Rhythmyx

Web Services Development Kit

Version 7.0.3

Printed on 5 June, 2009

Copyright and Licensing Statement

Licenses and Source Code

Rhythmyx uses Mozilla's JavaScript C API. See ***http://www.mozilla.org/source.html*** for the source code. In addition, see the Mozilla Public License (***http://www.mozilla.org/source.html***).

Netscape Public License

Apache Software License

IBM Public License

Lesser GNU Public License

Other Copyrights

Percussion Software
600 Unicorn Park Drive
Woburn, MA 01801 U.S.A.
781.438.9900
Internet E-Mail: technical_support@percussion.com
Website: http://www.percussion.com

Contents

CHAPTER 1

About Rhythmyx Web Services

Rhythmyx is an item-based (or modular) Content Management System for Web content. Rhythmyx manages Web content as discrete units, or Content Items. Content Items consist solely of raw data (either text or binary files). Formatting exists in separately in Templates defined using Apache Velocity. Users define Active Assembly Relationships between Content Items to form pages. Other Relationship Types are also available for such purposes as tracking translations and new versions of the Content Item.

When generating an output (publishing content), the raw Content Item data is merged with the formatting to produce the formatted pages. The related content is also merged into the output images during this process.

Rhythmyx provides a browser interface, Content Explorer, that allows users to interact with Content Items. Content Explorer also provides a Folder mechanism for organizing Content Items both for user convenience and to define the directory structure of the published output.

User access to Content Items is controlled in three ways. First, all Content Items exist in a Workflow, which defines the business process for creating and maintaining Content Items, publishing them, and archiving them. A user can only access a Content Item if the user is assigned to the current Workflow State of the Content Item. Second, all Content Items exist in a Community, which defines a set of Content Items and the users that have access to them. Users can only access a Content Item if they are currently logged in to the Community of that Content Item. Finally, Content Items exist in a Locale, which defines a language-region combination for Content Items. Similar to Communities, users can only access a Content Item if they are logged in to the Locale of the Content Item. Locales and Communities operate separately but in tandem. A user must be logged in to both the Community and Locale of a Content Item to access it. Users can change their Community and Locale using Content Explorer, but Workflow Roles are determined when the user logs in to the system.

For further detail about Rhythmyx functionality and concepts, see the *Rhythmyx Concepts Guide*.

Rhythmyx Web Services provide the ability to implement alternate clients to Content Explorer for the creation and maintenance of content, such as content loaders or custom user interfaces. Web Services clients can be used to:

- create and maintain Content Items;
- perform Workflow operations on Content Items;
- create Active Assembly Relationships and other types of Relationships between Content Items;
- assign Content Items to Folders and move Content Items from one Folder to another;
- log in and out of the system and change the users Community and Locale.

NOTE: Customers developing Web Services code in Java can use the pregenerated Web Services client .jar file (`<Rhythmyxroot>/WebServices/<version>/sample/loader/Java/build/dist/rxwebserviceclient.jar`). Customers developing Web Services code in other environments must generate proxy stubs for that environment from the Web Services WSDL files. These files can be accessed using the URL `http://localhost:9992/Rhythmyx/webservices/<webservicetype>SOAP?wsdl` where *localhost* is the name or IP address of the Rhythmyx server, *9992* is the Rhythmyx port, and *webservicetype* is the type of Web Service you want to include; options for webservicetype are *content*, *assembly*, *system*, *security*, and *ui*.

Operational Sequences

In some cases, a simple operation, such as creating Folder, may require the use of a single service. In many cases, however, completing an operation requires the use of multiple services in sequence. This section describes a few common sequences.

Logging in to Rhythmyx

The common sequence to log in to Rhythmyx is:

1. Establish a connection to the servlet container.
2. Log in to the Rhythmyx server.
3. Create proxies for the various service types.

Creating a Content Item

The minimal sequence to create a Content Item is:

1. Create the blank Content Item instance.
2. Add data to the Content Item fields.
3. Save the Content Item to the Repository

   Often the following additional processing is also performed:
4. Check in the Content Item.
5. Associate the Content Item with a Folder.

Creating a Content Item with Children

The minimal sequence to create a Content Item with children is:

1. Create a blank Content Item instance. This Content Item is the parent Content Item.
2. Add data to the fields of the parent Content Item.
3. Save the parent Content Item to the Repository, but do not check it in (leaving it still checked out to the user that created it).
4. Create child entries for the parent Content Item.
5. Add data to the child entries.
6. Save the child entries.
7. Check in the parent Content Item.

Modifying a Content Item

The basic sequence to modify a Content Item is:

1. Prepare the Content Item for editing. (If the Content Item is not already checked out, this request checks it out to the user.)
2. Load the Content Item.
3. Modify the Content Item data.
4. Save the modified Content Item.
5. Release the Content Item from edit. (Checks the Content Item in if it was checked out when it was prepared for edit.)

Modifying Child Content

Two options are available for modifying child content.

Modifying child content as part of the Content Item

Choose this option if you want to modify the data in the parent Content Item as well as the data in the children. The basic sequence to modify child content as part of a Content Item is:

1. Prepare the Content Item for editing. (If the Content Item is not already checked out, this request checks it out to the user.)
2. Load the Content Item, including the child content. You can choose to load only specific child Field Sets, or load all children.
3. Modify the Content Item data. These modifications may include adding or deleting child rows, as well as modifying child data.
4. Save the modified Content Item
5. Release the Content Item from edit.

Modifying only child content

Choose this option if you only want to modify the child content. The basic sequence to modify child content without loading and modifying the parent Content Item is:

1. Prepare the parent Content Item for editing. (If the Content Item is not already checked out, this request checks it out to the user).
2. To create new child entries:
   a) Create the new child entries.
   b) Add data to the new child entries.
   c) Save the new child entries
3. To modify existing child entries:
   a) Load the child entries you want to modify.

b) Change the data on the child entries.

c) Save the modified child entries.

4. Child entries can also be deleted.
5. When all modifications are complete, release the Content Item from edit. (Checks the Content Item in if it was checked out when it was prepared for edit.)

Assembling Content Items

The basic sequence for assembling Content Items is:

1. Prepare the Content Item for editing.
2. Load the Content Item.
3. Add Active Assembly Relationships.
4. Save the modified Content Item.
5. Release the Content Item from edit.

CHAPTER 2

Content Services

Rhythmyx Content web services provide functionality to access and manipulate Content Items and Folders. For convenience, these services are divided into the following groups:

- ***Item services*** (see page 18): services that allow the user to create, locate, modify, and delete Content Items;
- ***Item Children services*** (see page 43): services that allow the user to create, modify, rearrange, and delete child content;
- ***Related Item services*** (see page 50): services that allow the user to create, modify, and rearrange Active Assembly Relationships for a Content Item;
- ***Folder services*** (see page 60): services that allow users to create, move, and delete Folders, as well as to add Content Items to a Folder, move it between Folders, and rearrange it between Folders;
- ***Keyword services*** (see page 77): services that allow users to retrieve Keyword definitions;
- ***Localization services*** (see page 78): services used to retrieve localization data;
- ***Content Type services*** (see page 80): services used to retrieve data summaries about Content Types.

Item Services

Item services allow users to create, read, update, and delete Content Items, and to create Relationships of the default Relationship Types provided by Percussion Software.

CreateItem

Creates an array of new Content Items. When this service is submitted, the server constructs the new Content Items and returns them to the user with default values for any fields that have default values defined. The newly created Content Items are not added to the Repository until the SaveItems service is submitted for them.

Input Parameters

Parameter Name	Date Type	Description	Example
ContentType	String	Name of the Content Type for which you want to create the new Content Items. Must be an existing Content Type. Cannot be null.	Generic
Count	Integer	Optional. Specifies the number of new Content Items the user wants to create. If this parameter is not supplied, one new Content Item will be created. Must be greater than 0 if supplied.	3

Output

PSItem (see page 115): Array of new Content Items of the requested Content Type. Never null or empty. The returned Content Items are not saved to the Repository; to save the Content Items to the Repository, the user must submit the SaveItems service.

Faults

Fault Name	Returned
AxisFault	Any unexpected error.
PSContractViolationFault	If the value of the Count parameter is 0 or less than 0. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the user's current session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).

PSUnknownContentTypeFault	If the value submitted in the ContentType parameter does not exist. Returns an error message that the specified Content Type does not exist. Confirm that the Content Type specified in the request exists and is spelled correctly.
PSNotAuthorizedFault	If the requestor is not authorized to create Content Items of the specified Content Type. For more details see ***PSNotAuthorizedFault*** (on page 128)

FindItems

Finds all Content Items that match the criteria in the search specified in the PSSearch input parameter. Used to implement the functionality of either a Search or a View.

Input Parameters

Parameter Name	Date Type	Description	Example
PSSearch (see page 120)	String	The search used to perform the requested lookup. The value of this parameter can be either the name of a user-defined search saved in Content Explorer or a set of individual search parameters The value of this parameter cannot be null.	Saved user-defined search: MySavedSearch Set of Search Parameters: Community=EnterpriseInvestments, ContentType=Generic
LoadOperation	Boolean	If the value is specified as true, the PSSearchResults returned includes the set of available Actions that the user can take on the Content Items. (The list of available actions is filtered based on the user's Roles and Community. The default value is false, which specifies that the actions will not be returned.	false

Output

PSSearchResults: Array of search results of the Content Items that match the search criteria. Can never be null, but may be empty if no results are found that match the search criteria.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	If the value of the PSSearch parameter is null. For more details, see ***PSContractViolationFault*** (on page 126).

Fault Name	Returned
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).

LoadItems

Loads the requested Content Items using the options specified in the request.

Input Parameters

Parameter Name	Date Type	Description	Example
ID	Array	Array or IDs of the Content Items to load. Cannot be null or empty	27-5-198, 27-5-203, 27-5-497
FieldName	Array	Array of field names of fields to be included with the returned Content Items.	sys_Title, Callout, Body
IncludeBinary	Boolean	Flag specifying whether to include binary fields with the returned Content Items. Inclusion of this flag in a request is only valid if the FieldName parameter has been included in the request with at least one value. Defaults to False	False
IncludeChildren	Boolean	Flag specifying whether to include child content with the returned Content Items. If not included in the request, defaults to False.	False
ChildNames	Array	Fieldset names for all child field sets to load with the requested Content Item. If null or empty and the IncludeChildren flag is True, all child fieldsets are loaded. If no fieldset exists for a specified name, a PSContractViolationFault will be returned for the Content Item. If a requested fieldset has no data, the PSItem object returned for that Content Item will not include the PSChild node. If the value of the IncludeChildren flag is False, this parameter is ignored.	EventDate

Parameter Name	Date Type	Description	Example
IncludeRelated	Boolean	Flag specifying whether to include related Content Items with the returned Content Item. If not included in the request, defaults to False	False
Slot	Array	Array of names of Slots for which to load related Content Items. If null or empty and the IncludeRelated flag is set to True, all related Content Items are included for all Slots. If no Slot exists for a specified name, a PSContractViolationFault will be returned for the Content Item. If a requested Slot has no data, the PSItem object returned for that Content Item will not include the PSRelatedContent node. If the value of the IncludeRelated flag is false, this parameter is ignored.	Sidebar_Slot, Latest_News_Slot
IncludeFolderPath	Boolean	Flag specifying whether to include the Folder paths to all Folders that contain the requested Content Item. If not included in the request, defaults to False.	False

Output

PSItem (see page 115): Array of Content Items with the specified details, listed in the order that their IDs are specified in the ID parameter. The output is never null or empty. All returned Content Items are in read-only mode. Submit the ***PrepareForEdit service*** (see page 27) to make the Content Items editable.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned in the following circumstances: ▪ If the value of the ID parameter is empty or null, or if a specified Id does not exist in the system. ▪ If no fieldset exists in the Content Item with the fieldset name specified. ▪ If a no Slot exists on the Content Item with the name specified. For more details, see ***PSContractViolationFault*** (on page 126).

Fault Name	Returned
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSErrorResultsFault	If any requested Content Item fails to load. For details, see ***PSErrorResultsFault*** (on page 127)

SaveItems

Inserts or updates all the specified Content Items to the Repository.

All Content Items submitted with the request that do not include a valid Content Item ID for an existing Content Item will be inserted into the repository. All other Content Items will be updated.

When inserting Content Items, any child content, Active Assembly Relationships, or Folders included with a supplied Content Item will be created. To manipulate child content, use the ***Item Children services*** (see page 43). To manipulate Active Assembly Relationships, use the ***Related Item services*** (see page 50). To manipulate Folders, use the ***Folders services*** (see page 60).

When updating Content Items:

- If no child content is included with a Content Item being updated, any existing child content of that Content Item will be untouched.
- If child content is included with a Content Item being updated:
  - If new child content is included, new child entries will be added to the child tables in the Repository
  - If any existing child entries are included, those child entries will be updated with the data from the request.
  - Any entries missing from the request will be deleted from the Repository. Note that this behavior means that if you provide any child content with an update, you must provide ALL child content to prevent any child content from being deleted.
- If no Active Assembly Relationships are included for a Content Item being updated, any existing Active Assembly Relationships will be untouched.
- If Active Assembly Relationships are included for a Content Item being updated:
  - If a new Active Assembly Relationship is included, the new Relationship will be added to the Repository
  - If any existing Active Assembly Relationships are included, those Relationships will be updated with the data from the request;
  - Any Active Assembly Relationships missing from the request will be deleted from the Repository. Note that this behavior means that if you provide any Active Assembly Relationships with an update, you must include ALL Active Assembly Relationships to prevent any existing Active Assembly Relationships from being deleted.

- If no Folder path is included for a Content Item being updated, any existing Folder paths remain untouched.
- If Folder paths are included for a Content Item being updated:
  - If a new Folder path is included in the request, the new path will be added to the Repository;
  - If an existing Folder path is included in the request, it will be retained in its current condition;
  - Any Folder path that is missing from the request will be deleted from the Repository. Note that this behavior means that if you provide any Folder paths with an update, you must include ALL Folder paths to prevent any existing Folder paths from being deleted.

Input Parameters

Parameter Name	Date Type	Description	Example
PSItem (see page 115)	Array	Array of Content Items to be inserted or updated in the Repository. Cannot be null or empty.	
EnableRevisions	Boolean	Flag specifying whether to turn on the Revision Lock for new Content Items immediately. When the Revision Lock is turned on, each time a Content Item is checked out, a new Revision of that Content Item is created. (Note: Typically, the Revision Lock is turned on only after a Content Item goes Public). Defaults to False if not supplied. Note: This flag is ignored for Content Items being updated.	False
CheckIn	Boolean	Flag specifying whether to check in all Content Items in the request immediately after saving the changes. Defaults to False if not included in the request.	False

Output

Id: Array of IDs of the Content Items inserted or updated in the Repository. The order of IDs returned is the same order in which the Content Items were specified in the request.

Faults

Fault Name	Returned
AxisFault	Any unexpected error

Fault Name	Returned
PSContractViolationFault	If the PSItem parameter is null or empty. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSErrorResultsFault	If any submitted Content Item cannot be inserted. For details, see ***PSErrorResultsFault*** (on page 127)

DeleteItems

Deletes from the Repository all Content Items that match the IDs specified in the ID parameter.

This operation cannot be reversed.

All Content Items specified must be checked out to the requestor. All Content Items must also be in Edit mode (in other words, ***Prepare for Edit*** (see page 27) must have been submitted for all Content Items in the request without receiving an error result).

Input Parameters

Parameter Name	Date Type	Description	Example
ID	Array	Array of IDs of Content Items to delete from the Repository. If a specified ID does not exist, it is ignored. Cannot be null or empty.	27-5-189

Output

None

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	If the ID parameter is null or empty. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSErrorsFault	If any specified Content Item has not been prepared for edit, or cannot be deleted for some other reason. For more details, see ***PSErrorsFault*** (on page 127)

ViewItems

Returns all Content Items specified by the submitted IDs, using the specified loading options. All returned Content Items are read-only.

The behavior of this service is similar to the behavior of the ***Load Items*** (see page 20) service. The difference is that the ViewItems services allows you to specify the Revision of the Content Items you want to return by specifying a Revision-specific ID in the Id parameter. If the value of the Id parameter is a non-Revision-specific ID, the Current Revision of the specified Content Item will be returned. The ***PrepareForEdit*** (see page 27) service can only be submitted for the Current Revision. Before any other Revision can be edited, the user must submit the ***PromoteRevision*** (see page 38) service for it.

Input Parameters

Parameter Name	Date Type	Description	Example
Id	Array	Array of the IDs of the Content Items to load. Cannot be null or empty. If a specified Content Item ID does not exist, a PSContractViolationFault will be returned. All IDs can include a RevisionID. If this value is included, the specified Revision will be returned. If no RevisionId is specified, then the Current Revision is returned.	27-5-198, 27-5-203, 27-5-497
FieldName	Array	Array of field names to be included with the returned Content Items. If this parameter is not included, all fields in all Content Items will be returned. If a specified field does not exist, a PSContractViolationFault will be returned.	sys_Title, Body, Callout
IncludeBinary	Boolean	Flag specifying whether to include binary fields with the returned Content Items. Use of this flag is only valid if the FieldName parameter has been included with at least one value. If not included in the request, defaults to False	False
IncludeChildren	Boolean	Flag specifying whether to include child content with the returned Content Items. If not included in the request, defaults to False.	False

Parameter Name	Date Type	Description	Example
ChildNames	Array	Fieldset names for all child field sets to load with the requested Content Item. If null or empty, all child fieldsets are loaded. If no fieldset exists for a specified name, a PSContractViolationFault will be returned for the Content Item. If a requested fieldset has no data, the PSItem object returned for that Content Item will not include the PSChild node. If the value of the IncludeChildren flag is False, this parameter is ignored.	EventDate
IncludeRelated	Boolean	Flag specifying whether to include related Content Items with the returned Content Item. If not included in the request, defaults to False	False
Slot	Array	Array of names of Slots for which to load related Content Items. If null or empty, related Content Items are included for all Slots. If no Slot exists for a specified name, a PSContractViolationFault will be returned for the Content Item. If a requested Slot has no data, the PSItem object returned for that Content Item will not include the PSRelatedContent node. If the value of the IncludeRelated flag is false, this parameter is ignored.	Sidebar_Slot, Latest_News_Slot
IncludeFolderPath	Boolean	Flag specifying whether to include the Folder paths to all Folders that contain the requested Content Item. If not included in the request, defaults to False.	False

Output

PSItem (see page 115): Array of Content Items with the specified details, listed in the order of their IDs in the ID parameter. The output is never null or empty. All returned Content Items are in read-only mode. Content Items returned using this service cannot be made editable.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned in the following circumstances: ▪ If the value of the ID parameter is empty or null, or if a specified Id does not exist in the system. ▪ If no fieldset exists in the Content Item with the fieldset name specified. ▪ If a no Slot exists on the Content Item with the name specified. For more details, see ***PSContractViolationFault*** (on page 126)
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSErrorResultsFault	If any submitted Content Item cannot be inserted. For details, see ***PSErrorResultsFault*** (on page 127)

PrepareForEdit

Prepares the Content Items specified by the IDs submitted with the request for editing by the user issuing the request.

A user may resubmit this request multiple times for the same Content Item without submitting a ***ReleaseFromEdit*** (see page 28) request for that Content Item in between PrepareForEdit requests, but the user's client machine is responsible for maintaining the PSItemStatus returned the first time the call is submitted.

Input Parameters

Parameter Name	Date Type	Description	Example
Id	Array	Array of IDs of Content Items to be prepared for edit. Cannot be null or empty.	27-5-198, 27-5-203, 27-5-497

Output

PSItemStatus (see page 115): Array of PSItemStatus responses in the order of the IDs specified in the request. Stores the status data of the item before it was prepared for edit. Use this data to call the ReleaseFromEdit service.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned in the following circumstances: ▪ If the value of the Id parameter is empty or null, or if a specified ID does not exist in the system. ▪ If the Workflow requires a comment on checkout but the submitted request does not include the Comment parameter with valid string value. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSErrorResultsFault	If any requested Content Item cannot be prepared for edit. For details, see ***PSErrorResultsFault*** (on page 127)

ReleaseFromEdit

Releases the Content Items included in the PSItemStatus parameter from edit mode. If a specified Content Item is not in edit mode and checked out to the user, a PSContractViolationFault will be returned.

Input Parameters

Parameter Name	Date Type	Description	Example
PSItemStatus (see page 115)	Array	Array of PSItemStatus data, one for each Content Item that should be released from edit mode. Cannot be null or empty. For each Content Item, should be the original PSItemStatus returned when the first PrepareForEdit was submitted.	
CheckInOnly	Boolean	Flag specifying whether the ReleaseFromEdit service should simply check in the Content Item, or should attempt to execute the Transition behavior of ***PSItemStatus*** (see page 115). If not included in the request, defaults to False.	False

Output

None

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned in the following circumstances: ▪ If the value of the PSItmeStatus parameter is empty or null. ▪ If a specified Content Item is not in edit mode and checked out to the user submitting the request. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSErrorResultsFault	If any submitted Content Item cannot be inserted. For details, see ***PSErrorResultsFault*** (on page 127)

NewCopies

Creates a New Copy of each Content Item specified by the ID parameter. A New Copy is the simplest form of cloned Content Item created in Rhythmyx. A New Copy is a clone of a Content Item that does not involve any additional processing. Rhythmyx maintains a New Copy Relationship between the Owner and the Dependent to facilitate tracking of the two Content Items. For more details, see the *Rhythmyx Concepts Guide* or *Implementing the Relationship Engine*.

The new Content Items are inserted into the Repository and returned to the requestor in read-only mode.

The Path parameter can specify either a single path or one path for each ID specified in the ID parameter. (If multiple paths are specified but the number of paths specified does not match the number of IDs specified, a PSContractViolationFault will be returned.) If a single path is specified, all newly created Content Items will be added to the specified Folder. If multiple paths are specified, they are applied to the Content Items in the order specified. Thus, if we have the following parameters:

```
ID="27-5-189, 27-5-198, 27-5-203, 27-5-497"
Path="//Sites/EnterpriseInvestments/InvestmentAdvice/EstatePlanning,//Si
tes/EnterpriseInvestments/InvestmentAdvice/Retirement,
//Sites/EnterpriseInvestments/InvestmentAdvice/Retirement,
//Sitess/EnterpriseInvestments/InvestmentAdvice/Tax
```

The New Copies would be created in the following locations:

The New Copy of this Content Item	Would be created in this location
27-5-189	//Sites/EnterpriseInvestments/InvestmentAdvice/EstatePlanning
27-5-198	//Sites/EnterpriseInvestments/InvestmentAdvice/Retirement
27-5-203	//Sites/EnterpriseInvestments/InvestmentAdvice/Retirement

The New Copy of this Content Item	Would be created in this location
27-5-497	//Sites/EnterpriseInvestments/InvestmentAdvice/Tax

Whenever a New Copy Content Item is created using this service, Rhythmyx automatically creates a Relationship in the New Copy category. The Content Item specified in the ID parameter is the Owner in the new Relationship. The New Copy Content Item is the Dependent.

Input Parameters

Parameter Name	Date Type	Description	Example
ID	Array	Array of ID specifying the Content Items to be copied. Cannot be null or empty. If no Content Item exists with a specified ID, a PSContractViolationError is returned. If the same ID is included multiple times in a request, multiple copies will be created (on copy for each instance of the same Content ID).	27-5-189, 27-5-198, 27-5-203, 27-5-497
Path	Array	Array of Folder paths containing either a single path or one path for each Content Item specified in the ID parameter, specifying the Folder in which each New Copy Content Item should be created. If any specified Folder does not exist, it will be created. Cannot be null or empty. This array must contain the same number of entries as the ID parameter array.	//Sites/EnterpriseInvestments/InvestmentAdvice/EstatePlanning,//Sites/EnterpriseInvestments/InvestmentAdvice/Retirement, //Sites/EnterpriseInvestments/InvestmentAdvice/Retirement, //Sites/EnterpriseInvestments/InvestmentAdvice/Tax
Type	String	Name of the Relationship Type to use when creating the Relationship between the Content Item submitted with the request and the New Copy created. The Relationship Type specified must be in the New Copy Category. Defaults to the System/New Copy Relationship if not included in the request.	System/New Copy

Parameter Name	Date Type	Description	Example
EnableRevisions	Boolean	Flag specifying whether to turn on the Revision Lock for new Content Items immediately. When the Revision Lock is turned on, each time a Content Item is checked out, a new Revision of that Content Item is created. (Note: Typically, the Revision Lock is turned on only after a Content Item goes Public). Defaults to False if not included in the request.	False

Output

PSItem (see page 115): Array of New Copy Content Items. All new Content Items created as a result of this request are inserted into the Repository. The output is never null or empty. All returned Content Items are in read-only mode. To modify the newly created Content Items, the user must submit a PrepareForEdit request, and then must submit a SaveItems request to save any changes.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned in the following circumstances: ▪ If the value of the ID parameter is empty or null, or if a specified Id does not exist in the system. ▪ If the Path parameter is null or empty. ▪ If the Path parameter contains more than one path but the number of entries in the Path array does not correspond to the number of entries in the ID array. ▪ If the value of the Type parameter is not a valid Relationship in the New Copy Category. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSErrorResultsFault	If a New Copy cannot be created for any ID submitted. For details, see ***PSErrorResultsFault*** (on page 127)

NewPromotableVersions

Creates a new PromotableVersion Content Item. A Promotable Version is a cloned Content Item that supersedes the original when the clone becomes Public. The original Content Item is automatically Transitioned to an Archived State. Rhythmyx maintains a PromotableVersion Relationship between the two Content Items to facilitate this processing. For more details, see the *Rhythmyx Concepts Guide* or *Implementing the Relationship Engine*.

The new Content Items are inserted into the Repository and returned to the requestor in read-only mode.

The Path parameter can specify either a single path or one path for each ID specified in the ID parameter. (If multiple paths are specified but the number of paths specified does not match the number of IDs specified, a PSContractViolationFault will be returned.) If a single path is specified, all newly created Content Items will be added to the specified Folder. If multiple paths are specified, they are applied to the Content Items in the order specified. Thus, if we have the following parameters:

```
ID="27-5-189, 27-5-198, 27-5-203, 27-5-497"
Path="//Sites/EnterpriseInvestments/InvestmentAdvice/EstatePlanning,//Si
tes/EnterpriseInvestments/InvestmentAdvice/Retirement,
//Sites/EnterpriseInvestments/InvestmentAdvice/Retirement,
//Sites/EnterpriseInvestments/InvestmentAdvice/Tax
```

The New PromotableVersion Content Items would be created in the following locations:

The New Copy of this Content Item	Would be created in this location
27-5-189	//Sites/EnterpriseInvestments/InvestmentAdvice/EstatePlanning
27-5-198	//Sites/EnterpriseInvestments/InvestmentAdvice/Retirement
27-5-203	//Sites/EnterpriseInvestments/InvestmentAdvice/Retirement
27-5-497	//Sites/EnterpriseInvestments/InvestmentAdvice/Tax

Whenever a new PromotableVersion Content Item is created using this service, Rhythmyx automatically creates a Relationship in the Promotable category. The Content Item specified in the ID parameter is the Owner in the new Relationship. The New PromotableVersion Content Item is the Dependent.

Input Parameters

Parameter Name	Date Type	Description	Example
ID	Array	Array of ID specifying the Content Items to be copied. Cannot be null or empty. If no Content Item exists with a specified ID, a PSContractViolationError is returned. If the same ID is included multiple times in a request, multiple copies will be created (on copy for each instance of the same Content ID).	27-5-189, 27-5-198, 27-5-203, 27-5-497

Parameter Name	Date Type	Description	Example
Path	Array	Array of Folder paths containing either a single path or one path for each Content Item specified in the ID parameter, specifying the Folder in which each New PromotableVersion Content Item should be created. If any specified Folder does not exist, it will be created. Cannot be null or empty. This array must contain the same number of entries as the ID parameter array.	//Sites/EnterpriseInvestments/InvestmentAdvice/EstatePlanning,//Sites/EnterpriseInvestments/InvestmentAdvice/Retirement, //Sites/EnterpriseInvestments/InvestmentAdvice/Retirement, //Sites/EnterpriseInvestments/InvestmentAdvice/Tax
Type	String	Name of the Relationship Type to use when creating the Relationship between the Content Item submitted with the request and the New PromotableVersion Content Item created. The specified Relationship Type must be in the Promotable Version Category. Defaults to the System/PromotableVersion Relationship if not included in the request.	System/PromotableVersion
EnableRevisions	Boolean	Flag specifying whether to turn on the Revision Lock for new Content Items immediately. When the Revision Lock is turned on, each time a Content Item is checked out, a new Revision of that Content Item is created. (Note: Typically, the Revision Lock is turned on only after a Content Item goes Public). Defaults to False if not supplied.	False

Output

PSItem (see page 115): Array of New PromotableVersion Content Items. All new Content Items created as a result of this request are inserted into the Repository. The output is never null or empty. All returned Content Items are in read-only mode. To modify the newly created Content Items, the user must submit a PrepareForEdit request, and then must submit a SaveItems request to save any changes.

Faults

Fault Name	Returned
AxisFault	Any unexpected error

Fault Name	Returned
PSContractViolationFault	Returned in the following circumstances: ▪ If the value of the ID parameter is empty or null, or if a specified ID does not exist in the system. ▪ If the Path parameter is null or empty. ▪ If the Path parameter contains more than one path but the number of entries in the Path array does not correspond to the number of entries in the ID array. ▪ If the value of the Type parameter is not a valid Relationship in the Promotable Version Category.. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSErrorResultsFault	If a New Copy cannot be created for any ID submitted. For details, see ***PSErrorResultsFault*** (on page 127)

NewTranslations

Creates a new Translation Content Item of each Content Item specified by the ID parameter for each Locale specified in the PSAutoTranslation parameter. (NOTE: If this parameter is not included, or is null or empty, Translation Content Items are created in all Locales.) A Translation Content Item is a clone of a Content Item that is intended to be Translated into a different language (Rhythmyx does not perform the translation automatically. Rhythmyx maintains either a Translation Relationship or a Translation - Mandatory Relationship between the two Content Items. A Translation Relationship allows tracking of the two Content Items, while a Translation - Mandatory Relationship includes processing that prevents either Content Item from going Public until both are ready to be Public. For more details, see the *Rhythmyx Concepts Guide* or *Implementing the Relationship Engine*.

Note: If a Translation Content Item of a specified Content Item already exists in a specified Locale, that portion of the request will be ignored.

The new Content Items are inserted into the Repository and returned to the requestor in read-only mode.

Whenever a New Translation Content Item is created using this service, Rhythmyx automatically creates a Relationship in the Translation category. Currently, two Relationships are available in this category: Translation and Translation - Mandatory. The Content Item specified in the ID parameter is the Owner in the new Relationship. The New Translation Content Item is the Dependent.

Input Parameters

Parameter Name	Date Type	Description	Example
ID	Array	Array of ID specifying the Content Items for which to create Translation Content Items.. Cannot be null or empty. If no Content Item exists with a specified ID, a PSContractViolationError is returned.	27-5-189, 27-5-198, 27-5-203, 27-5-497
PSAutoTranslation (see page 113)	Array	Array of Automated Translation definitions to use when creating the Translation Content Items. If not specified, null or empty, Translation Content Items are created in all Locales.	
Type	String	Name of the Relationship Type to use when creating the Relationship between the Content Item submitted with the request and the New Translation Content Item created. The specified Relationship must be in the Translation Category. Defaults to the System/Translation Relationship if not included in the request.	System/Translation

Parameter Name	Date Type	Description	Example
EnableRevisions	Boolean	Flag specifying whether to turn on the Revision Lock for new Content Items immediately. When the Revision Lock is turned on, each time a Content Item is checked out, a new Revision of that Content Item is created. (Note: Typically, the Revision Lock is turned on only after a Content Item goes Public). Defaults to False if not supplied.	False

Output

PSItem (see page 115): Array of all specified Translation Content Items (including any Translation Content Items created by an earlier call). In the array, Content Items are ordered by Locale as specified in the PSAutoTranslation parameter, and within each Locale, in the order the Content Items were specified in the ID parameter. For example, assume the example specified were input, and that the PSAutoTranslation specified the Locales de-de, fr-fr. The array returned would be:

- de-de Translation of 27-5-189
- de-de Translation of 27-5-198
- de-de Translation of 27-5-203
- de-de Translation of 27-5-497
- fr-fr Translation of 27-5-189
- fr-fr Translation of 27-5-198
- fr-fr Translation of 27-5-203
- fr-fr Translation of 27-5-497

All new Content Items created as a result of this request are inserted into the Repository. The output is never null or empty. All returned Content Items are in read-only mode. To modify the newly created Content Items, the user must submit a PrepareForEdit request, and then submit a SaveItems request to save any changes.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned in the following circumstances: ▪ If the value of the ID parameter is empty or null, or if a specified Id does not exist in the system. ▪ If an automated translation configuration specified in the PSAutoTranslation parameter does not exist. ▪ If the value of the Type parameter is not a valid Relationship in the Translation Category. For more details, see ***PSContractViolationFault*** (on page 126).

Fault Name	Returned
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSErrorResultsFault	If a Translation Content cannot be created for any combination of ID and autotranslation configuration submitted for some reason other than that the requested Translation Content Item already exists. For details, see ***PSErrorResultsFault*** (on page 127)

FindRevisions

Finds all Revisions of the Content Items specified by the ID parameter.

Input Parameters

Parameter Name	Date Type	Description	Example
ID	Array	Specifies the Content Items for which to retrieve Revisions. Cannot be null or empty. If a specified ID does not exist, a PSContractViolationFault is returned.	<set of long values>

Output

PSRevisions (see page 120): Array of Revisions of the specified Content Items. Never null or empty.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	If a specified ID does not exist. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).

PromoteRevisions

Promotes each of the specified Content Item Revisions, making each the current Revision. All new Revision Content Items remain checked out to the user that issued the request. The user can thus modify these Revisions immediately, but must check in the new Revision Content Items manually.

Use the ***FindRevisions service*** (see page 37) to retrieve the Revisions-specific IDs.

Input Parameters

Parameter Name	Date Type	Description	Example
ID	Array	Array of Revision-specific Content Item IDs. For details about revision-specific Content Item IDs, see the ID data type. Cannot be null or empty. If a specified ID does not exist, a PSContractViolationFault will be returned. All Content Items must be checked in to promote a Revision.	27-5-198, 27-5-203, 27-5-497

Output

None

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	If the ID parameter is null or empty, or if a specified ID does not exist. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSErrorsFault	If any Revision cannot be promoted. For more details, see ***PSErrorsFault*** (on page 127).

CheckinItems

Checks in all Content Items specified by the ID parameter. If any specified Content Item is already checked in, the request is ignored for that Content Item.

Input Parameters

Parameter Name	Date Type	Description	Example
ID	Array	Array of IDs of Content Items to be checked in. Cannot be null or empty. If any specified Content Item does not exist, a PSContractViolationFault will be returned.	27-5-198, 27-5-203, 27-5-497
Comment	String	Comment in the checkin of the Content Item. This parameter is optional if comments are optional in the Workflow of the requested Content Item. If the Workflow requires comments, this parameter is required.	

Output

None

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned in the following circumstances: ▪ If the value of the Id parameter is empty or null, or if a specified ID does not exist in the system. ▪ If the Workflow requires a comment on checkout but the submitted request does not include the Comment parameter with valid string value. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSErrorsFault	If any requested Content Item cannot be checked in. For more details, see ***PSErrorsFault*** (on page 127).

CheckoutItems

Checks out all Content Items specified by the ID parameter. If a user attempts to check out a Content Item that is already checked out to them, the request is ignored. If a Content Item is already checked out to another user, a PSErrorsFault is returned.

Input Parameters

Parameter Name	Date Type	Description	Example
ID	Array	Array of IDs of Content Items to be checked out. Cannot be null or empty. If any specified Content Item does not exist, a PSContractViolationFault will be returned. If any specified Content Item is already checked out to another user, a PSErrorsFault will be returned.	27-5-198, 27-5-203, 27-5-497
Comment	String	Comment in the checkout of the Content Item. This parameter is optional if comments are optional in the Workflow of the requested Content Item. If the Workflow requires comments, this parameter is required.	

Output

None

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned in the following circumstances: ▪ If the value of the Id parameter is empty or null, or if a specified ID does not exist in the system. ▪ If the Workflow requires a comment on checkout but the submitted request does not include the Comment parameter with valid string value. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).

Fault Name	Returned
PSErrorsFault	If any specified Content Item is already checked out to another user, or cannot be checked out for some other reason. For more details, see ***PSErrorsFault*** (on page 127).

GetAssemblyUrls

Constructs an assembly URL for the specified Content Items using the included URL parameters. Executing the resulting URL generates a Preview of the specified Content Item.

Input Parameters

Parameter Name	Date Type	Description	Example
ID	Array	Array of IDs of Content Items for which to construct assembly URL. May be revision-specific IDs or non-Revision-specific IDs. If a Revision-specific ID is submitted, the assembly URL will be constructed for that Revision. If a non-Revision-specific URL is submitted, the assembly URL will be constructed for the Current Revision. Cannot be null or empty. If a specified Content Item does not exist, a PSContractVIolationFault is returned.	27-5-198, 27-5-203, 27-5-497
Template	String	Name of the Template to use when constructing the Assembly URL. Cannot be null or empty. If the specified Template does not exist, a PSContractViolationFault is returned.	P_EI_Generic
Context	Integer	The context for which to construct the assembly URL. Cannot be null or empty	1
Filter	String	The Item Filter to use when constructing the assembly URL. Cannot be null or empty.	
Site	ID	Name of the Site for which to construct the assembly URL. Cannot be null or empty.	27-8-314

Parameter Name	Date Type	Description	Example
FolderPath	String	The path to the Folder to use when constructing the assembly URL. Cannot be null or empty.	//Sites/EnterpriseInvestments/InvestmentAdvice/Retirement

Output

Url: Array of assembly URLs constructed using the data specified in the request. Never null or empty.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned in the following circumstances: ▪ If the value of any of the following parameters is null or empty: ▪ ID ▪ Template ▪ Context ▪ Site ▪ Filter ▪ FolderPath ▪ If the value of any of the following parameters does not exist in the system: ▪ ID ▪ Template ▪ Context ▪ Site For more details, see ***PSContractViolationFault*** (on page 126)
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).

Item Children Services

Item Children services allow users to create, read, update, and delete child entries on Content Items.

CreateChildEntries

Creates the specified number of child entries on the parent Content Item and saves them in the Repository. All entries are created with default data and returned to the user.

A user can only submit this service if the ***PrepareForEdit service*** (see page 27) has already been submitted for the parent Content Item.

Input Parameters

Parameter Name	Date Type	Description	Example
ID	ID	Id of the Content Item for which to create the new child entries. Cannot be null or empty. If the value of this parameter is not an existing ID, a PSContractViolationFault will be returned	27-5-189
Name	String	Name of the child fieldset in which to create the new child entries. Cannot be null or empty. If the value of this parameter is not the name of a fieldset of the Content Type of the parent Content Item, a PSUnknonwnChildFault will be returned.	Contact
Count	Integer	Specifies the number of new child entries to create. Defaults to 1 if this parameter is not included in the request. If the value of this parameter is 0 or less than 0, a PSContractViolationFault will be returned.	3

Output

PSChildEntry (see page 113): Array of new child entries for the specified Content Item, with default data. Never null or empty. The new entries are always saved to the Repository..

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned in the following circumstances: ▪ If the value of the ID parameter is empty or null, or if a specified Id does not exist in the system. ▪ If the value of the Count parameter is 0 or less than 0. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSUnknownChildFault	If the value submitted in the Name parameter is not a fieldset defined for the Content Type of the parent Content Item. For more details, see ***PSUnknownChildFault.*** (see "PSUnknownChildFault" on page 129)

LoadChildEntries

Loads all child entries of the specified parent Content Item in the specified fieldset.

A user can only submit this service if the ***PrepareForEdit service*** (see page 27) has already been submitted for the parent Content Item.

Input Parameters

Parameter Name	Date Type	Description	Example
ID	ID	Id of the Content Item for which to load the child entries. Cannot be null or empty. If the value of this parameter is not an existing ID, a PSContractViolationFault will be returned	27-5-189
Name	String	Name of the child fieldset whose entries will be loaded. Cannot be null or empty. If the value of this parameter is not the name of a fieldset of the Content Type of the parent Content Item, a PSUnknonwChildFault will be returned.	Contact

Parameter Name	Date Type	Description	Example
IncludeBinary	Boolean	Flag specifying whether to include binary fields with the returned child fieldset. Defaults to false if not included in the request.	False

Output

PSChildEntry (see page 113): Array of all loaded child entries for the specified fieldset on the specified Content Item. Never null or empty. For additional details, see the PSChild Data Type.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned if the value of the ID parameter is empty or null, or if a specified Id does not exist in the system. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSUnknownChildFault	If the value submitted in the Name parameter is not a fieldset defined for the Content Type of the parent Content Item. For more details, see ***PSUnknownChildFault.*** (see "PSUnknownChildFault" on page 129)

SaveChildEntries

Saves all child entries included with the request to the specified fieldset of the specified Content Item.

A user can only submit this service if the ***PrepareForEdit*** (see page 27) service has already been submitted for the parent Content Item.

Input Parameters

Parameter Name	Date Type	Description	Example
ID	ID	Id of the Content Item for which to save the child entries. Cannot be null or empty. If the value of this parameter is not an existing ID, a PSContractViolationFault will be returned	27-5-189

Name	String	Name of the child fieldset to which to save the child entries. Cannot be null or empty. If the value of this parameter is not the name of a fieldset of the Content Type of the parent Content Item, a PSUnknonwnChildFault will be returned.	Contact
PSChildEntry	Array	Array of the child entries to be saved. Cannot be null or empty. All submitted entries must be valid child field entries for the specified child fieldset of the Content Type of the specified parent Content Item. For additional details, see the PSChild Data Type.	

Output

None

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned in the following circumstances: ▪ If the value of the ID parameter is empty or null, or if a specified Id does not exist in the system. ▪ If the value of the PSChildEntry parameter is null or empty. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSUnknownChildFault	If the value submitted in the Name parameter is not a fieldset defined for the Content Type of the parent Content Item. For more details, see ***PSUnknownChildFault.*** (see "PSUnknownChildFault" on page 129)
PSErrorsFault	If any element of the array submitted in the PSChildEntry parameter does not conform to the requirements of the specified child fieldset of the Content Type of the specified parent Content Item, or if any submitted element of the array cannot be saved for some other reason. For more details, see ***PSErrorsFault*** (on page 127).

DeleteChildEntries

Deletes the specified child entries from the specified parent Content Item and child fieldset. This action is permanent and cannot be reversed.

A user can only submit this service if the ***PrepareForEdit service*** (see page 27) has already been submitted for the parent Content Item.

Input Parameters

Parameter Name	Date Type	Description	Example
ID	ID	Id of the Content Item from which to delete the child entries. Cannot be null or empty. If the value of this parameter is not an existing ID, a PSContractViolationFault will be returned	27-5-189
Name	String	Name of the child fieldset from which to delete the child entries. Cannot be null or empty. If the value of this parameter is not the name of a fieldset of the Content Type of the parent Content Item, a PSUnknonwnChildFault will be returned.	Contact
ChildId	Array	Array of IDs of child entries to be deleted. Cannot be null or empty. If a specified ChildId does not exist, it is ignored.	3

Output

None

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned if the value of the ID or ChildId parameter is empty or null, or if a specified Id or ChildId does not exist in the system. For more details, see ***PSContractViolationFault*** (on page 126).

PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSUnknownChildFault	If the value submitted in the Name parameter is not a fieldset defined for the Content Type of the parent Content Item. For more details, see ***PSUnknownChildFault.*** (on page 129)
PSErrorsFault	If any submitted element of the array cannot be deleted. For more details, see ***PSErrorsFault*** (on page 127).

ReorderChildEntries

Reorders all child entries from the specified parent Content Item in the specified child fieldset in the order requested.

A user can only submit this service if the ***PrepareForEdit service*** (see page 27) has already been submitted for the parent Content Item.

Input Parameters

Parameter Name	Date Type	Description	Example
ID	ID	Id of the Content Item for which to reorder the child entries. Cannot be null or empty. If the value of this parameter is not an existing ID, a PSContractViolationFault will be returned	27-5-189
Name	String	Name of the child fieldset for which to reorder the child entries. Cannot be null or empty. If the value of this parameter is not the name of a fieldset of the Content Type of the parent Content Item, a PSUnknonwnChildFault will be returned. If the specified child fieldset does not support reordering, a PSErrorsFault will be returned.	Contact

Parameter Name	Date Type	Description	Example
ChildId	Array	Array of IDs of all child entries of the specified fieldset in the new order. The sortrank of the specified entries starts at 0 for the first specified child and is incremented by 1 for each additional child ID. Any child entry that currently exists in the specified child fieldset on the specified parent Content Item that are not included in this array will be appended to the end of the fieldset in their current order. Cannot be null or empty. If a specified ChildId does not exist, it is ignored.	3,6,2,4,1,5

Output

None

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned if the value of the ID or ChildId parameter is empty or null, or if a specified Id or ChildId does not exist in the system. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSUnknownChildFault	If the value submitted in the Name parameter is not a fieldset defined for the Content Type of the parent Content Item. For more details, see ***PSUnknownChildFault.*** (see "PSUnknownChildFault" on page 129)
PSErrorsFault	If the specified child fieldset does not support sorting. For more details, see ***PSErrorsFault*** (on page 127).

Related Item Services

Related Item services are services that operate on ActiveAssembly Relationships between Content Items. Each request operates on the Relationships owned by one Content Item

Related Item services fall into two groups:

- Services that allow users to operate on the ActiveAssembly Relationships owned by a Content Item. In addition to allowing the user to create, read, update, and delete ActiveAssembly Relationships, these service also allow users to change the Template used in an ActiveAssembly Relationship, modify the order of Content Items in a Slot, or move Content Items from one Slot to another. Services in this group include:
  - ***AddContentRelations*** (see page 50)
  - ***LoadContentRelations*** (see page 52)
  - ***SaveContentRelations*** (see page 54)
  - ***DeleteContentRelations*** (see page 56)
  - ***ReorderContentRelations*** (see page 55)
- Services that allow users to navigate Active Assembly Relationship trees. Services in this group include:
  - ***FindDependents*** (see page 57)
  - ***FindOwners*** (see page 58)

AddContentRelations

Creates new ActiveAssembly Relationships between the Content Item specified in the ID parameter (Owner) and the Content Items specified in the RelatedId parameter (Dependents). The new ActiveAssembly Relationships are added to the Slot at the sortrank position specified by the Index parameter (indexing is 0-based).

This service cannot be run unless the PrepareForEdit service has been submitted for all Owner Content Items of the specified Relationships.

The Relationship Type of the ActiveAssembly Relationship is defined by the Relationship Type specified for the Slot of the Relationship.

Input Parameters

Parameter Name	Date Type	Description	Example
ID	ID	ID of the Content Item that will own the ActiveAssembly Relationships. Cannot be null or empty. If the specified ID is not found in the system, a PSContractViolationFault will be returned.	27-5-189

Parameter Name	Date Type	Description	Example
Slot	String	Name of the Slot in the ActiveAssembly Relationship. During assembly, the assembled Dependents will be added to this Slot. Cannot be null or empty If the value of this parameter is not a valid Slot name in the system, a PSContractViolationFault will be returned.	Sidebar
Template	ID	Name of the Template in the ActiveAssembly Relationship. During assembly, Rhythmyx uses this Template to format the Dependent Content Item before merging it into the Slot on the parent. Cannot be null or empty. If the value of this parameter is not the ID of an existing Template, a PSContractViolationFault will be returned.	rffListSlot
Index	Integer	The 0-based index indicating the sortrank position to insert the new ActiveAssembly Relationships in the Slot. May be null. If not included in the request, all new ActiveAssembly Relationships are appended to the end of the existing list of Relationships in the Slot, in the order the ID are specified in the RelatedId parameter.	3
RelatedId	Array	Array of IDs of Content Items to be associated with the Owner specified in the ID parameter of the request in the specified Slot. Cannot be null or empty. If any supplied ID does not exist, a PSContractViolationFault will be returned.	27-5-198,27-5-316,27-5-372

Output

PSAaRelationship (on page 110): Array consisting of all ActiveAssembly Relationships created as a result of the request. Never Null or empty. All newly-created Relationships are saved in the Repository.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned in the following circumstances: ▪ If the Content Item specified in the ID parameter does not exist in the system. ▪ If the specified Slot does not exist in the system. ▪ If the specified Template does not exist in the system. ▪ If any of the Content Items specified in the RelatedId parameter do not exist. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSNotAuthorizedFault	If the requestor is not authorized to create new ActiveAssembly Relationships for the Content Item specified in the Id parameter. For more details see ***PSNotAuthorizedFault*** (on page 128)
PSErrorResultsFault	If any requested ActiveAssembly Relationship cannot be created. For more details, see ***PSErrorResultsFault*** (on page 127).

LoadContentRelations

Loads all of the ActiveAssembly Relationships specified in the PSAaRealtionshipFilter parameter. A system may include thousands of ActiveAssembly Relationships, so a poorly specified request may degrade system performance, and may return more results than the user can work with effectively.

Input Parameters

Parameter Name	Date Type	Description	Example
PSAaRelationshipFilter (see page 110)	String	Defines the parameters used to filter the ActiveAssembly Relationships before returning them to the user. If not included, or if no parameters are specified, all Relationships in the ActiveAssembly category will be returned.	

Parameter Name	Date Type	Description	Example
LoadReferenceInfo	Boolean	If true, the following reference information will be returned with the Relationship: ▪ Sot ▪ Template ▪ Site name ▪ Folder name and path Defaults to false, which does not return the reference information.	false

Output

PSAaRelationship: Array consisting of all ActiveAssembly Relationships that meet the critiera defined in the PSAaRelationshipFilter parameter. Never null, but may be empty.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned in the following circumstances: ▪ If the Content Item specified in the ID parameter does not exist in the system. ▪ If the specified Slot does not exist in the system. ▪ If the specified Template does not exist in the system. ▪ If any of the Content Items specified in the RelatedId parameter do not exist. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSNotAuthorizedFault	If the requestor is not authorized to read or edit a Relationship that meets the criteria specified in the PSAaRelationshipFilter. For more details see ***PSNotAuthorizedFault*** (on page 128)
PSErrorResultsFault	If any ActiveAssembly Relationship that meets the criteria specified in the PSAaRelationshipFilter parameter cannot be loaded. For more details, see ***PSErrorResultsFault*** (on page 127).

SaveContentRelations

Saves all ActiveAssembly Relationships included in the PSAaRelationship parameter of the request.

This service cannot be run unless the PrepareForEdit service has been submitted for all Owner Content Items of the specified Relationships.

Input Parameters

Parameter Name	Date Type	Description	Example
PSAaRelationship	Array	Array consisting of all the ActiveAssembly Relationships to be saved. Cannot be null or empty. If any Content Item specified in either an ownerID attribute or a dependentID attribute does not exit, a PSContractViolationFault will be returned.	

Output

None

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned if any Content Item specified in an ownerID attribute or a dependentID attribute does not exist. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSNotAuthorizedFault	If the requestor is not authorized to read or edit a specified Relationship. For more details see ***PSNotAuthorizedFault*** (on page 128)
PSErrorResultsFault	If any requested ActiveAssembly Relationship cannot be saved. For more details, see ***PSErrorResultsFault*** (on page 127).

ReorderContentRelations

Updates the order of ActiveAssembly Relationships submitted with the order specified in the request. All specified Relationships must have the same Owner and be in the same Slot on that Owner. Any existing ActiveAssembly Relationships that are not specified in this request are moved to the end of the Slot.

This service cannot be run unless the PrepareForEdit service has been submitted the Owner Content Item of the specified Relationship.

Input Parameters

Parameter Name	Date Type	Description	Example
Id	Array	Array consisting of all the ActiveAssembly Relationships to be reordered, in the new order. Cannot be null or empty. If any specified Relationship does not exist it is ignored	
index	Integer	Specifies the index count of Relationships to which to move the first Content Item in the Id array. If the value of this parameter is -1 or is greater than the current number of Relationships in the Slot, the first Relationship is moved to the end of the list of Relationships.	

Output

None

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned if the value of the Id parameter is null or empty, or if the array includes Relationships associated with more than one Owner or Slot. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSErrorResultsFault	If the specified Content Item cannot be modified. For more details, see ***PSErrorResultsFault*** (on page 127).

DeleteContentRelations

Deletes the specified ActiveAssembly Relationships.

This service cannot be run unless the PrepareForEdit service has been submitted for all Owner Content Items of the specified Relationships.

Input Parameters

Parameter Name	Date Type	Description	Example
Id	Array	Array consisting of all the ActiveAssembly Relationships to be deleted. Cannot be null or empty. If any specified Relationship does not exist it is ignored.	

Output

None

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	If the value of the Id parameter is null or empty. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSNotAuthorizedFault	If the requestor is not authorized to read or edit a specified Relationship. For more details see ***PSNotAuthorizedFault*** (on page 128)
PSErrorResultsFault	If any specified ActiveAssembly Relationship cannot be deleted. For more details, see ***PSErrorResultsFault*** (on page 127).

FindDependents

Finds all ActiveAssembly Relationship Dependents of the specified Content Item.

Input Parameters

Parameter Name	Date Type	Description	Example
Id	ID	ID of the Content Item whose ActiveAssembly Dependents the user wants to find. Cannot be null or empty. If the specified value does not exist, a PSContractViolationFault will be returned.	27-5-189
PSAaRelationshipFilter (see page 110)	String	Defines the parameters to use when filtering the array of returned Content Item summaries. If not specified, or if specified without filter parameters, all ActiveAssembly Dependents of the Content Item specified in the Id parameter will be returned.	
LoadOperations	Boolean	If the value is specified as true, the PSItemsummary returned includes the set of available Actions that the user can take on the Content Items. (The list of available actions is filtered based on the user's Roles and Community. The default value is false, which specifies that the actions will not be returned.	False

Output

PSItemSummary (see page 116): Array of Content Item summaries of all Active Assembly Dependents of the Content Item specified by the Id parameter that meet the criteria specified in the PSAaRelationshipFilter parameter. Cannot be null, but may be empty.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned if the value of the Id parameter is null or empty. For more details, see ***PSContractViolationFault*** (on page 126).

Fault Name	Returned
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).

FindOwners

Finds all ActiveAssembly Relationship Owners of the specified Content Item.

Input Parameters

Parameter Name	Date Type	Description	Example
Id	ID	ID of the Content Item whose ActiveAssembly Owners the user wants to find. Cannot be null or empty. If the specified value does not exist, a PSContractViolationFault will be returned.	27-5-189
PSAaRelationshipFilter (see page 110)	String	Defines the parameters to use when filtering the array of returned Content Item summaries. If not specified, or if specified without filter parameters, all ActiveAssembly Dependents of the Content Item specified in the Id parameter will be returned.	
LoadOperation	Boolean	If the value is specified as true, the PSItemSummary returned includes the set of available Actions that the user can take on the Content Items. (The list of available actions is filtered based on the user's Roles and Community. The default value is false, which specifies that the actions will not be returned.	False

Output

PSItemSummary (see page 116): Array of Content Item summaries of all Active Assembly Owners of the Content Item specified by the Id parameter that meet the criteria specified in the PSAaRelationshipFilter parameter. Cannot be null, but may be empty.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned if the value of the Id parameter is null or empty. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).

Folders Services

Folders services allow users to access manipulate Folders and the Content Items associated with them.

Note that several of these services have two signatures, one of which allows the user to manipulate a Folder based on its ID, the other based on its path. Each signature of these services is documented separately. Folder services with multiple signatures include:

- LoadFolder
- FindFolderChildren
- AddFolderChildren
- RemoveFolderChildren
- MoveFolderChildren

LoadFolders

Load the Folders, including any contained Content Items, specified by the Id parameter.

Input Parameters

Parameter Name	Date Type	Description	Example
Id	Array	Array of IDs of the Folders to load. Cannot be null or empty. If any specified Folder does not exist, a PSContractViolationFault will be returned.	27-7-317

Output

PSFolder (see page 114): Array with all Folders (and their contained Content Items) requested, in the order specified in the Id parameter. Never null, but may be empty.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	If any Folder specified in the Id parameter does not exist. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSErrorResultsFault	If any requested Folder cannot be loaded. For more details see ***PSErrorResultsFault*** (on page 127).

LoadFolders

Load the Folders, including any contained Content Items, specified by the Path parameter.

Input Parameters

Parameter Name	Date Type	Description	Example
Path	Array	Array of fully-qualified paths to the Folders to load. Cannot be null or empty. To retrieve the root nodes (Folders and Sites), enter "/". If the user already has a Folder loaded, this value will return the root node of their current Folder. If any specified Folder does not exist, a PSContractViolationFault will be returned.	/Sites/EnterpriseInvestments/InvestmentAdvice/Retirement

Output

PSFolder (see page 114): Array with all Folders (and their contained Content Items) requested, in the order specified in the Id parameter. Never null, but may be empty.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	If any part of a path specified in the Path parameter is invalid or does not exist. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSErrorResultsFault	If any requested Folder cannot be loaded. For more details see ***PSErrorResultsFault*** (on page 127).

SaveFolders

Saves all Folders included with the request. Existing Folders are updated with the data included in the request. Any Folder included in the request that does not already exist will be created and saved to the Repository.

Input Parameters

Parameter Name	Date Type	Description	Example
PSFolder	Array	Array of PSFolder objects to save to the Repository. Cannot be null or empty.	27-7-317

Output

ID: Array of Folder IDs for all saved Folders, returned in the same order specified in the request.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	If the value of the PSFolder parameter is null or empty. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSErrorResultsFault	If any specified Folder cannot be saved. For more details see ***PSErrorResultsFault*** (on page 127).

DeleteFolders

Deletes all Folders specified in the Id parameter. The delete action is recursive; all Subfolders of the specified Folders will be deleted. Optionally, any Content Items contained by the specified Folders and their Subfolders can be purged as well. Note that the user submitting the request must have Admin rights to purge Content Items.

If a specified Folder does not exist, it is ignored.

Note that the delete action cannot be reversed.

Input Parameters

Parameter Name	Date Type	Description	Example
Id	Array	Array of IDs of Folders to delete from the system. All Folders specified will be deleted, as will all Descendant Subfolders.	27-7-317

Parameter Name	Date Type	Description	Example
PurgeItems	Boolean	Flag specifying whether to purge Content Items contained in any of the deleted Folders. Defaults to False if not included in the request. The user issuing the request must have Admin privileges to purge Content Items. If the user does not have Admin privileges, the purge action will fail and a PSErrorsFault will be returned.	False

Output

None

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	If the value of the Id parameter is not valid. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSErrorResultsFault	If any specified Folder cannot be deleted, or if the user submitting the request does not have Admin privileges. For more details see ***PSErrorResultsFault*** (on page 127).

AddFolder

Adds a new Folder with the specified Label to the specified parent Folder. The new Folder inherits the ACL of the parent Folder. Use the ***SaveFolders service*** (see page 62) to change the ACL of the new Folder.

The name of the Folder must be unique among the Folders contained by the parent Folder.

Input Parameters

Parameter Name	Date Type	Description	Example
Label	String	Label of the new Folder. Cannot be null or empty. Must be unique among the Labels of Folders contained by the parent Folder.	Retirement

Parameter Name	Date Type	Description	Example
Path	String	Fully-qualified path to the parent Folder to which the new Folder will be added. Cannot be null or empty. Must specify a path to an existing Folder. If the user submitting the request is not listed in the ACL of the parent Folder, the processing of this request will add them automatically.	/Sites/EnterpriseInvestments/InvestmentAdvice

Output

PSFolder (see page 114): The newly created Folder. Never Null or empty.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	If the value of the Path parameter is null or empty or contains an invalid Folder. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSErrorResultsFault	If the value in the Label parameter is the same as a Folder that already exists in the specified parent Folder, or if any specified Folder cannot be created in the specified parent Folder for some other reason. For more details see ***PSErrorResultsFault*** (on page 127).

AddFolderTree

Adds the Folder tree specified in the Path parameter to the location specified in that parameter. Both the existing and new trees must be fully-qualified. The new Folders will be created as children of the last existing Folder in the Path parameter. The new Folders inherit the Access Control List of the last existing Folder specified in the Path parameter.

Input Parameters

Parameter Name	Date Type	Description	Example
Path	String	Fully-qualified path to the parent Folder to which the new Folder will be added. Cannot be null or empty. Must specify a path to an existing Folder. If the user submitting the request is not listed in the ACL of the parent Folder, the processing of this request will add them automatically.	/Sites/EnterpriseInvestments/InvestmentAdvice/Education/SavingsPlans

Output

PSFolder (see page 114): The newly created Folders. Never Null or empty.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	If the value of the Path parameter is null or empty or contains an invalid Folder. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSErrorResultsFault	If the name of any newly-specified Folder is the same as any Folder that already exists in the immediate parent Folder, or if any specified Folder cannot be created in the specified parent Folder for some other reason. For more details see ***PSErrorResultsFault*** (on page 127).

FindFolderChildren

Finds all of the Content Items and Folders directly owned by the Folder specified in the Id parameter. (The objects contained in any child Folder are not returned by this request. To retrieve those objects, the service should be submitted again for with the ID of the Folder whose children the user wants to see.)

Input Parameters

Parameter Name	Date Type	Description	Example
Id	ID	ID of the Folder whose contents the user wants to find. Cannot be null. Must be the ID of an existing Folder.	27-7-317
LoadOperation	Boolean	If the value is specified as true, the PSItemSummary returned includes the set of available Actions that the user can take on the Content Items. (The list of available actions is filtered based on the user's Roles and Community. The default value is false, which specifies that the actions will not be returned.	False

Output

PSItemSummary (see page 116): Array of the Content Item and Folder summaries for all of the objects directly owned by the Folder specified in the Id parameter. Never null, but may be empty.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	If the value of the Id parameter is null or if the Folder specified in the Id parameter does not exist. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).

FindFolderChildren

Finds all of the Content Items and Folders directly owned by the Folder specified in the Path parameter. (The objects contained in any child Folder are not returned by this request. To retrieve those objects, the service should be submitted again for with the ID of the Folder whose children the user wants to see.)

Input Parameters

Parameter Name	Date Type	Description	Example
Path	String	Fully-qualified path to the Folder whose contents the user wants to retrieve. Cannot be null. To retrieve the contents of the root node of the current Folder (either Sites or Folder), submit "/" as the value of this parameter.	//Sites/EnterpriseInvestments/InvestmentAdvice
LoadOperation	Boolean	If the value is specified as true, the PSItemSummary returned includes the set of available Actions that the user can take on the Content Items. (The list of available actions is filtered based on the user's Roles and Community. The default value is false, which specifies that the actions will not be returned.	False

Output

PSItemSummary (see page 116): Array of the Content Item and Folder summaries for all of the objects directly owned by the Folder specified in the Path parameter. Never null, but may be empty..

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	If the value of the Path parameter is null or if any Folder specified in the Path parameter does not exist. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).

AddFolderChildren

Adds all of the Content Items and Folders specified in the ChildId parameter to the Folder specified in the ParentId parameter.

Input Parameters

Parameter Name	Date Type	Description	Example
ParentId	ID	ID of the Folder to which the Content Items and Folders specified in the ChildId parameter should be added. Cannot be null. Must be the ID of an existing Folder	27-7-317
ChildId	Array	Array of IDs of the Content Items and Folders to add to the Folder specified in the ParentId parameter. Cannot be null or empty. All Content Items and Folders specified must exist in the system. If the Label of any Folder specified matches the Label of a Folder that already exists in the Folder specified by the ParentId parameter, a PSErrorResultsFault will be returned.	27-5-326,27-5-379,27-5-402,27-5-529

Output

None

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	If the value of the ParentId parameter is null, or if the Folder specified by the ParentId parameter does not exist. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSErrorResultsFault	If the Label of any Folder specified in the ChildId parameter matches the label of a Folder that already exists in the Folder specified in the ParentId parameter, or if for some other reason any Content Item or Folder specified in the ChildId parameter cannot be added to the Folder specified in the ParentId parameter. For more details see ***PSErrorResultsFault*** (on page 127).

AddFolderChildren

Adds all of the Content Items and Folders specified in the ChildId parameter to the Folder specified in the Path parameter.

Input Parameters

Parameter Name	Date Type	Description	Example
Path	String	Fully-qualified Folder path to the Folder to which the user wants to add the Content Items and Folders specified in the ChildId parameter. Cannot be null. The path must lead to an existing Folder.	//Sites/EnterpriseInvestments/InvestmentAdvice
ChildId	Array	Array of IDs of the Content Items and Folders to add to the Folder specified in the Path parameter. Cannot be null or empty. All Content Items and Folders specified must exist in the system. If the Label of any Folder specified matches the Label of a Folder that already exists in the Folder specified by the Path parameter, a PSErrorResultsFault will be returned.	27-5-326,27-5-379,27-5-402,27-5-529

Output

None

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	If the value of the Path parameter is null, or if the Folder specified by the Path parameter does not exist. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).

Fault Name	Returned
PSErrorResultsFault	If the Label of any Folder specified in the ChildId parameter matches the label of a Folder that already exists in the Folder specified in the Path parameter, or if for some other reason any Content Item or Folder specified in the ChildId parameter cannot be added to the Folder specified in the ParentId parameter. For more details see ***PSErrorResultsFault*** (on page 127).

RemoveFolderChildren

Removes the Content Items and Folders specified in the ChildId parameter from the Folder specified in the ParentId parameter.

Ordinarily, these objects are simply removed from the Folder but remain in the system, but users with admin privileges have the option of purging them from the system permanently. Note that the purge action is not reversible.

Input Parameters

Parameter Name	Date Type	Description	Example
ParentId	ID	ID of the Folder from which to remove the Content Items and Folders specified in the ChildId parameter. Cannot be null. Must be the ID of an existing Folder.	27-7-317
ChildId	Array	Array of IDs of the Content Items and Folders to remove from the Folder specified in the ParentId parameter. If not specified, all contents of the Folder specified in the ParentId parameter will be removed. Any IDs that do not exist in the system are ignored.	27-5-326,27-5-379,27-5-402,27-5-529
PurgeItems	Boolean	Flag specifying whether to purge the Content Items and Folders specified in the ChildId parameter completely. Defaults to False. If this flag is True but the user submitting the request does not have admin privileges, the Content Items and Folders will be removed from the parent Folder, but will not be purged.	False

Output

None

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	If the value of the ParentId parameter is null, or if the Folder specified by the ParentId parameter does not exist. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).

RemoveFolderChildren

Removes the Content Items and Folders specified in the ChildId parameter from the Folder specified in the Path parameter.

Ordinarily, these objects are simply removed from the Folder but remain in the system, but users with admin privileges have the option of purging them from the system permanently. Note that the purge action is not reversible.

Input Parameters

Parameter Name	Date Type	Description	Example
Path	String	Fully-qualified path to the Folder from which to remove the Content Items and Folders specified in the ChildId parameter. Cannot be null. The Path must lead to an existing Folder.	//Sites/EnterpriseInvestments/InvestmentAdvice/Retirment
ChildId	Array	Array of IDs of the Content Items and Folders to remove from the Folder specified in the Path parameter. If not specified, all contents of the Folder specified in the Path parameter will be removed. Any IDs that do not exist in the system are ignored.	27-5-326,27-5-379,27-5-402,27-5-529

PurgeItems	Boolean	Flag specifying whether to purge the Content Items and Folders specified in the ChildId parameter completely. Defaults to False If this flag is True but the user submitting the request does not have admin privileges, the Content Items and Folders will be removed from the parent Folder, but will not be purged.	False

Output

None

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	If the value of the Path parameter is null, or if the Folder specified by the Path parameter does not exist. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).

MoveFolderChildren

Moves the Content Items and Folders specified in the ChildId parameter from the Folder specified in the SourceId parameter to the Folder specified in the TargetId parameter.

Input Parameters

Parameter Name	Date Type	Description	Example
SourceId	ID	ID of the Folder from which to move Content Items and Folders specified in the ChildId parameter. Cannot be null or empty. Must be the ID of an existing Folder	27-7-317
TargetId	ID	ID of the Folder to which to more the Content Items and Folders specified in the ChildId parameter. Cannot be null or empty. Must be the ID of an existing Folder.	27-7-497

Parameter Name	Date Type	Description	Example
ChildId	Array	Array of IDs of the Content Items and Folders to move from the Folder specified in the SourceId parameter to the Folder specified in the TargetId parameter.. Cannot be null or empty. All Content Items and Folders specified must exist in the system. If the Label of any Folder specified matches the Label of a Folder that already exists in the Folder specified by the TargetId parameter, a PSErrorResultsFault will be returned.	27-5-326,27-5-379,27-5-402,27-5-529

Output

None

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned in the following circumstances: ▪ if the value of the SourceId parameter is null or empty; ▪ if the Folder specified in the SourceId parameter does not exist in the system; ▪ if the value of the TargetId parameter is null or empty; ▪ if the Folder specified in the TargetId parameter does not exist in the system; ▪ if any of the Content Items or Folders specified in the ChildId parameter do not exist. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSErrorResultsFault	If the Label of any Folder specified in the ChildId parameter matches the label of a Folder that already exists in the Folder specified in the TargetId parameter, or if for some other reason any Content Item or Folder specified in the ChildId parameter cannot be moved from the Folder specified in the SourceId parameter to the Folder specified in the TargetId parameter. For more details see ***PSErrorResultsFault*** (on page 127).

MoveFolderChildren

Moves the Content Items and Folders specified in the ChildId parameter from the Folder specified in the SourcePath parameter to the Folder specified in the TargetPath parameter.

Input Parameters

Parameter Name	Date Type	Description	Example
SourcePath	ID	Fully-qualified path to the Folder from which to move Content Items and Folders specified in the ChildId parameter. Cannot be null or empty. The Path must specify an existing Folder	//Sites/EnterpriseInvestments/InvestmentAdvice/SavingforCollege
TargetId	ID	ID of the Folder to which to more the Content Items and Folders specified in the ChildId parameter. Cannot be null or empty. Must be the ID of an existing Folder.	//Sites/EnterpriseInvestments/InvestmentAdvice/Education
ChildId	Array	Array of IDs of the Content Items and Folders to move from the Folder specified in the SourcePath parameter to the Folder specified in the TargetPath parameter. Cannot be null or empty. All Content Items and Folders specified must exist in the system. If the Label of any Folder specified matches the Label of a Folder that already exists in the Folder specified by the ParentId parameter, a PSErrorResultsFault will be returned.	27-5-326,27-5-379,27-5-402,27-5-529

Output

None

Faults

Fault Name	Returned
AxisFault	Any unexpected error

Fault Name	Returned
PSContractViolationFault	Returned in the following circumstances: ▪ if the value of the SourcePath parameter is null or empty; ▪ if the Folder specified in the SourcePath parameter does not exist in the system; ▪ if the value of the TargetPath parameter is null or empty; ▪ if the Folder specified in the TargetPath parameter does not exist in the system; ▪ if any of the Content Items or Folders specified in the ChildId parameter do not exist. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSErrorResultsFault	If the Label of any Folder specified in the ChildId parameter matches the label of a Folder that already exists in the Folder specified in the TargetPath parameter, or if for some other reason any Content Item or Folder specified in the ChildId parameter cannot be moved from the Folder specified in the SourcePath parameter to the Folder specified in the TargetPath parameter. For more details see ***PSErrorResultsFault*** (on page 127).

FindFolderPath

Finds the path from the specified Content Item or Folder through all Ancestors to the root node of its currently location.

Input Parameters

Parameter Name	Date Type	Description	Example
Id	ID	ID of the Content Item or Folder whose path the user wants to return. . Cannot be null or empty. If any specified Content Item or Folder does not exist, a PSContractViolationFault will be returned.	27-7-317

Output

Path: Array of fully-qualified Folder paths from root node to the specified Content Item or Folder. Can never be null, but may be empty.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	If the value of the Id parameter is null or empty or if the specified Content Item or Folder does not exist. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).

FindPathIds

Finds the IDs of all Folders in the specified path. For example, if the input path was //Sites/EnterpriseInvestments/InvestmentAdvice/Retirement, an array of four IDs would be returned. The first ID would be that of //Sites, the next of /EnterpriseInvestments, and so forth.

Input Parameters

Parameter Name	Date Type	Description	Example
Path	String	The fully-qualified path for which to find the IDs. Cannot be null or empty. If the specified path does not exist, a PSContractViolationFault will be returned.	//Sites/ EnterpriseInvestments/ InvestmentAdvice/ Retirement

Output

ID: Array containing all the IDs of the Folders in the specified path. Never null or empty.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	If the value of the Path parameter is null or empty or if the specified path does not exist. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).

Keyword Services

Keyword services provide users with the ability to retrieve Keywords and their Choices.

LoadKeywords

Loads all Keyword definitions that match the value submitted in the Name parameter. The Keyword definitions are returned in read-only mode.

Input Parameters

Parameter Name	Date Type	Description	Example
Name	String	Name of the Keyword to load. If the value of this parameter is null or empty, all Keyword definitions will be returned. Wildcards are accepted in the value. For details about wildcards, see (XREF).	Offices

Output

PSKeyword (on page 116): Array with all loaded Keyword definitions, in read-only mode. Never null, but may be empty. The array is ordered alphabetically by Keyword name.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSNotAuthorizedFault	If the requestor is not authorized to read Keywords. For more details see ***PSNotAuthorizedFault*** (on page 128)

Localization Services

Localization services provide functionality to retrieve Localization data for changing Locales and creating Translation Content Items.

LoadLocales

Loads all Locale definitions that match the value submitted in the Name parameter. The Locale definitions are returned in read-only mode. The submitted request can include either the Code parameter or the Name parameter, or both. Note, however, that conflicting parameters produce no results. Thus, if a user submits a request where Code=en-us and Name=Canadian English, the value returned will be empty since the Name for the Locale Code en-us is "US English".

Input Parameters

Parameter Name	Date Type	Description	Example
Code	String	The Locale code of the Locale definition to load. Can be null or empty. If the value of this parameter is null or empty and the Name parameter is not included in the request or the value of the Name parameter is null or empty, all Locale definitions will be returned. For details about Locale Codes, see *Internationalizing and Localizing Rhythmyx*.	en-us
Name	String	Name of the Locale to load. May be null or empty. Wildcards are accepted in the value. For details about wildcards, see (XREF).	Offices

Output

PSLocale (see page 116): Array with all loaded Locale definitions, in read-only mode. Never null, but may be empty. The array is ordered alphabetically by Locale name.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).

Fault Name	Returned
PSNotAuthorizedFault	If the requestor is not authorized to read Locales. For more details see ***PSNotAuthorizedFault*** (on page 128)

LoadAutoTranslations

Loads all Automatic Translation configurations in read-only mode.

Input Parameters

None

Output

PSAutoTranslation (see page 113): Array with all loaded Automatic Translation configurations in read-only mode. Never null, but may be empty.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSNotAuthorizedFault	If the requestor is not authorized to read Automatic Translation Configurations. For more details see ***PSNotAuthorizedFault*** (on page 128)

Content Type Services

Content Type services provide functionality to retrieve data summaries of Content Types.

LoadContentTypes

Loads all Content Type summaries that match the value specified in the Name parameter. All Content Type summaries are returned in read-only mode.

Input Parameters

Parameter Name	Date Type	Description	Example
Name	String	Name of the Content Type whose summary to load. If null or empty all Content Type summaries will be returned.	Generic

Output

PSContentTypeSummary (see page 114): Array with all loaded Content Type summaries, in read-only mode. Never null, but may be empty. The array is ordered alphabetically by Content Type name.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSNotAuthorizedFault	If the requestor is not authorized to read Content Type definitions. For more details see ***PSNotAuthorizedFault*** (on page 128)

CHAPTER 3

Assembly Services

Rhythmyx Assembly Services provide functionality to retrieve data definitions for Content Assembly.

LoadSlots

Loads all Slot definitions that match the value specified in the Name parameter. All Slot definitions are returned in read-only mode.

Input Parameters

Parameter Name	Date Type	Description	Example
Name	String	Name of the Slot to load. If not included in the request, or if null or empty all Slots definitions will be returned. Wildcards are accepted in the value. For details about wildcards, see (XREF).	Sidebar

Output

PSTemplateSlot (on page 123): Array with all loaded Slot definitions, in read-only mode. Never null, but may be empty. The array is ordered alphabetically by Slot name.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSNotAuthorizedFault	If the requestor is not authorized to read Slot definitions. For more details see ***PSNotAuthorizedFault*** (on page 128)

LoadSlots

Loads all Template definitions that match the value specified in the Name parameter, the Content Type parameter, or both. All Template definitions are returned in read-only mode.

If a request includes neither the Name nor the ContentType parameter, all Templates in the system will be returned. If the request includes values in both the Name and ContentType parameters, then only Templates associated with the specified ContentType that have the specified Name will be returned.

Input Parameters

Parameter Name	Date Type	Description	Example
Name	String	Name of the Template to load. If null or empty all Template definitions will be returned. Wildcards are accepted in the value. For details about wildcards, see (XREF).	rffPgGeneric
ContentType	String	Name of the Content Type whose Templates to load. If null or empty, all Template definitions will be returned. Wildcards are accepted in the value.	rffGeneric

Output

PSAssemblyTemplate (on page 112): Array with all loaded Template definitions, in read-only mode. Never null, but may be empty. The array is ordered alphabetically by Template name.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSNotAuthorizedFault	If the requestor is not authorized to read Template definitions. For more details see ***PSNotAuthorizedFault*** (on page 128)

CHAPTER 4

System Services

Rhythmyx System services provide access to system functionality. System services are divided into the following groups:

- ***Switches services*** (see page 84): services that allow users to change Communities and Locales;
- ***Relationships services*** (see page 86): services that allow users to create, modify, and delete custom Relationships;
- ***RelationshipTypes services*** (see page 92): services that allow users to access Relationship Type definitions;
- ***Workflows Services*** (see page 94): services that allow users to access Workflow functionality;
- ***Menus services*** (see page 109): services that allow users to access Menu functionality

Switches Services

Switches services are services that allow the user to change from one Community or Locale to another.

SwtichCommunity

Changes the users logged Community to the Community specified in the Name parameter. Users can only change to Communities of which they are a Member. The list of available Communities is derived from the Communities node of the PSLogin object returned by the ***Login*** (see page 99) service.

Input Parameters

Parameter Name	Date Type	Description	Example
Name	string	Name of the Community to which to switch the user submitting the request. Cannot be null or empty. The specified Community must exist in the system.	EnterpriseInvestments

Output

None

Faults

Fault Name	Returned
AxisFault	Any unexpected error.
PSContractViolationFault	If the value of the Name parameter is null or empty, or it the specified Community does not exist. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSUserNotMemeberOfCommunityFault	If the user is not a Member of the Community specified in the Name parameter.

SwtichLocale

Changes the users current Locale to the Locale specified in the Code parameter. Users can only only change to Locales that are defined in the system and are currently enabled. The list of available Locales is derived from the Locales node of the ***PSLogin*** (see page 116) object returned by the ***Login*** (see page 99) service.

For more details about Locales, see *Internationalizing and Localizing Rhythmyx*.

Input Parameters

Parameter Name	Date Type	Description	Example
Code	string	The code of the Locale to which the user wants to switch. Cannot be null or empty. The specified Code must exist in the system and must currently be enabled.	en-us

Output

None

Faults

Fault Name	Returned
AxisFault	Any unexpected error.
PSContractViolationFault	If the value of the Code parameter is null or empty. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSInvalidLocale	If the specified Locale does not exist in the system, or if the Locale is not currently enabled.

Relationship Services

Relationship Services provide functionality to access, create, and manipulate custom Relationships. These services should only be used to modify custom Relationships. To modify default Relationships provided with Rhythmyx, use the services provided for those Relationships (such as New Copy, New Translation, New Promotable Version).

Relationship Services fall into two groups:

- Services that allow users to operate on Relationships, such as creating or modifying them. Services that fall into this group include:
  - ***CreateRelationship*** (see page 86)
  - ***LoadRelationship*** (see page 88)
  - ***SaveRelationship*** (see page 88)
  - ***DeleteRelationship*** (see page 89)
- Services that allow users to navigate Relationship trees. Services in this group include:
  - ***FindDependents*** (see page 90)
  - ***FindOwners*** (see page 91)

Create Relationship

Creates a new Relationship of the Relationship Type specified in the Name parameter between the Content Item specified in the OwnerId parameter (Owner of the Relationship) and the Content Item specified in the DependentId parameter (Dependent in the Relationship).

This Service can only be used for custom Relationships. To create a new Relationship of one of the default Relationship Types provided with Rhythmyx, use the services provided for that Relationship Type (such as New Copy or New Promotable Version).

Input Parameters

Parameter Name	Date Type	Description	Example
Name	String	Name of the Relationship Type that of which you want to create an instance. Cannot be null or empty. If this parameter specifies a Relationship Type that does not exist, a PSUnknownRelationshipTypeFault is returned.	Sample_Relationship

Parameter Name	Date Type	Description	Example
OwnerId	ID	The ID of the Content Item that will be the Owner in the Relationship. Cannot be null or empty. If the specified ID is not found in the system, a PSContractViolationFault will be returned.	5-27-312
DependentId	ID	The ID of the Content Item that will be the Dependent in the Relationship. Cannot be null or empty. If the specified ID is not found in the system, a PSContractViolationFault will be returned	5-27-312

Output

PSRelationship (see page 117): A new Relationship between the Content Item specified as the Owner in the Relationship and the Content Item specified as the Dependent. All Relationship properties are initialized with the defaults specified in the Relationship Type definition.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned if the Content Item specified in either the OwnerId or DependentId parameter does not exist in the system. For more details, see ***PSContractViolationFault*** (on page 126).
PSunknownRelationshipFault	Returned if the Relationship Type specified in the Name parameter does not exist. Confirm that the specified Relationship Type exists and is spelled correctly.
PSUseSpecificMethodsFault	Returned if the user attempts to create a new Relationship of one of the system-defined Relationship Types using this service. The request should be issued again using the correct service.

LoadRelationship

Loads all Relationships that pass the criteria specified in the PSRelationshipFilter parameter.

A system may have thousands of Relationships. Thus, the criteria in the PSREelationshipFilter parameter must be defined carefully and specifically to avoid hindering system performance.

Input Parameters

Parameter Name	Date Type	Description	Example
PSRelationshipFitler (see page 117)	Array	Array of parameters to use to filter the Relationships to return. If not specified, or if specified with no filter parameters, all Relationships in the system will be returned..	

Output

PSRelationship (see page 117): Array of Relationships that meet the criteria specified in the PSRelationshipFilter parameter. Relationships are ordered by Relationship ID. Never null, but may be empty..

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned if any of the parameters defined in the PSRelationshipFilter parameter are not valid. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSNotAuthorizedFault	If the requestor is not authorized to read or edit a Relationship that meets the criteria specified in the PSRelationshipFilter. For more details see ***PSNotAuthorizedFault*** (on page 128)

SaveRelationship

Saves the Relationships provided in the PSRelationship parameter to the Repository.

Input Parameters

Parameter Name	Date Type	Description	Example
PSRelationship (see page 117)	Array	Array of Relationships to save to the Repository. Cannot be null or empty.	

Output

None

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned if any of the parameters defined in the PSRelationship parameter are not valid. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSErrorResultsFault	If any requested Relationship cannot be saved. For more details, see ***PSErrorResultsFault*** (on page 127).

Delete Relationships

Deletes all Relationships specified in the Id parameter. The Relationships are deleted immediately. The deleted Relationship cannot be recovered.

Input Parameters

Parameter Name	Date Type	Description	Example
Id	Array	Array consisting of all the Relationships to be deleted. Cannot be null or empty. If any specified Relationship does not exist it is ignored.	

Output

None

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	If the value of the Id parameter is null or empty. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).

Fault Name	Returned
PSErrorResultsFault	If any specified Relationship cannot be deleted. For more details, see ***PSErrorResultsFault*** (on page 127).

FindRelationshipDependents

Finds all Relationship Dependents of the specified Content Item that meet the criteria specified in the PSRelationshipFilter parameter.

Input Parameters

Parameter Name	Date Type	Description	Example
Id	ID	ID of the Content Item whose Dependents the user wants to find. Cannot be null or empty. If the specified value does not exist, a PSContractViolationFault will be returned.	27-5-189
PSRelationshipFilter (see page 117)	Array	Defines the parameters to use when filtering the array of returned Content Item IDs. If not specified, or if specified without filter parameters, all Dependents of the Content Item specified in the Id parameter will be returned.	

Output

ID: Array of object IDs for all Dependents of the specified Content Item that meet the criteria specified in the PSRelationshipFilter. Never null, but may be empty.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned if the value of the Id parameter is null or empty. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).

FindRelationshipOwners

Finds all Relationship Owners of the specified Content Item that meet the criteria specified in the PSRelationshipFilter parameter.

Input Parameters

Parameter Name	Date Type	Description	Example
Id	ID	ID of the Content Item whose Owners the user wants to find. Cannot be null or empty. If the specified value does not exist, a PSContractViolationFault will be returned.	27-5-189
PSRelationshipFilter (see page 117)	Array	Defines the parameters to use when filtering the array of returned Relationship IDs. If not specified, or if specified without filter parameters, all Owners of the Content Item specified in the Id parameter will be returned.	

Output

ID: Array of object IDs for all Owners of the specified Content Item that meet the criteria specified in the PSRelationshipFilter. Never null, but may be empty.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned if the value of the Id parameter is null or empty. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).

Relationship Type Services

Relationship Type services provide functionality to access Relationship Types.

LoadRelationshipTypes

Loads all Relationships Type configurations that match the criteria specified in the service parameters. Relationship Type configurations are loaded in read-only mode.

Input Parameters

Parameter Name	Date Type	Description	Example
Name	String	The name of the Relationship Type to load. May be null or empty. Wildcards are accepted. If not included in the request or if empty, all Relationship Types will be loaded.	New*
Category	String	The Relationship Category whose configurations to load. May be null, but cannot be empty. Must be an existing Relationship Category. If not included in the request or if null, all Relationship Configurations will be returned.	Translation

Output

PSRelationshipConfigSummary (see page 117): Array of Relationship Type configurations that meet the criteria specified in the Name and Category parameters. Never null, but may be empty. Relationship Type configurations are ordered alphabetically by name.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned if any parameter is not valid. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).

Fault Name	Returned
PSNotAuthorizedFault	If the requestor is not authorized to read or edit a Relationship Type that meets the criteria specified in the request. For more details see ***PSNotAuthorizedFault*** (on page 128)

Workflow Services

Workflow Services provide functionality to access Workflow definitions and functions.

LoadAuditTrails

Loads the Workflow Audit Trail for each Content Item specified in the Id parameter.

Input Parameters

Parameter Name	Date Type	Description	Example
Id	Array	Array of IDs of the Content Items whose Audit Trails will be returned. Cannot be null or empty. If a Content Item with a specified ID does not exist, a PSContractViolationFault will be returned.	27-5-189

Output

PSAuditTrail (see page 113): Array of Audit Trails of the requested Content Items, in the order requested. Content Item Revisions are returned with the Audit Trails. Cannot be null or empty.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned if the value of the Id parameter is null or empty, or if a Content Item does not exist for a specified ID. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSErrorResultsFault	If any requested Audit Trail fails to load. For more details, see ***PSErrorResultsFault*** (on page 127).

TransitionItems

Transitions the Content Items specified in the Id parameter using the Transition specified in the Transition parameter. The same Transition is used for all Content Items. All Content Items specified by the Id parameter must be checked in.

Input Parameters

Parameter Name	Date Type	Description	Example
Id	Array	Array of IDs of the Content Items to be Transitioned. All specified Content Items must be checked in. Cannot be null or empty. If no Content Item exists with a specified ID, a PSContractViolationFault will be returned.	
Transition	String	Name of the Transition to use for the Content Items specified in the Id parameter. May be null or empty. If null or empty, the first Default Transition (in alphabetical order) for the current State of each Content Item will be used. If no Transition is specified for a Content Item, and no Default Transition exists from the current State of the Content Item, a PSErrorResultsFault will be returned. If no Transition exists for any of the specified Content Items in the current State, or if a specified Transition does not exist, a PSContractViolationFault will be returned.	Approve
Comment	String	Comment entered with the Transition. Whether the comment is required or optional is determined by the Workflow configuration.	Please check spelling.

Output

State: Array of names of the States to which the specified Content Items were Transitioned, in the order in which the Content Items were specified in the request. Never null or empty.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned if no Transition exists for a specified Content Item in its current State, or if a specified State does not exist. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSErrorResultsFault	Returned in the following circumstances: ▪ If no Transition is specified for a Content Item, and no Default Transition exists from the current State of the Content Item ▪ If the Transition of any Content Item fails. For more details, see ***PSErrorResultsFault*** (on page 127)

LoadWorkflows

Loads all Workflows specified in the Name parameter, in read-only mode.

Input Parameters

Parameter Name	Date Type	Description	Example
Name	String	Name of the Workflow to load. Can use wildcards. May be null or empty. If null or empty, all Workflows will be loaded.	S* (Loads all Workflows whose name begins with the letter "s".)

Output

PSWorkflow (see page 123): Array of all loaded Workflows, in read-only mode, ordered alphabetically by name. Never null, may be empty.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).

Fault Name	Returned
PSNotAuthorizedFault	If the requestor is not authorized to read Workflows. For more details see ***PSNotAuthorizedFault*** (on page 128)

GetAllowedTransitions

Returns the names of all Transitions that can be performed on the submitted Content Items based on their current State.

Input Parameters

Parameter Name	Date Type	Description	Example
Id	Array	Content Item ID of the Content Item for which to retrieve Transitions. If the request includes multiple Content Items, only the Transitions currently available to all specified Content Items will be returned. Cannot be null or empty. If no Content Item exists for the specified ID, a PSContractViolationFault will be returned..	5-26-483, 5-26-871

Output

Transition: Array of all Transition names available for the specified Content Item. If multiple Content Items were submitted, only the Transitions currently available to all specified Transitions are included in the results.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned if no Content Item exists for the value specified in the Id parameter, or if the Id parameter is otherwise invalid. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).

CHAPTER 5

Security Services

Rhythmyx Security services provide access to functionality controlling user access to the system. Security services are divided into the following groups:

- ***Login services*** (see page 99)
- ***Communities services*** (see page 102)
- ***Roles services*** (see page 104)

Login Services

Login services provide users with the ability to log in to and log out of the system, and to refresh their session.

Login

User login request. If the user submits valid credentials, a session ID is returned. This session ID must be attached to all other Web Services requests during the session.

Input Parameters

Parameter Name	Date Type	Description	Example
Username	String	The user name to be used to authenticate. Cannot be null or empty	admin1
Password	String	The password to be used to authenticate. Cannot be null or empty.	demo
Community	String	The name of the Community into which the requestor wants to be logged. If not included in the request and the user has not logged in before, they will be logged in to the first Community alphabetically by name into which they are eligible to log. If not included in the request and the user has logged in before, they will be logged in to the last Community into which they were logged.	EnterpriseInvestments
LocaleCode	String	The Locale Code of the Locale into which the user wants to be logged. If not included in the request and the user has not logged in before, they will be logged in to the default Locale, en-us. If not included in the request and the user has logged in before, they will be logged into the last Locale into which they were logged.	fr-fr

Parameter Name	Date Type	Description	Example
ClientId	String	Unique ID for the client logging in to the system. May be null or empty. This parameter is optional. If included in the request, it will be attached to the user's session ID. It is used internally for design object locks. If a client ID is included in the request, and the server later crashes, the user can recover the client by logging in again with the same client ID.	5-12-849

Output

PSLogin (see page 116): Login response containing the user's session ID, Community, Roles, and Locale. The session ID must be used with all subsequent Web Services requests. Never null or empty.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned in the following circumstances: ▪ If the Username parameter is null or empty. ▪ If the Password parameter is null or empty. ▪ If the value of any parameter is invalid. For more details, see ***PSContractViolationFault*** (on page 126).
PSNotAuthenticatedFault	If the user credentials (Username and Password) cannot be authenticated.
PSErrorResultsFault	If any requested Content Item fails to load. For details, see ***PSErrorResultsFault*** (on page 127)

Logout

User logout request. The session specified in the SessionID parameter will be invalidated and will not be usable again following this request.

Input Parameters

Parameter Name	Date Type	Description	Example
SessionId	String	Session ID of the session to be invalidated. Cannot be null or empty.	

Output

None

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned if the session ID included in the SessionId parameter is not valid. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	Returned if the session ID submitted in the SessionId parameter is not valid. For more details, see ***PSInvalidSessionFault*** (on page 128).

RefreshSession

Refreshes the session specified in the SessionId parameter. If the specified session is a valid session, the session timeout will be reset. If the specified session is not valid, the system returns a PSInvalidSessionFault.

Input Parameters

Parameter Name	Date Type	Description	Example
SessionId	String	Session ID of the session to be refreshed. Cannot be null or empty.	

Output

None

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned if the session ID included in the SessionId parameter is not valid. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	Returned if the session ID submitted in the SessionId parameter is not valid. For more details, see ***PSInvalidSessionFault*** (on page 128).

Communities Services

Communities services provide functionality to access Community definitions.

LoadCommunities

Loads all Communities that match the value specified in the Name parameter, in read-only mode.

Input Parameters

Parameter Name	Date Type	Description	Example
Name	String	Name of the Community to load. wildcards are accepted. May be null or empty. If null or empty, all Communities will be loaded.	Ent* (loads all Communities beginning with the string "Ent")

Output

PSCommunity (see page 113): Array containing all loaded Communities in read-only mode. Never null, but may be empty. Communities are ordered alphabetically by name.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSInvalidSessionFault	Returned if the session is not valid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSNotAuthorizedFault	Returned if the requestor is not authorized to read Communities. For more details see ***PSNotAuthorizedFault*** (on page 128)

FilterByRuntimeVisibility

Filters the list of submitted design object (such as a Content Item, Slot, Template, and so forth) IDs by the Community of the user submitting the request.

Input Parameters

Parameter Name	Date Type	Description	Example
Id	ID	IDs of the design objects to filter for visibility to the submitting user's Community. Cannot be null or empty. If no design object exists for the specified ID, a PSContractViolationFault will be returned..	5-26-483

Output

Id: Array IDs of design objects that are visible to the currently logged Community of the user that submitted the request.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSContractViolationFault	Returned if no design object exists for the value specified in the Id parameter, or if the Id parameter specifies an object that is not a design object. For more details, see ***PSContractViolationFault*** (on page 126).
PSInvalidSessionFault	If the session is invalid. For more details, see ***PSInvalidSessionFault*** (on page 128).

Roles Services

Roles services provide functionality to access Role definitions.

LoadRoles

Loads all Role definitions that match the value in the Name parameter, in read-only mode.

Input Parameters

Parameter Name	Date Type	Description	Example
Name	String	Name of the Role to load. wildcards are accepted. May be null or empty. If null or empty, all Roles will be loaded.	Ed (loads all Communities beginning with the string "Ed")

Output

PSRole (see page 120): Array containing all loaded Roles in read-only mode. Never null, but may be empty. Roles are ordered alphabetically by name.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSInvalidSessionFault	Returned if the session is not valid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSNotAuthorizedFault	Returned if the requestor is not authorized to read Roles. For more details see ***PSNotAuthorizedFault*** (on page 128)

CHAPTER 6

User Interface Services

Rhythmyx User Interface services provide the ability to load representations of the Content Explorer interface.

LoadActions

Loads the specified Action Menu entry in read-only mode.

Input Parameters

Parameter Name	Date Type	Description	Example
Name	String	Name of the Action Menu entry to load. Wildcards are accepted. May be null or empty. If null or empty, all Action Menu entries will be loaded.	Preview* (loads all Action Menu entries beginning with the string "Preview")

Output

PSAction (see page 112): Array containing all loaded Action Menu entries in read-only mode. Never null, but may be empty. Action Menu entries are ordered alphabetically by name.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSInvalidSessionFault	Returned if the session is not valid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSNotAuthorizedFault	Returned if the requestor is not authorized to read Communities. For more details see ***PSNotAuthorizedFault*** (on page 128)

LoadDisplayFormats

Loads the specified Display Format in read-only mode.

Input Parameters

Parameter Name	Date Type	Description	Example
Name	String	Name of the Display Format to load. Wildcards are accepted. May be null or empty. If null or empty, all Action Menu entries will be loaded.	Stan* (loads all Display Formats beginning with the string "Stan")

Output

PSDisplayFormat (see page 114): Array containing all loaded Display Formats in read-only mode. Never null, but may be empty. Display Formats are ordered alphabetically by name.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSInvalidSessionFault	Returned if the session is not valid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSNotAuthorizedFault	Returned if the requestor is not authorized to read Communities. For more details see ***PSNotAuthorizedFault*** (on page 128)

LoadSearches

Loads the specified Searches in read-only mode. Used to load both pre-defined Searches and user-defined saved searches.

Input Parameters

Parameter Name	Date Type	Description	Example
Name	String	Name of the Search to load. Wildcards are accepted. May be null or empty. If null or empty, all Action Menu entries will be loaded.	Gene* (loads all Searches beginning with the string "Gene")

Output

PSSearchDef (see page 122): Array containing all loaded Searches in read-only mode. Never null, but may be empty. Searches are ordered alphabetically by name.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSInvalidSessionFault	Returned if the session is not valid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSNotAuthorizedFault	Returned if the requestor is not authorized to read Communities. For more details see ***PSNotAuthorizedFault*** (on page 128)

LoadViews

Loads the specified Views in read-only mode.

Input Parameters

Parameter Name	Date Type	Description	Example
Name	String	Name of the View to load. Wildcards are accepted. May be null or empty. If null or empty, all Action Menu entries will be loaded.	*box* (loads all Views that contain the string "box")

Output

PSViewDef (see page 123): Array containing all loaded Views in read-only mode. Never null, but may be empty. Views are ordered alphabetically by name.

Faults

Fault Name	Returned
AxisFault	Any unexpected error
PSInvalidSessionFault	Returned if the session is not valid. For more details, see ***PSInvalidSessionFault*** (on page 128).
PSNotAuthorizedFault	Returned if the requestor is not authorized to read Communities. For more details see ***PSNotAuthorizedFault*** (on page 128)

CHAPTER 7

Common Data Types

This chapter describes data types that are used by Rhythmyx Web services.

Id

This Data Type is a container that contains one object ID. Object IDs are represented by long integers, but may be either an integer ID or a Globally Unique Identifier (GUID). In some cases, you may have an integer ID and need a GUID. To retrieve the GUID, submit a search on the integer ID. The GUID will be returned. Integer IDs are used in processing Relationships, and the query syntax of the fulltext search engine requires the integer version of the ID.

Most IDs are invariant between sessions and can be cached by Web Services clients. The one exception to this rule is the ID of data in Content Item child tables,. which may change when a Content Item is modified.

For Content Items and Folders, two forms of ID are available; one includes the Revision, the other does not. In most cases, either form of the ID can be used, but the following services require the Revision ID:

- FindRevisions

  The request need not contain the Revision ID, but the result will include Revision IDs
- PromoteRevisions

  The request requires a Revision ID.
- LoadAuditTrails

  The request need not contain the Revision ID, but the result will include Revision IDs
- ViewItems

  The request need not contain the Revision ID. If the request does not include the Revision ID, the current Revision will be returned. If the request does include the Revision ID, the specified Revision will be returned.
- GetAssemblyUrls

  The request need not contain the Revision ID. If the request does not include the Revision ID, the assembly URL of the current Revision is returned. If the request does include the Revision ID, the assembly URL of the specified Revision is returned.

Whenever a Revision ID is submitted, the system validates that it is valid to execute the requested action on the specified Revision. For example, if a user submits a Prepare for Edit request for a Revision that is not the Current Revision and the Content Item is not checked out to that user, the system returns an error.

PSAaRelationship

This Data Type is a container for one instance of a Relationship in the Active Assembly Category.

PSAaRelationshipFilter

This Data Type is a container that contains all properties that can be used to filter Relationships returned when a user submits a request to lookup Active Assembly Relationships. (For other Relationships, use the ***PSRelationshipFilter*** (see page 117).)

A number of filter options are available. One request can include multiple filters, and only Active Assembly Relationships that meet all filter criteria specified (in other words, the intersection of the results sets) will be returned. For example, if the request specifies both the slot property and the template property, then any Relationship that is associated with the specified Slot and that uses the specified Template will be returned. If mutually exclusive criteria are specified, no results will be returned. For example, if a request specifies the Id property and the slot property, but the specified Relationship does not exist in the specified Slot, an empty results set will be returned.

Unless otherwise specified, all filters can by used together in the same request.

Available filters include:

- To limit results to the user's current Community, specify the enableCommunityFilter property as *true*.
- To return a specific Active Assembly Relationship, specify the Relationship ID in the Id property.
- To return Active Assembly Relationships in a specific Slot, specify the Slot name in the slot property.
- To return Active Assembly Relationships that specify a particular Template, specify the Template name in the template property.
- To return Active Assembly Relationships associated with a specific Owner Content Item, specify the ID of the Content Item in the Owner property. Note that this filter allows you to specify only one ID The limitToOwnerRevisions and limitToEditOrCurrentRevions flags dictate the results returned for this option:

Owner ID Type	limitToOwnerRevision	limitToEditOrCurrentRevision	Revision Associated with the Returned Relationships
Revision-specific	false	false	All Revisions
Revision-specific	true	false	Specified Revision

Revision-specific	false	true	The Edit Revision if the Content Item is checked out, otherwise the Current Revision.
Revision-specific	true	true	If the specified Revisions is the Edit Revision and the Content Item is checked out, all Relationships associated with the Edit Revision will be returned. If the specified Revisions is the Current Revision and the Content Item is checked out, then all Relationships associated with the Current Revision will be returned. Otherwise, no Relationships are returned.
Non-Revision-specific	false	false	All Revisions
Non-Revision-specific	true	false	The Edit Revision if the user currently has the Content Item checked out, otherwise the Current Revision.
Non-Revision-specific	false	true	The Edit Revision if the Content Item is checked out, otherwise the Current Revision.
Non-Revision-specific	true	true	The Edit Revision if the user is currently has the Content Item checked out, otherwise the Current Revision.

- To return Active Assembly Relationships associated with a Dependent Content Item, specify the ID of the Dependent Content Items in the Dependent property. Note that this filter allows you to specify multiple IDs.
- To return Active Assembly Relationships that specify a particular Folder, specify the path to the Folder in the folderPath property. Folder paths must being with double slashes ("//") and the first Node must be either Sites (//Sites/) or Folders (//Folders/).
- To return Active Assembly Relationships that specify a particular Site, specify the name of the Site in the site property.

- To return Active Assembly Relationships where the Owner is of a particular Content Type, specify the ID of the Owner Content Type in the OwnerContentType property. Note that this filter allows you to specify only one ID. The name of the Content Type can be included as well, but the ID is required. This filter cannot be used in conjunction with the DependentContentType filter.
- To return Active Assembly Relationships where the Dependent is of a particular Content Type, specify the IDs of the Dependent Content Types in the DependentContentType property. Note that this filter allows you to specify multiple names. The names of the Content Type can be included as well, but the ID is required. This filter cannot be used in conjunction with the OwnerContentType filter.
- To return Active Assembly Relationships of a specific Relationship Type, specify the names of the Relationship configuration in the Name property. Multiple names can be specified.
- To return Active Assembly Relationships that have specific property values, specify the properties and their values in the Property class. Multiple properties can be specified, as can multiple values for a specific property.
- To return Active Assembly Relationships based on whether they are system-defined Relationship Types or a user-defined Relationship Types, use the RelationshipType property. Specify *system* as the value of this property to return system-defined Active Assembly Relationship Types. Specify *user* as the value of this property to return user-defined Active Assembly Relationship Types. Any other value for this node, including a null value, is treated as if the value is *system*.
- To filter based on the object type of the Owner in the Relationship,specify the OwnerObjectType property. Currently, valid values are null, *Content Type* and *Folder*. Any other value will return an error. Note that the Owner in an Active Assembly Relationship must be a Content Item, so if you specify *Folder* as the value for this property, the results set will be empty.
- To filter based on the object type of the Dependent, specify the DependentObjectType property. Currently, valid values are null, *Content Type* and *Folder*. Any other value will return an error. Note that the Dependent in an Active Assembly Relationship must be a Content Item, so if you specify *Folder* as the value for this property, the results set will be empty.

PSAction

This Data Type is a container that contains one Action Menu entry.

PSAssemblyTemplate

This Data Type is a container that contains one Template Definition, including any bindings, references to any Sites or Slots associated with the Template, and the Template HTML.

PSAuditTrail

This Data Type is a container that contains the audit trail for a specific Content Item.

PSAutoTranslation

This Data Type is a container for one automatic translation configuration. These configurations are used to generate Translation Content Items automatically when a Content Item reaches a specified State. The IDs of the Content Type, Community, and Workflow are required, as is the name of the Locale. Optionally, the names of the Content Type, Community, and Workflow can be included as well.

PSChildEntry

This Data Type is a container for a child table. Each instance of this node contains one child table, including all entries in the table. The data in the container specifies the action to perform on the child entry.

The Id property specifies the child entry on which to operate. If this node has no value, a new child entry will be created on the Content Item. If this node has a value and the Action node has no value, the child entry is updated. Otherwise, the action specified in the Action node will be executed on the specified child.

Services that return this Data Type set the Action property to the value required for the SaveChildEntries method. If this Data Type is returned by a call to CreateChildEntries, the Action property will be set to *insert*. If this Data Type is returned by a call to LoadChildEntries, the Action property will be set to update. These values should not be changed.

If the values of the Id and Action properties conflict (for example, if the Id node has a value but the Action node is included with a value of *insert*), the server will return an error.

Use the ***Content/Items services*** (see page 18) to operate on the complete set of child tables in a Content Item. Use the ***Content/ItemChildren*** (see page 43) services to operate on individual tables.

PSCommunity

This Data Type is a container that contains one Community definition.

PSContentType

This Data Type is a container for a Content Type, including the metadata for the Content Type and the Content Editor used to edit Content Items of the Type.

PSContentTypeSummary

This Data Type is a container that contains a summary of the definition of one Content Type, including all fields and all child tables.

PSDisplayFormat

This Data Type is a container that contains one Display Format definition.

PSError

This Data Type is a container that contains all of the errors and results of a request that returns an error. For each error returned, the container includes:

- the error code, which can be used by the client to determine how to handle the error;
- an error message that can be displayed to the user; and
- a stack trace that can be used for logging and debugging.

PSField

This Data Type contains the data defining a specific field.

If the value of the field is text data, the PSFieldValue property stores the text. If the value of the field is binary data, the PSFieldValue property stores the field value ID.

If multiple values are available for the field, they are stored in the ValueChoices property.

PSFolder

This Data Type is a container for the data defining one Folder.

The optional AclEntries specifies the set of users, Roles, and virtual groups that have access to the Folder. This property can be used to modify the security setting for the Folder.

The optional Property property specifies a property for the Folder. It can be used to add properties to the Folder, to remove properties, or to change the value of a property. Multiple properties can be specified.

The optional Display Format property specifies the Display Format to be used with the Folder. It can be used to specify a Display Format to use to render the Folder or to change the default Display Format of the Folder.

The optional Community property specifies the Communities that have access to the Folder. If the value of this property is null, all Communities will have access to the Folder.

PSItem

This Data Type is a container for all data of a Content Item requested by a user.

The Id property contains the ID of the Content Item as a GUID. The ID may be a revision-specific ID.

The Fields class contains all requested Content Item fields. When loading a Content Item, the user controls which fields are loaded. If the ***LoadItems*** (see page 20) service is submitted with the FieldName parameter, only the fields specified will be returned. If the FieldName parameter is not included in the request, all fields are returned. The IncludeBinary parameter specifies whether to include binary fields in the returned Content Item.

The Children class contains all child tables. When loading a Content Item, the user controls whether child tables are returned. If the LoadItems service is submitted with the IncludeChildren flag, the child tables specified in the ChildName parameter will be returned. If the ChildName parameter is not included in the request, or if it is specified as null, all child tables will be returned.

The Slots class contains all of the Slots on the Content Item. For each Slot, the PSRelatedContent child contains the related Content Items in that Slot. When loading a Content Item, the user controls whether the related content is returned. If the LoadItems service is submitted with the IncludeRelated flag, the Related Content in the Slots specified in the Slot parameter will be returned. If the Slot parameter is not included with the request, or if it is specified as null, all related content will be returned.

The Folders class contains all Folder paths associated with the Content Item. This class is included when PSItem is returned by a LoadItems request that specifies the IncludeFolderPath parameter as *true*. If the class is included with PSItem when included with a SaveItems request, the Content Item will be added to all specified Folder paths.

Any property not included in this Data Type when submitted as part of an update request is skipped when processing the Content Item.

PSItemStatus

This Data Type is returned whenever the PrepareForEdit service is submitted. It is a container for the status data of a Content Item before it was prepared for editing and summarizes the actions performed on the Content Item to prepare it for edit.

PSItemSummary

This Data Type contains is returned by lookup services for Content Items and Folders. It contains the following data:

- ID
- whether the object is a Content Item or a Folder
- if the object is a Content Item, its Content Type
- if the object is a Content Item, the value of the sys_title field; if the object is a Folder, its name.
- a collection of the operations that the user is allowed to perform on the object when requested.

Most of the data needed to implement an alternative user interface client is available in this summary. All of the data required to look up the complete object is available in the summary.

PSKeyword

This Data Type is a container for one complete keyword definition, including all choices.

PSLocale

This Data Type is a container that contains one Locale definition.

PSLogin

This Data Type is a container that stores all the data returned by a user request to log in to the system, including all Roles into which the user can be authenticated, and all Communities and Locales available to the user.

PSRelatedItem

This Data Type is a container for a complete Active Assembly Relationship with one Dependent Content Item.

When modifying a Content Item, an Active Assembly Relationship can be created in one of two ways:

- The request can include the Content Item ID in the PSItem class.
- The PSItem child can include a completely new Content Item to create as the new related Content Item. In this case, one or more SearchField properties must also be included so the system can determine whether the submitted child Content Item already exists. The set of fields specified should form a unique key for the Content Item being inserted.

The PSItem class contains the data of the Dependent Content Item in the Relationship, while the PSAaRelationship class contains the Relationship instance.

Relationships will be ordered according to the sequence of PSRelatedItem classes submitted with the request. If existing relationships are submitted, they will be reordered as specified in the request.

Note that if the sys_revision field is included in the set of fields specified in the PSItem child, it will be ignored. Best practice is not to include this field when PSItem is contained by PSRelatedItem.

PSRelationship

This Data Type is a container for a specific instance of a Relationship of any Category other than Active Assembly.

PSRelationshipConfigSummary

This Data Type is a container that contains a summary of one Relationship Type configuration.

PSRelationshipFilter

This Data Type is a container that contains all parameters that can be used to filter Relationships returned when a user submits a request to lookup Relationships other than Active Assembly Relationships. Use the ***PSAaRelationshipFilter*** (see page 110) to filter Active Assembly Relationships.

A number of filter options are available. One request can include multiple filters, and only Relationships that meet all filter criteria specified (in other words, the intersection of the results sets) will be returned. For example, if the request specifies both the slot property and the template property, then any Relationship that is associated with the specified Slot and that uses the specified Template will be returned. If mutually exclusive criteria are specified, no results will be returned. For example, if a request specifies the Id property and the slot property, but the specified Relationship does not exist in the specified Slot, an empty results set will be returned.

Unless otherwise specified, all filters can by used together in the same request.

Available filters include:

- To limit results to the user's current Community, specify the enableCommunityFilter property as *true*.
- To return a specific Relationship, specify the Relationship ID in the Id property.
- To return Relationships associated with a specific Owner Content Item, specify the ID of the Content Item in the Owner property. Note that this filter allows you to specify only one ID The limitToOwnerRevisions and limitToEditOrCurrentRevions flags dictate the results returned for this option:

Owner ID Type	limitToOwnerRevision	limitToEditOrCurrentRevision	Revision Associated with the Returned Relationships
Revision-specific	false	false	All Revisions
Revision-specific	true	false	Specified Revision
Revision-specific	false	true	The Edit Revision if the Content Item is checked out, otherwise the Current Revision.
Revision-specific	true	true	If the specified Revisions is the Edit Revision and the Content Item is checked out, all Relationships associated with the Edit Revision will be returned. If the specified Revisions is the Current Revision and the Content Item is checked out, then all Relationships associated with the Current Revision will be returned. Otherwise, no Relationships are returned.
Non-Revision-specific	false	false	All Revisions
Non-Revision-specific	true	false	The Edit Revision if the user currently has the Content Item checked out, otherwise the Current Revision.

Non-Revision-specific	false	true	The Edit Revision if the Content Item is checked out, otherwise the Current Revision.
Non-Revision-specific	true	true	The Edit Revision if the user is currently has the Content Item checked out, otherwise the Current Revision.

- To return Relationships associated with a Dependent Content Item, specify the ID of the Dependent Content Items in the Dependent property. Note that this filter allows you to specify multiple IDs.
- To return Relationships that specify a particular Folder, specify the path to the Folder in the folderPath property. Folder paths must being with double slashes ("//") and the first Node must be either Sites (//Sites/) or Folders (//Folders/).
- To return Relationships that specify a particular Site, specify the name of the Site in the site property.
- To return Relationships where the Owner is of a particular Content Type, specify the ID of the Owner Content Type in the OwnerContentType property. Note that this filter allows you to specify only one ID. The name of the Content Type can be included as well, but the ID is required. This filter cannot be used in conjunction with the DependentContentType filter.
- To return Relationships where the Dependent is of a particular Content Type, specify the IDs of the Dependent Content Types in the DependentContentType property. Note that this filter allows you to specify multiple names. The names of the Content Type can be included as well, but the ID is required. This filter cannot be used in conjunction with the OwnerContentType filter.
- To return Relationships of a specific Relationship Type, specify the names of the Relationship Types in the Name property. Multiple names can be specified.
- To return Relationships that have specific property values, specify the properties and their values in the Property class. Multiple properties can be specified, as can multiple values for a specific property.
- To return Relationships based on whether they are system-defined Relationship Types or a user-defined Relationship Types, use the RelationshipType property. Specify *system* as the value of this property to return system-defined Relationship Types. Specify *user* as the value of this property to return user-defined Relationship Types. Any other value for this node, including a null value, is treated as if the value is *system*.
- To filter based on the object type of the Owner in the Relationship,specify the OwnerObjectType property. Currently, valid values are null, *Content Type* and *Folder*. Any other value will return an error. Note that the Owner in an Active Assembly Relationship must be a Content Item, so if you specify *Folder* as the value for this property, the results set will be empty.
- To filter based on the object type of the Dependent, specify the DependentObjectType property. Currently, valid values are null, *Content Type* and *Folder*. Any other value will return an error. Note that the Dependent in an Active Assembly Relationship must be a Content Item, so if you specify *Folder* as the value for this property, the results set will be empty.

PSRevisions

This Data Type is a container that contains all the Revisions of a specific Content Item.

The EditRevision property contains the Revision-specific ID of the edit Revision of the Content Item. The edit Revision is usually the same as the current Revision. The edit Revision and the current Revision are different only when a Content Item is actively being edited.

The CurrentRevision property contains the Revision-specific ID of the current Revision of the Content Item.

A set of PSRevision properties contains the Revision-specific IDs of all Revisions of the Content Item. If the Content Item is new and does not have a revision history, this property may be null.

PSRole

This Data Type is a container that contains one Role definition.

PSSearch

This Data Type is a container for all the parameters used to search for Content Items. For details about the search options available using the full-text search engine, see the topic "Searching for Content Using the Full-text Search Engine" in the Content Explorer Help.

- If your system uses the full-text search engine, the FullTextQuery property specifies the query text. If your system does not have the full-text search engine enabled and you submit this property, the system returns an error.
- If your system uses the full-text search engine, the default search mode is concept search, which returns results that match the conceptual meaning of the search. You can control how closely the results match the concept by submitting the PSSearchProperty *expansionlevel* with one of the following values (examples assume the submitted string is "catch"):

Value	Expansion Level
1	Exact match (only exact matches on the specified string, or nearly identical variants, such as "catches", will be returned)
3	Simple variations (Content Items that include "caught) would be returned.
4	Most strongly related concepts (Content Items that include words such as "catchable" and "catching" would be returned. This is the default expansion level.)
5	Strongly related concepts (Content Items that include words such as "bag" or "arrest" would be returned.)
7	Weakly related concepts (Content Items that include words such as “capture”, “secure”, “seize”, or “nail” would be returned.)

If your system does not have the full-text search engine enabled and you submit this property, the system returns an error.

- If your system uses the full-text search engine and you want to execute a boolean search, submit the PSSearchProperty *querytype* with a value of *16*. A boolean search uses boolean operators entered in the search string to generate search results. If your system does not have the full-text search engine enabled and you submit this property, the system returns an error.
- If your system uses the full-text search engine and you want to execute a pattern search, submit the PSSearchProperty *querytype* with a value of *64*. A pattern search returns results that match the pattern of the submitted string (for example, if the submitted string were "catalog", Content Items that include the variant spelling "catalogue" would be returned). If your system does not have the full-text search engine enabled and you submit this property, the system returns an error.
- If your system does not have the full-text search engine enabled, you must use the database search engine. To use this engine, submit the PSSearchField class.
    - The Value property specifies the value you want to search for in the specified field.
    - The Operator property specifies the operator to pass to the database search engine. The following options are available:
        - equal
        - notequal
        - lessthan
        - lessthanequal
        - greaterthan
        - greaterthanequal
        - isnull
        - isnotnull
        - in
        - notin
        - like
        - notlike
    - The Connector property specifies how multiple PSSearchFields will be connected. Options are *and* and *or*.
- You can narrow the search results in the following ways:
    - To limit the results to a specific Folder, include the FolderFilter property with the path to the Folder whose contents you want to search. To include Subfolders in the search, set the includeSubfolders property to *true*.
    - To limit the results to Content Items of a specific Content Type, you can either

        - include the ContentType property with the name of the Content Type as the value (if you choose this option, you must create a PSSearchParamsContentType object in your code; you can also specify the ContentType property alone, which will return all Content Items of the specified Content Type, but you still must create a PSSeachItemsContentType object in your code); or
        - include an instance of PSSearchParams specifying the search field as *sys_contenttypeid*.
    - To limit the search results to Content Items with a specific value in the sys_title field, you can either:
        - include the Title property with the title string as the value; if you choose this option (you must create a PSSearchParamsTitle object in your code); or
        - include an instance of PSSearchParams, specifying the search field as *sys_title*.
    - To limit the search results to Content Items that also contain another string, submit the PSSearchProperty *bodyfilter* with the filter string as the value.
- The search results always include the following fields:
    - sys_contentid
    - sys_contenttypeid
    - sys_title
    - sys_workflowappid
    - sys_contentstateid
    - sys_contentcheckoutusername

  To include other fields in the results, include the SearchResults class with the set of fields you want to include in the results.
- To specify the maximum number of results, by passing the following:
    - PSSearchProperty *startindex* with a value of *0*.
    - PSSearchProperty *endindex* with a value of the maximum results you want.
- To search by Content ID or a Folder ID, you must use the legacy ID, but Web services returns the GUID. The legacy ID is stored in the lower 32 bits of the GUID and can be obtained by masking the desired bits in the GUID: GUID & 0xFFFFFFFF.

PSSearchDef

This Data Type is a container that contains one pre-defined Search definition or one user-defined saved Search definition.

PSTemplateSlot

This Data Type is a container that contains on Slot definition, including the metadata describing the Slot all allowed content in the Slot (combination of Content Type and Template) and any Content Finder arguments defined for the Slot.

PSViewDef

This Data Type is a container that contains one View definition.

PSWorkflow

This Data Type is a container that contains one complete Workflow definition.

CHAPTER 8

Common Faults

This chapter describes faults returned by more than one Rhythmyx Web Service. (Any fault returned only by one service is documented with that service.)

If a service returns multiple objects for a single request, a returned fault includes a result for each object. The result includes success or error information for each object in the same order that the objects were requested.

Each error result includes:

- the error code, which can be used by the client to determine how to handle the error;
- an error message that can be displayed to the user; and
- a stack trace that can be used for logging and debugging.

Axis Fault

This fault is returned for any unexpected error that occurs when processing any Web Service.

This fault is typically returned if the SOAP message is invalid.

To resolve the fault, debug your code to ensure that the SOAP messages are being formed correctly.

This fault can be returned by any Rhythmyx Web Service.

PSContractViolationFault

This fault is returned when any request violates the requirements of a specific service.

Typically this fault is returned if a required parameter is missing or if a submitted parameter value does not match the requirements of the service.

To resolve the fault, ensure that all required parameters are submitted with each request and that the values being submitted for each parameter match the requirements of the service.

This fault is returned by the following services:

- CreateItem
- FindItem
- LoadItem
- SaveItem
- DeleteItem
- ViewItem
- PrepareForEdit
- ReleaseFromEdit
- NewCopies
- NewPromotableVersions
- NewTranslations
- FindRevisions
- PromoteRevisions
- CheckinItems
- CheckoutItems
- GetAssemblyUrls
- CreateChildEntries
- LoadChildEntries
- SaveChildEntries
- DeleteChildEntries
- ReorderChildEntries
- AddContentRelations
- LoadContentRelations
- SaveContentRelations
- DeleteContentRelations
- ReorderContentRelations
- FindDependents
- FindOwners
- LoadFolders
- SaveFolders
- AddFolders
- AddFolderTree
- FindFolderChildren
- AddFolderChildren
- RemoveFolderChildren
- MoveFolderChildren
- FindFolderPaths
- FindPathIds
- SwtichCommunity
- SwitchLocale
- CreateRelationship
- LoadRelationship
- SaveRelationship
- DeleteRelationship
- FindDependents
- FindOwners
- LoadRelationshipTypes
- LoadAuditTrails
- TransitionItems
- GetAllowedTransitions
- Login
- Logout
- RefreshSession
- FilterByRuntimeVisibility

PSErrorsFault

This fault is returned by services that act on multiple input objects but do not return a result if the service is executed successfully. If the processing fails on any of the input objects, this fault is returned. It is a container for an array of results of the request. The array contains success or PSError information for each input object. The results are not necessarily returned in the order specified in the request, but the IDs of the service requests are included so the error results can be matched to the requests.

As a container, this fault does not have a specific resolution. Instead, resolve any of the contained faults.

This fault can be returned by the following services:

- CheckinItems
- CheckoutItems
- DeleteChildEntries
- DeleteFolders
- DeleteItems
- ReorderChildEntries
- SaveChildEntries

PSErrorResultsFault

This fault is returned by services that act on multiple input objects and return a result for each input object. If the processing fails on any of the input objects, this fault is returned. It is a container for an array of results of the request. The array contains the return object for each input object processed successfully and an instance of PSError for each object for which processing fails. The results not are included in the order in which the objects were submitted in the request, but the IDs of the services calls are listed so they can be matched.

This fault can be returned by the following services:

- AddContentRelations
- AddFolder
- AddFolderChildren
- AddFolderTree
- DeleteContentRelations
- DeleteCustomRelationship
- DeleteFolders
- LoadAuditTrails
- LoadContentRelations
- LoadFolder
- LoadItems

- Login
- MoveFolderChildren
- NewCopies
- NewPromotableVersions
- NewTranslations
- PrepareForEdit
- ReleaseFromEdit
- ReorderContentRelations
- SaveContentRelations
- SaveCustomRelationship
- SaveFolders
- SaveItems
- TransitionItems
- ViewItems

PSInvalidSessionFault

All Web Service requests must include the session ID in the request header. The session ID is derived from the ***PSLogin*** (see page 116) response returned by the server when the user logs in.

This fault is returned if the session has timed out, or is not valid for some other reason. Typically, the user must log in again for a new session ID.

This fault is returned by all services except Login.

PSNotAuthorizedFault

This fault is returned if the user attempts to execute a service that they are not authorized to use based on the current session ID and the access control list (ACL) of the objects the user has specified in the request. For example, if the user issues a LoadFolders request, but their Role is not in the ACL for the specified Folder, this fault will be returned. It would also be returned if the user issues a PrepareForEdit request for a Content Item but the user is not a Role assigned to the current Workflow State of that item.

In some cases, you may be able to address this fault. For example, if the fault is returned due because the user is not in the access control list, you can address the fault by adding the user's Role to the ACL. In other cases, the user may be able to address it; if the fault is returned because the user is not logged in to the Community of the Content Item, they may be able to change Communities. In a few cases, however, the fault may indicate that they system is functioning correctly. For example, if a user issues a PrepareForEdit request but they are not in a Role assigned to the current Workflow State of the Content Item, this fault will be returned, but that is correct behavior because the user should be denied access to the Content Item in its current State.

This fault may be returned by the following services:

- AddContentRelations
- CreateItem
- DeleteContentRelations
- LoadAssemblyTemplates
- LoadAutoTranslations
- LoadCommunities
- LoadContentRelations
- LoadContentTypes
- LoadCustomRelationship
- LoadFolders
- LoadKeywords
- LoadLocales
- LoadRelationshipTypes
- LoadRoles
- LoadSlots
- LoadWorkflows
- SaveContentRelations

PSNotFoundFault

This fault is returned if a system component specified in a request cannot be found.

PSUnknownChildFault

This fault is returned by services in the content/Item Children category if the Child table specified in the request does not exist. Typically, it results from the submission of incorrect data, such as misspelling the name of a child Field Set. Check the submitted data to ensure that the names of the child Field Sets being submitted match the names of the child Field Sets defined for the Content Types in the request.

This fault may be returned by the following services:

- CreateChildEntries
- DeleteChildEntries
- LoadChildEntries
- ReorderChildEntries
- SaveChildEntries

CHAPTER 9

Rhythmyx Web Services Behaviors

This chapter describes important behaviors of Rhythmyx Web Services.

Prepare for Edit

The system evaluates the current status of all requested Content Items and returns a PSItemStatus for each Content Item. Whether a particular Content Item is actually prepared for edit depends on its current status.

Item State	Action
Checked out by requestor	The Content Item is returned in its current status.
Checked out by someone else	If the requestor has admin rights,the system offers them the option of overwriting the current edit. If the requestor chooses this option, the Content Item is checked out to them and returned. If the user does not have admin rights, the system takes no action and returns an error to the requestor informing them that the Content Item is currently checked out to another user.
Checked in and in a non-Public State	The Content Item is checked out to the requestor and then returned to the requestor.
Checked in and in a Public State	The Content Item is Transitioned to the Quick Edit State, then checked out to the requestor, then returned to them. The Content Item is Transitioned using the first Transition defined for the current State of the Content Item that targets the Quick Edit State. If No Transition to the Quick Edit State exists for the current State of the Content Item, the request fails and a fault is returned to the requestor.

If the Content Item can be returned to the requestor in its current status, the it is prepared for edit, and returned with a PSItemStatus. The PSItemStatus must be submitted with the release from edit request for the Content Item to ensure that the Content Item returns to its status before it was prepared for edit.

Release from Edit

All Content Items specified in the request are released from edit. The release from edit request must include the ***PSItemStatus*** (see page 115) returned with the prepare for edit.

All Content Items specified in the request are returned to their previous State, which is included in the PSItemStatus. The first default Transition that leads to the State specified in PSItemStatus is used. If no default Transition is found, the first Transition from the current State of the Content Item to the State specified in PSItemStatus will be used. If no Transition is available from the current State of the Content Item to the State specified in PSItemStatus, no Transition will be executed.

Multi-Object Operations

Several Rhythmyx Web Services can operate on multiple objects. If a request is submitted for multiple objects, the system will attempt to process all of them. If all specified objects are processed successfully, the normal the normal results will be returned. If the processing for any object fails, all remaining objects will be processed, but instead of the normal output, a fault will be returned. The fault includes success or error information for each object in the request.

CHAPTER 10

Rhythmyx Web Services Samples

To illustrate the implementation of Rhythmyx Web Services, Percussion Software provides a simple loader application. Both Java and .NET (C#) versions of the application are available.

The loader illustrates a selected set of Rhythmyx functionality that can be accessed using Web Services. It reads Content Item data out of an XML file (`LoaderData.xml`) and either creates or updates Content Items based on that data. The location of the data file, the target Folder to be updated, and other properties, are defined in a properties file, `Loader.xml`.

Running the Loader

A compiled sample loader is provided ready to run with sample data. To run the loader, change to the directory `<Rhythmyxroot>/WebServices/<version>/sample/loader/Java` and run either `\bin\run.bat` (in a Windows environment) or `./bin/run.sh` (in a Unix or Linux environment). Note that the loader must be run from the Java directory.

Java Samples

The Java version of the loader is stored in the directory `<Rhythmyxroot>/WebServices/<version>/sample/loader/Java`. This directory contains the Eclipse project files (`.project` and `.classpath`) and Ant build script (`build.xml`).

It also contains the following subdirectories:

- `/bin`

  `This directory contains two executable files:`

  - `run.bat`

    This file is used to run the loader in a Windows environment.

  - `run.sh`

    This file is used to run the loader in a Unix/Linux environment.

- `/build/classes`

  This directory contains the Java classes compiled from the code in the `/src` and `/webservices/generated` directories. When the Eclipse project is compiled, the properties file (`Loader.xml`) and data file (`LoaderData.xml`) are copied to this directory. The loader expects the properties file to be in this directory. This is also the default location specified for the data file in the properties file, but the data file can be updated to specify a different location.

- `/lib`

  This directory contains all third party `.jar` files needed to run the loader:

  - activation.jar
  - axis-ant.jar
  - axis.jar
  - commons-discovery-0.2.jar
  - commons-logging-1.0.4.jar
  - jaxrpc.jar
  - mail.jar
  - saaj.jar
  - wsdl4j-1.5.1.jar

- `/src`

  This directory contains the loader source code.

- `/webservices/generated`

  This directory contains the proxy code generated by Axis 1.3

The loader consists of three classes:

- `PSWsUtils`

  This class consists of general utility methods used in the loader.

- `PSFileUtils`

  This class consists of utility methods specifically used to read the property and data files used by the loader

- `PSLoader`

  This class consists of the loader methods.

Login

The method `PSLoader.login()` relies on a set of methods in `PSWsUtils` both to log in and to create proxies for the different categories of Rhythmyx Web Services. In the sample, proxies are created for the security, system, and content services.

The method `PSWsUtils.login()` logs in to Rhythmyx using the location and authentication data defined in the loader properties file (`Loader.xml`).

Maintaining Sessions

Under Axis 1.3, different proxies cannot share the same JBoss session (JSESSION); each proxy must create and maintain its own JSESSION. To set the JSESSION, all methods that create a proxy maintain the session in Axis using the method `org.apache.axis.client.Stub.setMaintainSession`:

```
binding.setMaintainSession(true);
```

All proxies can share the same Rhythmyx session, which is maintained in the SOAP header. The proxies all use the method PSWsUtils.setRxSessionHeader to generate the SOAP header:

```
private static void setRxSessionHeader(Stub binding, String
   rxSession)
{
   binding.clearHeaders();
   binding.setHeader("urn:www.percussion.com/6.0.0/common",
      "session", rxSession);
}
```

Specifying the Host, Port, and Protocol for a Web Services Client

Proxy addresses in stub code are generated using the default connection data provided with the WebServices WSDLs.

Host:	localhost
Port:	9992
Protocol:	http

The Web Services client needs to update at least the host and port to run remotely or contact a Rhythmyx server on a port other than the default. You may need to update the protocol to enable secure communication between the server and client.

The class `PSWsUtils` illustrates how to update connection data. In this case, the correct address for the proxy is derived from the loader properties file (`Loader.xml`).

```
String address = getNewAddress(locator.getcontentSOAPAddress());
locator.setcontentSOAPEndpointAddress(address);
```

Creating a Content Item

Rhythmyx Web Services includes a createItems service, but this service merely creates a blank instance of a Content Item of the specified Content Type. The Content Item does not actually exist until it has been populated with data and saved to the Repository. The method `PSLoader.createItem` illustrates this minimal processing plus additional processing required to make the Content Item accessible to Publishing. This method performs the following processing:

1. Creates the blank Content Item instance.
   ```
   PSItem item = PSWsUtils.createItem(m_contService,
   (String)m_props.get(CONTENT_TYPE));
   ```
2. Populates the blank Content Item with field data.
   ```
   setItemFields(item, fields);
   ```
3. Saves the Content Item to the Repository. The Content Item exists in Rhythmyx at this point.
   ```
   long id = PSWsUtils.saveItem(m_contService, item);
   ```
4. Checks in the Content Item.
   ```
   PSWsUtils.checkinItem(m_contService, id);
   ```
5. Transitions the Content Item to Public. The Content Item is eligible to be published at this point.
   ```
   PSWsUtils.transitionItem(m_sysService, id, "DirecttoPublic");
   ```
6. Associates the Content Item with a Folder. If Site Folder Publishing is in use, this action is necessary to ensure that it can be published. In any case, a Content Item should always be associated with a Folder so users can find it easily.
   ```
   String path = m_props.getProperty(TARGET_FOLDER);
   PSWsUtils.addFolderChildren(m_contService, path, new
   long[]{id});
   ```

For details, see `PSLoader.createItem`:

Modifying a Content Item

Modifying a Content Item also requires several steps, as illustrated in the method `PSLoader.updateItem`:

1. Prepare the Content Item for editing.
   ```
   PSItemStatus status = PSWsUtils.prepareForEdit(m_contService, id);
   ```
2. Load the Content Item.
   ```
   PSItem item = PSWsUtils.loadItem(m_contService, id);
   ```

3. Modify the Content Item data.
```
setItemFields(item, fields);
```
4. Save the modified Content Item.
```
PSWsUtils.saveItem(m_contService, item);
```
5. Release the Content Item from edit.
```
PSWsUtils.releaseFromEdit(m_contService, status);
consoleMessage("Updated item: " + fields.get("sys_title"));
```

For details, see `PSLoader.updateItem`.

Uploading a Binary Attachment

When implementing Rhythmyx Web Services in Java, binary files are handled as attachments using SOAP with Attachments (SwA). The method `PSLoader.setFileItemFields` and `PSLoader.addAttachment` illustrates how to build a SOAP message with a binary attachment.

1. The method `PSLoader.setFileItemFields` builds the SOAP message. To build the binary data field, it calls the method `PSLoader.addAttachment`.
```
String attachmentId = addAttachment(m_contService, url);
newValue.setAttachmentId(attachmentId);
field.setPSFieldValue(new PSFieldValue[] { newValue });
```
This call passes the location of the binary file to `PSLoader.addAttachment`.
2. The method `PSLoader.addAttachment` attach the binary file at the specified location to the SOAP message and returns the attachment ID.
```
DataHandler handler = new DataHandler(new
URLDataSource(attachment));

AttachmentPart part = new AttachmentPart(handler);
binding.addAttachment(part);

return part.getContentId();
```
3. This ID is returned to the method `PSLoader.setFileItemFields`, which saves the ID to the `item` field in the SOAP message so the server can retrieve the attachment.

Retrieving a Binary Attachment

The method `PSLoader.retrieveBinaryData` illustrates how to retrieve a binary attachment from a SOAP message.

1. The incoming message is read to find the value of the `item` field, which identifies the attachment.
```
PSFieldValue[] values = field.getPSFieldValue();
attachmentId = values[0].getAttachmentId();
```
2. The attachment whose ID matches the value of the `item` field is found:
```
for (Object attachment : attachments)
{
   AttachmentPart part = (AttachmentPart) attachment;
   if (part.getContentId().equals(attachmentId))
```

```
      {
         reader = (InputStream) part.getContent();
         break;
      }
   }
```

3 The attachment that matches is read out.

```
byte[] content = new byte[reader.available()];
reader.read(content);
return content;
```

Error Handling in Java

The loader method of the class PSLoader illustrates a example approach to error handling in Java:

```
{
   PSLoader loader = new PSLoader();

   try
   {
      loader.readInputData();
      loader.login();
      loader.createTargetFolder();
      loader.uploadItems();
      loader.uploadDataFile();
      loader.logout();
   }
   catch (PSNotAuthenticatedFault e)
   {
      loader.consoleMessage("Caught PSNotAuthenticatedFault: "
            + e.getErrorMessage());
   }
   catch (PSContractViolationFault e)
   {
      loader.consoleMessage("Caught PSContractViolationFault: "
            + e.getErrorMessage());
   }
   catch (RemoteException e)
   {
      loader.consoleMessage("Caught RemoteException: "
            + e.getMessage());
   }
   catch (Exception e)
   {
      e.printStackTrace();
   }
}
```

Updating Generated Proxy Code

From time to time, Percussion Software may modify Rhythmyx Web Services WSDL files. Updated WSDL files will be in the directory `<Rhythmyxroot>/WebServices/<version>/Design`, where <version> is the new version in which the WSDL files changed. For example, suppose the WSDL files are updated in Version 7.5 of Rhythmyx. The new WSDL files will be located in the directory `<Rhythmyxroot>/WebServices/7.5.0/Design`.

To take advantage of these updates, you will need to update your generated proxy code based on the new WSDL files. Update the WSDL references in your project file to the location of the upgraded WSDL files and regenerate the proxies. You can access WSDL files using a URL. For example, to access the assembly WSDL, use the URL `http://localhost:9992/Rhythmyx/webservices/assemblySOAP?wsdl` where localhost is the name or IP address of the Rhythmyx server and 9992 is the Rhythmyx port.

C# Samples

The C# (.NET) version of the loader is stored in the directory `<Rhythmyxroot>\WebServices\<version>\sample\loader\csharp`. This directory contains all files specific to the C# version of the loader. It also contains the C# project file (`Loader.csproj`) and Solutions file (`Loader.sln`). Either file can be used to load the project into Visual Studio 2005, which was used to develop the code.

The loader consists of three classes:

- `PSWsUtils`

  This class consists of general utility methods used in the loader.

- `PSFileUtils`

  This class consists of utility methods specifically used to read the property and data files used by the loader

- `PSLoader`

  This class consists of the loader methods.

The properties file (`Loader.xml`) must be in the same directory as the loader executable (`Loader.exe`). By default, both are stored in `\csharp\bin\Debug`, as is the data file, `LoaderData.xml`.

Login

The method `PSLoader.Login()` relies on a set of methods in `PSWsUtils` both to log in and to create proxies for the different categories of Rhythmyx Web Services. In the sample, proxies are created for the security, system, and content services.

The method `PSWsUtils.Login()` logs in to Rhythmyx using the connection and authentication data defined in the loader properties file (`Loader.xml`), and establishes a Rhythmyx session:

Maintaining Sessions

In .NET, all proxies can use the same JBoss session (JSESSION) and the same Rhythmyx session. The JSESSION is maintained in a cookie by the cookie container instance created by the security proxy (`PSWsutils.GetSecurityService`). The same cookie container instance is used by all other proxies.

```
CookieContainer cookie = new CookieContainer();
securitySvc.CookieContainer = cookie;
```

The Rhythmyx session is maintained in the authentication header. The same instance of the authentication header is used by all proxies.

```
securitySvc.PSAuthenticationHeaderValue = new PSAuthenticationHeader();
securitySvc.PSAuthenticationHeaderValue.Session = rxSession;
```

Specifying the Host, Port, and Protocol for a Web Services Client

Proxy addresses in stub code are generated using the default connection data provided with the WebServices WSDLs.

Host: localhost

Port: 9992

Protocol: http

The Web Services client needs to update at least the host and port to run remotely or contact a Rhythmyx server on a port other than the default. You may need to update the protocol to enable secure communication between the server and client.

The class `PSWsUtils` illustrates how to update connection data. In this case, the correct address for the proxy is derived from the loader properties file (`Loader.xml`).

```
contentSOAP contentSvc = new contentSOAP();
contentSvc.Url = GetNewAddress(contentSvc.Url);
```

Creating a Content Item

Rhythmyx Web Services includes a createItems service, but this service merely creates a blank instance of a Content Item of the specified Content Type. The Content Item does not actually exist until it has been populated with data and saved to the Repository. The method `PSLoader.CreateItem` illustrates this minimal processing plus additional processing required to make the Content Item accessible to Publishing. This method performs the following processing:

1. Creates the blank Content Item instance.
   ```
   PSItem item = PSWsUtils.CreateItem(m_contService,
   m_props[CONTENT_TYPE]);
   ```
2. Populates the blank Content Item with field data.
   ```
   SetItemFields(item, fields);
   ```
3. Saves the Content Item to the Repository. The Content Item exists in Rhythmyx at this point.
   ```
   long id = PSWsUtils.SaveItem(m_contService, item);
   ```
4. Checks in the Content Item.
   ```
   PSWsUtils.CheckinItem(m_contService, id);
   ```
5. Transitions the Content Item to Public. The Content Item is eligible to be published at this point.
   ```
   PSWsUtils.TransitionItem(m_sysService, id, "DirecttoPublic");
   ```
6. Associates the Content Item with a Folder. If Site Folder Publishing is in use, this action is necessary to ensure that it can be published. In any case, a Content Item should always be associated with a Folder so users can find it easily.
   ```
   String path = m_props[TARGET_FOLDER];
   PSWsUtils.AddFolderChildren(m_contService, path, new long[]{id});
   ```

For details, see `PSLoader.CreateItem`:

Modifying a Content Item

Modifying a Content Item also requires several steps, as illustrated in the method `PSLoader.UpdateItem`:

1. Prepare the Content Item for editing.
   `PSItemStatus status = PSWsUtils.PrepareForEdit(m_contService, id);`
2. Load the Content Item.
   `PSItem item = PSWsUtils.LoadItem(m_contService, id);`
3. Modify the Content Item data.
   `SetItemFields(item, fields);`
4. Save the modified Content Item.
   `PSWsUtils.SaveItem(m_contService, item);`
5. Release the Content Item from edit.
   `PSWsUtils.ReleaseFromEdit(m_contService, status);`

For details, see `PSLoader.UpdateItem`.

Uploading a Binary File

In the .NET environment, the loader uploads binary files as base-64-encoded strings, as illustrated in the method `PSLoader.SetFileItemFields`:

```
field.PSFieldValue = new PSFieldValue[] { newValue };
```

Retrieving a Binary File

The method PSLoader.RetrieveBinaryData illustrates how binary files are retrieved and reconverted from base-64-encoded strings back to binary format:

```
byte[] binaryData = Convert.FromBase64String(value.RawData);
return binaryData;
```

Error Handling in C#

The loader method of the class PSLoader illustrates a example approach to error handling in C#, including how to process the returned XML document for detailed information (in this case, the error messages and the IDs of the successful objects).

```
static void Main(string[] args)
{
  PSLoader loader = new PSLoader();

    try
    {
      loader.ReadInputData();
      loader.Login();
      loader.CreateTargetFolder();
      loader.UploadItems();
      loader.UploadDataFile();
```

```
        loader.Logout();
      }
      catch (SoapException e)
      {
        IPSSoapFault fault = PSFaultFactory.GetFault(e);

        loader.ConsoleMessage("Caught Exception: " +
fault.GetFaultName());
           loader.ConsoleMessage("Error Message: " + fault.GetMessage());

        if (fault.GetFaultName() == "PSErrorsFault" ||
        fault.GetFaultName() == "PSErrorResultsFault")
        {
          PSErrorsFault errorsFault = (PSErrorsFault)fault;
          int errorCount = errorsFault.getErrorMessages().Count;
          int successCount = errorsFault.getSuccessIds().Count;
          loader.ConsoleMessage("There are " + errorCount + " error
messages.");
          loader.ConsoleMessage("There are " + successCount + " successful
ids.");
        }
      }
}
```

Updating Generated Proxy Code

From time to time, Percussion Software may modify Rhythmyx Web Services WSDL files. Updated WSDL files will be in the directory `<Rhythmyxroot>/WebServices/<version>/Design`, where <version> is the new version in which the WSDL files changed. For example, suppose the WSDL files are updated in Version 7.5 of Rhythmyx. The new WSDL files will be located in the directory `<Rhythmyxroot>/WebServices/7.5.0/Design`.

To take advantage of these updates, you will need to update your generated proxy code based on the new WSDL files. Update the WSDL references in your project file to the location of the upgraded WSDL files and regenerate the proxies.