



SERENA®

BUSINESS MANAGER

Web Services Developer's Guide

Serena Proprietary and Confidential Information

Copyright © 2001–2011 Serena Software, Inc. All rights reserved.

This document, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by such license, no part of this publication may be reproduced, photocopied, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Serena. Any reproduction of such software product user documentation, regardless of whether the documentation is reproduced in whole or in part, must be accompanied by this copyright statement in its entirety, without modification. This document contains proprietary and confidential information, and no reproduction or dissemination of any information contained herein is allowed without the express permission of Serena Software.

The content of this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Serena. Serena assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

Trademarks

Serena, TeamTrack, StarTool, PVCS, Collage, Comparex, Dimensions, Serena Dimensions, Mashup Composer, Mashup Exchange, Prototype Composer, Mariner, and ChangeMan are registered trademarks of Serena Software, Inc. The Serena logo, Version Manager, Meritage, and Mover are trademarks of Serena Software, Inc. All other products or company names are used for identification purposes only, and may be trademarks of their respective owners.

U.S. Government Rights

Any Software product acquired by Licensee under this Agreement for or on behalf of the U.S. Government, its agencies and instrumentalities is "commercial software" as defined by the FAR. Use, duplication, and disclosure by the U.S. Government is subject to the restrictions set forth in the license under which the Software was acquired. The manufacturer is Serena Software, Inc., 1900 Seaport Boulevard, 2nd Floor, Redwood City, CA 94063-5587.

Part number: Product version: 2009 R4.02

Publication date: 2011-06-21

Table of Contents

Chapter 1: Preface	11
Guide to SBM Documentation	11
Chapter 2: Getting Started	13
About the SBM Application Engine Web Services API	13
About Web Services	13
SOAP Requests	13
Before You Begin	14
System Requirements	14
Licensing	14
Security	14
Setting Up the Development Environment	15
Generating Web Services Stubs	15
Setting Up Microsoft Visual Studio	15
Upgrading Visual Studio	15
Setting Up Web Service Stub Files	15
Installing the SBM Web Services API	16
Web Service and Script Execution Order	17
Error Message Logs	18
SOAP Faults	18
Chapter 3: Web Services API Reference	21
SBM Application Web Services	21
Calls Available	21
CreateAuxItem	24
CreateAuxItems	27
CreateFileAttachment	30
CreateNoteAttachment	32
CreatePrimaryItem	33
CreatePrimaryItems	37
DeleteAttachment	43

DeleteItems	44
DeleteItemsByQuery	45
GetApplications	47
GetAvailableSubmitTransitions	48
GetAvailableTransitions	49
GetFileAttachment	50
GetItem	52
GetItems	53
GetItemsByQuery	56
GetNoteLoggerInfo	59
GetReports	60
GetSolutions	62
GetStateChangeHistory	63
GetSubmitProjects	64
GetTables	65
GetUsers	67
GetVersion	69
IsUserValid	70
LinkSubtask	71
Logout	72
RunReport	73
UpdateFileAttachment	75
TransitionItem	77
TransitionItems	80
Common Types	83
ApplicationIdentifier	86
Attachment-Access-Type	87
ContactIdentifier	88
FieldIdentifier	89

FieldValue	89
FileAttachment	90
FileAttachmentContents	91
FileBufferBase64	92
FileContents	92
GroupIdentifier	93
Identifier	94
ItemIdentifier	95
ItemLink	96
ItemLink-Type	97
Note	98
ProjectIdentifier	100
QueryRange	100
ReportAccessLevel	101
ReportCategory	102
ReportIdentifier	104
ReportType	104
SolutionIdentifier	106
Solution-Type	106
StateIdentifier	107
Subtasks	108
TableIdentifier	109
Table-Type	110
TransitionIdentifier	111
Transition-Type	111
TTItem	114
URLAttachment	137
UserIdentifier	138
WorkflowIdentifier	139

Arguments	139
Auth	141
ExtendedData	142
ExtraValue	143
GetTransitionOptions	144
MultipleOption	144
MultipleOptions	145
MultipleResponseItemOptions	146
NameValue	148
NoteAttachmentContents	150
Options	151
ReportsFilter	151
ResponseItemOptions	154
SectionsOption	156
Set-Value-By	158
Set-Value-Method	159
UserResponseOptions	162
UserSingleResponseOptions	164
Responses	166
AccessType	169
ApplicationData	170
DatePreference	171
Field	171
Field-Type	173
FieldWithValue	175
GetReportsResult	175
GetStateChangeHistoryResult	177
NoteLoggerInfo	179
OrderBy	179
ProjectData	180

ReportDefinition.....	181
ReportInfo	182
ReportResult	184
RunReportResult	185
SolutionData	191
StateChangeHistory	192
Status	193
StatusEnum	194
TableData	195
TimePreference	196
Transition	197
TTItemHolder	198
TTItemList	201
UserHolder	202
UserInfo	206
UserSolutionData	211
SBM Administrative Web Services	212
Calls Available	212
CreateGroups	214
CreateProject	217
CreateUsers	220
DeleteProcessApp	224
GetGroups	225
GetUsers	227
GetUserPrivileges	229
GetVersion	232
HasUserPrivilege	232
IsValid	234
Logout	235
UpdateGroups	236

UpdateUsers	239
Common Types.....	243
AccessType	246
ApplicationIdentifier	246
Attachment-Access-Type.....	247
ContactIdentifier	248
DatePreference	249
FieldIdentifier	249
FileBufferBase64	250
GroupInfo	251
GroupIdentifier	252
Identifier	253
ItemIdentifier	254
ProjectIdentifier.....	255
ReportIdentifier	256
SolutionIdentifier	256
StateIdentifier	257
TableIdentifier	258
TimePreference	259
TransitionIdentifier	259
UserIdentifier	260
UserInfo	261
UserSolutionData	266
WorkflowIdentifier.....	268
Arguments	269
Auth	270
ExtendedData.....	271
ExtraValue	272
GroupListAction	272
Options	274

MultipleOption	274
MultipleOptions	275
MultipleResponseItemOptions	276
SectionsOption	278
UserResponseOptions	280
Responses	282
GroupHolder	283
NewUser	285
Privilege	287
ProjectGeneralData	289
Status	290
StatusEnum	291
UserHolder	292
Chapter 4: Tips for Writing Applications	297
Authentication Methods.....	297
Argument	297
HTTP Basic.....	297
WS-Security	297
C++ Constants	298
Supported Character Encoding	298
Supported Date/Time Formats	298
Application, Table, and Project IDs	299
Chapter 5: Sample Programs	301
About the Sample Programs	301
Solutions and Tables	301
CreateItems	302
DeleteItems	302
GetItems	302
Reports	302
UpdateItems	302

Users and Groups.....	303
Using the Sample Programs.....	303
Requirements	303
Choosing Authentication.....	304
Editing the Web Services URL	304
Rebuilding the Executables	304

Chapter 1: Preface

This document describes how to use the SBM Application Engine Web Services API, a product of Serena Software, Inc.. SBM is a Web-architected change request and process management solution that empowers application development teams to improve communication and development processes across the enterprise.

Through the Web services API, you can develop integrations with SBM that create, read, update, and delete primary and auxiliary items within SBM. The Web services API also returns details for states, transitions, projects, tables, fields, users, applications and more.

Audience and Scope

This manual is intended for experienced developers interested in integrating their products with SBM.



Important: Support for development efforts writing Web services is provided by Professional Services. Questions regarding use of Web services operations in orchestration processes as documented are handled by customer support.

Guide to SBM Documentation

The SBM documentation set includes the following manuals in PDF format. Most documents are installed with SBM; to obtain a document that is not installed with the product or to download the complete documentation set, visit <http://www.serena.com/support>.

Manual	Description
<i>Serena[®] Business Manager Installation and Configuration Guide</i>	Provides information on installing SBM and creating a database. Database and Web server configuration information is also provided.
<i>SBM Composer Guide</i>	Provides details on using SBM Composer to create the tables, fields, workflows, orchestrations, and other design elements comprised in process apps. Information about saving, versioning, importing, and exporting process apps is also provided. This document is intended for individuals who want to design and maintain process apps.
<i>SBM Application Administrator Guide</i>	Provides information on using Application Administrator to deploy process apps to runtime environments and to promote configured applications from one environment to another.

Manual	Description
<i>SBM System Administrator Guide</i>	Provides information on configuring and administering the SBM Application Engine. Instructions for managing projects, user accounts, system settings, and authentication are included.
<i>Serena[®] Business Manager User's Guide</i>	Provides information about the SBM User Workspace and is intended for end users. Instructions on using the SBM User Workspace, including the robust reporting feature in SBM, are included. To ease the process of providing a copy for every user in your system, the <i>Serena[®] Business Manager User's Guide</i> is provided in PDF and can be accessed from the Product Information tab of the About page in the SBM User Workspace.
<i>Serena[®] Business Manager Licensing Guide</i>	Explains how to manage licenses for Serena [®] Business Manager. License types are discussed, along with instructions for installing and using the Serena License Manager. This guide is intended for administrators who will install and implement Serena [®] Business Manager.
<i>Serena[®] Business Manager Web Services Developer's Guide</i>	Provides an overview of all SBM Web services, including descriptions for all calls, arguments, and responses. Installation instructions and information about the sample Web service programs are also provided.
<i>SBM AppScript Reference</i>	Provides information on customizing SBM using SBM AppScript, a programming language built around VBScript 4.0. This guide is intended for VBScript programmers who want to use SBM AppScript to implement custom features in an SBM system.
<i>Moving to Serena[®] Business Manager</i>	Provides migration information for existing TeamTrack customers who are moving to SBM. It explains how to upgrade your existing system, and it explains the expanded SBM paradigm in relation to the TeamTrack paradigm.

Chapter 2: Getting Started

- [About the SBM Application Engine Web Services API \[page 13\]](#)
- [About Web Services \[page 13\]](#)

About the SBM Application Engine Web Services API

With the SBM Application Engine Web Services API you can access key SBM features from your own applications. This enables you to build your own front-end clients for your users' most common tasks, including:

- Submitting items into projects or auxiliary tables
- Transitioning and updating items
- Deleting items
- Viewing item details
- Listing applications and projects
- Finding and running Listing reports

These items can either be project-based primary items or auxiliary items.

About Web Services

Web services are applications that are accessible using standard Internet protocols and formats such as Extensible Markup Language (XML), Hypertext Transfer Protocol (HTTP), or Simple Object Access Protocol (SOAP). You can implement applications that interact with Web services on any platform in any programming language, as long as the language can create and respond to messages that are sent using SOAP over HTTP.

The SBM Application Engine Web Services API is supported on Windows using the IIS Web server for the GSOAP framework.



Note: To use the SBM Web services from Perl, you need Soap::Lite version 0.69 or later.

SOAP Requests

Serena Business Manager supports the SOAP protocol for calling Web service operations over HTTP or HTTPS. Web service SOAP messages are essentially specially formatted XML data packages sent between a client and a server. The SOAP protocol is popular since it communicates over HTTP, which typically allows access through company firewalls. Since the data is in XML format, different programming languages on different operating systems can send, receive, and process SOAP messages. For detailed information about the SOAP protocol, visit <http://www.w3.org/> and search for SOAP.

SOAP Web services are described via a WSDL file. A WSDL is simply an XML dataset that defines the calls, arguments, and responses in Web service interactions. A WSDL can be imported into a development environment and integrated with an application using a SOAP toolkit. For more information on setting up your development environment to create applications that can send, receive, and process SOAP messages, see [Setting Up the Development Environment \[page 15\]](#).

There are two sets of Web service calls (comprised of two separate WSDLs) that are available in SBM: services based on *administrative* functions and services based on *application* or *item* functions.

The URL for SOAP-based Web services that perform functions using SBM User Workspace items and data is: `http://serverName:aePort/gsoap/gsoap_ssl.dll?sbmappservices72`.

The URL for SOAP-based Web services that perform functions for administrative tasks is: `http://serverName:aePort/gsoap/gsoap_ssl.dll?sbmadminservices72`.

A detailed list of all supported SOAP calls can be found in [Calls Available \[page 21\]](#).

Before You Begin

Before you install and use the SBM Application Engine Web Services API, there are a few things to consider.

System Requirements

The SBM Application Engine Web Services API is supported on Windows operating systems with the Microsoft Internet Information Services (IIS) Web server. For the specific versions that are supported, refer to the SBM readme.

Licensing

Use of the SBM Application Engine Web Services API consumes a license. If a user has already checked out a license, an additional license is not checked out for use of the Web services. If the user logs out, but a Web service continues to run for that user, a license is not checked out while the Web service runs.

Additionally, when administrators or designers log in to Application Administrator or contact Application Administrator through SBM Composer to check out a file, a license is consumed. If that person already has a license checked out for the SBM Application Engine or for SBM System Administrator, SBM uses the existing license.

Security

The SBM Application Engine Web Services API supports the use of https to connect from the client. Use of SSL (Secure Sockets Layer) is recommended for any customers connecting to their Web services server in a non-secure environment. Using SSL prevents credentials from being extracted from the messages that are sent. IIS should be configured to allow or require SSL to connect to the Web services.

For information on setting up SSL, see the IIS documentation.

Authentication

SBM supports the following authentication types:

-
- LDAP (Lightweight Directory Access Protocol)
 - NT Challenge/Response
 - SBM Internal Passwords authentication

Setting Up the Development Environment

This section provides important information on:

- Generating code stubs in your integrated development environment (IDE) from the SBM Web services definition (WSDL) files.
- Setting up Microsoft Visual Studio .NET.

Generating Web Services Stubs

The SBM Web services are defined in two separate WSDL files. You can build an application that interacts with the SBM Web services in any IDE that can generate code stubs from these WSDL files. You can create the stub files using the WSDL files available at the following URLs:

- `http://serverName:aePort/gsoap/sbmappservices72.wsdl`
- `http://serverName:aePort/gsoap/sbmadminservices72.wsdl`

Setting Up Microsoft Visual Studio

To create applications that interact with the SBM Web services, you must upgrade Visual Studio and then set up Web service stub files.

Upgrading Visual Studio

To use WS-Security with the SBM Web services, you must install the Microsoft Web Service Enhancements (WSE) product and then enable the WSE for your project.

For information on WS-Security, see [Authentication Methods \[page 297\]](#).

To upgrade Visual Studio .NET 2003 or 2005:

1. Download the appropriate version of WSE from [here](#).
2. After installing the WSE, open your solution in Visual Studio .NET.
3. Right-click your project and select **WSE Settings**.
4. Select the **Enable this project for Web Services Enhancements** check box.
5. Click **OK**.

You can now start building applications using the SBM Web services with WS-Security.

Setting Up Web Service Stub Files

To access the SBM Web services from Visual Studio .NET, you must add references to the WSDL file. When you do this, Visual Studio creates stub classes that your application can use to access the Web service methods. You can create these stub classes for Visual Basic or C#.

Once you have created the stub files, you can update them to use the client protocol provided by the WSE.

To set up the Web service stub files:

1. In Visual Studio .NET, create or open a project.
2. Select Project | Add Web Reference.
3. In the dialog box that appears, enter the URL for either of the SBM Web services in the **URL** field:
 - `http://serverName:aePort/gsoap/sbmappservices72.wsdl`
 - `http://serverName:aePort/gsoap/sbmadminservices72.wsdl`
4. Enter a name in the **Web reference name** field. This name will be used in your code to refer to the web reference. The sample programs uses `aeweb` as the web reference name.
5. Click **Add Reference**.

Installing the SBM Web Services API

To use the SBM Application Engine Web Services API, you must install SBM version 2009 or later. The SBM Application Engine Web Services API is installed when you perform either a **Complete** installation or a **Custom** installation that includes the SBM Application Engine. By default, the SBM Application Engine Web Services API is installed in the following location:

```
\installDirectory\Application Engine\webservices
```

On-Demand users can access the SBM Web services here:

```
https://serenasupport.serenamashups.com/gsoap/sbmappservices72.wsdl
```

For details on the installation process, see the *Serena[®] Business Manager Installation and Configuration Guide*.

The latest Web service calls for SBM Web services version 7.2 can be found in the `sbmappservices72` and `sbmadminservices72` WSDLs. However, all TeamTrack Web services and earlier SBM Web services are still compatible with this release. These prior Web services include `ttwebservices`, `aeweb services70`, and `aeweb services71`. You can download Web services guides for earlier versions at <http://www.serena.com/support>.

It is recommended that you upgrade existing applications that use the older Web services (ttwebservices, awebservices70, and awebservices71) to use the new Web services available in SBM Web services version 7.2; however, you can continue to use existing applications in tandem with new or upgraded applications that take advantage of the enhancements introduced in SBM Web services version 7.2. To upgrade, simply update your existing endpoints with the new WSDL name and change the arguments in your existing call list as necessary. If you are currently using the calls found in ttwebservices, awebservices70, and awebservices71, compare your existing calls with those in SBM Web services version 7.2 and consider upgrading in order to take advantage of the expanded identifiers, options, and tighter processing control.



Note: As of SBM Web services version 7.2, Multi-Relational fields no longer accept a comma-separated list of internal (tableid:internal-item-id) values on update. Instead, you must specify values individually in an array of values to update a Multi-Relational field. For example:

```
<urn:extendedField>
  <urn:id>
    <urn:displayName>MULTI-RELATIONAL</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName></urn:dbName>
  </urn:id>
  <urn:setValueBy>DISPLAY-VALUE</urn:setValueBy>
  <urn:setValueMethod>APPEND-VALUES</urn:setValueMethod>
  <urn:value>
    <urn:displayValue>Value1</urn:displayValue>
    <urn:internalName></urn:internalName>
    <urn:internalValue></urn:internalValue>
    <urn:uuid></urn:uuid>
  </urn:value>
  <urn:value>
    <urn:displayValue>Value2</urn:displayValue>
    <urn:internalName></urn:internalName>
    <urn:internalValue></urn:internalValue>
    <urn:uuid></urn:uuid>
  </urn:value>
</urn:extendedField>
```

SBM provides sample programs written in C# that interact with the SBM Web services API. These samples are discussed in more detail in [Chapter 5: Sample Programs \[page 301\]](#).

Web Service and Script Execution Order

The SBM Application Engine executes Web service functions (see note below), SBM AppScript scripts, transition attribute scripts, transition actions and state actions, and process app events in the following order:

1. Web service function for the pre-transition context
2. SBM AppScript for the pre-transition context
3. Transition attribute scripts for the pre-transition context

4. Transition executed by users.
5. SBM AppScript for the post-transition context
6. Transition attribute scripts for the post-transition context
7. Web service function for the post-transition context
8. SBM AppScript for the post-state context
9. Web Service function for the post-state context
10. SBM AppScript for the pre-state context
11. Web Service function for pre-state context
12. Transition completed and recorded in the database
13. Transition actions
14. Events are emitted
15. Subtasks and posted items are submitted
16. State actions are performed



Note: An orchestration workflow with a reply is treated the same as a Web service function, in terms of when it's executed.

Error Message Logs

All Web service error messages are recorded in the Event Log on the SBM Web server. You can view the messages in the Event Viewer, along with the SBM Web server messages.

If IIS uses Anonymous Access as its authentication model, you will need to remove the IUSR account from the Guests group and add it to another group (like Users). This will allow the Web services API to write messages to the System Event Log.

SOAP Faults

SOAP Faults are generated when there is an error processing a Web service request (also known as a SOAP request). A SOAP Fault is made up of three elements:

- **faultcode**—Used by the software making the Web service call to take action based on the type of error that occurred.
- **faultstring**—Contains the human-readable, localized error message.
- **detail**—Contains exception-specific information about the error. The detail section is normally an `AEWebServicesFault` that includes the error message generated by the Web service call.

All Web service calls perform user authentication and license checking. If an error occurs as a result of authenticating a user or allocating a license, a SOAP Fault is returned. A list of the error messages follows:

-
- ae:Client.LoginNoConcurrentLicense - No concurrent licenses available
 - ae:Client.LoginConcurrentExpired - User had but lost concurrent license
 - ae:Client.LoginServerDown - LDAP Server Down
 - ae:Client.LoginUserDisabled - Disabled user account
 - ae:Client.LoginNamespaceDisabled - Disabled namespace
 - ae:Client.LoginLicenseViolation - System is in seat license violation
 - ae:Client.LoginInvalidUserCredentials - Userid or password is invalid

The faultstring and AEWebServicesFault will contain a human-readable, localized error message that can be displayed to the user in the SBM User Workspace.

Chapter 3: Web Services API Reference

This section provides an overview of all SBM Web services version 7.2 calls, arguments, and responses. For more information on calling out to Web services, see the *SBM System Administrator Guide*.

- [SBM Application Web Services \[page 21\]](#)
- [SBM Administrative Web Services \[page 212\]](#)

SBM Application Web Services

This section contains reference material for all of the SBM Application Web Services (as described in the `sbmappservices72` WSDL).

- [Calls Available \[page 21\]](#)
- [Common Types \[page 83\]](#)
- [Arguments \[page 139\]](#)
- [Responses \[page 166\]](#)

Calls Available

This section describes the Application Web service calls that are available in SBM. These calls represent the Web service operations that are invoked from a client and performed on the SBM Application Engine Web Server. The calls receive one or more arguments from the client, perform an operation on the server, and return an XML response to the client when applicable.

The following table lists all supported calls in alphabetical order, followed by a brief description of each operation. Select a call to view detailed information including:

- **Description**
A brief description of the call.
- **Arguments**
A table describing the arguments for each call. Both simple and complex types are listed for each argument. For each complex type argument, you can click the argument name for a detailed description.
- **Response**
A brief description of the call's response. For each complex type response, you can click the response name for a detailed description.
- **Usage**
Any notes, additional details, and concerns regarding the call are addressed here.
- **Faults**

Possible error values are listed here.

- **XML**

An example of the actual XML payload that is sent is displayed here. The XML not only shows the call and its respective elements, you can also see detailed examples of each argument and how to format the expected data.

Calls Available

Call	Description
CreateAuxItem [page 24]	This service creates a single auxiliary item within the same table.
CreateAuxItems [page 27]	This service creates multiple auxiliary items within the same table.
CreateFileAttachment [page 30]	This service creates a new file attachment associated with an item.
CreateNoteAttachment [page 32]	This service creates a new note on an existing item.
CreatePrimaryItem [page 33]	This service submits a single primary item within the specified project.
CreatePrimaryItems [page 37]	This service submits multiple primary items within the same project using the data supplied.
DeleteAttachment [page 43]	This service deletes an existing attachment, which can be a note, item link, URL attachment, or file attachment.
DeleteItems [page 44]	This service uses the delete transition to delete multiple items.
DeleteItemsByQuery [page 45]	This service deletes all the items that match the specified <i>where</i> clause.
GetApplications [page 47]	This service returns a list of available applications.

Call	Description
GetAvailableSubmitTransitions [page 48]	This service returns all submit transitions for the specified project.
GetAvailableTransitions [page 49]	This service returns a list of available transitions for the specified item.
GetFileAttachment [page 50]	This service gets an existing file attachment.
GetItem [page 52]	This service returns a single item, given the table ID and internal item ID.
GetItems [page 53]	This service returns one or more items, given the table ID and internal item ID for each item.
GetItemsByQuery [page 56]	This service returns multiple items found using a <i>where</i> clause and an <i>order by</i> clause to determine the set of items returned.
GetNoteLoggerInfo [page 59]	This service returns the e-mail address of the E-mail Recorder feature.
GetReports [page 60]	This service returns a list of reports within a specified range, limited by one or more optional filters.
GetSolutions [page 62]	This service returns a list of solutions that can be accessed by the user.
GetStateChangeHistory [page 63]	This service returns a specified range of state change history for an item.
GetSubmitProjects [page 64]	This service returns a list of projects into which the user can submit items.

Call	Description
GetTables [page 65]	This service returns a list of tables optionally filtered by solution or table type.
GetUsers [page 67]	This service returns one or more user records.
GetVersion [page 69]	This service returns the SBM version number.
IsValidUser [page 70]	This service determines whether a specified user is valid or not.
LinkSubtask [page 71]	This service links one item to another to create a subtask relationship.
Logout [page 72]	This service releases any licenses and resources associated with the session.
RunReport [page 73]	This service runs a specified report, given the proper privileges.
UpdateFileAttachment [page 75]	This service updates an existing file attachment for a specified item.
TransitionItem [page 77]	This service transitions an existing item using a specified transition.
TransitionItems [page 80]	This service transitions one or more existing items using a specified transition.

CreateAuxItem

Description

This service creates a single auxiliary item in the specified table.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The Auth type supplies credentials and optionally, a host name for licensing. The userId and password can be specified with HTTP BASIC or WS-SECURITY instead.
table (required)	TableIdentifier [page 109]	Specifies the table where the item is created.
item (required)	TTItem [page 114]	The items to be created. The TTItem types hold the generic data for the item.
options (optional)	ResponseItemOptions [page 154]	Enables you to limit the data that is returned in the response.

Response

TTItemHolder is returned for the item that is specified in the call. The new auxiliary item is returned with updated item data, which shows the unique TS_ID of the record and the TS_ID of the table to which it was added. For more detail, see [TTItemHolder \[page 198\]](#)

Usage

The CreateAuxItem call provides a method to add a new record to a given auxiliary table. If you have the proper privileges, you can add new records to both custom and system auxiliary tables.

To create notes, item links, and URL attachments on the new auxiliary item, add records to the lists that are defined in TTItem. To create a file attachment, see [CreateFileAttachment \[page 30\]](#).



Tip: You must have the table ID and item ID of the auxiliary item prior to attaching a file to the item because the IDs are required in the ItemIdentifier argument of CreateFileAttachment.

Use the following elements in the options parameter to control how this call is processed and what data is returned:

- **sections** and **specifiedSections** — Enables you to specify which parts of an item should be returned. This allows you to limit the data that is returned for a given item. The sections that aren't specified are not included in the response. For example, if the items you are creating have a large number of item links, notes, URL and file attachments that don't need to be returned in the response, use the sections parameter to return only the sections of an item you want.
- **limitedField** — Enables you to specify which fields you want returned in the response.

For more information on the options elements, see [ResponseItemOptions \[page 154\]](#).

Faults

- Invalid database pointer.

- The table ID is not valid.
- The user lacks sufficient permission.
- Creating the record fails.
- Reading the item fails.

XML

The following XML is a snippet of the payload that is sent with CreateAuxItem.

```
<urn:CreateAuxItem>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:table>
    <urn:displayName></urn:displayName>
    <urn:id>1004</urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName></urn:dbName>
  </urn:table>
  <urn:item>
    <urn:id>
      <urn:displayName></urn:displayName>
      <urn:id></urn:id>
      <urn:uuid></urn:uuid>
      <urn:tableId></urn:tableId>
      <urn:tableIdItemId></urn:tableIdItemId>
      <urn:issueId></urn:issueId>
    </urn:id>
    <urn:itemType></urn:itemType>
    <urn:title>New Aux Item 1</urn:title>
    <urn:description></urn:description>
    <urn:subtasks/>
    <urn:extendedField>
      <urn:id>
        <urn:displayName>Field Name</urn:displayName>
        <urn:id>153</urn:id>
        <urn:uuid></urn:uuid>
        <urn:dbName>FIELD_NAME</urn:dbName>
      </urn:id>
      <urn:setValueBy>DISPLAY-VALUE</urn:setValueBy>
      <urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
      <urn:value>
        <urn:displayValue>test text</urn:displayValue>
        <urn:internalValue></urn:internalValue>
        <urn:uuid></urn:uuid>
      </urn:value>
    </urn:extendedField>
  </urn:item>
  <urn:options>
    <urn:extraOption>
```

```

        <urn:name></urn:name>
        <urn:value></urn:value>
    </urn:extraOption>
    <urn:sections>SECTIONS-SPECIFIED</urn:sections>
    <urn:specifiedSections>SECTION:FIXED,SECTION:EXTENDED</urn:specifiedSections>
    <urn:limitedField></urn:limitedField>
</urn:options>
</urn:CreateAuxItem>

```

CreateAuxItems

Description

This service creates one or more auxiliary items within the same table.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The Auth type supplies credentials and optionally, a host name for licensing. The userId and password can be specified with HTTP BASIC or WS-SECURITY instead.
table (required)	TableIdentifier [page 109]	Specifies the table where the item or items are created.
item (required)	TTItem [page 114]	The list of items that are created. The TTItem types hold the generic data for each item.
options (optional)	MultipleResponseItemOptions [page 146]	Specifies whether the service should continue if an error is encountered, or stop. Also enables you to limit the data that is returned in the response.

Response

TTItemHolder is returned, one for each item that is specified in the call. The new auxiliary items are returned with updated item data, which shows the unique TS_IDs of each record and TS_IDs of the table to which they were added. For more detail, see [TTItemHolder \[page 198\]](#)

Usage

The CreateAuxItems call provides a method to add new records to a given auxiliary table. If you have the proper privileges, you can add new records to both custom and system auxiliary tables.

To create notes, item links, and URL attachments on the new auxiliary items, add records to the lists that are defined in `TTItem`. To create a file attachment, see [CreateFileAttachment \[page 30\]](#).



Tip: You must have the table ID and item ID of the auxiliary item prior to attaching a file to the item because the IDs are required in the `ItemIdentifier` argument of `CreateFileAttachment`.

Use the following elements in the options parameter to control how this call is processed and what data is returned:

- **multiOption** — Enables you to specify whether the service should continue if an error is encountered, or stop and throw an error. If any failures occur, each successive error message is appended to the string that is returned. Multiple error messages are separated by a single newline. Failures do not result in a return before all items have been processed.
- **sections** and **specifiedSections** — Enables you to specify which parts of an item should be returned. This allows you to limit the data that is returned for a given item. The sections that aren't specified are not included in the response. For example, if the items you are creating have a large number of item links, notes, URL and file attachments that don't need to be returned in the response, use the sections parameter to return only the sections of an item you want.
- **limitedField** — Enables you to specify which fields you want returned in the response. For example, you can specify one or more fields to limit a service response to return only the fields that you want returned.

For more information on the options elements, see [MultipleResponseItemOptions \[page 146\]](#).

Faults

- Invalid database pointer.
- The table ID is not valid.
- The user lacks sufficient permission.
- Creating the record fails.
- Reading the item fails.

XML

The following XML is a snippet of the payload that is sent with `CreateAuxItems`.

```
<urn:CreateAuxItems>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:table>
    <urn:displayName></urn:displayName>
    <urn:id>1004</urn:id>
```

```

    <urn:uuid></urn:uuid>
    <urn:dbName></urn:dbName>
</urn:table>
<urn:item>
  <urn:id>
    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:tableId></urn:tableId>
    <urn:tableIdItemId></urn:tableIdItemId>
    <urn:issueId></urn:issueId>
  </urn:id>
  <urn:itemType></urn:itemType>
  <urn:title>New Aux Item 1</urn:title>
  <urn:description></urn:description>
  <urn:subtasks/>
  <urn:extendedField>
    <urn:id>
      <urn:displayName>Field Name</urn:displayName>
      <urn:id>153</urn:id>
      <urn:uuid></urn:uuid>
      <urn:dbName>FIELD_NAME</urn:dbName>
    </urn:id>
    <urn:setValueBy>DISPLAY-VALUE</urn:setValueBy>
    <urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
    <urn:value>
      <urn:displayValue>test text</urn:displayValue>
      <urn:internalValue></urn:internalValue>
      <urn:uuid></urn:uuid>
    </urn:value>
  </urn:extendedField>
</urn:item>
<urn:item>
  <urn:id>
    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:tableId></urn:tableId>
    <urn:tableIdItemId></urn:tableIdItemId>
    <urn:issueId></urn:issueId>
  </urn:id>
  <urn:itemType></urn:itemType>
  <urn:title>New Aux Item 2</urn:title>
  <urn:description></urn:description>
  <urn:extendedField>
    <urn:id>
      <urn:displayName>Field Name</urn:displayName>
      <urn:id>153</urn:id>
      <urn:uuid></urn:uuid>
      <urn:dbName>FIELD_NAME</urn:dbName>
    </urn:id>
    <urn:setValueBy>DISPLAY-VALUE</urn:setValueBy>
    <urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
    <urn:value>
      <urn:displayValue>test text</urn:displayValue>

```

```

        <urn:internalValue></urn:internalValue>
        <urn:uuid></urn:uuid>
    </urn:value>
</urn:extendedField>
</urn:item>
<urn:options>
    <urn:extraOption>
        <urn:name></urn:name>
        <urn:value></urn:value>
    </urn:extraOption>
    <urn:sections>SECTIONS-SPECIFIED</urn:sections>
    <urn:specifiedSections>SECTION:FIXED,SECTION:EXTENDED</urn:specifiedSections>
    <urn:limitedField></urn:limitedField>
    <urn:multiOption>CONTINUE-ON-FAILURE</urn:multiOption>
</urn:options>
</urn:CreateAuxItems>


```

CreateFileAttachment

Description

This service creates a new file attachment associated with an item.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The <i>Auth</i> type supplies credentials and optionally, a host name for licensing. The <i>userId</i> and <i>password</i> can be specified with HTTP BASIC or WS-SECURITY instead.
item (required)	ItemIdentifier [page 95]	Specifies the item that receives the new file attachment.
attachmentContents (required)	FileAttachmentContents [page 91]	<p>The file attachment details and content.</p> <p> Note: attachmentContents is of type FileAttachmentContents, but also includes attachment detail found in FileAttachment. The XML example below shows the parameters from both FileAttachment and FileAttachmentContents. See FileAttachment [page 90] for more information.</p>

Argument	Type	Description
options (optional)	Options [page 151]	Holds name value pairing for future arguments.

Response

FileAttachment is returned. The newly added file attachment details are returned (not the content itself). For more detail, see [FileAttachment \[page 90\]](#).

Usage

The CreateFileAttachment call provides a method to add a single attachment to an auxiliary or primary item, given the proper privileges. If the call fails, a file attachment will not be added. To create multiple file attachments for a single item, CreateFileAttachment must be called for each attachment. You must have the table ID and item ID of the auxiliary or primary item prior to calling CreateFileAttachment because the IDs are required in the ItemIdentifier argument.



Note: The file to be attached is sent as a base64 encoded attachment.

Faults

- Invalid database pointer.
- The item ID is not valid.
- The user lacks sufficient permission.
- Creating the record fails.
- Failed to create the attachment.

XML

The following XML is a snippet of the payload that is sent with CreateFileAttachment.

```
<urn:CreateFileAttachment>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname>localhost</urn:hostname>
  </urn:auth>
  <urn:item>
    <urn:displayName></urn:displayName>
    <urn:id>109</urn:id>
    <urn:uuid></urn:uuid>
    <urn:tableId>1000</urn:tableId>
    <urn:tableIdItemId></urn:tableIdItemId>
    <urn:issueId></urn:issueId>
  </urn:item>
  <urn:attachmentContents>
    <urn:id>16</urn:id>
    <urn:name>pdf_doc</urn:name>
    <urn:fileName>relnotes.pdf</urn:fileName>
  </urn:attachmentContents>
</urn:CreateFileAttachment>
```

```

    <urn:showAsImage>false</urn:showAsImage>
    <urn:modificationDateTime></urn:modificationDateTime>
    <urn:url></urn:url>
    <urn:accessType>ATTACHACCESS-DEFAULT</urn:accessType>
    <urn:contentsBase64>
      <urn:data>cid:981662964041</urn:data>
    </urn:contentsBase64>
  </urn:attachmentContents>
</urn:CreateFileAttachment>

```

CreateNoteAttachment

Description

This service creates a new note attachment in an existing item.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The Auth type supplies credentials and optionally, a host name for licensing. The userId and password can be specified with HTTP BASIC or WS-SECURITY instead.
item (required)	ItemIdentifier [page 95]	Specifies the item that receives the new note.
author (optional)	UserIdentifier [page 138]	Indicates the author of the note.
noteContents (required)	NoteAttachmentContents [page 150]	Holds the time, title, content, and access-type for the note.

Response

A boolean is returned. True indicates the note was created successfully; false indicates the note was not created.

```

<ae:CreateNoteAttachmentResponse>
  <ae:return>true</ae:return>
</ae:CreateNoteAttachmentResponse>

```

To verify the contents of the note, view the `<ae:note>` element in the `TTItem` response of the `GetItem` call.

Usage

The `CreateNoteAttachment` call provides a method to add a single note to an auxiliary or primary item, given the proper privileges. If the call fails, the note is not be added. To create multiple notes for a single item, you must call `CreateNoteAttachment` for each note. You must have the table ID and item ID of the auxiliary or primary item prior to calling `CreateNoteAttachment` because the IDs are required in the `ItemIdentifier` argument.

Faults

- Invalid database pointer.
- The item ID is not valid.
- The user lacks sufficient permission.
- Creating the record fails.
- Failed to create the note.

XML

The following XML is a snippet of the payload that is sent with CreateNoteAttachment.

```
<urn:CreateNoteAttachment>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:item>
    <urn:displayName></urn:displayName>
    <urn:id>142</urn:id>
    <urn:uuid></urn:uuid>
    <urn:tableId>1000</urn:tableId>
    <urn:tableIdItemId></urn:tableIdItemId>
    <urn:issueId></urn:issueId>
  </urn:item>
  <urn:author>
    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:loginId></urn:loginId>
  </urn:author>
  <urn:noteContents>
    <urn:time></urn:time>
    <urn:title>Note Title</urn:title>
    <urn:body>This is a note.</urn:body>
    <urn:accessType>ATTACHACCESS-DEFAULT</urn:accessType>
  </urn:noteContents>
</urn:CreateNoteAttachment>
```

CreatePrimaryItem

Description

This service creates a single primary item within the same project using the data that is supplied.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The Auth type supplies credentials and optionally, a host name for licensing. The userId and password can be specified with HTTP BASIC or WS-SECURITY instead.
project (required)	ProjectIdentifier [page 100]	The project in which the new item will be created. You must at least specify the TS_ID of the project.
parentItem	ItemIdentifier [page 95]	If specified, the item that is created using CreatePrimaryItem will become a sub-item of this item.
item (required)	TTItem [page 114]	Holds one the item to be created. The TTItem type holds the generic data for the item.
submitTransition (optional)	TransitionIdentifier [page 111]	Only used if you want to use an alternate submit transition.
options (optional)	ResponseItemOptions [page 154]	Enables you to limit the data that is returned in the response.

Response

TTItemHolder is returned for the item that is specified in the call. The new primary item is returned with updated item data, which shows the unique TS_ID of the record and the TS_ID of the table to which it was added. For more detail, see [TTItemHolder \[page 198\]](#).

Usage

The CreatePrimaryItem call provides a method to add a single new record to a given primary table. You can add new records to both custom and system primary tables, given the proper privileges.

To create notes, item links, and URL attachments on the new primary item, add records to these elements as defined in TTItem. To create a file attachment, see [CreateFileAttachment \[page 30\]](#).



Tip: You must have the table ID and item ID of the primary item prior to attaching a file to the item because the IDs are required in the ItemIdentifier argument of CreateFileAttachment.

Use the following elements in the options parameter to control how this call is processed and what data is returned:

- **sections** and **specifiedSections** — Enables you to specify which parts of an item should be returned. This allows you to limit the data that is returned for a given item. The sections that aren't specified are not included in the response. For example, if the items you are creating have a large number of item links, notes, URL

and file attachments that don't need to be returned in the response, use the sections parameter to return only the sections of an item you want.

- **limitedField** — Enables you to specify which fields you want returned in the response.

For more information on the options elements, see [ResponseItemOptions \[page 154\]](#).

Faults

- Invalid database pointer.
- The project ID is not valid.
- The user lacks sufficient permission.
- Creating the record fails.
- The submit transition fails.
- Validation constraint violation: data type mismatch in element

XML

The following XML is a snippet of the payload that is sent with CreatePrimaryItem.

```
<urn:CreatePrimaryItem>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:project>
    <urn:displayName>Animation Pro</urn:displayName>
    <urn:id>2</urn:id>
    <urn:uuid></urn:uuid>
    <urn:fullyQualifiedName>Base Project||Base IDT Project||
    Software Development||Animation Pro</urn:fullyQualifiedName>
  </urn:project>
  <urn:parentItem>
</urn:parentItem>
  <urn:item>
    <urn:id>
      <urn:displayName></urn:displayName>
      <urn:id></urn:id>
      <urn:uuid></urn:uuid>
      <urn:tableId></urn:tableId>
      <urn:tableIdItemId></urn:tableIdItemId>
      <urn:issueId></urn:issueId>
    </urn:id>
    <urn:itemType></urn:itemType>
  <urn:project>
    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:fullyQualifiedName></urn:fullyQualifiedName>
```

```
</urn:project>
<urn:title>Test item</urn:title>
<urn:description>This is a test item.</urn:description>
<urn:createdBy>
  <urn:displayName></urn:displayName>
  <urn:id></urn:id>
  <urn:uuid></urn:uuid>
  <urn:loginId></urn:loginId>
</urn:createdBy>
<urn:createDate></urn:createDate>
<urn:modifiedBy>
  <urn:displayName></urn:displayName>
  <urn:id></urn:id>
  <urn:uuid></urn:uuid>
  <urn:loginId></urn:loginId>
</urn:modifiedBy>
<urn:modifiedDate></urn:modifiedDate>
<urn:activeInactive></urn:activeInactive>
<urn:state>
  <urn:displayName></urn:displayName>
  <urn:id></urn:id>
  <urn:uuid></urn:uuid>
  <urn:isClosed></urn:isClosed>
</urn:state>
<urn:owner>
  <urn:displayName></urn:displayName>
  <urn:id></urn:id>
  <urn:uuid></urn:uuid>
  <urn:loginId></urn:loginId>
</urn:owner>
<urn:url/>
<urn:subtasks/>
<urn:extendedField>
  <urn:id>
    <urn:displayName>Severity</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName>SEVERITY</urn:dbName>
  </urn:id>
  <urn:setValueBy>DISPLAY-VALUE</urn:setValueBy>
  <urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
  <urn:value>
    <urn:displayValue>Critical</urn:displayValue>
    <urn:internalValue></urn:internalValue>
    <urn:uuid></urn:uuid>
  </urn:value>
</urn:extendedField>
<urn:extendedField>
  <urn:id>
    <urn:displayName>How Found</urn:displayName>
    <urn:id>74</urn:id>
    <urn:uuid>b999082f-ef27-47c9-890f-b4d80a3c4c23</urn:uuid>
    <urn:dbName>HOW_FOUND</urn:dbName>
  </urn:id>
  <urn:setValueBy>INTERNAL-VALUE</urn:setValueBy>
```

```

        <urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
        <urn:value>
            <urn:displayValue>Code Review</urn:displayValue>
            <urn:internalValue>24</urn:internalValue>
            <urn:uuid>931259b4-dc0a-46c5-b567-ff04dd5c9395</urn:uuid>
        </urn:value>
    </urn:extendedField>
    <urn:extendedData>
        <urn:data>
            <urn:name></urn:name>
            <urn:value></urn:value>
        </urn:data>
    </urn:extendedData>
</urn:item>
<urn:submitTransition>
    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
</urn:submitTransition>
<urn:options>
    <urn:extraOption>
        <urn:name></urn:name>
        <urn:value></urn:value>
    </urn:extraOption>
    <urn:sections>SECTIONS-ALL</urn:sections>
    <urn:specifiedSections></urn:specifiedSections>
    <urn:limitedField/>
</urn:options>
</urn>CreatePrimaryItem>

```

CreatePrimaryItems

Description

This service creates one or more primary items within the same project using the data that is supplied.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The Auth type supplies credentials and optionally, a host name for licensing. The userId and password can be specified with HTTP BASIC or WS-SECURITY instead.
project (required)	ProjectIdentifier [page 100]	The project in which new items are created. You must at least specify the TS_ID of the project.

Argument	Type	Description
parentItem	ItemIdentifier [page 95]	If specified, items that are created using CreatePrimaryItems will become sub-items of this item.
item (required)	TTItem [page 114]	Holds one or more items to be created. The TTItem type holds the generic data for each item.
submitTransition (optional)	TransitionIdentifier [page 111]	Only used if you want to use an alternative submit transition.
options (optional)	MultipleResponseItemOptions [page 146]	Specifies whether the service should continue if an error is encountered or stop. Also enables you to limit the data that is returned in the response.

Response

TTItemHolder is returned, one for each item that is specified in the call. The new primary items are returned with updated item data, which shows the unique TS_IDs of each record and TS_IDs of the table to which they were added. For more detail, see [TTItemHolder \[page 198\]](#).

Usage

The CreatePrimaryItems call provides a method to add multiple new records to a given primary table. You can add new records to both custom and system primary tables, given the proper privileges.

To create notes, item links, and URL attachments on the new primary items, add records to these elements as defined in TTItem. To create a file attachment, see [CreateFileAttachment \[page 30\]](#).



Tip: You must have the table ID and item ID of the primary item prior to attaching a file to the item because the IDs are required in the ItemIdentifier argument of CreateFileAttachment.

Use the following elements in the options parameter to control how this call is processed and what data is returned:

- **multiOption** — Enables you to specify whether the service should continue if an error is encountered, or stop and throw an error. If any failures occur, each successive error message is appended to the string that is returned. Multiple error messages are separated by a single newline. Failures do not result in a return before all items have been processed.
- **sections** and **specifiedSections** — Enables you to specify which parts of an item should be returned. This allows you to limit the data that is returned for a given item. The sections that aren't specified are not included in the response. For example, if the items you are creating have a large number of item links, notes, URL and file attachments that don't need to be returned in the response, use the sections parameter to return only the sections of an item you want.

-
- **limitedField** — Enables you to specify which fields you want returned in the response. For example, you can specify one or more fields to limit a service response to return only the fields that you want returned.

For more information on the options elements, see [MultipleResponseItemOptions](#) [page 146].

Faults

- Invalid database pointer.
- The project ID is not valid.
- The user lacks sufficient permission.
- Creating the record fails.
- The submit transition fails.
- Validation constraint violation: data type mismatch in element

XML

The following XML is a snippet of the payload that is sent with `CreatePrimaryItems`.

```
<urn:CreatePrimaryItems>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:project>
    <urn:displayName>Animation Pro</urn:displayName>
    <urn:id>2</urn:id>
    <urn:uuid></urn:uuid>
    <urn:fullyQualifiedName>Base Project||Base IDT Project||
    Software Development||Animation Pro</urn:fullyQualifiedName>
  </urn:project>
  <urn:parentItem>
</urn:parentItem>
  <urn:item>
    <urn:id>
      <urn:displayName></urn:displayName>
      <urn:id></urn:id>
      <urn:uuid></urn:uuid>
      <urn:tableId></urn:tableId>
      <urn:tableIdItemId></urn:tableIdItemId>
      <urn:issueId></urn:issueId>
    </urn:id>
    <urn:itemType></urn:itemType>
    <urn:project>
      <urn:displayName></urn:displayName>
      <urn:id></urn:id>
      <urn:uuid></urn:uuid>
      <urn:fullyQualifiedName></urn:fullyQualifiedName>
    </urn:project>
  </urn:item>
</urn:CreatePrimaryItems>
```

```
<urn:title>Test item</urn:title>
<urn:description>This is a test item.</urn:description>
<urn:createdBy>
  <urn:displayName></urn:displayName>
  <urn:id></urn:id>
  <urn:uuid></urn:uuid>
  <urn:loginId></urn:loginId>
</urn:createdBy>
<urn:createDate></urn:createDate>
<urn:modifiedBy>
  <urn:displayName></urn:displayName>
  <urn:id></urn:id>
  <urn:uuid></urn:uuid>
  <urn:loginId></urn:loginId>
</urn:modifiedBy>
<urn:modifiedDate></urn:modifiedDate>
<urn:activeInactive></urn:activeInactive>
<urn:state>
  <urn:displayName></urn:displayName>
  <urn:id></urn:id>
  <urn:uuid></urn:uuid>
  <urn:isClosed></urn:isClosed>
</urn:state>
<urn:owner>
  <urn:displayName></urn:displayName>
  <urn:id></urn:id>
  <urn:uuid></urn:uuid>
  <urn:loginId></urn:loginId>
</urn:owner>
<urn:url/>
<urn:subtasks/>
<urn:extendedField>
  <urn:id>
    <urn:displayName>Severity</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName>SEVERITY</urn:dbName>
  </urn:id>
  <urn:setValueBy>DISPLAY-VALUE</urn:setValueBy>
  <urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
  <urn:value>
    <urn:displayValue>Critical</urn:displayValue>
    <urn:internalValue></urn:internalValue>
    <urn:uuid></urn:uuid>
  </urn:value>
</urn:extendedField>
<urn:extendedField>
  <urn:id>
    <urn:displayName>How Found</urn:displayName>
    <urn:id>74</urn:id>
    <urn:uuid>b999082f-ef27-47c9-890f-b4d80a3c4c23</urn:uuid>
    <urn:dbName>HOW_FOUND</urn:dbName>
  </urn:id>
  <urn:setValueBy>INTERNAL-VALUE</urn:setValueBy>
  <urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
```

```

        <urn:value>
            <urn:displayValue>Code Review</urn:displayValue>
            <urn:internalValue>24</urn:internalValue>
            <urn:uuid>931259b4-dc0a-46c5-b567-ff04dd5c9395</urn:uuid>
        </urn:value>
    </urn:extendedField>
</urn:extendedData>
    <urn:data>
        <urn:name></urn:name>
        <urn:value></urn:value>
    </urn:data>
</urn:extendedData>
</urn:item>
<urn:item>
    <urn:id>
        <urn:displayName></urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
        <urn:tableId></urn:tableId>
        <urn:tableIdItemId></urn:tableIdItemId>
        <urn:issueId></urn:issueId>
    </urn:id>
    <urn:itemType></urn:itemType>
    <urn:project>
        <urn:displayName></urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
        <urn:fullyQualifiedNames></urn:fullyQualifiedNames>
    </urn:project>
    <urn:title>Test item</urn:title>
    <urn:description>This is a second test item.</urn:description>
    <urn:createdBy>
        <urn:displayName></urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
        <urn:loginId></urn:loginId>
    </urn:createdBy>
    <urn:createDate></urn:createDate>
    <urn:modifiedBy>
        <urn:displayName></urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
        <urn:loginId></urn:loginId>
    </urn:modifiedBy>
    <urn:modifiedDate></urn:modifiedDate>
    <urn:activeInactive></urn:activeInactive>
    <urn:state>
        <urn:displayName></urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
        <urn:isClosed></urn:isClosed>
    </urn:state>
    <urn:owner>
        <urn:displayName></urn:displayName>
        <urn:id></urn:id>

```

```
        <urn:uuid></urn:uuid>
        <urn:loginId></urn:loginId>
    </urn:owner>
    <urn:url/>
    <urn:subtasks/>
    <urn:extendedField>
        <urn:id>
            <urn:displayName>Severity</urn:displayName>
            <urn:id></urn:id>
            <urn:uuid></urn:uuid>
            <urn:dbName>SEVERITY</urn:dbName>
        </urn:id>
        <urn:setValueBy>DISPLAY-VALUE</urn:setValueBy>
        <urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
        <urn:value>
            <urn:displayValue>Critical</urn:displayValue>
            <urn:internalValue></urn:internalValue>
            <urn:uuid></urn:uuid>
        </urn:value>
    </urn:extendedField>
    <urn:extendedField>
        <urn:id>
            <urn:displayName>How Found</urn:displayName>
            <urn:id>74</urn:id>
            <urn:uuid>b999082f-ef27-47c9-890f-b4d80a3c4c23</urn:uuid>
            <urn:dbName>HOW_FOUND</urn:dbName>
        </urn:id>
        <urn:setValueBy>INTERNAL-VALUE</urn:setValueBy>
        <urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
        <urn:value>
            <urn:displayValue>Code Review</urn:displayValue>
            <urn:internalValue>24</urn:internalValue>
            <urn:uuid>931259b4-dc0a-46c5-b567-ff04dd5c9395</urn:uuid>
        </urn:value>
    </urn:extendedField>
    <urn:extendedData>
        <urn:data>
            <urn:name></urn:name>
            <urn:value></urn:value>
        </urn:data>
    </urn:extendedData>
</urn:item>
<urn:submitTransition>
    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
</urn:submitTransition>
<urn:options>
    <urn:extraOption>
        <urn:name></urn:name>
        <urn:value></urn:value>
    </urn:extraOption>
    <urn:sections>SECTIONS-ALL</urn:sections>
    <urn:specifiedSections></urn:specifiedSections>
</urn:limitedField/>
```

```
    <urn:multiOption>STOP-ON-FAILURE</urn:multiOption>
  </urn:options>
</urn:CreatePrimaryItems>
```

DeleteAttachment

Description

This service deletes an existing attachment, which can be a note, item link, URL attachment, or file attachment.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The <i>Auth</i> type supplies credentials and optionally, a host name for licensing. The <i>userId</i> and <i>password</i> can be specified with HTTP BASIC or WS-SECURITY instead.
attachmentID (required)	integer	This is the internal <i>TS_ID</i> of the attachment from the <i>TS_ATTACHMENTS</i> table.
options	Options [page 151]	Holds name value pairing for future arguments.

Response

An empty XML response is returned:

```
<ae>DeleteAttachmentResponse></ae>DeleteAttachmentResponse>
```

and the specified note, item link, URL, or file attachment is deleted.

Usage

You can use [GetItems \[page 53\]](#) to find the *TS_ID* of the attachment to be deleted. If the item has any attachments, they are listed in the `<ae:note>`, `<ae:itemLink>`, `<ae:urlAttachment>`, or `<ae:fileAttachment>` parameters.

Faults

- Invalid database pointer.
- The attachment ID is not valid.
- Creating the record fails.
- The user lacks sufficient permission.
- Failed to delete the attachment.

XML

The following XML is a snippet of the payload that is sent with `DeleteAttachment`.

```

<urn:DeleteAttachment>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname>localhost</urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:attachmentID>47</urn:attachmentID>
</urn:DeleteAttachment>

```

DeleteItems

Description

This service uses the delete transition to delete multiple items.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The <i>Auth</i> type supplies credentials and optionally, a host name for licensing. The <i>userId</i> and <i>password</i> can be specified with HTTP BASIC or WS-SECURITY instead.
itemID (required)	ItemIdentifier [page 95]	The item or items that you want to delete.
options	MultipleOptions [page 145]	Holds name value pairing for future elements and enumeration to determine if the service should stop on failure and send an error message or continue processing.

Response

An empty XML response is returned:

```
<ae:DeleteItemsResponse></ae:DeleteItemsResponse>
```

and the items are deleted using the default delete transition. Failure will not delete items.

Usage

This call permanently deletes the primary or auxiliary items you specify. Any items referring to these items will be reduced or set to (None). All items in the list are processed. If any failures occur, each successive error message is appended to the string that is returned. Multiple error messages are separated by a single newline. Failures do not result in a return before all items have been processed.

Faults

- Invalid database pointer.
- The item ID is not valid.
- The user lacks sufficient permission.

- The delete transition fails to execute.

XML

The following XML is a snippet of the payload being sent with DeleteItems.

```
<urn:DeleteItems>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname>localhost</urn:hostname>
  </urn:auth>
  <urn:itemId>
    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:tableId></urn:tableId>
    <urn:tableIdItemId>1000:153</urn:tableIdItemId>
    <urn:issueId></urn:issueId>
  </urn:itemId>
  <urn:itemId>
    <urn:displayName></urn:displayName>
    <urn:id>154</urn:id>
    <urn:uuid></urn:uuid>
    <urn:tableId>1000</urn:tableId>
    <urn:tableIdItemId></urn:tableIdItemId>
    <urn:issueId></urn:issueId>
  </urn:itemId>
  <urn:options>
    <urn:extraOption></urn:extraOption>
    <urn:multiOption>CONTINUE-ON-FAILURE</urn:multiOption>
  </urn:options>
</urn:DeleteItems>
```

DeleteItemsByQuery

Description

This service deletes all the items that match the specified *where* clause.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The <i>Auth</i> type supplies credentials and optionally, a host name for licensing. The <i>userId</i> and <i>password</i> can be specified with HTTP BASIC or WS-SECURITY instead.
table (required)	TableIdentifier [page 109]	Specifies the table in which items are deleted.

Argument	Type	Description
queryWhereClause (required)	string	An SQL statement to find items with TS_ID>0. Sending an empty <i>where</i> clause will delete all items.
options	MultipleOptions [page 145]	Holds name value pairing for future elements and enumeration to determine if the service should stop on failure and send an error message or continue processing.

Response

An empty XML response is returned:

```
<ae:DeleteItemByQueryResponse></ae:DeleteItemsByQueryResponse>
```

and the items are deleted using the default delete transition. Failure will not delete items.

Usage

This call permanently deletes the primary or auxiliary items you specify. Any items referring to these items will be reduced or set to (None). The items that are deleted are determined via the table and *where* clause.



Note: You need only pass the conditions in the *where* clause itself. For example:

```
<urn:queryWhereClause>TS_ISSUEID LIKE '00106' OR TS_ISSUEID LIKE '00029'  
OR TS_ISSUEID LIKE '00105'</urn:queryWhereClause>
```

```
<urn:queryWhereClause>TS_ISSUEID='00038'</urn:queryWhereClause>
```

```
<urn:queryWhereClause>TS_ID='4'</urn:queryWhereClause>
```

To delete all items, send a null query:

```
<urn:queryWhereClause></urn:queryWhereClause>
```

Faults

- Invalid database pointer.
- The table ID is not valid.
- The user lacks sufficient permission.
- The delete transition fails to execute.

XML

The following XML is a snippet of the payload that is sent with DeleteItemsByQuery.

```

<urn:DeleteItemsByQuery>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
  </urn:auth>
  <urn:table>
    <urn:displayName></urn:displayName>
    <urn:id>1000</urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName></urn:dbName>
  </urn:table>
  <urn:queryWhereClause>TS_ISSUEID LIKE '000208' OR TS_ISSUEID LIKE '000209'
OR TS_ISSUEID LIKE '000210'</urn:queryWhereClause>
  <urn:options>
    <urn:extraOption></urn:extraOption>
    <urn:multiOption>CONTINUE-ON-FAILURE</urn:multiOption>
  </urn:options>
</urn:DeleteItemsByQuery>

```

GetApplications

Description

This service returns identification information for available applications in an SBM Application Engine database.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The <i>Auth</i> type supplies credentials and optionally, a host name for licensing. The <i>userId</i> and <i>password</i> can be specified with HTTP BASIC or WS-SECURITY instead.
options	Options [page 151]	Holds name value pairing for future arguments on certain calls.

Response

ApplicationData (a list of applications) is returned in the response. Each return element shows the *ApplicationIdentifier*, the description, the application definition UUID, and the revision number. See [ApplicationData](#) [page 170] for more information.

Usage

Use *GetApplications* to return the available applications and information about each revision. This information is useful when you need to retrieve historical information about deployed applications.

Faults

- Invalid database pointer.
- No applications are available for the user.

XML

The following XML is a snippet of the payload that is sent with GetApplications.

```

<urn:GetApplications>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:options>
    <urn:extraOption>
      <urn:name></urn:name>
      <urn:value></urn:value>
    </urn:extraOption>
  </urn:options>
</urn:GetApplications>

```

GetAvailableSubmitTransitions

Description

This service returns all submit transitions for the specified project.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The <i>Auth</i> type supplies credentials and optionally, a host name for licensing. The <i>userId</i> and <i>password</i> can be specified with HTTP BASIC or WS-SECURITY instead.
project (required)	ProjectIdentifier [page 100]	Specifies the project for which all available submit transitions are returned.
attributename (optional)	string	Returns only transitions that have this transition attribute (see TS_TRANSATTRS for selections). This argument is only used for transitions created via an integration.
options	Options [page 151]	Holds name value pairing for future arguments on certain calls.

Response

One or more transitions is returned, though the list can be empty. See the [Transition \[page 197\]](#) type for more detail.

Usage

The transition or transitions returned is limited by the project you specify.

Faults

- Invalid database pointer.
- The project ID is not valid.
- Reading transition attribute fails.

XML

The following XML is a snippet of the payload being sent with `GetAvailableSubmitTransitions`.

```
<urn:GetAvailableSubmitTransitions>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname>localhost</urn:hostname>
  </urn:auth>
  <urn:projectId>6</urn:projectId>
  <urn:attributeName></urn:attributeName>
</urn:GetAvailableSubmitTransitions>
```

GetAvailableTransitions

Description

This service returns a list of available transitions for the specified item. That list can be filtered to include only the transitions that have the specified attribute.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The <i>Auth</i> type supplies credentials and optionally, a host name for licensing. The <i>userId</i> and <i>password</i> can be specified with HTTP BASIC or WS-SECURITY instead.
item (required)	ItemIdentifier [page 95]	Specifies the item for which available transitions are returned.
transitionOptions	GetTransitionOptions [page 144]	Indicates the type of transitions to return. If empty or TRANSITIONS-ALL is specified, all transition types are returned. Specify TRANSITIONS-QUICK to return only quick transitions.
attributeName (optional)	string	Returns only transitions that have this transition attribute (see <code>TS_TRANSATTRS</code> for selections). This argument is only used for transitions created via an integration.
options	Options [page 151]	Holds name value pairing for future arguments on certain calls.

Response

One or more transitions are returned, though the list can be empty. See the [Transition \[page 197\]](#) type for more detail.

Usage

The transition or transitions returned are limited by the item you specify. Only transitions available to the user are returned.

Faults

- Invalid database pointer.
- The item ID is not valid.
- Reading transition attribute fails.

XML

The following XML is a snippet of the payload that is sent with `GetAvailableTransitions`.

```

<urn:GetAvailableTransitions>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:item>
    <urn:displayName></urn:displayName>
    <urn:id>25</urn:id>
    <urn:uuid></urn:uuid>
    <urn:tableId>1000</urn:tableId>
    <urn:tableIdItemId></urn:tableIdItemId>
    <urn:issueId></urn:issueId>
  </urn:item>
  <urn:transitionOptions>TRANSITIONS-ALL</urn:transitionOptions>
  <urn:attributeName></urn:attributeName>
</urn:GetAvailableTransitions>

```

GetFileAttachment

Description

This service gets an existing file attachment.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The <i>Auth</i> type supplies credentials and optionally, a host name for licensing. The <i>userId</i> and <i>password</i> can be specified with HTTP BASIC or WS-SECURITY instead.

Argument	Type	Description
item (required)	ItemIdentifier [page 95]	Specifies the item that contains the attachment.
attachmentID (required)	integer	This is the internal TS_ID of the attachment from the TS_ATTACHMENTS table.
options	Options [page 151]	Holds name value pairing for future arguments on certain calls.

Response

FileAttachmentContents is returned. The file attachment is returned as a base64 encoded attachment. See [FileAttachmentContents \[page 91\]](#) for more information.

Usage

You can use the ID returned in the FileAttachment parameter of a returned TTIItem in the attachmentID argument of GetFileAttachment.

Faults

- Invalid database pointer.
- The item ID is not valid.
- The user lacks sufficient permission.
- Creating the record fails.
- The attachment ID is not valid.
- Failed to read the attachment contents.

XML

The following XML is a snippet of the payload being sent with GetFileAttachment.

```
<urn:GetFileAttachment>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
  </urn:auth>
  <urn:item>
    <urn:displayName></urn:displayName>
    <urn:id>109</urn:id>
    <urn:uuid></urn:uuid>
    <urn:tableId>1000</urn:tableId>
    <urn:tableIdItemId>1000:109</urn:tableIdItemId>
    <urn:issueId>BUG000173</urn:issueId>
  </urn:item>
  <urn:attachmentID>104</urn:attachmentID>
</urn:GetFileAttachment>
```

GetItem

Description

This service returns a single item, given the table ID and internal item ID.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The <i>Auth</i> type supplies credentials and optionally, a host name for licensing. The <i>userId</i> and <i>password</i> can be specified with HTTP BASIC or WS-SECURITY instead.
itemId (required)	ItemIdentifier [page 95]	The item that should be returned.
options (optional)	ResponseItemOptions [page 154]	Enables you to limit the data that is returned in the response.

Response

TTItemHolder is returned for the item that is specified in the call. The *itemIdIdentifier*, which shows the unique TS_ID for the record and TS_ID of the table to which it belongs is displayed. For more detail, see [TTItemHolder \[page 198\]](#).

Usage

The GetItem call enables you to see a snapshot of an auxiliary or primary item, without invoking any actual changes against the specified item. To return results, you must provide the table ID and item ID for the item.

You can use the following elements in the options argument to control the service handling and response:

- **sections** and **specifiedSections** – Use these elements to specify which parts of an item should be returned in order to limit the data that is returned for a given item. The sections that are not specified are not included in the response. For example, if the items have a large number of item links, notes, URL and file attachments that don't need to be returned in the response, use these parameters to return only the sections of an item you want.
- **limitedField** – Use the limitedField element to specify specific fields to control the item data that is returned. For example, you can specify one or more fields to limit a service response to return only the fields that you want returned.

For more information on the options elements, see [ResponseItemOptions \[page 154\]](#).

Faults

- Invalid item id 0 for table 0.
- The item ID is not valid.
- The user lacks sufficient permission.

-
- Validation constraint violation: data type mismatch.

XML

The following XML is a snippet of the payload that is sent with GetItem.

```
<urn:GetItem>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
    <urn:extendedData>
      <urn:data>
        <urn:name></urn:name>
        <urn:value></urn:value>
      </urn:data>
    </urn:extendedData>
  </urn:auth>
  <urn:itemId>
    <urn:displayName>BUG000164</urn:displayName>
    <urn:id>106</urn:id>
    <urn:uuid></urn:uuid>
    <urn:tableId>1000</urn:tableId>
    <urn:tableIdItemId>1000:106</urn:tableIdItemId>
    <urn:issueId>000164</urn:issueId>
  </urn:itemId>
  <urn:options>
    <urn:extraOption>
      <urn:name></urn:name>
      <urn:value></urn:value>
    </urn:extraOption>
    <urn:sections>SECTIONS-SPECIFIED</urn:sections>
    <urn:specifiedSections>SECTION:FIXED</urn:specifiedSections>
    <urn:limitedField>
      <urn:displayName></urn:displayName>
      <urn:id></urn:id>
      <urn:uuid></urn:uuid>
      <urn:dbName></urn:dbName>
    </urn:limitedField>
  </urn:options>
</urn:GetItem>
```

GetItems

Description

This service returns one or more items, given the table ID and internal item ID for each item.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The <i>Auth</i> type supplies credentials and optionally, a host name for licensing. The <i>userId</i> and <i>password</i> can be specified with HTTP BASIC or WS-SECURITY instead.
itemId (required)	ItemIdentifier [page 95]	The item or items that should be returned.
options (optional)	MultipleResponseItemOptions [page 146]	Specifies whether the service should continue if an error is encountered or stop. Also enables you to limit the data that is returned in the response.

Response

TTItemHolder is returned, one for each item that is specified in the call. The *itemIdentifiers*, which show the unique TS_ID for each record and TS_ID of the table to which they belong is displayed. For more detail, see [TTItemHolder \[page 198\]](#).

Usage

The GetItems call enables you to see a snapshot of one or more auxiliary or primary items, without invoking any actual changes against the specified items. Specify the items that you want to return in one or more *itemId* arguments. To return results, you must provide the table ID and item ID for each item.

You can use the following elements in the *options* argument to control the service handling and response:

- **multiOption** – Use the *multiOption* element to specify whether the service should continue if an error is encountered, or stop and return an error. If any failures occur, each successive error message is appended to the string that is returned. Multiple error messages are separated by a single newline. If you specify CONTINUE-ON-FAILURE, then failures do not result in a return before all items have been processed.
- **sections** and **specifiedSections** – Use these elements to specify which parts of an item should be returned in order to limit the data that is returned for a given item. The sections that are not specified are not included in the response. For example, if the items have a large number of item links, notes, URL and file attachments that don't need to be returned in the response, use these parameters to return only the sections of an item you want.
- **limitedField** – Use the *limitedField* element to specify specific fields to control the item data that is returned. For example, you can specify one or more fields to limit a service response to return only the fields that you want returned.

For more information on the options elements, see [MultipleResponseItemOptions \[page 146\]](#).

Faults

-
- Invalid item id 0 for table 0.
 - The item ID is not valid.
 - The user lacks sufficient permission.
 - Validation constraint violation: data type mismatch.

XML

The following XML is a snippet of the payload that is sent with GetItems.

```
<urn:GetItems>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
    <urn:extendedData>
      <urn:data>
        <urn:name></urn:name>
        <urn:value></urn:value>
      </urn:data>
    </urn:extendedData>
  </urn:auth>
  <urn:itemId>
    <urn:displayName>BUG000164</urn:displayName>
    <urn:id>106</urn:id>
    <urn:uuid></urn:uuid>
    <urn:tableId>1000</urn:tableId>
    <urn:tableIdItemId>1000:106</urn:tableIdItemId>
    <urn:issueId>000164</urn:issueId>
  </urn:itemId>
  <urn:itemId>
    <urn:displayName>ENH000169</urn:displayName>
    <urn:id>107</urn:id>
    <urn:uuid></urn:uuid>
    <urn:tableId>1000</urn:tableId>
    <urn:tableIdItemId>1000:107</urn:tableIdItemId>
    <urn:issueId>000169</urn:issueId>
  </urn:itemId>
  <urn:itemId>
    <urn:displayName>BUG000173</urn:displayName>
    <urn:id>109</urn:id>
    <urn:uuid></urn:uuid>
    <urn:tableId>1000</urn:tableId>
    <urn:tableIdItemId>1000:109</urn:tableIdItemId>
    <urn:issueId>000173</urn:issueId>
  </urn:itemId>
  <urn:options>
    <urn:extraOption>
      <urn:name></urn:name>
      <urn:value></urn:value>
    </urn:extraOption>
  <urn:sections>SECTIONS-SPECIFIED</urn:sections>
  <urn:specifiedSections>SECTION:FIXED</urn:specifiedSections>
```

```

    <urn:limitedField>
      <urn:displayName></urn:displayName>
      <urn:id></urn:id>
      <urn:uuid></urn:uuid>
      <urn:dbName></urn:dbName>
    </urn:limitedField>
    <urn:multiOption>CONTINUE-ON-FAILURE</urn:multiOption>
  </urn:options>
</urn:GetItems>

```

GetItemsByQuery

Description

This service returns multiple items found using a *where* clause and an *order by* clause to determine the set of items returned.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The <i>Auth</i> type supplies credentials and optionally, a host name for licensing. The <i>userId</i> and <i>password</i> can be specified with HTTP BASIC or WS-SECURITY instead.
table (required)	TableIdentifier [page 109]	Specifies the table that contains the items that you want to return.
queryWhereClause (optional)	string	An SQL statement to find items with <i>TS_ID</i> >0. If not provided, all items in the table are returned.
orderByClause (optional)	string	An SQL statement to order the returned items. Enter a null or empty string for no ordering.
firstRecord	integer	Used in combination with <i>maxReturnSize</i> to return chunks of records.
maxReturnSize (optional)	integer	Enter the number of items to return.
options (optional)	MultipleResponseItemOptions [page 146]	Specifies whether the service should continue if an error is encountered or stop. Also enables you to limit the data that is returned in the response.

Response

TTItemList is returned, one for each item that is specified in the call. The response is a list of items in the specified table that match the query *where* clause. The list is ordered and limited as specified. For more detailed information, see [TTItemList \[page 201\]](#).

Usage

You use the `firstRecord` argument in combination with `maxReturnSize` to return chunks of records, rather than all the records in a query. If 1000 records match the query, you can specify `firstRecord` and `maxReturnSize` together to return only 100 items per page. For example, first you specify:

```
<urn:firstRecord>0</urn:firstRecord>
<urn:maxReturnSize>100</urn:maxReturnSize>
```

Then:

```
<urn:firstRecord>100</urn:firstRecord>
<urn:maxReturnSize>100</urn:maxReturnSize>
```

```
<urn:firstRecord>200</urn:firstRecord>
<urn:maxReturnSize>100</urn:maxReturnSize>
```

Up to the 900th record:

```
<urn:firstRecord>900</urn:firstRecord>
<urn:maxReturnSize>100</urn:maxReturnSize>
```

In the `maxReturnSize` element, you can also set the value to zero to use the system "Listing Report Items" limit. A number greater than zero will limit the return item list to the number you specify. If the number of items that would be returned exceeds the system "Listing Report Items" limit, no items will be returned and an error will be generated.



Note: You need only pass the conditions in the *where* clause itself. For example:

```
<urn:queryWhereClause>TS_ISSUEID LIKE '00106' OR TS_ISSUEID LIKE '00029'
OR TS_ISSUEID LIKE '00105'</urn:queryWhereClause>
```

```
<urn:queryWhereClause>TS_ISSUEID='00038'</urn:queryWhereClause>
```

```
<urn:queryWhereClause>TS_ID='4'</urn:queryWhereClause>
```

To return all items, send a null query:

```
<urn:queryWhereClause></urn:queryWhereClause>
```

You can use the following elements in the options argument to control the service handling and response:

- **multiOption** – Use the `multiOption` element to specify whether the service should continue if an error is encountered, or stop and return an error. If any failures occur,

each successive error message is appended to the string that is returned. Multiple error messages are separated by a single newline.

If you specify CONTINUE-ON-FAILURE, then failures do not result in a return before all items

have been processed.

- **sections** and **specifiedSections** – Use these elements to specify which parts of an item should be returned in order to limit the data that is returned for a given item. The sections that are not specified are not included in the response. For example, if the items have a large number of item links, notes, URL and file attachments that don't need to be returned in the response, use these parameters to return only the sections of an item you want.
- **limitedField** – Use the limitedField element to specify specific fields to control the item data that is returned. For example, you can specify one or more fields to limit a service response to return only the fields that you want to return.

For more information on the options elements, see [MultipleResponseItemOptions \[page 146\]](#).

Faults

- Invalid database pointer.
- The table ID is not valid.
- The user lacks sufficient permission.
- Creating the record fails.
- Query exceeds system "Listing Report Items" limit.
- Reading the item fails.

XML

The following XML is a snippet of the payload being sent with GetItemsByQuery.

```
<urn:GetItemsByQuery>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname>localhost</urn:hostname>
  </urn:auth>
  <urn:table>
    <urn:displayName></urn:displayName>
    <urn:id>1000</urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName></urn:dbName>
  </urn:table>
  <urn:queryWhereClause>TS_TITLE LIKE 'New Item'</urn:queryWhereClause>
  <urn:orderByClause>TS_TITLE</urn:orderByClause>
  <urn:firstRecord>0</urn:firstRecord>
  <urn:maxReturnSize>100</urn:maxReturnSize>
  <urn:options>
```

```

    <urn:extraOption>
      <urn:name></urn:name>
      <urn:value></urn:value>
    </urn:extraOption>
    <urn:sections>SECTIONS-SPECIFIED</urn:sections>
    <urn:specifiedSections>SECTION:FIXED</urn:specifiedSections>
    <urn:limitedField></urn:limitedField>
    <urn:multiOption></urn:multiOption>
  </urn:options>
</urn:GetItemsByQuery>

```

GetNoteLoggerInfo

Description

This service returns the mail box e-mail address that is configured for the E-mail Recorder feature.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The <i>Auth</i> type supplies credentials and optionally, a host name for licensing. The <i>userId</i> and <i>password</i> can be specified with HTTP BASIC or WS-SECURITY instead.
options	Options [page 151]	Holds name value pairing for future arguments.

Response

NoteLoggerInfo is returned in the response. The NoteLoggerInfo type contains the e-mail address that has been designated for the E-mail Recorder (located in the SBM System Administrator Mail Client properties). See [NoteLoggerInfo \[page 179\]](#) for more information.

Usage

This service is only used by the SBM Mobile application.

Faults

- There is no note logger mailbox configured.

XML

The following XML is a snippet of the payload being sent with GetNoteLoggerInfo.

```

<urn:GetNoteLoggerInfo>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname>localhost</urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
</urn:GetNoteLoggerInfo>

```

```
</urn:auth>
</urn:GetNoteLoggerInfo>
```

GetReports

Description

This service returns a list of reports within a specified range, limited by one or more optional filters.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The Auth type supplies credentials and optionally, a host name for licensing. The userId and password can be specified with HTTP BASIC or WS-SECURITY instead.
queryRange (optional)	QueryRange [page 100]	The QueryRange type allows you to specify the number of reports to return. This enables you to limit the number of reports that are returned.
reportsFilter (optional)	ReportsFilter [page 151]	The ReportsFilter type allows you to filter the reports that are returned based on optional parameters similar to the Find Reports command in the SBM User Workspace.
options	Options [page 151]	Holds name value pairing for future arguments.

Response

GetReportsResult is returned. A filtered list of reports, within the specified range, is displayed in the response. The response contains the number of reports returned and a high-level description for each report. For more detail, see [GetReportsResult \[page 175\]](#).

Usage

The GetReports call finds a list of available reports without actually running any of the reports themselves. You can use GetReports to search for all reports that you have privileges to run, modify, and delete within the provided range . You can also use this call to search for auxiliary table reports that you can run, modify, and delete. You can use this call to identify which report you would like to run using the UUID of the report. To run a given report in the returned list, use [RunReport \[page 73\]](#). You can optionally use the returned report URL to run the report in a Web browser.

Faults

- Invalid database pointer.
- The project ID is not valid.
- The user lacks sufficient permission.
- Returning the report fails.

XML

The following XML is a snippet of the payload that is sent with GetReports.

```
<urn:GetReports>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:queryRange>
    <urn:startIndex>1</urn:startIndex>
    <urn:fetchSize>4</urn:fetchSize>
    <urn:totalCount></urn:totalCount>
  </urn:queryRange>
  <urn:reportsFilter>
    <urn:solution>
      <urn:displayName></urn:displayName>
      <urn:id>1</urn:id>
      <urn:uuid></urn:uuid>
      <urn:uniqueName></urn:uniqueName>
      <urn:tabName></urn:tabName>
    </urn:solution>
    <urn:project>
      <urn:displayName></urn:displayName>
      <urn:id>4</urn:id>
      <urn:uuid></urn:uuid>
      <urn:fullyQualifiedNames></urn:fullyQualifiedNames>
    </urn:project>
    <urn:table>
      <urn:displayName></urn:displayName>
      <urn:id>1000</urn:id>
      <urn:uuid></urn:uuid>
      <urn:dbName></urn:dbName>
    </urn:table>
    <urn:author>
      <urn:displayName></urn:displayName>
      <urn:id></urn:id>
      <urn:uuid></urn:uuid>
      <urn:loginId></urn:loginId>
    </urn:author>
    <urn:reportType>LISTING</urn:reportType>
    <urn:reportCategory>ALL</urn:reportCategory>
    <urn:reportAccessLevel>USER</urn:reportAccessLevel>
    <urn:report>
      <urn:displayName></urn:displayName>
      <urn:id></urn:id>
      <urn:uuid></urn:uuid>
    </urn:report>
    <urn:searchByName></urn:searchByName>
    <urn:includeSubProjects>true</urn:includeSubProjects>
    <urn:createdDateFrom></urn:createdDateFrom>
    <urn:createdDateTo></urn:createdDateTo>
    <urn:extendedData>
```

```

        <urn:data>
            <urn:name></urn:name>
            <urn:value></urn:value>
        </urn:data>
    </urn:extendedData>
</urn:reportsFilter>
<urn:options>
    <urn:extraOption>
        <urn:name></urn:name>
        <urn:value></urn:value>
    </urn:extraOption>
</urn:options>
</urn:GetReports>

```

GetSolutions

Description

This service returns a list of solutions that can be accessed by the user.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The <i>Auth</i> type supplies credentials and optionally, a host name for licensing. The <i>userId</i> and <i>password</i> can be specified with HTTP BASIC or WS-SECURITY instead.
options	Options [page 151]	Holds name value pairing for future arguments.

Response

SolutionData (a list of solutions) is returned in the response. Each return element contains a *SolutionIdentifier*, and additional information including the *UUID*. See [SolutionData](#) [page 191] for more information.

Usage

This call is a good starting point for using the SBM Web services. After calling *GetSolutions*, you can call [GetTables](#) [page 65] to get available tables. You can then use the table identifiers to get items with [GetItemsByQuery](#) [page 56], or create auxiliary items with [CreateAuxItems](#) [page 27]. To create primary items, use [GetSubmitProjects](#) [page 64] followed by [CreatePrimaryItems](#) [page 37].



Note: Primary items require the *ProjectIdentifier*, while auxiliary items require the *TableIdentifier*.

Faults

- Invalid database pointer.
- No solutions are available for the user.

XML

The following XML is a snippet of the payload that is sent with GetSolutions.

```
<urn:GetSolutions>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
    <urn:extendedData>
      <urn:data>
        <urn:name></urn:name>
        <urn:value></urn:value>
      </urn:data>
    </urn:extendedData>
  </urn:auth>
  <urn:options>
    <urn:extraOption>
      <urn:name></urn:name>
      <urn:value></urn:value>
    </urn:extraOption>
  </urn:options>
</urn:GetSolutions>
```

GetStateChangeHistory

Description

This service returns a specified range of state change history for an item.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The <i>Auth</i> type supplies credentials and optionally, a host name for licensing. The <i>userId</i> and <i>password</i> can be specified with HTTP BASIC or WS-SECURITY instead.
item (required)	ItemIdentifier [page 95]	Specifies the item for which state change history is returned.
queryRange (optional)	QueryRange [page 100]	The <i>QueryRange</i> type allows you to specify the number of change records to return. It can be used to limit the number of state changes that are returned.
options	Options [page 151]	Holds name value pairing for future arguments.

Response

GetStateChangeHistoryResult is returned. A list containing each state change record, within the specified range, is displayed in the response. The response contains detailed

information about each state change, including the state, the user who performed the transition, and the owner. For more detail, see [GetStateChangeHistoryResult \[page 177\]](#).

Usage

You use `GetStateChangeHistory` to return either the entire state change history for an item or only a portion of the history. The information that is returned is the same information that is stored in the `TS_CHANGEACTIONS` table in the database. Use the `GetStateChangeHistory` call if you want to retrieve the details of an item's lifecycle for audit purposes.

Faults

- Invalid User ID or Password.
- Invalid item id 0 for table 0.

XML

The following XML is a snippet of the payload that is sent with `GetStateChangeHistory` to only return history through the first two states.

```
<urn:GetStateChangeHistory>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:item>
    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:tableId></urn:tableId>
    <urn:tableIdItemId>1000:2</urn:tableIdItemId>
    <urn:issueId></urn:issueId>
  </urn:item>
  <urn:queryRange>
    <urn:startIndex></urn:startIndex>
    <urn:fetchSize>2</urn:fetchSize>
    <urn:totalCount></urn:totalCount>
  </urn:queryRange>
</urn:GetStateChangeHistory>
```

GetSubmitProjects

Description

This service returns a list of projects into which the user can submit. If a table ID is provided, only projects from that table are listed.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The <i>Auth</i> type supplies credentials and optionally, a host name for licensing. The <i>userId</i> and <i>password</i> can be specified with HTTP BASIC or WS-SECURITY instead.
table (optional)	TableIdentifier [page 109]	Specify a table to limit the projects that are returned.

Response

ProjectData (a list of projects, if multiple are available) is returned in the response. See [ProjectData \[page 180\]](#) for more information.

Usage

The project or projects that are returned are limited by the table you specify. Only projects available to the user are returned. Use [CreatePrimaryItems \[page 37\]](#) to submit after the desired project is found.

Faults

- Invalid database pointer.
- The table ID is not valid.
- Error reading transitions table.

XML

The following XML is a snippet of the payload that is sent with `GetSubmitProjects`.

```
<urn:GetSubmitProjects>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:table>
    <urn:displayName>Issues</urn:displayName>
    <urn:id>1000</urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName>UBG_ISSUES</urn:dbName>
  </urn:table>
</urn:GetSubmitProjects>
```

GetTables

Description

This service returns a list of available tables that a user can access, optionally filtered by solution and table type.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The <i>Auth</i> type supplies credentials and optionally, a host name for licensing. The <i>userId</i> and <i>password</i> can be specified with HTTP BASIC or WS-SECURITY instead.
solution (optional)	SolutionIdentifier [page 106]	Specify a solution to only return the tables in that solution.
tableType (optional)	Table-Type [page 110]	The type of table to return. If a table type is specified, only available tables of that type are listed.
options	Options [page 151]	Holds name value pairing for future arguments.

Response

TableData is returned. The list of available tables and their fields are returned, optionally filtered by either solution or table type. For more information, see [TableData \[page 195\]](#).

Usage

If no solution ID is supplied, available tables are listed for all solutions. Use [GetSolutions \[page 62\]](#) to return available solution IDs.

Faults

- Invalid database pointer.
- Reading the tables fails.
- GetTables not implemented for table type.

XML

The following XML is a snippet of the payload that is sent with GetTables.

```
<urn:GetTables>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname>localhost</urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
    <urn:extendedData>
      <urn:data>
        <urn:name></urn:name>
        <urn:value></urn:value>
      </urn:data>
    </urn:extendedData>
  </urn:auth>
  <urn:solution>
    <urn:displayName>Issue Defect Management</urn:displayName>
    <urn:id>1</urn:id>
```

```

    <urn:uuid></urn:uuid>
    <urn:uniqueName></urn:uniqueName>
    <urn:tabName></urn:tabName>
  </urn:solution>
  <urn:tableType>PRIMARY-TABLE</urn:tableType>
  <urn:options>
    <urn:extraOption>
      <urn:name></urn:name>
      <urn:value></urn:value>
    </urn:extraOption>
  </urn:options>
</urn:GetTables>

```

GetUsers

Description

This services returns identification information about a user and the user's preferences.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The Auth type supplies credentials and optionally, a host name for licensing. The userId and password can be specified with HTTP BASIC or WS-SECURITY instead.
getCurrentUser	boolean	Specify true to return user info for the user specified in the auth argument of the GetUsers call. Specify false to use the other arguments to return users instead. Default value is false.
searchByName	string	Used to specify the name of the user as a search parameter.
user	UserIdentifier [page 138]	Used to return a specific user or list of users. Specify multiple UserIdentifiers to return multiple users.
options	UserResponseOptions [page 162]	Specifies whether the service should continue if an error is encountered or stop. Also enables you to limit the data that is returned in the response.

Response

UserHolder is returned. The UserHolder response contains one or more UserIdentifiers and additional information from the users preferences. For more specific information, see [UserHolder \[page 202\]](#).

Usage

The GetUsers call is useful when you need data about a user account. The GetUsers call retrieves data for a user account as it exists in the TS_USERS table of the database. To retrieve privileges for a given user account, use the admin Web service call [GetUserPrivileges \[page 229\]](#). To determine if a specific user has a given privilege, use the admin Web service call [HasUserPrivilege \[page 232\]](#).

Use the following arguments to determine which users are returned in the response:

- **getCurrentUser** — Enables you to specify whether the service should only return user information for the user that is making the call. If you set getCurrentUser to true, the current user is returned, regardless of the data sent in the searchByName and user arguments.
- **searchByName** — Enables you to enter a single string to return user accounts. For example, if you enter Joe as the searchByName value, user records with following login IDs are returned: Joe, Joel, Joey. You can not enter multiple strings. The searchByName argument is ignored if getCurrentUser is set to true or if a UserIdentifier is specified in the user argument.



Note: To return all user records in the database, leave getCurrentUser set to false, do not provide a value for searchByName, and do not enter specific UserIdentifiers in the user argument. If you have a large number of users in your database, consider limiting the return data using one or more arguments.

- **user** — Enables you to specify one or more specific user accounts to be returned. If you want to return multiple users, you must specify each desired user by providing the UserIdentifier in a list of multiple user arguments. The user argument takes precedence over the searchByName argument.

Use the following elements in the options parameter to control how this call is processed and what data is returned:

- **multiOption** — Enables you to specify whether the service should continue if an error is encountered, or stop and throw an error. If any failures occur, each successive error message is appended to the string that is returned. Multiple error messages are separated by a single newline. Failures do not result in a return before all users have been processed.
- **sections** and **specifiedSections** — Enables you to specify which parts of a user record should be returned. This allows you to limit the data that is returned for a given user. The sections that aren't specified are not included in the response. For example, if you only need basic user information in the response, use the sections parameter to return only the STANDARD section.

For more information on the options elements, see [UserResponseOptions \[page 162\]](#).

Faults

- Invalid database pointer.
- The user ID is not valid.
- The user lacks sufficient permission.

XML

The following XML is a snippet of the payload this is sent with `GetUsers`. In this example, `searchByName` is used to find all users with the string `Joe` in the user login ID.

```
<urn:GetUsers>
  <urn:auth>
    <urn:userId>Admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:getCurrentUser></urn:getCurrentUser>
  <urn:searchByName>Joe</urn:searchByName>
  <urn:user>
    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:loginId></urn:loginId>
  </urn:user>
</urn:GetUsers>
```

In this example, `getCurrentUser` is empty (or false), `searchByName` is left empty, and no user is provided in the user argument. This call returns all the users in the system.

```
<urn:GetUsers>
  <urn:auth>
    <urn:userId>Admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:getCurrentUser></urn:getCurrentUser>
  <urn:searchByName></urn:searchByName>
  <urn:user>
    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:loginId></urn:loginId>
  </urn:user>
</urn:GetUsers>
```

GetVersion

Description

This service returns the SBM version number.

Arguments

None.

Response

A string is returned, showing the version number. For example:

```
<ae:GetVersionResponse>
  <ae:return>Version 2010 R1.200</ae:return>
</ae:GetVersionResponse>
```

Usage

None.

Faults

None.

XML

The following XML shows the payload that is sent with GetVersion.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:urn="urn:sbmappservices72">
  <soap:Header/>
  <soap:Body>
    <urn:GetVersion/>
  </soap:Body>
</soap:Envelope>
```

IsUserValid**Description**

This service determines whether a specified user is valid.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 270]	The Auth type supplies credentials and optionally, a host name for licensing. The userId and password can be specified with HTTP BASIC or WS-SECURITY instead.
user (optional)	UserIdentifier [page 138]	The user to be checked.

Response

A boolean is returned, showing whether the user is valid (true) or not (false). The user account is checked in the context of either the calling user or the specified user. For example:

```
<ae:IsUserValidResponse>
  <ae:return>>true</ae:return>
</ae:IsUserValidResponse>
```

Usage

IsUserValid will return true if the specified user was found in the database and is not deleted or disabled. Otherwise false is returned.

Faults

- Invalid database pointer.
- The login ID is not valid.
- The user lacks sufficient permission.

XML

The following XML is a snippet of the payload this is sent with `IsValidUser`.

```
<urn:IsUserValid>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
  </urn:auth>
  <urn:user>
    <urn:displayName>carmen</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:loginId></urn:loginId>
  </urn:user>
</urn:IsUserValid>
```

LinkSubtask

Description

This service links one item to another as a subtask.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The <i>Auth</i> type supplies credentials and optionally, a host name for licensing. The <code>userId</code> and <code>password</code> can be specified with HTTP BASIC or WS-SECURITY instead.
parentItem (optional)	ItemIdentifier [page 95]	The parent item of the subtask.
childItem (optional)	ItemIdentifier [page 95]	The child item that will become the subtask.
options	Options [page 151]	Holds name value pairing for future arguments.

Response

An empty response is returned: `<ae:LinkSubtaskResponse/>`.

Usage

Use this call to create a subtask relationship between primary items.

Faults

- Invalid database pointer.
- Reading the tables fails.
- GetTables not implemented for table type.

XML

The following XML is a snippet of the payload that is sent with LinkSubtask.

```
<urn:LinkSubtask>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:parentItem>
    <urn:displayName></urn:displayName>
    <urn:id>109</urn:id>
    <urn:uuid></urn:uuid>
    <urn:tableId></urn:tableId>
    <urn:tableIdItemId>1000:109</urn:tableIdItemId>
    <urn:issueId></urn:issueId>
  </urn:parentItem>
  <urn:childItem>
    <urn:displayName></urn:displayName>
    <urn:id>79</urn:id>
    <urn:uuid></urn:uuid>
    <urn:tableId></urn:tableId>
    <urn:tableIdItemId>1000:79</urn:tableIdItemId>
    <urn:issueId></urn:issueId>
  </urn:childItem>
</urn:LinkSubtask>
```

Logout

Description

This service releases any licenses and resources associated with the session.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The Auth type supplies credentials and optionally, a host name for licensing. The userId and password can be specified with HTTP BASIC or WS-SECURITY instead.
options	Options [page 151]	Holds name value pairing for future arguments.

Response

An empty XML response is returned:

```
<ae:LogoutResponse/>
```

and the session is ended. Failure will keep the session open.

Usage

The Logout call logs out the user from the current active session. There is no effect if previous Web service calls are not made before Logout is called.

Faults

- Authentication error if invalid credentials.

XML

The following XML is a snippet of the payload that is sent with Logout.

```
<urn:Logout>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
  </urn:auth>
</urn:Logout>
```

RunReport

Description

This service runs a specified report, given the proper privileges.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The Auth type supplies credentials and optionally, a host name for licensing. The userId and password can be specified with HTTP BASIC or WS-SECURITY instead.
queryRange (optional)	QueryRange [page 100]	Enables you to specify the number of reports to return. Used to limit the number of reports that are returned.
report (required)	ReportIdentifier [page 104]	Specifies the report to run.
solution (optional)	SolutionIdentifier [page 106]	Specifies the solution that the report is based on.
project (optional)	ProjectIdentifier [page 100]	Specifies the project that the report was created against.

Argument	Type	Description
table (optional)	TableIdentifier [page 109]	Specifies the table that the report is based on.
reportCategory (optional)	ReportCategory [page 102]	A broader enumeration that limits the response based on the category of report (built-in reports, application reports, reports you authored).
reportAccessLevel (optional)	ReportAccessLevel [page 101]	An enumeration that limits the response based on the report's access level (PRIVATE, GUEST, USER, or MANAGER).
options	Options [page 151]	Holds name value pairing for future arguments.

Response

RunReportResult is returned. A high-level description of the report is returned, along with a description of each column in the report. The fields used to order the results are shown as well. In the result parameter, the actual field values can be found. For more detail, see [RunReportResult](#) [page 185].

Usage

The RunReport call executes a given report, assuming you have the proper privilege. The GetReports call finds a list of available reports without actually running any of the reports themselves. In order to run one of those returned reports, use RunReport. If no fetchSize is specified in the QueryRange, up to 1000 items can be returned.

In order to run a Built-in report, you must provide either solutionID or solutionName. Since Built-in reports aren't tied to a specific application, you must provide the RunReport call with the solution ID or name (which can be found in the GetSolutionsWithUniqueName call). User-created reports (those stored in the TS_REPORTS table) do not require solutionID or solutionName. To run a user-created report, you simply need to provide the report ID (which can be obtained from the GetReports call).

Faults

- Invalid database pointer.
- The user lacks sufficient permission.
- Executing the report fails.
- Could not run built-in report because it needs correct solutionName (or ID) parameter.
- Two or more reports exist with the name *<reportName>*. Please provide a solution ID, solution name, or use additional parameters to identify which report to run.

XML

The following XML is a snippet of the payload that is sent with RunReport.


```
<urn:RunReport>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:queryRange>
    <urn:startIndex></urn:startIndex>
    <urn:fetchSize></urn:fetchSize>
    <urn:totalCount></urn:totalCount>
  </urn:queryRange>
  <urn:report>
    <urn:displayName>Test Listing Report</urn:displayName>
    <urn:id>118</urn:id>
    <urn:uuid>45e5b0ce-fb41-4a67-be7a-5941cdd04b90</urn:uuid>
  </urn:report>
  <urn:solution>
    <urn:displayName>Issue Defect Management</urn:displayName>
    <urn:id>2</urn:id>
    <urn:uuid>cea0a86c-5d74-4e12-b8d6-9d6b90186f1e</urn:uuid>
    <urn:uniqueName>ISSUE_DEFECT_MANAGEMENT</urn:uniqueName>
    <urn:tabName>IDM</urn:tabName>
  </urn:solution>
  <urn:project>
    <urn:displayName>Animation Pro</urn:displayName>
    <urn:id>6</urn:id>
    <urn:uuid>2ac5ef27-71da-4b07-ab7e-dddbc9c2d8c7</urn:uuid>
    <urn:fullyQualifiedName>Base Project||Base IDT Project||
    Software Development||Animation Pro</urn:fullyQualifiedName>
  </urn:project>
  <urn:table>
    <urn:displayName>Issues</urn:displayName>
    <urn:id>1000</urn:id>
    <urn:uuid>dc8cd329-b430-436f-bb75-bf90008e6a50</urn:uuid>
    <urn:dbName>UBG_ISSUES</urn:dbName>
  </urn:table>
  <urn:reportCategory>USERREPORTS</urn:reportCategory>
  <urn:reportAccessLevel>GUEST</urn:reportAccessLevel>
</urn:RunReport>
```

UpdateFileAttachment

Description

This service updates an existing file attachment for a specified item.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The Auth type supplies credentials and optionally, a host name for licensing. The userId and password can be specified with HTTP BASIC or WS-SECURITY instead.
item (required)	ItemIdentifier [page 95]	The item that contains the file attachment.
attachmentContents (required)	FileAttachmentContents [page 91]	<p>The file attachment details and content, which are all optional except for the attachment ID.</p> <p> Note: attachmentContents is of type FileAttachmentContents, but also includes attachment detail found in FileAttachment. The XML example below shows the parameters from both FileAttachment and FileAttachmentContents. See FileAttachment [page 90] for more information.</p>
options	Options [page 151]	Holds name value pairing for future arguments.

Response

FileAttachment is returned. The newly updated file attachment details are returned (not the content itself). For more detail, see [FileAttachment \[page 90\]](#)

Usage

The UpdateFileAttachment call provides a method to update a single attachment on an auxiliary or primary item, given the proper privileges. If the call fails, the file attachment will not be updated. To update multiple file attachments for a single item, UpdateFileAttachment must be called for each attachment. Any data that is provided is updated as appropriate. Attachment ID and modification time cannot be set.

Faults

- Invalid database pointer.
- The item ID is not valid.
- The user lacks sufficient permission.

XML

The following XML is a snippet of the payload that is sent with UpdateFileAttachment.

```
<urn:UpdateFileAttachment>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname>localhost</urn:hostname>
  </urn:auth>
  <urn:item>
    <urn:displayName></urn:displayName>
    <urn:id>109</urn:id>
    <urn:uuid></urn:uuid>
    <urn:tableId>1000</urn:tableId>
    <urn:tableIdItemId>1000:109</urn:tableIdItemId>
    <urn:issueId>BUG000173</urn:issueId>
  </urn:item>
  <urn:attachmentContents>
    <urn:id>51</urn:id>
    <urn:name>file attach updated MER00023!</urn:name>
    <urn:fileName>wslog.txt</urn:fileName>
    <urn:showAsImage>false</urn:showAsImage>
    <urn:modificationDateTime></urn:modificationDateTime>
    <urn:url></urn:url>
    <urn:accessType>ATTACHACCESS-RESTRICTED</urn:accessType>
    <urn:contentsBase64>
      <urn:data>cid:1155248127636</urn:data>
    </urn:contentsBase64>
  </urn:attachmentContents>
</urn:UpdateFileAttachment>
```

TransitionItem

Description

This service transitions an item.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The <i>Auth</i> type supplies credentials and optionally, a host name for licensing. The <i>userId</i> and <i>password</i> can be specified with HTTP BASIC or WS-SECURITY instead.
item (required)	TTItem [page 114]	The items to be transitioned. The <i>TTItem</i> type holds the generic data for the item. You must have the <i>itemID</i> filled in at a minimum.
transition (optional)	TransitionIdentifier [page 111]	The transition that you intend to invoke against the item.
breakLock	boolean	Specify True to break any existing item lock. Specify False to honor the item lock.

Argument	Type	Description
options (optional)	ResponseItemOptions [page 154]	Enables you to limit the data that is returned in the response.

Response

TTItemHolder is returned for the transitioned item. The item is returned with updated item data, which shows the unique TS_ID the each record and the TS_ID of the table. For more detail, see [TTItemHolder \[page 198\]](#).

Usage

The TransitionItem call provides a method to transition or update a single record in a primary table or auxiliary table. If a transition of 0 is specified, the default update transition is used. You can update records in both custom and system primary tables, given the proper privileges.

If you specify a specific transition for the call to use, that transition needs to be a valid transition for the items' project.

To create notes, item links, and URL attachments on the item, add records to the lists that are defined in TTItem. To update a file attachment, see [UpdateFileAttachment \[page 75\]](#).

Use the following elements in the options parameter to control how this call is processed and what data is returned:

- **sections** and **specifiedSections** — Enables you to specify which parts of an item should be returned. This allows you to limit the data that is returned for a given item. The sections that aren't specified are not included in the response. For example, if the items have a large number of item links, notes, URL and file attachments that don't need to be returned in the response, use the sections parameter to return only the sections of an item you want.
- **limitedField** — Enables you to specify which fields you want returned in the response. For example, you can specify one or more fields to limit a service response to return only the fields that you want returned.

For more information on the options elements, see [ResponseItemOptions \[page 154\]](#).

Faults

- Invalid database pointer.
- Invalid item.
- The user lacks sufficient permission.
- The transaction fails.
- Reading the item fails.
- Invalid project.

XML

The following XML is a snippet of the payload that is sent with TransitionItem.

```

<urn:TransitionItem>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:item>
    <urn:id>
      <urn:displayName>000231</urn:displayName>
      <urn:id>197</urn:id>
      <urn:uuid></urn:uuid>
      <urn:tableId>1000</urn:tableId>
      <urn:tableIdItemId>1000:197</urn:tableIdItemId>
      <urn:issueId></urn:issueId>
    </urn:id>
    <urn:itemType></urn:itemType>
    <urn:project>
      <urn:displayName>Animation Pro</urn:displayName>
      <urn:id>6</urn:id>
      <urn:uuid></urn:uuid>
      <urn:fullyQualifiedName>Base Project||Base IDT Project||
        Software Development||Animation Pro</urn:fullyQualifiedName>
    </urn:project>
    <urn:title></urn:title>
    <urn:description>This is a test item.</urn:description>
    <urn:extendedField>
      <urn:id>
        <urn:displayName>Developer</urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
        <urn:dbName>DEVELOPER</urn:dbName>
      </urn:id>
      <urn:setValueBy>DISPLAY-VALUE</urn:setValueBy>
      <urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
      <urn:value>
        <urn:displayValue>admin</urn:displayValue>
        <urn:internalValue></urn:internalValue>
        <urn:uuid></urn:uuid>
      </urn:value>
    </urn:extendedField>
  </urn:item>
  <urn:transition>
    <urn:displayName>Fix</urn:displayName>
    <urn:id>8</urn:id>
    <urn:uuid></urn:uuid>
  </urn:transition>
  <urn:breakLock>true</urn:breakLock>
  <urn:options>
    <urn:sections>SECTIONS-SPECIFIED</urn:sections>
    <urn:specifiedSections>SECTION:FIXED</urn:specifiedSections>
    <urn:limitedField></urn:limitedField>
  </urn:options>
</urn:TransitionItems>

```

TransitionItems

Description

This service transitions one or more items.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 141]	The <i>Auth</i> type supplies credentials and optionally, a host name for licensing. The <i>userId</i> and <i>password</i> can be specified with HTTP BASIC or WS-SECURITY instead.
item (required)	TTItem [page 114]	The list of items to be transitioned. The <i>TTItem</i> types hold the generic data for each item. You must have the <i>itemID</i> filled in at a minimum for each item in the list.
transition (optional)	TransitionIdentifier [page 111]	The transition that you intend to invoke against one or more items.
breakLock	boolean	Specify True to break any existing item lock. Specify False to honor the item lock.
options (optional)	MultipleResponseItemOptions [page 146]	Specifies whether the service should continue if an error is encountered or stop. Also enables you to limit the data that is returned in the response.

Response

TTItemHolder is returned, one for each transitioned item. The primary items are returned with updated item data, which shows the unique *TS_IDs* of each record and *TS_IDs* of the table. For more detail, see [TTItemHolder \[page 198\]](#).

Usage

The *TransitionItems* call provides a method to transition or update multiple records in a primary table or auxiliary table. If a transition of 0 is specified, the default update transition is used. If any failures occur, each successive error message is appended to the string that is returned. Multiple error messages are separated by a single newline. Failures do not result in a return before all items have been processed. You can update records in both custom and system primary tables, given the proper privileges.

If you specify a specific transition for the call to use, that transition needs to be a valid transition for the items' project.

To create notes, item links, and URL attachments on the items, add records to the lists that are defined in each *TTItem*. To update a file attachment, see [UpdateFileAttachment \[page 75\]](#).

Use the following elements in the options parameter to control how this call is processed and what data is returned:

- **multiOption** — Enables you to specify whether the service should continue if an error is encountered, or stop and throw an error. If any failures occur, each successive error message is appended to the string that is returned. Multiple error messages are separated by a single newline. Failures do not result in a return before all items have been processed.
- **sections** and **specifiedSections** — Enables you to specify which parts of an item should be returned. This allows you to limit the data that is returned for a given item. The sections that aren't specified are not included in the response. For example, if the items have a large number of item links, notes, URL and file attachments that don't need to be returned in the response, use the sections parameter to return only the sections of an item you want.
- **limitedField** — Enables you to specify which fields you want returned in the response. For example, you can specify one or more fields to limit a service response to return only the fields that you want returned.

For more information on the options elements, see [MultipleResponseItemOptions \[page 146\]](#).

Faults

- Invalid database pointer.
- Invalid item.
- The user lacks sufficient permission.
- The transaction fails.
- Reading the item fails.
- Invalid project.

XML

The following XML is a snippet of the payload that is sent with TransitionItems.

```
<urn:TransitionItems>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:item>
    <urn:id>
      <urn:displayName>000231</urn:displayName>
      <urn:id>197</urn:id>
      <urn:uuid></urn:uuid>
      <urn:tableId>1000</urn:tableId>
      <urn:tableIdItemId>1000:197</urn:tableIdItemId>
      <urn:issueId></urn:issueId>
    </urn:id>
  </urn:item>
</urn:TransitionItems>
```

```
<urn:itemType></urn:itemType>
<urn:project>
  <urn:displayName>Animation Pro</urn:displayName>
  <urn:id>6</urn:id>
  <urn:uuid></urn:uuid>
  <urn:fullyQualifiedName></urn:fullyQualifiedName>
</urn:project>
<urn:title></urn:title>
<urn:description>This is a test item.</urn:description>
<urn:extendedField>
  <urn:id>
    <urn:displayName>Developer</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName>DEVELOPER</urn:dbName>
  </urn:id>
  <urn:setValueBy>DISPLAY-VALUE</urn:setValueBy>
  <urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
  <urn:value>
    <urn:displayValue>admin</urn:displayValue>
    <urn:internalValue></urn:internalValue>
    <urn:uuid></urn:uuid>
  </urn:value>
</urn:extendedField>
</urn:item>
<urn:item>
  <urn:id>
    <urn:displayName>000232</urn:displayName>
    <urn:id>198</urn:id>
    <urn:uuid></urn:uuid>
    <urn:tableId>1000</urn:tableId>
    <urn:tableIdItemId>1000:198</urn:tableIdItemId>
    <urn:issueId></urn:issueId>
  </urn:id>
  <urn:itemType></urn:itemType>
  <urn:project>
    <urn:displayName>Animation Pro</urn:displayName>
    <urn:id>6</urn:id>
    <urn:uuid></urn:uuid>
    <urn:fullyQualifiedName></urn:fullyQualifiedName>
  </urn:project>
  <urn:title></urn:title>
  <urn:description>Another test.</urn:description>
  <urn:subtasks/>
  <urn:extendedField>
    <urn:id>
      <urn:displayName>Priority</urn:displayName>
      <urn:id></urn:id>
      <urn:uuid></urn:uuid>
      <urn:dbName>PRIORITY</urn:dbName>
    </urn:id>
    <urn:setValueBy>DISPLAY-VALUE</urn:setValueBy>
    <urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
    <urn:value>
      <urn:displayValue>1</urn:displayValue>
```

```
        <urn:internalValue></urn:internalValue>
        <urn:uuid></urn:uuid>
    </urn:value>
</urn:extendedField>
</urn:item>
<urn:transition>
    <urn:displayName>Fix</urn:displayName>
    <urn:id>8</urn:id>
    <urn:uuid></urn:uuid>
</urn:transition>
<urn:breakLock>true</urn:breakLock>
<urn:options>
    <urn:sections>SECTIONS-SPECIFIED</urn:sections>
    <urn:specifiedSections>SECTION:FIXED</urn:specifiedSections>
    <urn:limitedField></urn:limitedField>
    <urn:multiOption>CONTINUE-ON-FAILURE</urn:multiOption>
</urn:options>
</urn:TransitionItems>
```

Common Types

This section provides detailed descriptions of SBM Application Web service types that appear as both arguments and response elements. The types listed here contain one or more parameters, which make up the data being sent to or returned by the Web service. The parameters listed in each types are either simple or complex types themselves. If the type is complex, a link to further detail of that type will be provided in the **Type** column.

The following table lists all supported arguments in alphabetical order, followed by a brief description of each type. Select an argument to view detailed information including:

- **Description**
A brief description of the type.
- **Parameters**
A table describing the parameters for each type. Both simple and complex types are listed for each type. For each complex type, you can click the type name for a detailed description.
- **Usage**
Any notes, additional details, and concerns regarding the type are addressed here.
- **XML**
An example of the actual XML being sent is displayed here. The XML not only shows the type and its respective elements, you can also see detailed examples of each element and how to format the expected data.

Common Types

Type	Description
ApplicationIdentifier [page 86]	Holds the complete identification information for an application.
Attachment-Access-Type [page 87]	Indicates the type of attachment on an item.
ContactIdentifier [page 88]	Holds the complete identification information for a contact.
FieldIdentifier [page 89]	Holds the complete identification information for a field.
FieldValue [page 89]	Holds generic identification information for a field value.
FileAttachment [page 90]	Holds the details of a file that you upload to the server.
FileAttachmentContents [page 91]	Holds the actual contents of a file attachment.
FileBufferBase64 [page 92]	Holds the binary contents of a file in base64.

Type	Description
FileContents [page 92]	Holds the actual contents of a file.
GroupIdentifier [page 93]	Holds the complete identification information for a group.
Identifier [page 94]	Holds generic identification information.
ItemIdentifier [page 95]	Holds the complete identification information for an item.
ItemLink [page 96]	Holds information about an item link.
ItemLink-Type [page 97]	Indicates the type of item link.
Note [page 98]	Holds information about a note.
ProjectIdentifier [page 100]	Holds the complete identification information for a project.
QueryRange [page 100]	Limits the number of records returned in a query.
ReportAccessLevel [page 101]	Indicates the access level that is assigned to a report.
ReportCategory [page 102]	A filter that describes a grouping of reports.
ReportIdentifier [page 104]	Holds the complete identification information for a report.
ReportType [page 104]	Holds information about a type of report.
SolutionIdentifier [page 106]	Holds the complete identification information for a solution.

Type	Description
StateIdentifier [page 107]	Holds the complete identification information for a state.
Subtasks [page 108]	Holds information about a subtask relationship.
TableIdentifier [page 109]	Holds the complete identification information for a table.
Table-Type [page 110]	Indicates the type of table.
TransitionIdentifier [page 111]	Holds the complete identification information for a transition.
Transition-Type [page 111]	Indicates the type of transition.
TTItem [page 114]	Holds all of the SBM field information for an item.
URLAttachment [page 137]	Holds information about a URL attached to an auxiliary or primary item.
UserIdentifier [page 138]	Holds the complete identification information for a user.
WorkflowIdentifier [page 139]	Holds the complete identification information for a workflow.

ApplicationIdentifier

Description

The ApplicationIdentifier type holds the identification information for an application. The ApplicationIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 94] .	Generic identification extension base. Holds the displayName, id, and uuid for an application.

Usage

The ApplicationIdentifier is the identifier that can be used in Web service methods to uniquely identify an application. The ApplicationIdentifier contains the generic information about an application (including the display name, ID, and UUID).



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML snippet shows ApplicationIdentifier in the return element of the GetApplications response.

```
<ae:return>
  <ae:application xsi:type="ae:ApplicationIdentifier">
    <ae:displayName>Incident Management</ae:displayName>
    <ae:id>3</ae:id>
    <ae:uuid>955e8e0e-9342-46ed-ba06-d1bfcd1cdf80</ae:uuid>
  </ae:application>
  <ae:description>Provides simple Incident Management
  →functionality for small Support teams.</ae:description>
  <ae:appDefUUID/>
  <ae:revision>2</ae:revision>
</ae:return>
```

Attachment-Access-Type

Description

Attachment-Access-Type indicates the type of attachment on an item. It is used for note, item link, URL, and file attachments. This type is used to determine whether the attachment has a restriction, is unrestricted, or if neither is set.

Parameters

Name	Type	Description
ATTACHACCESS-DEFAULT	string	Restrict the attachment only as specified by user privileges.
ATTACHACCESS-RESTRICTED	string	Makes the attachment visible only to users who can view the item.
ATTACHACCESS-UNRESTRICTED	string	Makes the attachment visible to all users who can view the item.

Usage

The Unrestricted status makes the file visible to all users who can view the item. You can set a file to have Default status to restrict the file as specified by user privileges. The Unrestricted status is disabled if you do not have privileges to set file attachments as unrestricted for the selected project or auxiliary table.

XML

The following XML shows Attachment-Access-Type in the `<urn:accessType>` element of a typical call.

```
<urn:attachmentContents>
  <urn:id>16</urn:id>
  <urn:name>pdf_doc</urn:name>
  <urn:fileName>relnotes.pdf</urn:fileName>
  <urn:showAsImage>false</urn:showAsImage>
  <urn:modificationDateTime></urn:modificationDateTime>
  <urn:url></urn:url>
  <urn:accessType>ATTACHACCESS-DEFAULT</urn:accessType>
  <urn:extendedData>
    <urn:data>
      <urn:name></urn:name>
      <urn:value></urn:value>
    </urn:data>
  </urn:extendedData>
  <urn:contentsBase64></urn:contentsBase64>
</urn:attachmentContents>
```

ContactIdentifier

Description

The ContactIdentifier type holds the identification information for a user's contact record. The ContactIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 94] .	Generic identification extension base. Holds the displayName, id, and uuid for a contact.

Usage

The ContactIdentifier is the identifier that can be used in Web service methods to uniquely identify a user's contact record. The ContactIdentifier contains the generic information about a contact (including the display name, ID, and UUID).



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows ContactIdentifier as seen in a typical call.

```
<urn:contact>
  <urn:displayName>Joe Manager</urn:displayName>
  <urn:id>1</urn:id>
  <urn:uuid>69d03cc6-e635-47d8-ab2e-a8be48a22f0a</urn:uuid>
</urn:contact>
```

FieldIdentifier

Description

The FieldIdentifier type holds the generic data for a field. The FieldIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 94] .	Generic identification extension base. Holds the displayName, id, and uuid for a field.
dbName	string	The unique database field name.

Usage

The FieldIdentifier is the identifier that can be used in Web service methods to uniquely identify a field. The FieldIdentifier contains the generic information about a field (including the display name, ID, and UUID) in addition to the database field name information for the field.



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows FieldIdentifier in the <urn:id> element in a typical call.

```
<urn:id>
  <urn:displayName>Actual Time to Fix</urn:displayName>
  <urn:id>59</urn:id>
  <urn:uuid>838fbaff-e74d-4d47-b415-85b502ea4676</urn:uuid>
  <urn:dbName>ACTUAL_TIME_TO_FIX</urn:dbName>
</urn:id>
```

FieldValue

Description

The FieldValue type holds generic identification information for a field value. The FieldValue parameters are listed below.

Parameters

Name	Type	Description
displayValue	string	The display name of the field value.
internalValue	string	The internal database name for the field value.
uuid	string	Alternate unique identifier for the field value.

Usage

FieldValue is used in combination with the Field argument to completely describe a field value pairing in FieldWithValue. For more information, see [FieldWithValue \[page 175\]](#).

XML

The following XML shows FieldValue in the <urn:value> element of the extendedField argument.

```
<urn:extendedField>
  <urn:id>
    <urn:displayName>SINGLE SELECTION</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName>SINGLE_SELECTION</urn:dbName>
  </urn:id>
  <urn:setValueBy>PRECEDENCE-VALUE</urn:setValueBy>
  <urn:setValueMethod></urn:setValueMethod>
  <urn:value>
    <urn:displayValue>yellow</urn:displayValue>
    <urn:internalValue>77</urn:internalValue>
    <urn:uuid>2d16b39d-1be9-44a0-805c-1f9074c8ac35</urn:uuid>
  </urn:value>
</urn:extendedField>
```

FileAttachment

Description

The FileAttachment type holds the details of a file that is attached to an item. The FileAttachment type parameters are listed below.

Parameters

Name	Type	Description
id	integer	This is the internal TS_ID of the attachment from the TS_ATTACHMENTS table.
name	string	The name you give the attachment in SBM.
fileName	string	The name of the file as it exists on the file system.
showAsImage	boolean	This flag indicates whether or not graphic attachments are shown as images in the SBM User Workspace.
modificationDateTime	dateTime	The date and time when the attachment was last modified. See Supported Date/Time Formats [page 298] for more information.

Name	Type	Description
url	string	The URL for the attachment. The file can be downloaded from this URL.
accessType	Attachment-Access-Type [page 87]	Shows the access type for the attachment. The value is either DEFAULT, RESTRICTED, or UNRESTRICTED.
extendedData	ExtendedData [page 142]	Placeholder for future arguments.

Usage

FileAttachmentContents is inherited from FileAttachment. FileAttachment is used with FileAttachmentContents to completely describe a file attachment. The file attachment detail consists of an ID, name, and URL. With the URL, client code can download the file directly.

XML

The following XML snippet shows the FileAttachment type in the `<urn:fileAttachment>` parameter of TTIItem.

```

<urn:fileAttachment>
  <urn:id>39</urn:id>
  <urn:name>test</urn:name>
  <urn:fileName>wslog.txt</urn:fileName>
  <urn:showAsImage>>false</urn:showAsImage>
  <urn:modificationDateTime>2010-06-20T15:35:38-07:00
  →</urn:modificationDateTime>
  <urn:url>http://server:80/tmtrack/tmtrack.dll?AttachmentPage
  →&AttachmentID=39
  </urn:url>
  <urn:accessType>ATTACHACCESS-DEFAULT</urn:accessType>
  <urn:extendedData>
    <urn:data>
      <urn:name></urn:name>
      <urn:value></urn:value>
    </urn:data>
  </urn:extendedData>
</urn:fileAttachment>

```

FileAttachmentContents

Description

The FileAttachmentContents type holds the actual contents of a file that you upload to the server. The FileAttachmentContents type parameters are listed below.

Parameters

Name	Type	Description
contentsBase64	FileBufferBase64 [page 92]	Holds the base64 encoded contents of the file.

Usage

Inherited from FileAttachment. FileAttachmentContents is used with FileAttachment to completely describe a file attachment. For more information, see [FileAttachment \[page 90\]](#).

XML

See [FileBufferBase64 \[page 92\]](#).

FileBufferBase64

Description

The FileBufferBase64 type holds the actual contents of a file that you upload to the server in Base64. The FileBufferBase64 type parameters are listed below.

Parameters

Name	Type	Description
data	base64Binary	Holds the Base64 encoded contents of the file.

Usage

None.

XML

The following XML is a snippet of the FileBufferBase64 type.

```
<ae:GetFileAttachmentResponse>
  <ae:return xsi:type="ae:FileAttachmentContents">
    <ae:id>104</ae:id>
    <ae:name>my attachment</ae:name>
    <ae:fileName>fileName.txt</ae:fileName>
    <ae:showAsImage>>false</ae:showAsImage>
    <ae:modificationDateTime>2010-09-17T18:46:25Z</ae:modificationDateTime>
    <ae:accessType>ATTACHACCESS-RESTRICTED</ae:accessType>
    <ae:contentsBase64>
      <ae:data>c2FtcGxlIGZpbGUgYXR0YWNobWVudA==</ae:data>
    </ae:contentsBase64>
  </ae:return>
</ae:GetFileAttachmentResponse>
```

FileContents

Description

The FileContents type holds the actual contents of a file returned from an Export operation or supplied to an Import operation. The FileContents type parameters are listed below.

Parameters

Name	Type	Description
contentsBase64	FileBufferBase64 [page 92]	Holds the base64 encoded contents of the file.

Usage

FileContents holds the actual contents of a file and is used to pass a file to or receive a file from the client.

XML

See [FileBufferBase64 \[page 92\]](#).

GroupIdentifier

Description

The GroupIdentifier type holds the identification information for a user group. The GroupIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 94] .	Generic identification extension base. Holds the displayName, id, and uuid for a group.

Usage

The GroupIdentifier is the identifier that can be used in Web service methods to uniquely identify a group. The GroupIdentifier contains the generic information about a group (including the display name, ID, and UUID).



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows GroupIdentifier as seen in a typical call.

```
<urn:group>
  <urn:displayName>CR Submitters</urn:displayName>
  <urn:id>11</urn:id>
  <urn:uuid>cf83a358-d7fb-4b96-8f98-ed532c66cd0a</urn:uuid>
</urn:group>
```

Identifier

Description

The Identifier type holds generic identification information. The Identifier type parameters are listed below.

Parameters

Name	Type	Description
displayName	string	The display name of the object.
id	integer	The TS_ID of the object.
uuid	string	An alternative unique identifier for the object.

Usage

The Identifier is a structure that contains generic identification information about an object in SBM. The identifier is a common set of parameters that are combined with other elements to uniquely define objects in the database. For example, the identifier element is used in combination with other elements like tableId, tableItemId, and issueId to uniquely describe an item in several different ways. The Identifier is used in the following common types:

- [ApplicationIdentifier \[page 86\]](#)
- [ContactIdentifier \[page 88\]](#)
- [FieldIdentifier \[page 89\]](#)
- [GroupIdentifier \[page 93\]](#)
- [ItemIdentifier \[page 95\]](#)
- [ProjectIdentifier \[page 100\]](#)
- [ReportIdentifier \[page 104\]](#)
- [SolutionIdentifier \[page 106\]](#)
- [StateIdentifier \[page 107\]](#)
- [TableIdentifier \[page 109\]](#)
- [TransitionIdentifier \[page 111\]](#)
- [UserIdentifier \[page 138\]](#)
- [WorkflowIdentifier \[page 139\]](#)



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows Identifier in the <ae:report> element of a typical call.

```
<urn:report>
  <urn:displayName>Change Requests By Issue</urn:displayName>
  <urn:id>8</urn:id>
  <urn:uuid>40e8bb61-14fe-409c-aa5e-6399cf3e26a8</urn:uuid>
</urn:report>
```

ItemIdentifier

Description

The ItemIdentifier type holds the identification information for an item. The ItemIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 94].	Generic identification extension base. Holds the displayName, id, and uuid for an item.
tableID	integer	The TS_ID of the table that contains this item.
tableIdItemId	string	The TS_ID of the table, followed by the TS_ID of the item in that table (for example, 1000:164).
issueId	string	Item name for display purposes.

Usage

The ItemIdentifier is the identifier that can be used in Web service methods to uniquely identify an item. The ItemIdentifier contains the generic information about an item (including the display name, ID, and UUID) in addition to table ID and issue ID information for the item.



Note: You can send any one of the elements in the identifier—you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object. You must provide either `tableIdItemId` or both `tableid` and `id` in calls that take an ItemIdentifier as input (such as `GetItem`).

XML

The following XML shows ItemIdentifier as seen in a typical call.

```
<urn:item>
  <urn:displayName>BUG000059</urn:displayName>
  <urn:id>25</urn:id>
  <urn:uuid>7d4703a0-302d-4da3-891e-1d36d43613f2</urn:uuid>
```

```

    <urn:tableId>1000</urn:tableId>
    <urn:tableIdItemId>1000:25</urn:tableIdItemId>
    <urn:issueId>000059</urn:issueId>
  </urn:item>

```

ItemLink

Description

The ItemLink type holds information about an item link, which consists of an ItemIdentifier and the type of item link. The ItemLink type parameters are listed below.

Parameters

Name	Type	Description
id	integer	This is the internal TS_ID of the item link from the TS_ATTACHMENTS table.
itemID	ItemIdentifier [page 95]	The ID of the linked item.
linkType	ItemLink-Type [page 97]	Describes the type of item link. See ItemLink-Type [page 97] for a list of possible values.
modificationDateTime	dateTime	The date and time when the item link was last modified. See Supported Date/Time Formats [page 298] for more information.
accessType	Attachment-Access-Type [page 87]	Shows the access type for the item link. The value is either DEFAULT, RESTRICTED, or UNRESTRICTED.
extendedData	ExtendedData [page 142]	Placeholder for future arguments.

Usage

The ItemLink type allows you to specify an item link to another item. The itemID element used uniquely identify a linked item.

XML

The following XML snippet shows the ItemLink type in the <urn:itemLink> parameter of TItem.

```

<urn:itemLink>
  <urn:id>52</urn:id>
  <urn:itemID>
    <urn:displayName>000220</urn:displayName>
    <urn:id>173</urn:id>
    <urn:uuid>0de24a1f-34a8-4f15-a72f-6ce923f0a212</urn:uuid>
  </urn:itemID>
</urn:itemLink>

```



```

    <urn:tableId>1000</urn:tableId>
    <urn:tableIdItemId>1000:173</urn:tableIdItemId>
    <urn:issueId>000220</urn:issueId>
  </urn:itemID>
  <urn:linkType>DEFAULT-ITEM-LINK</urn:linkType>
  <urn:modificationDateTime>2008-03-11T22:17:12-07:00
  →</urn:modificationDateTime>
  <urn:accessType>ATTACHACCESS-DEFAULT</urn:accessType>
  <urn:extendedData>
    <urn:data>
      <urn:name></urn:name>
      <urn:value></urn:value>
    </urn:data>
  </urn:extendedData>
</urn:itemLink>

```

ItemLink-Type

Description

The ItemLink-Type indicates the type of item link.

Parameters

Name	Type	Description
DEFAULT-ITEM-LINK	string	During creation of a new item link, the DEFAULT_ITEM_LINK is equivalent to TWOWAY_NO_TRIGGERS; otherwise, DEFAULT_ITEM_LINK indicates no type change. Note that returned item links always have DEFAULT_ITEM_LINK as the type.
TWOWAY-NO-TRIGGERS	string	Creates a two-way link between the current item and the selected item without Transition Triggers. This is the default link type.
ONEWAY-NO-TRIGGERS	string	Creates a one-way link from the current item to the selected item without Transition Triggers.
ONEWAY-CURRENT-TRIGGERS-LINKED	string	Creates a one-way link from the current item to the selected item. When the current item transitions, it also triggers the selected item to transition.
TWOWAY-CURRENT-TRIGGERS-LINKED	string	Creates a two-way link between the current item and the selected item. When the current item transitions, it also triggers the selected item to transition.

Name	Type	Description
TWOWAY-LINKED-TRIGGERS-CURRENT	string	Creates a two-way link between the current item and the selected item. When the selected item transitions, it also triggers the current item to transition.
TWOWAY-BOTH-TRIGGERS	string	Creates a two-way link between the current item and the selected item. When either item transitions, it also triggers the linked item to transition.

Usage

Linked items can trigger one another based on certain predefined actions defined in your workflow. You can specify a link type that triggers items, but triggers fire only if they are configured for your workflow.

XML

The following XML snippet shows ItemLink-Type in the <urn:linkType> element in a typical call.

```

<urn:itemLink>
  <urn:id>52</urn:id>
  <urn:itemID>
    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:tableId>1000</urn:tableId>
    <urn:tableIdItemId>1000:173</urn:tableIdItemId>
    <urn:issueId></urn:issueId>
  </urn:itemID>
  <urn:linkType>DEFAULT-ITEM-LINK</urn:linkType>
  <urn:modificationDateTime>2008-03-11T22:17:12-07:00
  →</urn:modificationDateTime>
  <urn:accessType>ATTACHACCESS-DEFAULT</urn:accessType>
  <urn:extendedData>
    <urn:data>
      <urn:name></urn:name>
      <urn:value></urn:value>
    </urn:data>
  </urn:extendedData>
</urn:itemLink>

```

Note

Description

The Note type holds information about a note. The Note type parameters are listed below.

Parameters

Name	Type	Description
id	integer	This is the internal TS_ID of the note from the TS_ATTACHMENTS table.
title	string	The title of the note.
note	string	The text of the note.
author	UserIdentifier [page 138]	The author of the note.
modificationDateTime	dateTime	The date and time when the note was last modified. See Supported Date/Time Formats [page 298] for more information.
accessType	Attachment-Access-Type [page 87]	Shows the access type for the note. The value is either DEFAULT, RESTRICTED, or UNRESTRICTED.
extendedData	ExtendedData [page 142]	Placeholder for future arguments.

Usage

The note title is limited to 255 unicode characters. The note body is limited to 65,535 characters.

XML

The following XML snippet shows the Note type in the `<urn:note>` parameter of TTIItem.

```
<urn:note>
  <urn:id>54</urn:id>
  <urn:title>Attention</urn:title>
  <urn:note>This is a note</urn:note>
  <urn:author>
    <urn:displayName>Administrator</urn:displayName>
    <urn:id>8</urn:id>
    <urn:uuid>9f9146a3-a273-4411-8000-8396688b7554</urn:uuid>
    <urn:loginId>admin</urn:loginId>
  </urn:author>
  <urn:modificationDateTime>2008-03-11T22:17:15-07:00
  →</urn:modificationDateTime>
  <urn:accessType>ATTACHACCESS-DEFAULT</urn:accessType>
  <urn:extendedData>
    <urn:data>
      <urn:name></urn:name>
      <urn:value></urn:value>
    </urn:data>
```

```

    </urn:extendedData>
  </urn:note>

```

ProjectIdentifier

Description

The ProjectIdentifier type holds the identification information for a project. The ProjectIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 94] .	Generic identification extension base. Holds the displayName, id, and uuid for a project.
fullyQualifiedName	string	The fully qualified name of the project. Specifies the project name as it exists in the project hierarchy with respect to project inheritance. To specify the fully qualified name for a project, you start with the Base Project, followed by any other projects in the inheritance chain that lead to your ultimate project.

Usage

The ProjectIdentifier is the identifier that can be used in Web service methods to uniquely identify a project. The ProjectIdentifier contains the generic information about a project (including the display name, ID, and UUID) in addition to the fully qualified name.



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows ProjectIdentifier as seen in a typical call.

```

<urn:project>
  <urn:displayName>Animation Pro</urn:displayName>
  <urn:id>2</urn:id>
  <urn:uuid>0b87f347-a00c-4359-9c16-625e847bfdab</urn:uuid>
  <urn:fullyQualifiedName>Base Project||Base IDT Project||
    Software Development||Animation Pro</urn:fullyQualifiedName>
</urn:project>

```

QueryRange

Description

The QueryRange type allows you to limit the number of records returned in a query. The QueryRange type parameters are listed below.

Parameters

Name	Type	Description
startIndex	integer	This is the first record where the query should start.
fetchSize	integer	The fetchSize number is used to limit the number records that should be returned from the entire set of records in the totalCount.
totalCount	integer	The number of all the records in the query.

Usage

You can limit the number of reports returned in [GetReports \[page 60\]](#) using QueryRange. For example, if GetReports returns 20 items without a QueryRange, you can set startIndex to 5 and fetchSize to 7 to return reports 5 through 12 in the list of available reports. You can also apply QueryRange in the RunReport call to limit the number of records returned via the fetchSize.

XML

The following XML snippet shows QueryRange as seen in a typical call.

```
<urn:queryRange>
  <urn:startIndex>5</urn:startIndex>
  <urn:fetchSize>7</urn:fetchSize>
  <urn:totalCount></urn:totalCount>
</urn:queryRange>
```

ReportAccessLevel

Description

ReportAccessLevel is used to describe the access level assigned to a report. The available report access levels are listed below.

Parameters

Name	Type	Description
PRIVATE	string	This access level enables individual users to manage reports they create. Only the user who creates a private report can access, modify, or delete private reports and only if this user is granted "Manage Private Reports" privileges.
GUEST	string	This access level denotes a guest-level report. Users with guest-level report privileges can perform report actions for guest-level reports.

Name	Type	Description
USER	string	This access level denotes a user-level report. Users with user-level report privileges can perform report actions for user-level reports.
MANAGER	string	This access level denotes a manager-level report. Users with manager-level report privileges can perform report actions for manager-level reports.

Usage

ReportAccessLevel is used to limit the reports returned in the reportsFilter argument of the GetReports call. If ReportAccessLevel is not specified, the GetReports response will not be limited by any type of access level. See [ReportsFilter \[page 151\]](#) for further usage.

XML

The following XML snippet shows ReportAccessLevel as seen in a typical call.

```

<urn:reportsFilter>
  <urn:solution></urn:solution>
  <urn:project>
    <urn:displayName></urn:displayName>
    <urn:id>4</urn:id>
    <urn:uuid></urn:uuid>
    <urn:fullyQualifiedName></urn:fullyQualifiedName>
  </urn:project>
  <urn:table>
    <urn:displayName></urn:displayName>
    <urn:id>1000</urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName></urn:dbName>
  </urn:table>
  <urn:author></urn:author>
  <urn:reportType>LISTING</urn:reportType>
  <urn:reportCategory>ALL</urn:reportCategory>
  <urn:reportAccessLevel>USER</urn:reportAccessLevel>
  <urn:report>
    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
  </urn:report>
  <urn:searchByName>ALL</urn:searchByName>
  <urn:includeSubProjects>true</urn:includeSubProjects>
  <urn:createdDateFrom>2007-06-20T15:35:38-07:00</urn:createdDateFrom>
  <urn:createdDateTo>2007-07-20T15:35:38-07:00</urn:createdDateTo>
</urn:reportsFilter>

```

ReportCategory

Description

ReportCategory is used to describe a grouping of reports. The available report categories are listed below.

Parameters

Name	Type	Description
ALL	string	This is the default ReportCategory. No filter is applied and all reports should be returned.
APPLICATION	string	Return only Application Reports (process app-specific listing reports designed in SBM Composer).
BUILTIN	string	Return only built-in reports.
MY	string	Return all reports that display in My Reports in the SBM User Workspace. Contains all reports that you have authored.
QUICKLINKS	string	Return only reports that are saved as Quick Links.
USERREPORTS	string	Return all reports, with the exception of built-in reports.

Usage

ReportCategory is used to limit the reports returned in the reportsFilter argument of the GetReports call. The default setting is "ALL" which will returns all reports (no filter is applied via this parameter in that case). See [ReportsFilter \[page 151\]](#) for further usage.

XML

The following XML snippet shows ReportCategory as seen in a typical call.

```
<urn:reportsFilter>
  <urn:solution>
    <urn:displayName></urn:displayName>
    <urn:id>1</urn:id>
    <urn:uuid></urn:uuid>
    <urn:uniqueName>BASE_ISSUE_DEFECT_TRACKING</urn:uniqueName>
    <urn:tabName></urn:tabName>
  </urn:solution>
  <urn:project>
    <urn:displayName></urn:displayName>
    <urn:id>4</urn:id>
    <urn:uuid></urn:uuid>
    <urn:fullyQualifiedName></urn:fullyQualifiedName>
  </urn:project>
  <urn:table>
    <urn:displayName></urn:displayName>
    <urn:id>1000</urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName></urn:dbName>
  </urn:table>
  <urn:author></urn:author>
  <urn:reportType>LISTING</urn:reportType>
  <urn:reportCategory>ALL</urn:reportCategory>
  <urn:reportAccessLevel>USER</urn:reportAccessLevel>
  <urn:report>
```

```

    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
  </urn:report>
  <urn:searchByName>ALL</urn:searchByName>
  <urn:includeSubProjects>true</urn:includeSubProjects>
  <urn:createdDateFrom>2007-06-20T15:35:38-07:00</urn:createdDateFrom>
  <urn:createdDateTo>2007-07-20T15:35:38-07:00</urn:createdDateTo>
</urn:reportsFilter>

```

ReportIdentifier

Description

The ReportIdentifier type holds the identification information for a report. The ReportIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 94] .	Generic identification extension base. Holds the displayName, id, and uuid for a report.

Usage

The ReportIdentifier is the identifier that can be used in Web service methods to uniquely identify a report. The ReportIdentifier contains the generic information about a report (including the display name, ID, and UUID).



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows ReportIdentifier as seen in a typical call.

```

<urn:report>
  <urn:displayName>Change Requests By Issue</urn:displayName>
  <urn:id>8</urn:id>
  <urn:uuid>40e8bb61-14fe-409c-aa5e-6399cf3e26a8</urn:uuid>
</urn:report>

```

ReportType

Description

ReportType is used to describe the type of report. The available report types are listed below.

Parameters

Name	Type	Description
LISTING	string	Indicates a Listing report type.
ALLTYPES	string	Returns all available report types.

Usage

ReportType is used to limit the types of reports returned in the reportsFilter argument of the GetReports call. The default setting is "1" or "LISTING" to indicate a Listing report. See [ReportsFilter \[page 151\]](#) for further usage.

XML

The following XML snippet shows ReportType as seen in a typical call.

```
<urn:reportsFilter>
  <urn:solution>
    <urn:displayName></urn:displayName>
    <urn:id>1</urn:id>
    <urn:uuid></urn:uuid>
    <urn:uniqueName>BASE_ISSUE_DEFECT_TRACKING</urn:uniqueName>
    <urn:tabName></urn:tabName>
  </urn:solution>
  <urn:project>
    <urn:displayName></urn:displayName>
    <urn:id>4</urn:id>
    <urn:uuid></urn:uuid>
    <urn:fullyQualifiedName></urn:fullyQualifiedName>
  </urn:project>
  <urn:table>
    <urn:displayName></urn:displayName>
    <urn:id>1000</urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName></urn:dbName>
  </urn:table>
  <urn:author></urn:author>
  <urn:reportType>LISTING</urn:reportType>
  <urn:reportCategory>ALL</urn:reportCategory>
  <urn:reportAccessLevel>USER</urn:reportAccessLevel>
  <urn:report>
    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
  </urn:report>
  <urn:searchByName>ALL</urn:searchByName>
  <urn:includeSubProjects>true</urn:includeSubProjects>
  <urn:createdDateFrom>2007-06-20T15:35:38-07:00</urn:createdDateFrom>
  <urn:createdDateTo>2007-07-20T15:35:38-07:00</urn:createdDateTo>
</urn:reportsFilter>
```

SolutionIdentifier

Description

The SolutionIdentifier type holds the identification information for a solution. The SolutionIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 94] .	Generic identification extension base. Holds the displayName, id, and uuid for a solution.
uniqueName	string	The database name of the solution.
tabName	string	The name that is displayed on the solution tab in the SBM User Workspace.

Usage

The SolutionIdentifier is the identifier that you use in Web service methods to uniquely identify a solution. The SolutionIdentifier contains the generic information about a solution (including the display name, ID, and UUID) in addition to the database name and SBM User Workspace tab name.



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows SolutionIdentifier as seen in a typical call.

```
<urn:solution>
  <urn:displayName>Issue Defect Management</urn:displayName>
  <urn:id>1</urn:id>
  <urn:uuid>cea0a86c-5d74-4e12-b8d6-9d6b90186f1e</urn:uuid>
  <urn:uniqueName>ISSUE_DEFECT_MANAGEMENT</urn:uniqueName>
  <urn:tabName>IDM</urn:tabName>
</urn:solution>
```

Solution-Type

Description

Solution-Type indicates the type of solution.

Parameters

Name	Type	Description
TEAMTRACK-SOLUTION	string	Indicates that the solution was created by Serena.
USER-SOLUTION	string	Indicates that the solution was created by a user.
THIRDPARTY-SOLUTION	string	Indicates that the solution was created by a third party.

Usage

The Solution-Type helps identify the type of solution returned in the SolutionData response. For more information, see [SolutionData \[page 191\]](#).

XML

The following XML shows Solution-Type as seen in a typical response.

```
<ae:GetSolutionsResponse>
  <ae:return>
    <ae:solution xsi:type="ae:SolutionIdentifier">
      <ae:displayName>Issue Defect Management</ae:displayName>
      <ae:id>1</ae:id>
      <ae:uuid>cea0a86c-5d74-4e12-b8d6-9d6b90186f1e</ae:uuid>
      <ae:uniqueName>ISSUE_DEFECT_MANAGEMENT</ae:uniqueName>
      <ae:tabName>IDM</ae:tabName>
    </ae:solution>
    <ae:type>USER-SOLUTION</ae:type>
    <ae:prefix>UBG</ae:prefix>
    <ae:description>Last updated 1/26/09</ae:description>
  </ae:return>
```

StateIdentifier

Description

The StateIdentifier type holds the identification information for a state. The StateIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 94] .	Generic identification extension base. Holds the displayName, id, and uuid for a state.
isClosed	boolean	Boolean value to indicate if the state is active or inactive.

Usage

The StateIdentifier is the identifier that can be used in Web service methods to uniquely identify a state. The StateIdentifier contains the generic information about a state (including the display name, ID, and UUID), in addition to whether or not the state is active or inactive.



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows StateIdentifier as seen in a typical call.

```
<urn:state>
  <urn:displayName>Evaluating Issue</urn:displayName>
  <urn:id>1</urn:id>
  <urn:uuid>985caf28-7a1c-4038-b6e2-c11703b214cd</urn:uuid>
  <urn:isClosed>>false</urn:isClosed>
</urn:state>
```

Subtasks

Description

The Subtasks type holds information about a subtask relationship. The Subtasks type is used only for informational purposes in the TTItem response. The Subtasks type parameters are listed below.

Parameters

Name	Type	Description
parentItemId	ItemIdentifier [page 95]	Identifies the parent item.
subtask	ItemIdentifier [page 95]	Identifies the subtask item.
extendedData	ExtendedData [page 142]	Placeholder for future arguments.

Usage

The Subtasks type contains the parent and child items that exist in a subtask relationship. The Subtasks type is returned in the TTItem response to inform you about subtask relationships on returned items. The Subtasks argument is left empty in the CreatePrimaryItems, CreateAuxiliaryItems, and TransitionItems calls. Instead, you must use the LinkSubtask call to create subtask relationships between items.

XML

The following XML shows the parent and subtask items in the `<ae:subtasks>` parameter of two separate TTItem responses. In this example, a subtask relationship was created using the LinkSubtask call, followed by a GetItems call against each item in the relationship.

GetItems against the parent item (where the TS_ID of the parent is 74) returns:

```

<ae:subtasks>
  <ae:subtask xsi:type="ae:ItemIdentifier">
    <ae:displayName>HRD000196</ae:displayName>
    <ae:id>75</ae:id>
    <ae:uuid>79e8af21-760d-44a4-af84-aec56935f88f</ae:uuid>
    <ae:tableId>1002</ae:tableId>
    <ae:tableIdItemId>1002:75</ae:tableIdItemId>
    <ae:issueId>000196</ae:issueId>
  </ae:subtask>
</ae:subtasks>

```

GetItems against the child item (where the TS_ID of the child is 75) returns:

```

<ae:subtasks>
  <ae:parentItemId xsi:type="ae:ItemIdentifier">
    <ae:displayName>HRD000195</ae:displayName>
    <ae:id>74</ae:id>
    <ae:uuid>5bf6642c-da90-436a-9fd6-2ec06363cdd0</ae:uuid>
    <ae:tableId>1002</ae:tableId>
    <ae:tableIdItemId>1002:74</ae:tableIdItemId>
    <ae:issueId>000195</ae:issueId>
  </ae:parentItemId>
</ae:subtasks>

```

TableIdentifier

Description

The TableIdentifier type holds the identification information for a table. The TableIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 94] .	Generic identification extension base. Holds the displayName, id, and uuid for a table.
dbName	string	The unique database name of the table

Usage

The TableIdentifier is the identifier that can be used in Web service methods to uniquely identify a table. The TableIdentifier contains the generic information about a table (including the display name, ID, and UUID) in addition to the database name of the table.



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows TableIdentifier as seen in a typical call.

```

<urn:table>
  <urn:displayName>Issues</urn:displayName>
  <urn:id>1000</urn:id>
  <urn:uuid>dc8cd329-b430-436f-bb75-bf90008e6a50</urn:uuid>
  <urn:dbName>UBG_ISSUES</urn:dbName>
</urn:table>

```

Table-Type

Description

Table-Type indicates the type of table. The available table types are listed below.

Parameters

Name	Type	Description
NOT-SPECIFIED	string	Used to not specify a table type.
SYSTEM-TABLE	string	A table type that is inherent to any SBM database.
PRIMARY-TABLE	string	A table that stores information about primary items, which follow an application workflow process.
AUXILIARY-TABLE	string	A table that stores information that may be needed repeatedly. Stores records that do not follow an application workflow process.
SYSTEM-AUXILIARY-TABLE	string	An built-in auxiliary table provided by SBM
ARCHIVE-TABLE	string	A built-in table used to store archived primary or auxiliary records.

Usage

Table-Type can be used to limit the types of tables returned in [GetTables \[page 65\]](#). It can also be used to identify the type of table returned in the GetTables response.

XML

The following XML shows Table-Type as seen in the type element of a GetTables response.

```

<ae:GetTablesResponse>
  <ae:return>
    <ae:table xsi:type="ae:TableIdentifier">
      <ae:displayName>Issues</ae:displayName>
      <ae:id>1000</ae:id>
      <ae:uuid>dc8cd329-b430-436f-bb75-bf90008e6a50</ae:uuid>
      <ae:dbName>UBG_ISSUES</ae:dbName>
    </ae:table>
  </ae:return>
</ae:GetTablesResponse>

```

```

</ae:table>
<ae:solution xsi:type="ae:SolutionIdentifier">
  <ae:displayName>Issue Defect Management</ae:displayName>
  <ae:id>1</ae:id>
  <ae:uuid>cea0a86c-5d74-4e12-b8d6-9d6b90186f1e</ae:uuid>
  <ae:uniqueName>ISSUE_DEFECT_MANAGEMENT</ae:uniqueName>
  <ae:tabName>IDM</ae:tabName>
</ae:solution>
<ae:type>PRIMARY-TABLE</ae:type>

```

TransitionIdentifier

Description

The TransitionIdentifier type holds the identification information for a transition. The TransitionIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 94] .	Generic identification extension base. Holds the displayName, id, and uuid for a transition.

Usage

The TransitionIdentifier is the identifier that can be used in Web service methods to uniquely identify a transition. The TransitionIdentifier contains the generic information about a transition (including the display name, ID, and UUID).



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows TransitionIdentifier as seen in a typical call.

```

<urn:transition>
  <urn:displayName>Approve</urn:displayName>
  <urn:id>4</urn:id>
  <urn:uuid>a78f0a30-1305-46c2-b661-df8219c105b2</urn:uuid>
</urn:transition>

```

Transition-Type

Description

Transition-Type indicates the type of transition. The available transition types are listed below.

Parameters

Name	Type	Description
TRANSITION-REGULAR	string	Denotes a Regular transition.
TRANSITION-COPY	string	Denotes a Copy transition.
TRANSITION-POST	string	Denotes a Post transition that submits a record into an application table based on the transition of a primary item.
TRANSITION-SUBMITPROBLEM	string	Denotes a Publish transition that creates a Knowledge Base problem or resolution.
TRANSITION-MOBILE	string	Denotes a transition that is available to Mobile Connect users.
TRANSITION-SUBTASK	string	Denotes a transition that creates a Subtask.
TRANSITION-UPDATE	string	Denotes an Update transition.
TRANSITION-DELETE	string	Denotes a Delete transition.
TRANSITION-EXTERNALPOST	string	Denotes a transition that will submit a record into an external database by sending an email message to perform a special kind of e-mail submission.

Usage

Transition-Type is used to describe the types of transition returned in the Transition response returned from the various GetTransition calls. For more information see [GetAvailableTransitions \[page 49\]](#).

XML

The following XML shows Transition-Type as seen in the type element of a GetAvailableTransitionsResponse.

```
<ae:GetAvailableTransitionsResponse>
  <ae:return>
    <ae:transition xsi:type="ae:TransitionIdentifier">
      <ae:displayName>Fix</ae:displayName>
      <ae:id>8</ae:id>
      <ae:uuid>7d095afe-1679-4e68-b492-0ad574bcdb2b</ae:uuid>
    </ae:transition>
    <ae:fromState xsi:type="ae:StateIdentifier">
      <ae:displayName>Evaluating Issue</ae:displayName>
      <ae:id>1</ae:id>
    </ae:fromState>
  </ae:return>
</ae:GetAvailableTransitionsResponse>
```

```

    <ae:uuid>985caf28-7a1c-4038-b6e2-c11703b214cd</ae:uuid>
    <ae:isClosed>>false</ae:isClosed>
  </ae:fromState>
  <ae:toState xsi:type="ae:StateIdentifier">
    <ae:displayName>Fixing Issue</ae:displayName>
    <ae:id>4</ae:id>
    <ae:uuid>a555a40c-7554-46f2-80bb-a6ff8b9ec917</ae:uuid>
    <ae:isClosed>>false</ae:isClosed>
  </ae:toState>
  <ae:type>TRANSITION-REGULAR</ae:type>
</ae:return>
<ae:return>
  <ae:transition xsi:type="ae:TransitionIdentifier">
    <ae:displayName>Need More Info</ae:displayName>
    <ae:id>4</ae:id>
    <ae:uuid>c2abdb46-47fa-46e3-a5f2-b90a401668e9</ae:uuid>
  </ae:transition>
  <ae:fromState xsi:type="ae:StateIdentifier">
    <ae:displayName>Evaluating Issue</ae:displayName>
    <ae:id>1</ae:id>
    <ae:uuid>985caf28-7a1c-4038-b6e2-c11703b214cd</ae:uuid>
    <ae:isClosed>>false</ae:isClosed>
  </ae:fromState>
  <ae:toState xsi:type="ae:StateIdentifier">
    <ae:displayName>Returned</ae:displayName>
    <ae:id>2</ae:id>
    <ae:uuid>2b2f2218-5d29-403d-b7e2-779d14b111b4</ae:uuid>
    <ae:isClosed>>false</ae:isClosed>
  </ae:toState>
  <ae:type>TRANSITION-REGULAR</ae:type>
</ae:return>
<ae:return>
  <ae:transition xsi:type="ae:TransitionIdentifier">
    <ae:displayName>Defer</ae:displayName>
    <ae:id>6</ae:id>
    <ae:uuid>ef75fd06-3a00-4e14-a1f9-e9b7b9c5d340</ae:uuid>
  </ae:transition>
  <ae:fromState xsi:type="ae:StateIdentifier">
    <ae:displayName>Evaluating Issue</ae:displayName>
    <ae:id>1</ae:id>
    <ae:uuid>985caf28-7a1c-4038-b6e2-c11703b214cd</ae:uuid>
    <ae:isClosed>>false</ae:isClosed>
  </ae:fromState>
  <ae:toState xsi:type="ae:StateIdentifier">
    <ae:displayName>Deferred</ae:displayName>
    <ae:id>3</ae:id>
    <ae:uuid>65b066a7-45f8-49cc-b5f4-0237f2fd3a2f</ae:uuid>
    <ae:isClosed>true</ae:isClosed>
  </ae:toState>
  <ae:type>TRANSITION-REGULAR</ae:type>
</ae:return>
<ae:return>
  <ae:transition xsi:type="ae:TransitionIdentifier">
    <ae:displayName>Copy</ae:displayName>
    <ae:id>19</ae:id>

```

```

    <ae:uuid>34e41d1a-9e30-4811-ad40-202bd397ea68</ae:uuid>
  </ae:transition>
  <ae:type>TRANSITION-COPY</ae:type>
</ae:return>
<ae:return>
  <ae:transition xsi:type="ae:TransitionIdentifier">
    <ae:displayName>Close</ae:displayName>
    <ae:id>20</ae:id>
    <ae:uuid>970ae164-c2fb-4709-8539-7f295add3de</ae:uuid>
  </ae:transition>
  <ae:fromState xsi:type="ae:StateIdentifier">
    <ae:displayName>Evaluating Issue</ae:displayName>
    <ae:id>1</ae:id>
    <ae:uuid>985caf28-7a1c-4038-b6e2-c11703b214cd</ae:uuid>
    <ae:isClosed>>false</ae:isClosed>
  </ae:fromState>
  <ae:toState xsi:type="ae:StateIdentifier">
    <ae:displayName>Resolved</ae:displayName>
    <ae:id>7</ae:id>
    <ae:uuid>ea57582f-5c62-4bcd-b250-5e21cad308e3</ae:uuid>
    <ae:isClosed>>true</ae:isClosed>
  </ae:toState>
  <ae:type>TRANSITION-REGULAR</ae:type>
</ae:return>
<ae:return>
  <ae:transition xsi:type="ae:TransitionIdentifier">
    <ae:displayName>P4 Update</ae:displayName>
    <ae:id>66</ae:id>
    <ae:uuid>e3a41611-f1e3-4a94-bc54-304f0fcd2227</ae:uuid>
  </ae:transition>
  <ae:type>TRANSITION-REGULAR</ae:type>
</ae:return>
<ae:return>
  <ae:transition xsi:type="ae:TransitionIdentifier">
    <ae:displayName>Update</ae:displayName>
    <ae:id>1</ae:id>
    <ae:uuid>update</ae:uuid>
  </ae:transition>
  <ae:type>TRANSITION-UPDATE</ae:type>
</ae:return>
<ae:return>
  <ae:transition xsi:type="ae:TransitionIdentifier">
    <ae:displayName>Delete</ae:displayName>
    <ae:id>2</ae:id>
    <ae:uuid>delete</ae:uuid>
  </ae:transition>
  <ae:type>TRANSITION-DELETE</ae:type>
</ae:return>
</ae:GetAvailableTransitionsResponse>

```

TTItem

Description

The TTIItem type holds all of the SBM field information for an item along with attached notes and item links. The TTIItem type parameters are listed below.

Parameters

Name	Type	Description
id	ItemIdentifier [page 95]	Holds the identification information for an item.
itemType	string	Describes items, such as Defects or Change Requests, tracked in a workflow.
project	ProjectIdentifier [page 100]	Holds the identification information for the item's project.
title	string	The 80 character fixed length title of the item.
description	string	A description of the item. This is the value given to the system Description field, which is a Text/Memo field.
createdBy	UserIdentifier [page 138]	Holds the identification information for the user who created the item.
createDate	dateTime	The date the item was created. See Supported Date/Time Formats [page 298] for more information.
modifiedBy	UserIdentifier [page 138]	Holds the identification information of the user who last modified the item.
modifiedDate	dateTime	The date and time when the item was last modified. See Supported Date/Time Formats [page 298] for more information.
activeInactive	string	Indicates whether the item is active (true) or inactive (false). Defaults to true.
state	StateIdentifier [page 107]	Holds the identification information of the current state of the item.
owner	UserIdentifier [page 138]	Holds the identification information of the user who currently owns the item.
url	string	The exact URL of the item.
extendedField	NameValue [page 148]	Additional fields and values not otherwise specified in TTIItem are set using the extendedField element.
note	Note [page 98]	The list of notes attached to the item.

Name	Type	Description
itemLink	ItemLink [page 96]	The list of items linked to this item.
urlAttachment	URLAttachment [page 137]	The list of URL attachments associated with this item.
fileAttachment	FileAttachment [page 90]	The list of file attachments associated with this item.
subtasks	Subtasks [page 108]	Holds information about an existing subtask relationship. Left empty when sent as an argument. Used only for informational purposes in a TTItem response.
extendedData	ExtendedData [page 142]	Placeholder for future arguments.

Usage

The TTitem type is essentially used describe the fields of an auxiliary or primary item in SBM. The following TTitem parameters are described in further detail:

- **itemType**

The itemType parameter is used to describe the item. The available values for itemType are derived from selections in the Item Type field. When you create an Item Type value in SBM Composer, you create a value and assign it a prefix. In the itemType parameter, you enter only the value, not the prefix. See the XML sample below for an example. Item Type is only applicable to primary table items.

- **project**

The project parameter is used to assign an item to a project. When creating items using Web services, make sure the proper project is used with respect to the itemtype, state, and owner. Note that the project parameter is only applicable to primary table items.

- **state**

The available states for an item are determined by the project the item is in. If you change the value of the state field for a specific item, you are also moving that item to the specified state in the workflow and it could follow a different process than you intended. If not specified, the initial state is the default. State is only applicable to primary table items.

- **owner**

The available owner of an item is determined by the state the item is in. If not explicitly set, the value for owner will be derived from the value of the user field designated as the owner of the state. Owner is only applicable to primary table items.

- **url**

The detailed view of the item can be accessed via this URL. The URL is of the form "tmtrack.dll?IssuePage&RecordId=10&Template=view&TableId=1002", so the "http://host:port/tmtrack/" must be pre-pended.

- **extendedField**

The extendedField is a collection of fields and field values that aren't specifically set elsewhere in TItem. The XML example below shows each field type in SBM, sent via TItem in a CreatePrimaryItems call. The example shows how the data for each field type should be entered, using either the display or internal value. Both the display and internal values are shown in the example, though you can specify either the display, internal, or UUID value instead. Note that Multi-Select and Multi-Relational fields are included as well, with each value being sent in its own element.



Tip: In order to create a new auxiliary or primary item with a file attachment, that attachment must exist in the TS_ATTACHMENTS table of SBM already. Otherwise, you can use CreateFileAttachment after the item has been created to associate it with a new attachment that isn't already in TS_ATTACHMENTS.

XML

The following XML shows TItem as seen in the CreatePrimaryItems call. Throughout the extendedFields in this example, all of the SBM field types are used in the name element to illustrate how to format the data for that given field type.

```
<urn:CreatePrimaryItems>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:project>
    <urn:displayName>Animation Pro</urn:displayName>
    <urn:id>6</urn:id>
    <urn:uuid></urn:uuid>
    <urn:fullyQualifiedName>Base Project||Base IDT Project||
      Software Development||Animation Pro</urn:fullyQualifiedName>
  </urn:project>
  <urn:parentItem></urn:parentItem>
  <urn:item>
    <urn:id>
      <urn:displayName>000230</urn:displayName>
      <urn:id>196</urn:id>
      <urn:uuid></urn:uuid>
      <urn:tableId>1000</urn:tableId>
      <urn:tableIdItemId>1000:196</urn:tableIdItemId>
      <urn:issueId>000230</urn:issueId>
    </urn:id>
    <urn:itemType></urn:itemType>
    <urn:project>
      <urn:displayName></urn:displayName>
      <urn:id></urn:id>
      <urn:uuid></urn:uuid>
      <urn:fullyQualifiedName></urn:fullyQualifiedName>
    </urn:project>
  </urn:item>
</urn:CreatePrimaryItems>
```

```
<urn:title>Test Item for TTItem</urn:title>
<urn:description>This is the item description.</urn:description>
<urn:createdBy>
  <urn:displayName></urn:displayName>
  <urn:id></urn:id>
  <urn:uuid></urn:uuid>
  <urn:loginId></urn:loginId>
</urn:createdBy>
<urn:createDate></urn:createDate>
<urn:modifiedBy>
  <urn:displayName></urn:displayName>
  <urn:id></urn:id>
  <urn:uuid></urn:uuid>
  <urn:loginId></urn:loginId>
</urn:modifiedBy>
<urn:modifiedDate></urn:modifiedDate>
<urn:activeInactive></urn:activeInactive>
<urn:state>
  <urn:displayName></urn:displayName>
  <urn:id></urn:id>
  <urn:uuid></urn:uuid>
  <urn:isClosed></urn:isClosed>
</urn:state>
<urn:owner>
  <urn:displayName></urn:displayName>
  <urn:id></urn:id>
  <urn:uuid></urn:uuid>
  <urn:loginId></urn:loginId>
</urn:owner>
<urn:url></urn:url>
<urn:extendedField>
  <urn:id>
    <urn:displayName>BINARY TRINARY</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName>BINARY_TRINARY</urn:dbName>
  </urn:id>
  <urn:setValueBy>PRECEDENCE-VALUE</urn:setValueBy>
  <urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
  <urn:value>
    <urn:displayValue>True</urn:displayValue>
    <urn:internalValue></urn:internalValue>
    <urn:uuid></urn:uuid>
  </urn:value>
</urn:extendedField>
<urn:extendedField>
  <urn:id>
    <urn:displayName>DATE AND TIME</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName>DATE_AND_TIME</urn:dbName>
  </urn:id>
  <urn:setValueBy>PRECEDENCE-VALUE</urn:setValueBy>
  <urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
  <urn:value>
```

```

        <urn:displayValue>2010-05-01T06:00:00+00:00</urn:displayValue>
        <urn:internalValue>2010-05-01T06:00:00+00:00</urn:internalValue>
        <urn:uuid></urn:uuid>
    </urn:value>
</urn:extendedField>
<urn:extendedField>
    <urn:id>
        <urn:displayName>DATE ONLY</urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
        <urn:dbName>DATE_ONLY</urn:dbName>
    </urn:id>
    <urn:setValueBy>PRECEDENCE-VALUE</urn:setValueBy>
    <urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
    <urn:value>
        <urn:displayValue>2010-05-01</urn:displayValue>
        <urn:internalValue>2008-05-01T00:00:00+00:00</urn:internalValue>
        <urn:uuid></urn:uuid>
    </urn:value>
</urn:extendedField>
<urn:extendedField>
    <urn:id>
        <urn:displayName>TIME OF DAY</urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
        <urn:dbName>TIME_OF_DAY</urn:dbName>
    </urn:id>
    <urn:setValueBy>PRECEDENCE-VALUE</urn:setValueBy>
    <urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
    <urn:value>
        <urn:displayValue>17:36:39</urn:displayValue>
        <urn:internalValue>63399</urn:internalValue>
        <urn:uuid></urn:uuid>
    </urn:value>
</urn:extendedField>
<urn:extendedField>
    <urn:id>
        <urn:displayName>ELAPSED TIME</urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
        <urn:dbName>ELAPSED_TIME</urn:dbName>
    </urn:id>
    <urn:setValueBy>PRECEDENCE-VALUE</urn:setValueBy>
    <urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
    <urn:value>
        <urn:displayValue>7 17:36:39</urn:displayValue>
        <urn:internalValue>668199</urn:internalValue>
        <urn:uuid></urn:uuid>
    </urn:value>
</urn:extendedField>
<urn:extendedField>
    <urn:id>
        <urn:displayName>FOLDER</urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>

```

```
        <urn:dbName>FOLDER</urn:dbName>
    </urn:id>
    <urn:setValueBy>PRECEDENCE-VALUE</urn:setValueBy>
    <urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
    <urn:value>
        <urn:displayValue>InBox</urn:displayValue>
        <urn:internalValue></urn:internalValue>
        <urn:uuid></urn:uuid>
    </urn:value>
</urn:extendedField>
<urn:extendedField>
    <urn:id>
        <urn:displayName>MULTI  GROUP</urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
        <urn:dbName>MULTI_GROUP</urn:dbName>
    </urn:id>
    <urn:setValueBy>PRECEDENCE-VALUE</urn:setValueBy>
    <urn:setValueMethod></urn:setValueMethod>
    <urn:value>
        <urn:displayValue>IDM Administrator</urn:displayValue>
        <urn:internalValue></urn:internalValue>
        <urn:uuid></urn:uuid>
    </urn:value>
    <urn:value>
        <urn:displayValue>IDM Team</urn:displayValue>
        <urn:internalValue></urn:internalValue>
        <urn:uuid></urn:uuid>
    </urn:value>
</urn:extendedField>
<urn:extendedField>
    <urn:id>
        <urn:displayName>MULTI  RELATIONAL</urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
        <urn:dbName>MULTI_RELATIONAL</urn:dbName>
    </urn:id>
    <urn:setValueBy>PRECEDENCE-VALUE</urn:setValueBy>
    <urn:setValueMethod></urn:setValueMethod>
    <urn:value>
        <urn:displayValue>App1 - test123</urn:displayValue>
        <urn:internalValue>1004:1</urn:internalValue>
        <urn:uuid></urn:uuid>
    </urn:value>
    <urn:value>
        <urn:displayValue>App2 - field456</urn:displayValue>
        <urn:internalValue>1004:2</urn:internalValue>
        <urn:uuid></urn:uuid>
    </urn:value>
</urn:extendedField>
<urn:extendedField>
    <urn:id>
        <urn:displayName>MULTI  SELECTION</urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
```

```

        <urn:dbName>MULTI_SELECTION</urn:dbName>
    </urn:id>
    <urn:setValueBy>PRECEDENCE-VALUE</urn:setValueBy>
    <urn:setValueMethod></urn:setValueMethod>
    <urn:value>
        <urn:displayValue>red</urn:displayValue>
        <urn:internalValue></urn:internalValue>
        <urn:uuid></urn:uuid>
    </urn:value>
    <urn:value>
        <urn:displayValue>blue</urn:displayValue>
        <urn:internalValue></urn:internalValue>
        <urn:uuid></urn:uuid>
    </urn:value>
        <urn:value>
            <urn:displayValue>green</urn:displayValue>
            <urn:internalValue></urn:internalValue>
            <urn:uuid></urn:uuid>
        </urn:value>
    </urn:extendedField>
</urn:extendedField>
<urn:extendedField>
    <urn:id>
        <urn:displayName>MULTI_USER</urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
        <urn:dbName>MULTI_USER</urn:dbName>
    </urn:id>
    <urn:setValueBy>PRECEDENCE-VALUE</urn:setValueBy>
    <urn:setValueMethod></urn:setValueMethod>
    <urn:value>
        <urn:displayValue>admin</urn:displayValue>
        <urn:internalValue></urn:internalValue>
        <urn:uuid></urn:uuid>
    </urn:value>
    <urn:value>
        <urn:displayValue>bill</urn:displayValue>
        <urn:internalValue></urn:internalValue>
        <urn:uuid></urn:uuid>
    </urn:value>
</urn:extendedField>
</urn:extendedField>
<urn:extendedField>
    <urn:id>
        <urn:displayName>NUMERIC</urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
        <urn:dbName>NUMERIC</urn:dbName>
    </urn:id>
    <urn:setValueBy>PRECEDENCE-VALUE</urn:setValueBy>
    <urn:setValueMethod></urn:setValueMethod>
    <urn:value>
        <urn:displayValue>55</urn:displayValue>
        <urn:internalValue>55</urn:internalValue>
        <urn:uuid></urn:uuid>
    </urn:value>
</urn:extendedField>

```

```
<urn:extendedField>
  <urn:id>
    <urn:displayName>SINGLE RELATIONAL</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName>SINGLE_RELATIONAL</urn:dbName>
  </urn:id>
  <urn:setValueBy>PRECEDENCE-VALUE</urn:setValueBy>
  <urn:setValueMethod></urn:setValueMethod>
  <urn:value>
    <urn:displayValue>Appl - test123</urn:displayValue>
    <urn:internalValue>1004:1</urn:internalValue>
    <urn:uuid></urn:uuid>
  </urn:value>
</urn:extendedField>
<urn:extendedField>
  <urn:id>
    <urn:displayName>SINGLE SELECTION</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName>SINGLE_SELECTION</urn:dbName>
  </urn:id>
  <urn:setValueBy>PRECEDENCE-VALUE</urn:setValueBy>
  <urn:setValueMethod></urn:setValueMethod>
  <urn:value>
    <urn:displayValue>yellow</urn:displayValue>
    <urn:internalValue></urn:internalValue>
    <urn:uuid></urn:uuid>
  </urn:value>
</urn:extendedField>
<urn:extendedField>
  <urn:id>
    <urn:displayName>NUMERIC 2 FOR SUM</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName>NUMERIC_2_FOR_SUM</urn:dbName>
  </urn:id>
  <urn:setValueBy>PRECEDENCE-VALUE</urn:setValueBy>
  <urn:setValueMethod></urn:setValueMethod>
  <urn:value>
    <urn:displayValue>10</urn:displayValue>
    <urn:internalValue>10</urn:internalValue>
    <urn:uuid></urn:uuid>
  </urn:value>
</urn:extendedField>
<urn:extendedField>
  <urn:id>
    <urn:displayName>SUMMATION</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName>SUMMATION</urn:dbName>
  </urn:id>
  <urn:setValueBy>PRECEDENCE-VALUE</urn:setValueBy>
  <urn:setValueMethod></urn:setValueMethod>
  <urn:value>
```

```

        <urn:displayValue></urn:displayValue>
        <urn:internalValue></urn:internalValue>
        <urn:uuid></urn:uuid>
    </urn:value>
</urn:extendedField>
<urn:extendedField>
    <urn:id>
        <urn:displayName>TEXT</urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
        <urn:dbName>TEXT</urn:dbName>
    </urn:id>
    <urn:setValueBy>PRECEDENCE-VALUE</urn:setValueBy>
    <urn:setValueMethod></urn:setValueMethod>
    <urn:value>
        <urn:displayValue>This is some text.</urn:displayValue>
        <urn:internalValue>This is some text.</urn:internalValue>
        <urn:uuid></urn:uuid>
    </urn:value>
</urn:extendedField>
<urn:extendedField>
    <urn:id>
        <urn:displayName>USER</urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
        <urn:dbName>USER</urn:dbName>
    </urn:id>
    <urn:setValueBy>PRECEDENCE-VALUE</urn:setValueBy>
    <urn:setValueMethod></urn:setValueMethod>
    <urn:value>
        <urn:displayValue>carmen</urn:displayValue>
        <urn:internalValue></urn:internalValue>
        <urn:uuid></urn:uuid>
    </urn:value>
</urn:extendedField>
<urn:note>
    <urn:id></urn:id>
    <urn:title>Note Title</urn:title>
    <urn:note>this is a note</urn:note>
    <urn:author>
        <urn:displayName>admin</urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
        <urn:loginId></urn:loginId>
    </urn:author>
    <urn:modificationDateTime></urn:modificationDateTime>
    <urn:accessType>ATTACHACCESS-DEFAULT</urn:accessType>
</urn:note>
<urn:itemLink>
    <urn:id></urn:id>
    <urn:itemID>
        <urn:displayName></urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
        <urn:tableId>1000</urn:tableId>

```

```
        <urn:tableItemId>1000:190</urn:tableItemId>
        <urn:issueId></urn:issueId>
    </urn:itemID>
    <urn:linkType>DEFAULT-ITEM-LINK</urn:linkType>
    <urn:modificationDateTime></urn:modificationDateTime>
    <urn:accessType>ATTACHACCESS-DEFAULT</urn:accessType>
</urn:itemLink>
<urn:fileAttachment>
    <urn:id></urn:id>
    <urn:name></urn:name>
    <urn:fileName></urn:fileName>
    <urn:showAsImage></urn:showAsImage>
    <urn:modificationDateTime></urn:modificationDateTime>
    <urn:url></urn:url>
    <urn:accessType>ATTACHACCESS-DEFAULT</urn:accessType>
</urn:fileAttachment>
<urn:subtasks/>
</urn:item>
<urn:submitTransition>
    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
</urn:submitTransition>
<urn:options>
    <urn:extraOption></urn:extraOption>
    <urn:multiOption>STOP-ON-FAILURE</urn:multiOption>
    <urn:sections>SECTIONS-ALL</urn:sections>
    <urn:specifiedSections></urn:specifiedSections>
    <urn:limitedField></urn:limitedField>
</urn:options>
</urn>CreatePrimaryItems>
```

The following XML shows TItem as seen in the CreatePrimaryItems response.

```
<ae:CreatePrimaryItemsResponse>
  <ae:return>
    <ae:item>
      <ae:id xsi:type="ae:ItemIdentifier">
        <ae:displayName>000230</ae:displayName>
        <ae:id>196</ae:id>
        <ae:uuid>fafaf647-22f5-4579-b60c-3f90e61bdcf0</ae:uuid>
        <ae:tableId>1000</ae:tableId>
        <ae:tableItemId>1000:196</ae:tableItemId>
        <ae:issueId>000230</ae:issueId>
      </ae:id>
      <ae:itemType/>
      <ae:project xsi:type="ae:ProjectIdentifier">
        <ae:displayName>Animation Pro</ae:displayName>
        <ae:id>6</ae:id>
        <ae:uuid>2ac5ef27-71da-4b07-ab7e-dddbc9c2d8c7</ae:uuid>
        <ae:fullyQualifiedName>Base Project||Base IDT Project||
          Software Development||Animation Pro</ae:fullyQualifiedName>
      </ae:project>
      <ae:title>Test Item for TItem</ae:title>
```

```
<ae:description>This is the item description.</ae:description>
<ae:createdBy xsi:type="ae:UserIdentifier">
  <ae:displayName>Administrator</ae:displayName>
  <ae:id>8</ae:id>
  <ae:uuid>9f9146a3-a273-4411-8000-8396688b7554</ae:uuid>
  <ae:loginId>admin</ae:loginId>
</ae:createdBy>
<ae:createDate>2010-09-14T22:39:41Z</ae:createDate>
<ae:modifiedBy xsi:type="ae:UserIdentifier">
  <ae:displayName/>
  <ae:id>0</ae:id>
  <ae:uuid/>
  <ae:loginId/>
</ae:modifiedBy>
<ae:modifiedDate>2010-09-14T22:39:41Z</ae:modifiedDate>
<ae:activeInactive>>false</ae:activeInactive>
<ae:state xsi:type="ae:StateIdentifier">
  <ae:displayName/>
  <ae:id>0</ae:id>
  <ae:uuid/>
  <ae:isClosed>>false</ae:isClosed>
</ae:state>
<ae:owner xsi:type="ae:UserIdentifier">
  <ae:displayName/>
  <ae:id>0</ae:id>
  <ae:uuid/>
  <ae:loginId/>
</ae:owner>
<ae:url>http://cs1372:80/tmtrack/tmtrack.dll?IssuePage&
→RecordId=196&Template=view&TableId=1000</ae:url>
<ae:extendedField>
  <ae:id xsi:type="ae:FieldIdentifier">
    <ae:displayName>Steps to Reproduce</ae:displayName>
    <ae:id>56</ae:id>
    <ae:uuid>9a9545f3-984a-4f1c-92eb-481ebbaab733</ae:uuid>
    <ae:dbName>STEPS_TO_REPRODUCE_</ae:dbName>
  </ae:id>
  <ae:value>
    <ae:displayValue/>
    <ae:internalValue/>
    <ae:uuid/>
  </ae:value>
</ae:extendedField>
<ae:extendedField>
  <ae:id xsi:type="ae:FieldIdentifier">
    <ae:displayName>QA APP BUG ID</ae:displayName>
    <ae:id>57</ae:id>
    <ae:uuid>f44ff842-2973-486e-8298-4a1332111998</ae:uuid>
    <ae:dbName>QA_APP_BUG_ID</ae:dbName>
  </ae:id>
  <ae:value>
    <ae:displayValue/>
    <ae:internalValue/>
    <ae:uuid/>
  </ae:value>
```

```
</ae:extendedField>
<ae:extendedField>
  <ae:id xsi:type="ae:FieldIdentifier">
    <ae:displayName>Actual Time to Fix</ae:displayName>
    <ae:id>59</ae:id>
    <ae:uuid>838fbaff-e74d-4d47-b415-85b502ea4676</ae:uuid>
    <ae:dbName>ACTUAL_TIME_TO_FIX</ae:dbName>
  </ae:id>
  <ae:value>
    <ae:displayValue>0.00</ae:displayValue>
    <ae:internalValue>0.00</ae:internalValue>
    <ae:uuid/>
  </ae:value>
</ae:extendedField>
<ae:extendedField>
  <ae:id xsi:type="ae:FieldIdentifier">
    <ae:displayName>Time Remaining</ae:displayName>
    <ae:id>60</ae:id>
    <ae:uuid>e3326ea7-a02e-4651-a24f-805b2980fbc2</ae:uuid>
    <ae:dbName>TIME_REMAINING</ae:dbName>
  </ae:id>
  <ae:value>
    <ae:displayValue>0.00</ae:displayValue>
    <ae:internalValue>0.00</ae:internalValue>
    <ae:uuid/>
  </ae:value>
</ae:extendedField>
<ae:extendedField>
  <ae:id xsi:type="ae:FieldIdentifier">
    <ae:displayName>Est Time to Fix</ae:displayName>
    <ae:id>61</ae:id>
    <ae:uuid>7077c520-cf68-45fc-97aa-3721687ec7cf</ae:uuid>
    <ae:dbName>EST_TIME_TO_FIX</ae:dbName>
  </ae:id>
  <ae:value>
    <ae:displayValue>0.00</ae:displayValue>
    <ae:internalValue>0.00</ae:internalValue>
    <ae:uuid/>
  </ae:value>
</ae:extendedField>
<ae:extendedField>
  <ae:id xsi:type="ae:FieldIdentifier">
    <ae:displayName>Est Date to Fix</ae:displayName>
    <ae:id>62</ae:id>
    <ae:uuid>1a47ca48-5bfd-47e5-b62f-92081484de0a</ae:uuid>
    <ae:dbName>EST_DATE_TO_FIX</ae:dbName>
  </ae:id>
  <ae:value>
    <ae:displayValue/>
    <ae:internalValue/>
    <ae:uuid/>
  </ae:value>
</ae:extendedField>
<ae:extendedField>
  <ae:id xsi:type="ae:FieldIdentifier">
```

```

        <ae:displayName>Tester</ae:displayName>
        <ae:id>63</ae:id>
        <ae:uuid>34b5810f-c1ee-49e9-b3c2-15949e13e54b</ae:uuid>
        <ae:dbName>TESTER</ae:dbName>
    </ae:id>
    <ae:value>
        <ae:displayValue/>
        <ae:internalValue>0</ae:internalValue>
        <ae:uuid/>
    </ae:value>
</ae:extendedField>
<ae:extendedField>
    <ae:id xsi:type="ae:FieldIdentifier">
        <ae:displayName>Resolution</ae:displayName>
        <ae:id>64</ae:id>
        <ae:uuid>cee5ba03-235b-484a-bd28-b425a73fa849</ae:uuid>
        <ae:dbName>RESOLUTION</ae:dbName>
    </ae:id>
    <ae:value>
        <ae:displayValue/>
        <ae:internalValue>0</ae:internalValue>
        <ae:uuid/>
    </ae:value>
</ae:extendedField>
<ae:extendedField>
    <ae:id xsi:type="ae:FieldIdentifier">
        <ae:displayName>Status Log</ae:displayName>
        <ae:id>65</ae:id>
        <ae:uuid>c96df948-ce4a-4851-9c4d-5adalf20ed0d</ae:uuid>
        <ae:dbName>STATUS_LOG</ae:dbName>
    </ae:id>
    <ae:value>
        <ae:displayValue/>
        <ae:internalValue/>
        <ae:uuid/>
    </ae:value>
</ae:extendedField>
<ae:extendedField>
    <ae:id xsi:type="ae:FieldIdentifier">
        <ae:displayName>Developer</ae:displayName>
        <ae:id>66</ae:id>
        <ae:uuid>21aead0a-8127-4685-b4f1-4b79cd74504b</ae:uuid>
        <ae:dbName>DEVELOPER</ae:dbName>
    </ae:id>
    <ae:value>
        <ae:displayValue/>
        <ae:internalValue>0</ae:internalValue>
        <ae:uuid/>
    </ae:value>
</ae:extendedField>
<ae:extendedField>
    <ae:id xsi:type="ae:FieldIdentifier">
        <ae:displayName>Team Lead</ae:displayName>
        <ae:id>67</ae:id>
        <ae:uuid>0a460f3e-fe35-4078-a358-9961e41bebe5</ae:uuid>

```

```
        <ae:dbName>TEAM_LEAD</ae:dbName>
    </ae:id>
    <ae:value>
        <ae:displayValue>joe</ae:displayValue>
        <ae:internalValue>2</ae:internalValue>
        <ae:uuid>85a33f0b-9542-43fe-90c1-e152eeee777f</ae:uuid>
    </ae:value>
</ae:extendedField>
<ae:extendedField>
    <ae:id xsi:type="ae:FieldIdentifier">
        <ae:displayName>Peer Reviewer</ae:displayName>
        <ae:id>68</ae:id>
        <ae:uuid>56f5fa31-be98-4100-a78d-c0bd45115209</ae:uuid>
        <ae:dbName>PEER_REVIEWER</ae:dbName>
    </ae:id>
    <ae:value>
        <ae:displayValue/>
        <ae:internalValue>0</ae:internalValue>
        <ae:uuid/>
    </ae:value>
</ae:extendedField>
<ae:extendedField>
    <ae:id xsi:type="ae:FieldIdentifier">
        <ae:displayName>Severity</ae:displayName>
        <ae:id>69</ae:id>
        <ae:uuid>f4eff572-2e29-4d7f-8478-8e9e16865c2c</ae:uuid>
        <ae:dbName>SEVERITY</ae:dbName>
    </ae:id>
    <ae:value>
        <ae:displayValue/>
        <ae:internalValue>0</ae:internalValue>
        <ae:uuid/>
    </ae:value>
</ae:extendedField>
<ae:extendedField>
    <ae:id xsi:type="ae:FieldIdentifier">
        <ae:displayName>Reproducible</ae:displayName>
        <ae:id>70</ae:id>
        <ae:uuid>6b21b7e8-f496-43fd-a04d-4a46f714822e</ae:uuid>
        <ae:dbName>REPRODUCIBLE_</ae:dbName>
    </ae:id>
    <ae:value>
        <ae:displayValue>No</ae:displayValue>
        <ae:internalValue>0</ae:internalValue>
        <ae:uuid/>
    </ae:value>
</ae:extendedField>
<ae:extendedField>
    <ae:id xsi:type="ae:FieldIdentifier">
        <ae:displayName>Found in Version</ae:displayName>
        <ae:id>71</ae:id>
        <ae:uuid>08f99c88-6163-457e-b985-f2a70f935cc3</ae:uuid>
        <ae:dbName>FOUND_IN_VERSION</ae:dbName>
    </ae:id>
    <ae:value>
```

```

        <ae:displayValue/>
        <ae:internalValue/>
        <ae:uuid/>
    </ae:value>
</ae:extendedField>
<ae:extendedField>
    <ae:id xsi:type="ae:FieldIdentifier">
        <ae:displayName>Regression</ae:displayName>
        <ae:id>72</ae:id>
        <ae:uuid>d7790d89-c772-4b5c-a6e9-5e9ebb27ff5a</ae:uuid>
        <ae:dbName>REGRESSION_</ae:dbName>
    </ae:id>
    <ae:value>
        <ae:displayValue>(Not Checked)</ae:displayValue>
        <ae:internalValue>0</ae:internalValue>
        <ae:uuid/>
    </ae:value>
</ae:extendedField>
<ae:extendedField>
    <ae:id xsi:type="ae:FieldIdentifier">
        <ae:displayName>Priority</ae:displayName>
        <ae:id>73</ae:id>
        <ae:uuid>3141e866-7b20-4916-af87-9de705cc4b43</ae:uuid>
        <ae:dbName>PRIORITY</ae:dbName>
    </ae:id>
    <ae:value>
        <ae:displayValue/>
        <ae:internalValue>0</ae:internalValue>
        <ae:uuid/>
    </ae:value>
</ae:extendedField>
<ae:extendedField>
    <ae:id xsi:type="ae:FieldIdentifier">
        <ae:displayName>How Found</ae:displayName>
        <ae:id>74</ae:id>
        <ae:uuid>b999082f-ef27-47c9-890f-b4d80a3c4c23</ae:uuid>
        <ae:dbName>HOW_FOUND</ae:dbName>
    </ae:id>
    <ae:value>
        <ae:displayValue/>
        <ae:internalValue>0</ae:internalValue>
        <ae:uuid/>
    </ae:value>
</ae:extendedField>
<ae:extendedField>
    <ae:id xsi:type="ae:FieldIdentifier">
        <ae:displayName>Functional Area</ae:displayName>
        <ae:id>75</ae:id>
        <ae:uuid>165fe23f-ffea-4958-979c-f7d233b7df84</ae:uuid>
        <ae:dbName>FUNCTIONAL_AREA</ae:dbName>
    </ae:id>
    <ae:value>
        <ae:displayValue/>
        <ae:internalValue>0</ae:internalValue>
        <ae:uuid/>

```

```
</ae:value>
</ae:extendedField>
<ae:extendedField>
  <ae:id xsi:type="ae:FieldIdentifier">
    <ae:displayName>QA Reject Count</ae:displayName>
    <ae:id>76</ae:id>
    <ae:uuid>d2cb9f78-f29e-43c7-9fd2-b43c565fd5cc</ae:uuid>
    <ae:dbName>QA_REJECT_COUNT</ae:dbName>
  </ae:id>
  <ae:value>
    <ae:displayValue>0</ae:displayValue>
    <ae:internalValue>0</ae:internalValue>
    <ae:uuid/>
  </ae:value>
</ae:extendedField>
<ae:extendedField>
  <ae:id xsi:type="ae:FieldIdentifier">
    <ae:displayName>Secondary Owner</ae:displayName>
    <ae:id>78</ae:id>
    <ae:uuid>bcc74000-2e86-4813-84f2-c46c7229fef4</ae:uuid>
    <ae:dbName>SECONDARYOWNER</ae:dbName>
  </ae:id>
  <ae:value>
    <ae:displayValue>-2</ae:displayValue>
    <ae:internalValue>-2</ae:internalValue>
    <ae:uuid>0</ae:uuid>
  </ae:value>
</ae:extendedField>
<ae:extendedField>
  <ae:id xsi:type="ae:FieldIdentifier">
    <ae:displayName>Development Team</ae:displayName>
    <ae:id>79</ae:id>
    <ae:uuid>88a2d035-80ce-472c-8d32-90e7abb85633</ae:uuid>
    <ae:dbName>DEVELOPMENT_TEAM</ae:dbName>
  </ae:id>
  <ae:value>
    <ae:displayValue/>
    <ae:internalValue/>
    <ae:uuid/>
  </ae:value>
</ae:extendedField>
<ae:extendedField>
  <ae:id xsi:type="ae:FieldIdentifier">
    <ae:displayName>Replicated Issue URL</ae:displayName>
    <ae:id>156</ae:id>
    <ae:uuid>54acff35-aeb3-4245-8e8b-357d77b10b7c</ae:uuid>
    <ae:dbName>REPLICATED_ISSUE_URL</ae:dbName>
  </ae:id>
  <ae:value>
    <ae:displayValue/>
    <ae:internalValue/>
    <ae:uuid/>
  </ae:value>
</ae:extendedField>
<ae:extendedField>
```

```

    <ae:id xsi:type="ae:FieldIdentifier">
      <ae:displayName>Percentage Complete</ae:displayName>
      <ae:id>163</ae:id>
      <ae:uuid>f4f8cc08-5c1d-478c-9ac1-c9b5cb0692e8</ae:uuid>
      <ae:dbName>PERCENTAGE_COMPLETE</ae:dbName>
    </ae:id>
    <ae:value>
      <ae:displayValue>0</ae:displayValue>
      <ae:internalValue>0</ae:internalValue>
      <ae:uuid/>
    </ae:value>
  </ae:extendedField>
<ae:extendedField>
  <ae:id xsi:type="ae:FieldIdentifier">
    <ae:displayName>Close Date</ae:displayName>
    <ae:id>164</ae:id>
    <ae:uuid>68a285b6-92f7-46da-8c9c-7a376a5e83b2</ae:uuid>
    <ae:dbName>CLOSEDATE</ae:dbName>
  </ae:id>
  <ae:value>
    <ae:displayValue>2010-09-14T22:39:41+00:00</ae:displayValue>
    <ae:internalValue>2010-09-14T22:39:41+00:00</ae:internalValue>
    <ae:uuid/>
  </ae:value>
</ae:extendedField>
<ae:extendedField>
  <ae:id xsi:type="ae:FieldIdentifier">
    <ae:displayName>Last State Change Date</ae:displayName>
    <ae:id>165</ae:id>
    <ae:uuid>f2ef8a18-61f8-4272-8353-2f2796223e33</ae:uuid>
    <ae:dbName>LASTSTATECHANGEDATE</ae:dbName>
  </ae:id>
  <ae:value>
    <ae:displayValue>2010-09-14T22:39:41+00:00</ae:displayValue>
    <ae:internalValue>2010-09-14T22:39:41+00:00</ae:internalValue>
    <ae:uuid/>
  </ae:value>
</ae:extendedField>
<ae:extendedField>
  <ae:id xsi:type="ae:FieldIdentifier">
    <ae:displayName>P4Status</ae:displayName>
    <ae:id>166</ae:id>
    <ae:uuid>e0e50784-5cb8-424f-9e2b-3406ac1fd9b1</ae:uuid>
    <ae:dbName>P4STATUS</ae:dbName>
  </ae:id>
  <ae:value>
    <ae:displayValue>suspended</ae:displayValue>
    <ae:internalValue>2</ae:internalValue>
    <ae:uuid/>
  </ae:value>
</ae:extendedField>
<ae:extendedField>
  <ae:id xsi:type="ae:FieldIdentifier">
    <ae:displayName>Case URL</ae:displayName>
    <ae:id>167</ae:id>

```

```
        <ae:uuid>d8cf42f2-8a50-42f4-8798-f8b1a8391852</ae:uuid>
        <ae:dbName>CASE_URL</ae:dbName>
    </ae:id>
    <ae:value>
        <ae:displayValue/>
        <ae:internalValue/>
        <ae:uuid/>
    </ae:value>
</ae:extendedField>
<ae:extendedField>
    <ae:id xsi:type="ae:FieldIdentifier">
        <ae:displayName>BINARY TRINARY</ae:displayName>
        <ae:id>188</ae:id>
        <ae:uuid>bdcecc1a-d7b8-4571-819c-2fb5df8f5b5e</ae:uuid>
        <ae:dbName>BINARY_TRINARY</ae:dbName>
    </ae:id>
    <ae:value>
        <ae:displayValue>True</ae:displayValue>
        <ae:internalValue>0</ae:internalValue>
        <ae:uuid/>
    </ae:value>
</ae:extendedField>
<ae:extendedField>
    <ae:id xsi:type="ae:FieldIdentifier">
        <ae:displayName>DATE AND TIME</ae:displayName>
        <ae:id>189</ae:id>
        <ae:uuid>b0edf3b4-771b-488d-896d-cafe5a349de7</ae:uuid>
        <ae:dbName>DATE_AND_TIME</ae:dbName>
    </ae:id>
    <ae:value>
        <ae:displayValue>2010-05-01T06:00:00+00:00</ae:displayValue>
        <ae:internalValue>2010-05-01T06:00:00+00:00</ae:internalValue>
        <ae:uuid/>
    </ae:value>
</ae:extendedField>
<ae:extendedField>
    <ae:id xsi:type="ae:FieldIdentifier">
        <ae:displayName>DATE ONLY</ae:displayName>
        <ae:id>190</ae:id>
        <ae:uuid>b47a0270-95de-49db-b83f-0a1506c9bd63</ae:uuid>
        <ae:dbName>DATE_ONLY</ae:dbName>
    </ae:id>
    <ae:value>
        <ae:displayValue>2008-05-01</ae:displayValue>
        <ae:internalValue>2008-05-01T00:00:00+00:00</ae:internalValue>
        <ae:uuid/>
    </ae:value>
</ae:extendedField>
<ae:extendedField>
    <ae:id xsi:type="ae:FieldIdentifier">
        <ae:displayName>TIME OF DAY</ae:displayName>
        <ae:id>191</ae:id>
        <ae:uuid>2e7a47e0-546c-4842-af1d-81b0438ef398</ae:uuid>
        <ae:dbName>TIME_OF_DAY</ae:dbName>
    </ae:id>
```

```

    <ae:value>
      <ae:displayValue>17:36:39</ae:displayValue>
      <ae:internalValue>63399</ae:internalValue>
      <ae:uuid/>
    </ae:value>
  </ae:extendedField>
  <ae:extendedField>
    <ae:id xsi:type="ae:FieldIdentifier">
      <ae:displayName>ELAPSED TIME</ae:displayName>
      <ae:id>192</ae:id>
      <ae:uuid>9060a8db-ae8d-4dbe-b790-c33a3f0724fd</ae:uuid>
      <ae:dbName>ELAPSED_TIME</ae:dbName>
    </ae:id>
    <ae:value>
      <ae:displayValue>7 17:36:39</ae:displayValue>
      <ae:internalValue>668199</ae:internalValue>
      <ae:uuid/>
    </ae:value>
  </ae:extendedField>
  <ae:extendedField>
    <ae:id xsi:type="ae:FieldIdentifier">
      <ae:displayName>FOLDER</ae:displayName>
      <ae:id>193</ae:id>
      <ae:uuid>62830604-fa8b-48af-bb81-eba5c992d2f1</ae:uuid>
      <ae:dbName>FOLDER</ae:dbName>
    </ae:id>
    <ae:value>
      <ae:displayValue>InBox</ae:displayValue>
      <ae:internalValue>45</ae:internalValue>
      <ae:uuid>feb880b1-1156-4b3f-88d4-de41dfeeaf8a</ae:uuid>
    </ae:value>
  </ae:extendedField>
  <ae:extendedField>
    <ae:id xsi:type="ae:FieldIdentifier">
      <ae:displayName>MULTI GROUP</ae:displayName>
      <ae:id>194</ae:id>
      <ae:uuid>6d8bdef8-1136-4747-b88c-23c3e428e841</ae:uuid>
      <ae:dbName>MULTI_GROUP</ae:dbName>
    </ae:id>
    <ae:value>
      <ae:displayValue>IDM Team</ae:displayValue>
      <ae:internalValue>2</ae:internalValue>
      <ae:uuid>634aed4a-bc98-49d3-86f0-6095c2f7b9b6</ae:uuid>
    </ae:value>
    <ae:value>
      <ae:displayValue>IDM Administrator</ae:displayValue>
      <ae:internalValue>4</ae:internalValue>
      <ae:uuid>9156f029-6421-4f15-b84f-0ba67d593b3b</ae:uuid>
    </ae:value>
  </ae:extendedField>
  <ae:extendedField>
    <ae:id xsi:type="ae:FieldIdentifier">
      <ae:displayName>MULTI RELATIONAL</ae:displayName>
      <ae:id>195</ae:id>
      <ae:uuid>d5df0429-0bab-47ab-ba0b-8d5befce0bf0</ae:uuid>

```

```
    <ae:dbName>MULTI_RELATIONAL</ae:dbName>
  </ae:id>
  <ae:value>
    <ae:displayValue>App1 - test123</ae:displayValue>
    <ae:internalValue>1004:1</ae:internalValue>
    <ae:uuid>6c3741ff-5272-4659-9f0e-f7d8d6b0e616</ae:uuid>
  </ae:value>
  <ae:value>
    <ae:displayValue>App2 - field456</ae:displayValue>
    <ae:internalValue>1004:2</ae:internalValue>
    <ae:uuid>bcfa0ff7-0b0a-4bda-835c-19f7b3a3f9c8</ae:uuid>
  </ae:value>
</ae:extendedField>
<ae:extendedField>
  <ae:id xsi:type="ae:FieldIdentifier">
    <ae:displayName>MULTI_SELECTION</ae:displayName>
    <ae:id>196</ae:id>
    <ae:uuid>aa9eb88c-aa9a-477a-b3ea-92fd7bc4a257</ae:uuid>
    <ae:dbName>MULTI_SELECTION</ae:dbName>
  </ae:id>
  <ae:value>
    <ae:displayValue>Blue</ae:displayValue>
    <ae:internalValue>68</ae:internalValue>
    <ae:uuid>2b51dc08-f59c-4e20-a5a3-98527ce3172b</ae:uuid>
  </ae:value>
  <ae:value>
    <ae:displayValue>Green</ae:displayValue>
    <ae:internalValue>69</ae:internalValue>
    <ae:uuid>7f406b54-c629-42ec-be0a-52de6250f9da</ae:uuid>
  </ae:value>
  <ae:value>
    <ae:displayValue>Red</ae:displayValue>
    <ae:internalValue>70</ae:internalValue>
    <ae:uuid>823c84f1-7fab-436b-ad1f-8b344676d62e</ae:uuid>
  </ae:value>
</ae:extendedField>
<ae:extendedField>
  <ae:id xsi:type="ae:FieldIdentifier">
    <ae:displayName>MULTI_USER</ae:displayName>
    <ae:id>197</ae:id>
    <ae:uuid>67fba7ee-f62b-4603-8066-5d76f92c408a</ae:uuid>
    <ae:dbName>MULTI_USER</ae:dbName>
  </ae:id>
  <ae:value>
    <ae:displayValue>admin</ae:displayValue>
    <ae:internalValue>8</ae:internalValue>
    <ae:uuid>9f9146a3-a273-4411-8000-8396688b7554</ae:uuid>
  </ae:value>
  <ae:value>
    <ae:displayValue>bill</ae:displayValue>
    <ae:internalValue>27</ae:internalValue>
    <ae:uuid>225b4498-ce4b-4332-8775-6ccca90bebb4</ae:uuid>
  </ae:value>
</ae:extendedField>
<ae:extendedField>
```

```
<ae:id xsi:type="ae:FieldIdentifier">
  <ae:displayName>NUMERIC</ae:displayName>
  <ae:id>198</ae:id>
  <ae:uuid>e98538af-d1f1-4619-b4b4-90476529377c</ae:uuid>
  <ae:dbName>NUMERIC</ae:dbName>
</ae:id>
<ae:value>
  <ae:displayValue>55</ae:displayValue>
  <ae:internalValue>55</ae:internalValue>
  <ae:uuid/>
</ae:value>
</ae:extendedField>
<ae:extendedField>
  <ae:id xsi:type="ae:FieldIdentifier">
    <ae:displayName>SINGLE RELATIONAL</ae:displayName>
    <ae:id>199</ae:id>
    <ae:uuid>985c6a50-a523-4c81-975b-0336fed82b2a</ae:uuid>
    <ae:dbName>SINGLE_RELATIONAL</ae:dbName>
  </ae:id>
  <ae:value>
    <ae:displayValue>App1 - test123</ae:displayValue>
    <ae:internalValue>1004:1</ae:internalValue>
    <ae:uuid>6c3741ff-5272-4659-9f0e-f7d8d6b0e616</ae:uuid>
  </ae:value>
</ae:extendedField>
<ae:extendedField>
  <ae:id xsi:type="ae:FieldIdentifier">
    <ae:displayName>SINGLE SELECTION</ae:displayName>
    <ae:id>200</ae:id>
    <ae:uuid>6467afee-cd2d-431c-9d98-9059ab293bf7</ae:uuid>
    <ae:dbName>SINGLE_SELECTION</ae:dbName>
  </ae:id>
  <ae:value>
    <ae:displayValue>Yellow</ae:displayValue>
    <ae:internalValue>77</ae:internalValue>
    <ae:uuid>2d16b39d-1be9-44a0-805c-1f9074c8ac35</ae:uuid>
  </ae:value>
</ae:extendedField>
<ae:extendedField>
  <ae:id xsi:type="ae:FieldIdentifier">
    <ae:displayName>NUMERIC 2 FOR SUM</ae:displayName>
    <ae:id>201</ae:id>
    <ae:uuid>d4a2d02d-f52a-40a8-82de-edc8f26c9c87</ae:uuid>
    <ae:dbName>NUMERIC_2_FOR_SUM</ae:dbName>
  </ae:id>
  <ae:value>
    <ae:displayValue>10</ae:displayValue>
    <ae:internalValue>10</ae:internalValue>
    <ae:uuid/>
  </ae:value>
</ae:extendedField>
<ae:extendedField>
  <ae:id xsi:type="ae:FieldIdentifier">
    <ae:displayName>SUMMATION</ae:displayName>
    <ae:id>202</ae:id>
```

```
        <ae:uuid>2eb93809-e071-42eb-8ac4-af6cdf06fd43</ae:uuid>
        <ae:dbName>SUMMATION</ae:dbName>
    </ae:id>
    <ae:value>
        <ae:displayValue>0</ae:displayValue>
        <ae:internalValue>0</ae:internalValue>
        <ae:uuid/>
    </ae:value>
</ae:extendedField>
<ae:extendedField>
    <ae:id xsi:type="ae:FieldIdentifier">
        <ae:displayName>TEXT</ae:displayName>
        <ae:id>203</ae:id>
        <ae:uuid>84272380-923e-40e2-91e0-b505fe3081b3</ae:uuid>
        <ae:dbName>TEXT</ae:dbName>
    </ae:id>
    <ae:value>
        <ae:displayValue>This is some text.</ae:displayValue>
        <ae:internalValue>This is some text.</ae:internalValue>
        <ae:uuid/>
    </ae:value>
</ae:extendedField>
<ae:extendedField>
    <ae:id xsi:type="ae:FieldIdentifier">
        <ae:displayName>USER</ae:displayName>
        <ae:id>204</ae:id>
        <ae:uuid>61f40abc-8285-4485-96d6-ead4701b74a9</ae:uuid>
        <ae:dbName>USER</ae:dbName>
    </ae:id>
    <ae:value>
        <ae:displayValue>carmen</ae:displayValue>
        <ae:internalValue>10</ae:internalValue>
        <ae:uuid>c4a1c377-569c-469a-913b-9e1e94e88f26</ae:uuid>
    </ae:value>
</ae:extendedField>
<ae:note>
    <ae:id>103</ae:id>
    <ae:title>Note Title</ae:title>
    <ae:note>this is a note</ae:note>
    <ae:author xsi:type="ae:UserIdentifier">
        <ae:displayName>Administrator</ae:displayName>
        <ae:id>8</ae:id>
        <ae:uuid>9f9146a3-a273-4411-8000-8396688b7554</ae:uuid>
        <ae:loginId>admin</ae:loginId>
    </ae:author>
    <ae:modificationDateTime>2010-09-14T22:39:41Z</ae:modificationDateTime>
    <ae:accessType>ATTACHACCESS-RESTRICTED</ae:accessType>
</ae:note>
<ae:subtasks/>
</ae:item>
</ae:return>
</ae>CreatePrimaryItemsResponse>
```

URLAttachment

Description

The URLAttachment type holds information about a URL attached to a primary or auxiliary item. The URLAttachment type parameters are listed below.

Parameters

Name	Type	Description
id	integer	This is the internal TS_ID of the URL attachment from the TS_ATTACHMENTS table.
name	string	The name you give the URL attachment in SBM.
url	string	The actual URL itself.
showAsImage	boolean	This flag indicates whether or not graphic images in a URL are shown in the SBM User Workspace.
modificationDateTime	dateTime	The date and time when the URL attachment was last modified. See Supported Date/Time Formats [page 298] for more information.
accessType	Attachment-Access-Type [page 87]	Shows the access type for the URL attachment. The value is either DEFAULT, RESTRICTED, or UNRESTRICTED.
extendedData	ExtendedData [page 142]	Placeholder for future arguments.

Usage

You can add a URL to a primary or auxiliary item using the URLAttachment argument. URLs can be links to an external Web site or to a page within SBM.



Note: If you set `<urn:showAsImage>` to true, the image appears instead of a hyperlink when viewing the item in the SBM User Workspace. If you enter a URL to a graphic file such as `http://www.acme.com/image.gif`, the graphic appears. To show a URL graphically, an image file must be part of the URL. If you enter `www.acme.com` and select to show the URL as an image, an image does not appear because there is not an image file specified in the URL. Typically, this feature works for GIF and JPEG files, but can work in some browsers for PNG and BMP files as well. The file type that you can use for this feature depends on the file types your browser supports.



Tip: Internet Explorer is the only browser that supports URLs to files. Links to files do not work for users accessing the SBM User Workspace from other browser types.

XML

The following XML snippet shows the URLAttachment type in the <urn:urlAttachment> parameter of TTIItem.

```
<urn:urlAttachment>
  <urn:id>38</urn:id>
  <urn:name>test url</urn:name>
  <urn:url>http://www.acme.com/image.gif</urn:url>
  <urn:showAsImage>true</urn:showAsImage>
  <urn:modificationDateTime>2007-06-20T15:35:38-07:00
→</urn:modificationDateTime>
  <urn:accessType>ATTACHACCESS-DEFAULT</urn:accessType>
  <urn:extendedData>
    <urn:data>
      <urn:name></urn:name>
      <urn:value></urn:value>
    </urn:data>
  </urn:extendedData>
</urn:urlAttachment>
```

UserIdentifier

Description

The UserIdentifier type holds the identification information for a user. The UserIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 94] .	Generic identification extension base. Holds the displayName, id, and uuid for a user.
loginId	string	The unique login ID for the user.

Usage

The UserIdentifier is the identifier that can be used in Web service methods to uniquely identify a user. The UserIdentifier contains the generic information about a user (including the display name, ID, and UUID) in addition to the login ID for the user.



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows UserIdentifier as seen in a typical call.

```
<urn:user>
  <urn:displayName>Joe Manager</urn:displayName>
  <urn:id>2</urn:id>
  <urn:uuid>85a33f0b-9542-43fe-90c1-e152eeee777f</urn:uuid>
```

```
<urn:loginId>joe</urn:loginId>
</urn:user>
```

WorkflowIdentifier

Description

The WorkflowIdentifier type holds the identification information for a workflow. The WorkflowIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 94] .	Generic identification extension base. Holds the displayName, id, and uuid for a workflow.

Usage

The WorkflowIdentifier is the identifier that can be used in Web service methods to uniquely identify a workflow. The WorkflowIdentifier contains the generic information about a workflow (including the display name, ID, and UUID).



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows WorkflowIdentifier as seen in a typical call.

```
<urn:workflow>
  <urn:displayName>IDM</urn:displayName>
  <urn:id>2</urn:id>
  <urn:uuid>5296c4b1-4bab-48a9-83b3-1f633894ef33</urn:uuid>
</urn:workflow>
```

Arguments

This section provides detailed descriptions of SBM Application Web service arguments. The arguments are the request parameters that can be used by SBM Application Web service operations. The arguments listed here contain one or more parameters, which make up the data being sent to the Web service. The parameters listed in each argument are either simple or complex types themselves. If the type is complex, a link to further detail of that type will be provided in the **Type** column.

The following table lists all supported arguments in alphabetical order, followed by a brief description of each type. Select an argument to view detailed information including:

- **Description**
A brief description of the argument.
- **Parameters**

A table describing the types for each argument. Both simple and complex types are listed for each argument. For each complex type, you can click the type name for a detailed description.

- **Usage**

Any notes, additional details, and concerns regarding the argument are addressed here.

- **XML**

An example of the actual XML being sent is displayed here. The XML not only shows the argument and its respective elements, you can also see detailed examples of each element and how to format the expected data.

Arguments

Argument	Description
Auth [page 141]	Supplies credentials and optionally, a host name for licensing.
ExtendedData [page 142]	Placeholder argument for future argument elements.
ExtraValue [page 143]	Holds a name value pairing for future elements.
GetTransitionOptions [page 144]	Indicates the type of transitions to return.
MultipleOption [page 144]	Specifies whether a service should stop on failure and send an error message or continue processing.
MultipleOptions [page 145]	Holds the Options type and an enumeration to determine if a service should stop on failure and send an error message or continue processing.
MultipleResponseItemOptions [page 146]	Holds the Options type, as well as an enumeration to determine if a service should stop on failure and send an error message or continue processing. Also enables you to limit the data that is returned in a response.

Argument	Description
NameValue [page 148]	Holds a field and indicates how the field value is set.
NoteAttachmentContents [page 150]	Holds information, title, and content of a note.
Options [page 151]	Holds the ExtraValue type, which contains name value pairing for future arguments on certain calls.
ReportsFilter [page 151]	Allows you to filter reports.
ResponseItemOptions [page 154]	Holds the Options type and enables you to limit the data that is returned in a response.
SectionsOption [page 156]	Determines the section of an item to return.
Set-Value-By [page 158]	Indicates how the value in the NameValue type should be set on an update or create.
Set-Value-Method [page 159]	Indicates the operation that should be performed with the values in the NameValue type.
UserResponseOptions [page 162]	Holds the UserSingleResponseOptions type, as well as an enumeration to determine if a service should stop on failure and send an error message or continue processing.
UserSingleResponseOptions [page 164]	Holds the Options type and enables you to limit the data that is returned in a response.

Auth

Description

The Auth type supplies credentials and optionally, a host name for licensing. The Auth type parameters are listed below.

Parameters

Name	Type	Description
userId	string	The SBM user Login ID. If you are only specifying the host name, then <code>userId</code> is optional
password	string	The password for the user. If you are only specifying the host name, then <code>password</code> is optional.
hostname	string	The host name of the client.
loginAsUserId	string	User ID for the SBM login you wish to impersonate. If you are only specifying the host name, then <code>loginAsUserId</code> is optional.
extendedData	ExtendedData [page 142]	Placeholder for future arguments.

Usage

The Auth type allows credentials to be provided if not using WS-SECURITY or HTTP BASIC to pass the credentials. The hostname element is only needed in case you want to override the client's IP address for licensing purposes, forcing Serena License Manager to use a particular client host. If it's not provided, the code gets the client hostname from the socket.

XML

The following XML shows Auth as seen in a typical call.

```
<urn:auth>
  <urn:userId>admin</urn:userId>
  <urn:password>password</urn:password>
  <urn:hostname>serverName</urn:hostname>
  <urn:loginAsUserId></urn:loginAsUserId>
  <urn:extendedData></urn:extendedData>
</urn:auth>
```

ExtendedData

Description

The ExtendedData type is a placeholder for future argument or response elements. The ExtendedData type parameters are listed below.

Parameters

Name	Type	Description
data	ExtraValue [page 143]	Holds the name value pair for an additional argument or response element.

Usage

The ExtendedData type is simply a placeholder for future arguments that might be added to a given call. Responses have a similar placeholder for future response elements.

XML

The following XML shows ExtendedData as seen within the auth argument of a typical call.

```
<urn:auth>
  <urn:userId>admin</urn:userId>
  <urn:password>password</urn:password>
  <urn:hostname>serverName</urn:hostname>
  <urn:loginAsUserId></urn:loginAsUserId>
  <urn:extendedData>
    <urn:data>
      <urn:name></urn:name>
      <urn:value></urn:value>
    </urn:data>
  </urn:extendedData>
</urn:auth>
```

ExtraValue

Description

ExtraValue holds a name value pairing for future elements.

Parameters

Name	Type	Description
name	string	Holds the name of a future element.
value	string	Holds the value of a future element.

Usage

None.

XML

The following XML shows ExtraValue in the <urn:extraOption> element in a typical call.

```
<urn:options>
  <urn:extraOption>
    <urn:name></urn:name>
    <urn:value></urn:value>
```

```

    </urn:extraOption>
  </urn:options>

```

GetTransitionOptions

Description

GetTransitionOptions indicates the type of transitions that are returned for an item.

Parameters

Name	Type	Description
TRANSITIONS-ALL	string	Returns all transition types. This is the default value. If no value is specified, ALL is the assumed value.
TRANSITIONS-QUICK	string	Returns only quick transitions.

Usage

Use the `transitionOptions` argument to specify which types of transitions are returned for a given item.

XML

The following XML shows GetTransitionOptions in the `<urn:transitionOptions>` argument of the GetAvailableTransitions call.

```

<urn:GetAvailableTransitions>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:item>
    <urn:displayName></urn:displayName>
    <urn:id>25</urn:id>
    <urn:uuid></urn:uuid>
    <urn:tableId>1000</urn:tableId>
    <urn:tableIdItemId></urn:tableIdItemId>
    <urn:issueId></urn:issueId>
  </urn:item>
  <urn:transitionOptions>TRANSITIONS-ALL</urn:transitionOptions>
  <urn:attributeName></urn:attributeName>
</urn:GetAvailableTransitions>

```

MultipleOption

Description

The MultipleOption enumeration enables you to specify whether a Web service should stop on failure and send an error message or continue processing.

Parameters

Name	Type	Description
CONTINUE-ON-FAILURE	string	If a failure is encountered, continue processing the rest of the items in the call.
STOP-ON-FAILURE	string	If a failure is encountered, stop processing items and return an error.

Usage

The `MultipleOption` element enables you to specify whether the service should continue if an error is encountered, or stop and throw an error. If any failures occur, each successive error message is appended to the string that is returned. Multiple error messages are separated by a single newline. Failures do not result in a return before all items have been processed. See [TTItemHolder \[page 198\]](#) for an example of a `GetItems` call that returns an error, but continues to process the rest of the items.

XML

The following XML shows `MultipleOption` in the `<urn:multiOption>` element of a typical call.

```
<urn:options>
  <urn:extraOption>
    <urn:name></urn:name>
    <urn:value></urn:value>
  </urn:extraOption>
  <urn:multiOption>CONTINUE-ON-FAILURE</urn:multiOption>
  <urn:sections>SECTIONS-ALL</urn:sections>
  <urn:specifiedSections></urn:specifiedSections>
  </urn:limitedField>
</urn:options>
```

MultipleOptions

Description

`MultipleOptions` holds the `Options` type and enables you to specify whether a Web service should continue if an error is encountered, or stop and throw an error.

Parameters

Name	Type	Description
options	Extension. See Options [page 151] .	Options extension base. Holds name value pairing for future arguments on certain calls.
multiOption	MultipleOption [page 144]	Enumeration element that holds the option to continue or stop processing of items.

Usage

MultipleOptions holds the Options type and the MultipleOption type. You use the multiOption parameter to control service handling as follows:

- **multiOption** – Use the multiOption element to specify whether the service should continue if an error is encountered, or stop and return an error. If any failures occur, each successive error message is appended to the string that is returned. Multiple error messages are separated by a single newline. If you specify CONTINUE-ON-FAILURE, then failures do not result in a return before all items have been processed. For more information, see [MultipleOption \[page 144\]](#).

XML

The following XML shows MultipleOptions in the <urn:options> element in a typical call.

```
<urn:options>
  <urn:extraOption>
    <urn:name></urn:name>
    <urn:value></urn:value>
  </urn:extraOption>
  <urn:multiOption>CONTINUE-ON-FAILURE</urn:multiOption>
</urn:options>
```

MultipleResponseItemOptions

Description

MultipleResponseItemOptions holds the ResponseItemOptions type, as well as an enumeration to determine if a service should stop on failure and send an error message or continue processing.

Parameters

Name	Type	Description
ResponseItemOptions	Extension. See ResponseItemOptions [page 154] .	ResponseItemOptions extension base. Holds a name value pairing for future elements and enables you to limit the data returned in the response.
multiOption	MultipleOption [page 144]	Enumeration element that holds the option to continue processing or stop when an error is encountered.

Usage

The MultipleResponseItemOptions contains the same functionality as ResponseItemOptions, but it also enables you to specify an enumeration value that determines if the service should continue processing or stop when an error is encountered.

You use the following parameters in ResponseItemOptions and the multiOption parameter to control service handling and the amount of data that is returned in the response:

- **sections** and **specifiedSections** – Use these elements to specify which parts of an item should be returned in order to limit the data that is returned for a given item. The sections that are not specified are not included in the response. For example, if the items have a large number of item links, notes, URL and file attachments that don't need to be returned in the response, use these parameters to return only the sections of an item you want. For more information, see [SectionsOption \[page 156\]](#). In the sections parameter, specify one of the following enumeration options:
 - **SECTIONS-ALL** – Returns all sections. This is the default value for the sections argument. If no value is specified, ALL is the assumed value.
 - **SECTIONS-NONE** – No sections are returned (only the ItemIdentifier is returned). Note that the ItemIdentifier is always returned, regardless of the value or values specified in the sections parameter.
 - **SECTIONS-SPECIFIED** – Returns sections that you specify.

You can use a comma-separated list in specifiedSections to return only the sections of an item you want. For example:

```
<urn:specifiedSections>SECTION:FIXED,SECTION:EXTENDED</urn:specifiedSections>
```

This ensures that only the fixed and extended sections of an item are returned. Here are some of the possible sections you can specify:

- **SECTION:FIXED** – All parameters in TTIItem (from <urn:itemType> to <urn:url>) prior to the extendedField parameter are returned.
- **SECTION:EXTENDED** – Returns all of the extendedFields in TTIItem.
- **SECTION:ATTACHMENTS** – Returns all of the attachment sections of TTIItem. You can return the next four sections simply by specifying:

```
<urn:specifiedSections>SECTION:ATTACHMENTS</urn:specifiedSections>
```

- **SECTION:NOTES** – Returns all note sections of TTIItem.
- **SECTION:ITELINKS** – Returns all itemLink sections of TTIItem.
- **SECTION:URLATTACHMENTS** – Returns all urlAttachment sections of TTIItem.
- **SECTION:FILEATTACHMENTS** – Returns all fileAttachment sections of TTIItem.



Note: If you specify SECTION:NONE after other sections, those preceding sections will not be returned. For example, SECTION:FIXED,SECTION:NONE,SECTION:EXTENDED will only return the extendedField sections.

- **limitedField** – Use the limitedField element to use specific fields to limit the item data that is returned. For example, you can specify one or more fields to limit a service response to return only the fields that you want to return. In the event that limitedField contradicts the sections value, the sections specification takes precedence. The following XML shows how to limit return results using the limitedField option.

```

<urn:limitedField>
  <urn:displayName>Severity</urn:displayName>
  <urn:id>69</urn:id>
  <urn:uuid>f4eff572-2e29-4d7f-8478-8e9e16865c2c</urn:uuid>
  <urn:dbName>SEVERITY</urn:dbName>
</urn:limitedField>

```

- **multiOption** – Use the multiOption element to specify whether the service should continue if an error is encountered, or stop and return an error. If any failures occur, each successive error message is appended to the string that is returned. Multiple error messages are separated by a single newline. If you specify CONTINUE-ON-FAILURE, then failures do not result in a return before all items have been processed. For more information, see [MultipleOption \[page 144\]](#).

XML

The following XML shows MultipleResponseItemOptions in the <urn:options> element of a typical call.

```

<urn:options>
  <urn:extraOption>
    <urn:name></urn:name>
    <urn:value></urn:value>
  </urn:extraOption>
  <urn:sections>SECTIONS-SPECIFIED</urn:sections>
  <urn:specifiedSections>SECTION:EXTENDED</urn:specifiedSections>
  </urn:limitedField>
  <urn:multiOption>CONTINUE-ON-FAILURE</urn:multiOption>
</urn:options>

```

NameValue

Description

The NameValue type holds a field name with either a single value or a list of values and determines how the field value is set. The NameValue parameters are listed below.

Parameters

Name	Type	Description
id	FieldIdentifier [page 89]	The field identifier.
setValueBy	Set-Value-By [page 158]	Determines how to set the value (which value takes precedence).
setValueMethod	Set-Value-Method [page 159]	Use setValueMethod to append, remove, or replace a value.
value	FieldValue [page 89]	Holds information about a field value.

Usage

When specifying a value or values, you can use the display, internal, or UUID value. In addition, you can use the setValueBy element to specify which type of value it is, though it is not required. The setValueBy element is mainly used if you are passing in an empty value. Otherwise, SBM will determine which type of value is set by checking for a non-empty value.

To set an empty value, you must use the SET-VALUE-BY parameter that corresponds to the empty <value> parameter. For example, to set an empty value for a field using the internalValue parameter, you could specify:

```
<urn:setValueBy>INTERNAL-VALUE</urn:setValueBy>
<urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
<urn:value>
  <urn:internalValue></urn:internalValue>
```

Alternatively, to set an empty value for a field using the displayValue parameter, you could specify:

```
<urn:setValueBy>DISPLAY-VALUE</urn:setValueBy>
<urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
<urn:value>
  <urn:displayValue></urn:displayValue>
```



Note: To set an empty value for a single or Multi-Relational field, you must use INTERNAL-VALUE in SetValueBy and empty tags in the internalValue parameter.

XML

The following XML shows the NameValue type as seen in a typical call.

```
<urn:extendedField>
  <urn:id>
    <urn:displayName>Severity</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName>SEVERITY</urn:dbName>
  </urn:id>
  <urn:setValueBy>DISPLAY-VALUE</urn:setValueBy>
  <urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
  <urn:value>
    <urn:displayValue>Critical</urn:displayValue>
    <urn:internalValue></urn:internalValue>
    <urn:uuid></urn:uuid>
  </urn:value>
</urn:extendedField>
```

The following XML shows the NameValue type with multiple values as seen in a typical call.

```
<urn:extendedField>
  <urn:id>
    <urn:displayName>MULTI USER</urn:displayName>
    <urn:id>178</urn:id>
```

```

    <urn:uuid>f62c6b63-2531-441a-9fff-9cd471bc61ca</urn:uuid>
    <urn:dbName>MULT_USER</urn:dbName>
  </urn:id>
  <urn:setValueBy>PRECEDENCE-VALUE</urn:setValueBy>
  <urn:setValueMethod>APPEND-VALUES</urn:setValueMethod>
    <urn:value>
      <urn:displayValue>admin</urn:displayValue>
      <urn:internalValue>1</urn:internalValue>
      <urn:uuid>d2d60592-656e-4103-a20d-f12da9305fe4</urn:uuid>
    </urn:value>
    <urn:value>
      <urn:displayValue>bill</urn:displayValue>
      <urn:internalValue>10</urn:internalValue>
      <urn:uuid>7130c9c3-abb6-41f5-bd7a-30c40f47b824</urn:uuid>
    </urn:value>
    <urn:value>
      <urn:displayValue>carmen</urn:displayValue>
      <urn:internalValue>11</urn:internalValue>
      <urn:uuid>9d71b19e-9b72-4731-bec3-3eba938da0de</urn:uuid>
    </urn:value>
  </urn:extendedField>

```

Using the example above, if this payload was sent with TransitionItems, the call would append admin, bill, and carmen to the current selections that exist in the "MULTI USER" field.

NoteAttachmentContents

Description

The NoteAttachmentContents type holds the actual contents of a note. The NoteAttachmentContents type parameters are listed below.

Parameters

Name	Type	Description
time	integer	The time at which the note was created.
title	string	The title of the note.
body	string	Holds the contents of the body of the note.
accessType	Attachment-Access-Type [page 87]	Shows the access type for the note. The value is either DEFAULT, RESTRICTED, or UNRESTRICTED.
extendedData	ExtendedData [page 142]	Placeholder for future arguments.

Usage

NoteAttachmentContents is used to describe a note that is attached to an item. You use NoteAttachmentsContents in the CreateNoteAttachment call to add a note to an existing item. For more information, see [CreateNoteAttachment \[page 32\]](#).

XML

The following XML snippet shows NoteAttachmentContents in the `<urn:noteContents>` argument of a CreateNoteAttachment call.

```
<urn:noteContents>
  <urn:time></urn:time>
  <urn:title>Note Title</urn:title>
  <urn:body>This is a note.</urn:body>
  <urn:accessType>ATTACHACCESS-DEFAULT</urn:accessType>
</urn:noteContents>
```

Options

Description

Options holds the ExtraValue type, which contains a name value pairing for future arguments on certain calls.

Parameters

Name	Type	Description
extraOption	ExtraValue [page 143]	Holds a name value pairing for future elements.

Usage

None.

XML

The following XML shows Options as seen in a typical call.

```
<urn:options>
  <urn:extraOption>
    <urn:name></urn:name>
    <urn:value></urn:value>
  </urn:extraOption>
</urn:options>
```

ReportsFilter

Description

The ReportsFilter type allows you to filter the reports returned via the GetReports call. The ReportsFilter type parameters are listed below.

Parameters

Name	Type	Description
solution	SolutionIdentifier [page 106]	Holds the identification information for a solution.
project	ProjectIdentifier [page 100]	Holds the identification information for a project.
table	TableIdentifier [page 109]	Holds the identification information for a table.
author	UserIdentifier [page 138]	Holds the identification information for a user.
reportType	ReportType [page 104]	An enumeration that indicates the type of report. The default value is 1 for LISTING.
reportCategory	ReportCategory [page 102]	A broader enumeration that limits the response based on the category of report (built-in reports, application reports, reports you authored). The default value is "ALL."
reportAccessLevel	ReportAccessLevel [page 101]	An enumeration that limits the response based on the report's access level (PRIVATE, GUEST, USER, or MANAGER).
report	ReportIdentifier [page 104]	Holds the identification information for a report.
searchByName	string	Specifies the name of the report as a search parameter.
includeSubProjects	boolean	This flag indicates whether or not to include reports that are based on sub-projects of the project that is specified. The default value is "false."
createdDateFrom	dateTime	Filter reports created after this date.
createdDateTo	dateTime	Filter reports created before this date.
extendedData	ExtendedData [page 142]	Placeholder for future arguments.

Usage

You can filter the reports returned in [GetReports \[page 60\]](#) using ReportsFilter. For example, you can specify "joe" in the displayName of the author element to only return reports that Joe created. Keep in mind that projects do not exist for auxiliary tables so the ProjectIdentifier should be left empty when searching for reports against auxiliary tables.



Tip: You can easily find the proper solution's uniqueName by looking at the Internal Name of the application in SBM Composer.



Note: You can treat the searchByName as a search parameter and enter only part of the report name. In the example below, "All" is sent in searchByName to filter on reports that include the word "All" in the report title. See the XML in [RunReportResult \[page 185\]](#) for an example of the response.

XML

The following XML snippet shows ReportsFilter as seen in a typical call.

```
<urn:reportsFilter>
  <urn:solution>
    <urn:displayName></urn:displayName>
    <urn:id>1</urn:id>
    <urn:uuid></urn:uuid>
    <urn:uniqueName>BASE_ISSUE_DEFECT_TRACKING</urn:uniqueName>
    <urn:tabName></urn:tabName>
  </urn:solution>
  <urn:project>
    <urn:displayName></urn:displayName>
    <urn:id>4</urn:id>
    <urn:uuid></urn:uuid>
    <urn:fullyQualifiedName></urn:fullyQualifiedName>
  </urn:project>
  <urn:table>
    <urn:displayName></urn:displayName>
    <urn:id>1000</urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName></urn:dbName>
  </urn:table>
  <urn:author>
    <urn:displayName>Joe</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:loginId></urn:loginId>
  </urn:author>
  <urn:reportType>LISTING</urn:reportType>
  <urn:reportCategory>ALL</urn:reportCategory>
  <urn:reportAccessLevel>USER</urn:reportAccessLevel>
  <urn:report>
    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
  </urn:report>
  <urn:searchByName>ALL</urn:searchByName>
  <urn:includeSubProjects>true</urn:includeSubProjects>
  <urn:createdDateFrom>2007-06-20T15:35:38-07:00</urn:createdDateFrom>
  <urn:createdDateTo>2007-07-20T15:35:38-07:00</urn:createdDateTo>
  <urn:extendedData>
```

```

    <urn:data>
      <urn:name></urn:name>
      <urn:value></urn:value>
    </urn:data>
  </urn:extendedData>
</urn:reportsFilter>

```

ResponseItemOptions

Description

ResponseItemOptions holds the Options type and it enables you to limit the data that is returned in a response.

Parameters

Name	Type	Description
options	Extension. See Options [page 151] .	Options extension base. Holds a name value pairing for future elements.
sections	SectionsOption [page 156]	Enumeration element that controls the sections of an item that should be returned.
specifiedSections	string	If SECTIONS-SPECIFIED is used in the sections element above, enter the specified section or sections here. The available options are described below in the Usage section.
limitedField	FieldIdentifier [page 89]	Enables you to limit the return results based on one or more fields that you identify.

Usage

The ResponseItemOptions type is used to limit the return results based on item sections or specific fields that you identify.

You use the following parameters to control the amount of data that is returned in the response:

- **sections** and **specifiedSections** – Use these elements to specify which parts of an item should be returned in order to limit the data that is returned for a given item. The sections that are not specified are not included in the response. For example, if the items have a large number of item links, notes, URL and file attachments that don't need to be returned in the response, use these parameters to return only the sections of an item you want. For more information, see [SectionsOption \[page 156\]](#). In the sections parameter, specify one of the following enumeration options:
 - **SECTIONS-ALL** – Returns all sections. This is the default value for the sections argument. If no value is specified, ALL is the assumed value.

- **SECTIONS-NONE** – No sections are returned (only the ItemIdentifier is returned). Note that the ItemIdentifier is always returned, regardless of the value or values specified in the sections parameter.
- **SECTIONS-SPECIFIED** – Returns sections that you specify.

You can use a comma-separated list in specifiedSections to return only the sections of an item you want. For example:

```
<urn:specifiedSections>SECTION:FIXED,SECTION:EXTENDED</urn:specifiedSections>
```

This ensures that only the fixed and extended sections of an item are returned. Here are some of the possible sections you can specify:

- **SECTION:FIXED** – All parameters in TItem (from <urn:itemType> to <urn:url>) prior to the extendedField parameter are returned.
- **SECTION:EXTENDED** – Returns all of the extendedFields in TItem.
- **SECTION:ATTACHMENTS** – Returns all of the attachment sections of TItem. You can return the next four sections simply by specifying:

```
<urn:specifiedSections>SECTION:ATTACHMENTS</urn:specifiedSections>
```

- **SECTION:NOTES** – Returns all note sections of TItem.
- **SECTION:ITELINKS** – Returns all itemLink sections of TItem.
- **SECTION:URLATTACHMENTS** – Returns all urlAttachment sections of TItem.
- **SECTION:FILEATTACHMENTS** – Returns all fileAttachment sections of TItem.



Note: If you specify SECTION:NONE after other sections, those preceding sections will not be returned. For example, SECTION:FIXED,SECTION:NONE,SECTION:EXTENDED will only return the extendedField sections.

- **limitedField** – Use the limitedField element to use specific fields to limit the item data that is returned. For example, you can specify one or more fields to limit a service response to return only the fields that you want to return. In the event that limitedField contradicts the sections value, the sections specification takes precedence. The following XML shows how to limit return results using the limitedField option.

```
<urn:limitedField>
  <urn:displayName>Severity</urn:displayName>
  <urn:id>69</urn:id>
  <urn:uuid>f4eff572-2e29-4d7f-8478-8e9e16865c2c</urn:uuid>
  <urn:dbName>SEVERITY</urn:dbName>
</urn:limitedField>
```

XML

The following XML shows ResponseItemOptions in the <urn:options> element of a typical call.

```

<urn:options>
  <urn:extraOption>
    <urn:name></urn:name>
    <urn:value></urn:value>
  </urn:extraOption>
  <urn:sections>SECTIONS-SPECIFIED</urn:sections>
  <urn:specifiedSections>SECTION:EXTENDED</urn:specifiedSections>
  <urn:limitedField>
</urn:options>

```

SectionsOption

Description

SectionsOption indicates the sections that should be returned for a record.

Parameters

Name	Type	Description
SECTIONS-ALL	string	Returns all sections. This is the default value for the sections argument. If no value is specified, ALL is the assumed value.
SECTIONS-NONE	string	No sections are returned (only the Identifier is returned). Note that the Identifier element is always returned, regardless of the value or values specified in the sections parameter.
SECTIONS-SPECIFIED	string	Returns sections that you specify.

Usage

Use the `sections` and `specifiedSections` arguments to specify which parts of a record should be returned in order to limit the total amount of data that is returned. The sections that are not specified are not included in the response. For example, if the items that you want to return have a large number of item links, notes, URL and file attachments that don't need to be returned in the response, use these parameters to return only the sections of an item you want. If you only need basic user information in the `GetUsers` response, use the `specifiedSections` parameter to return only the `STANDARD` section. If you need basic user information and group information, specify `STANDARD` and `GROUPS`.

In the `sections` parameter, specify one of the following enumeration options:

- **SECTIONS-ALL** – Returns all sections. This is the default value for the sections argument. If no value is specified, ALL is the assumed value. For example:

```

<urn:sections>SECTIONS-ALL</urn:sections>
<urn:specifiedSections></urn:specifiedSections>

```

- **SECTIONS-NONE** – No sections are returned (only the Identifier element is returned). For example:

```
<urn:sections>SECTIONS-NONE</urn:sections>  
<urn:specifiedSections></urn:specifiedSections>
```



Note: The Identifier is always returned, regardless of the value or values specified in the sections parameter.

- **SECTIONS-SPECIFIED** – Returns sections that you specify. For example:

```
<urn:sections>SECTIONS-SPECIFIED</urn:sections>  
<urn:specifiedSections>SECTION:EXTENDED</urn:specifiedSections>
```

You can also use a comma-separated list in specifiedSections to return only the sections of a record that you want. For example:

```
<urn:specifiedSections>SECTION:FIXED,SECTION:EXTENDED</urn:specifiedSections>
```

This ensures that only the fixed and extended sections of an item are returned. Below are the sections that you can specify for items:

- **SECTION:FIXED** – All parameters in TTIItem (from <urn:itemType> to <urn:url>) prior to the extendedField parameter are returned.
- **SECTION:EXTENDED** – Returns all of the extendedFields in TTIItem.
- **SECTION:ATTACHMENTS** – Returns all of the attachment sections of TTIItem. You can return the next four sections simply by specifying:

```
<urn:specifiedSections>SECTION:ATTACHMENTS</urn:specifiedSections>
```

- **SECTION:NOTES** – Returns all note sections of TTIItem.
- **SECTION:ITEMLINKS** – Returns all itemLink sections of TTIItem.
- **SECTION:URLATTACHMENTS** – Returns all urlAttachment sections of TTIItem.
- **SECTION:FILEATTACHMENTS** – Returns all fileAttachment sections of TTIItem.

Below are the possible sections that you can specify for user records:

- **SECTION:STANDARD** – Returns parameters from UserInfo such as the accessType, e-mail address, contact, date and time information, phone number, locale, and the various display preferences.
- **SECTION:GROUPS** – Returns one or more GroupIdentifiers for each group to which the user belongs.
- **SECTION:SOLUTIONS** – Returns the preferredSolution and solutionData elements.



Note: If you specify SECTION:NONE after other sections, those preceding sections will not be returned. For example, SECTION:FIXED,SECTION:NONE,SECTION:EXTENDED will only return the extendedField sections. For the GetUsers call, SECTION:STANDARD,SECTION:NONE,SECTION:GROUPS will only return the groups section.

XML

The following XML shows SectionsOption in the `<urn:sections>` element of a typical call.

```
<urn:options>
  <urn:extraOption>
    <urn:name></urn:name>
    <urn:value></urn:value>
  </urn:extraOption>
  <urn:multiOption>CONTINUE-ON-FAILURE</urn:multiOption>
  <urn:sections>SECTIONS-SPECIFIED</urn:sections>
  <urn:specifiedSections>SECTION:FIXED</urn:specifiedSections>
  <urn:limitedField>
</urn:options>
```

Set-Value-By

Description

Set-Value-By indicates how the value in the NameValue type should be set.

Parameters

Name	Type	Description
PRECEDENCE-VALUE	string	If the PRECEDENCE-VALUE is chosen, then the value will be set by looking at the values supplied in order of precedence.
INTERNAL-VALUE	string	Set the value using the internal value
UUID-VALUE	string	Set the value using the UUID value.
DISPLAY-VALUE	string	Set the value using the display value

Usage

If the PRECEDENCE-VALUE is chosen or the Set-Value-By parameter is omitted, then the value will be set by looking at the values supplied in order of precedence. The order is internal, uuid, followed by display. For example, if the internal value is present, it will be used. If the internal value is not present, the uuid value will be used. If neither the internal or uuid values are specified, the display value will be used.

You can use the `setValueBy` element to specify which type of value it is, though it is not required. The `setValueBy` element is mainly used if you are passing in an empty value. Otherwise, SBM will determine which type of value is set by checking for a non-empty value.

To set an empty value, you must use the SET-VALUE-BY parameter that corresponds to the empty `<value>` parameter. For example, to set an empty value for a field using the `internalValue` parameter, you could specify:

```
<urn:setValueBy>INTERNAL-VALUE</urn:setValueBy>
<urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
<urn:value>
  <urn:internalValue></urn:internalValue>
```

Alternatively, to set an empty value for a field using the `displayValue` parameter, you could specify:

```
<urn:setValueBy>DISPLAY-VALUE</urn:setValueBy>
<urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
<urn:value>
  <urn:displayValue></urn:displayValue>
```



Note: To set an empty value for a Single or Multi-Relational field, you must use `INTERNAL-VALUE` in `SetValueBy` and empty tags in the `internalValue` parameter.

XML

The following XML shows `Set-Value-By` in the `<urn:extendedField>` element in a typical call.

```
<urn:extendedField>
  <urn:id>
    <urn:displayName>Severity</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName>SEVERITY</urn:dbName>
  </urn:id>
  <urn:setValueBy>PRECEDENCE-VALUE</urn:setValueBy>
  <urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
  <urn:value>
    <urn:displayValue>High</urn:displayValue>
    <urn:internalValue>122</urn:internalValue>
    <urn:uuid>4ad0961d-30dc-4198-8167-5224dcb6c065</urn:uuid>
  </urn:value>
</urn:extendedField>
```

Set-Value-Method

Description

`Set-Value-Method` indicates how to process values in a text field or any field that contains multiple values such as Multi-Selection, Multi-Relational, Multi-Group, or Multi-User.

Parameters

Name	Type	Description
REPLACE-VALUES	string	If you choose <code>REPLACE-VALUES</code> , <code>Set-Value-Method</code> replaces the current value. Applies to all fields. If a <code>Set-Value-Method</code> is not specified, <code>REPLACE-VALUES</code> is assumed by default.

Name	Type	Description
APPEND-VALUES	string	If you choose APPEND-VALUES, Set-Value-Method appends the value to the previous entry. Applies to Text and all multi-value fields.
REMOVE-VALUES	string	If you choose REMOVE-VALUES, Set-Value-Method removes the value. Only applies to multi-value fields.

Usage

You can use Set-Value-Method to specify how text fields or any field that holds multiple values should process the value or values that are passed in the `setValueBy` argument. For example, you could use this argument to *replace* one or more values in a Multi-Relational field, *append* new text to a Journal field, or *remove* specified values from a Multi-Selection list. Specifying a Set-Value-Method parameter is useful if you plan to have multiple Web service calls updating the same field on the same item. By using Set-Value-Method, field values can be managed dynamically; there is no need to ensure that the last update made contains the definitive selections for a given field. Otherwise, if Set-Value-Method is not supplied, the values will be replaced or an empty value will be set, depending on how the value or values are sent.

XML

The following XML snippets show each Set-Value-Method in the `<urn:extendedField>` element, with a brief description of the results.

```
<urn:extendedField>
  <urn:id>
    <urn:displayName>Text Field</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName>TEXT_FIELD</urn:dbName>
  </urn:id>
  <urn:setValueBy>INTERNAL-VALUE</urn:setValueBy>
  <urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
  <urn:value>
    <urn:displayValue>11</urn:displayValue>
    <urn:internalValue>11</urn:internalValue>
  </urn:value>
</urn:extendedField>
<urn:extendedField>
  <urn:id>
    <urn:displayName>Text Field</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName>TEXT_FIELD</urn:dbName>
  </urn:id>
  <urn:setValueBy>INTERNAL-VALUE</urn:setValueBy>
  <urn:setValueMethod>APPEND-VALUES</urn:setValueMethod>
  <urn:value>
    <urn:displayValue>22</urn:displayValue>
    <urn:internalValue>22</urn:internalValue>
  </urn:value>
</urn:extendedField>
```

```
</urn:value>
</urn:extendedField>
```

This call would replace the current text field value with "11" and then append "22" to that value:

```
11 22
```

```
<urn:extendedField>
  <urn:id>
    <urn:displayName>WCRS</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName>WCRS</urn:dbName>
  </urn:id>
  <urn:setValueBy>INTERNAL-VALUE</urn:setValueBy>
  <urn:setValueMethod>REPLACE-VALUES</urn:setValueMethod>
  <urn:value>
    <urn:displayValue>ZMF00026: Release Package - Construction</urn:displayValue>
    <urn:internalValue>1015:10</urn:internalValue>
    <urn:uuid>af93aad0-8245-4e49-ac91-574725f7041f</urn:uuid>
  </urn:value>
  <urn:value>
    <urn:displayValue>ZMF00028: Release Package - Construction</urn:displayValue>
    <urn:internalValue>1015:11</urn:internalValue>
    <urn:uuid>e16d9550-c2bc-41b9-99c6-f716cb5a7022</urn:uuid>
  </urn:value>
</urn:extendedField>
```

This call would replace the current selections in the Multi-Relational WCRS field with two new records (10 and 11) from table 1015.

```
<urn:extendedField>
  <urn:id>
    <urn:displayName>WCRS</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName>WCRS</urn:dbName>
  </urn:id>
  <urn:setValueBy>INTERNAL-VALUE</urn:setValueBy>
  <urn:setValueMethod>APPEND-VALUES</urn:setValueMethod>
  <urn:value>
    <urn:displayValue>ZMF00032: Release Unit - Construction</urn:displayValue>
    <urn:internalValue>1015:12</urn:internalValue>
    <urn:uuid>15ce4bc2-347e-41bb-acaf-f3dd434a9633</urn:uuid>
  </urn:value>
  <urn:value>
    <urn:displayValue>ZMF00033: Release Unit - Construction</urn:displayValue>
    <urn:internalValue>1015:13</urn:internalValue>
    <urn:uuid>5e8d1cb0-b177-4aa5-ae8c-bd169e31f6da</urn:uuid>
  </urn:value>
</urn:extendedField>
```

This call would add two additional selections to the Multi-Relational WCRS field. Records 12 and 13 from table 1015 would now be selected in this field along with the previous selections that were made in the field.

```
<urn:extendedField>
  <urn:id>
    <urn:displayName>WCRS</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName>WCRS</urn:dbName>
  </urn:id>
  <urn:setValueBy>INTERNAL-VALUE</urn:setValueBy>
  <urn:setValueMethod>REMOVE-VALUES</urn:setValueMethod>
  <urn:value>
    <urn:displayValue>ZMF00032: Release Unit - Construction</urn:displayValue>
    <urn:internalValue>1015:12</urn:internalValue>
    <urn:uuid>15ce4bc2-347e-41bb-acaf-f3dd434a9633</urn:uuid>
  </urn:value>
  <urn:value>
    <urn:displayValue>ZMF00026: Release Package - Construction</urn:displayValue>
    <urn:internalValue>1015:10</urn:internalValue>
    <urn:uuid>af93aad0-8245-4e49-ac91-574725f7041f</urn:uuid>
  </urn:value>
</urn:extendedField>
```

This call would simply remove records 10 and 12 from the Multi-Relational WCRS field.

```
<urn:extendedField>
  <urn:id>
    <urn:displayName>Colors</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:dbName>COLORS</urn:dbName>
  </urn:id>
  <urn:id>178</urn:id>
  <urn:setValueBy>INTERNAL-VALUE</urn:setValueBy>
  <urn:setValueMethod>REMOVE-VALUES</urn:setValueMethod>
  <urn:value>
    <urn:displayValue>Red</urn:displayValue>
    <urn:internalValue>5</urn:internalValue>
  </urn:value>
  <urn:value>
    <urn:displayValue>Green</urn:displayValue>
    <urn:internalValue>6</urn:internalValue>
  </urn:value>
</urn:extendedField>
```

This call would remove the "Red" and "Green" selections from the Multi-Selection "Colors" field.

UserResponseOptions

Description

UserResponseOptions holds the UserSingleResponseOptions type, as well as an enumeration to determine if a service should stop on failure and send an error message or continue processing.

Parameters

Name	Type	Description
UserSingleResponseOptions	Extension. See UserSingleResponseOptions [page 164] .	UserSingleResponseOptions extension base. Holds a name value pairing for future elements and enables you to limit the data returned in the response.
multiOption	MultipleOption [page 144]	Enumeration element that holds the option to continue processing or stop when an error is encountered.

Usage

The UserResponseOptions contains the same functionality as UserSingleResponseOptions, but it also enables you to specify an enumeration value that determines if the service should continue processing or stop when an error is encountered.

You use the following parameters in UserSingleResponseOptions to control service handling and the amount of data that is returned in the response:

- **sections** and **specifiedSections** – Use these elements to specify which parts of a user record should be returned in order to limit the amount of data that is returned. The sections that are not specified are not included in the response. For example, if you only need basic user information in the response, use the specifiedSections parameter to return only the STANDARD section. If you need basic user information and group information, specify STANDARD and GROUPS. For more information, see [SectionsOption \[page 156\]](#).

In the sections parameter, specify one of the following enumeration options:

- **SECTIONS-ALL** – Returns all sections. This is the default value for the sections argument. If no value is specified, ALL is the assumed value.
- **SECTIONS-NONE** – No sections are returned (only the UserIdentifier is returned). Note that the UserIdentifier is always returned, regardless of the value or values specified in the sections parameter.
- **SECTIONS-SPECIFIED** – Returns sections that you specify.

You can use a comma-separated list in specifiedSections to return only the sections of a user record that you want. For example:

```
<urn:specifiedSections>SECTION:STANDARD,SECTION:GROUPS</urn:specifiedSections>
```

This ensures that only the standard and groups sections of a user record are returned. Here are some of the possible sections you can specify in GetUsers:

- **SECTION:STANDARD** – Returns parameters from UserInfo such as the accessType, e-mail address, contact, date and time information, phone number, locale, and the various display preferences.
- **SECTION:GROUPS** – Returns one or more GroupIdentifiers for each group to which the user belongs.
- **SECTION:SOLUTIONS** – Returns the preferredSolution and solutionData elements.



Note: If you specify SECTION:NONE after other sections, those preceding sections will not be returned. For example, SECTION:STANDARD,SECTION:NONE,SECTION:GROUPS will only return the groups section.

- **multiOption** – Use the multiOption element to specify whether the service should continue if an error is encountered or stop and return an error. If any failures occur, each successive error message is appended to the string that is returned. Multiple error messages are separated by a single newline. If you specify CONTINUE-ON-FAILURE, then failures do not result in a return before all records have been processed. For more information, see [MultipleOption \[page 144\]](#).

XML

The following XML shows UserResponseOptions in the <urn:options> element of the GetUsers call.

```
<urn:options>
  <urn:extraOption>
    <urn:name></urn:name>
    <urn:value></urn:value>
  </urn:extraOption>
  <urn:sections>SECTIONS-SPECIFIED</urn:sections>
  <urn:specifiedSections>SECTION:STANDARD</urn:specifiedSections>
  <urn:multiOption>CONTINUE-ON-FAILURE</urn:multiOption>
</urn:options>
```

UserSingleResponseOptions

Description

UserSingleResponseOptions holds the Options type and enables you to limit the data that is returned in a response.

Parameters

Name	Type	Description
options	Extension. See Options [page 151] .	Options extension base. Holds a name value pairing for future elements.

Name	Type	Description
sections	SectionsOption [page 156]	Enumeration element that controls the sections of a user record that should be returned.
specifiedSections	string	If SECTIONS-SPECIFIED is used in the sections element above, enter the specified section or sections here. The available options are described below in the Usage section.

Usage

The UserSingleResponseOptions type is used to limit the return results based on specific sections that you identify.

You use the following parameters to control the amount of data that is returned in the response:

- **sections** and **specifiedSections** – Use these elements to specify which parts of a user record should be returned in order to limit the amount of data that is returned. The sections that are not specified are not included in the response. For example, if you only need basic user information in the response, use the specifiedSections parameter to return only the STANDARD section. If you need basic user information and group information, specify STANDARD and GROUPS. For more information, see [SectionsOption \[page 156\]](#).

In the sections parameter, specify one of the following enumeration options:

- **SECTIONS-ALL** – Returns all sections. This is the default value for the sections argument. If no value is specified, ALL is the assumed value.
- **SECTIONS-NONE** – No sections are returned (only the UserIdentifier is returned). Note that the UserIdentifier is always returned, regardless of the value or values specified in the sections parameter.
- **SECTIONS-SPECIFIED** – Returns sections that you specify.

You can use a comma-separated list in specifiedSections to return only the sections of a user record that you want. For example:

```
<urn:specifiedSections>SECTION:STANDARD,SECTION:GROUPS</urn:specifiedSections>
```

This ensures that only the standard and groups sections of a user record are returned. Here are some of the possible sections you can specify in GetUsers:

- **SECTION:STANDARD** – Returns parameters from UserInfo such as the accessType, e-mail address, contact, date and time information, phone number, locale, and the various display preferences.
- **SECTION:GROUPS** – Returns one or more GroupIdentifiers for each group to which the user belongs.
- **SECTION:SOLUTIONS** – Returns the preferredSolution and solutionData elements.



Note: If you specify SECTION:NONE after other sections, those preceding sections will not be returned. For example, SECTION:STANDARD,SECTION:NONE,SECTION:GROUPS will only return the groups section.

XML

The following XML shows UserSingleResponseOptions in the <urn:options> element of the GetUsers call.

```
<urn:options>
  <urn:extraOption>
    <urn:name></urn:name>
    <urn:value></urn:value>
  </urn:extraOption>
  <urn:sections>SECTIONS-SPECIFIED</urn:sections>
  <urn:specifiedSections>SECTION:STANDARD</urn:specifiedSections>
</urn:options>
```

Responses

This section provides detailed descriptions of SBM Application Web service responses. The responses are the data elements that are returned from SBM Application Web service operations. The responses listed here contain one or more parameters, which make up the data being sent back from the Web service. The parameters listed in each argument are either simple or complex types themselves. If the type is complex, a link to further detail of that type will be provided in the **Type** column.

The following table lists all supported responses in alphabetical order, followed by a brief description of each type. Select a response to view detailed information including:

- **Description**

A brief description of the response.

- **Parameters**

A table describing the types for each response. Both simple and complex types are listed for each response. For each complex type, you can click the type name for a detailed description.

- **Usage**

Any notes, additional details, and concerns regarding the response are addressed here.

- **XML**

An example of the actual XML being sent is displayed here. The XML not only shows the response and its respective elements, you can also see detailed examples of each element and how the expected data is formatted.

Responses

Response	Description
AccessType [page 169]	Holds the user's product access type.
ApplicationData [page 170]	Holds the name and other information about an application.
DatePreference [page 171]	Indicates a user's preferred date format.
Field [page 171]	Used to completely describe a field.
Field-Type [page 173]	Indicates the type of field.
FieldWithValue [page 175]	Combines the Field response type with the field's value.
GetReportsResult [page 175]	Holds the number of reports returned, as well as high-level information for each report.
GetStateChangeHistoryResult [page 177]	Holds the state change history for an item and the query range used to return the history.
NoteLoggerInfo [page 179]	Holds the e-mail address of the E-mail Recorder.
OrderBy [page 179]	Holds the ORDER BY definition for a report.
ProjectData [page 180]	Holds the name and other information about a project.
ReportDefinition [page 181]	Holds the columns and any ORDER BY definition for a report.

Response	Description
ReportInfo [page 182]	Holds a high-level description for a report.
ReportResult [page 184]	Holds the actual item data returned in a report.
RunReportResult [page 185]	Holds the results or output of a report.
SolutionData [page 191]	Holds the name and other information about a solution.
StateChangeHistory [page 192]	Holds the state change history for an item.
Status [page 193]	Holds status information for a Web service operation.
StatusEnum [page 194]	Indicates the type of status.
TableData [page 195]	Holds the name and other information about a table.
TimePreference [page 196]	Indicates a user's preferred time format.
Transition [page 197]	Holds the name and other information about a transition.
TTItemHolder [page 198]	Holds all of the SBM field information for an item and any applicable error messages.
TTItemList [page 201]	Contains the items and the total count of items that are returned from GetItemsByQuery.
UserHolder [page 202]	Holds user information and any applicable error messages.

Response	Description
UserInfo [page 206]	Holds the name and additional information about a user.
UserSolutionData [page 211]	Holds an ordered-list of applications that are accessible to a user. Also returns the user's home page report for each application and the preferred project list.

AccessType

Description

AccessType indicates the type of product-access a user is granted. The product access is used to determine the possible privileges that are available to users.

Parameters

Name	Type	Description
ACCESS-NONE	string	Indicates that the user's product access is set to None.
ACCESS-USER	string	Indicates Regular User product access.
ACCESS-OCCASIONAL	string	Indicates Occasional User product access.
ACCESS-EXTERNAL	string	Indicates External User product access.
ACCESS-ADMIN	string	Indicates Managed Administrator product access.
ACCESS-APIScript	string	Indicates API/Script product access.

Usage

The AccessType enumeration indicates the product access that is set for a user that is returned by the GetUsers call. For more information about each product access type, see the *SBM System Administrator Guide*.

XML

The following XML shows AccessType in the return element of the GetUsers response.

```
<ae:accessType>ACCESS-USER</ae:accessType>
```

ApplicationData

Description

The ApplicationData type holds the name and other information about an application in SBM. The ApplicationData type parameters are listed below.

Parameters

Name	Type	Description
application	ApplicationIdentifier [page 86]	Holds the complete identification information for an application.
description	string	The description of the application. Derived from the TS_DESCRIPTION column in TS_APPLICATIONS.
appDefUUID	string	The unique identifier that applies to the process app definition in the Application Administrator repository.
revision	string	The revision number.

Usage

The ApplicationData type holds ApplicationIdentifier, description, and UUID used to describe an application. You can use [GetApplications](#) [page 47] to retrieve the application data shown here.

XML

The following XML snippet shows the ApplicationData type in the return element of the GetApplications response.

```
<ae:GetApplicationsResponse>
  <ae:return>
    <ae:application xsi:type="ae:ApplicationIdentifier">
      <ae:displayName>Global Process App (eval)</ae:displayName>
      <ae:id>1</ae:id>
      <ae:uuid>global-d0f243dd-5ba7-44aa-a7fe-db1bd8a9c3bd</ae:uuid>
    </ae:application>
    <ae:description/>
    <ae:appDefUUID/>
    <ae:revision/>
  </ae:return>
  <ae:return>
    <ae:application xsi:type="ae:ApplicationIdentifier">
      <ae:displayName>Issue Defect Management</ae:displayName>
      <ae:id>2</ae:id>
      <ae:uuid>d365ac15-6b52-47a5-b82c-259e4591d022</ae:uuid>
    </ae:application>
    <ae:description>This process app manages the capture, triage and resolution process for software or hardware issues (bugs).</ae:description>
    <ae:appDefUUID/>
  </ae:return>
</ae:GetApplicationsResponse>
```

```

    <ae:revision/>
  </ae:return>
</ae:return>
  <ae:application xsi:type="ae:ApplicationIdentifier">
    <ae:displayName>Incident Management</ae:displayName>
    <ae:id>3</ae:id>
    <ae:uuid>955e8e0e-9342-46ed-ba06-d1bfdc1cdf80</ae:uuid>
  </ae:application>
  <ae:description>Provides simple Incident Management functionality
  →for small Support teams.</ae:description>
  <ae:appDefUUID/>
  <ae:revision>-</ae:revision>
</ae:return>
</ae:GetApplicationsResponse>

```

DatePreference

Description

DatePreference indicates a user's preferred date format. The available options are listed below.

Parameters

Name	Type	Description
DATE-FORMAT-FROM-LOCALE	string	Use the format based on the user's locale.
DATE-FORMAT-MM-DD-YYYY	string	Use a MM-DD-YYYY format for dates.
DATE-FORMAT-DD-MM-YYYY	string	Use a DD-MM-YYYY format for dates.
DATE-FORMAT-DD-MM-YYYY.S	string	Use a DD.MM.YYYY format for dates.
DATE-FORMAT-YYYY-MM-DD	string	Use a YYYY-MM-DD format for dates.

Usage

DatePreference is used to determine how dates display to a user in the SBM User Workspace. The various date formats are returned in the datePreference parameter of the GetUsers response. See [UserInfo \[page 206\]](#) for more information.

XML

The following XML shows DatePreference as seen in the return element of the GetUsers call.

```
<ae:datePreference>DATE-FORMAT-MM-DD-YYYY</ae:datePreference>
```

Field

Description

The Field type holds the name and other information about a field in SBM. The field type parameters are listed below.

Parameters

Name	Type	Description
field	FieldIdentifier [page 89]	Holds the identification information for a field.
fieldType	Field-Type [page 173]	Describes the type of field.
attribute	integer	Indicates a field attribute. Used to describe types of text fields, numeric fields, date/time, and binary fields as described in the TS_FIELDS table in the <i>Database Schema Reference</i> guide.
properties	integer	Indicates field properties. Used to indicate read-only status, numeric field calculation settings, and whether or not to display field values as checkboxes as described in the TS_FIELDS table in the <i>Database Schema Reference</i> guide.

Usage

The Field type completely describes an available field in SBM. Use GetTables to retrieve a list of fields available for a specified table. The list of fields appears in the field element.

XML

The following XML snippet shows the Field type in the fieldList element of the GetTables response.

```
<ae:GetTablesResponse>
  <ae:return>
    <ae:table xsi:type="ae:TableIdentifier">
      <ae:displayName>Issues</ae:displayName>
      <ae:id>1000</ae:id>
      <ae:uuid>dc8cd329-b430-436f-bb75-bf90008e6a50</ae:uuid>
      <ae:dbName>UBG_ISSUES</ae:dbName>
    </ae:table>
    <ae:solution xsi:type="ae:SolutionIdentifier">
      <ae:displayName>Issue Defect Management</ae:displayName>
      <ae:id>1</ae:id>
      <ae:uuid>cea0a86c-5d74-4e12-b8d6-9d6b90186f1e</ae:uuid>
      <ae:uniqueName>ISSUE_DEFECT_MANAGEMENT</ae:uniqueName>
      <ae:tabName>IDM</ae:tabName>
    </ae:solution>
    <ae:type>PRIMARY-TABLE</ae:type>
    <ae:description/>
    <ae:field>
      <ae:field xsi:type="ae:FieldIdentifier">
        <ae:displayName>Type</ae:displayName>
```

```

    <ae:id>52</ae:id>
    <ae:uuid>7718fd10-ff7a-4fc9-9f8e-581820cb77bf</ae:uuid>
    <ae:dbName>ISSUETYPE</ae:dbName>
  </ae:field>
  <ae:fieldType>FLDTYPE-SELECTION</ae:fieldType>
  <ae:attribute>0</ae:attribute>
  <ae:properties>0</ae:properties>
</ae:field>
</ae:return>
</ae:GetTablesResponse>

```

Field-Type

Description

Field-Type indicates the type of field. The available field types are listed below.

Parameters

Name	Type	Description
FLDTYPE_UNKNOWN	string	Indicates an unknown field.
FLDTYPE_NUMERIC	string	Indicates a numeric field.
FLDTYPE_TEXT	string	Indicates a text field.
FLDTYPE_DATETIME	string	Indicates a date/time field.
FLDTYPE_SELECTION	string	Indicates a single select field.
FLDTYPE_BINARY	string	Indicates a binary field.
FLDTYPE_STATE	string	Indicates a state field.
FLDTYPE_USER	string	Indicates a user field.
FLDTYPE_PROJECT	string	Indicates a project field.
FLDTYPE_SUMMATION	string	Indicates a summation field.
FLDTYPE_MULTIPLE_SELECTION	string	Indicates a multi-select field.
FLDTYPE_CONTACT	string	Indicates a contact field.
FLDTYPE_INCIDENT	string	Indicates a field unique to Incidents.
FLDTYPE_FOLDER	string	Indicates a folder type field.
FLDTYPE_RELATIONAL	string	Indicates a single relational field.

Name	Type	Description
FLDTYPE_SUBRELATIONAL	string	Indicates a sub-relational field.
FLDTYPE_SYSTEM	string	Indicates a system field.
FLDTYPE_MULTIPLE_RELATIONAL	string	Indicates a multi-relational field.
FLDTYPE_MULTIPLE_GROUP	string	Indicates a multi-group field.
FLDTYPE_MULTIPLE_USERGROUP	string	Indicates a multi-user field.

Usage

Field-Type can be used to identify the type of field that is returned in the GetTables response.

XML

The following XML shows Field-Type in the GetTables response.

```

<ae:GetTablesResponse>
  <ae:return>
    <ae:table xsi:type="ae:TableIdentifier">
      <ae:displayName>Issues</ae:displayName>
      <ae:id>1000</ae:id>
      <ae:uuid>dc8cd329-b430-436f-bb75-bf90008e6a50</ae:uuid>
      <ae:dbName>UBG_ISSUES</ae:dbName>
    </ae:table>
    <ae:solution xsi:type="ae:SolutionIdentifier">
      <ae:displayName>Issue Defect Management</ae:displayName>
      <ae:id>1</ae:id>
      <ae:uuid>cea0a86c-5d74-4e12-b8d6-9d6b90186f1e</ae:uuid>
      <ae:uniqueName>ISSUE_DEFECT_MANAGEMENT</ae:uniqueName>
      <ae:tabName>IDM</ae:tabName>
    </ae:solution>
    <ae:type>PRIMARY-TABLE</ae:type>
    <ae:description/>
    <ae:field>
      <ae:field xsi:type="ae:FieldIdentifier">
        <ae:displayName>Type</ae:displayName>
        <ae:id>52</ae:id>
        <ae:uuid>7718fd10-ff7a-4fc9-9f8e-581820cb77bf</ae:uuid>
        <ae:dbName>ISSUETYPE</ae:dbName>
      </ae:field>
      <ae:fieldType>FLDTYPE-SELECTION</ae:fieldType>
      <ae:attribute>0</ae:attribute>
      <ae:properties>0</ae:properties>
    </ae:field>
  </ae:return>
</ae:GetTablesResponse>

```

FieldWithValue

Description

The FieldWithValue type is used to completely describe a field value pairing in SBM. The FieldWithValue parameters are listed below.

Parameters

Name	Type	Description
field	Extension. See Field [page 171] .	Used to completely describe a field.
value	FieldValue [page 89]	Holds the generic identification information for a field value.

Usage

Field is used in combination with the FieldValue argument to completely describe a field value pairing in FieldWithValue.

XML

The following XML shows the `<ae:fieldValue>` response.

```
<ae:fieldValue xsi:type="ae:FieldWithValue">
  <ae:field xsi:type="ae:FieldIdentifier">
    <ae:displayName>Item Id</ae:displayName>
    <ae:id>53</ae:id>
    <ae:uuid>8a46043d-8cff-4871-a106-f0646ed3c58f</ae:uuid>
    <ae:dbName>ISSUEID</ae:dbName>
  </ae:field>
  <ae:fieldType>FLDTYPE-TEXT</ae:fieldType>
  <ae:attribute>0</ae:attribute>
  <ae:properties>0</ae:properties>
  <ae:value>
    <ae:displayValue>BUG000077</ae:displayValue>
    <ae:internalValue>000077</ae:internalValue>
    <ae:uuid/>
  </ae:value>
</ae:fieldValue>
```

GetReportsResult

Description

The GetReportsResult type holds the number of reports returned, as well as high-level information for each report. The GetReportsResult type parameters are listed below.

Parameters

Name	Type	Description
queryRange	QueryRange [page 100]	Specifies the number of reports that are returned.
report	ReportInfo [page 182]	Holds a description of the report.

Usage

The `GetReportsResult` type is sent back in response to a `GetReports` call. `GetReportsResult` summarizes the range of reports returned and provides an overview of each report. The actual contents of a report are obtained in the `RunReportResult` response. You can use [GetReports \[page 60\]](#) to retrieve the report data shown here.

XML

The following XML snippet shows the `GetReportsResult` type in the return element of the `GetReports` response.

```
<ae:GetReportsResponse>
  <ae:return>
    <ae:queryRange>
      <ae:startIndex>0</ae:startIndex>
      <ae:fetchSize>0</ae:fetchSize>
      <ae:totalCount>1</ae:totalCount>
    </ae:queryRange>
    <ae:report>
      <ae:report xsi:type="ae:ReportIdentifier">
        <ae:displayName>SCM Association Listing</ae:displayName>
        <ae:id>1</ae:id>
        <ae:uuid>4440bb69-40fa-4bbe-8fd8-41f113210243</ae:uuid>
      </ae:report>
      <ae:reportURL>tmtrack.dll?ReportPage&Template=reports
        →%2Flistframe&ReportId=1</ae:reportURL>
      <ae:reportType>LISTING</ae:reportType>
      <ae:reportCategory>USERREPORTS</ae:reportCategory>
      <ae:reportAccessLevel>GUEST</ae:reportAccessLevel>
      <ae:solution xsi:type="ae:SolutionIdentifier">
        <ae:displayName>Issue Defect Management</ae:displayName>
        <ae:id>1</ae:id>
        <ae:uuid>cea0a86c-5d74-4e12-b8d6-9d6b90186f1e</ae:uuid>
        <ae:uniqueName>ISSUE_DEFECT_MANAGEMENT</ae:uniqueName>
        <ae:tabName>IDM</ae:tabName>
      </ae:solution>
      <ae:table xsi:type="ae:TableIdentifier">
        <ae:displayName>SCM Associations</ae:displayName>
        <ae:id>1001</ae:id>
        <ae:uuid>50db5a94-645a-4e8b-9500-3f2950aa4f45</ae:uuid>
        <ae:dbName>UBG_SCM_ASSOCIATIONS</ae:dbName>
      </ae:table>
      <ae:project xsi:type="ae:ProjectIdentifier">
        <ae:displayName>Base Project</ae:displayName>
        <ae:id>1</ae:id>
    </ae:report>
  </ae:return>
</ae:GetReportsResponse>
```



```

        <ae:uuid>ROOTPROJECT</ae:uuid>
        <ae:fullyQualifiedName>Base Project</ae:fullyQualifiedName>
    </ae:project>
    <ae:createdBy xsi:type="ae:UserIdentifier">
        <ae:displayName>Joe Manager</ae:displayName>
        <ae:id>2</ae:id>
        <ae:uuid>85a33f0b-9542-43fe-90c1-e152eaaa777f</ae:uuid>
        <ae:loginId>joe</ae:loginId>
    </ae:createdBy>
    <ae:createDate>2009-02-05T22:10:01Z</ae:createDate>
    <ae:modifiedBy xsi:type="ae:UserIdentifier">
        <ae:displayName>Joe Manager</ae:displayName>
        <ae:id>2</ae:id>
        <ae:uuid>85a33f0b-9542-43fe-90c1-e152eaaa777f</ae:uuid>
        <ae:loginId>joe</ae:loginId>
    </ae:modifiedBy>
    <ae:modifiedDate>2009-03-04T22:49:31Z</ae:modifiedDate>
    <ae:execDate>2010-09-17T18:45:19Z</ae:execDate>
    <ae:isQueryAtRuntime>true</ae:isQueryAtRuntime>
</ae:report>
</ae:return>
</ae:GetReportsResponse>

```

GetStateChangeHistoryResult

Description

The `GetStateChangeHistoryResult` type holds the state change history for an item and the query range used to return the history. The `GetStateChangeHistoryResult` type parameters are listed below.

Parameters

Name	Type	Description
queryRange	QueryRange [page 100]	Specifies the number of state change history records that are returned.
stateChangeHistory	StateChangeHistory [page 192]	Holds the state change history for an item.

Usage

The `GetStateChangeHistoryResult` type is sent back in response to a `GetStateChangeHistory` call. `GetStateChangeHistoryResult` summarizes the range of state change records and provides detailed state change history information. See [GetStateChangeHistory](#) [page 63] for more information.

XML

The following XML snippet shows the `GetStateChangeHistoryResult` type in the `GetStateChangeHistory` response.

```
<ae:GetStateChangeHistoryResponse>
  <ae:return>
    <ae:queryRange>
      <ae:startIndex>0</ae:startIndex>
      <ae:fetchSize>2</ae:fetchSize>
      <ae:totalCount>4</ae:totalCount>
    </ae:queryRange>
    <ae:stateChangeHistory>
      <ae:newState xsi:type="ae:StateIdentifier">
        <ae:displayName>Evaluating Issue</ae:displayName>
        <ae:id>1</ae:id>
        <ae:uuid>985caf28-7a1c-4038-b6e2-c11703b214cd</ae:uuid>
        <ae:isClosed>false</ae:isClosed>
      </ae:newState>
      <ae:transition xsi:type="ae:TransitionIdentifier">
        <ae:displayName>Submit</ae:displayName>
        <ae:id>3</ae:id>
        <ae:uuid>a78f0a30-1305-46c2-b661-df8219c105b2</ae:uuid>
      </ae:transition>
      <ae:time>2009-02-05T22:04:32Z</ae:time>
      <ae:user xsi:type="ae:UserIdentifier">
        <ae:displayName>Joe Manager</ae:displayName>
        <ae:id>2</ae:id>
        <ae:uuid>85a33f0b-9542-43fe-90c1-e152eeea777f</ae:uuid>
        <ae:loginId>joe</ae:loginId>
      </ae:user>
      <ae:owner xsi:type="ae:UserIdentifier">
        <ae:displayName>Joe Manager</ae:displayName>
        <ae:id>2</ae:id>
        <ae:uuid>85a33f0b-9542-43fe-90c1-e152eeea777f</ae:uuid>
        <ae:loginId>joe</ae:loginId>
      </ae:owner>
    </ae:stateChangeHistory>
    <ae:stateChangeHistory>
      <ae:newState xsi:type="ae:StateIdentifier">
        <ae:displayName>Fixing Issue</ae:displayName>
        <ae:id>4</ae:id>
        <ae:uuid>a555a40c-7554-46f2-80bb-a6ff8b9ec917</ae:uuid>
        <ae:isClosed>false</ae:isClosed>
      </ae:newState>
      <ae:transition xsi:type="ae:TransitionIdentifier">
        <ae:displayName>Fix</ae:displayName>
        <ae:id>8</ae:id>
        <ae:uuid>7d095afe-1679-4e68-b492-0ad574bcdb2b</ae:uuid>
      </ae:transition>
      <ae:time>2009-02-05T22:23:06Z</ae:time>
      <ae:user xsi:type="ae:UserIdentifier">
        <ae:displayName>Joe Manager</ae:displayName>
        <ae:id>2</ae:id>
        <ae:uuid>85a33f0b-9542-43fe-90c1-e152eeea777f</ae:uuid>
        <ae:loginId>joe</ae:loginId>
      </ae:user>
      <ae:owner xsi:type="ae:UserIdentifier">
        <ae:displayName>Newton Engineer</ae:displayName>
        <ae:id>4</ae:id>
```

```
<ae:uuid>42a11cba-c422-442e-b299-5f3fe58a40e9</ae:uuid>
  <ae:loginId>newton</ae:loginId>
</ae:owner>
</ae:stateChangeHistory>
</ae:return>
</ae:GetStateChangeHistoryResponse>
```

NoteLoggerInfo

Description

The NoteLoggerInfo type holds the e-mail address for the E-mail Recorder mail box. The NoteLoggerInfo type parameters are listed below.

Parameters

Name	Type	Description
emailAddress	string	This is the e-mail address that is specified in the Mailbox Configuration for the E-mail Recorder.

Usage

The NoteLoggerInfo type holds the e-mail address that is configured for the E-mail Recorder mail box. You use [GetNoteLoggerInfo \[page 59\]](#) to retrieve the data shown here.

XML

The following XML snippet shows the NoteLoggerInfo type in the return element of the GetNoteLoggerInfo response.

```
<ae:GetNoteLoggerInfoResponse>
  <ae:return>
    <ae:emailAddress>test@companyName.com</ae:emailAddress>
  </ae:return>
</ae:GetNoteLoggerInfoResponse>
```

OrderBy

Description

The OrderBy type contains the ORDER BY definition for a report, if ORDER BY is specified for the report. The OrderBy type parameters are listed below.

Parameters

Name	Type	Description
firstFieldName	string	The first field used to order report results.
secondFieldName	string	The second field used to order report results.

Usage

The OrderBy type describes which field or fields are used to order the results returned in a report. OrderBy may have an empty return value even though certain fields are designated in the report definition to sort the results.

XML

The following XML snippet shows the OrderBy type in the reportDefinition element of the RunReport response.

```
<ae:reportDefinition>
  <ae:orderBy/>
```

ProjectData

Description

The ProjectData type holds the name and other information about a project in SBM. The ProjectData type parameters are listed below.

Parameters

Name	Type	Description
project	ProjectIdentifier [page 100]	The project that is returned.
description	string	The description of the project. Derived from the TS_DESCRIPTION column in TS_PROJECTS.

Usage

The ProjectData type holds the ID, UUID, name and description used to describe a project. You can use [GetSubmitProjects \[page 64\]](#) to retrieve the project data shown here.

XML

The following XML snippet shows the ProjectData type in the return element of the GetSubmitProjects response.

```
<ae:GetSubmitProjectsResponse>
  <ae:return>
    <ae:project xsi:type="ae:ProjectIdentifier">
      <ae:displayName>IDM Project</ae:displayName>
      <ae:id>2</ae:id>
      <ae:uuid>0b87f347-a00c-4359-9c16-625e847bfdab</ae:uuid>
      <ae:fullyQualifiedName>Base Project||IDM Project</ae:fullyQualifiedName>
    </ae:project>
    <ae:description/>
  </ae:return>
  <ae:return>
    <ae:project xsi:type="ae:ProjectIdentifier">
      <ae:displayName>Animation Pro</ae:displayName>
      <ae:id>6</ae:id>
      <ae:uuid>2ac5ef27-71da-4b07-ab7e-dddbc9c2d8c7</ae:uuid>
```

```

        <ae:fullyQualifiedName>Base Project||Base IDT Project||
        Software Development||Animation Pro</ae:fullyQualifiedName>
    </ae:project>
    <ae:description/>
</ae:return>
<ae:return>
    <ae:project xsi:type="ae:ProjectIdentifier">
        <ae:displayName>Image Builder</ae:displayName>
        <ae:id>8</ae:id>
        <ae:uuid>d1f727e2-9f70-4bee-afd5-bccaf0e71cda</ae:uuid>
        <ae:fullyQualifiedName>Base Project||Base IDT Project||
        Software Development||Image Builder</ae:fullyQualifiedName>
    </ae:project>
    <ae:description/>
</ae:return>
</ae:GetSubmitProjectsResponse>

```

ReportDefinition

Description

The ReportDefinition type contains columns and the order by definition for a report. The ReportDefinition type parameters are listed below.

Parameters

Name	Type	Description
orderBy	OrderBy [page 179]	Indicates which fields are used to organize the report results.
column	Field [page 171]	Holds a description the fields used in the report.

Usage

The ReportDefinition describes the basic structure of a report. The ReportDefinition type contains a description of each of the fields used in the report. If the report uses any of the columns to order the results, the OrderBy parameter will indicate which column(s) are used.

XML

The following XML snippet shows the ReportDefintion type in the return element of the RunReport response.

```

<ae:reportDefinition>
  <ae:orderBy/>
  <ae:column>
    <ae:field xsi:type="ae:FieldIdentifier">
      <ae:displayName>Item Id</ae:displayName>
      <ae:id>53</ae:id>
      <ae:uuid>8a46043d-8cff-4871-a106-f0646ed3c58f</ae:uuid>
      <ae:dbName>ISSUEID</ae:dbName>
    </ae:field>
  </ae:column>
</ae:reportDefinition>

```

```

    <ae:fieldType>FLDTYPE-TEXT</ae:fieldType>
    <ae:attribute>0</ae:attribute>
    <ae:properties>0</ae:properties>
  </ae:column>
  <ae:column>
    <ae:field xsi:type="ae:FieldIdentifier">
      <ae:displayName>Title</ae:displayName>
      <ae:id>54</ae:id>
      <ae:uuid>9bf79ebb-c66f-43dc-b608-5edc0677d4ca</ae:uuid>
      <ae:dbName>TITLE</ae:dbName>
    </ae:field>
    <ae:fieldType>FLDTYPE-TEXT</ae:fieldType>
    <ae:attribute>0</ae:attribute>
    <ae:properties>0</ae:properties>
  </ae:column>
</ae:reportDefinition>

```

ReportInfo

Description

The ReportInfo type contains a high-level description for a report. The ReportInfo type parameters are listed below.

Parameters

Name	Type	Description
report	ReportIdentifier [page 104]	Holds the identification information for a report.
reportURL	string	This is the URL for the report.
reportType	ReportType [page 104]	An enumeration that indicates the type of report.
reportCategory	ReportCategory [page 102]	A broader enumeration that indicates the category of report (built-in reports, application reports, reports you authored).
reportAccessLevel	ReportAccessLevel [page 101]	An enumeration that indicates the report's access level (PRIVATE, GUEST, USER, or MANAGER).
solution	SolutionIdentifier [page 106]	Holds the identification information for the solution that the report is based on.
table	TableIdentifier [page 109]	Holds the identification information for the table that the report is based on.
project	ProjectIdentifier [page 100]	Holds the identification information for the project that the report was created against.

Name	Type	Description
createdBy	UserIdentifier [page 138]	Holds the identification information for the user who authored the report.
createDate	dateTime	The date the report was created. See Supported Date/Time Formats [page 298] for more information.
modifiedBy	UserIdentifier [page 138]	Holds the identification information for the user that last modified the report.
modifiedDate	dateTime	The date the report was last modified. See Supported Date/Time Formats [page 298] for more information.
execDate	dateTime	The date the report was last executed. See Supported Date/Time Formats [page 298] for more information.
isQueryAtRuntime	boolean	This flag is used to indicate whether or not the report is a Query at Runtime report.

Usage

The ReportInfo type holds data used to describe a report. You can use [GetReports \[page 60\]](#) to retrieve the report data shown here.

XML

The following XML snippet shows ReportInfo in the `<ae:report>` return element of the GetReports response.

```
<ae:report>
  <ae:report xsi:type="ae:ReportIdentifier">
    <ae:displayName>My Test Report</ae:displayName>
    <ae:id>117</ae:id>
    <ae:uuid>fa736261-ef83-483f-b13c-1f1c4176a8ea</ae:uuid>
  </ae:report>
  <ae:reportURL>tmtrack.dll?ReportPage&Template=reports%
  →2Flistframe&ReportId=117</ae:reportURL>
  <ae:reportType>LISTING</ae:reportType>
  <ae:reportCategory>USERREPORTS</ae:reportCategory>
  <ae:reportAccessLevel>GUEST</ae:reportAccessLevel>
  <ae:project xsi:type="ae:ProjectIdentifier">
    <ae:displayName>Animation Pro</ae:displayName>
    <ae:id>0</ae:id>
    <ae:uuid>2ac5ef27-71da-4b07-ab7e-dddbc9c2d8c7</ae:uuid>
    <ae:fullyQualifiedName>Base Project||Base IDT Project||
    Software Development||Animation Pro<ae:fullyQualifiedName>
  </ae:project>
  <ae:createdBy xsi:type="ae:UserIdentifier">
    <ae:displayName>Administrator</ae:displayName>
```

```

    <ae:id>8</ae:id>
    <ae:uuid>9f9146a3-a273-4411-8000-8396688b7554</ae:uuid>
    <ae:loginId>admin</ae:loginId>
  </ae:createdBy>
  <ae:createDate>1969-12-31T23:59:58Z</ae:createDate>
  <ae:modifiedBy xsi:type="ae:UserIdentifier">
    <ae:displayName/>
    <ae:id>0</ae:id>
    <ae:uuid/>
    <ae:loginId/>
  </ae:modifiedBy>
  <ae:modifiedDate>1969-12-31T23:59:58Z</ae:modifiedDate>
  <ae:execDate>1969-12-31T23:59:58Z</ae:execDate>
  <ae:isQueryAtRuntime>false</ae:isQueryAtRuntime>
</ae:report>

```

ReportResult

Description

The ReportResult type contains the actual item data returned in a report. The ReportResult type parameters are listed below.

Parameters

Name	Type	Description
itemURL	string	This is the URL for the item returned in the report results.
item	ItemIdentifier [page 95]	The item described in the report.
fieldValue	FieldWithValue [page 175]	Holds the field type and field value for each field returned in the report results.

Usage

The ReportResult is sent back in response to [RunReport](#) [page 73]. ReportResult contains the URL for each item returned by the report. You can prepend `http://serverName/tmtrack/` to the itemURL contents and access the item from the SBM User Workspace. For every column defined in the report, the field type and field value are returned as well in the fieldValue element.

XML

The following XML snippet shows ReportResult in the return element of the RunReport response.

```

<ae:result>
  <ae:itemURL>tmtrack.dll?IssuePage&amp;RecordId=40&amp;
  →Template=view&amp;TableId=1000</ae:itemURL>
  <ae:item xsi:type="ae:ItemIdentifier">
    <ae:displayName>BUG000077</ae:displayName>

```



```

<ae:id>40</ae:id>
<ae:uuid>2841fdfb-9db3-43bb-a2e1-3bd54b64dbac</ae:uuid>
<ae:tableId>1000</ae:tableId>
<ae:tableIdItemId>1000:40</ae:tableIdItemId>
<ae:issueId>000077</ae:issueId>
</ae:item>
<ae:fieldValue xsi:type="ae:FieldWithValue">
  <ae:field xsi:type="ae:FieldIdentifier">
    <ae:displayName>Item Id</ae:displayName>
    <ae:id>53</ae:id>
    <ae:uuid>8a46043d-8cff-4871-a106-f0646ed3c58f</ae:uuid>
    <ae:dbName>ISSUEID</ae:dbName>
  </ae:field>
  <ae:fieldType>FLDTYPE-TEXT</ae:fieldType>
  <ae:attribute>0</ae:attribute>
  <ae:properties>0</ae:properties>
  <ae:value>
    <ae:displayValue>BUG000077</ae:displayValue>
    <ae:internalValue>000077</ae:internalValue>
    <ae:uuid/>
  </ae:value>
</ae:fieldValue>
<ae:fieldValue xsi:type="ae:FieldWithValue">
  <ae:field xsi:type="ae:FieldIdentifier">
    <ae:displayName>Title</ae:displayName>
    <ae:id>54</ae:id>
    <ae:uuid>9bf79ebb-c66f-43dc-b608-5edc0677d4ca</ae:uuid>
    <ae:dbName>TITLE</ae:dbName>
  </ae:field>
  <ae:fieldType>FLDTYPE-TEXT</ae:fieldType>
  <ae:attribute>0</ae:attribute>
  <ae:properties>0</ae:properties>
  <ae:value>
    <ae:displayValue>Scanner feature is not working correctly</ae:displayValue>
    <ae:internalValue>Scanner feature is not working correctly</ae:internalValue>
    <ae:uuid/>
  </ae:value>
</ae:fieldValue>
</ae:result>

```

RunReportResult

Description

The RunReportResult type contains the results or output of a report. The RunReportResult type parameters are listed below.

Parameters

Name	Type	Description
queryRange	QueryRange [page 100]	The QueryRange type specifies the number of reports returned.

Name	Type	Description
reportInfo	ReportInfo [page 182]	Holds a description of the report.
reportDefinition	ReportDefinition [page 181]	Holds the columns in a report and indicates which columns are used to order the results.
result	ReportResult [page 184]	Holds the actual data in the columns.

Usage

The RunReportResult is sent back in response to [RunReport](#) [page 73]. The RunReportResult type contains the actual data you would see in response to executing a report in the SBM User Workspace.

XML

The following XML snippet shows the ReportInfo, ReportDefintion, and ReportResult in the return element of the RunReport response.

```
<ae:RunReportResponse>
  <ae:return>
    <ae:queryRange>
      <ae:startIndex>2</ae:startIndex>
      <ae:fetchSize>4</ae:fetchSize>
      <ae:totalCount>47</ae:totalCount>
    </ae:queryRange>
    <ae:reportInfo>
      <ae:report xsi:type="ae:ReportIdentifier">
        <ae:displayName>My Test Report</ae:displayName>
        <ae:id>117</ae:id>
        <ae:uuid>fa736261-ef83-483f-b13c-1f1c4176a8ea</ae:uuid>
      </ae:report>
      <ae:reportURL>tmtrack.dll?ReportPage&Template=reports%2
      →Flistframe&ReportId=117</ae:reportURL>
      <ae:reportType>LISTING</ae:reportType>
      <ae:reportCategory>USERREPORTS</ae:reportCategory>
      <ae:reportAccessLevel>GUEST</ae:reportAccessLevel>
      <ae:solution xsi:type="ae:SolutionIdentifier">
        <ae:displayName>Issue Defect Management</ae:displayName>
        <ae:id>1</ae:id>
        <ae:uuid>cea0a86c-5d74-4e12-b8d6-9d6b90186f1e</ae:uuid>
        <ae:uniqueName>ISSUE_DEFECT_MANAGEMENT</ae:uniqueName>
        <ae:tabName>IDM</ae:tabName>
      </ae:solution>
      <ae:table xsi:type="ae:TableIdentifier">
        <ae:displayName>Issues</ae:displayName>
        <ae:id>1000</ae:id>
        <ae:uuid>dc8cd329-b430-436f-bb75-bf90008e6a50</ae:uuid>
        <ae:dbName>UBG_ISSUES</ae:dbName>
      </ae:table>
    <ae:project xsi:type="ae:ProjectIdentifier">
```

```

    <ae:displayName>Animation Pro</ae:displayName>
    <ae:id>6</ae:id>
    <ae:uuid>2ac5ef27-71da-4b07-ab7e-dddbc9c2d8c7</ae:uuid>
    <ae:fullyQualifiedName>Base Project||Base IDT Project||
    Software Development||Animation Pro</ae:fullyQualifiedName>
</ae:project>
<ae:createdBy xsi:type="ae:UserIdentifier">
    <ae:displayName>Administrator</ae:displayName>
    <ae:id>8</ae:id>
    <ae:uuid>9f9146a3-a273-4411-8000-8396688b7554</ae:uuid>
    <ae:loginId>admin</ae:loginId>
</ae:createdBy>
<ae:createDate>2010-09-29T16:10:10Z</ae:createDate>
<ae:modifiedBy xsi:type="ae:UserIdentifier">
    <ae:displayName/>
    <ae:id>0</ae:id>
    <ae:uuid/>
    <ae:loginId/>
</ae:modifiedBy>
<ae:modifiedDate>1969-12-31T23:59:58Z</ae:modifiedDate>
<ae:execDate>2010-09-29T16:10:14Z</ae:execDate>
<ae:isQueryAtRuntime>false</ae:isQueryAtRuntime>
</ae:reportInfo>
<ae:reportDefinition>
    <ae:orderBy/>
    <ae:column>
        <ae:field xsi:type="ae:FieldIdentifier">
            <ae:displayName>Item Id</ae:displayName>
            <ae:id>53</ae:id>
            <ae:uuid>8a46043d-8cff-4871-a106-f0646ed3c58f</ae:uuid>
            <ae:dbName>ISSUEID</ae:dbName>
        </ae:field>
        <ae:fieldType>FLDTYPE-TEXT</ae:fieldType>
        <ae:attribute>0</ae:attribute>
        <ae:properties>0</ae:properties>
    </ae:column>
    <ae:column>
        <ae:field xsi:type="ae:FieldIdentifier">
            <ae:displayName>Title</ae:displayName>
            <ae:id>54</ae:id>
            <ae:uuid>9bf79ebb-c66f-43dc-b608-5edc0677d4ca</ae:uuid>
            <ae:dbName>TITLE</ae:dbName>
        </ae:field>
        <ae:fieldType>FLDTYPE-TEXT</ae:fieldType>
        <ae:attribute>0</ae:attribute>
        <ae:properties>0</ae:properties>
    </ae:column>
</ae:reportDefinition>
<ae:result>
    <ae:itemURL>tmtrack.dll?IssuePage&RecordId=40&
    →Template=view&TableId=1000</ae:itemURL>
    <ae:item xsi:type="ae:ItemIdentifier">
        <ae:displayName>BUG000077</ae:displayName>
        <ae:id>40</ae:id>
        <ae:uuid>2841fdfb-9db3-43bb-a2e1-3bd54b64dbac</ae:uuid>

```

```
<ae:tableId>1000</ae:tableId>
<ae:tableIdItemId>1000:40</ae:tableIdItemId>
<ae:issueId>000077</ae:issueId>
</ae:item>
<ae:fieldValue xsi:type="ae:FieldWithValue">
  <ae:field xsi:type="ae:FieldIdentifier">
    <ae:displayName>Item Id</ae:displayName>
    <ae:id>53</ae:id>
    <ae:uuid>8a46043d-8cff-4871-a106-f0646ed3c58f</ae:uuid>
    <ae:dbName>ISSUEID</ae:dbName>
  </ae:field>
  <ae:fieldType>FLDTYPE-TEXT</ae:fieldType>
  <ae:attribute>0</ae:attribute>
  <ae:properties>0</ae:properties>
  <ae:value>
    <ae:displayValue>BUG000077</ae:displayValue>
    <ae:internalValue>000077</ae:internalValue>
    <ae:uuid/>
  </ae:value>
</ae:fieldValue>
<ae:fieldValue xsi:type="ae:FieldWithValue">
  <ae:field xsi:type="ae:FieldIdentifier">
    <ae:displayName>Title</ae:displayName>
    <ae:id>54</ae:id>
    <ae:uuid>9bf79ebb-c66f-43dc-b608-5edc0677d4ca</ae:uuid>
    <ae:dbName>TITLE</ae:dbName>
  </ae:field>
  <ae:fieldType>FLDTYPE-TEXT</ae:fieldType>
  <ae:attribute>0</ae:attribute>
  <ae:properties>0</ae:properties>
  <ae:value>
    <ae:displayValue>Scanner feature is not working correctly</ae:displayValue>
    <ae:internalValue>Scanner feature is not working correctly</ae:internalValue>
    <ae:uuid/>
  </ae:value>
</ae:fieldValue>
</ae:result>
<ae:result>
  <ae:itemURL>tmtrack.dll?IssuePage&amp;RecordId=41&amp;
  →Template=view&amp;TableId=1000</ae:itemURL>
  <ae:item xsi:type="ae:ItemIdentifier">
    <ae:displayName>BUG000078</ae:displayName>
    <ae:id>41</ae:id>
    <ae:uuid>24a726fb-3a43-44a2-aa94-9794ad9063d4</ae:uuid>
    <ae:tableId>1000</ae:tableId>
    <ae:tableIdItemId>1000:41</ae:tableIdItemId>
    <ae:issueId>000078</ae:issueId>
  </ae:item>
  <ae:fieldValue xsi:type="ae:FieldWithValue">
    <ae:field xsi:type="ae:FieldIdentifier">
      <ae:displayName>Item Id</ae:displayName>
      <ae:id>53</ae:id>
      <ae:uuid>8a46043d-8cff-4871-a106-f0646ed3c58f</ae:uuid>
      <ae:dbName>ISSUEID</ae:dbName>
    </ae:field>
```

```

    <ae:fieldType>FLDTYPE-TEXT</ae:fieldType>
    <ae:attribute>0</ae:attribute>
    <ae:properties>0</ae:properties>
    <ae:value>
      <ae:displayValue>BUG000078</ae:displayValue>
      <ae:internalValue>000078</ae:internalValue>
      <ae:uuid/>
    </ae:value>
  </ae:fieldValue>
  <ae:fieldValue xsi:type="ae:FieldWithValue">
    <ae:field xsi:type="ae:FieldIdentifier">
      <ae:displayName>Title</ae:displayName>
      <ae:id>54</ae:id>
      <ae:uuid>9bf79ebb-c66f-43dc-b608-5edc0677d4ca</ae:uuid>
      <ae:dbName>TITLE</ae:dbName>
    </ae:field>
    <ae:fieldType>FLDTYPE-TEXT</ae:fieldType>
    <ae:attribute>0</ae:attribute>
    <ae:properties>0</ae:properties>
    <ae:value>
      <ae:displayValue>WebSite Links</ae:displayValue>
      <ae:internalValue>WebSite Links</ae:internalValue>
      <ae:uuid/>
    </ae:value>
  </ae:fieldValue>
</ae:result>
<ae:result>
  <ae:itemURL>tmtrack.dll?IssuePage&amp;RecordId=42&amp;
  →Template=view&amp;TableId=1000</ae:itemURL>
  <ae:item xsi:type="ae:ItemIdentifier">
    <ae:displayName>ENH000079</ae:displayName>
    <ae:id>42</ae:id>
    <ae:uuid>d67f64c6-59a0-4db7-b195-abbb7434a4b5</ae:uuid>
    <ae:tableId>1000</ae:tableId>
    <ae:tableIdItemId>1000:42</ae:tableIdItemId>
    <ae:issueId>000079</ae:issueId>
  </ae:item>
  <ae:fieldValue xsi:type="ae:FieldWithValue">
    <ae:field xsi:type="ae:FieldIdentifier">
      <ae:displayName>Item Id</ae:displayName>
      <ae:id>53</ae:id>
      <ae:uuid>8a46043d-8cff-4871-a106-f0646ed3c58f</ae:uuid>
      <ae:dbName>ISSUEID</ae:dbName>
    </ae:field>
    <ae:fieldType>FLDTYPE-TEXT</ae:fieldType>
    <ae:attribute>0</ae:attribute>
    <ae:properties>0</ae:properties>
    <ae:value>
      <ae:displayValue>ENH000079</ae:displayValue>
      <ae:internalValue>000079</ae:internalValue>
      <ae:uuid/>
    </ae:value>
  </ae:fieldValue>
  <ae:fieldValue xsi:type="ae:FieldWithValue">
    <ae:field xsi:type="ae:FieldIdentifier">

```

```
        <ae:displayName>Title</ae:displayName>
        <ae:id>54</ae:id>
        <ae:uuid>9bf79ebb-c66f-43dc-b608-5edc0677d4ca</ae:uuid>
        <ae:dbName>TITLE</ae:dbName>
    </ae:field>
    <ae:fieldType>FLDTYPE-TEXT</ae:fieldType>
    <ae:attribute>0</ae:attribute>
    <ae:properties>0</ae:properties>
    <ae:value>
        <ae:displayValue>Zoom tool needs more options</ae:displayValue>
        <ae:internalValue>Zoom tool needs more options</ae:internalValue>
        <ae:uuid/>
    </ae:value>
</ae:fieldValue>
</ae:result>
<ae:result>
    <ae:itemURL>tmtrack.dll?IssuePage&amp;RecordId=43&amp;
    →Template=view&amp;TableId=1000</ae:itemURL>
    <ae:item xsi:type="ae:ItemIdentifier">
        <ae:displayName>BUG000080</ae:displayName>
        <ae:id>43</ae:id>
        <ae:uuid>1d83e8dc-e0e4-495f-b858-8e7673abca47</ae:uuid>
        <ae:tableId>1000</ae:tableId>
        <ae:tableIdItemId>1000:43</ae:tableIdItemId>
        <ae:issueId>000080</ae:issueId>
    </ae:item>
    <ae:fieldValue xsi:type="ae:FieldWithValue">
        <ae:field xsi:type="ae:FieldIdentifier">
            <ae:displayName>Item Id</ae:displayName>
            <ae:id>53</ae:id>
            <ae:uuid>8a46043d-8cff-4871-a106-f0646ed3c58f</ae:uuid>
            <ae:dbName>ISSUEID</ae:dbName>
        </ae:field>
        <ae:fieldType>FLDTYPE-TEXT</ae:fieldType>
        <ae:attribute>0</ae:attribute>
        <ae:properties>0</ae:properties>
        <ae:value>
            <ae:displayValue>BUG000080</ae:displayValue>
            <ae:internalValue>000080</ae:internalValue>
            <ae:uuid/>
        </ae:value>
    </ae:fieldValue>
    <ae:fieldValue xsi:type="ae:FieldWithValue">
        <ae:field xsi:type="ae:FieldIdentifier">
            <ae:displayName>Title</ae:displayName>
            <ae:id>54</ae:id>
            <ae:uuid>9bf79ebb-c66f-43dc-b608-5edc0677d4ca</ae:uuid>
            <ae:dbName>TITLE</ae:dbName>
        </ae:field>
        <ae:fieldType>FLDTYPE-TEXT</ae:fieldType>
        <ae:attribute>0</ae:attribute>
        <ae:properties>0</ae:properties>
        <ae:value>
            <ae:displayValue>Setup is not installing examples</ae:displayValue>
            <ae:internalValue>Setup is not installing examples</ae:internalValue>
```

```

        <ae:uuid/>
    </ae:value>
</ae:fieldValue>
</ae:result>
</ae:return>
</ae:RunReportResponse>

```

SolutionData

Description

The SolutionData type holds the name and other information about an solution in SBM. The SolutionData type parameters are listed below.

Parameters

Name	Type	Description
solution	SolutionIdentifier [page 106]	Holds the identification information for a solution.
type	SolutionIdentifier [page 106]	The type of solution. The possible values are: TEAMTRACK_SOLUTION, USER_SOLUTION, and THIRD_PARTYSOLUTION.
prefix	string	The prefix of the solution. Derived from the TS_PREFIX column in TS_SOLUTIONS.
description	string	The description of the solution. Derived from the TS_DESCRIPTION column in TS_SOLUTIONS.
processAppUUID	string	The UUID of the associated process app.

Usage

The SolutionData type holds the SolutionIdentifier and additional information used to completely describe a solution. You can use [GetSolutions \[page 62\]](#) to retrieve the solution data shown here.

XML

The following XML snippet shows the SolutionData type in the return element of the GetSolutions response.

```

<ae:GetSolutionsResponse>
  <ae:return>
    <ae:solution xsi:type="ae:SolutionIdentifier">
      <ae:displayName>Issue Defect Management</ae:displayName>
      <ae:id>1</ae:id>
      <ae:uuid>cea0a86c-5d74-4e12-b8d6-9d6b90186f1e</ae:uuid>
      <ae:uniqueName>ISSUE_DEFECT_MANAGEMENT</ae:uniqueName>
      <ae:tabName>IDM</ae:tabName>
    </ae:solution>
    <ae:type>USER-SOLUTION</ae:type>
  </ae:return>
</ae:GetSolutionsResponse>

```

```

    <ae:prefix>UBG</ae:prefix>
    <ae:description>Last updated 1/26/09</ae:description>
    <ae:processAppUUID>d365ac15-6b52-47a5-b82c-259e4591d022</ae:processAppUUID>
  </ae:return>
</ae:return>
  <ae:solution xsi:type="ae:SolutionIdentifier">
    <ae:displayName>Incident Management</ae:displayName>
    <ae:id>2</ae:id>
    <ae:uuid>c6f06a70-4d06-42a4-a3a9-50b2120dca41</ae:uuid>
    <ae:uniqueName>INCIDENT_MANAGEMENT</ae:uniqueName>
    <ae:tabName>Incident Mgmt</ae:tabName>
  </ae:solution>
  <ae:type>USER-SOLUTION</ae:type>
  <ae:prefix>UIM</ae:prefix>
  <ae:description/>
  <ae:processAppUUID>955e8e0e-9342-46ed-ba06-d1bfdc1cdf80</ae:processAppUUID>
</ae:return>
</ae:return>
  <ae:solution xsi:type="ae:SolutionIdentifier">
    <ae:displayName>Change Request Management</ae:displayName>
    <ae:id>3</ae:id>
    <ae:uuid>0ea28a74-6dde-406f-b19a-4c45aec40294</ae:uuid>
    <ae:uniqueName>CHANGE_REQUEST_MANAGEMENT</ae:uniqueName>
    <ae:tabName>Change Requests</ae:tabName>
  </ae:solution>
  <ae:type>USER-SOLUTION</ae:type>
  <ae:prefix>UCA</ae:prefix>
  <ae:description/>
  <ae:processAppUUID>660af90a-7d96-405c-a859-7115a9114ed3</ae:processAppUUID>
</ae:return>
</ae:GetSolutionsResponse>

```

StateChangeHistory

Description

The StateChangeHistory type holds the state change history for an item. The StateChangeHistory type parameters are listed below.

Parameters

Name	Type	Description
newState	StateIdentifier [page 107]	The state the item is in after the transition.
transition	TransitionIdentifier [page 111]	The transition that was performed.
time	dateTime	The timestamp used when the change record was created.

Name	Type	Description
user	UserIdentifier [page 138]	The user who performed the transition.
owner	UserIdentifier [page 138]	The current owner of the item.

Usage

The StateChangeHistory type is part of the response to the GetStateChangeHistory call. StateChangeHistory holds the same information that you see in the details of the State Change History for an item in the SBM User Workspace. You can use [GetStateChangeHistory \[page 63\]](#) to retrieve the data shown here.

XML

The following XML snippet shows the StateChangeHistory type in the return element of the GetStateChangeHistory response.

```
<ae:stateChangeHistory>
  <ae:newState xsi:type="ae:StateIdentifier">
    <ae:displayName>Evaluating Issue</ae:displayName>
    <ae:id>1</ae:id>
    <ae:uuid>985caf28-7a1c-4038-b6e2-c11703b214cd</ae:uuid>
    <ae:isClosed>>false</ae:isClosed>
  </ae:newState>
  <ae:transition xsi:type="ae:TransitionIdentifier">
    <ae:displayName>Submit</ae:displayName>
    <ae:id>3</ae:id>
    <ae:uuid>a78f0a30-1305-46c2-b661-df8219c105b2</ae:uuid>
  </ae:transition>
  <ae:time>2009-02-05T22:04:32Z</ae:time>
  <ae:user xsi:type="ae:UserIdentifier">
    <ae:displayName>Joe Manager</ae:displayName>
    <ae:id>2</ae:id>
    <ae:uuid>85a33f0b-9542-43fe-90c1-e152eeee777f</ae:uuid>
    <ae:loginId>joe</ae:loginId>
  </ae:user>
  <ae:owner xsi:type="ae:UserIdentifier">
    <ae:displayName>Joe Manager</ae:displayName>
    <ae:id>2</ae:id>
    <ae:uuid>85a33f0b-9542-43fe-90c1-e152eeee777f</ae:uuid>
    <ae:loginId>joe</ae:loginId>
  </ae:owner>
</ae:stateChangeHistory>
```

Status

Description

The Status type holds status information for a Web service operation. The Status type parameters are listed below.

Parameters

Name	Type	Description
status	StatusEnum [page 194] .	Status enumeration that indicates whether the message is a warning message, informational message, or error message.
code	string	The message code that is returned.
message	string	The message string that is returned.

Usage

The Status type is used to return status messages from service operations.

XML

The following XML shows Status as seen in the return element of a GetItems call that did not send the proper ItemIdentifier content.

```
<ae:return>
  <ae:status>
    <ae:status>IS-ERROR</ae:status>
    <ae:message>Invalid item id 0 for table 1000.</ae:message>
  </ae:status>
</ae:return>
```

StatusEnum**Description**

StatusEnum indicates the type of message that is returned in the Status response.

Parameters

Name	Type	Description
IS-WARNING	string	Indicates a warning message.
IS-INFORMATION	string	Indicates an informational message.
IS-ERROR	string	Indicates an error message.

Usage

None.

XML

The following XML shows StatusEnum in the `<ae:status>` return element of a typical response.

```
<ae:return>
  <ae:status>
    <ae:status>IS-ERROR</ae:status>
    <ae:message>Invalid project 0.</ae:message>
  </ae:status>
</ae:return>
```

TableData

Description

The TableData type holds the name and other information about a table in SBM. The TableData type parameters are listed below.

Parameters

Name	Type	Description
table	TableIdentifier [page 109]	Holds the identification information for a table.
solution	SolutionIdentifier [page 106]	Holds the identification information for a solution.
type	Table-Type [page 110]	The type of table returned.
description	string	The description of the table. Derived from the TS_DESCRIPTION column in TS_TABLES.
field	Field [page 171]	Fields that are defined for the table.

Usage

The TableData type holds the TableIdentifier and additional information used to completely describe a table. You can use [GetTables \[page 65\]](#) to retrieve the table data shown here.

XML

The following XML snippet shows the TableData type in the return element of the GetTables response.

```
<ae:GetTablesResponse>
  <ae:return>
    <ae:table xsi:type="ae:TableIdentifier">
      <ae:displayName>Issues</ae:displayName>
      <ae:id>1000</ae:id>
      <ae:uuid>dc8cd329-b430-436f-bb75-bf90008e6a50</ae:uuid>
      <ae:dbName>UBG_ISSUES</ae:dbName>
    </ae:table>
```

```

<ae:solution xsi:type="ae:SolutionIdentifier">
  <ae:displayName>Issue Defect Management</ae:displayName>
  <ae:id>1</ae:id>
  <ae:uuid>cea0a86c-5d74-4e12-b8d6-9d6b90186f1e</ae:uuid>
  <ae:uniqueName>ISSUE_DEFECT_MANAGEMENT</ae:uniqueName>
  <ae:tabName>IDM</ae:tabName>
</ae:solution>
<ae:type>PRIMARY-TABLE</ae:type>
<ae:description/>
  <ae:field>
    <ae:field xsi:type="ae:FieldIdentifier">
      <ae:displayName>Type</ae:displayName>
      <ae:id>52</ae:id>
      <ae:uuid>7718fd10-ff7a-4fc9-9f8e-581820cb77bf</ae:uuid>
      <ae:dbName>ISSUETYPE</ae:dbName>
    </ae:field>
    <ae:fieldType>FLDTYPE-SELECTION</ae:fieldType>
    <ae:attribute>0</ae:attribute>
    <ae:properties>0</ae:properties>
  </ae:field>
  <ae:field>
    <ae:field xsi:type="ae:FieldIdentifier">
      <ae:displayName>Item Id</ae:displayName>
      <ae:id>53</ae:id>
      <ae:uuid>8a46043d-8cff-4871-a106-f0646ed3c58f</ae:uuid>
      <ae:dbName>ISSUEID</ae:dbName>
    </ae:field>
    <ae:fieldType>FLDTYPE-TEXT</ae:fieldType>
    <ae:attribute>0</ae:attribute>
    <ae:properties>0</ae:properties>
  </ae:field>
</ae:GetTablesResponse>

```

TimePreference

Description

TimePreference indicates a user's preferred time format. The available options are listed below.

Parameters

Name	Type	Description
TIME-FORMAT-12HOUR	string	Indicates a 12-hour clock preference.
TIME-FORMAT-24HOUR	string	Indicates a 24-hour clock preference.
TIME-FORMAT-USE-GMT-OFFSET	string	Indicates the use of a GMT offset.
TIME-FORMAT-HONOR-DAYLIGHT	string	Indicates time format with daylight savings honored.

Usage

TimePreference is used to determine how time is displayed to a user in the SBM User Workspace. The various time formats are returned in the timePreference parameter of the GetUsers response. See the [UserInfo \[page 206\]](#) response for more information.

XML

The following XML shows TimePreference as seen in the return element of a GetUsers response.

```
<ae:timePreference>TIME-FORMAT-12HOUR</ae:timePreference>
```

Transition

Description

The Transition type holds the name and other information about a transition in SBM. The Transition type parameters are listed below.

Parameters

Name	Type	Description
transition	TransitionIdentifier [page 111]	Holds identification information for a transition.
fromState	StateIdentifier [page 107]	The state from which the transition begins.
toState	StateIdentifier [page 107]	The state to which the transition moves or ends.
type	Transition-Type [page 111]	The type of transition.
project	ProjectIdentifier [page 100]	For a TRANSITION_POST type of transition, this indicates the project into which the item is to be submitted.
transitionAttributes	string	The list of transition attributes associated with this transition.

Usage

The Transition type completely describes an available transition or transitions on an item in SBM. Use [GetAvailableTransitions \[page 49\]](#) to retrieve a list of transitions available for a specified item.

The transitionAttributes argument is optional and is only used as another means to describe the transition. A transition attribute is typically associated with unique enabler license - integration transitions.

XML

The following XML snippet shows the Transition type in the return element of the `GetAvailableTransitions` response.

```
<ae:GetAvailableTransitionsResponse>
  <ae:return>
    <ae:transition xsi:type="ae:TransitionIdentifier">
      <ae:displayName>Post Issue</ae:displayName>
      <ae:id>8</ae:id>
      <ae:uuid>7d095afe-1679-4e68-b492-0ad574bcdb2b</ae:uuid>
    </ae:transition>
    <ae:fromState xsi:type="ae:StateIdentifier">
      <ae:displayName>Evaluating Issue</ae:displayName>
      <ae:id>1</ae:id>
      <ae:uuid>985caf28-7a1c-4038-b6e2-c11703b214cd</ae:uuid>
      <ae:isClosed>false</ae:isClosed>
    </ae:fromState>
    <ae:toState xsi:type="ae:StateIdentifier">
      <ae:displayName>Fixing Issue</ae:displayName>
      <ae:id>4</ae:id>
      <ae:uuid>a555a40c-7554-46f2-80bb-a6ff8b9ec917</ae:uuid>
      <ae:isClosed>false</ae:isClosed>
    </ae:toState>
    <ae:type>TRANSITION-POST</ae:type>
    <ae:project xsi:type="ae:ProjectIdentifier">
      <ae:displayName>Animation Pro</ae:displayName>
      <ae:id>6</ae:id>
      <ae:uuid>2ac5ef27-71da-4b07-ab7e-dddbc9c2d8c7</ae:uuid>
      <ae:fullyQualifiedName>Base Project||Base IDT Project||
        Software Development||Animation Pro</ae:fullyQualifiedName>
    </ae:project>
  </ae:return>
</ae:GetAvailableTransitionsResponse>
```

TTItemHolder

Description

The `TTItemHolder` type contains the `TTItem` response and any returned errors for items that were not successfully created or updated. The `TTItemHolder` type parameters are listed below.

Parameters

Name	Type	Description
item	TTItem [page 114]	The <code>TTItem</code> type holds the generic data for an item.
status	Status [page 193]	Holds status information for any messages or failures that are encountered.

Usage

The `TTItemHolder` type holds all of the field information for an item along with attached notes and item links. If any errors occur during the call, they are returned in the status element in the order in which they were processed.

XML

The following XML snippet shows `TTItemHolder` in the `<ae:item>` return element of the `GetItems` response. Note that an error message is returned in this example (inside the `message` element). In this example, three items were requested using the `GetItems` call; however, the second item was not given an internal ID. Therefore, an error was returned for the second item, though the service continued to run and returned the third item (BUG000173).

```
<ae:item>
  <ae:id xsi:type="ae:ItemIdentifier">
    <ae:displayName>BUG000164</ae:displayName>
    <ae:id>106</ae:id>
    <ae:uuid>7fe3elf4-c8a8-4a99-8aec-fd236699a516</ae:uuid>
    <ae:tableId>1000</ae:tableId>
    <ae:tableIdItemId>1000:106</ae:tableIdItemId>
    <ae:issueId>000164</ae:issueId>
  </ae:id>
  <ae:itemType>Bug Report</ae:itemType>
  <ae:project xsi:type="ae:ProjectIdentifier">
    <ae:displayName>IDM Project</ae:displayName>
    <ae:id>2</ae:id>
    <ae:uuid>0b87f347-a00c-4359-9c16-625e847bfdab</ae:uuid>
    <ae:fullyQualifiedName>Base Project||IDM Project</ae:fullyQualifiedName>
  </ae:project>
  <ae:title>I'm getting a 502 error</ae:title>
  <ae:description>When I access the HR system, I'm getting a 502 error.
  </ae:description>
  <ae:createdBy xsi:type="ae:UserIdentifier">
    <ae:displayName>Jan Lvl 2 Support</ae:displayName>
    <ae:id>18</ae:id>
    <ae:uuid>09c6aa7e-e2b9-4ff4-9377-0d4090cab4fe</ae:uuid>
    <ae:loginId>jan</ae:loginId>
  </ae:createdBy>
  <ae:createDate>2009-02-18T18:18:11Z</ae:createDate>
  <ae:modifiedBy xsi:type="ae:UserIdentifier">
    <ae:displayName>Rhadika Tester</ae:displayName>
    <ae:id>28</ae:id>
    <ae:uuid>43a37118-c82a-4fbd-803f-a1f0ad127c12</ae:uuid>
    <ae:loginId>rhadika</ae:loginId>
  </ae:modifiedBy>
  <ae:modifiedDate>2009-02-18T18:22:53Z</ae:modifiedDate>
  <ae:activeInactive>false</ae:activeInactive>
  <ae:state xsi:type="ae:StateIdentifier">
    <ae:displayName>Resolved</ae:displayName>
    <ae:id>7</ae:id>
    <ae:uuid>ea57582f-5c62-4bcd-b250-5e21cad308e3</ae:uuid>
    <ae:isClosed>true</ae:isClosed>
  </ae:state>
  <ae:owner xsi:type="ae:UserIdentifier">
    <ae:displayName/>
```

```
<ae:id>0</ae:id>
<ae:uuid/>
<ae:loginId/>
</ae:owner>
<ae:url>http://serverName:80/tmtrack/tmtrack.dll?IssuePage&
→RecordId=106&Template=view&TableId=1000</ae:url>
</ae:item>
</ae:return>
<ae:return>
  <ae:status>
    <ae:status>IS-ERROR</ae:status>
    <ae:message>Invalid item id 0 for table 1000.</ae:message>
  </ae:status>
</ae:return>
<ae:return>
<ae:item>
  <ae:id xsi:type="ae:ItemIdentifier">
    <ae:displayName>BUG000173</ae:displayName>
    <ae:id>109</ae:id>
    <ae:uuid>c1e62a05-5dd1-46e7-9cdc-b91518893b23</ae:uuid>
    <ae:tableId>1000</ae:tableId>
    <ae:tableIdItemId>1000:109</ae:tableIdItemId>
    <ae:issueId>000173</ae:issueId>
  </ae:id>
  <ae:itemType>Bug Report</ae:itemType>
  <ae:project xsi:type="ae:ProjectIdentifier">
    <ae:displayName>IDM Project</ae:displayName>
    <ae:id>2</ae:id>
    <ae:uuid>0b87f347-a00c-4359-9c16-625e847bdfdb</ae:uuid>
    <ae:fullyQualifiedName>Base Project||IDM Project</ae:fullyQualifiedName>
  </ae:project>
  <ae:title>Image Builder crashes my system.</ae:title>
  <ae:description>When user logs on to Image Builder, she gets a blue
→screen.</ae:description>
  <ae:createdBy xsi:type="ae:UserIdentifier">
    <ae:displayName>Jan Lvl 2 Support</ae:displayName>
    <ae:id>18</ae:id>
    <ae:uuid>09c6aa7e-e2b9-4ff4-9377-0d4090cab4fe</ae:uuid>
    <ae:loginId>jan</ae:loginId>
  </ae:createdBy>
  <ae:createDate>2009-02-19T19:44:33Z</ae:createDate>
  <ae:modifiedBy xsi:type="ae:UserIdentifier">
    <ae:displayName>Administrator</ae:displayName>
    <ae:id>8</ae:id>
    <ae:uuid>9f9146a3-a273-4411-8000-8396688b7554</ae:uuid>
    <ae:loginId>admin</ae:loginId>
  </ae:modifiedBy>
  <ae:modifiedDate>2010-09-08T21:55:07Z</ae:modifiedDate>
  <ae:activeInactive>true</ae:activeInactive>
  <ae:state xsi:type="ae:StateIdentifier">
    <ae:displayName>Testing Issue</ae:displayName>
    <ae:id>5</ae:id>
    <ae:uuid>4c9d41b8-d5b3-49ef-a69c-d9e6320ad85a</ae:uuid>
    <ae:isClosed>>false</ae:isClosed>
```



```

</ae:state>
<ae:owner xsi:type="ae:UserIdentifier">
  <ae:displayName>Rhadika Tester</ae:displayName>
  <ae:id>28</ae:id>
  <ae:uuid>43a37118-c82a-4fbd-803f-alf0ad127c12</ae:uuid>
  <ae:loginId>rhadika</ae:loginId>
</ae:owner>
<ae:url>http://serverName:80/tmtrack/tmtrack.dll?IssuePage&
→RecordId=109&Template=view&TableId=1000</ae:url>
</ae:item>

```

TTItemList

Description

The TTItemList type contains the items and the total count of items that are returned from GetItemsByQuery. The TTItemList type parameters are listed below.

Parameters

Name	Type	Description
item	TTItem [page 114]	The <i>TTItem</i> type holds the generic data for an item.
totalCount	integer	The total number of items returned by the query.

Usage

The TTItemList type holds all of the field information for items that are returned by GetItemsByQuery, along with a count of the total number of returned items.

XML

The following XML snippet shows TTItemList in the return element of the GetItemsByQuery response.

```

<ae:GetItemsByQueryResponse>
  <ae:return>
    <ae:item>
      <ae:id xsi:type="ae:ItemIdentifier">
        <ae:displayName>BUG000030</ae:displayName>
        <ae:id>14</ae:id>
        <ae:uuid>79fd7446-f96e-4265-b765-cbcd0e66357f</ae:uuid>
        <ae:tableId>1000</ae:tableId>
        <ae:tableIdItemId>1000:14</ae:tableIdItemId>
        <ae:issueId>000030</ae:issueId>
      </ae:id>
      <detailed_item_data>
    </ae:item>
    <ae:item>
      <ae:id xsi:type="ae:ItemIdentifier">
        <ae:displayName>ENH000032</ae:displayName>
        <ae:id>15</ae:id>
        <ae:uuid>49a065dd-cf7f-499c-852f-a78c751fdd2f</ae:uuid>

```

```

    <ae:tableId>1000</ae:tableId>
    <ae:tableIdItemId>1000:15</ae:tableIdItemId>
    <ae:issueId>000032</ae:issueId>
  </ae:id>
  <detailed_item_data>
</ae:item>
  <ae:totalCount>2</ae:totalCount>
</ae:return>
</ae:GetItemsByQueryResponse>

```

UserHolder

Description

The UserHolder type contains the UserInfo response and any errors for user records that were not successfully returned. The UserHolder type parameters are listed below.

Parameters

Name	Type	Description
user	UserInfo [page 206]	The userInfo type holds the complete informaton that describes a user.
status	Status [page 193]	Holds status information for any messages or failures that are encountered.

Usage

The UserHolder type holds detailed information for a user and his or her preferences. If any errors occur during the call, they are returned in the status element in the order in which they were processed.

XML

The following XML snippet shows UserHolder in the `<ae:user>` return element of the GetUsers response. Note that an error message is returned in this example (inside the `message` element). In this example, three users were requested using the GetUsers call; however, the second user (Kathy) was not properly identified. Therefore, an error was returned for the second user, though the service continued to run and returned the third user (Laura).

```

<ae:GetUsersResponse>
  <ae:return>
    <ae:user>
      <ae:id xsi:type="ae:UserIdentifier">
        <ae:displayName>John Support Manager</ae:displayName>
        <ae:id>21</ae:id>
        <ae:uuid>08784a43-970f-4d28-9a6e-c301077ca653</ae:uuid>
        <ae:loginId>john</ae:loginId>
      </ae:id>
      <ae:accessType>ACCESS-USER</ae:accessType>
      <ae:email>john@companyName.com</ae:email>
      <ae:emailCC/>
    </ae:user>
  </ae:return>
</ae:GetUsersResponse>

```

```
<ae:timezone/>
<ae:offsetFromGMT>-25200000</ae:offsetFromGMT>
<ae:dstSavings>3600000</ae:dstSavings>
<ae:datePreference>DATE-FORMAT-FROM-LOCALE</ae:datePreference>
<ae:timePreference>TIME-FORMAT-12HOUR</ae:timePreference>
<ae:namespaceName>00000</ae:namespaceName>
<ae:phoneNumber/>
<ae:locale>en_US</ae:locale>
<ae:isDeleted>>false</ae:isDeleted>
<ae:maxNotes>10</ae:maxNotes>
<ae:maxChangeHistory>10</ae:maxChangeHistory>
<ae:maxItemsPerPage>20</ae:maxItemsPerPage>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>Everyone</ae:displayName>
  <ae:id>1</ae:id>
  <ae:uuid>ade39c21-e7b2-4dcb-a231-d3d872671b59</ae:uuid>
</ae:group>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>IDM View Only</ae:displayName>
  <ae:id>5</ae:id>
  <ae:uuid>af38532d-e79c-495f-a3e6-f4bf784cc492</ae:uuid>
</ae:group>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>IM Technician</ae:displayName>
  <ae:id>6</ae:id>
  <ae:uuid>902a1300-6ae1-44d7-b46a-e420babe8497</ae:uuid>
</ae:group>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>IM Manager</ae:displayName>
  <ae:id>7</ae:id>
  <ae:uuid>b70af5de-6642-4228-b9bb-d0da59bb6909</ae:uuid>
</ae:group>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>IM Administrator</ae:displayName>
  <ae:id>8</ae:id>
  <ae:uuid>b77b6e9d-e75a-4841-be1b-9c358affb797</ae:uuid>
</ae:group>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>CR Approval Board Members</ae:displayName>
  <ae:id>13</ae:id>
  <ae:uuid>ea167b8c-e9ea-4196-9727-dfd2f10fd751</ae:uuid>
</ae:group>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>CR Submitters</ae:displayName>
  <ae:id>11</ae:id>
  <ae:uuid>cf83a358-d7fb-4b96-8f98-ed532c66cd0a</ae:uuid>
</ae:group>
<ae:fieldsMask>1</ae:fieldsMask>
<ae:notesMask>4</ae:notesMask>
<ae:changeHistoryMask>4</ae:changeHistoryMask>
<ae:browserMask>13635632</ae:browserMask>
<ae:preferredSolution xsi:type="ae:SolutionIdentifier">
  <ae:displayName>Incident Management</ae:displayName>
  <ae:id>2</ae:id>
  <ae:uuid>c6f06a70-4d06-42a4-a3a9-50b2120dca41</ae:uuid>
```

```
<ae:uniqueName>INCIDENT_MANAGEMENT</ae:uniqueName>
  <ae:tabName>Incident Mgmt</ae:tabName>
</ae:preferredSolution>
<ae:solutionData>
  <ae:solution xsi:type="ae:SolutionIdentifier">
    <ae:displayName>Incident Management</ae:displayName>
    <ae:id>2</ae:id>
    <ae:uuid>c6f06a70-4d06-42a4-a3a9-50b2120dca41</ae:uuid>
    <ae:uniqueName>INCIDENT_MANAGEMENT</ae:uniqueName>
    <ae:tabName>Incident Mgmt</ae:tabName>
  </ae:solution>
  <ae:homeReport xsi:type="ae:ReportIdentifier">
    <ae:displayName>Trend Of Incidents Closed On Initial Call</ae:displayName>
    <ae:id>38</ae:id>
    <ae:uuid>f8a5ce79-4b40-45cf-9f6d-735d060e90de</ae:uuid>
  </ae:homeReport>
</ae:solutionData>
<ae:solutionData>
  <ae:solution xsi:type="ae:SolutionIdentifier">
    <ae:displayName>Change Request Management</ae:displayName>
    <ae:id>3</ae:id>
    <ae:uuid>0ea28a74-6dde-406f-b19a-4c45aec40294</ae:uuid>
    <ae:uniqueName>CHANGE_REQUEST_MANAGEMENT</ae:uniqueName>
    <ae:tabName>Change Requests</ae:tabName>
  </ae:solution>
  <ae:homeReport xsi:type="ae:ReportIdentifier">
    <ae:displayName>Built-In: All Active Items I Own</ae:displayName>
    <ae:id>-6</ae:id>
    <ae:uuid>-6</ae:uuid>
  </ae:homeReport>
</ae:solutionData>
<ae:solutionData>
  <ae:solution xsi:type="ae:SolutionIdentifier">
    <ae:displayName>Issue Defect Management</ae:displayName>
    <ae:id>1</ae:id>
    <ae:uuid>cea0a86c-5d74-4e12-b8d6-9d6b90186f1e</ae:uuid>
    <ae:uniqueName>ISSUE_DEFECT_MANAGEMENT</ae:uniqueName>
    <ae:tabName>IDM</ae:tabName>
  </ae:solution>
  <ae:homeReport xsi:type="ae:ReportIdentifier">
    <ae:displayName>Built-In: All Active Items I Own</ae:displayName>
    <ae:id>-6</ae:id>
    <ae:uuid>-6</ae:uuid>
  </ae:homeReport>
</ae:solutionData>
</ae:user>
</ae:return>
<ae:return>
  <ae:status>
    <ae:status>IS-ERROR</ae:status>
    <ae:message>Invalid user Kath.</ae:message>
  </ae:status>
</ae:return>
<ae:return>
  <ae:user>
```

```
<ae:id xsi:type="ae:UserIdentifier">
  <ae:displayName>Laura Engineer</ae:displayName>
  <ae:id>5</ae:id>
  <ae:uuid>e0538593-21aa-4ca5-a229-473563c21470</ae:uuid>
  <ae:loginId>laura</ae:loginId>
</ae:id>
<ae:accessType>ACCESS-USER</ae:accessType>
<ae:email>laura@companyName.com</ae:email>
<ae:emailCC/>
<ae:timezone/>
<ae:offsetFromGMT>-25200000</ae:offsetFromGMT>
<ae:dstSavings>3600000</ae:dstSavings>
<ae:datePreference>DATE-FORMAT-FROM-LOCALE</ae:datePreference>
<ae:timePreference>TIME-FORMAT-12HOUR</ae:timePreference>
<ae:namespaceName>00000</ae:namespaceName>
<ae:phoneNumber/>
<ae:locale>en_US</ae:locale>
<ae:isDeleted>false</ae:isDeleted>
<ae:maxNotes>10</ae:maxNotes>
<ae:maxChangeHistory>10</ae:maxChangeHistory>
<ae:maxItemsPerPage>20</ae:maxItemsPerPage>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>Everyone</ae:displayName>
  <ae:id>1</ae:id>
  <ae:uuid>ade39c21-e7b2-4dcb-a231-d3d872671b59</ae:uuid>
</ae:group>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>IDM Team</ae:displayName>
  <ae:id>2</ae:id>
  <ae:uuid>634aed4a-bc98-49d3-86f0-6095c2f7b9b6</ae:uuid>
</ae:group>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>CR Technical Analysts</ae:displayName>
  <ae:id>12</ae:id>
  <ae:uuid>661c1c5b-2e1d-4ad1-9d43-60a3dad6f931</ae:uuid>
</ae:group>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>CR Submitters</ae:displayName>
  <ae:id>11</ae:id>
  <ae:uuid>cf83a358-d7fb-4b96-8f98-ed532c66cd0a</ae:uuid>
</ae:group>
<ae:fieldsMask>1</ae:fieldsMask>
<ae:notesMask>4</ae:notesMask>
<ae:changeHistoryMask>4</ae:changeHistoryMask>
<ae:browserMask>13635632</ae:browserMask>
<ae:preferredSolution xsi:type="ae:SolutionIdentifier">
  <ae:displayName>Issue Defect Management</ae:displayName>
  <ae:id>1</ae:id>
  <ae:uuid>cea0a86c-5d74-4e12-b8d6-9d6b90186f1e</ae:uuid>
  <ae:uniqueName>ISSUE_DEFECT_MANAGEMENT</ae:uniqueName>
  <ae:tabName>IDM</ae:tabName>
</ae:preferredSolution>
<ae:solutionData>
  <ae:solution xsi:type="ae:SolutionIdentifier">
    <ae:displayName>Issue Defect Management</ae:displayName>
```

```
        <ae:id>1</ae:id>
        <ae:uuid>cea0a86c-5d74-4e12-b8d6-9d6b90186f1e</ae:uuid>
        <ae:uniqueName>ISSUE_DEFECT_MANAGEMENT</ae:uniqueName>
        <ae:tabName>IDM</ae:tabName>
    </ae:solution>
    <ae:homeReport xsi:type="ae:ReportIdentifier">
        <ae:displayName>My In Box</ae:displayName>
        <ae:id>54</ae:id>
        <ae:uuid>034982d7-822e-421e-88cf-1fe2cb436785</ae:uuid>
    </ae:homeReport>
</ae:solutionData>
<ae:solutionData>
    <ae:solution xsi:type="ae:SolutionIdentifier">
        <ae:displayName>Incident Management</ae:displayName>
        <ae:id>2</ae:id>
        <ae:uuid>c6f06a70-4d06-42a4-a3a9-50b2120dca41</ae:uuid>
        <ae:uniqueName>INCIDENT_MANAGEMENT</ae:uniqueName>
        <ae:tabName>Incident Mgmt</ae:tabName>
    </ae:solution>
    <ae:homeReport xsi:type="ae:ReportIdentifier">
        <ae:displayName>Built-In: All Items I Submitted</ae:displayName>
        <ae:id>-11</ae:id>
        <ae:uuid>-11</ae:uuid>
    </ae:homeReport>
</ae:solutionData>
<ae:solutionData>
    <ae:solution xsi:type="ae:SolutionIdentifier">
        <ae:displayName>Change Request Management</ae:displayName>
        <ae:id>3</ae:id>
        <ae:uuid>0ea28a74-6dde-406f-b19a-4c45aec40294</ae:uuid>
        <ae:uniqueName>CHANGE_REQUEST_MANAGEMENT</ae:uniqueName>
        <ae:tabName>Change Requests</ae:tabName>
    </ae:solution>
    <ae:homeReport xsi:type="ae:ReportIdentifier">
        <ae:displayName>Built-In: All Items I Submitted</ae:displayName>
        <ae:id>-11</ae:id>
        <ae:uuid>-11</ae:uuid>
    </ae:homeReport>
</ae:solutionData>
</ae:user>
</ae:return>
</ae:GetUsersResponse>
```

UserInfo

Description

The UserInfo type holds the name and other information about a user in SBM. The UserInfo type parameters are listed below.

Parameters

Name	Type	Description
id	UserIdentifier [page 138]	The login ID of the user.
accessType	AccessType [page 169]	Shows the user's product-access type.
email	string	The user's primary email address.
emailCC	string	The user's CC email addresses.
timezone	string	The user's time zone.
offsetFromGMT	integer	This is the time zone's raw GMT offset.
dstSavings	integer	The amount of time in milliseconds to be added to local standard time to get local wall clock time.
datePreference	DatePreference [page 171]	The user's date preference.
timePreference	TimePreference [page 196]	The user's time preference.
namespaceName	string	The user's namespace name.
phoneNumber	string	The user's phone number.
locale	string	The user's designated locale.
isDeleted	boolean	Indicates whether the user is marked as deleted or not.
contact	ContactIdentifier [page 88]	Holds identification information for the user's associated contact record.
maxNotes	integer	Indicates the maximum number of notes to display on an item.
maxChangeHistory	integer	Indicates the maximum number of change history records to display on an item.
maxItemsPerPage	integer	Indicates the maximum number of items to display per page in the SBM User Workspace.

Name	Type	Description
group	GroupIdentifier [page 93]	Holds the identification information for groups to which the user belongs. One or more GroupIdentifiers are returned for each group.
fieldsMask	integer	Bit mask that indicates which field sections the user would like displayed as described in the TS_USERS table in the <i>Database Schema Reference</i> .
notesMask	integer	Bit mask that indicates the user's preference for displaying notes as described in the TS_USERS table in the <i>Database Schema Reference</i> .
changeHistoryMask	integer	Bit mask that indicates the user's preference for displaying change history information as described in the TS_USERS table in the <i>Database Schema Reference</i> .
browserMask	integer	Bit mask that indicates the user's browser preferences as described in the TS_USERS table in the <i>Database Schema Reference</i> .
preferredSolution	SolutionIdentifier [page 106]	Holds the identification information for the user's preferred application. This setting determines which application tab is selected by default when the user first enters the SBM User Workspace.
solutionData	UserSolutionData [page 211]	Contains the ordered-list of application tabs that appear for the user in the SBM User Workspace. The application identification information, home page report, and user's preferred projects are returned for each application.

Usage

The UserInfo type provides a user account in SBM. Use [GetUsers \[page 67\]](#) to retrieve a list of attributes for a specified user.

The timezone parameter is a programmatic ID; for example, "America/Los_Angeles". This ID is used to call up a specific real-world time zone. It corresponds to the IDs defined in the standard Olson data used by UNIX systems, and has the format continent/city or ocean/city.

The offsetFromGMT parameter is the time zone's raw GMT offset (i.e., the number of milliseconds to add to GMT to get local time, before taking Daylight Saving Time into account). If DST is in effect for a given date, use the dstSavings value to adjust this offset.

The `dstSavings` parameter is the amount of time in milliseconds to be added to local standard time to get local wall clock time. If Daylight Saving Time is not observed in this user's timezone, this value will be 0. This value should be used only to adjust a date/time that is within the DST observation period.

The `namespaceName` parameter returns the name of the namespace that was generated while provisioning the customer environment. If the user does not belong to a namespace, then the default namespace name ("00000") is returned.

XML

The following XML snippet shows the `UserInfo` type in the return element of the `GetUsers` response.

```
<ae:user>
  <ae:id xsi:type="ae:UserIdentifier">
    <ae:displayName>John Support Manager</ae:displayName>
    <ae:id>21</ae:id>
    <ae:uuid>08784a43-970f-4d28-9a6e-c301077ca653</ae:uuid>
    <ae:loginId>john</ae:loginId>
  </ae:id>
  <ae:accessType>ACCESS-USER</ae:accessType>
  <ae:email>john@companyName.com</ae:email>
  <ae:emailCC/>
  <ae:timezone/>
  <ae:offsetFromGMT>-25200000</ae:offsetFromGMT>
  <ae:dstSavings>3600000</ae:dstSavings>
  <ae:datePreference>DATE-FORMAT-FROM-LOCALE</ae:datePreference>
  <ae:timePreference>TIME-FORMAT-12HOUR</ae:timePreference>
  <ae:namespaceName>00000</ae:namespaceName>
  <ae:phoneNumber/>
  <ae:locale>en_US</ae:locale>
  <ae:isDeleted>>false</ae:isDeleted>
  <ae:maxNotes>10</ae:maxNotes>
  <ae:maxChangeHistory>10</ae:maxChangeHistory>
  <ae:maxItemsPerPage>20</ae:maxItemsPerPage>
  <ae:group xsi:type="ae:GroupIdentifier">
    <ae:displayName>Everyone</ae:displayName>
    <ae:id>1</ae:id>
    <ae:uuid>ade39c21-e7b2-4dcb-a231-d3d872671b59</ae:uuid>
  </ae:group>
  <ae:group xsi:type="ae:GroupIdentifier">
    <ae:displayName>IDM View Only</ae:displayName>
    <ae:id>5</ae:id>
    <ae:uuid>af38532d-e79c-495f-a3e6-f4bf784cc492</ae:uuid>
  </ae:group>
  <ae:group xsi:type="ae:GroupIdentifier">
    <ae:displayName>IM Technician</ae:displayName>
    <ae:id>6</ae:id>
    <ae:uuid>902a1300-6ae1-44d7-b46a-e420babe8497</ae:uuid>
  </ae:group>
  <ae:group xsi:type="ae:GroupIdentifier">
    <ae:displayName>IM Manager</ae:displayName>
    <ae:id>7</ae:id>
    <ae:uuid>b70af5de-6642-4228-b9bb-d0da59bb6909</ae:uuid>
  </ae:group>
</ae:user>
```

```
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>IM Administrator</ae:displayName>
  <ae:id>8</ae:id>
  <ae:uuid>b77b6e9d-e75a-4841-be1b-9c358affb797</ae:uuid>
</ae:group>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>CR Approval Board Members</ae:displayName>
  <ae:id>13</ae:id>
  <ae:uuid>ea167b8c-e9ea-4196-9727-dfd2f10fd751</ae:uuid>
</ae:group>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>CR Submitters</ae:displayName>
  <ae:id>11</ae:id>
  <ae:uuid>cf83a358-d7fb-4b96-8f98-ed532c66cd0a</ae:uuid>
</ae:group>
<ae:fieldsMask>1</ae:fieldsMask>
<ae:notesMask>4</ae:notesMask>
<ae:changeHistoryMask>4</ae:changeHistoryMask>
<ae:browserMask>13635632</ae:browserMask>
<ae:preferredSolution xsi:type="ae:SolutionIdentifier">
  <ae:displayName>Incident Management</ae:displayName>
  <ae:id>2</ae:id>
  <ae:uuid>c6f06a70-4d06-42a4-a3a9-50b2120dca41</ae:uuid>
  <ae:uniqueName>INCIDENT_MANAGEMENT</ae:uniqueName>
  <ae:tabName>Incident Mgmt</ae:tabName>
</ae:preferredSolution>
<ae:solutionData>
  <ae:solution xsi:type="ae:SolutionIdentifier">
    <ae:displayName>Incident Management</ae:displayName>
    <ae:id>2</ae:id>
    <ae:uuid>c6f06a70-4d06-42a4-a3a9-50b2120dca41</ae:uuid>
    <ae:uniqueName>INCIDENT_MANAGEMENT</ae:uniqueName>
    <ae:tabName>Incident Mgmt</ae:tabName>
  </ae:solution>
  <ae:homeReport xsi:type="ae:ReportIdentifier">
    <ae:displayName>Trend Of Incidents Closed On Initial Call</ae:displayName>
    <ae:id>38</ae:id>
    <ae:uuid>f8a5ce79-4b40-45cf-9f6d-735d060e90de</ae:uuid>
  </ae:homeReport>
</ae:solutionData>
<ae:solutionData>
  <ae:solution xsi:type="ae:SolutionIdentifier">
    <ae:displayName>Change Request Management</ae:displayName>
    <ae:id>3</ae:id>
    <ae:uuid>0ea28a74-6dde-406f-b19a-4c45aec40294</ae:uuid>
    <ae:uniqueName>CHANGE_REQUEST_MANAGEMENT</ae:uniqueName>
    <ae:tabName>Change Requests</ae:tabName>
  </ae:solution>
  <ae:homeReport xsi:type="ae:ReportIdentifier">
    <ae:displayName>Built-In: All Active Items I Own</ae:displayName>
    <ae:id>-6</ae:id>
    <ae:uuid>-6</ae:uuid>
  </ae:homeReport>
</ae:solutionData>
<ae:solutionData>
```

```

    <ae:solution xsi:type="ae:SolutionIdentifier">
      <ae:displayName>Issue Defect Management</ae:displayName>
      <ae:id>1</ae:id>
      <ae:uuid>cea0a86c-5d74-4e12-b8d6-9d6b90186f1e</ae:uuid>
      <ae:uniqueName>ISSUE_DEFECT_MANAGEMENT</ae:uniqueName>
      <ae:tabName>IDM</ae:tabName>
    </ae:solution>
    <ae:homeReport xsi:type="ae:ReportIdentifier">
      <ae:displayName>Built-In: All Active Items I Own</ae:displayName>
      <ae:id>-6</ae:id>
      <ae:uuid>-6</ae:uuid>
    </ae:homeReport>
  </ae:solutionData>
</ae:user>

```

UserSolutionData

Description

The UserSolutionData type holds identification information and user settings for a solution (also known as an application) in SBM. The UserSolutionData type parameters are listed below.

Parameters

Name	Type	Description
solution	SolutionIdentifier [page 106]	Holds the identification information for a solution.
homeReport	ReportIdentifier [page 104]	Holds the identification information for a report. The report that is returned indicates the user's home page report.
preferredProject	ProjectIdentifier [page 100]	Holds the identification for a project. One or more projects are returned. The list of projects indicates the user's preferred projects.

Usage

The UserSolutionData type holds the SolutionIdentifier and additional information used to describe the settings for an application that a user has access to in the SBM User Workspace. You can use [GetUsers \[page 67\]](#) to retrieve the user solution data shown here. The GetUsers response contains one or more UserSolutionData elements that comprise an ordered-list of the application tabs that are available to the user in the SBM User Workspace.

XML

The following XML snippet shows the UserSolutionData type in the return element of the GetUsers response.

```

<ae:solutionData>
  <ae:solution xsi:type="ae:SolutionIdentifier">

```

```
<ae:displayName>Incident Management</ae:displayName>
<ae:id>2</ae:id>
<ae:uuid>c6f06a70-4d06-42a4-a3a9-50b2120dca41</ae:uuid>
<ae:uniqueName>INCIDENT_MANAGEMENT</ae:uniqueName>
<ae:tabName>Incident Mgmt</ae:tabName>
</ae:solution>
<ae:homeReport xsi:type="ae:ReportIdentifier">
  <ae:displayName>Built-In: All Items I Submitted</ae:displayName>
  <ae:id>-11</ae:id>
  <ae:uuid>-11</ae:uuid>
</ae:homeReport>
</ae:solutionData>
<ae:solutionData>
  <ae:solution xsi:type="ae:SolutionIdentifier">
    <ae:displayName>Change Request Management</ae:displayName>
    <ae:id>3</ae:id>
    <ae:uuid>0ea28a74-6dde-406f-b19a-4c45aec40294</ae:uuid>
    <ae:uniqueName>CHANGE_REQUEST_MANAGEMENT</ae:uniqueName>
    <ae:tabName>Change Requests</ae:tabName>
  </ae:solution>
  <ae:homeReport xsi:type="ae:ReportIdentifier">
    <ae:displayName>Built-In: All Items I Submitted</ae:displayName>
    <ae:id>-11</ae:id>
    <ae:uuid>-11</ae:uuid>
  </ae:homeReport>
</ae:solutionData>
```

SBM Administrative Web Services

This section contains reference material for all of the SBM Administrative Web Services (as described in the sbmadminsolutions72 WSDL).

- [Calls Available \[page 212\]](#)
- [Common Types \[page 243\]](#)
- [Arguments \[page 269\]](#)
- [Responses \[page 282\]](#)

Calls Available

This section describes the Administrative Web service API calls that are available in SBM. These calls represent the Web service operations that are invoked from a client and performed on the SBM Application Engine Web Server. The calls receive one or more arguments from the client, perform an operation on the server, and return an XML response to the client when applicable.

The following table lists all supported calls in alphabetical order, followed by a brief description of each operation. Select a call to view detailed information including:

- **Description**
A brief description of the call.
- **Arguments**

A table describing the arguments for each call. Both simple and complex types are listed for each argument. For each complex type argument, you can click the argument name for a detailed description.

- **Response**

A brief description of the call's response. For each complex type response, you can click the response name for a detailed description.

- **Usage**

Any notes, additional details, and concerns regarding the call are addressed here.

- **Faults**

Possible error values are listed here.

- **XML**

An example of the actual XML payload that is sent is displayed here. The XML not only shows the call and its respective elements, you can also see detailed examples of each argument and how to format the expected data.

Calls Available

Call	Description
CreateGroups [page 214]	This service creates one or more new groups.
CreateProject [page 217]	This service creates a new project.
CreateUsers [page 220]	This service creates one or more new users.
DeleteProcessApp [page 224]	This service deletes a specified process app.
GetGroups [page 225]	This service returns one or more existing groups.
GetUsers [page 227]	This service returns one or more user records.
GetUserPrivileges [page 229]	This services returns a list of privileges for a specified user.
GetVersion [page 232]	This service returns the SBM version and build number.

Call	Description
HasUserPrivilege [page 232]	This service checks for a specified privilege by name.
IsValidUser [page 234]	This service determines whether a specified user is valid or not.
Logout [page 235]	This service releases any licenses and resources associated with the session.
UpdateGroups [page 236]	This service updates one or more existing groups.
UpdateUsers [page 239]	This service updates one or more existing users.

CreateGroups

Description

This service creates one or more new groups.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 270]	The Auth type supplies credentials and optionally, a host name for licensing. The userId and password can be specified with HTTP BASIC or WS-SECURITY instead.
templateGroup	GroupIdentifier [page 252]	Specifies an existing group that should be used as a template for creating new group. New groups are created based on the group account that you provide in the GroupIdentifier.
updateIfExists	boolean	Specify true to update any existing groups that are found with the same displayName.
group (required)	GroupInfo [page 251]	Describes information about a group. Specify one or more group arguments to create one or more new groups.

Argument	Type	Description
options	MultipleOptions [page 275]	Holds name value pairing for future elements and enumeration to determine if the service should stop on failure and send an error message, or continue processing.

Response

GroupHolder is returned. The GroupHolder response contains the GroupIdentifier and additional information about the group. For more specific information, see [GroupHolder \[page 283\]](#).

Usage

The CreateGroups call provides a method to create one or more new user groups, given the proper privileges. Use the following arguments to control how the new user groups are created:

- **templateGroup** — Enables you to specify an existing group in SBM to act as a template for each new group that you create. If you specify a template group as well as additional details in the group argument, the group parameters that you send will overwrite the values provided by the templateGroup.



Note: You can not use the templateGroup argument in combination with an existing group in the group argument to update an existing group (the system returns "Cannot create a duplicate group"). To update existing groups, see [UpdateGroups \[page 236\]](#).

If you do not want to create new groups based on an existing group, do not specify a group in the templateGroup argument. The templateGroup argument is ignored if the updateIfExists argument is set to true.

- **updateIfExists** — Enables you to specify whether existing groups are updated or not. When updateIfExists is true, each group with a display name that matches an existing group is updated based on the parameters sent in the group argument. For example, if the Engineering group already exists in SBM, then the group's current product access is replaced by the accessType value you set in the group argument for the Engineers groups. If you send an empty value and updateIfExists is set to true, then the group's existing value is not changed. To remove existing values, use the UpdateGroups call and set AllowEmptyValues to true. The templateGroup argument is ignored if updateIfExists is set to true.
- **group** — To create new groups, specify one or more unique group arguments. For each unique group argument that you send, an additional group is created in SBM. You must provide the displayName for each new group account. If updateIfExists is set to true and the group names that you send match existing group accounts, then matching groups are updated. If you specify a templateGroup instead, then any additional group information that you provide in the group argument overwrites the corresponding data supplied by the template account. For example, if you specify Engineering as a template group, then the Engineering group's product access and memo value are copied into each new group that you create unless you specifically overwrite these parameters in each group argument.

Faults

- Invalid database pointer.
- The user lacks sufficient permission to create a group.

XML

The following XML is a snippet of the payload that is sent with CreateGroups. In this example, two new groups are created without a template group.

```
<urn:CreateGroups>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:templateGroup>
    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
  </urn:templateGroup>
  <urn:updateIfExists></urn:updateIfExists>
  <urn:group>
    <urn:id>
      <urn:displayName>New Group 1</urn:displayName>
      <urn:id></urn:id>
      <urn:uuid></urn:uuid>
    </urn:id>
    <urn:accessType>ACCESS-USER</urn:accessType>
    <urn:memo>This group contains regular users.</urn:memo>
    <urn:isDeleted></urn:isDeleted>
  </urn:group>
  <urn:group>
    <urn:id>
      <urn:displayName>New Group 2</urn:displayName>
      <urn:id></urn:id>
      <urn:uuid></urn:uuid>
    </urn:id>
    <urn:accessType>ACCESS-EXTERNAL</urn:accessType>
    <urn:memo>This group contains external users.</urn:memo>
    <urn:isDeleted></urn:isDeleted>
  </urn:group>
  <urn:options>
    <urn:multiOption>CONTINUE-ON-FAILURE</urn:multiOption>
  </urn:options>
</urn:CreateGroups>
```

In this example, the CR Submitters group is used as a template to create the new group:

```
<urn:CreateGroups>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
```

```

</urn:auth>
<urn:templateGroup>
  <urn:displayName>CR Submitters</urn:displayName>
  <urn:id></urn:id>
  <urn:uuid></urn:uuid>
</urn:templateGroup>
<urn:updateIfExists></urn:updateIfExists>
<urn:group>
  <urn:id>
    <urn:displayName>New Group</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
  </urn:id>
  <urn:accessType></urn:accessType>
  <urn:memo>This is a copy of the CR Submitters group.</urn:memo>
  <urn:isDeleted></urn:isDeleted>
</urn:group>
<urn:options>
  <urn:multiOption>CONTINUE-ON-FAILURE</urn:multiOption>
</urn:options>
</urn:CreateGroups>

```

CreateProject

Description

This service creates a new project.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 270]	The Auth type supplies credentials and optionally, a host name for licensing. The userId and password can be specified with HTTP BASIC or WS-SECURITY instead.
projectName (required)	string	The name of the new project. Note this is not the fully qualified project name.
parentProject (required)	ProjectIdentifier [page 255]	Used to specify the new project's parent project.
workflow (required)	WorkflowIdentifier [page 268]	The workflow is required if the Base Project is parent. Otherwise it is optional. The default value is the parent project's workflow.

Argument	Type	Description
useParentProjectWorkflow	boolean	Determines if the project should use the parent project's workflow. Default is true if parent is not Base Project. Default is false if parent is Base Project.
allowSubmit	boolean	Allow items to be submitted into the new project. Default is true .
useParentSequenceNumbers	boolean	Determines if items submitted to the new project are numbered in sequence with items in the parent project. False means the new project will number its items independently of items in the parent project. Default is true .
lastItemSequenceNumber	integer	The next item submitted to the new project will be numbered one greater than this number. 0 means the first item will be numbered 1. Default is 0 . Ignored if useParentSequenceNumbers is true.
zeroFillTo	integer	Zero-fill item numbers within the project to a certain number of digits. For example, 5 would fill to five digits: 00001. Default is 5 . Ignored if useParentSequenceNumbers is true.
allowAnonymousSubmit	boolean	Allow users without user accounts to submit items into the project. Default is false . Ignored if allowSubmit is false.
altName	string	Alternate project name to display to users who do not have view privileges on the project. Default is same as projectName.
description	string	Project description. Default is empty string.
options	Options [page 274]	Holds name value pairing for future arguments.

Response

ProjectData for the newly added project is returned. For more detail, see [ProjectGeneralData \[page 289\]](#).

Usage

The CreateProject call provides a method to add a single project, given the proper privileges. A successfully created project will be added underneath the specified parent project. If sibling projects exist underneath the parent, the newly created project will be placed last. If the call fails, the project is not added.

Faults

- Invalid database pointer.
- The user lacks sufficient permission to create a project.
- The Parent project is not found.
- Workflow is not specified when parent project is Base Project.
- ProjectName is empty, too long, or not unique among siblings.
- Alternate name too long.
- Description too long.
- Workflow type, that is table ID, for the new project's workflow differs from table ID for parent project's workflow.
- zeroFillTo out of range.
- lastItemSequenceNumber out of range.
- Database failure.

XML

The following XML is a snippet of the payload that is sent with CreateProject.

```
<urn:CreateProject>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:projectName>My new project</urn:projectName>
  <urn:parentProject>
    <urn:displayName>IDM Project</urn:displayName>
    <urn:id>2</urn:id>
    <urn:uuid></urn:uuid>
    <urn:fullyQualifiedName></urn:fullyQualifiedName>
  </urn:parentProject>
  <urn:workflow>
    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
```

```

    <urn:uuid></urn:uuid>
  </urn:workflow>
  <urn:useParentProjectWorkflow>true</urn:useParentProjectWorkflow>
  <urn:allowSubmit>true</urn:allowSubmit>
  <urn:useParentSequenceNumbers>true</urn:useParentSequenceNumbers>
  <urn:lastItemSequenceNumber>0</urn:lastItemSequenceNumber>
  <urn:zeroFillTo>5</urn:zeroFillTo>
  <urn:allowAnonymousSubmit>>false</urn:allowAnonymousSubmit>
  <urn:altName></urn:altName>
  <urn:description>This is a description.</urn:description>
</urn:CreateProject>

```

CreateUsers

Description

This services creates one or more new users.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 270]	The Auth type supplies credentials and optionally, a host name for licensing. The userId and password can be specified with HTTP BASIC or WS-SECURITY instead.
templateUser	UserIdentifier [page 260]	Specifies a user account that should be used as a template for creating new users. New users are created based on the user account that you provide in the UserIdentifier.
cloneGroups	boolean	Specify true to create one or more users with the same group membership as the templateUser. Default is false.
updateIfExists	boolean	Specify true to update any existing user accounts that are found with the same displayName and loginId.
user (required)	UserInfo [page 261]	Describes information about a user. Specify one or more user arguments to create one or more new users.
options	UserResponseOptions [page 280]	Specifies whether the service should continue if an error is encountered or stop. Also enables you to limit the data that is returned in the response.

Response

NewUser is returned. The NewUser response contains one UserHolder element for each new user that is created. A temporary password is also returned for each new user account. For more specific information, see [NewUser \[page 285\]](#).

Usage

The CreateUsers call enables you create one or more new user accounts in SBM. Use the following arguments to control how the new user accounts are created:

- **templateUser** — Enables you to specify an existing user account in SBM to act as a template for each new user account that you create. If you specify a template user as well as additional user details in the user argument, the user parameters that you send will overwrite the values provided by the templateUser.



Note: You can not use the templateUser argument in combination with an existing user in the user argument to update an existing user account (the system returns "Cannot create a duplicate user"). To update existing user accounts, see [UpdateUsers \[page 239\]](#).

If you do not want to create new user accounts based on an existing account, do not specify a user in the templateUser argument. The templateUser argument is ignored if the updateIfExists argument is set to true.

- **cloneGroups** — Enables you to specify whether or not the group membership of the template user should be replicated for each new user account that you create. If you specify false, new users are created without group membership unless you specify one or more groups in the group element of the user argument. If you specify true, the group membership is cloned and the new user is also added to any additional groups that are specified in the user argument.
- **updateIfExists** — Enables you to specify whether existing user accounts are updated or not. When updateIfExists is true, each user with a login ID or display name that matches an existing user is updated based on the parameters sent in the user argument. For example, if Bill's login ID already exists in SBM, then Bill's e-mail address is replaced by the e-mail address you set in the user argument for Bill. If you send an empty e-mail address and updateIfExists is set to true, then Bill's existing e-mail address is not changed. To remove existing values, use the UpdateUsers call and set AllowEmptyValues to true. The templateUser argument is ignored if updateIfExists is set to true.
- **user** — To create new users, specify one or more unique user arguments. For each unique user argument that you send, an additional user is created in SBM. You must provide both the displayName and loginId for each new user account. If updateIfExists is set to true and the user IDs that you send match existing user accounts, the matching users are updated. If you specify a templateUser account instead, then any additional user information that you provide in the user argument overwrites the corresponding data supplied by the template account. For example, if you specify Joe as a template user, then Joe's product access, e-mail address, and phone number are copied into each new user account that you create unless you specifically overwrite these parameters in each user argument.

Use the following elements in the options parameter to control how this call is processed and what data is returned:

- **multiOption** — Enables you to specify whether the service should continue if an error is encountered, or stop and throw an error. If any failures occur, each successive error message is appended to the string that is returned. Multiple error

messages are separated by a single newline. Failures do not result in a return before all users have been processed.

- **sections** and **specifiedSections** — Enables you to specify which parts of a user record should be returned. This allows you to limit the data that is returned for a given user. The sections that aren't specified are not included in the response. For example, if you only need basic user information in the response, use the sections parameter to return only the STANDARD section.

For more information on the options elements, see [UserResponseOptions \[page 280\]](#).

Faults

- Invalid database pointer.
- The user ID is not valid.
- The user lacks sufficient permission.
- Create user requires a user login id.
- Create user requires a user display name.

XML

The following XML is a snippet of the payload this is sent with CreateUsers.

```
<urn:CreateUsers>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:templateUser>
    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:loginId></urn:loginId>
  </urn:templateUser>
  <urn:cloneGroups>>false</urn:cloneGroups>
  <urn:updateIfExists></urn:updateIfExists>
  <urn:user>
    <urn:id>
      <urn:displayName>John Doe</urn:displayName>
      <urn:id></urn:id>
      <urn:uuid></urn:uuid>
      <urn:loginId>JDoe</urn:loginId>
    </urn:id>
    <urn:accessType>ACCESS-USER</urn:accessType>
    <urn:email>johndoe@companyName.com</urn:email>
    <urn:emailCC></urn:emailCC>
    <urn:timezone></urn:timezone>
    <urn:offsetFromGMT>-25200000</urn:offsetFromGMT>
    <urn:dstSavings>3600000</urn:dstSavings>
    <urn:datePreference>DATE-FORMAT-FROM-LOCALE</urn:datePreference>
    <urn:timePreference>TIME-FORMAT-12HOUR</urn:timePreference>
```

```

    <urn:namespaceName></urn:namespaceName>
    <urn:phoneNumber>111-111-1111</urn:phoneNumber>
    <urn:locale>en_US</urn:locale>
    <urn:isDeleted></urn:isDeleted>
    <urn:contact></urn:contact>
    <urn:maxNotes>10</urn:maxNotes>
    <urn:maxChangeHistory>10</urn:maxChangeHistory>
    <urn:maxItemsPerPage>20</urn:maxItemsPerPage>
    <urn:group>
      <urn:displayName>CR Submitters</urn:displayName>
    </urn:group>
    <urn:fieldsMask></urn:fieldsMask>
    <urn:notesMask></urn:notesMask>
    <urn:changeHistoryMask></urn:changeHistoryMask>
    <urn:browserMask></urn:browserMask>
    <urn:preferredSolution>
      <urn:displayName>Issue Defect Management</urn:displayName>
    </urn:preferredSolution>
    <urn:solutionData>
      <urn:solution>
        <urn:displayName>Issue Defect Management</urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
        <urn:uniqueName></urn:uniqueName>
        <urn:tabName>IDM</urn:tabName>
      </urn:solution>
      <urn:homeReport>
        <urn:displayName>Built-In: All Items</urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
      </urn:homeReport>
      <urn:preferredProject>
        <urn:displayName>Animation Pro</urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
        <urn:fullyQualifiedName></urn:fullyQualifiedName>
      </urn:preferredProject>
    </urn:solutionData>
  </urn:user>
  <urn:options>
    <urn:multiOption></urn:multiOption>
    <urn:sections>SECTIONS-ALL</urn:sections>
    <urn:specifiedSections></urn:specifiedSections>
  </urn:options>
</urn:CreateUsers>

```

In this example, a new user is created based on Joe's existing user account. Joe's group membership is replicated as well:

```

<urn:CreateUsers>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>

```

```

    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:templateUser>
    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:loginId>Joe</urn:loginId>
  </urn:templateUser>
  <urn:cloneGroups>true</urn:cloneGroups>
  <urn:updateIfExists></urn:updateIfExists>
  <urn:user>
    <urn:id>
      <urn:displayName>John Doe</urn:displayName>
      <urn:id></urn:id>
      <urn:uuid></urn:uuid>
      <urn:loginId>JDoe</urn:loginId>
    </urn:id>
  </urn:user>
</urn:deleteProcessApp>

```

DeleteProcessApp

Description

This service deletes a specified process app and all its data, given the proper privilege.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 270]	The Auth type supplies credentials and optionally, a host name for licensing. The userId and password can be specified with HTTP BASIC or WS-SECURITY instead.
processAppName (required)	string	The name of the process app to be deleted.

Response

An empty XML response is returned:

```
<ae:DeleteProcessAppResponse></ae:DeleteProcessAppResponse>
```

and the process app is deleted. Failure will not delete the process app.

Usage

This call permanently deletes the process app that you specify.



Note: DeleteProcessApp not only deletes the specified process app, but also all the data in that process app. All tables defined by the process app and all data in those tables is permanently deleted. Since the data cannot be recovered, you should consider performing a backup of your database prior to deleting the process app.

Faults

- Invalid database pointer.

- The process app name is not valid.
- The user lacks sufficient permission.
- The delete process app fails to execute.

XML

The following XML is a snippet of the payload being sent with DeleteProcessApp.

```
<urn:DeleteProcessApp>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname>localhost</urn:hostname>
  </urn:auth>
  <urn:processAppName>Incident Management</urn:processAppName>
  <urn:options></urn:options>
</urn:DeleteProcessApp>
```

GetGroups

Description

This services returns one or more existing groups.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 270]	The Auth type supplies credentials and optionally, a host name for licensing. The userId and password can be specified with HTTP BASIC or WS-SECURITY instead.
searchByName	string	Used to specify the name of the group as a search parameter.
group (required)	GroupIdentifier [page 252]	The group or groups that you wish to return.
options	MultipleOptions [page 275]	Holds name value pairing for future elements and enumeration to determine if the service should stop on failure and send an error message or continue processing.

Response

GroupHolder is returned. The GroupHolder response contains the GroupIdentifier and additional information about the group. For more specific information, see [GroupHolder \[page 283\]](#).

Usage

The GetGroups call is useful when you need data about a group. The GetGroups call retrieves data for a group as it exists in the TS_GROUPS table of the database. To retrieve information for a given user account, use [GetUsers \[page 227\]](#).

Use the following arguments to determine which groups are returned in the response:

- **searchByName** — Enables you to enter a single string to return groups. For example, if you enter Engineer as the searchByName value, groups with following display names are returned: Engineers, Engineering, EngineerExperts. You can not enter multiple strings.



Note: To return all user records in the database, do not provide a value for searchByName, and do not enter specific GroupIdentifiers in the group argument. If you have a large number of groups in your database, consider limiting the return data using one or more arguments.

- **group** — Enables you to specify one or more specific groups to be returned. If you want to return multiple groups, you must specify each desired group by providing the GroupIdentifier in a list of multiple group arguments. The group argument takes precedence over the searchByName argument.

Faults

- Invalid database pointer.
- The group ID is not valid.
- The user lacks sufficient permission.

XML

The following XML is a snippet of the payload this is sent with GetGroups.

```
<urn:GetGroups>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:searchByName></urn:searchByName>
  <urn:group>
    <urn:displayName>CR Submitters</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
  </urn:group>
  <urn:group>
    <urn:displayName>IDM Team</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
  </urn:group>
  <urn:options>
    <urn:multiOption></urn:multiOption>
  </urn:options>
</urn:GetGroups>
```

In this example, only group names that contain the string "IDM" will be returned:

```

<urn:GetGroups>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:searchByName>IDM</urn:searchByName>
  <urn:group>
    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
  </urn:group>
  <urn:options>
    <urn:multiOption></urn:multiOption>
  </urn:options>
</urn:GetGroups>

```

GetUsers

Description

This services returns one or more users.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 270]	The Auth type supplies credentials and optionally, a host name for licensing. The userId and password can be specified with HTTP BASIC or WS-SECURITY instead.
getCurrentUser	boolean	Specify true to return user info for the user specified in the auth argument of the GetUsers call. Specify false to use the other arguments to return users instead. Default value is false.
searchByName	string	Used to specify the name of the user as a search parameter.
user	UserIdentifier [page 260]	Used to return a specific user or list of users. Specify multiple UserIdentifiers to return multiple users.
options	UserResponseOptions [page 280]	Specifies whether the service should continue if an error is encountered or stop. Also enables you to limit the data that is returned in the response.

Response

UserHolder is returned. The UserHolder response contains one or more UserIdentifiers and additional information from the users preferences. For more specific information, see [UserHolder \[page 292\]](#).

Usage

The GetUsers call is useful when you need data about a user account. The GetUsers call retrieves data for a user account as it exists in the TS_USERS table of the database. To retrieve privileges for a given user account, use [GetUserPrivileges \[page 229\]](#). To determine if a specific user has a given privilege, use [HasUserPrivilege \[page 232\]](#).

Use the following arguments to determine which users are returned in the response:

- **getCurrentUser** — Enables you to specify whether the service should only return user information for the user that is making the call. If you set getCurrentUser to true, the current user is returned, regardless of the data sent in the searchByName and user arguments.
- **searchByName** — Enables you to enter a single string to return user accounts. For example, if you enter Joe as the searchByName value, user records with following display names are returned: Joe, Joel, Joey. You can not enter multiple strings. The searchByName argument is ignored if getCurrentUser is set to true or if a UserIdentifier is specified in the user argument.



Note: To return all user records in the database, leave getCurrentUser set to false, do not provide a value for searchByName, and do not enter specific UserIdentifiers in the user argument. If you have a large number of users in your database, consider limiting the return data using one or more arguments.

- **user** — Enables you to specify one or more specific user accounts to be returned. If you want to return multiple users, you must specify each desired user by providing the UserIdentifier in a list of multiple user arguments. The user argument takes precedence over the searchByName argument.

Use the following elements in the options parameter to control how this call is processed and what data is returned:

- **multiOption** — Enables you to specify whether the service should continue if an error is encountered, or stop and throw an error. If any failures occur, each successive error message is appended to the string that is returned. Multiple error messages are separated by a single newline. Failures do not result in a return before all users have been processed.
- **sections** and **specifiedSections** — Enables you to specify which parts of a user record should be returned. This allows you to limit the data that is returned for a given user. The sections that aren't specified are not included in the response. For example, if you only need basic user information in the response, use the sections parameter to return only the STANDARD section.

For more information on the options elements, see [UserResponseOptions \[page 280\]](#).

Faults

- Invalid database pointer.
- The user ID is not valid.

-
- The user lacks sufficient permission.

XML

The following XML is a snippet of the payload this is sent with `GetUsers`. In this example, `searchByName` is used to find all users with the string `Joe` in the user login ID.

```
<urn:GetUsers>
  <urn:auth>
    <urn:userId>Admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:getCurrentUser></urn:getCurrentUser>
  <urn:searchByName>Joe</urn:searchByName>
  <urn:user>
    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:loginId></urn:loginId>
  </urn:user>
</urn:GetUsers>
```

In this example, `getCurrentUser` is empty (or false), `searchByName` is left empty, and no user is provided in the user argument. This call returns all the users in the system.

```
<urn:GetUsers>
  <urn:auth>
    <urn:userId>Admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:getCurrentUser></urn:getCurrentUser>
  <urn:searchByName></urn:searchByName>
  <urn:user>
    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:loginId></urn:loginId>
  </urn:user>
</urn:GetUsers>
```

GetUserPrivileges

Description

This service returns a list of privileges for a specified user.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 270]	The Auth type supplies credentials and optionally, a host name for licensing. The userId and password can be specified with HTTP BASIC or WS-SECURITY instead.
privilegeType (optional)	string	Indicates the privilege is either a normal user privilege or an administrator privilege. The types of privilege for each are listed below.
objectId (optional)	string	The ID of the object you wish to check privileges against. If provided, privilegeType becomes required.
user (optional)	UserIdentifier [page 260]	Indicates the user for which privileges should be returned. If this parameter is not provided, the privilege query is made against the user account that sends the call.

Response

Privilege is returned. The privileges are returned in the context of either the calling user or the specified user. The privilege name, type, and object UUID (if the privilege applies only to a particular object like a project or table) are returned. For more information, see [Privilege \[page 287\]](#).

Usage

The objectId argument is used to limit the privilege check to a certain database object. For example, if want to query a table privilege, you specify the table ID of the table. If you want to query a project privilege, you specify the project ID. To check for system privileges, send an empty objectId argument. If the object ID is not composed of all numeric digits, the ID is interpreted as the UUID for the object. Note that if this parameter is provided, then the privilegeType argument is required.

If an invalid privilege type or no privilege type is specified at all, the call will succeed, but no privilege information will be returned. The tables below contain the possible privilege types.

The available privilege types for a normal user are:

TS_PRIVTYPE_USERSYS
TS_PRIVTYPE_USERPRJ
TS_PRIVTYPE_USERWKF
TS_PRIVTYPE_USERFLD
TS_PRIVTYPE_USERTBL

The available privilege types for an administrator are:

TS_PRIVTYPE_ADMSYS
TS_PRIVTYPE_ADMPRJ
TS_PRIVTYPE_ADMWKF
TS_PRIVTYPE_ADMFLD_PRJ
TS_PRIVTYPE_ADMFLD_WKF
TS_PRIVTYPE_ADMCON
TS_PRIVTYPE_UNKNOWN
TS_PRIVTYPE_SYSMASK
TS_PRIVTYPE_ADMMASK
TS_PRIVTYPE_TBLMASK

Faults

- Invalid database pointer.
- The privilege type name is not valid.
- The login ID is not valid.
- The user lacks sufficient permission.

XML

The following XML is a snippet of the payload that is sent with GetUserPrivileges.

```
<urn:GetUserPrivileges>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname>localhost</urn:hostname>
  </urn:auth>
  <urn:privilegeType>TS_PRIVTYPE_USERTBL</urn:privilegeType>
  <urn:objectId>1003</urn:objectId>
  <urn:user>
    <urn:displayName>Bill Admin</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:loginId></urn:loginId>
  </urn:user>
</urn:GetUserPrivileges>
```

Another example:

```
<urn:GetUserPrivileges>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
  </urn:auth>
  <urn:privilegeType>TS_PRIVTYPE_ADMCON</urn:privilegeType>
  <urn:objectId></urn:objectId>
  <urn:user>
    <urn:displayName>Bill Admin</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:loginId></urn:loginId>
  </urn:user>
</urn:GetUserPrivileges>
```

GetVersion

Description

This service returns the SBM version number.

Arguments

None.

Response

A string is returned, showing the version number. For example:

```
<ae:GetVersionResponse>
  <ae:return>Version 2010 R1.200</ae:return>
</ae:GetVersionResponse>
```

Usage

None.

Faults

None.

XML

The following XML shows the payload that is sent with GetVersion.

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:urn="urn:sbmappservices72">
  <soap:Header/>
  <soap:Body>
    <urn:GetVersion/>
  </soap:Body>
</soap:Envelope>
```

HasUserPrivilege

Description

This service checks for a specified privilege by name.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 270]	The <i>Auth</i> type supplies credentials and optionally, a host name for licensing. The <i>userId</i> and <i>password</i> can be specified with HTTP BASIC or WS-SECURITY instead.
privilegeName (required)	string	name of the privilege (for example, "TS_USRSYSPRIV_APIACCESS")
objectId (optional)	string	The ID of the object to check privileges against.
user	UserIdentifier [page 260]	The user to check privileges against.
options	Options [page 274]	Holds name value pairing for future arguments.

Response

A boolean is returned, showing whether the user has the privilege (true) or not (false). The privilege is checked in the context of either the calling user or the specified user. For example:

```
<ae:HasUserPrivilegeResponse>  
  <ae:return>true</ae:return>  
</ae:HasUserPrivilegeResponse>
```

Usage

You can call [GetUserPrivileges \[page 229\]](#) for the administrator account or for a user account (assuming it has all privileges) to return the name of every privilege. You can then use these privilege names in the *HasUserPrivilege* call to determine if other users have the same privilege.

The *objectId* argument is used to limit the privilege check to a certain database object. For example, if want to query a table privilege, you specify the table ID of the table. If want to query a project privilege, you specify the project ID. To check for system privileges, send an empty *objectID* argument. If the object ID is not composed of all numeric digits, the ID is interpreted as the UUID for the object.

Faults

- Invalid database pointer.
- The privilege name is not valid.
- The login ID is not valid.
- The user lacks sufficient permission.

XML

The following XML is a snippet of the payload this is sent with HasUserPrivilege.

```

<urn:HasUserPrivilege>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname>localhost</urn:hostname>
  </urn:auth>
  <urn:privilegeName>TS_ADMCONPRIV_DEPLOY_APPLICATION</urn:privilegeName>
  <urn:objectId></urn:objectId>
  <urn:user>
    <urn:displayName>Joe</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:loginId></urn:loginId>
  </urn:user>
</urn:HasUserPrivilege>

```

IsUserValid

Description

This service determines whether a specified user is valid.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 270]	The Auth type supplies credentials and optionally, a host name for licensing. The userId and password can be specified with HTTP BASIC or WS-SECURITY instead.
user (optional)	UserIdentifier [page 260]	The user to be checked.

Response

A boolean is returned, showing whether the user is valid (true) or not (false). The user account is checked in the context of either the calling user or the specified user. For example:

```

<ae:IsUserValidResponse>
  <ae:return>true</ae:return>
</ae:IsUserValidResponse>

```

Usage

IsUserValid will return true if the specified user was found in the database and is not deleted or disabled. Otherwise false is returned.

Faults

- Invalid database pointer.

- The login ID is not valid.
- The user lacks sufficient permission.

XML

The following XML is a snippet of the payload this is sent with IsUserValid.

```
<urn:IsUserValid>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
  </urn:auth>
  <urn:user>
    <urn:displayName>carmen</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:loginId></urn:loginId>
  </urn:user>
</urn:IsUserValid>
```

Logout

Description

This service releases any licenses and resources associated with the session.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 270]	The Auth type supplies credentials and optionally, a host name for licensing. The userId and password can be specified with HTTP BASIC or WS-SECURITY instead.
options	Options [page 274]	Holds name value pairing for future arguments.

Response

An empty XML response is returned:

```
<ae:LogoutResponse/>
```

and the session is ended. Failure will keep the session open.

Usage

The Logout call logs out the user from the current active session. There is no effect if previous Web service calls are not made before Logout is called.

Faults

- Authentication error if invalid credentials.

XML

The following XML is a snippet of the payload that is sent with Logout.

```

<urn:Logout>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
  </urn:auth>
</urn:Logout>

```

UpdateGroups

Description

This services updates one or more existing groups.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 270]	The Auth type supplies credentials and optionally, a host name for licensing. The userId and password can be specified with HTTP BASIC or WS-SECURITY instead.
allowEmptyValues	boolean	Specify true in order to honor empty values that are sent in the group argument. (This removes the current value that is specified on the group account). Default is false.
group (required)	GroupInfo [page 251]	Used to identify a specific group or list of groups for update. Specify multiple group arguments to update multiple groups.
options	MultipleOptions [page 275]	Holds name value pairing for future elements and enumeration to determine if the service should stop on failure and send an error message or continue processing.

Response

GroupHolder is returned. The GroupHolder response contains one or more GroupIdentifiers and additional information from the groups record. For more specific information, see [GroupHolder \[page 283\]](#).

Usage

Use the UpdateGroups call when you need to update one or more groups in SBM. Use the following arguments to handle how groups are updated by the service:

- **allowEmptyValues** — Enables you to send an empty value in the group argument that is honored by the update. For example, if you need to remove the memo for one

or more groups, set `allowEmptyValues` to true and send an empty value in the group element of each group argument you send. The default value is false, which means that empty values in the group argument are not honored.



CAUTION: Use care when updating existing group accounts. Important group information might be mistakenly altered or removed if an existing group match is found and the group parameters you send are incorrect or empty. For example, to delete the memo for one or more groups, do not just clear the memo value and set `allowEmptyValues` to true. This will remove all the other group settings including the product access (which will be set to None if you send an empty value). When you want to clear values using `allowEmptyValues`, you must also provide values for elements in `GroupInfo` that you want to preserve

- **group** — Enables you to specify one or more specific group accounts to be updated. If you want to update multiple groups, you must specify each desired group by providing the `GroupIdentifier` in a list of multiple group arguments. To delete groups, specify true in the `isDeleted` element of the group argument.

Faults

- Invalid database pointer.
- The user ID is not valid.
- The user lacks sufficient permission.

XML

The following XML is a snippet of the payload this is sent with `UpdateGroups`.

```
<urn:UpdateGroups>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:allowEmptyValues>>false</urn:allowEmptyValues>
  <urn:group>
    <urn:id>
      <urn:displayName>CR Submitters</urn:displayName>
      <urn:id></urn:id>
      <urn:uuid></urn:uuid>
    </urn:id>
    <urn:accessType>ACCESS-ADMIN</urn:accessType>
    <urn:memo>Changed this group to admin access.</urn:memo>
    <urn:isDeleted></urn:isDeleted>
  </urn:group>
  <urn:group>
    <urn:id>
      <urn:displayName>IDM Team</urn:displayName>
      <urn:id></urn:id>
      <urn:uuid></urn:uuid>
    </urn:id>
    <urn:accessType>ACCESS-USER</urn:accessType>
```

```
    <urn:memo>This team has user access now.</urn:memo>
    <urn:isDeleted></urn:isDeleted>
  </urn:group>
<urn:options>
  <urn:multiOption>CONTINUE-ON-FAILURE</urn:multiOption>
</urn:options>
</urn:UpdateGroups>
```

In this example, the current memo is deleted on the CR Submitters group:

```
<urn:UpdateGroups>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:allowEmptyValues>true</urn:allowEmptyValues>
  <urn:group>
    <urn:id>
      <urn:displayName>CR Submitters</urn:displayName>
    </urn:id>
    <urn:accessType>ACCESS-ADMIN</urn:accessType>
    <urn:memo></urn:memo>
    <urn:isDeleted></urn:isDeleted>
  </urn:group>
  <urn:options>
    <urn:multiOption>CONTINUE-ON-FAILURE</urn:multiOption>
  </urn:options>
</urn:UpdateGroups>
```

In this example, the CR Submitters group is deleted:

```
<urn:UpdateGroups>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:allowEmptyValues>>false</urn:allowEmptyValues>
  <urn:group>
    <urn:id>
      <urn:displayName>CR Submitters</urn:displayName>
    </urn:id>
    <urn:accessType>ACCESS-ADMIN</urn:accessType>
    <urn:memo></urn:memo>
    <urn:isDeleted>true</urn:isDeleted>
  </urn:group>
  <urn:options>
    <urn:multiOption>CONTINUE-ON-FAILURE</urn:multiOption>
  </urn:options>
</urn:UpdateGroups>
```

UpdateUsers

Description

This services updates one or more existing users.

Arguments

Argument	Type	Description
auth (optional)	Auth [page 270]	The Auth type supplies credentials and optionally, a host name for licensing. The userId and password can be specified with HTTP BASIC or WS-SECURITY instead.
allowEmptyValues	boolean	Specify true in order to honor empty values that are sent in the user argument. (This removes the current value that is specified on the user account). Default is false.
groupAction	GroupListAction [page 272]	Specifies whether to add, replace, or remove groups.
user (required)	UserInfo [page 261]	Used to identify a specific user or list of users for update. Specify multiple user arguments to update multiple users.
options	UserResponseOptions [page 280]	Specifies whether the service should continue if an error is encountered or stop. Also enables you to limit the data that is returned in the response.

Response

UserHolder is returned. The UserHolder response contains one or more UserIdentifiers and additional information from the users preferences. For more specific information, see [UserHolder \[page 292\]](#).

Usage

Use the UpdateUsers call when you need to update one or more user accounts in SBM. Use the following arguments to handle how users are updated by the service:

- **allowEmptyValues** — Enables you to send an empty value in the user argument that is honored by the update. For example, if you need to remove the e-mail address for one or more users, set `allowEmptyValues` to `true` and send an empty value in the `phoneNumber` element of each user argument you send. The default value is `false`, which means that empty values in the user argument are not honored.



CAUTION: Use care when updating existing user accounts. Important user information might be mistakenly altered or removed if an existing user match is found and the user parameters you send are incorrect or empty. For example, to delete the phone number for one or more users, do not just clear the phone number value and set `allowEmptyValues` to `true`. This will remove all the other user preference settings including the product access (which will be set to `None` if you send an empty value). When you want to clear values using `allowEmptyValues`, you must also provide values for elements in `UserInfo` that you want to preserve.

- **groupAction** — Enables you to add, replace, or remove existing group membership for a user account on update. If you specify `ADD-GROUPS`, then the user added to each group that is specified in the user argument. If you specify `REPLACE-GROUPS`, then each group that is specified in the user argument replaces the current group membership designated for the user. If you specify `REMOVE-GROUPS`, then the user is removed from the groups that are specified in the user argument.
- **user** — Enables you to specify one or more specific user accounts to be updated. If you want to update multiple users, you must specify each desired user by providing the `UserIdentifier` in a list of multiple user arguments. To delete users, specify `true` in the `isDeleted` element of the user argument.

Use the following elements in the options parameter to control how this call is processed and what data is returned:

- **multiOption** — Enables you to specify whether the service should continue if an error is encountered, or stop and throw an error. If any failures occur, each successive error message is appended to the string that is returned. Multiple error messages are separated by a single newline. Failures do not result in a return before all users have been processed.
- **sections** and **specifiedSections** — Enables you to specify which parts of a user record should be returned. This allows you to limit the data that is returned for a given user. The sections that aren't specified are not included in the response. For example, if you only need basic user information in the response, use the `sections` parameter to return only the `STANDARD` section.

For more information on the options elements, see [UserResponseOptions \[page 280\]](#).

Faults

- Invalid database pointer.
- The user ID is not valid.
- The user lacks sufficient permission.

XML

The following XML is a snippet of the payload this is sent with `UpdateUsers`. This XML shows an update against Nancy's user account. In this example, Nancy is added to the CR

Submitters group, her preferred application is set to IDM, and the tab order is set to begin with IDM when she logs in.

```
<urn:UpdateUsers>
  <urn:auth>
    <urn:userId>Admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:allowEmptyValues>>false</urn:allowEmptyValues>
  <urn:groupAction>ADD-GROUPS</urn:groupAction>
  <urn:user>
    <urn:id>
      <urn:displayName>Nancy</urn:displayName>
      <urn:id></urn:id>
      <urn:uuid></urn:uuid>
      <urn:loginId>Nancy</urn:loginId>
    </urn:id>
    <urn:accessType>ACCESS-USER</urn:accessType>
    <urn:email></urn:email>
    <urn:emailCC></urn:emailCC>
    <urn:timezone></urn:timezone>
    <urn:offsetFromGMT>-2520000</urn:offsetFromGMT>
    <urn:dstSavings>360000</urn:dstSavings>
    <urn:datePreference>DATE-FORMAT-FROM-LOCALE</urn:datePreference>
    <urn:timePreference>TIME-FORMAT-12HOUR</urn:timePreference>
    <urn:namespaceName></urn:namespaceName>
    <urn:phoneNumber>888-888-8888</urn:phoneNumber>
    <urn:locale>en_US</urn:locale>
    <urn:isDeleted></urn:isDeleted>
    <urn:contact></urn:contact>
    <urn:maxNotes>10</urn:maxNotes>
    <urn:maxChangeHistory>10</urn:maxChangeHistory>
    <urn:maxItemsPerPage>20</urn:maxItemsPerPage>
    <urn:group>
      <urn:displayName>CR Submitters</urn:displayName>
      <urn:id></urn:id>
      <urn:uuid></urn:uuid>
    </urn:group>
    <urn:fieldsMask>1</urn:fieldsMask>
    <urn:notesMask>4</urn:notesMask>
    <urn:changeHistoryMask>4</urn:changeHistoryMask>
    <urn:browserMask></urn:browserMask>
    <urn:preferredSolution>
      <urn:displayName></urn:displayName>
      <urn:id></urn:id>
      <urn:uuid></urn:uuid>
      <urn:uniqueName></urn:uniqueName>
      <urn:tabName>IDM</urn:tabName>
    </urn:preferredSolution>
    <urn:solutionData>
      <urn:solution>
        <urn:displayName></urn:displayName>
        <urn:id></urn:id>
      </urn:solution>
    </urn:solutionData>
  </urn:user>
</urn:UpdateUsers>
```

```
        <urn:uuid></urn:uuid>
        <urn:uniqueName>ISSUE_DEFECT_MANAGEMENT</urn:uniqueName>
        <urn:tabName>IDM</urn:tabName>
    </urn:solution>
    <urn:homeReport>
        <urn:displayName>Built-In: All Active Items I Own</urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
    </urn:homeReport>
    <urn:preferredProject>
        <urn:displayName>Animation Pro</urn:displayName>
        <urn:id>6</urn:id>
        <urn:uuid></urn:uuid>
        <urn:fullyQualifiedName></urn:fullyQualifiedName>
    </urn:preferredProject>
</urn:solutionData>
<urn:solutionData>
    <urn:solution>
        <urn:displayName></urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
        <urn:uniqueName>INCIDENT_MANAGEMENT</urn:uniqueName>
        <urn:tabName>Incident Mgmt</urn:tabName>
    </urn:solution>
    <urn:homeReport>
        <urn:displayName>Built-In: All Active Items I Own</urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
    </urn:homeReport>
    <urn:preferredProject>
        <urn:displayName>IM Project</urn:displayName>
        <urn:id>3</urn:id>
        <urn:uuid></urn:uuid>
        <urn:fullyQualifiedName></urn:fullyQualifiedName>
    </urn:preferredProject>
</urn:solutionData>
</urn:user>
<urn:options>
    <urn:extraOption>
        <urn:name></urn:name>
        <urn:value></urn:value>
    </urn:extraOption>
    <urn:multiOption></urn:multiOption>
    <urn:sections>SECTIONS-ALL</urn:sections>
    <urn:specifiedSections></urn:specifiedSections>
</urn:options>
</urn:UpdateUsers>
```

In this example, Nancy and Carmen are removed from the CR Submitters group:

```
<urn:UpdateUsers>
  <urn:auth>
    <urn:userId>Admin</urn:userId>
    <urn:password></urn:password>
```

```

    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:allowEmptyValues>>false</urn:allowEmptyValues>
  <urn:groupAction>REMOVE-GROUPS</urn:groupAction>
  <urn:user>
    <urn:id>
      <urn:loginId>Nancy</urn:loginId>
    </urn:id>
    <urn:isDeleted></urn:isDeleted>
    <urn:group>
      <urn:displayName>CR Submitters</urn:displayName>
      <urn:id></urn:id>
      <urn:uuid></urn:uuid>
    </urn:group>
  </urn:user>
  <urn:user>
    <urn:id>
      <urn:loginId>Carmen</urn:loginId>
    </urn:id>
    <urn:isDeleted></urn:isDeleted>
    <urn:group>
      <urn:displayName>CR Submitters</urn:displayName>
      <urn:id></urn:id>
      <urn:uuid></urn:uuid>
    </urn:group>
  </urn:user>
</urn:UpdateUsers>

```

In this example, the John Doe is deleted from the system:

```

<urn:UpdateUsers>
  <urn:auth>
    <urn:userId>admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:allowEmptyValues>>false</urn:allowEmptyValues>
  <urn:groupAction>ADD-GROUPS</urn:groupAction>
  <urn:user>
    <urn:id>
      <urn:displayName>John Doe</urn:displayName>
      <urn:loginId>JDoe</urn:loginId>
    </urn:id>
    <urn:isDeleted>true</urn:isDeleted>

```

Common Types

This section provides detailed descriptions of SBM Administrative Web service types that appear as both arguments and response elements. The types listed here contain one or more parameters, which make up the data being sent to or returned by the Web service. The parameters listed in each types are either simple or complex types themselves. If the type is complex, a link to further detail of that type will be provided in the **Type** column.

The following table lists all supported arguments in alphabetical order, followed by a brief description of each type. Select an argument to view detailed information including:

- **Description**

A brief description of the argument.

- **Parameters**

A table describing the types for each argument. Both simple and complex types are listed for each argument. For each complex type, you can click the type name for a detailed description.

- **Usage**

Any notes, additional details, and concerns regarding the argument are addressed here.

- **XML**

An example of the actual XML being sent is displayed here. The XML not only shows the argument and its respective elements, you can also see detailed examples of each element and how to format the expected data.

Common Types

Type	Description
AccessType [page 246]	Holds the user's product access type.
ApplicationIdentifier [page 246]	Holds the complete identification information for an application.
Attachment-Access-Type [page 247]	Indicates the type of attachment on an item.
ContactIdentifier [page 248]	Holds the complete identification information for a contact.
DatePreference [page 249]	Indicates a user's preferred date format.
FieldIdentifier [page 249]	Holds the complete identification information for a field.
FileBufferBase64 [page 250]	Holds the binary contents of a file in base64.
GroupIdentifier [page 252]	Holds the complete identification information for a group.

Type	Description
GroupInfo [page 251]	Holds the name and additional information about a user.
Identifier [page 253]	Holds generic identification information.
ItemIdentifier [page 254]	Holds the complete identification information for an item.
ProjectIdentifier [page 255]	Holds the complete identification information for a project.
ReportIdentifier [page 256]	Holds the complete identification information for a report.
SolutionIdentifier [page 256]	Holds the complete identification information for a solution.
StateIdentifier [page 257]	Holds the complete identification information for a state.
TableIdentifier [page 258]	Holds the complete identification information for a table.
TimePreference [page 259]	Indicates a user's preferred time format.
TransitionIdentifier [page 259]	Holds the complete identification information for a transition.
UserIdentifier [page 260]	Holds the complete identification information for a user.
UserInfo [page 261]	Holds the name and additional information about a user.

Type	Description
UserSolutionData [page 266]	Holds an ordered-list of applications that are accessible to a user. Also returns the user's home page report for each application and the preferred project list.
WorkflowIdentifier [page 268]	Holds the complete identification information for a workflow.

AccessType

Description

AccessType indicates the type of product-access a user is granted. The product access is used to determine the possible privileges that are available to users.

Parameters

Name	Type	Description
ACCESS-NONE	string	Indicates that the user's product access is set to None.
ACCESS-USER	string	Indicates Regular User product access.
ACCESS-OCCASIONAL	string	Indicates Occasional User product access.
ACCESS-EXTERNAL	string	Indicates External User product access.
ACCESS-ADMIN	string	Indicates Managed Administrator product access.
ACCESS-APIScript	string	Indicates API/Script product access.

Usage

The AccessType enumeration indicates the product access that is set for a user. For more information about each product access type, see the *SBM System Administrator Guide*.

XML

The following XML shows AccessType in the return element of the GetUsers response.

```
<ae:accessType>ACCESS-USER</ae:accessType>
```

ApplicationIdentifier

Description

The ApplicationIdentifier type holds the identification information for an application. The ApplicationIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 253] .	Generic identification extension base. Holds the displayName, id, and uuid for an application.

Usage

The ApplicationIdentifier is the identifier that can be used in Web service methods to uniquely identify an application. The ApplicationIdentifier contains the generic information about an application (including the display name, ID, and UUID).



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML snippet shows ApplicationIdentifier in the return element of a response.

```
<ae:return>
  <ae:application xsi:type="ae:ApplicationIdentifier">
    <ae:displayName>Incident Management</ae:displayName>
    <ae:id>3</ae:id>
    <ae:uuid>955e8e0e-9342-46ed-ba06-d1bfdc1cdf80</ae:uuid>
  </ae:application>
  <ae:description>Provides simple Incident Management
  →functionality for small Support teams.</ae:description>
  <ae:appDefUUID/>
  <ae:revision>2</ae:revision>
</ae:return>
```

Attachment-Access-Type

Description

Attachment-Access-Type indicates the type of attachment on an item. It is used for note, item link, URL, and file attachments. This type is used to determine whether the attachment has a restriction, is unrestricted, or if neither is set.

Parameters

Name	Type	Description
ATTACHACCESS-DEFAULT	string	Restrict the attachment only as specified by user privileges.
ATTACHACCESS-RESTRICTED	string	Makes the attachment visible only to users who can view the item.

Name	Type	Description
ATTACHACCESS-UNRESTRICTED	string	Makes the attachment visible to all users who can view the item.

Usage

The Unrestricted status makes the file visible to all users who can view the item. You can set a file to have Default status to restrict the file as specified by user privileges. The Unrestricted status is disabled if you do not have privileges to set file attachments as unrestricted for the selected project or auxiliary table.

XML

The following XML shows Attachment-Access-Type in the `<urn:accessType>` element of a typical call.

```
<urn:attachmentContents>
  <urn:id>16</urn:id>
  <urn:name>pdf_doc</urn:name>
  <urn:fileName>relnotes.pdf</urn:fileName>
  <urn:showAsImage>>false</urn:showAsImage>
  <urn:modificationDateTime></urn:modificationDateTime>
  <urn:url></urn:url>
  <urn:accessType>ATTACHACCESS-DEFAULT</urn:accessType>
  <urn:extendedData>
    <urn:data>
      <urn:name></urn:name>
      <urn:value></urn:value>
    </urn:data>
  </urn:extendedData>
  <urn:contentsBase64></urn:contentsBase64>
</urn:attachmentContents>
```

ContactIdentifier

Description

The ContactIdentifier type holds the identification information for a user's contact record. The ContactIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 253] .	Generic identification extension base. Holds the displayName, id, and uuid for a contact.

Usage

The ContactIdentifier is the identifier that can be used in Web service methods to uniquely identify a user's contact record. The ContactIdentifier contains the generic information about a contact (including the display name, ID, and UUID).



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows ContactIdentifier as seen in a typical call.

```
<urn:contact>
  <urn:displayName>Joe Manager</urn:displayName>
  <urn:id>1</urn:id>
  <urn:uuid>69d03cc6-e635-47d8-ab2e-a8be48a22f0a</urn:uuid>
</urn:contact>
```

DatePreference

Description

DatePreference indicates a user's preferred date format. The available options are listed below.

Parameters

Name	Type	Description
DATE-FORMAT-FROM-LOCALE	string	Use the format based on the user's locale.
DATE-FORMAT-MM-DD-YYYY	string	Use a MM-DD-YYYY format for dates.
DATE-FORMAT-DD-MM-YYYY	string	Use a DD-MM-YYYY format for dates.
DATE-FORMAT-DD-MM-YYYY.S	string	Use a DD.MM.YYYY format for dates.
DATE-FORMAT-YYYY-MM-DD	string	Use a YYYY-MM-DD format for dates.

Usage

DatePreference is used to determine how dates display to a user in the SBM User Workspace. The various date formats are returned in the datePreference parameter of the GetUsers response. See [UserInfo \[page 261\]](#) for more information.

XML

The following XML shows DatePreference as seen in the return element of the GetUsers call.

```
<ae:datePreference>DATE-FORMAT-MM-DD-YYYY</ae:datePreference>
```

FieldIdentifier

Description

The FieldIdentifier type holds the generic data for a field. The FieldIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 253] .	Generic identification extension base. Holds the displayName, id, and uuid for a field.
dbName	string	The unique database field name.

Usage

The FieldIdentifier is the identifier that can be used in Web service methods to uniquely identify a field. The FieldIdentifier contains the generic information about a field (including the display name, ID, and UUID) in addition to the database field name information for the field.



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows FieldIdentifier in the <urn:id> element in a typical call.

```
<urn:id>
  <urn:displayName>Actual Time to Fix</urn:displayName>
  <urn:id>59</urn:id>
  <urn:uuid>838fbaff-e74d-4d47-b415-85b502ea4676</urn:uuid>
  <urn:dbName>ACTUAL_TIME_TO_FIX</urn:dbName>
</urn:id>
```

FileBufferBase64

Description

The FileBufferBase64 type holds the actual contents of a file that you upload to the server in Base64. The FileBufferBase64 type parameters are listed below.

Parameters

Name	Type	Description
data	base64Binary	Holds the Base64 encoded contents of the file.

Usage

None.

XML

The following XML is a snippet of the FileBufferBase64 type.

```

<ae:GetFileAttachmentResponse>
  <ae:return xsi:type="ae:FileAttachmentContents">
    <ae:id>104</ae:id>
    <ae:name>my attachment</ae:name>
    <ae:fileName>fileName.txt</ae:fileName>
    <ae:showAsImage>false</ae:showAsImage>
    <ae:modificationDateTime>2010-09-17T18:46:25Z</ae:modificationDateTime>
    <ae:accessType>ATTACHACCESS-RESTRICTED</ae:accessType>
    <ae:contentsBase64>
      <ae:data>c2FtcGxlIGZpbGUgYXR0YWNobWVudA==</ae:data>
    </ae:contentsBase64>
  </ae:return>
</ae:GetFileAttachmentResponse>

```

GroupInfo

Description

The GroupInfo type holds the name and other information about a group in SBM. The GroupInfo type parameters are listed below.

Parameters

Name	Type	Description
id	GroupIdentifier [page 252]	Contains the complete identification information for a group.
accessType	AccessType [page 246]	Shows the group's product-access type.
memo	string	Contains the contents of the group's memo field.
isDeleted	boolean	Specify true to delete a group during UpdateGroups. The default is false. When returned in the response, it indicates whether or not the group is marked as deleted.

Usage

The GroupInfo type provides complete information about a group in SBM. Use [GetGroups](#) [page 225] to retrieve a list of attributes for a desired group.

XML

The following XML snippet shows the GroupInfo type in the group argument of the CreateGroups call.

```

<urn:group>
  <urn:id>
    <urn:displayName>New_Group1</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
  </urn:id>

```

```

    <urn:accessType>ACCESS-USER</urn:accessType>
    <urn:memo>This group contains regular users.</urn:memo>
    <urn:isDeleted></urn:isDeleted>
  </urn:group>

```

The following XML snippet shows the GroupInfo type in the return element of the GetGroups response.

```

<ae:return>
  <ae:group>
    <ae:id xsi:type="ae:GroupIdentifier">
      <ae:displayName>New_Group1</ae:displayName>
      <ae:id>18</ae:id>
      <ae:uuid>cb90f9de-5cc8-4056-bbc6-29caf39bd047</ae:uuid>
    </ae:id>
    <ae:accessType>ACCESS-USER</ae:accessType>
    <ae:memo>This group contains regular users.</ae:memo>
    <ae:isDeleted>>false</ae:isDeleted>
  </ae:group>
</ae:return>

```

GroupIdentifier

Description

The GroupIdentifier type holds the identification information for a user group. The GroupIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 253] .	Generic identification extension base. Holds the displayName, id, and uuid for a group.

Usage

The GroupIdentifier is the identifier that can be used in Web service methods to uniquely identify a group. The GroupIdentifier contains the generic information about a group (including the display name, ID, and UUID).



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows GroupIdentifier as seen in a typical call.

```

<urn:group>
  <urn:displayName>CR Submitters</urn:displayName>
  <urn:id>11</urn:id>
  <urn:uuid>cf83a358-d7fb-4b96-8f98-ed532c66cd0a</urn:uuid>
</urn:group>

```

Identifier

Description

The Identifier type holds generic identification information. The Identifier type parameters are listed below.

Parameters

Name	Type	Description
displayName	string	The display name of the object.
id	integer	The TS_ID of the object.
uuid	string	An alternate unique identifier for the object.

Usage

The Identifier is a structure that contains generic identification information about an object in SBM. The identifier is a common set of parameters that are combined with other elements to uniquely define objects in the database. For example, the identifier element is used in combination with other elements like tableId, tableItemId, and issueId to uniquely describe an item in several different ways. The Identifier is used in the following common types:

- [ApplicationIdentifier \[page 246\]](#)
- [ContactIdentifier \[page 248\]](#)
- [FieldIdentifier \[page 249\]](#)
- [GroupIdentifier \[page 252\]](#)
- [ItemIdentifier \[page 254\]](#)
- [ProjectIdentifier \[page 255\]](#)
- [ReportIdentifier \[page 256\]](#)
- [SolutionIdentifier \[page 256\]](#)
- [StateIdentifier \[page 257\]](#)
- [TableIdentifier \[page 258\]](#)
- [TransitionIdentifier \[page 259\]](#)
- [UserIdentifier \[page 260\]](#)
- [WorkflowIdentifier \[page 268\]](#)



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows Identifier in the <ae:project> element of a typical call.

```

<urn:project>
  <urn:displayName>Animation Pro</urn:displayName>
  <urn:id>2</urn:id>
  <urn:uuid>0b87f347-a00c-4359-9c16-625e847bfdab</urn:uuid>
  <urn:fullyQualifiedName>Base Project||Base IDT Project||
    Software Development||Animation Pro</urn:fullyQualifiedName>
</urn:project>

```

ItemIdentifier

Description

The ItemIdentifier type holds the identification information for an item. The ItemIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 253] .	Generic identification extension base. Holds the displayName, id, and uuid for an item.
tableID	integer	The TS_ID of the table that contains this item.
tableIdItemId	string	The TS_ID of the table, followed by the TS_ID of the item in that table (for example, 1000:164).
issueId	string	Item name for display purposes.

Usage

The ItemIdentifier is the identifier that can be used in Web service methods to uniquely identify an item. The ItemIdentifier contains the generic information about an item (including the display name, ID, and UUID) in addition to table ID and issue ID information for the item.



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows ItemIdentifier as seen in a typical call.

```

<urn:item>
  <urn:displayName>BUG000059</urn:displayName>
  <urn:id>25</urn:id>
  <urn:uuid>7d4703a0-302d-4da3-891e-1d36d43613f2</urn:uuid>

```

```
<urn:tableId>1000</urn:tableId>
<urn:tableIdItemId>1000:25</urn:tableIdItemId>
<urn:issueId>000059</urn:issueId>
</urn:item>
```

ProjectIdentifier

Description

The ProjectIdentifier type holds the identification information for a project. The ProjectIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 253] .	Generic identification extension base. Holds the displayName, id, and uuid for a project.
fullyQualifiedName	string	The fully qualified name of the project. Specifies the project name as it exists in the project hierarchy with respect to project inheritance. To specify the fully qualified name for a project, you start with the Base Project, followed by any other projects in the inheritance chain that lead to your ultimate project.

Usage

The ProjectIdentifier is the identifier that can be used in Web service methods to uniquely identify a project. The ProjectIdentifier contains the generic information about a project (including the display name, ID, and UUID) in addition to the fully qualified name.



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows ProjectIdentifier as seen in a typical call.

```
<urn:project>
  <urn:displayName>Animation Pro</urn:displayName>
  <urn:id>2</urn:id>
  <urn:uuid>0b87f347-a00c-4359-9c16-625e847bfdab</urn:uuid>
  <urn:fullyQualifiedName>Base Project||Base IDT Project||
    Software Development||Animation Pro</urn:fullyQualifiedName>
</urn:project>
```

ReportIdentifier

Description

The ReportIdentifier type holds the identification information for a report. The ReportIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 253] .	Generic identification extension base. Holds the displayName, id, and uuid for a report.

Usage

The ReportIdentifier is the identifier that can be used in Web service methods to uniquely identify a report. The ReportIdentifier contains the generic information about a report (including the display name, ID, and UUID).



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows ReportIdentifier as seen in a typical call.

```
<urn:report>
  <urn:displayName>Change Requests By Issue</urn:displayName>
  <urn:id>8</urn:id>
  <urn:uuid>40e8bb61-14fe-409c-aa5e-6399cf3e26a8</urn:uuid>
</urn:report>
```

SolutionIdentifier

Description

The SolutionIdentifier type holds the identification information for a solution. The SolutionIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 253] .	Generic identification extension base. Holds the displayName, id, and uuid for a solution.
uniqueName	string	The database name of the solution.

Name	Type	Description
tabName	string	The name that is displayed on the solution tab in the SBM User Workspace.

Usage

The SolutionIdentifier is the identifier that you use in Web service methods to uniquely identify a solution. The SolutionIdentifier contains the generic information about a solution (including the display name, ID, and UUID) in addition to the database name and SBM User Workspace tab name.



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows SolutionIdentifier as seen in a typical call.

```
<urn:solution>
  <urn:displayName>Issue Defect Management</urn:displayName>
  <urn:id>1</urn:id>
  <urn:uuid>cea0a86c-5d74-4e12-b8d6-9d6b90186f1e</urn:uuid>
  <urn:uniqueName>ISSUE_DEFECT_MANAGEMENT</urn:uniqueName>
  <urn:tabName>IDM</urn:tabName>
</urn:solution>
```

StateIdentifier

Description

The StateIdentifier type holds the identification information for a state. The StateIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 253] .	Generic identification extension base. Holds the displayName, id, and uuid for a state.
isClosed	boolean	Boolean value to indicate if the state is active or inactive.

Usage

The StateIdentifier is the identifier that can be used in Web service methods to uniquely identify a state. The StateIdentifier contains the generic information about a state (including the display name, ID, and UUID), in addition to whether or not the state is active or inactive.



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows StateIdentifier as seen in a typical call.

```
<urn:state>
  <urn:displayName>Evaluating Issue</urn:displayName>
  <urn:id>1</urn:id>
  <urn:uuid>985caf28-7a1c-4038-b6e2-c11703b214cd</urn:uuid>
  <urn:isClosed>>false</urn:isClosed>
</urn:state>
```

TableIdentifier

Description

The TableIdentifier type holds the identification information for a table. The TableIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 253] .	Generic identification extension base. Holds the displayName, id, and uuid for a table.
dbName	string	The unique database name of the table

Usage

The TableIdentifier is the identifier that can be used in Web service methods to uniquely identify a table. The TableIdentifier contains the generic information about a table (including the display name, ID, and UUID) in addition to the database name of the table.



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows TableIdentifier as seen in a typical call.

```
<urn:table>
  <urn:displayName>Issues</urn:displayName>
  <urn:id>1000</urn:id>
  <urn:uuid>dc8cd329-b430-436f-bb75-bf90008e6a50</urn:uuid>
```

```
<urn:dbName>UBG_ISSUES</urn:dbName>
</urn:table>
```

TimePreference

Description

TimePreference indicates a user's preferred time format. The available options are listed below.

Parameters

Name	Type	Description
TIME-FORMAT-12HOUR	string	Indicates a 12-hour clock preference.
TIME-FORMAT-24HOUR	string	Indicates a 24-hour clock preference.
TIME-FORMAT-USE-GMT-OFFSET	string	Indicates the use of a GMT offset.
TIME-FORMAT-HONOR-DAYLIGHT	string	Indicates time format with daylight savings honored.

Usage

TimePreference is used to determine how time is displayed to a user in the SBM User Workspace. The various time formats are returned in the timePreference parameter of the GetUsers response. See [UserInfo \[page 261\]](#) for more information.

XML

The following XML shows TimePreference as seen in the return element of a GetUsers response.

```
<ae:timePreference>TIME-FORMAT-12HOUR</ae:timePreference>
```

TransitionIdentifier

Description

The TransitionIdentifier type holds the identification information for a transition. The TransitionIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 253] .	Generic identification extension base. Holds the displayName, id, and uuid for a transition.

Usage

The TransitionIdentifier is the identifier that can be used in Web service methods to uniquely identify a transition. The TransitionIdentifier contains the generic information about a transition (including the display name, ID, and UUID).



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows TransitionIdentifier as seen in a typical call.

```
<urn:transition>
  <urn:displayName>Approve</urn:displayName>
  <urn:id>4</urn:id>
  <urn:uuid>a78f0a30-1305-46c2-b661-df8219c105b2</urn:uuid>
</urn:transition>
```

UserIdentifier

Description

The UserIdentifier type holds the identification information for a user. The UserIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 253] .	Generic identification extension base. Holds the displayName, id, and uuid for a user.
loginId	string	The unique login ID for the user.

Usage

The UserIdentifier is the identifier that can be used in Web service methods to uniquely identify a user. The UserIdentifier contains the generic information about a user (including the display name, ID, and UUID) in addition to the login ID for the user.



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows UserIdentifier as seen in a typical call.

```
<urn:user>
  <urn:displayName>Joe Manager</urn:displayName>
  <urn:id>2</urn:id>
  <urn:uuid>85a33f0b-9542-43fe-90c1-e152eeee777f</urn:uuid>
  <urn:loginId>joe</urn:loginId>
</urn:user>
```

UserInfo

Description

The UserInfo type holds the name and other information about a user in SBM. The UserInfo type parameters are listed below.

Parameters

Name	Type	Description
id	UserIdentifier [page 260]	Holds the complete identification information and login ID for a user account.
accessType	AccessType [page 246]	Shows the user's product-access type.
email	string	The user's primary email address.
emailCC	string	The user's CC email addresses.
timezone	string	The user's time zone.
offsetFromGMT	integer	This is the time zone's raw GMT offset.
dstSavings	integer	The amount of time in milliseconds to be added to local standard time to get local wall clock time.
datePreference	DatePreference [page 249]	The user's date preference.
timePreference	TimePreference [page 259]	The user's time preference.
namespaceName	string	The user's namespace name.
phoneNumber	string	The user's phone number.
locale	string	The user's designated locale.
isDeleted	boolean	Specify true to delete a user during UpdateUsers. The default is false. When returned in the response, it indicates whether or not the user is marked as deleted.
contact	ContactIdentifier [page 248]	Holds identification information for the user's associated contact record.
maxNotes	integer	Indicates the maximum number of notes to display on an item.

Name	Type	Description
maxChangeHistory	integer	Indicates the maximum number of change history records to display on an item.
maxItemsPerPage	integer	Indicates the maximum number of items to display per page in the SBM User Workspace.
group	GroupIdentifier [page 252]	Holds the identification information for groups to which the user belongs. One or more GroupIdentifiers are returned for each group.
fieldsMask	integer	Bit mask that indicates which field sections the user would like displayed as described in the TS_USERS table in the <i>Database Schema Reference</i> .
notesMask	integer	Bit mask that indicates the user's preference for displaying notes as described in the TS_USERS table in the <i>Database Schema Reference</i> .
changeHistoryMask	integer	Bit mask that indicates the user's preference for displaying change history information as described in the TS_USERS table in the <i>Database Schema Reference</i> .
browserMask	integer	Bit mask that indicates the user's browser preferences as described in the TS_USERS table in the <i>Database Schema Reference</i> .
preferredSolution	SolutionIdentifier [page 256]	Holds the identification information for the user's preferred application. This setting determines which application tab is selected by default when the user first enters the SBM User Workspace.
solutionData	UserSolutionData [page 266]	Contains the ordered-list of application tabs that appear for the user in the SBM User Workspace. The application identification information, home page report, and user's preferred projects are returned for each application.

Usage

The UserInfo type provides a user account in SBM. Use [GetUsers \[page 227\]](#) to retrieve a list of attributes for a specified user.

The timezone parameter is a programmatic ID; for example, "America/Los_Angeles". This ID is used to call up a specific real-world time zone. It corresponds to the IDs defined in

the standard Olson data used by UNIX systems, and has the format continent/city or ocean/city.

The `offsetFromGMT` parameter is the time zone's raw GMT offset (i.e., the number of milliseconds to add to GMT to get local time, before taking Daylight Saving Time into account). If DST is in effect for a given date, use the `dstSavings` value to adjust this offset.

The `dstSavings` parameter is the amount of time in milliseconds to be added to local standard time to get local wall clock time. If Daylight Saving Time is not observed in this user's timezone, this value will be 0. This value should be used only to adjust a date/time that is within the DST observation period.

The `namespaceName` parameter returns the name of the namespace that was generated while provisioning the customer environment. If the user does not belong to a namespace, then the default namespace name ("00000") is returned.

XML

The following XML snippet shows the `UserInfo` argument in the `CreateUsers` call.

```
<urn:user>
  <urn:id>
    <urn:displayName>John Doe</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:loginId>JDoe</urn:loginId>
  </urn:id>
  <urn:accessType>ACCESS-USER</urn:accessType>
  <urn:email>johndoe@companyName.com</urn:email>
  <urn:emailCC></urn:emailCC>
  <urn:timezone></urn:timezone>
  <urn:offsetFromGMT>-25200000</urn:offsetFromGMT>
  <urn:dstSavings>3600000</urn:dstSavings>
  <urn:datePreference>DATE-FORMAT-FROM-LOCALE</urn:datePreference>
  <urn:timePreference>TIME-FORMAT-12HOUR</urn:timePreference>
  <urn:namespaceName></urn:namespaceName>
  <urn:phoneNumber>111-111-1111</urn:phoneNumber>
  <urn:locale>en_US</urn:locale>
  <urn:isDeleted></urn:isDeleted>
  <urn:contact></urn:contact>
  <urn:maxNotes>10</urn:maxNotes>
  <urn:maxChangeHistory>10</urn:maxChangeHistory>
  <urn:maxItemsPerPage>20</urn:maxItemsPerPage>
  <urn:group>
    <urn:displayName>CR Submitters</urn:displayName>
  </urn:group>
  <urn:fieldsMask></urn:fieldsMask>
  <urn:notesMask></urn:notesMask>
  <urn:changeHistoryMask></urn:changeHistoryMask>
  <urn:browserMask></urn:browserMask>
  <urn:preferredSolution>
    <urn:displayName>Issue Defect Management</urn:displayName>
  </urn:preferredSolution>
  <urn:solutionData>
    <urn:solution>
      <urn:displayName>Issue Defect Management</urn:displayName>
```

```
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
        <urn:uniqueName></urn:uniqueName>
        <urn:tabName>IDM</urn:tabName>
    </urn:solution>
    <urn:homeReport>
        <urn:displayName>Built-In: All Items</urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
    </urn:homeReport>
    <urn:preferredProject>
        <urn:displayName>Animation Pro</urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
        <urn:fullyQualifiedName></urn:fullyQualifiedName>
    </urn:preferredProject>
</urn:solutionData>
</urn:user>
```

The following XML snippet shows the UserInfo type in the return element of the GetUsers response.

```
<ae:user>
  <ae:id xsi:type="ae:UserIdentifier">
    <ae:displayName>John Support Manager</ae:displayName>
    <ae:id>21</ae:id>
    <ae:uuid>08784a43-970f-4d28-9a6e-c301077ca653</ae:uuid>
    <ae:loginId>john</ae:loginId>
  </ae:id>
  <ae:accessType>ACCESS-USER</ae:accessType>
  <ae:email>john@companyName.com</ae:email>
  <ae:emailCC/>
  <ae:timezone/>
  <ae:offsetFromGMT>-25200000</ae:offsetFromGMT>
  <ae:dstSavings>3600000</ae:dstSavings>
  <ae:datePreference>DATE-FORMAT-FROM-LOCALE</ae:datePreference>
  <ae:timePreference>TIME-FORMAT-12HOUR</ae:timePreference>
  <ae:namespaceName>00000</ae:namespaceName>
  <ae:phoneNumber/>
  <ae:locale>en_US</ae:locale>
  <ae:isDeleted>>false</ae:isDeleted>
  <ae:maxNotes>10</ae:maxNotes>
  <ae:maxChangeHistory>10</ae:maxChangeHistory>
  <ae:maxItemsPerPage>20</ae:maxItemsPerPage>
  <ae:group xsi:type="ae:GroupIdentifier">
    <ae:displayName>Everyone</ae:displayName>
    <ae:id>1</ae:id>
    <ae:uuid>ade39c21-e7b2-4dcb-a231-d3d872671b59</ae:uuid>
  </ae:group>
  <ae:group xsi:type="ae:GroupIdentifier">
    <ae:displayName>IDM View Only</ae:displayName>
    <ae:id>5</ae:id>
    <ae:uuid>af38532d-e79c-495f-a3e6-f4bf784cc492</ae:uuid>
  </ae:group>
```

```

<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>IM Technician</ae:displayName>
  <ae:id>6</ae:id>
  <ae:uuid>902a1300-6ae1-44d7-b46a-e420babe8497</ae:uuid>
</ae:group>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>IM Manager</ae:displayName>
  <ae:id>7</ae:id>
  <ae:uuid>b70af5de-6642-4228-b9bb-d0da59bb6909</ae:uuid>
</ae:group>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>IM Administrator</ae:displayName>
  <ae:id>8</ae:id>
  <ae:uuid>b77b6e9d-e75a-4841-be1b-9c358affb797</ae:uuid>
</ae:group>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>CR Approval Board Members</ae:displayName>
  <ae:id>13</ae:id>
  <ae:uuid>ea167b8c-e9ea-4196-9727-dfd2f10fd751</ae:uuid>
</ae:group>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>CR Submitters</ae:displayName>
  <ae:id>11</ae:id>
  <ae:uuid>cf83a358-d7fb-4b96-8f98-ed532c66cd0a</ae:uuid>
</ae:group>
<ae:fieldsMask>1</ae:fieldsMask>
<ae:notesMask>4</ae:notesMask>
<ae:changeHistoryMask>4</ae:changeHistoryMask>
<ae:browserMask>13635632</ae:browserMask>
<ae:preferredSolution xsi:type="ae:SolutionIdentifier">
  <ae:displayName>Incident Management</ae:displayName>
  <ae:id>2</ae:id>
  <ae:uuid>c6f06a70-4d06-42a4-a3a9-50b2120dca41</ae:uuid>
  <ae:uniqueName>INCIDENT_MANAGEMENT</ae:uniqueName>
  <ae:tabName>Incident Mgmt</ae:tabName>
</ae:preferredSolution>
<ae:solutionData>
  <ae:solution xsi:type="ae:SolutionIdentifier">
    <ae:displayName>Incident Management</ae:displayName>
    <ae:id>2</ae:id>
    <ae:uuid>c6f06a70-4d06-42a4-a3a9-50b2120dca41</ae:uuid>
    <ae:uniqueName>INCIDENT_MANAGEMENT</ae:uniqueName>
    <ae:tabName>Incident Mgmt</ae:tabName>
  </ae:solution>
  <ae:homeReport xsi:type="ae:ReportIdentifier">
    <ae:displayName>Trend Of Incidents Closed On Initial Call</ae:displayName>
    <ae:id>38</ae:id>
    <ae:uuid>f8a5ce79-4b40-45cf-9f6d-735d060e90de</ae:uuid>
  </ae:homeReport>
</ae:solutionData>
<ae:solutionData>
  <ae:solution xsi:type="ae:SolutionIdentifier">
    <ae:displayName>Change Request Management</ae:displayName>
    <ae:id>3</ae:id>
    <ae:uuid>0ea28a74-6dde-406f-b19a-4c45aec40294</ae:uuid>

```

```

    <ae:uniqueName>CHANGE_REQUEST_MANAGEMENT</ae:uniqueName>
    <ae:tabName>Change Requests</ae:tabName>
  </ae:solution>
  <ae:homeReport xsi:type="ae:ReportIdentifier">
    <ae:displayName>Built-In: All Active Items I Own</ae:displayName>
    <ae:id>-6</ae:id>
    <ae:uuid>-6</ae:uuid>
  </ae:homeReport>
</ae:solutionData>
<ae:solutionData>
  <ae:solution xsi:type="ae:SolutionIdentifier">
    <ae:displayName>Issue Defect Management</ae:displayName>
    <ae:id>1</ae:id>
    <ae:uuid>cea0a86c-5d74-4e12-b8d6-9d6b90186f1e</ae:uuid>
    <ae:uniqueName>ISSUE_DEFECT_MANAGEMENT</ae:uniqueName>
    <ae:tabName>IDM</ae:tabName>
  </ae:solution>
  <ae:homeReport xsi:type="ae:ReportIdentifier">
    <ae:displayName>Built-In: All Active Items I Own</ae:displayName>
    <ae:id>-6</ae:id>
    <ae:uuid>-6</ae:uuid>
  </ae:homeReport>
</ae:solutionData>
</ae:user>

```

UserSolutionData

Description

The UserSolutionData type holds identification information and user settings for a solution (also known as an application) in SBM. The UserSolutionData type parameters are listed below.

Parameters

Name	Type	Description
solution	SolutionIdentifier [page 256]	Holds the identification information for a solution.
homeReport	ReportIdentifier [page 256]	Holds the identification information for a report. The report that is returned indicates the user's home page report.
preferredProject	ProjectIdentifier [page 255]	Holds the identification for a project. One or more projects are returned. The list of projects indicates the user's preferred projects.

Usage

The UserSolutionData type holds the SolutionIdentifier and additional information used to describe the settings for an application that a user has access to in the SBM User Workspace. You can use [GetUsers \[page 227\]](#) to retrieve the user solution data shown here. The GetUsers response contains one or more UserSolutionData elements that

comprise an ordered-list of the application tabs that are available to the user in the SBM User Workspace. You can use [UpdateUsers \[page 239\]](#) to re-establish this order, update the home page report, and designate the user's preferred projects.

XML

The following XML snippet shows UserSolutionData in the `<urn:solutionData>` element of the UpdateUsers call.

```
<urn:solutionData>
  <urn:solution>
    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:uniqueName>ISSUE_DEFECT_MANAGEMENT</urn:uniqueName>
    <urn:tabName>IDM</urn:tabName>
  </urn:solution>
  <urn:homeReport>
    <urn:displayName>Built-In: All Active Items I Own</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
  </urn:homeReport>
  <urn:preferredProject>
    <urn:displayName>Animation Pro</urn:displayName>
    <urn:id>6</urn:id>
    <urn:uuid></urn:uuid>
    <urn:fullyQualifiedName></urn:fullyQualifiedName>
  </urn:preferredProject>
</urn:solutionData>
<urn:solutionData>
  <urn:solution>
    <urn:displayName></urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
    <urn:uniqueName>INCIDENT_MANAGEMENT</urn:uniqueName>
    <urn:tabName>Incident Mgmt</urn:tabName>
  </urn:solution>
  <urn:homeReport>
    <urn:displayName>Built-In: All Active Items I Own</urn:displayName>
    <urn:id></urn:id>
    <urn:uuid></urn:uuid>
  </urn:homeReport>
  <urn:preferredProject>
    <urn:displayName>IM Project</urn:displayName>
    <urn:id>3</urn:id>
    <urn:uuid></urn:uuid>
    <urn:fullyQualifiedName></urn:fullyQualifiedName>
  </urn:preferredProject>
</urn:solutionData>
```

The following XML snippet shows the UserSolutionData type in the return element of the UpdateUsers response.

```
<ae:solutionData>
  <ae:solution xsi:type="ae:SolutionIdentifier">
```

```

<ae:displayName>Incident Management</ae:displayName>
<ae:id>2</ae:id>
<ae:uuid>c6f06a70-4d06-42a4-a3a9-50b2120dca41</ae:uuid>
<ae:uniqueName>INCIDENT_MANAGEMENT</ae:uniqueName>
<ae:tabName>Incident Mgmt</ae:tabName>
</ae:solution>
<ae:homeReport xsi:type="ae:ReportIdentifier">
  <ae:displayName>Built-In: All Items I Submitted</ae:displayName>
  <ae:id>-11</ae:id>
  <ae:uuid>-11</ae:uuid>
</ae:homeReport>
</ae:solutionData>
<ae:solutionData>
  <ae:solution xsi:type="ae:SolutionIdentifier">
    <ae:displayName>Change Request Management</ae:displayName>
    <ae:id>3</ae:id>
    <ae:uuid>0ea28a74-6dde-406f-b19a-4c45aec40294</ae:uuid>
    <ae:uniqueName>CHANGE_REQUEST_MANAGEMENT</ae:uniqueName>
    <ae:tabName>Change Requests</ae:tabName>
  </ae:solution>
  <ae:homeReport xsi:type="ae:ReportIdentifier">
    <ae:displayName>Built-In: All Items I Submitted</ae:displayName>
    <ae:id>-11</ae:id>
    <ae:uuid>-11</ae:uuid>
  </ae:homeReport>
</ae:solutionData>

```

WorkflowIdentifier

Description

The WorkflowIdentifier type holds the identification information for a workflow. The WorkflowIdentifier type parameters are listed below.

Parameters

Name	Type	Description
identifier	Extension. See Identifier [page 253] .	Generic identification extension base. Holds the displayName, id, and uuid for a workflow.

Usage

The WorkflowIdentifier is the identifier that can be used in Web service methods to uniquely identify a workflow. The WorkflowIdentifier contains the generic information about a workflow (including the display name, ID, and UUID).



Note: You can send any one of the elements in the identifier--you do not need to provide values for every element. You only need to provide more than one element in the event that the first element does not uniquely identify the object.

XML

The following XML shows WorkflowIdentifier as seen in a typical call.

```

<urn:workflow>
  <urn:displayName>IDM</urn:displayName>
  <urn:id>2</urn:id>
  <urn:uuid>5296c4b1-4bab-48a9-83b3-1f633894ef33</urn:uuid>
</urn:workflow>

```

Arguments

This section provides detailed descriptions of SBM Administrative Web service arguments. The arguments are the request parameters that can be used by SBM Administrative Web service operations. The arguments listed here contain one or more parameters, which make up the data being sent to the Web service. The parameters listed in each argument are either simple or complex types themselves. If the type is complex, a link to further detail of that type will be provided in the **Type** column.

The following table lists all supported arguments in alphabetical order, followed by a brief description of each type. Select an argument to view detailed information including:

- **Description**

A brief description of the argument.

- **Parameters**

A table describing the types for each argument. Both simple and complex types are listed for each argument. For each complex type, you can click the type name for a detailed description.

- **Usage**

Any notes, additional details, and concerns regarding the argument are addressed here.

- **XML**

An example of the actual XML being sent is displayed here. The XML not only shows the argument and its respective elements, you can also see detailed examples of each element and how to format the expected data.

Arguments

Argument	Description
Auth [page 270]	Supplies credentials and optionally, a host name for licensing.
ExtendedData [page 271]	Placeholder argument for future argument elements.
ExtraValue [page 272]	Holds a name value pairing for future elements.
GroupListAction [page 272]	Specifies whether the service should add, remove, or replace groups.

Argument	Description
Options [page 274]	Holds the ExtraValue type, which contains name value pairing for future arguments on certain calls.
MultipleOption [page 274]	Specifies whether a service should stop on failure and send an error message or continue processing.
MultipleOptions [page 275]	Holds the Options type and an enumeration to determine if a service should stop on failure and send an error message or continue processing.
MultipleResponseItemOptions [page 276]	Holds the Options type, as well as an enumeration to determine if a service should stop on failure and send an error message or continue processing. Also enables you to limit the data that is returned in a response.
SectionsOption [page 278]	Determines the section of an item to return.
UserResponseOptions [page 280]	Holds the Options type, as well as an enumeration to determine if a service should stop on failure and send an error message or continue processing. Also enables you to limit the data that is returned in a response.

Auth

Description

The Auth type supplies credentials and optionally, a host name for licensing. The Auth type parameters are listed below.

Parameters

Name	Type	Description
userID	string	The SBM user Login ID. If you are only specifying the host name, then userID is optional
password	string	The password for the user. If you are only specifying the host name, then password is optional.
hostname	string	The host name of the client.
loginAsUserId	string	User ID for the SBM login you wish to impersonate. If you are only specifying the host name, then loginAsUserId is optional.
extendedData	ExtendedData [page 271]	Placeholder for future arguments.

Usage

The Auth type allows credentials to be provided if not using WS-SECURITY or HTTP BASIC to pass the credentials. The hostname element is only needed in case you want to override the client's IP address for licensing purposes, forcing Serena License Manager to use a particular client host. If it's not provided, the code gets the client hostname from the socket.

XML

The following XML shows Auth as seen in a typical call.

```
<urn:auth>
  <urn:userId>admin</urn:userId>
  <urn:password>password</urn:password>
  <urn:hostname>localhost</urn:hostname>
  <urn:loginAsUserId></urn:loginAsUserId>
  <urn:extendedData></urn:extendedData>
</urn:auth>
```

ExtendedData

Description

The ExtendedData type is a placeholder for future argument or response elements. The ExtendedData type parameters are listed below.

Parameters

Name	Type	Description
data	ExtraValue [page 272]	Holds the name value pair for an additional argument or response element.

Usage

The ExtendedData type is simply a placeholder for future arguments that might be added to a given call. Responses have a similar placeholder for future response elements.

XML

The following XML shows ExtendedData as seen within the auth argument of a typical call.

```
<urn:auth>
  <urn:userId>admin</urn:userId>
  <urn:password>password</urn:password>
  <urn:hostname>localhost</urn:hostname>
  <urn:loginAsUserId></urn:loginAsUserId>
  <urn:extendedData>
    <urn:data>
      <urn:name></urn:name>
      <urn:value></urn:value>
    </urn:data>
  </urn:extendedData>
</urn:auth>
```

ExtraValue

Description

ExtraValue holds a name value pairing for future elements.

Parameters

Name	Type	Description
name	extraValue	Holds the name of a future element.
value	string	Holds the value of a future element.

Usage

None.

XML

The following XML shows ExtraValue in the `<urn:extraOption>` element in a typical call.

```
<urn:options>
  <urn:extraOption>
    <urn:name></urn:name>
    <urn:value></urn:value>
  </urn:extraOption>
</urn:options>
```

GroupListAction

Description

The GroupListAction enumeration enables you to specify whether groups should be added, replaced, or removed during the UpdateUsers call.

Parameters

Name	Type	Description
ADD-GROUPS	string	Adds the specified groups on update.
REPLACE-GROUPS	string	Replaces the existings groups on update.
REMOVE-GROUPS	string	Removes the specified groups on update.

Usage

The GroupListAction element enables you to control how user groups are processed during the UpdateUsers call. Using the groupAction argument, you set one to these options to add, replace, or remove existing groups. See [UpdateUsers \[page 239\]](#) for additional information.

XML

The following XML shows `GroupListAction` in the `<urn:groupAction>` element of the `UpdateUsers` call. In this example, Joe is added to the CR Submitters group:

```
<urn:UpdateUsers>
  <urn:auth>
    <urn:userId></urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:allowEmptyValues>false</urn:allowEmptyValues>
  <urn:groupAction>ADD-GROUPS</urn:groupAction>
  <urn:user>
    <urn:id>
      <urn:displayName>Joe</urn:displayName>
      <urn:id></urn:id>
      <urn:uuid></urn:uuid>
      <urn:loginId>Joe</urn:loginId>
    </urn:id>
    <urn:contact>
      <urn:displayName></urn:displayName>
      <urn:id></urn:id>
      <urn:uuid></urn:uuid>
    </urn:contact>
    <urn:group>
      <urn:displayName>CR Submitters</urn:displayName>
      <urn:id></urn:id>
      <urn:uuid></urn:uuid>
    </urn:group>
  </urn:user>
</urn:UpdateUsers>
```

In this example, the current group membership for Nancy and Carmen is replaced. Nancy and Carmen now only belong to CR Submitters:

```
<urn:UpdateUsers>
  <urn:auth>
    <urn:userId>Admin</urn:userId>
    <urn:password></urn:password>
    <urn:hostname></urn:hostname>
    <urn:loginAsUserId></urn:loginAsUserId>
  </urn:auth>
  <urn:allowEmptyValues>false</urn:allowEmptyValues>
  <urn:groupAction>REPLACE-GROUPS</urn:groupAction>
  <urn:user>
    <urn:id>
      <urn:loginId>Nancy</urn:loginId>
    </urn:id>
    <urn:contact>
      <urn:displayName></urn:displayName>
    </urn:contact>
    <urn:group>
      <urn:displayName>CR Submitters</urn:displayName>
    </urn:group>
  </urn:user>
```

```

    <urn:user>
      <urn:id>
        <urn:loginId>Carmen</urn:loginId>
      </urn:id>
      <urn:contact>
        <urn:displayName></urn:displayName>
        <urn:id></urn:id>
        <urn:uuid></urn:uuid>
      </urn:contact>
      <urn:group>
        <urn:displayName>CR Submitters</urn:displayName>
      </urn:group>
    </urn:user>
  </urn:UpdateUsers>

```

Options

Description

Options holds the ExtraValue type, which contains a name value pairing for future arguments on certain calls.

Parameters

Name	Type	Description
extraOption	ExtraValue [page 272]	Holds a name value pairing for future elements.

Usage

None.

XML

The following XML shows Options as seen in a typical call.

```

<urn:options>
  <urn:extraOption>
    <urn:name></urn:name>
    <urn:value></urn:value>
  </urn:extraOption>
</urn:options>

```

MultipleOption

Description

The MultipleOption enumeration enables you to specify whether a Web service should stop on failure and send an error message or continue processing.

Parameters

Name	Type	Description
CONTINUE-ON-FAILURE	string	If a failure is encountered, continue processing the rest of the items in the call.
STOP-ON-FAILURE	string	If a failure is encountered, stop processing items and return an error.

Usage

The `MultipleOption` element enables you to specify whether the service should continue if an error is encountered, or stop and throw an error. If any failures occur, each successive error message is appended to the string that is returned. Multiple error messages are separated by a single newline. Failures do not result in a return before all items have been processed. See [GroupHolder \[page 283\]](#) for an example of a `GetGroups` call that returns an error, but continues to process the rest of the records.

XML

The following XML shows `MultipleOption` in the `<urn:multiOption>` element of a typical call.

```
<urn:options>
  <urn:extraOption>
    <urn:name></urn:name>
    <urn:value></urn:value>
  </urn:extraOption>
  <urn:multiOption>CONTINUE-ON-FAILURE</urn:multiOption>
  <urn:sections>SECTIONS-ALL</urn:sections>
  <urn:specifiedSections></urn:specifiedSections>
  </urn:limitedField>
</urn:options>
```

MultipleOptions

Description

`MultipleOptions` holds the `Options` type and enables you to specify whether a Web service should continue if an error is encountered, or stop and throw an error.

Parameters

Name	Type	Description
options	Extension. See Options [page 274] .	Options extension base. Holds name value pairing for future arguments on certain calls.
multiOption	MultipleOption [page 274]	Enumeration element that holds the option to continue or stop processing of items.

Usage

MultipleOptions holds the Options type and the MultipleOption type. You use the multiOption parameter to control service handling as follows:

- **multiOption** – Use the multiOption element to specify whether the service should continue if an error is encountered, or stop and return an error. If any failures occur, each successive error message is appended to the string that is returned. Multiple error messages are separated by a single newline. If you specify CONTINUE-ON-FAILURE, then failures do not result in a return before all items have been processed. For more information, see [MultipleOption \[page 274\]](#).

XML

The following XML shows MultipleOptions in the <urn:options> element in a typical call.

```
<urn:options>
  <urn:extraOption>
    <urn:name></urn:name>
    <urn:value></urn:value>
  </urn:extraOption>
  <urn:multiOption>CONTINUE-ON-FAILURE</urn:multiOption>
</urn:options>
```

MultipleResponseItemOptions

Description

MultipleResponseItemOptions holds the Options type, as well as an enumeration to determine if a service should stop on failure and send an error message or continue processing. It also enables you to limit the data that is returned in a response.

Parameters

Name	Type	Description
options	Extension. See Options [page 274] .	Options extension base. Holds a name value pairing for future elements.
multiOption	MultipleOption [page 274]	Enumeration element that holds the option to continue processing or stop when an error is encountered.
sections	SectionsOption [page 278]	Enumeration element that controls the sections of an item that should be returned.
specifiedSections	string	If SECTIONS-SPECIFIED is used in the sections element above, enter the specified section or sections here. The available options are described below in the Usage section.
limitedField	FieldIdentifier [page 249]	Enables you to limit the return results based on one or more fields that you identify.

Usage

The `MultipleResponseItemOptions` contains the same functionality as `MultipleOptions`, but it also enables you to limit the return results based on item sections or specific fields that you identify.

You use the following parameters to control service handling and the amount of data that is returned in the response:

- **multiOption** – Use the `multiOption` element to specify whether the service should continue if an error is encountered, or stop and return an error. If any failures occur, each successive error message is appended to the string that is returned. Multiple error messages are separated by a single newline. If you specify `CONTINUE-ON-FAILURE`, then failures do not result in a return before all items have been processed. For more information, see [MultipleOption \[page 274\]](#).
- **sections** and **specifiedSections** – Use these elements to specify which parts of an item should be returned in order to limit the data that is returned for a given item. The sections that are not specified are not included in the response. For example, if the items have a large number of item links, notes, URL and file attachments that don't need to be returned in the response, use these parameters to return only the sections of an item you want. For more information, see [SectionsOption \[page 278\]](#). In the `sections` parameter, specify one of the following enumeration options:
 - **SECTIONS-ALL** – Returns all sections. This is the default value for the `sections` argument. If no value is specified, `ALL` is the assumed value.
 - **SECTIONS-NONE** – No sections are returned (only the `ItemIdentifier` is returned). Note that the `ItemIdentifier` is always returned, regardless of the value or values specified in the `sections` parameter.
 - **SECTIONS-SPECIFIED** – Returns sections that you specify.

You can use a comma-separated list in `specifiedSections` to return only the sections of an item you want. For example:

```
<urn:specifiedSections>SECTION:FIXED,SECTION:EXTENDED</urn:specifiedSections>
```

This ensures that only the fixed and extended sections of an item are returned. Here are some of the possible sections you can specify:

- **SECTION:FIXED** – All parameters in `TTItem` (from `<urn:itemType>` to `<urn:url>`) prior to the `extendedField` parameter are returned.
- **SECTION:EXTENDED** – Returns all of the `extendedFields` in `TTItem`.
- **SECTION:ATTACHMENTS** – Returns all of the attachment sections of `TTItem`. You can return the next four sections simply by specifying:

```
<urn:specifiedSections>SECTION:ATTACHMENTS</urn:specifiedSections>
```

- **SECTION:NOTES** – Returns all note sections of `TTItem`.
- **SECTION:ITELINKS** – Returns all `itemLink` sections of `TTItem`.
- **SECTION:URLATTACHMENTS** – Returns all `urlAttachment` sections of `TTItem`.

- **SECTION:FILEATTACHMENTS** – Returns all fileAttachment sections of TTIItem.



Note: If you specify SECTION:NONE after other sections, those preceding sections will not be returned. For example, SECTION:FIXED,SECTION:NONE,SECTION:EXTENDED will only return the extendedField sections.

- **limitedField** – Use the limitedField element to use specific fields to limit the item data that is returned. For example, you can specify one or more fields to limit a service response to return only the fields that you want to return. In the event that limitedField contradicts the sections value, the sections specification takes precedence. The following XML shows how to limit return results using the limitedField option.

```
<urn:limitedField>
  <urn:displayName>Severity</urn:displayName>
  <urn:id>69</urn:id>
  <urn:uuid>f4eff572-2e29-4d7f-8478-8e9e16865c2c</urn:uuid>
  <urn:dbName>SEVERITY</urn:dbName>
</urn:limitedField>
```

XML

The following XML shows MultipleResponseItemOptions in the <urn:options> element of a typical call.

```
<urn:options>
  <urn:extraOption>
    <urn:name></urn:name>
    <urn:value></urn:value>
  </urn:extraOption>
  <urn:multiOption>CONTINUE-ON-FAILURE</urn:multiOption>
  <urn:sections>SECTIONS-SPECIFIED</urn:sections>
  <urn:specifiedSections>SECTION:EXTENDED</urn:specifiedSections>
  </urn:limitedField>
</urn:options>
```

SectionsOption

Description

SectionsOption indicates the sections that should be returned for a record.

Parameters

Name	Type	Description
SECTIONS-ALL	string	Returns all sections. This is the default value for the sections argument. If no value is specified, ALL is the assumed value.
SECTIONS-NONE	string	No sections are returned (only the Identifier is returned). Note that the Identifier element is always returned, regardless of the value or values specified in the sections parameter.

Name	Type	Description
SECTIONS-SPECIFIED	string	Returns sections that you specify.

Usage

Use the `sections` and `specifiedSections` arguments to specify which parts of a record should be returned in order to limit the total amount of data that is returned. The sections that are not specified are not included in the response. For example, if the items have a large number of item links, notes, URL and file attachments that don't need to be returned in the response, use these parameters to return only the sections of an item you want. If you only need basic user information in the `GetUsers` response, use the `specifiedSections` parameter to return only the `STANDARD` section. If you need basic user information and group information, specify `STANDARD` and `GROUPS`.

In the `sections` parameter, specify one of the following enumeration options:

- **SECTIONS-ALL** – Returns all sections. This is the default value for the `sections` argument. If no value is specified, `ALL` is the assumed value. For example:

```
<urn:sections>SECTIONS-ALL</urn:sections>
<urn:specifiedSections></urn:specifiedSections>
```

- **SECTIONS-NONE** – No sections are returned (only the Identifier element is returned). For example:

```
<urn:sections>SECTIONS-NONE</urn:sections>
<urn:specifiedSections></urn:specifiedSections>
```



Note: The Identifier is always returned, regardless of the value or values specified in the `sections` parameter.

- **SECTIONS-SPECIFIED** – Returns sections that you specify. For example:

```
<urn:sections>SECTIONS-SPECIFIED</urn:sections>
<urn:specifiedSections>SECTION:EXTENDED</urn:specifiedSections>
```

You can also use a comma-separated list in `specifiedSections` to return only the sections of a record you want. For example:

```
<urn:specifiedSections>SECTION:FIXED,SECTION:EXTENDED</urn:specifiedSections>
```

This ensures that only the fixed and extended sections of an item are returned. Below are the sections that you can specify for items:

- **SECTION:FIXED** – All parameters in `TTItem` (from `<urn:itemType>` to `<urn:url>`) prior to the `extendedField` parameter are returned.
- **SECTION:EXTENDED** – Returns all of the `extendedFields` in `TTItem`.
- **SECTION:ATTACHMENTS** – Returns all of the attachment sections of `TTItem`. You can return the next four sections simply by specifying:

```
<urn:specifiedSections>SECTION:ATTACHMENTS</urn:specifiedSections>
```

- **SECTION:NOTES** – Returns all note sections of TTItem.
- **SECTION:ITELINKS** – Returns all itemLink sections of TTItem.
- **SECTION:URLATTACHMENTS** – Returns all urlAttachment sections of TTItem.
- **SECTION:FILEATTACHMENTS** – Returns all fileAttachment sections of TTItem.

Below are the possible sections that you can specify for user records:

- **SECTION:STANDARD** – Returns parameters from UserInfo such as the accessType, e-mail address, contact, date and time information, phone number, locale, and the various display preferences.
- **SECTION:GROUPS** – Returns one or more GroupIdentifiers for each group to which the user belongs.
- **SECTION:SOLUTIONS** – Returns the preferredSolution and solutionData elements.



Note: If you specify SECTION:NONE after other sections, those preceding sections will not be returned. For example, SECTION:FIXED,SECTION:NONE,SECTION:EXTENDED will only return the extendedField sections. For the GetUsers call, SECTION:STANDARD,SECTION:NONE,SECTION:GROUPS will only return the groups section.

XML

The following XML shows SectionsOption in the `<urn:sections>` element of a typical call.

```
<urn:options>
  <urn:extraOption>
    <urn:name></urn:name>
    <urn:value></urn:value>
  </urn:extraOption>
  <urn:multiOption>CONTINUE-ON-FAILURE</urn:multiOption>
  <urn:sections>SECTIONS-SPECIFIED</urn:sections>
  <urn:specifiedSections>SECTION:STANDARD</urn:specifiedSections>
  </urn:limitedField>
</urn:options>
```

UserResponseOptions

Description

UserResponseOptions holds the Options type, as well as an enumeration to determine if a service should stop on failure and send an error message or continue processing. It also enables you to limit the data that is returned in a response.

Parameters

Name	Type	Description
options	Extension. See Options [page 274] .	Options extension base. Holds a name value pairing for future elements.
multiOption	MultipleOption [page 274]	Enumeration element that holds the option to continue processing or stop when an error is encountered.
sections	SectionsOption [page 278]	Enumeration element that controls the sections of a user record that should be returned.
specifiedSections	string	If SECTIONS-SPECIFIED is used in the sections element above, enter the specified section or sections here. The available options are described below in the Usage section.

Usage

The UserResponseOptions contains the same functionality as MultipleOptions, but it also enables you to limit the return results based on sections of a user record.

You use the following parameters to control service handling and the amount of data that is returned in the response:

- **multiOption** – Use the multiOption element to specify whether the service should continue if an error is encountered, or stop and return an error. If any failures occur, each successive error message is appended to the string that is returned. Multiple error messages are separated by a single newline. If you specify CONTINUE-ON-FAILURE, then failures do not result in a return before all records have been processed. For more information, see [MultipleOption \[page 274\]](#).
- **sections** and **specifiedSections** – Use these elements to specify which parts of a user record should be returned in order to limit the amount of data that is returned. The sections that are not specified are not included in the response. For example, if you only need basic user information in the response, use the specifiedSections parameter to return only the STANDARD section. If you need basic user information and group information, specify STANDARD and GROUPS. For more information, see [SectionsOption \[page 278\]](#).

In the sections parameter, specify one of the following enumeration options:

- **SECTIONS-ALL** – Returns all sections. This is the default value for the sections argument. If no value is specified, ALL is the assumed value.
- **SECTIONS-NONE** – No sections are returned (only the UserIdentifier is returned). Note that the UserIdentifier is always returned, regardless of the value or values specified in the sections parameter.
- **SECTIONS-SPECIFIED** – Returns sections that you specify.

You can use a comma-separated list in specifiedSections to return only the sections of a user record that you want. For example:

```
<urn:specifiedSections>SECTION:STANDARD,SECTION:GROUPS</urn:specifiedSections>
```

This ensures that only the standard and groups sections of a user record are returned. Here are some of the possible sections you can specify:

- **SECTION:STANDARD** – Returns parameters from UserInfo such as the accessType, e-mail address, contact, date and time information, phone number, locale, and the various display preferences.
- **SECTION:GROUPS** – Returns one or more GroupIdentifiers for each group to which the user belongs.
- **SECTION:SOLUTIONS** – Returns the preferredSolution and solutionData elements.



Note: If you specify SECTION:NONE after other sections, those preceding sections will not be returned. For example, SECTION:STANDARD,SECTION:NONE,SECTION:GROUPS will only return the groups section.

XML

The following XML shows UserResponseOptions in the `<urn:options>` element of the GetUsers call.

```
<urn:options>
  <urn:extraOption>
    <urn:name></urn:name>
    <urn:value></urn:value>
  </urn:extraOption>
  <urn:multiOption>CONTINUE-ON-FAILURE</urn:multiOption>
  <urn:sections>SECTIONS-SPECIFIED</urn:sections>
  <urn:specifiedSections>SECTION:STANDARD</urn:specifiedSections>
</urn:options>
```

Responses

This section provides detailed descriptions of SBM Administrative Web service responses. The responses are the data elements that are returned from SBM Administrative Web service operations. The responses listed here contain one or more parameters, which make up the data being sent back from the Web service. The parameters listed in each argument are either simple or complex types themselves. If the type is complex, a link to further detail of that type will be provided in the **Type** column.

The following table lists all supported responses in alphabetical order, followed by a brief description of each type. Select a response to view detailed information including:

- **Description**

A brief description of the response.

- **Parameters**

A table describing the types for each response. Both simple and complex types are listed for each response. For each complex type, you can click the type name for a detailed description.

- **Usage**

Any notes, additional details, and concerns regarding the response are addressed here.

- **XML**

An example of the actual XML being sent is displayed here. The XML not only shows the response and its respective elements, you can also see detailed examples of each element and how the expected data is formatted.

Responses

Response	Description
GroupHolder [page 283]	Holds all of the information for a group and any applicable error messages.
GroupHolder [page 283]	Holds the name and additional information about a group.
NewUser [page 285]	Holds user information, a temporary password, and any applicable error messages that occurred while creating users.
Privilege [page 287]	Holds the name and other information about a privilege.
Privilege [page 287]	Holds the name and other information about a project.
Status [page 290]	Holds status information for a Web service operation.
StatusEnum [page 291]	Indicates the type of status.
UserHolder [page 292]	Holds user information and any applicable error messages.

GroupHolder

Description

The GroupHolder type contains the GroupInfo response and any errors for group records that were not successfully returned. The GroupHolder type parameters are listed below.

Parameters

Name	Type	Description
group	GroupInfo [page 251]	The GroupInfo type holds the complete informaton that describes a group.
status	Status [page 290]	Holds status information for any messages or failures that are encountered.

Usage

The GroupHolder type holds detailed information for a group including the access type and status (whether it is marked as deleted or not). If any errors occur during the call, they are returned in the status element in the order in which they were processed.

XML

The following XML snippet shows GroupHolder in the return element of the GetGroups response. Note that an error message is returned in this example (inside the `message` element). In this example, three groups were requested using the GetGroups call; however, the second group (CR Submitters) was not properly identified. Therefore, an error was returned for the second group, though the service continued to run and returned the third group (IDM Manager).

```
<ae:GetGroupsResponse>
  <ae:return>
    <ae:group>
      <ae:id xsi:type="ae:GroupIdentifier">
        <ae:displayName>IDM Team</ae:displayName>
        <ae:id>2</ae:id>
        <ae:uuid>634aed4a-bc98-49d3-86f0-6095c2f7b9b6</ae:uuid>
      </ae:id>
      <ae:accessType>ACCESS-USER</ae:accessType>
      <ae:memo/>
      <ae:isDeleted>>false</ae:isDeleted>
    </ae:group>
  </ae:return>
  <ae:return>
    <ae:status>
      <ae:status>IS-ERROR</ae:status>
      <ae:message>The specified group id or name 'CR Submitter'
        → does not exist.</ae:message>
    </ae:status>
  </ae:return>
  <ae:return>
    <ae:group>
      <ae:id xsi:type="ae:GroupIdentifier">
        <ae:displayName>IDM Manager</ae:displayName>
        <ae:id>3</ae:id>
        <ae:uuid>b2654094-0ec6-460d-856b-9b72709d183b</ae:uuid>
      </ae:id>
    </ae:group>
  </ae:return>
</ae:GetGroupsResponse>
```

```

    </ae:id>
    <ae:accessType>ACCESS-USER</ae:accessType>
    <ae:memo/>
    <ae:isDeleted>>false</ae:isDeleted>
  </ae:group>
</ae:return>
</ae:GetGroupsResponse>

```

NewUser

Description

The NewUser type contains one or more new or existing users and errors for user records that were not successfully returned. The NewUser type parameters are listed below.

Parameters

Name	Type	Description
userHolder	UserHolder [page 292]	The userInfo type holds the complete informaton that describes a user.
isNew	boolean	Specifies whether the user is new (true) or if it is an existing account that was updated (false).
temporaryPassword	string	Indicates the temporary password that has been established for the user.

Usage

The NewUser type holds detailed information for a new or existing user and his or her preferences, as well as the user's temporary password . If any errors occur during the call, they are returned in the status element in the order in which they were processed.

XML

The following XML snippet shows NewUser in the return element of the CreateUsers response. Note that an error message is returned in this example (inside the `message` element). In this example, three users were created; however, the second user (Johnny_Doe) was not properly identified (no login ID was specified). Therefore, an error was returned for the second user, though the service continued to run and created the third user (Jane_Doe).

```

<ae:CreateUsersResponse>
  <ae:return>
    <ae:userHolder>
      <ae:user>
        <ae:id xsi:type="ae:UserIdentifier">
          <ae:displayName>John_Doe</ae:displayName>
          <ae:id>47</ae:id>
          <ae:uuid>893de970-fafa-4907-84d3-50e7b82fcaaa</ae:uuid>
          <ae:loginId>J_Doe</ae:loginId>
        </ae:id>
        <ae:accessType>ACCESS-USER</ae:accessType>

```

```
<ae:email>johndoe@companyName.com</ae:email>
<ae:emailCC/>
<ae:timezone/>
<ae:offsetFromGMT>-25200000</ae:offsetFromGMT>
<ae:dstSavings>3600000</ae:dstSavings>
<ae:datePreference>DATE-FORMAT-FROM-LOCALE</ae:datePreference>
<ae:timePreference>TIME-FORMAT-12HOUR</ae:timePreference>
<ae:namespaceName>00000</ae:namespaceName>
<ae:phoneNumber/>
<ae:locale>en_US</ae:locale>
<ae:isDeleted>>false</ae:isDeleted>
<ae:maxNotes>10</ae:maxNotes>
<ae:maxChangeHistory>10</ae:maxChangeHistory>
<ae:maxItemsPerPage>20</ae:maxItemsPerPage>
<ae:fieldsMask>1</ae:fieldsMask>
<ae:notesMask>4</ae:notesMask>
<ae:changeHistoryMask>4</ae:changeHistoryMask>
<ae:browserMask>1052724</ae:browserMask>
<ae:preferredSolution xsi:type="ae:SolutionIdentifier">
  <ae:id>-1</ae:id>
</ae:preferredSolution>
</ae:user>
</ae:userHolder>
<ae:isNew>>true</ae:isNew>
<ae:temporaryPassword>WSNP</ae:temporaryPassword>
</ae:return>
<ae:return>
  <ae:userHolder>
    <ae:status>
      <ae:status>IS-ERROR</ae:status>
      <ae:message>Create user requires a user login id.</ae:message>
    </ae:status>
  </ae:userHolder>
  <ae:isNew>>false</ae:isNew>
</ae:return>
<ae:return>
  <ae:userHolder>
    <ae:user>
      <ae:id xsi:type="ae:UserIdentifier">
        <ae:displayName>Jane_Doe</ae:displayName>
        <ae:id>48</ae:id>
        <ae:uuid>5ddb7a7-bae0-467d-9e71-9390f0932541</ae:uuid>
        <ae:loginId>Ja_Doe</ae:loginId>
      </ae:id>
      <ae:accessType>ACCESS-USER</ae:accessType>
      <ae:email>jadoe@companyName.com</ae:email>
      <ae:emailCC/>
      <ae:timezone/>
      <ae:offsetFromGMT>-25200000</ae:offsetFromGMT>
      <ae:dstSavings>3600000</ae:dstSavings>
      <ae:datePreference>DATE-FORMAT-FROM-LOCALE</ae:datePreference>
      <ae:timePreference>TIME-FORMAT-12HOUR</ae:timePreference>
      <ae:namespaceName>00000</ae:namespaceName>
      <ae:phoneNumber/>
      <ae:locale>en_US</ae:locale>
```

```

    <ae:isDeleted>>false</ae:isDeleted>
    <ae:maxNotes>10</ae:maxNotes>
    <ae:maxChangeHistory>10</ae:maxChangeHistory>
    <ae:maxItemsPerPage>20</ae:maxItemsPerPage>
    <ae:fieldsMask>1</ae:fieldsMask>
    <ae:notesMask>4</ae:notesMask>
    <ae:changeHistoryMask>4</ae:changeHistoryMask>
    <ae:browserMask>1052724</ae:browserMask>
    <ae:preferredSolution xsi:type="ae:SolutionIdentifier">
      <ae:id>-1</ae:id>
    </ae:preferredSolution>
  </ae:user>
</ae:userHolder>
<ae:isNew>>true</ae:isNew>
<ae:temporaryPassword>VWWR</ae:temporaryPassword>
</ae:return>
</ae:CreateUsersResponse>

```

Privilege

Description

The Privilege type holds the name and other information about a privilege in SBM. The Privilege type parameters are listed below.

Parameters

Name	Type	Description
name	string	The name of the privilege.
objectUUID	string	The UUID of the object to which the privilege applies. May be set to null if not scoped to an object. May return the ID and not UUID in some cases.
type	string	The type of privilege. See below.

Usage

The Privilege type completely describes an available privilege in SBM. Use [GetUserPrivileges \[page 229\]](#) to retrieve a list of privileges available for a specified user.

The available privilege types for a normal user are:

TS_PRIVTYPE_USERSYS
TS_PRIVTYPE_USERPRJ
TS_PRIVTYPE_USERWKF
TS_PRIVTYPE_USERFLD

TS_PRIVTYPE_USERTBL

The available privilege types for an administrator are:

TS_PRIVTYPE_ADMSYS
TS_PRIVTYPE_ADMPRJ
TS_PRIVTYPE_ADMWKF
TS_PRIVTYPE_ADMFLD_PRJ
TS_PRIVTYPE_ADMFLD_WKF
TS_PRIVTYPE_ADMCON
TS_PRIVTYPE_UNKNOWN
TS_PRIVTYPE_SYSMASK
TS_PRIVTYPE_ADMMASK
TS_PRIVTYPE_TBLMASK

XML

The following XML snippet shows the Privilege type in the return element of the GetUserPrivileges response.

```
<ae:GetUserPrivilegesResponse>
  <ae:return>
    <ae:name>TS_ADMPRJPRIV_ADDPROJECT</ae:name>
    <ae:objectUUID>ROOTPROJECT</ae:objectUUID>
    <ae:type>TS_PRIVTYPE_ADMPRJ</ae:type>
  </ae:return>
  <ae:return>
    <ae:name>TS_ADMPRJPRIV_EDITPROJECT</ae:name>
    <ae:objectUUID>ROOTPROJECT</ae:objectUUID>
    <ae:type>TS_PRIVTYPE_ADMPRJ</ae:type>
  </ae:return>
  <ae:return>
    <ae:name>TS_ADMPRJPRIV_DELETEPROJECT</ae:name>
    <ae:objectUUID>ROOTPROJECT</ae:objectUUID>
    <ae:type>TS_PRIVTYPE_ADMPRJ</ae:type>
  </ae:return>
  <ae:return>
    <ae:name>TS_ADMPRJPRIV_ASSIGNPROJECTPRIVS</ae:name>
    <ae:objectUUID>ROOTPROJECT</ae:objectUUID>
    <ae:type>TS_PRIVTYPE_ADMPRJ</ae:type>
  </ae:return>
  <ae:return>
```



```

<ae:name>TS_ADMPRJPRIV_ADDPROJECT</ae:name>
<ae:objectUUID>1bb8e27a-3156-49e6-8257-f7379e6aa498</ae:objectUUID>
<ae:type>TS_PRIVTYPE_ADMPRJ</ae:type>
</ae:return>
</ae:GetUserPrivilegesResponse>

```

ProjectGeneralData

Description

The ProjectGeneralData type holds the name and other information about a project in SBM. The ProjectGeneralData type parameters are listed below.

Parameters

Name	Type	Description
project	ProjectIdentifier [page 255]	Specifies the project that is created.
parentProject	ProjectIdentifier [page 255]	Specifies the new project's parent project.
workflow	WorkflowIdentifier [page 268]	Specifies the new project's parent workflow.
useParentProjectWorkflow	boolean	Determines if the project uses the parent project's workflow. Default is true if parent is not Base Project. Default is false if parent is Base Project.
projectSequence	integer	Indicates the sequence of the project in the order of siblings.
allowSubmit	boolean	Indicates whether or not the project allows items to be submitted.
useParentSequenceNumbers	boolean	Determines if items submitted to the new project are numbered in sequence with items in the parent project. False means the new project will number its items independently of items in the parent project. Default is true .

Name	Type	Description
lastItemSequenceNumber	integer	The next item submitted to the new project will be numbered one greater than this number. 0 means the first item will be numbered 1. Default is 0 . Ignored if useParentSequenceNumbers is true.
zeroToFill	integer	Zero-fill item numbers within the project to a certain number of digits. For example, 5 would fill to five digits: 00001. Default is 5 . Ignored if useParentSequenceNumbers is true.
allowAnonymousSubmit	boolean	Indicates whether or not the project allows items to be submitted anonymously.
altName	string	Alternate project name to display to users who do not have view privileges on the project. Default is same as projectName.
description	string	The description of the project. Derived from the TS_DESCRIPTION column in TS_PROJECTS.

Usage

The ProjectGeneralData type holds information that completely describes a project that you create using CreateProject. For more information, see [CreateProject \[page 217\]](#).

XML

The following XML snippet shows the ProjectGeneralData type in the return element of the CreateProject response.

Status

Description

The Status type holds status information for a Web service operation. The Status type parameters are listed below.

Parameters

Name	Type	Description
status	StatusEnum [page 291] .	Status enumeration that indicates whether the message is a warning message, informational message, or error message.
code	string	The message code that is returned.
message	string	The message string that is returned.

Usage

The Status type is used to return status messages from service operations.

XML

The following XML shows Status as seen in the return element of a GetGroups call that did not include the group name.

```
<ae:return>
  <ae:status>
    <ae:status>IS-ERROR</ae:status>
    <ae:message>A group name must be specified to create a new group.</ae:message>
  </ae:status>
</ae:return>
```

StatusEnum

Description

StatusEnum indicates the type of message that is returned in the Status response.

Parameters

Name	Type	Description
IS-WARNING	string	Indicates a warning message.
IS-INFORMATION	string	Indicates an informational message.
IS-ERROR	string	Indicates an error message.

Usage

None.

XML

The following XML shows StatusEnum in the `<ae:status>` return element of a typical response.

```
<ae:return>
  <ae:status>
    <ae:status>IS-ERROR</ae:status>
    <ae:message>Invalid project 0.</ae:message>
  </ae:status>
</ae:return>
```

UserHolder

Description

The UserHolder type contains the UserInfo response and any errors for user records that were not successfully returned. The UserHolder type parameters are listed below.

Parameters

Name	Type	Description
user	UserInfo [page 261]	The userInfo type holds the complete informaton that describes a user.
status	Status [page 290]	Holds status information for any messages or failures that are encountered.

Usage

The UserHolder type holds detailed information for a user and his or her preferences. If any errors occur during the call, they are returned in the status element in the order in which they were processed.

XML

The following XML snippet shows UserHolder in the `<ae:user>` return element of the UpdateUsers response. Note that an error message is returned in this example (inside the `message` element). In this example, three users were updated using the UpdateUsers call; however, the second user (Kathy) was not properly identified. Therefore, an error was returned for the second user, though the service continued to run and returned the third user (Laura).

```
<ae:UpdateUsersResponse>
  <ae:return>
    <ae:user>
      <ae:id xsi:type="ae:UserIdentifier">
        <ae:displayName>John Support Manager</ae:displayName>
        <ae:id>21</ae:id>
        <ae:uuid>08784a43-970f-4d28-9a6e-c301077ca653</ae:uuid>
        <ae:loginId>john</ae:loginId>
      </ae:id>
      <ae:accessType>ACCESS-USER</ae:accessType>
      <ae:email>john@companyName.com</ae:email>
      <ae:emailCC/>
```

```
<ae:timezone/>
<ae:offsetFromGMT>-25200000</ae:offsetFromGMT>
<ae:dstSavings>3600000</ae:dstSavings>
<ae:datePreference>DATE-FORMAT-FROM-LOCALE</ae:datePreference>
<ae:timePreference>TIME-FORMAT-12HOUR</ae:timePreference>
<ae:namespaceName>00000</ae:namespaceName>
<ae:phoneNumber/>
<ae:locale>en_US</ae:locale>
<ae:isDeleted>>false</ae:isDeleted>
<ae:maxNotes>10</ae:maxNotes>
<ae:maxChangeHistory>10</ae:maxChangeHistory>
<ae:maxItemsPerPage>20</ae:maxItemsPerPage>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>Everyone</ae:displayName>
  <ae:id>1</ae:id>
  <ae:uuid>ade39c21-e7b2-4dcb-a231-d3d872671b59</ae:uuid>
</ae:group>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>IDM View Only</ae:displayName>
  <ae:id>5</ae:id>
  <ae:uuid>af38532d-e79c-495f-a3e6-f4bf784cc492</ae:uuid>
</ae:group>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>IM Technician</ae:displayName>
  <ae:id>6</ae:id>
  <ae:uuid>902a1300-6ae1-44d7-b46a-e420babe8497</ae:uuid>
</ae:group>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>IM Manager</ae:displayName>
  <ae:id>7</ae:id>
  <ae:uuid>b70af5de-6642-4228-b9bb-d0da59bb6909</ae:uuid>
</ae:group>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>IM Administrator</ae:displayName>
  <ae:id>8</ae:id>
  <ae:uuid>b77b6e9d-e75a-4841-be1b-9c358affb797</ae:uuid>
</ae:group>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>CR Approval Board Members</ae:displayName>
  <ae:id>13</ae:id>
  <ae:uuid>ea167b8c-e9ea-4196-9727-dfd2f10fd751</ae:uuid>
</ae:group>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>CR Submitters</ae:displayName>
  <ae:id>11</ae:id>
  <ae:uuid>cf83a358-d7fb-4b96-8f98-ed532c66cd0a</ae:uuid>
</ae:group>
<ae:fieldsMask>1</ae:fieldsMask>
<ae:notesMask>4</ae:notesMask>
<ae:changeHistoryMask>4</ae:changeHistoryMask>
<ae:browserMask>13635632</ae:browserMask>
<ae:preferredSolution xsi:type="ae:SolutionIdentifier">
  <ae:displayName>Incident Management</ae:displayName>
  <ae:id>2</ae:id>
  <ae:uuid>c6f06a70-4d06-42a4-a3a9-50b2120dca41</ae:uuid>
```

```
<ae:uniqueName>INCIDENT_MANAGEMENT</ae:uniqueName>
  <ae:tabName>Incident Mgmt</ae:tabName>
</ae:preferredSolution>
<ae:solutionData>
  <ae:solution xsi:type="ae:SolutionIdentifier">
    <ae:displayName>Incident Management</ae:displayName>
    <ae:id>2</ae:id>
    <ae:uuid>c6f06a70-4d06-42a4-a3a9-50b2120dca41</ae:uuid>
    <ae:uniqueName>INCIDENT_MANAGEMENT</ae:uniqueName>
    <ae:tabName>Incident Mgmt</ae:tabName>
  </ae:solution>
  <ae:homeReport xsi:type="ae:ReportIdentifier">
    <ae:displayName>Trend Of Incidents Closed On Initial Call</ae:displayName>
    <ae:id>38</ae:id>
    <ae:uuid>f8a5ce79-4b40-45cf-9f6d-735d060e90de</ae:uuid>
  </ae:homeReport>
</ae:solutionData>
<ae:solutionData>
  <ae:solution xsi:type="ae:SolutionIdentifier">
    <ae:displayName>Change Request Management</ae:displayName>
    <ae:id>3</ae:id>
    <ae:uuid>0ea28a74-6dde-406f-b19a-4c45aec40294</ae:uuid>
    <ae:uniqueName>CHANGE_REQUEST_MANAGEMENT</ae:uniqueName>
    <ae:tabName>Change Requests</ae:tabName>
  </ae:solution>
  <ae:homeReport xsi:type="ae:ReportIdentifier">
    <ae:displayName>Built-In: All Active Items I Own</ae:displayName>
    <ae:id>-6</ae:id>
    <ae:uuid>-6</ae:uuid>
  </ae:homeReport>
</ae:solutionData>
<ae:solutionData>
  <ae:solution xsi:type="ae:SolutionIdentifier">
    <ae:displayName>Issue Defect Management</ae:displayName>
    <ae:id>1</ae:id>
    <ae:uuid>cea0a86c-5d74-4e12-b8d6-9d6b90186f1e</ae:uuid>
    <ae:uniqueName>ISSUE_DEFECT_MANAGEMENT</ae:uniqueName>
    <ae:tabName>IDM</ae:tabName>
  </ae:solution>
  <ae:homeReport xsi:type="ae:ReportIdentifier">
    <ae:displayName>Built-In: All Active Items I Own</ae:displayName>
    <ae:id>-6</ae:id>
    <ae:uuid>-6</ae:uuid>
  </ae:homeReport>
</ae:solutionData>
</ae:user>
</ae:return>
<ae:return>
  <ae:status>
    <ae:status>IS-ERROR</ae:status>
    <ae:message>Invalid user Kath.</ae:message>
  </ae:status>
</ae:return>
<ae:return>
  <ae:user>
```

```
<ae:id xsi:type="ae:UserIdentifier">
  <ae:displayName>Laura Engineer</ae:displayName>
  <ae:id>5</ae:id>
  <ae:uuid>e0538593-21aa-4ca5-a229-473563c21470</ae:uuid>
  <ae:loginId>laura</ae:loginId>
</ae:id>
<ae:accessType>ACCESS-USER</ae:accessType>
<ae:email>laura@companyName.com</ae:email>
<ae:emailCC/>
<ae:timezone/>
<ae:offsetFromGMT>-25200000</ae:offsetFromGMT>
<ae:dstSavings>3600000</ae:dstSavings>
<ae:datePreference>DATE-FORMAT-FROM-LOCALE</ae:datePreference>
<ae:timePreference>TIME-FORMAT-12HOUR</ae:timePreference>
<ae:namespaceName>00000</ae:namespaceName>
<ae:phoneNumber/>
<ae:locale>en_US</ae:locale>
<ae:isDeleted>false</ae:isDeleted>
<ae:maxNotes>10</ae:maxNotes>
<ae:maxChangeHistory>10</ae:maxChangeHistory>
<ae:maxItemsPerPage>20</ae:maxItemsPerPage>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>Everyone</ae:displayName>
  <ae:id>1</ae:id>
  <ae:uuid>ade39c21-e7b2-4dcb-a231-d3d872671b59</ae:uuid>
</ae:group>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>IDM Team</ae:displayName>
  <ae:id>2</ae:id>
  <ae:uuid>634aed4a-bc98-49d3-86f0-6095c2f7b9b6</ae:uuid>
</ae:group>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>CR Technical Analysts</ae:displayName>
  <ae:id>12</ae:id>
  <ae:uuid>661c1c5b-2e1d-4ad1-9d43-60a3dad6f931</ae:uuid>
</ae:group>
<ae:group xsi:type="ae:GroupIdentifier">
  <ae:displayName>CR Submitters</ae:displayName>
  <ae:id>11</ae:id>
  <ae:uuid>cf83a358-d7fb-4b96-8f98-ed532c66cd0a</ae:uuid>
</ae:group>
<ae:fieldsMask>1</ae:fieldsMask>
<ae:notesMask>4</ae:notesMask>
<ae:changeHistoryMask>4</ae:changeHistoryMask>
<ae:browserMask>13635632</ae:browserMask>
<ae:preferredSolution xsi:type="ae:SolutionIdentifier">
  <ae:displayName>Issue Defect Management</ae:displayName>
  <ae:id>1</ae:id>
  <ae:uuid>cea0a86c-5d74-4e12-b8d6-9d6b90186f1e</ae:uuid>
  <ae:uniqueName>ISSUE_DEFECT_MANAGEMENT</ae:uniqueName>
  <ae:tabName>IDM</ae:tabName>
</ae:preferredSolution>
<ae:solutionData>
  <ae:solution xsi:type="ae:SolutionIdentifier">
    <ae:displayName>Issue Defect Management</ae:displayName>
```

```
    <ae:id>1</ae:id>
    <ae:uuid>cea0a86c-5d74-4e12-b8d6-9d6b90186f1e</ae:uuid>
    <ae:uniqueName>ISSUE_DEFECT_MANAGEMENT</ae:uniqueName>
    <ae:tabName>IDM</ae:tabName>
  </ae:solution>
  <ae:homeReport xsi:type="ae:ReportIdentifier">
    <ae:displayName>My In Box</ae:displayName>
    <ae:id>54</ae:id>
    <ae:uuid>034982d7-822e-421e-88cf-1fe2cb436785</ae:uuid>
  </ae:homeReport>
</ae:solutionData>
<ae:solutionData>
  <ae:solution xsi:type="ae:SolutionIdentifier">
    <ae:displayName>Incident Management</ae:displayName>
    <ae:id>2</ae:id>
    <ae:uuid>c6f06a70-4d06-42a4-a3a9-50b2120dca41</ae:uuid>
    <ae:uniqueName>INCIDENT_MANAGEMENT</ae:uniqueName>
    <ae:tabName>Incident Mgmt</ae:tabName>
  </ae:solution>
  <ae:homeReport xsi:type="ae:ReportIdentifier">
    <ae:displayName>Built-In: All Items I Submitted</ae:displayName>
    <ae:id>-11</ae:id>
    <ae:uuid>-11</ae:uuid>
  </ae:homeReport>
</ae:solutionData>
<ae:solutionData>
  <ae:solution xsi:type="ae:SolutionIdentifier">
    <ae:displayName>Change Request Management</ae:displayName>
    <ae:id>3</ae:id>
    <ae:uuid>0ea28a74-6dde-406f-b19a-4c45aec40294</ae:uuid>
    <ae:uniqueName>CHANGE_REQUEST_MANAGEMENT</ae:uniqueName>
    <ae:tabName>Change Requests</ae:tabName>
  </ae:solution>
  <ae:homeReport xsi:type="ae:ReportIdentifier">
    <ae:displayName>Built-In: All Items I Submitted</ae:displayName>
    <ae:id>-11</ae:id>
    <ae:uuid>-11</ae:uuid>
  </ae:homeReport>
</ae:solutionData>
</ae:user>
</ae:return>
</ae:UpdateUsersResponse>
```

Chapter 4: Tips for Writing Applications

This section provides guidelines for getting started writing applications that interact with the SBM Web services.

- [Authentication Methods \[page 297\]](#)
- [C++ Constants \[page 298\]](#)
- [Supported Character Encoding \[page 298\]](#)
- [Supported Date/Time Formats \[page 298\]](#)
- [Application, Table, and Project IDs \[page 299\]](#)

Authentication Methods

The SBM Web services API supports the following authentication methods. Choose the method most appropriate for your environment. Use of SSL is recommended with any of these authentication methods.

Argument

By default, the SBM Web services use the argument method of authentication. Authentication occurs each time a method is called. The Auth argument passes the SBM user ID and password in plain text. You can also use this argument to specify the host name for licensing purposes, instead of using the client's IP address.

HTTP Basic

HTTP Basic authentication is defined in the HTTP header. The SBM user ID and password are passed with Base64 encoding.

To define HTTP Basic authentication in Visual Studio .NET, add an authorization header to the request. Typically you do this by overriding the GetWebRequest method. In the sample programs, you can use the `-basic` command-line argument to do this.

WS-Security

WS-Security (Web Services Security) authentication creates a security token in the SOAP header. The SBM credentials are passed as a Username token and a Base64-encoded password.

To define WS-Security in Visual Studio .NET, add a Username token to the SOAP header, which contains the plain text user ID and the Base64-encoded password. You must also have Microsoft's WSE installed.

C++ Constants

For reference, SBM provides constants in C++ format. You can find these constants in the `TSDef.h` file, located in the SBM API package (`tsapi.zip`).

For information on the SBM database, see the `schema.doc` file, also located in the API package.

Supported Character Encoding

The SBM Web services use UTF-8 encoding. Single-byte characters are automatically supported with UTF-8. To enable support for multi-byte characters, you must set the expected encoding to UTF-8 on the client side. In Visual Studio 2005 C#, you can do this by overriding the `GetReaderFromMessage` method and setting the reader's encoding to UTF-8.

Supported Date/Time Formats

SBM uses the ICU (International Components for Unicode) library (version 3.6) for date/time support. The SBM *Date/Time* field values are strings and must be in XML date/time format. The value is expected to be in coordinated universal time (UTC). Below are four examples of date/time strings that you can send to the SBM Web services:

#	Example	Explanation
1	2006-04-28	No offset from 00:00:00, April 28, 2006 — or — 2006-04-28T00:00:00+00:00
2	2006-04-28T13:20-6:00	6-hour offset from 1:20:00 p.m., April 28, 2006 — or — 2006-04-28T19:20:00+00:00
3	2006-04-28T19:20:30+0:00	No offset from 7:20:30 p.m., April 28, 2006
4	2006-04-28T13:20:30.45-6:00	6-hour offset from 1:20:30 p.m., April 28, 2006 — or — 2006-04-28T19:20:30+00:00

If you use the format shown in example 1, the SBM Web services return *Date Only* field values in the same format. If you use the formats shown in examples 2, 3, or 4, the SBM Web services return *Date/Time* field values in the same format as example 3.



Note: SBM does not currently store the fractions of a second when the format in example four is used.

Web service date/time values that are sent with milliseconds must be 29 characters in length. For example, `2006-04-28T13:20:30.45-6:00` results in an invalid value response; however, `2006-04-28T13:20:30.450-06:00` is accepted. Therefore, specify the value with

milliseconds in exactly 29 characters or remove the milliseconds from the value (2006-04-28T13:20:30-6:00).

The SBM Web services API has two special system *Date/Time* fields, `item.createDate` and `item.modifiedDate`, which have values in seconds since January 1, 1970 (the modified Julian date). These fields are automatically populated by SBM and it is recommended that you do not modify them. SOAP will convert these values to its `dateTime` format for transport in XML.

You can explicitly set these values as shown in the following C# example code:

```
string MyString = "Jan 1, 2006 12:12:12 am";

DateTime MyDateTime = DateTime.Parse(MyString);

web.TTItem item = new web.TTItem();

item.createDate = MyDateTime;

item.modifiedDate = MyDateTime;
```

Application, Table, and Project IDs

The `GetSolutions`, `GetTables`, and `GetSubmitProjects` methods are a useful starting point to retrieve IDs of applications, tables, and projects. These IDs are required when you use certain methods, such as `CreatePrimaryItem` and `DeleteItem`.

Chapter 5: Sample Programs

- [About the Sample Programs \[page 301\]](#)
- [Using the Sample Programs \[page 303\]](#)

About the Sample Programs

SBM provides seven sample program source files that you can use with the SBM sample database. Use these programs to see functional examples of C# source code for applications that interact with SBM Web services. You can find these samples in your installation under `Application Engine\webservices\samples`.

The samples include:

- [Solutions and Tables \[page 301\]](#)
- [CreateItems \[page 302\]](#)
- [DeleteItems \[page 302\]](#)
- [GetItems \[page 302\]](#)
- [Reports \[page 302\]](#)
- [UpdateItems \[page 302\]](#)
- [Users and Groups \[page 303\]](#)



Important: Support for development efforts writing Web services is provided by Professional Services. Questions regarding use of Web services operations in orchestration processes as documented are handled by customer support.

Solutions and Tables

Demonstrates returning a list of applications, tables, and projects.

This program:

1. Returns the applications for which the user has permissions.
2. Returns the tables and fields for the Issue Defect Management application.
3. Returns all auxiliary tables and fields.
4. Returns all primary tables and fields.
5. Returns the tables and fields for which the user has permissions.
6. Returns the projects for which the user has submit permissions for the Issues table.
7. Returns the projects for which the user has submit permissions.

CreateItems

Demonstrates submitting items into a specified SBM project, and creating auxiliary items within a specified table.

This program:

1. Creates items in the Contacts auxiliary table.
2. Creates items in the Image Builder project.

DeleteItems

Demonstrates deleting specified items.

This program:

1. Creates an item in the Image Builder project and then deletes the item.
2. Creates two items in the Image Builder project and then deletes the items.
3. Creates two items in the Image Builder project and then deletes the item using a query string.

GetItems

Demonstrates returning items based on user privileges.

This program:

1. Returns all items from the Issues table.
2. Deals with file attachments on an item.

Reports

Demonstrates getting and executing reports.

This program:

1. Gets all built-in reports.
2. Gets all reports that were created by Joe.
3. Gets three reports from the IDM application that were created by Joe.
4. Executes the "All Active Escalated Incidents" report by report name and returns all items.
5. Executes the "All Issues I Own" report by UUID and returns four items.
6. Executes the "Built-In: All Inactive Items" built-in report in the Incident Management application and returns all items.

UpdateItems

Demonstrates updating item fields and updating items using a specified transition.

This program:

1. Creates an item in the Image Builder project.
2. Updates the title and description of the item.
3. Updates the item using the transition ID 0.
4. Creates two more items in the Image Builder project.
5. Updates the two items.

Users and Groups

Demonstrates getting, updating, and creating users and groups.

This program:

1. Gets users and groups.
2. Updates a user and a group.
3. Creates a user and a group.
4. Marks a user and a group as deleted.

Using the Sample Programs

The following sections discuss the requirements for using the sample programs and what you should modify in the programs:

- [Requirements \[page 303\]](#)
- [Choosing Authentication \[page 304\]](#)
- [Editing the Web Services URL \[page 304\]](#)
- [Rebuilding the Executables \[page 304\]](#)

Requirements

You must have the following installed:

- SBM 2009 R1 or later, connected to the sample database
- One of the following:
 - Microsoft Visual Studio .NET 2005
- For use with WS-Security authentication (optional):
 - WSE 3.0 (download from [here](#)).



Note: Later versions of Visual Studio may be used; however, WSE is not supported in later versions, so the WSE specific code must be removed or replaced with Windows Communication Foundation (WCF) code. This only applicable when WS-Security authentication is used.

Choosing Authentication

By default, the samples use the argument method for authentication. You can change the authentication method to HTTP Basic by using the `-basic` command-line argument.

For a description of the authentication methods, see [Authentication Methods \[page 297\]](#).

Editing the Web Services URL

By default, the SBM Web services URL points to a server called `localhost:80`. To point to your actual server, locate the following files in each sample program directory:

- `Sample*.cs`

Change the server name and port number in this line:

```
sbmserv.Url = "http://serverName:aePort/gsoap/gsoap_ssl.dll?sbmappservices72";
```

The port number can be changed from the default value of 80 using the `- port` argument.

Rebuilding the Executables

The sample programs include pre-built executables. If you have changed the Web services URL or the authentication method, rebuild the solution and use the new executables.

You can find the executables in the `bin\release` folder.