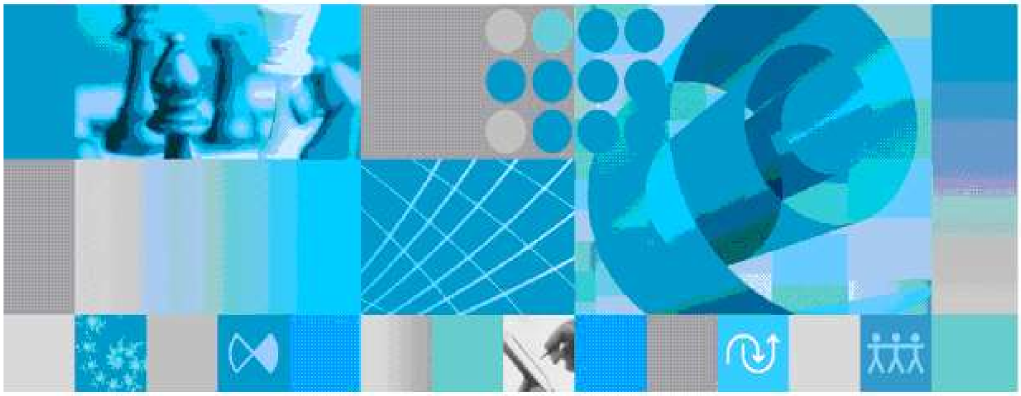
Rational. Focal Point

IBM



Web Services API version 2.0 Reference Manual

Note: Before using this information and the product it supports, read the information in the Notices section.

Table of contents

Rational Focal Point Web Services API

This document describes version 2.0 of the Web Services API for IBM® Rational® Focal Point™ versions 6.5 and later.

You can refer to the SOAP Web Services API examples to learn how to use the APIs for Rational Focal Point.

The SOAP Web Services API examples are available in Rational Focal Point install directory\apiexamples\JavaWebServices2Example.zip. An administrator having access to the application server on which Rational Focal Point in installed can provide you with the example package. For more information about the source files and the process to compile the code, see the readme file in the example package.

.

WSDL

The Rational Focal Point Web Services API uses Web Services Description Language (WSDL) to define services. The Web Services API uses WSDL version 1.1.

You can download the current WSDL from a Rational Focal Point server at http://*your_Rational_Focal_Point_server_URL*/fp/services2/FPServices?wsdl.
For example, http://focalpoint.example.com/fp/services2/FPServices?wsdl.

Data types are defined by using an included XML Schema. For a list of the data types, see the "Schema guide" section. This document describes the data types, and the XML Schema describes their details. You need both documents to fully understand the Rational Focal Point Web Services API.

The WSDL uses the document/literal wrapped pattern. For more information about that pattern, see "Which style of WSDL should I use?" at http://www.ibm.com/developerworks/webservices/library/ws-whichwsdl/.

SOAP

The Rational Focal Point Web Services API uses SOAP to call the services. The server uses SOAP version 1.1.

The URL to the SOAP endpoint is http://*your_Rational_Focal_Point_server_URL*/fp/services2/FPServices. For example, http://focalpoint.example.com/fp/services2/FPServices. You can also find the URL through the WSDL.

Faults

If an error occurs, a service might need to return a SOAP fault instead of a value. A service might return a fault because of a user error, such as referencing an element that does not exist, or because of an internal server error. Both this document and the WSDL describe the faults.

Security

When SOAP messages are sent over Hypertext Transfer Protocol (HTTP), they are sent in clear text, so that anyone who can access the traffic can read the sent data, for example a user name and password.

For security reasons, use Hypertext Transfer Protocol Secure (HTTPS) instead of HTTP. When you use HTTPS, all communication with the Rational Focal Point server is encrypted. For a description of how to set up HTTPS for Rational Focal Point, see the Installation Guide in the Rational Focal Point Information Center at http://www.ibm.com/support/docview.wss?uid=swg27016559.

Aliases and resources

In Rational Focal Point, resources are identified by aliases. For example, each workspace can be identified by a unique alias. The following resources can have aliases: workspaces, modules, views, elements, and attributes. Modules and views are grouped and are collectively known as domains. Resources are assigned a default random alias, which you can later change. To edit workspace, domain, and attribute aliases in each workspace, click **Configure** > **Alias**.

An alias is a string that can contain any Unicode character; however, avoid using special or invisible characters, such as line breaks.

When you export and import a workspace, aliases are preserved for domains, attributes, and workspace, if possible.

Elements

Aliases for elements are unique for a Rational Focal Point installation. Two elements cannot use the same alias.

When you create an element by using addElement, addElements, writeElement, or writeElements, you can assign element alias. However, you cannot change the alias. If you do not supply an alias when you create an element, a random alias is generated. New aliases are generated when a module or workspace is imported.

Domains

Modules and views are collectively known as *domains*. A domain can be a module or a view. Within a workspace, aliases for domains are unique. For example, a module in workspace A and a view in workspace B can both use the alias *requirements,* but they cannot be in the same a workspace. A domain is uniquely identified by using a workspace alias and a domain alias.

When a domain is created, you can assign its alias. A workspace administrator can later change the alias. When a workspace is imported, domain aliases are preserved.

Workspaces

Aliases for workspaces are unique within your instance of Rational Focal Point. Two workspaces cannot have the same alias.

When a workspace is created, you can assign its alias. A workspace administrator can later change the alias. When a workspace is imported, the alias is preserved unless another workspace uses the same alias.

Attributes

Aliases for attributes are unique within a module. The same attribute can exist in different modules, but two attributes in the same module cannot have the same alias. For example, a workspace might have two modules with the aliases requirements and defects, and each module has the estimated_time attribute alias. Another workspace contains a requirements module with an estimated_time attribute. An attribute is uniquely defined by its workspace alias, module alias, and attribute alias.

When an attribute is created, you can assign its alias. A workspace administrator can later change the alias. When a workspace or module is imported, aliases are preserved.

Services

The Rational Focal Point Web Services API calls the following services. Each service is defined in this document.

Name	**Short description**	**Rational Focal Point version**
addElement	Creates an element	6.2 and later
addElements	Creates several elements in one batch	6.2 and later
authenticate	Authenticates, or logs in, a user	6.2 and later
endSession	Invalidates a session and releases licenses (logs out)	6.2 and later
getAttributeDefinitions	Retrieves a list of attribute definitions for a specified domain in a workspace	6.2 and later

getAttributeHistory	Retrieves the change history for an attribute of an element	6.2 and later
getChangedElements	Retrieves the aliases for elements that have changed within a specified time range	6.2 and later
getElement	Retrieves an element and its attributes	6.2 and later
getElementHistory	Retrieves the change history for an element	6.2 and later
getElements	Retrieves elements for a workspace, domain, or both	6.2 and later
getElementsByAlias	Retrieves elements from a list of element aliases	6.2 and later
getExternalLink	Retrieves a URL to an element.	6.3 and later
getFileContent	Retrieves the file data contents of a file attribute	6.2 and later
getHistory	Retrieves the change history for a domain	6.2 and later
getModules	Retrieves a list of modules in a workspace	6.2 and later
getViews	Retrieves a list of views that the current user can access	6.2 and later
getWorkspaces	Retrieves a list of workspaces that the current user can access	6.2 and later
lookupAttributeAlias	Translates an attribute ID into an alias	6.3 and later
lookupElementAlias	Translates an element ID into an alias	6.3 and later
lookupViewAlias	Translates a view ID into an alias	6.3 and later
lookupWorkspaceAlias	Translates a workspace ID into an alias	6.3 and later
updateElement	Changes the specified attributes of an element	6.2 and later
updateElements	Updates several elements in one batch	6.2 and later
writeElement	Creates an element, or if the element already exists, updates it	6.2 and later
writeElements	Writes several elements in one batch	6.2 and later

Authentication

authenticate

The authenticate service logs in a user. Before any service is used, this service must be called. A user is authenticated as long as the session is active. After 1-1.5 hours of inactivity, the session times out. You can log in by using the web interface and this API concurrently, as the logins are handled as two different sessions.

To explicitly end a session, call *endSession*. Licenses are held until the session ends or times out.

Returns

The authenticate service returns an XML Schema string as a token. You can use this token for later calls.

Parameters

Name	**Description**	**Optional**	**Type**
username	The user's login name	No	XML Schema string
password	The user's password	No	XML Schema string

Faults

Name	**Returned when**
AuthenticationFailedFault	The user name and password combination is not accepted.

AuthenticationFailedFault contains one of these authenticationFailedReasons:

Reason	**Description**
accountLocked	The account is locked for login.
waitForDelay	If login delay is enabled, a delay occurs between every login attempt. The delay is 5 seconds after the first attempt, 15 seconds after the second attempt, 60 seconds after the third attempt, 5 minutes after the fourth attempt, and 1 hour after more than four attempts.
incorrectPassword	The password was not correct for this user.
licenseError	While retrieving the license, an error occurred.
unspecified	Authentication failed for any other reason.

endSession

The endSession service logs out a user. The session and token are invalidated, and held licenses are released.

Returns

Nothing

Parameters

Name	Description	Optional	Type
token	Token for the session to end	No	XML Schema string

Faults

Name	Returned when
NotAuthenticatedFault	No active session exists for this token

Reading data

getElement

The getElement service retrieves an element and its attributes. You must specify the element alias. The view can be empty, nil/null, or left out. If a view is not specified, the element is retrieved from its module.

Note: Regular users can access elements by using only the views that they can access. To retrieve an element from a module, the current user must be a workspace administrator. For a regular user, a view that the user can access must be specified, and the element must be part of that view.

Returns

The getElement service returns an element. For details, see the “Schema guide” section.

Attribute definitions are not included and can be retrieved independently by calling getAttributeDefinitions.

Parameters

Name	**Description**	**Optional**	**Type**
element	Alias for the element	No	XML Schema string
view	Alias for a view	Yes	XML Schema string
token	Authentication token	No	XML Schema string

Faults

Name	**Returned when**
ElementNotFoundFault	No element matches the alias.
AccessDeniedFault	The current user does not have access to element or the view, or the module, if no view was specified.
DomainNotFoundFault	The specified view or the module cannot be found.
InvalidArgumentFault	No, null, or an empty element alias is supplied
NotAuthenticatedFault	No session is active for the supplied token

Examples

A module with the alias module1 contains two views based on module1, view1 and view2. Module1 contains an element with the alias element1. The element is part of view1 but not view2.

Scenario 1: A regular user who can access view1 calls getElement by setting the element parameter to `element1` and the view parameter to `view1`. Result: element1 is returned with the attributes visible in view1.

Scenario 2: A workspace administrator calls getElement by setting the element parameter to `element1` and the view parameter to `nil/null`. Result: element1 is returned with the attributes in module1.

Scenario 3: A regular user who can access view2 calls getElement by setting the element parameter to `element1` and the view parameter to `view2`. Result: An AccessDeniedFault fault is returned because element1 is not part of view2.

Scenario 4: A workspace administrator calls getElement by setting the element parameter to `element2` and the view parameter to `nil/null`. Result: An ElementNotFoundFault fault is returned because no element has the element2 alias.

Scenario 5: A user of any type calls getElement by setting the view parameter to `view1`. The element parameter is empty. Result: An InvalidArgumentFault fault is returned.

Scenario 6: A workspace administrator calls getElement by setting the element parameter to `element1` and the view parameter to `module1`. Result: A DomainNotFoundFault fault is returned because module1 is not a view.

getElements

The getElements service retrieves elements for a workspace, domain, or both.

If both workspace and domain are specified, elements for that combination are retrieved. If a workspace is not specified, the elements for the domain alias in all workspaces that the current user can access are retrieved.

If the user is a workspace administrator, the domain can be a module or a view. If the user is a regular user, the domain must be a view that the user has access to.

Returns

The getElements service returns an ElementsResult. For more information, see the “Schema guide” section.

Parameters

Name	**Description**	**Optional**	**Type**
workspace	Alias for a workspace	Yes	XML Schema string
domain	Alias for a domain	No	XML Schema string
token	Authentication token	No	XML Schema string

Faults

Name	Returned when
AccessDeniedFault	The user does not have access to the workspace
DomainNotFoundFault	No matching domains are found
InvalidArgumentFault	Domain is left out, empty, or nil/null
NotAuthenticatedFault	No session is active for the supplied token
WorkspaceNotFoundFault	The workspace cannot be found

Examples

Two workspaces exist, one with the alias workspace1 and the other with the alias workspace2. Both workspaces contain modules with the alias module1 and views that are based on module1 with the alias view1.

Scenario 1: A regular user who can access view1 of workspace1 calls getElements by setting the workspace parameter to `workspace1` and the domain parameter to `view1`. Result: The elements in view1 of workspace1 are returned.

Scenario 2: A regular user can access view1 of workspace1 and view1 of workspace2. The user calls getElements by setting the workspace parameter to `nil/null` and the domain parameter to `view1`. Result: The elements in view1 of workspace1 and the elements in view1 of workspace2 are returned.

Scenario 3: A regular user who cannot access workspace1 calls getElements by setting the workspace parameter to `workspace1` and the domain parameter to `view1`. Result: An AccessDeniedFault fault is returned.

Scenario 4: A regular user who can access workspace1 calls getElements by setting the workspace parameter to `workspace1` and the domain parameter to `module1`. Result: A DomainNotFoundFault fault is returned because regular users cannot access modules.

Scenario 5: A workspace administrator can access view1 of workspace1 but cannot access view1 of workspace2. The administrator calls getElements by setting the workspace parameter to `nil/null` and the view parameter to `view1`. Result: An ElementsResult is returned with the elements in view1 of workspace1, and a domainNoAccess error occurs because the user cannot access view1 in workspace2.

Scenario 6: A regular user calls getElements by setting the workspace parameter to `workspace3` and the domain parameter to `view1`. Result: A WorkspaceNotFoundFault fault is returned because no workspace has the alias workspace3.

Scenario 7: A regular user who can access workspace1 calls getElements by setting the workspace parameter to `workspace1` and the domain parameter to `nil/null`. Result: An InvalidArgumentFault fault is returned.

getElementsByAlias

The getElementsByAlias service retrieves elements from a list of element aliases. You might use this service as an alternative to calling getElement for each element that you want to retrieve. The getElementsByAlias service includes information about attributes, such as AttributeDefinitions. You can retrieve up to 100 elements. To retrieve more than 100 elements, you must call getElementsByAlias again.

Views can be empty. If view is not specified, the module is determined from each element alias and the current user must be a workspace administrator. If the user is a regular user, a view that the user has access to must be specified.

Returns

The getElementsByAlias service returns ElementsResult. For more information, see the "Schema guide" section.

Parameters

Name	**Description**	**Optional**	**Type**
elementAliases	A list of aliases for the elements to be retrieved	No	XML Schema string
view	Alias for a view	Yes	XML Schema string
token	Authentication token	No	XML Schema string

Faults

Name	**Returned when**
InvalidArgumentFault	More than 100 aliases are in the elementAliases list
NotAuthenticatedFault	No session is active for the supplied token

Examples

Two workspaces exist, one with the alias workspace1 and the other with the alias workspace2. Both workspaces contain modules with the alias module1 and views that are based on module1 with the alias view1. In workspace1, module1 contains two elements: element1, which is a part of view1, and element2, which is not part of view1. In workspace2, module1 contains two elements: element3, which is a part of view1, and element4, which is not part of view1.

Scenario 1: A workspace administrator can access workspace1 and workspace2.

The administrator calls getElementsByAlias by setting the elementAliases parameter to `[element1, element2, element3, element4]` and the view parameter to `nil/null`. Result: All four elements are returned.

Scenario 2: A regular user can access workspace1 and workspace2 and view1 in both workspaces. The user calls getElementsByAlias by setting the elementAliases parameter to `[element1, element2, element3, element4]` and the view parameter to `view1`. Result: Because element1 and element3 are part of view1, they are returned. In the ElementsResult, errors are displayed for element2 and element4.

Scenario 3: A workspace administrator who can access both workspaces calls getElementsByAlias by setting the elementAliases parameter to `[element1, element2, element3, element4]` and the view parameter to `view1`. Result: Because element1 and element3 are part of view1, they are returned. In the ElementsResult, errors are displayed for the other elements.

Scenario 4: A workspace administrator who can access workspace1 calls getElementsByAlias by setting the elementAliases parameter to `[element1, element2, element3, element4]` and the view parameter to `nil/null`. Result: Because the administrator cannot access workspace 2, only element1 and element2 are returned. In the ElementsResult, errors are displayed for the other elements.

getAttributeDefinitions

The getAttributeDefinitions service retrieves a list of attribute definitions for a specified domain in a workspace.

If both the workspace and domain are empty, the default Add Domain is used. To configure the default Add Domain, click **Application** > **Data Access**.

Returns

The getAttributeDefinitions service returns AttributeDefinitionsResult.

Property name	**Description**	**Type**
attributeDefinitions	A list of AttributeDefinitions	AttributeDefinition
errors	A list of errors, if any errors or warnings occurred while retrieving AttributeDefinitions	Error

Parameters

Name	**Description**	**Optional**
workspace	Alias for a workspace	Might be empty if the domain is empty

domain	Alias for a domain	Might be empty if the workspace is empty
token	Authentication token	No

Faults

Name	**Returned when**
AccessDeniedFault	The user cannot access the workspace or the domain
DomainNotFoundFault	The specified domain cannot be found
InvalidArgumentFault	The workspace or domain is nil/null, empty, or left out.
NotAuthenticatedFault	No session is active for the supplied token
WorkspaceNotFoundFault	The specified workspace cannot be found

Examples

A workspace with the alias workspace1 including a module with alias module1 and a view with alias view1 which is an add view. The Add Domain is set to view1 in workspace1.

Scenario 1: A workspace administrator calls getAttributeDefinitions by setting the workspace parameter to `workspace1` and the domain parameter to `module1`. Result: The AttributeDefinitions for module1 in workspace1 are returned.

Scenario 2: A regular user calls getAttributeDefinitions by setting the workspace parameter to `workspace1` and the domain parameter to `module1`. Result: An AccessDeniedFault is returned, because regular users cannot access modules.

Scenario 3: A regular user who can access view1 in workspace1 calls getAttributeDefinitions by setting the workspace and domain parameters to `nil/null`. Result: The AttributeDefinitions for view1 in workspace1 are returned.

Scenario 4: A user calls getAttributeDefinitions by setting the workspace parameter to `nil/null` and the domain parameter to `view1`. Result: An InvalidArgumentFault fault is returned because no workspace was specified.

Scenario 5: A user calls getAttributeDefinitions by setting the workspace parameter to `workspace2` and the domain parameter to `view1`. Result: A WorkspaceNotFoundFault fault is returned, because workspace2 does not exist.

getFileContent

The getFileContent service retrieves the file data contents of a file attribute. This service is the only way to retrieve file data from a file attribute, because file attributes include only metadata about the files.

If the view is nil/null, empty or left out, the module that the element belongs to is used.

Returns

Binary data in XML Schema base 64 binary format

Parameters

Name	Description	Optional	Type
element	Alias for an element	No	XML Schema string
attribute	Alias for a file attribute	No	XML Schema string
view	Alias for a view	Yes	XML Schema string
fileNumber	The number of files in the file attribute. Use 0 if the file attribute can contain only one file	No	XML Schema int
token	Authentication token	No	XML Schema string

Faults

Name	Returned when
AccessDeniedFault	The user cannot access the workspace, domain, element, or attribute
AttributeNotFoundFault	The supplied attribute cannot be found, the attribute is not a file attribute, or the fileNumber cannot be found in the file attribute
DomainNotFoundFault	The supplied view cannot be found
ElementNotFoundFault	The supplied element cannot be found
InvalidArgumentFault	The element is empty, the attribute is empty, or fileNumber is not a positive integer
NotAuthenticatedFault	No session is active for the supplied token

Examples

A module with the alias module1 contains two views: one with the alias view1 and another with the alias view2. The module contains an element with the alias element1 and a file attribute with the alias fileAttribute. The element is part of view1 and view2, and it has a fileAttribute that contains two files with the numbers 0 and 1. The fileAttribute is visible in view1 but not in view2.

Scenario 1: A workspace administrator calls getFileContent by specifying the following parameters:

- element: element1
- attribute: fileAttribute
- view: nil/null
- fileNumber: `0`

Result: The data for the fileNumber in the fileAttribute of element1 is returned.

Scenario 2: A regular user calls getFileContent by specifying the following parameters:

- element: `element1`
- attribute: fileAttribute
- view: view1
- fileNumber: `0`

Result: The data for fileNumber in the fileAttribute of element1 is returned.

Scenario 3: A regular user calls getFileContent by specifying the following parameters:

- element: `element1`
- attribute: fileAttribute
- view: view2
- fileNumber: `0`

Result: An AccessDeniedFault fault is returned because fileAttribute is not visible in view2.

Scenario 4: A regular user calls getFileContent by specifying the following parameters:

- element: `element1`
- attribute: fileAttribute
- view: view1
- fileNumber: `3`

Result: An AttributeNotFoundFault fault is returned because file number 3 does not exist in fileAttribute.

getModules

The getModules service retrieves a list of modules in a workspace. To use this service, you must be a workspace administrator in the specified workspace.

Returns

The getModules service returns a list of Domains. For more information, see the "Schema guide" section.

Parameters

Name	Description	Optional	Type
workspace	Alias for a workspace	No	XML Schema string
token	Authentication token	No	XML Schema string

Faults

Name	Returned when
AccessDeniedFault	The user is not a workspace administrator
InvalidArgumentFault	Workspace is nil/null, empty, or left out
NotAuthenticatedFault	No session is active for the supplied token
WorkspaceNotFoundFault	The supplied workspace cannot be found

Examples

A workspace has the alias workspace1.

Scenario 1: A workspace administrator calls getModules by setting the workspace parameter to `workspace1`. Result: The modules in workspace1 are returned.

Scenario 2: A regular user calls getModules by setting the workspace parameter to `workspace1`. Result: An AccessDeniedFault fault is returned.

getViews

The getViews service retrieves a list of views that the current user can access.

You can specify the workspace, module, or both. If you specify the workspace, only the views for that workspace are retrieved. If the workspace is empty, all workspaces that you can access are evaluated. If you specify a module, only the views for that module are retrieved. You cannot use this service if both the module and workspace are empty.

Returns

The getViews service returns a list of Domains. For more information, see the "Schema guide" section.

Parameters

Name	Description	Optional	Type
workspace	Alias for a workspace	If the module has a value	XML Schema string
module	Alias for a module	If the workspace has a value	XML Schema string
token	Authentication token	No session is active for the token	XML Schema string

Faults

Name	Returned when
AccessDeniedFault	The user cannot access the workspace
InvalidArgumentFault	Both the workspace and module are nil/null, empty, or left out
NotAuthenticatedFault	No session is active for this token
WorkspaceNotFoundFault	The supplied workspace cannot be found

Examples

Two workspaces, with the aliases workspace1 and workspace2, each have two modules, with the aliases module1 and module2. Each workspace also has two views, view1, which is based on module1, and view2, which is based on module2.

Scenario 1: A regular user who can access all the views calls getViews by setting the workspace parameter to `workspace1` and the module parameter to `module1`. Result: view1 for workspace1 is returned.

Scenario 2: A regular user who can access all the views calls getViews by setting the workspace parameter to `nil/null` and the module parameter to `module1`. Result: view1 for workspace1 and view1 for workspace2 are returned.

Scenario 3: A regular user who can access all the views calls getViews by setting the workspace parameter to `workspace1` and the module parameter to `nil/null`. Result: view1 and view2 for workspace1 are returned.

Scenario 4: A regular user calls getViews by setting both the workspace and view parameters to `nil/null`. Result: An InvalidArgumentFault fault is returned.

getWorkspaces

The getWorkspaces service retrieves a list of workspaces that you can access.

Returns

A list of workspaces that includes the following information:

Property name	Description	Type
alias	The alias for this workspace	XML Schema string
name	The display name for this workspace	XML Schema string
settings	A list of name/value pairs that contain various settings for workspaces. Currently not used.	Setting

Parameters

Name	Description	Optional	Type
token	Authentication token	No	XML Schema string

Faults

Name	Returned when
NotAuthenticatedFault	No session is active for this token

getExternalLink

The getExternalLink service retrieves a URL to an element. To display the element, enter the URL in a web browser.

This service was released in version 6.3 of Telelogic Focal Point.

Returns

The getExternalLink service returns a URL as an XML Schema string.

Parameters

Name	Description	Optional
element	Alias for the element	No
token	Authentication token	No

Faults

Name	Returned when
ElementNotFoundFault	The supplied element cannot be found
InvalidArgumentFault	Element is nil/null, empty, or left out
NotAuthenticatedFault	No session is active for this token

Change history

getChangedElements

The getChangedElements service retrieves aliases for elements that have changed within a specified time range. An element is considered changed if at least one attribute that logs history has changed.

If both the workspace and domain are specified, only the elements that are in the workspace and domain are included in the result. If a workspace is not specified, the elements for the domain alias in the workspaces that the current user can access are included in the result.

Only the aliases for elements are returned. To retrieve the elements, including the current attribute values, use the getElementsByAlias service.

Returns

A ChangedElementsResult with the following information:

Property name	Description	Type
changedDomains	A list of ChangedDomains	ChangedDomain
errors	A list of errors that occurred when the changedElements were retrieved	Error

A ChangedDomain includes the following information:

Property name	Description	Type
module	The alias of the module where the changes occurred	XML Schema string
view	The alias of the view where the changes occurred. If the changes were retrieved from a module, this property can be empty, left out, or nil/null.	XML Schema string
errors	A list of errors that occurred when the changes were retrieved from this domain	Error
workspace	The alias for the workspace where the changes occurred	XML Schema string
elements	A list of aliases for elements that have changed	XML Schema string

Parameters

Name	Description	Optional	Type
workspace	Alias for a workspace	Yes	XML Schema string

domain	Alias for a domain	No	XML Schema string
start	The beginning of the time range for which the history is retrieved	No	XML Schema dateTime
end	The end of the time range for which the history is retrieved	No	XML Schema dateTime
token	Authentication token	No	XML Schema string

Faults

Name	**Returned when**
AccessDeniedFault	The user cannot access the workspace
DomainNotFoundFault	Matching domains are not found
InvalidArgumentFault	The domain, start, or end is left out, empty, or nil/null, or the end is before the start
NotAuthenticatedFault	No session is active for the supplied token
WorkspaceNotFoundFault	The workspace cannot be found

Examples

Two workspaces, with the aliases workspace1 and workspace2, each have two modules with the aliases module1 and module2. In workspace1, module1 contains element1, which was changed on 1 September 2008. Also in workspace1, module2 contains element2, which was changed on 2 September 2008.

In workspace2, module1 contains element3, which was changed on 3 September 2008. Also in workspace2, module2 contains element4, which was changed on 4 September 2008.

Scenario 1: A workspace administrator calls getChangedElements by specifying the following parameters:

- workspace: `nil/null`
- domain: module1
- start: 2 September 2008
- end: 4 September 2008

Result: The alias for element3 is returned.

Scenario 2: A workspace administrator calls getChangedElements by specifying the following parameters:

- workspace: workspace1
- domain: module2

- start: 2 September 2008
- end: 4 September 2008

Result: The alias for element2 is returned.

getHistory

The getHistory service retrieves the change history for a domain.

If both the workspace and domain are specified, the history for that combination is retrieved. If a workspace is not specified, the history for the domain alias in all workspaces that the current user has access to will be retrieved.

The HistoryResult includes the history entries, sorted by date order, starting with the most recent date. The history is limited to 100 entries for each workspace. If more than 100 entries exist, the HistoryResult uses the hasMore and continueDate properties, and the hasMore property is set to true. To retrieve more history items, call getHistory again by using the value of the continueDate property as the end parameter.

Returns

The getHistory service returns a list of HistoryResults for each workspace. For more information, see the "Schema guide" section.

Parameters

Name	Description	Optional	Type
workspace	Alias for a workspace	Yes	XML Schema string
domain	Alias for a domain	No	XML Schema string
start	The beginning of the range for which the history is retrieved	No	XML Schema dateTime
end	The end of the range for which the history is retrieved	No	XML Schema dateTime
token	Authentication token	No	XML Schema string

Faults

Name	Returned when
AccessDeniedFault	The user cannot access the specified workspace
DomainNotFoundFault	The specified domain cannot be found in a workspace
InvalidArgumentFault	The domain, start, or end is left out, empty, or nil/null, or the end is before the start
NotAuthenticatedFault	No session is active for the supplied token
WorkspaceNotFoundFault	The workspace cannot be found

getElementHistory

The getElementsHistory service retrieves the change history for an element.

If view is specified, the history for the element in that view is retrieved. Otherwise, the history for the element in the module is retrieved.

The HistoryResult includes the history entries, which are sorted by date order. The most recent history is listed first. The history is limited to 100 entries. If more than 100 entries exist, the HistoryResult uses the hasMore and continueDate properties, and the hasMore property is set to true. To retrieve more history items, call getElementHistory again by using the value of the continueDate property as the end parameter.

Returns

The getElementHistory service returns a HistoryResult. For more information, see the “Schema guide” section.

Parameters

Name	Description	Optional	Type
element	The alias for an element	No	XML Schema string
view	The alias for a view	Yes	XML Schema string
start	The beginning of the range for which the history is retrieved	No	XML Schema dateTime
end	The end of the time range for which the history is retrieved	No	XML Schema dateTime
token	The authentication token	No	XML Schema string

Faults

Name	Returned when
AccessDeniedFault	The user cannot access the element or the view
DomainNotFoundFault	The view cannot be found
ElementNotFoundFault	The element cannot be found
InvalidArgumentFault	The element, start, or end is left out, empty, or nil/null, or the end is before the start
NotAuthenticatedFault	No session is active for the supplied token

getAttributeHistory

The getAttributeHistory retrieves the change history for an attribute of an element. If a view is specified, the history for the attribute of the element in that view is retrieved. Otherwise, the history for the element in the module is retrieved.

The HistoryResult includes the history entries, which are sorted by date. The most recent history is listed first. The history is limited to 100 entries. If more than 100 entries exist, the HistoryResult uses the hasMore and continueDate properties, and the hasMore property is set to true. To retrieve more history items, call getElementHistory again by using the value of the continueDate property as the end parameter.

Returns

The getAttributeHistory service returns a HistoryResult. For more information, see the "Schema guide" section.

Parameters

Name	**Description**	**Optional**	**Type**
attribute	The alias for an attribute	No	XML Schema string
element	The alias for an element	No	XML Schema string
view	The alias for a view	Yes	XML Schema string
start	The beginning of the range for which the history is retrieved	No	XML Schema dateTime
end	The end of the range for which the history is retrieved	No	XML Schema dateTime
token	The authentication token	No	XML Schema string

Faults

Name	**Returned when**
AccessDeniedFault	The user cannot access the element or the view
AttributeNotFoundFault	The attribute cannot be found
DomainNotFoundFault	The view cannot be found
ElementNotFoundFault	The element cannot be found
InvalidArgumentFault	The attribute, element, start, or end is left out, empty, or nil/ null, or the end is before the start
NotAuthenticatedFault	No session is active for the supplied token

Additions and updates

addElement

The addElement service creates an element. You can specify the workspace and domain aliases. The domain can be a view or a module, in which case the user must be a workspace administrator. When you add elements, you can use add views only. If both the workspace and domain are empty or null, the default Add Domain is used.

You can supply an element alias. If you specify an alias, when the element is created, it uses the alias. If you do not supply an alias, the new element is given a random alias.

A list of attribute changes might be included, and the new element is created with the supplied values. The aliases and attribute types must match the aliases and types in the module where the element is added. Attributes that are not included are assigned their default values.

Returns

The addElement service returns an ElementAddResult. For more information, see the “Schema guide” section.

Parameters

Name	**Description**	**Optional**	**Type**
workspace	The alias for the workspace where the element is added	Is empty if the domain is empty	XML Schema string
domain	The alias for the domain where the element is added. The domain must be a module or an add view	Is empty if the workspace is empty	XML Schema string
element	The alias for the element that is to be created. If you specify this alias, when the element is created, it uses this alias. If you do not specify this alias, when the element is created, it uses a new alias.	Yes	XML Schema string
attributes	A list of attribute changes that are assigned to the new	Yes	AttributeChange

	element. Attributes that use the default value are not included.		
token	The authentication token	No	XML Schema string

Faults

Name	Returned when
AccessDeniedFault	The user cannot access the workspace or domain
DomainNotFoundFault	The domain cannot be found or, if the domain is a view, its module cannot be found
ElementNotCreatedFault	The element could not be created
ElementAlreadyExistsFault	An element with the specified alias already exists
InvalidArgumentFault	The domain or the workspace parameter is nil/null, empty, or left out
NotAuthenticatedFault	No session is active for the supplied token
WorkspaceNotFoundFault	The workspace cannot be found

addElements

The addElements service creates several elements in one batch. You can create up to 100 elements for each call. If you need to add more than 100 elements, call addElements several times.

Returns

The addElements service returns a list of ElementAddResults. Each ElementAddResult contains the alias of the created element and a list of errors that might have occurred during creation.

Parameters

Name	Description	Optional	Type
elements	A list of NewElements	No	NewElement
token	An authentication token	No	XML Schema string

The NewElement type uses the following properties:

Property name	Description	Type
workspace	The alias for the workspace where the	XML Schema string

	element is added. This value is empty if the domain is empty	
domain	The alias for the domain where the element is added. This domain must be a module or an add view. The domain can be empty if the workspace is empty.	XML Schema string
element	The alias for the element to be created. If this alias is specified, when the element is created, it uses this alias. If this alias is not specified, when the element is created, it uses a new alias.	XML Schema string
attributeChanges	A list of attribute changes that are assigned to the new element. Attributes that use the default value are not included.	AttributeChange

Faults

Name	**Returned when**
InvalidArgumentFault	More than 100 elements are supplied
NotAuthenticatedFault	No session is active for the supplied token

updateElement

The updateElement service changes the specified attributes of an existing element. If you are a regular user, you must specify a view. If you are a workspace administrator, specifying a view is optional.

Because the update can include several attribute updates, some attributes might be saved and some might result in errors. Errors that occur during the update are returned.

Important: In the list of attributes, include only the attributes that must change. When you call updateElement, any changes that you made on the server after the value was read are overwritten because the values of the attribute list replace the current values on the server.

For example, user 1 reads the title attribute of element1, and then user 2 changes the title of element 1. User 1 calls updateElement, including the old value for the title of element 1. The title of element 1 is reverted to the previous value because user 1 read the value before it was changed by another user.

Returns

The updateElement service returns a list of *errors* that occurred during the update. For more information, see the “Schema guide” section.

Parameters

Name	Description	Optional
element	An alias for the element that will be updated	No
view	An alias for the view that will be used during the update	Yes
attributes	A list of attribute changes that are to be applied to the element	No
token	The authentication token	No

Faults

Name	Returned when
AccessDeniedFault	The user cannot access the view, module, or workspace, or the element is not part of the view
DomainNotFoundFault	The view cannot be found
ElementNotFoundFault	The element cannot be found
ElementUpdateFault	An error occurred while the content was being updated
InvalidArgumentFault	The element alias is nil/null, empty, or left out
NotAuthenticatedFault	No session is active for the supplied token

updateElements

The updateElements service updates several elements in one batch.

The specified attributes of each ElementUpdate are updated. You must specify the element alias for the ElementUpdate. If you are a regular user, you must specify a view. If you are an administrator, specifying a view is optional. Because the update might include the update of several elements and attributes, some elements and attributes might be saved and some elements and attributes might have errors. Errors that occur during the update are returned.

You can update up to 100 elements for each call. If you need to update more than 100 elements, call updateElements several times.

Important: In the list of attributes, include only the attributes that should change. When you call updateElements, any changes that were made on the server after the value was read are overwritten because the values of the attribute list replace the current values on the server.

For example, user 1 reads the title attribute of element1, and then user 2 changes the title of element 1. User 1 calls updateElements*,* including the old value for the title of element 1. The title of element 1 is reverted to the previous value because user 1 read the value before it was changed by another user.

Returns

The updateElements service returns a list of ElementUpdateResults. If errors occur during the element update, the results include a set of errors for each affected element.

Property name	Description	Type
errors	A list of errors	Error
element	The alias of the element for which the errors occurred	XML Schema string

Parameters

Name	Description	Optional	Type
updates	A list of ElementUpdates	No	ElementUpdate
token	The authentication token	No	XML Schema string

The ElementUpdate type uses these properties:

Property name	Description	Type
attributeChanges	A list of attribute changes to be applied to the element	AttributeChange
element	An alias for the element that will be updated	XML Schema string
view	An alias for the view to use during the update	XML Schema string

Faults

Name	Returned when
InvalidArgumentFault	More than 100 elements must be updated
NotAuthenticatedFault	No session is active for the supplied token

writeElement

The writeElement service creates an element. If an element already exists, this service updates that element.

You must specify the element alias. If you specify an alias that does not match an element, an element is created. If you specify an alias that matches an element, that element is updated.

You can specify the workspace and domain. When you create an element, the workspace and domain are used to determine the module or add view to use. If you do not specify the workspace or the domain, the default Add Domain is used.

When you update an element, the domain is used to determine which view to use for the update. If the domain is empty, the module is used. The workspace is determined by the existing element. Only workspace administrators can access modules.

Returns

The writeElement service returns a list of errors that occurred during the update. For more information, see the "Schema guide" section.

Parameters

Name	Description	Optional	Type
workspace	The alias for the workspace where the element is to be created	Yes	XML Schema string
domain	The alias for the domain that is used to create or update the element	Yes	XML Schema string
element	The alias for the element	No	XML Schema string
attributes	A list of attribute changes that are to be applied	Yes	AttributeChange
token	The authentication token	No	XML Schema string

Faults

Name	Returned when
AccessDeniedFault	The user cannot access the domain or the element
DomainNotFoundFault	The specified domain does not exist
ElementUpdateFault	An error occurred during the update of the element
ElementNotCreatedFault	An element should have been, but was not created
InvalidArgumentFault	The element is nil/null, empty, or left out
NotAuthenticatedFault	No session is active for the supplied token
WorkspaceNotFoundFault	The supplied workspace cannot be found

writeElements

The writeElements service writes several elements in one batch. One element is written for each ElementChange.

You must specify element aliases for each ElementChange. If no element exists with that alias, a new element is created. If an element already exists, it is updated.

For each ElementChange, you can specify the workspace and domain. When you create an element, the workspace and domain are used to determine the module or add view to use. If you do not specify the workspace or domain, the default Add Domain is used.

When you update elements, the domain is used to determine which view to use for the update. If the domain is empty, the module is used. The workspace is determined by the existing element. Only workspace administrators can access modules.

You can write up to 100 elements for each call. To write more than 100 elements, call writeElements several times.

Returns

The writeElements service returns a list of ElementWriteResults. If errors occur during the write process, the results include a set of errors for each element that was affected.

Property name	**Description**	**Type**
errors	A list of errors	Error
element	An alias of the element for which the errors occurred	XML Schema string

Parameters

Name	**Description**	**Optional**	**Type**
changes	A list of ElementChanges	No	ElementChange
token	The authentication token	No	XML Schema string

The ElementChange type uses these properties:

Property name	**Description**	**Type**
workspace	The alias for the workspace where the element is created	XML Schema string
domain	The alias for the domain that is used to create or update the element	XML Schema string
element	The alias for the element	XML Schema string
attributeChanges	A list of attribute changes that are to be applied	AttributeChange

Faults

Name	Returned when
InvalidArgumentFault	The changes contain more than 100 elements
NotAuthenticatedFault	No session is active for the supplied token

lookupWorkspaceAlias

The lookupWorkspaceAlias service translates a workspace ID into an alias. This service is included for compatibility with the previous Web Services API, which used internal identifiers instead of aliases. You can also use this service to find an alias if you know the internal identifier, but not the alias, of a workspace.

This service was released in version 6.3 of Telelogic Focal Point.

Returns

The lookupWorkspaceAlias service returns a workspace alias as an XML Schema string.

Parameters

Name	Description	Optional	Type
workspaceId	An internal identifier for a workspace	No	XML Schema int

Faults

Name	Returned when
WorkspaceNotFoundFault	A workspace with the supplied identifier cannot be found

lookupViewAlias

The lookupViewAlias service translates a view ID into an alias. This service is included for compatibility with the previous Web Services API, which used internal identifiers instead of aliases. You can also use this service to find an alias if you know the internal identifier, but not the alias, of a view.

This service was released in version 6.3 of Telelogic Focal Point.

Returns

The lookupViewAlias returns the view alias as an XML Schema string.

Parameters

Name	Description	Optional	Type

viewId	An internal identifier for a view	No	XML Schema int
workspaceId	An internal identifier for a workspace	No	XML Schema int

Faults

Name	**Returned when**
DomainNotFoundFault	A view with the supplied identifier cannot be found

lookupElementAlias

The lookupElementAlias service translates an element ID into an alias. This service is included for compatibility with the previous Web Services API, which used internal identifiers instead of aliases. You can also use this service to find an alias if you know the internal identifier, but not the alias, of an element.

This service was released in version 6.3 of Telelogic Focal Point.

Returns

The lookupElementAlias service returns an element alias as an XML Schema string.

Parameters

Name	**Description**	**Optional**	**Type**
elementId	An internal identifier for an element	No	XML Schema int
workspaceId	An internal identifier for a workspace	No	XML Schema int

Faults

Name	**Returned when**
ElementNotFoundFault	An element with the supplied identifier cannot be found

lookupAttributeAlias

The lookupAttributeAlias service translates an attribute ID into an alias. This service is included for compatibility with the previous Web Services API, which used internal identifiers instead of aliases. You can also use this service to find an alias if you know the internal identifier, but not the alias, of an attribute.

This service was released in version 6.3 of Telelogic Focal Point.

Returns

The lookupAttributeAlias service returns the attribute alias as an XML Schema string.

Parameters

Name	**Description**	**Optional**	**Type**
attributeId	An internal identifier for an attribute	No	XML Schema int
workspaceId	An internal identifier for an attribute	No	

Faults

Name	**Returned when**
AttributeNotFoundFault	An attribute with the supplied identifier cannot be found

Schema guide

This section describes the data types in the XML Schema that is included in the WSDL. For more information, also read the XML Schema.

ElementsResult

Property name	Description	Type
Domains	A list of ElementSets, one for each domain	ElementSet
Errors	Errors or warnings that occur when accessing a domain	Error

ElementSet

Property name	Description	Type
attributeDefinitions	A list of the names and definitions of attributes	AttributeDefinition
Elements	A list of elements	Element
Module	The alias of the module from which this element set was retrieved	XML Schema string
View	The alias of the view from which this element set was retrieved. If the element set was retrieved from a module, the alias can be nil/ null, empty, or left out.	XML Schema string
Errors	A list of errors and warnings that occurred when reading the elements or attributeDefinitions	Error
Workspace	The alias of the workspace from which this element set was retrieved	XML Schema string

Element

An element has the following properties:

Property name	Description	Type
alias	The alias that identifies this element	XML Schema string
Attributes	A list of attributes	Attribute
Module	The alias of the module that the element belongs to	XML Schema string
View	The alias of view that the element was retrieved from. If the element was retrieved from a module, this alias might be empty.	XML Schema string

Workspace	The alias of the workspace that the element belongs to	XML Schema string
Errors	A list of errors or warnings that occurred when the attribute values were read	Error

Attribute

An attribute has the following properties. The attribute value is represented by exactly one property, but the name and type of that property are different for each type of attribute. For example, a text attribute has a text property, and an integer attribute has an integer property. The Other property is currently not used.

Property name	**Description**	**Type**
Alias	The alias of the attribute in its module	XML Schema string
Type	The type of the attribute	AttributeType
Writeable	If the attribute value can be changed, this property is true. If the attribute value cannot be changed, this property is false.	XML Schema Boolean
One of these properties:	A representation of the attribute value. This property depends on the attribute type.	One of the following types:
checkBox		CheckBox
choice		Choice
date		Date
float		Float
file		File
integer		Integer
link		Link
textList		TextList
linkList		LinkList
matrix		Matrix
multichoice		Multichoice
text		Text
uniqueId		UniqueId
url		Url
version		Version
other		XML Schema anyType

CheckBox

Property name	Description	Type
Value	If the check box is selected, this property is true. Otherwise, this property is false.	XML Schema Boolean

Choice

Property name	Description	Type
value	The selected choice item for this attribute. The value is the name of the choice item.	XML Schema string
availableValues	A list of choice items that this choice can be changed to	XML Schema string

Date

Property name	Description	Type
dateTime	The value of a date expressed as a date and time in UTC. A date attribute is stored at the time 00:00 (midnight) for the time zone of the server. If the server is in the GMT+2 time zone, the date 2008-08-13 is displayed as 2008-08-12T22:00. The Last Changed Date and Created Date attributes include the exact time when the element was created or changed.	XML Schema dateTime
Expression	A date attribute can contain an expression. If the attribute does not have an expression, this property is nil/null.	Expression

Float

Property name	Description	Type
Value	The float value	XML Schema double
Expression	A float attribute can contain an expression. If the attribute does not contain an expression, this	Expression

property is nil/null.

File

Property name	Description	Type
fileInfos	A list of FileInfo attributes, one for each file in the file attribute	FileInfo

Note: The file content is not included in the file attribute; the attribute contains only metadata about the files. You can retrieve the file content by calling getFileContent.

FileInfo

Property name	Description	Type
filename	The name of the file	XML Schema string
contentType	The MIME type of the file	XML Schema string
Size	The size of the file, in bytes	XML Schema long
fileNumber	The identifier for the file within a particular file attribute	XML Schema int

Integer

Property name	Description	Type
Value	The integer value	XML Schema long
Expression	An integer attribute can contain an expression. If the attribute does not contain an expression, this property is nil/null.	Expression

Link

Property name	Description	Type
displayText	The display text for a link	XML Schema string
Element	The alias of the linked element	XML Schema string
Module	The alias of the module of the linked element	XML Schema string
Workspace	The alias of the workspace of the linked element	XML Schema string

Note: If a link is not set, or has no value, all of these properties are left out or nil/ null.

TextList

A List (Text) attribute represents the value of a list attribute of the type Text.

Property name	**Description**	**Type**
textListEntries	A list of TextListEntries, one for each entry in the list	TextListEntry

TextListEntry

Property name	**Description**	**Type**
Id	An identifier for the entry within its list	XML Schema string
Author	A link to the workspace member who created the entry	Link
createdDate	The date and time when the entry was created	XML Schema dateTime
lastChangedBy	A link to the workspace member who last edited this entry	Link
lastChangedDate	The date and time when this entry was last edited	XML Schema dateTime
formattedTextValue	The text value of the entry. This value can be formatted by using the following HTML rich text tags: <B> for bold <I> for italics <U> for underlined <S> for strike-through <OL> for numbered list <UL> for unordered list	XML Schema string
textValue	The unformatted text value of the entry	XML Schema string

LinkList

A List (Link) attribute represents the value of a list attribute of the type Link

Property name	Description	Type
Links	A list of links	Link

Matrix

Property name	Description	Type
Rows	A list of the rows that the matrix contains	MatrixRow

MatrixRow

Property name	Description	Type
rowed	The ID for a row in a matrix. The ID of the column title row is ColumnTitle. The ID of the next row is 1.	XML Schema string
Cells	A list of the matrix cells in this row	MatrixCell

MatrixCell

Property name	Description	Type
rowed	The ID for the row in which this cell is placed. The ID of the column title row is ColumnTitle, and the ID of the next row is 1. The IDs of the rows that follow row 1 are 2, 3, and so on.	XML Schema string
columned	The ID for the column in which this cell is placed. The ID of the row title column is RowTitle, and the ID of the next column is A. The IDs of the columns that follow column A are B, C, and so on.	XML Schema string
Value	The value of the cell as an unformatted text	XML Schema string

formattedText	The text value of the cell, which you can format by using the following HTML rich text tags: <B> for bold <I> for italics <U> for underlined <S> for strike-through <OL> for numbered list <UL> for unordered list	XML Schema string
Expression	A matrix cell can contain an expression. If the cell does not contain an expression, this property is nil/null.	Expression
Locked	If this property is true, the cell cannot be edited.	XML Schema Boolean

Multichoice

Property name	Description	Type
Selected	A list of the selected items of the multi choice attribute. The value of each item is the name of the item.	XML Schema string

Text

Property name	Description	Type
formattedTextValue	The value of the text attribute, which you can format by using the following HTML rich text tags: <B> for bold <I> for italics <U> for underlined <S> for strike-through <OL> for numbered list <UL> for unordered list	XML Schema string
textValue	The unformatted value of the text attribute	XML Schema string
Expression	A text attribute can contain an expression. If the attribute does not contain an expression, this property is nil/null.	Expression

UniqueId

Property name	Description	Type
Value	The value of a unique ID attribute	XML Schema string

Url

Property name	Description	Type
Value	The value of a URL attribute	XML Schema string

Version

Property name	Description	Type
Major	The major identifier in a version. For example, in version x.y.z, the major identifier is x.	XML Schema string
Minor	The minor identifier of a version. For example, in version x.y.z, the minor identifier is y.	XML Schema string
Patch	The patch identifier of a version. For example, in version x.y.z, the patch identifier is z.	XML Schema string

Other

This type can contain any XML, and might be used for attribute values that cannot be represented in another way. This type of attribute is currently not used.

Expression

Text, integer, float, and date attributes can contain an expression. If an attribute contains an expression, the value of the attribute contains the result of the expression, unless the expression is pending an update or is not valid.

Property name	Description	Type
Expression	The actual expression, which is	XML Schema string

	represented without the initial equals symbol (=) that is used in the web interface.	
pendingUpdate	If this property is set to true, the expression is waiting to be evaluated.	XML Schema Boolean
Valid	If this property is set to false, an error exists in the expression. For example, an error might occur because of incorrect expression syntax.	XML Schema Boolean

AttributeDefinition

Property name	**Description**	**Type**
alias	The alias of the attribute	XML Schema string
availableValues	A list of the values that are valid for this attribute. This property is used for choice and multi choice attributes only.	XML Schema string
description	The description of the attribute	XML Schema string
mandatory	Whether this attribute must contain a value	XML Schema Boolean
module	The alias of the module that this attribute belongs to	XML Schema string
name	The name of the attribute	XML Schema string
settings	A list of name/value pairs that contain various settings for attributes. Currently not used.	Setting
sortOrder	The order for this attribute relative to other attributes. This property is used when displaying element to ensure that attributes are sorted correctly.	XML Schema int
systemName	Provides a special meaning to an attribute. For example, the systemName “Title” means that the attribute is the title attribute. For a list of systemNames, see the systemName table.	XML Schema string
type	The type of attribute	AttributeType
view	The alias of the view that this attribute was retrieved from. If the attribute was retrieved from	XML Schema string

	a module, this property can be empty, nil/null, or left out.	
workspace	The alias of the workspace that this attribute belongs to	XML Schema string
writeable	If the value of the attribute can be edited, this property is true. Otherwise, this property is false. A specific attribute for an element might not be editable for other reasons, for example, if the element is locked. For more information, see the writeable property in the “Attribute” section.	XML Schema Boolean
mirror	If this attribute is a mirror attribute, which reflects the value of another attribute, this property is true. If the property is true, the type of this attribute is the same as the mirrored attribute.	XML Schema Boolean

systemName

The systemName property provides a special meaning to an attribute.

systemName property	Description
Title	The Title attribute
Descr	The Description attribute
createdby	The Creator attribute
createddate	The Created Date attribute
lastchangedby	The Last Changed By attribute
lastchangedate	The Last Changed Date attribute
isGroup	This property is a check box attribute that indicates whether an element is a folder. You can turn an element into a folder or turn a folder into an element by changing the value of this attribute.
parent	This property indicates the folder in which an element can be found
Prefix	The Prefix attribute, which you can configure
lock	A check box attribute that you can add to a module. If this property is set to true, the

	element or some attributes of the element are locked for editing.
change_assigned	A hidden check box attribute, which is used only when integration with IBM Rational® Change™ is enabled

Error

Property name	Description	Type
code	The error code, which indicates the type of error	ErrorCode
message	A description of the error	XML Schema string
alias	An alias that is associated with the error. The type of the alias depends on the error code. For example, if the error code is attributeReadError, the alias is an attribute alias that indicates the attribute for which the error occurred. This property can be nil/null, empty, or left out.	XML Schema string

ErrorCode

The ErrorCode property defines a set of errors that might occur. For more information, see the “Error” section. This property can use one of the following codes:

Error code	Description
attributeNotEditable	The current user tried to change the value of an attribute that cannot be edited
attributeNotPartOfDomain	The current user tried to read or change an attribute that was not part of a specified domain
attributeNotFound	An alias was specified for an attribute that does not exist
attributeWriteError	While the value of an attribute was being changed, an error occurred
attributeReadError	While the value of an attribute was being read, an error occurred
emptyDomain	A domain with no elements was specified
domainNotFound	An alias was specified for a domain that does not exist
domainNoAccess	The current user tried to access a domain that

	they cannot access
domainError	While a domain was being read, an error occurred
elementWriteError	While an element was being created or updated, an error occurred
elementNotFound	An alias was specified for an element that does not exist
elementNotCreated	While an element was being created, an error occurred and the element was not created
elementNoAccess	The current user tried to access an element that they cannot access
elementAlreadyExists	The current user tried to create an element with a specified alias but an element with that alias already exists
invalidArgument	An input parameter has a value that is not allowed or does not contain a value
accessDenied	The current user tried to access something that they cannot access
workspaceNotFound	An alias was specified for a workspace that does not exist
workspaceNoAccess	The current user tried to access a workspace that they cannot access
expressionError	An error that is related to an expression occurred. For example, the user added an expression that contained incorrect syntax.
internalExpressionError	An error occurred for an internal, or systems, expression. If this error occurs, the Rational Focal Point server is not functioning correctly.
generalError	An unspecified error occurred. The message property might include details.
internalError	An unspecified internal error occurred. If this error occurs, the Rational Focal Point server is not functioning correctly. This error might occur if the database is inconsistent.

AttributeType

The AttributeType property can be one of the following values:

- CheckBox
- Choice
- Date
- File
- Float
- Heading

- Integer
- Link
- TextList
- LinkList
- Matrix
- Multi choice
- Text
- Url
- UniqueId
- Version
- Other

Heading attributes never have a value. Those attributes are included only as separators between attributes.

The Other type is currently not used.

The Mirror attribute does not have its own type. Instead, a mirror attribute uses the type of the attribute that it mirrors, and its AttributeDefinition property is true.

Domain

A domain represents the metadata for a module or a view.

Property name	**Description**	**Type**
workspace	The alias of the workspace that the domain belongs to	XML Schema string
alias	The alias of the domain	XML Schema string
name	The display name of the domain	XML Schema string
type	“View” or “Module”	XML Schema string
module	If the domain is a view, this property is the module that the domain is based on. If the domain is a module, this property has the same value as the alias property.	XML Schema string
addable	If elements can be added to the domain, this property is true.	XML Schema Boolean
displayable	If the domain is used to read elements, this property is true.	XML Schema Boolean
settings	A list of name/value pairs that contain various settings for domains. Currently not used.	Setting

HistoryResult

Property name	Description	Type
historyEntries	A list of HistoryEntries for this result	HistoryEntry
errors	A list of errors that occurred when retrieving history	Error
hasMore	If the limit for the number of HistoryEntries has been reached, but there are more HistoryEntries to retrieve, this property is true.	XML Schema Boolean
continueDate	If the hasMore property is true, a continueDate is added. This date can be used to retrieve the remaining HistoryEntries.	XML Schema dateTime

HistoryEntry

A HistoryEntry represents one change that was made at a given time for an attribute in an element.

Property name	Description	Type
username	The name of the user who made the change	XML Schema string
userAlias	The alias of the user who made the change	XML Schema string
date	The date and time when the change occurred	XML Schema dateTime
workspace	The alias of the workspace that this HistoryEntry comes from	XML Schema string
module	The alias of the module that this HistoryEntry comes from	XML Schema string
view	The alias of the view that this HistoryEntry comes from. If the HistoryEntry was not retrieved from a view, this property can be nil/null, empty, or left out.	XML Schema string
element	The alias of the element	XML Schema string
attribute	The value that the attribute had at the time of the change	Attribute
errors	A list of errors that occurred when retrieving the HistoryEntry	Error

ElementAddResult

Property name	Description	Type
element	The alias of the created element	XML Schema string
errors	A list of errors that occurred when adding the element	Error

AttributeChange

Property name	Description	Type
alias	The alias of the attribute	XML Schema string
Exactly one of the following values: checkBox choice date float file integer link textList linkList matrix multichoice text uniqueId url version other	A representation of the attribute value. The property depends on the attribute type.	One of the following types: CheckBox ChoiceChange DateChange FloatChange FileChange IntegerChange LinkChange TextListChange LinkListChange MatrixChange Multichoice TextChange UniqueId Url VersionChange XML Schema anyType

ChoiceChange

Property name	Description	Type
value	The name of the choice item that the choice attribute should be changed to.	XML Schema string

DateChange

The DateChange must have exactly one of the following properties:

Property name	Description	Type
value	A new date value. The date must be within the range of the date for the time zone of the server. If the server is in the GMT+2 time zone, the range 2008-08-12T22:00 to 2008-08-13T22:00 represents the date 2008-08-13.	XML Schema dateTime
expression	A new expression for the date attribute. For instructions to write expressions, see the Rational Focal Point help at http://www.ibm.com/support/docview.wss?uid=swg27016559. You can omit the initial equals symbol (=).	XML Schema string

FloatChange

The FloatChange must have exactly one of the following properties:

Property name	Description	Type
value	A new float value	XML Schema double
expression	A new expression for the float attribute. For instructions to write expressions, see the Rational Focal Point help at http://www.ibm.com/support/docview.wss?uid=swg27016559. You can omit the initial equal sign (=).	XML Schema string

FileChange

Property name	Description	Type
removeFiles	A list of files to remove from the attribute	RemoveFile
addFiles	A list of files, including binary content, to add to the attribute	AddFile

RemoveFile

Property name	Description	Type
fileNumber	The number of the file that is to be removed	XML Schema int

AddFile

Property name	Description	Type

fileName	The name of the file	XML Schema string
contentType	The MIME type of the file	XML Schema string
fileData	The content of the file	XML Schema base64Binary

IntegerChange

An IntegerChange must have exactly one of the following properties:

Property name	**Description**	**Type**
value	A new integer value	XML Schema long
expression	A new expression for the integer attribute. For instructions to write expressions, see the Rational Focal Point help at http://www.ibm.com/support/docview.wss?uid=swg27016559. You can omit the initial equal sign (=).	XML Schema string

LinkChange

Property name	**Description**	**Type**
element	The alias of the element that the link attribute will link to	XML Schema string

TextListChange

Property name	**Description**	**Type**
deleteEntries	A list of DeleteTextListEntries	DeleteTextListEntry
changeEntries	A list of TextListEntryChanges	TextListEntryChange
addEntries	A list of AddTextListEntries	AddTextListEntry

DeleteTextListEntry

Property name	**Description**	**Type**
id	The identifier of the entry in a text list attribute that is to be deleted	XML Schema string

ChangeEntries

Property name	**Description**	**Type**

id		
textValue or formattedTextValue	A plain text value or a formattedTextValue. You can format the text by using a quoted limited subset of HTML.	XML Schema string

AddTextListEntry

Property name	**Description**	**Type**
textValue or formattedTextValue	A plain text value or a formattedTextValue. You can format the text by using a quoted limited subset of HTML.	XML Schema string

LinkListChange

Property name	**Description**	**Type**
id	The identifier of the entry in a text list attribute that is to be changed	XML Schema string
textValue or formattedTextValue	A plain text value or a formattedTextValue. You can format the text by using a quoted limited subset of HTML.	XML Schema string

MatrixChange

A MatrixChange must have exactly one of the following properties:

Property name	**Description**	**Type**
resetMatrix	This property deletes an entire matrix and replace it with new values	ResetMatrix
changeCells	The changes to apply to the specified cells	ChangeCells
addRows	The rows to add to the matrix	AddRows
addColumns	The columns to add to the	AddColumns

	matrix	
deleteRow	The ID of a row to delete. The ID of the column title row is ColumnTitle, and the ID of the next row is 1. The IDs of the rows that follow row 1 are 2, 3, and so on.	XML Schema string
deleteColumn	The identifier of a column to delete. The ID of the row title column is RowTitle, and the ID of the next column is A. The IDs of the columns after column A are B, C, and so on.	XML Schema string

ResetMatrix

Property name	**Description**	**Type**
newRows	A list of NewCells to add to the matrix. Each row must have one set of NewCells.	NewCells

ChangeCells

Property name	**Description**	**Type**
cellChanges	A list of MatrixCellChanges to apply to a matrix attribute. New cells are not created, and you can change existing cells only.	MatrixCellChange

AddRows

Property name	**Description**	**Type**
rows	A list if NewCells to add to the matrix. Each new row must have one set of NewCells.	NewCells

AddColumns

Property name	Description	Type
columns	A list of NewCells to add to the matrix. Each new column must have one set of NewCells.	NewCells

NewCells

Property name	Description	Type
newCells	A list of MatrixCellValueChanges. Each property represents one cell.	MatrixCellValueChange
x	A placeholder that is ignored	N/A

MatrixCellChange

Property name	Description	Type
rowId	The ID of the row in which the cell is found. The ID of the column title row is ColumnTitle, and the ID of the next row is 1. The IDs of the rows that follow row 1 are 2, 3, and so on.	XML Schema string
columnId	The ID of the column in which the cell is found. The ID of the row title column is RowTitle, and the ID of the next column is A. The IDs of the columns that follow column A are B, C, and so on.	XML Schema string
valueChange	The new value for the cell	MatrixCellValueChange

MatrixCellValueChange

A MatrixCellValueChange must have exactly one of the following properties:

Property name	Description	Type
value	A new value for a matrix cell. The value can be text or a number.	XML Schema string

formattedText	A formatted text value for a matrix cell. You can format text by using a quoted limited subset of HTML.	XML Schema string
expression	A new expression for the matrix cell. For instructions to write expressions, see the Rational Focal Point help at http://www.ibm.com/support/docview.wss?uid=swg27016559. You can omit the initial equal sign (=).	XML Schema string

TextChange

A TextChange must have exactly one of the following properties:

Property name	**Description**	**Type**
textValue	A new value as unformatted text	XML Schema string
formattedText	A new formatted text value. You can format text by using a quoted limited subset of HTML.	XML Schema string
expression	A new expression for the text attribute. For instructions to write expressions, see the Rational Focal Point help at http://www.ibm.com/support/docview.wss?uid=swg27016559. You can omit the initial equal sign (=).	XML Schema string

VersionChange

Property name	**Description**	**Type**
nextMajorVersion	If this property is true, a version attribute must be increased to the next major version. If the property is false, nothing happens.	XML Schema Boolean

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Intellectual Property Dept. for Rational Software
IBM Corporation
5 Technology Park Drive
Westford, MA 01886
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information in softcopy, the photographs and color illustrations may not appear.

Trademark acknowledgments

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A