



Adaptive Planning Data Integration API

Copyright 2010-2012 Adaptive Planning, Inc. All Rights Reserved. This work contains trade secrets and confidential material of Adaptive Planning, Inc., and its use or disclosure in whole or in part without the express written permission of Adaptive Planning, Inc., is prohibited.

Last Updated January 2012

Adaptive Planning, Inc. Data Integration API

Table of Contents

1.	INTRODUCTION	1
2.	WEB SERVICES URL AND NAME SPACE.....	1
3.	SECURITY.....	1
4.	AUTHENTICATION.....	1
5.	IMPORTDATASERVICE.....	3
	STANDARD IMPORT REQUEST FORMAT	3
	CUBE SHEET IMPORT REQUEST FORMAT	4
	MODELED SHEET IMPORT REQUEST FORMAT	4
	TRANSACTION IMPORT REQUEST FORMAT	5
	TAG DETAILS	5
	RESPONSE FORMAT.....	8
6.	EXPORTDATASERVICE.....	10
	EXPORT REQUEST FORMAT	10
	TAG DETAILS	10
	RESPONSE FORMAT.....	14
7.	APPENDIX A – SAMPLE SESSION	15

Adaptive Planning, Inc. Data Integration API

1. Introduction

Adaptive Planning offers a set of SOAP-based web services available for access by any customer. These Document/literal web services are available by accessing a specified URL for each service over HTTPS.

2. Web Services URL and Name Space

The URL to which web service calls should be directed is <https://live.adaptiveplanning.com>. Note that web services for Adaptive Planning are only available over HTTPS. There is no non-SSL access available. All web service clients calling Adaptive Planning web services must be able to encrypt and decrypt standard SSL communications.

The Adaptive Planning web service namespace name is <http://webservices.adaptiveplanning.com/>. To avoid collision with non-Adaptive Planning web services, client requests need to indicate this namespace name. Similarly, client programs should note it when parsing responses to their Adaptive Planning requests. See **Appendix A** for an example of the namespace name in use.

3. Security

We apply the SSL and SOAP standards to our web services. Our web services are stateless, single-action requests, freeing us from security issues associated with more involved web services constructs such as conversations and global resources.

The user must be authenticated on each separate invocation, and thus there is not any possibility of an intruder attempting to hijack an existing SSO session.

We require that all interactions with our web services be encrypted, and we ensure this by requiring the use of SSL to contact our web services layer; the SSL layer performs the encryption.

The authentication credentials (login ID and password) are transmitted to Adaptive Planning as part of the web services request. This means that the login ID and password are also encrypted by the SSL layer before they leave the computer generating the web services request, and are only decrypted once received by the target Adaptive Planning server.

4. Authentication

Authentication of users is done within the body of the web services call, using an Authentication XML tag. A user's email and password must be transmitted in the web service request. These credentials are protected by the fact that communications with Adaptive Planning web services are all encrypted using SSL.

Authenticating a user does not create a persistent "user session" for this user – each separate web service call must authenticate the user separately.

Adaptive Planning, Inc. Data Integration API

There is no special permission required for an Adaptive Planning user to be able to access web services. However, the user making the web service request must have the required permissions to actually *perform* the action being requested. For example, to call the importData web service, a user must have the Import permission.

Each individually-listed Web Service in this document will list the permission required of users to invoke the service.

Adaptive Planning, Inc. Data Integration API

5. ImportDataService

The ImportData WSDL is available over the web at:

<https://live.adaptiveplanning.com/importData?wsdl>

The schema is available at:

<https://live.adaptiveplanning.com/importData?xsd=1>

Web service name	ImportDataService
Web service function/operation name	importData
Permission required to invoke web service	Import Capabilities

The ImportData web service accepts one of three types of data:

- Standard data for standard accounts: GL accounts, Custom accounts, or Assumptions.
- Sheet-based data for modeled sheets or cube sheets. Sheet-based imports must include a **Sheet** element to indicate which sheet's data is being imported.
- Transaction data for importing transaction data into companies that have the Transactions feature enabled.

All ImportData invocations include **RowHeader** and **Row** elements to specify the individual rows and columns of values to be imported.

Here are examples of each import request. The examples are followed by the details of each tag.

Standard Import Request Format

```
<Request>
  <Authentication email="myemail@company.com" password="my_pwd" />
  <Data importType="Actuals" moveBPointer="true"
    parallelActualsImportAllowed="true" locale="en-US">
    <Version name="Actuals" isDefault="true/false">
      <RowHeader val="Account|Plan|Split Label|Region|Sep-2009|Oct-2009">
        <Row val="Code1|Plan 1|Split Label|Region3|1.12|23.35"/>
        <Row val="Code1|Plan 2|Split Label|Region3|10|20"/>
        <Row val="Code1|Plan 3||100.999|222.234"/>
        <Row val="Code2|Plan 1||82213|82213"/>
      </RowHeader>
    </Version>
  </Data>
</Request>
```

Standard imports do not have a Sheet element (in contrast see, for example, the Cube Sheet Import Request Format below)

Note that, similar to Adaptive Planning's import spreadsheet format, some restrictions apply to the format of RowHeader and Row elements for standard import: "Account" must be the first column name and "Plan" must be the second column name in the RowHeader's val attribute. The values for the accounts and plans must therefore be the first and second values in the Row element's val attribute, respectively.

Adaptive Planning, Inc.

Data Integration API

Cube Sheet Import Request Format

```
<Request>
  <Authentication email="myemail@company.com" password="my_pwd" />
  <Data importType="Actuals" moveBPointer="true"
    parallelActualsImportAllowed="true" locale="en-US">
    <Sheet type="Cube" name="Sales Cube" isPlanIndependent="false"/>
    <Version name="Actuals" isDefault="false">
      <RowHeader val="ProductFurniture|CountryRegion|FabricationMachine|
        Customer|Plan|Account|Jan-04|Feb-04"/>
      <Row val="Coffee Table|Argentina|Do-All 15 Vertical|Aeropostale|Corporate|
        UnitsSold|12|55"/>
      <Row val="Coffee Table|Brazil|Do-All 15 Vertical|Aeropostale|Corporate|
        UnitsSold|0|2"/>
    </RowHeader>
  </Version>
</Data>
</Request>
```

Cube sheet imports have a **Sheet** element, with the *type* set to "Cube".

Modeled Sheet Import Request Format

```
<Request>
  <Authentication email="myemail@company.com" password="my_pwd" />
  <Data importType="Plan" moveBPointer="false"
    parallelActualsImportAllowed="true" locale="en-US">
    <Sheet type="Modeled" name="Personnel" isPlanIndependent="false"/>
    <Version name="Budget 2010" isDefault="true">
      <RowHeader val="Plan|Title|JobCode|Region|Benefits|per|Last Name|First Name|
        ID|Start|End|Hr/Week|Pay Rate|Pay Rate Display Column"/>
      <Row val="plan name|CEO|E1|W-US|Yes|Yr|Topdog|Andy|0001|01/15/2000||
        40|200000|216000"/>
      <Row val="plan name|Doc Mgr|E2|W-US|Yes|Hr|Docs|Lori|Contor|04/21/2003||
        15|35|38"/>
    </RowHeader>
  </Version>
</Data>
</Request>
```

Modeled sheet imports have a **Sheet** element, with the *type* set to "Modeled".

For all standard and sheet-based import types, month column names must be the last set of column names in **RowHeader's** *val* tag if month columns are present in the data. The corresponding values under those month column headers in each **Row's** *val* elements must therefore be the last set of values. Of course, month columns are optional in modeled sheet imports, and may be omitted entirely, and are not used at all in transaction imports.

Adaptive Planning, Inc.

Data Integration API

Transaction Import Request Format

```
<Request>
  <Authentication email="myemail@company.com" password="my_pwd" />
  <Data importType="Transactions" parallelActualsImportAllowed="true"
    deleteStart="01/15/2010" deleteEnd="01/15/2011"
    transactionTypes="Invoice|Purchase Order"
    locale="en-US">
    <Version name="Budget 2010" isDefault="true">
      <RowHeader val="Transaction Number|Transaction Type|Transaction|Date|Memo|
        Store|Account|Plan|Description|Quantity|Amount">
        <Row val="INV100234|Invoice|03/05/2011|CustNo: 119872|Mountain View|mapped
          account|mapped plan|Logitech Mouse|1|49.99"/>
        <Row val="INV100234|Invoice|03/05/2011|CustNo: 119872|Mountain View|mapped
          account|mapped plan|Microsoft Natural Keyboard|1|69.99"/>
      </RowHeader>
    </Version>
  </Data>
</Request>
```

Tag Details

Authentication tag:

Attributes:

- **email:** (required) the user's login email
- **password:** (required) the user's password
- **instanceCode:** (optional) the alphanumeric code for the target instance, if the user has access to more than one instance in a multi-instance system. If not set, the user's default instance is used.
- **tid:** (optional): reserved for internal use

Contents:

None

Data tag:

Attributes:

- **importType:** (required) the literal string "Actuals", "Plan", or "Transactions", indicating whether this data will be inserted into the Actuals version, a Plan version, or Transactions in Adaptive Planning.
- **moveBPointer:** (required for standard and sheet-based imports, not used for transactions) either "true" or "false". If set to "true", then an import of Actuals will make the actuals "available" in the Adaptive Planning system up to and including the latest month of data found in this import. Actuals which are not yet made "available" do not appear in Adaptive Planning except when explicitly viewing the Actuals versions (that is, they do not overlay plan versions, as "available" actuals do). If set to "false", an import of actuals data will have no impact on the availability of actuals. Note that this attribute is ignored if Plan data is being imported instead of Actuals data. This attribute can be thought of as marking an import of a given month's actuals as being "final" – until an import is marked "final", all Actuals data is considered incomplete and is not generally shown to end users.
- **parallelActualsImportAllowed:** (required) either "true" or "false". If set to "true", then an import will proceed even if there is already another actuals or transactions import in process for this customer. If set to "false", then an attempt to import will fail if there is already an actuals or transactions import being processed for this customer.

Adaptive Planning, Inc. Data Integration API

- **locale:** (required) name of the locale under which the data should be interpreted. Note that this is really only significant to allow Adaptive Planning to correctly interpret dates in the RowHeader and Row elements, and decimal separators in currency amounts and other number amounts. Some locales use commas to separate decimal portions of currencies and other numbers, while others use commas for thousands separators and periods for decimal separators.
- **deleteStart** and **deleteEnd:** (optional for transactions, not used in others) the start and end dates, respectively of the date range for deletion of existing transactions currently stored in the system. If present, all existing transactions that fall within the date range will be deleted during the import. If only one of these is specified, the other is unbounded. For example, if **deleteStart** is not specified, but **deleteEnd** is specified, the date range will consist of all dates up through **deleteEnd**. Can be used in combination with **transactionTypes**.
- **transactionTypes:** (optional for transactions, not used in standard and sheet-based) a pipe-separated list of the transaction types of existing transactions to be deleted. If present, all transactions currently stored in the system with the types listed will be deleted during the import. When used in combination with **deleteStart** and **deleteEnd**, all transactions of the types listed within the specified date range will be deleted during the import.

Contents:

One optional Sheet tag, and one required Version tag

Sheet tag (required for modeled and cube imports, not used for transactions):

The sheet tag is used only for import into modeled sheets or cube sheets, and is omitted for standard model imports.

Attributes:

- **type:** (required) either "Modeled" or "Cube", indicating whether this is a modeled sheet import or a cube sheet import.
- **name:** (required) the name of the sheet into which imported data will be placed.
- **isPlanIndependent:** (required) either "true" or "false", indicates whether the sheet named in the **name** attribute is a plan-independent sheet (found under the "Assumptions" tab), or a normal plan-dependent sheet (found under the "Sheets" tab).

Contents:

None

Version tag:

Attributes:

- **name:** (required) either the literal string "Actuals" if the import is being sent into the Actuals version, otherwise the name of the version into which the import should be performed. For transaction imports any existing version name is valid.
- **isDefault:** (optional) either "true" or "false". If set to "true", the import will take place in the version currently set to be the default working version, and the version's name will not be considered. Otherwise, if set to "false", the import will take place in the version named in the **name** attribute above.

Contents:

Exactly one RowHeader tag

RowHeader tag:

Attributes:

Adaptive Planning, Inc. Data Integration API

- **val:** (required) a pipe-separated list of column headers. These are the equivalent column headers as are found in Excel spreadsheets being uploaded through the standard GUI interface. The data imported through the web service simulates importation of a spreadsheet, with data being entered row by row. The RowHeader tag instructs the system how to interpret the various columns found on each row of data. These column headers must come in the following order, and contain:
 - **Account:** (required for standard, cube, and transactions, prohibited for modeled). For standard, cube, and transaction imports, one column of data in the import must indicate the code of the account being imported for each row. Accounts are prohibited for modeled imports.
 - **Plan:** (required). One column of data in the import must indicate the name of the plan to which each row of data will be imported.
 - **Split Label:** (required for standard import, prohibited for modeled, cube, and transactions). A Split Label column must be specified in the RowHeader, and all Rows must specify a split (any rows which do not target a split must leave that column blank). Rows targeting a split may leave the Split Label column blank if they specify dimension values instead (see below).
 - **DimensionName:** (optional for standard, required for modeled, cube, and transactions; multiple dimensions are allowed). For standard imports, if any of the enclosed Row tags of data are intended to import into individual splits and those splits are intended to be tagged with particular dimensional values, the name of the dimension with which those splits will be tagged must appear as a column header. If more than one dimension is to be used to tag splits created during import, then each dimension must have its own column and its name must appear in the RowHeader. Dimension names must appear exactly as they appear in the Adaptive Planning interface in order to be accepted. For modeled, cube, and transaction imports, all dimensions defined on the sheet must appear as column headers.
 - **OtherColumn:** (prohibited for standard and cube, optional for modeled and transactions; multiple columns are allowed). For modeled imports and transactions, data which is to be imported into non-dimensional columns in the modeled sheet and in transactions (e.g. text columns, date columns, or numeric columns) can be specified by naming the columns to be imported in the RowHeader. For transaction imports, columns must be specified for each of the following required transaction columns: Transaction Type, Transaction Date, and Transaction Amount. Names for these columns must match those specified in the transaction builder user interface.
 - **Month:** (at least one required for standard and cube, optional for modeled, prohibited for transactions). For standard and cube imports, at least one column header found in the RowHeader tag must be a month to which data should be imported. The formatting of the date is as seen in the printable view of a sheet for the locale specified for this Data (formats for different locales at http://en.wikipedia.org/wiki/Calendar_date). Month names are not case-sensitive. Multiple months may be specified, all separated by the pipe symbol, in which case this import will simultaneously import into multiple months. For modeled imports, months may appear as column headers, but they are not required, and are only allowed if the modeled sheet has a Timespan (set of month columns) on it.

Contents:

One or more Row tags

Row tag:

Attributes:

Adaptive Planning, Inc. Data Integration API

- **val:** (required) a pipe-separated list of values to be imported. These values must come in the same order as the column headers found in the enclosing RowHeader element.
 - **Account:** (required for standard, cube, and transactions, prohibited for modeled). Every standard and cube row of data must specify an account into which to import. The external account code given here will be mapped using pre-existing Import Mappings. If no mapping is found for an imported account, an error will result and the import will be cancelled.
 - **Plan:** (required). Every row of data must specify a plan into which to import. The plan name given here will be mapped using the pre-existing Import Mappings. If no mapping is found for an imported plan, an error will result and the import will be cancelled.
 - **Split Label:** (optional for standard import). Each standard import row may list an optional split label into which to import. If this row of data is not intended to be imported into an individual split, this field should be left blank (but must still be present, to keep the number of columns consistent between rows).
 - **DimensionName:** (optional for standard and transactions, required for modeled and cube). For each Dimension column header listed in the RowHeader tag, each row must list a dimensional value from that dimension with which to tag any splits of data. The dimensional value given here will be mapped using the pre-existing Import Mappings. If no mapping is found for an imported plan, an error will result. For standard import, if this row of data is not intended to be imported into an individual split, this field should be left blank (but must still be present, to keep the number of columns consistent between rows). Similarly, for transaction imports, the field may be left blank if desired.
 - **OtherColumn:** (prohibited for standard and cube, optional for modeled and transactions). For each column listed in the RowHeader, the corresponding data value must be inserted into the Row. For transaction imports, values must be entered for each of the following required transaction fields: Transaction Type, Transaction Date, and Transaction Amount. Numbers and dates are interpreted using the locale specified in the Data element.
 - **Month:** (at least one required for standard and cube, optional for modeled, prohibited for transactions). The value to be imported into the month specified in the RowHeader for this account, plan, and split/dimension combination. The value will be interpreted using the locale specified in the Data element. Blank values are not currently supported during import: every month on a row must have a non-blank numeric value.

Contents:
None

Response Format

```
<Response success="true/false">  
  <Messages>  
    <Message> Authorization failed for user myemail@company.com </Message>  
  </Messages>  
</Response>
```

A successful response (i.e. ones with success="true") may contain **Message** tags within the enclosing **Messages** element indicating individual rows that were not imported, along with the cause of each row's import failure. Individual rows may not be imported if they have invalid or missing critical portions of their *val* attribute, such as account code or plan name, or have invalid/unmapped dimension or other values.

Adaptive Planning, Inc. Data Integration API

A failed import response message (with success="false") contains **Message** tags indicating why the import failed. Errors fall into a few categories:

- Authentication and authorization failure
- Invalid combination of attributes and values, such as *importType* set to "Actuals" while *Sheet type* is set to "Modeled", or attempting to import into a locked version.
- Invalid **RowHeader** attribute values, or invalid **Sheet** or **Version** names
- Improper import mappings or invalid locale

Adaptive Planning, Inc. Data Integration API

6. ExportDataService

The ExportData WSDL is available over the web at:

<https://live.adaptiveplanning.com/exportData?wsdl>

The schema is available at:

<https://live.adaptiveplanning.com/exportData?xsd=1>

Web service name	ExportDataService
Web service function/operation name	exportData
Permission required to invoke web service	Export Capabilities

Export Request Format

```
<ExportRequest>
  <Authentication email="myemail@company.com" password="my_pwd" />
  <Version name="Budget 2010" isDefault="false"/>
  <Format useInternalCodes="true" includeUnmappedItems="false"
    locale="en-US"/>
  <Filters>
    <Accounts>
      <Account code="A100" isAssumption="true"
        includeDescendants="false"/>
      <Account code="L100" isAssumption="false"
        includeDescendants="true"/>
    </Accounts>
    <Plans>
      <Plan name="Development" isRollup="true"
        includeDescendants="true"/>
      <Plan name="QA" isRollup="false" includeDescendants="false"/>
    </Plans>
    <TimeSpan start="Jan-2010" end="Dec-2010"/>
  </Filters>
  <Dimensions>
    <Dimension name="Product"/>
    <Dimension name="CountryRegion"/>
  </Dimensions>
  <Rules useCorporateCurrency="true" includeZeroRows="false"
    includeRollups="false"/>
  <Type>D</Type>
</ExportRequest>
```

Tag Details

Authentication tag (required):

Attributes:

- **email:** (required) the user's login email
- **password:** (required) the user's password
- **instanceCode:** (optional) the alphanumeric code for the target instance, if the user has access to more than one instance in a multi-instance system. If not set, the user's default instance is used.
- **tid:** (optional) reserved for internal use

Contents:

None

Adaptive Planning, Inc. Data Integration API

Type tag (optional): must be listed as the last element in the request, if present. Contains one of the literal strings “D”, “V”, “A”, “P”, or “DV”. The default is “D” for data export. Values other than “D” ignore any format, filters, versions, dimensions, or rules tags specified in the request and cause the response to contain the corresponding metadata in XML form, instead of the data of the version.

Attributes:

None

Contents:

A short one or two character literal string, one of:

D = Data export (export of data; requires Export permission)

V = Versions (export of version metadata; requires Version Admin permission)

A = Accounts (export of account metadata; requires Model Admin permission)

P = Plans (export of plan tree metadata; requires full Plan Admin permission)

DV = Dimensions/DimValues (metadata; requires Model Admin permission)

Version tag (optional):

Attributes:

- **name:** (optional) either the literal string “Actuals” if the data to be exported is in the Actuals version, otherwise the name of the version from which the export should be performed.
- **isDefault:** (optional) either “true” or “false”. If set to “true”, the data will be exported from the default working version, and the version’s name will not be considered. Otherwise, if set to “false”, the export will take place in the version named in the **name** attribute above.
- **id:** (optional) id of the version from which export should be performed. Set it to null if version name is specified.

Contents:

None

Format tag (optional):

Attributes:

- **useInternalCodes:** (optional) either “true” or “false”. Default is “false”. If set to “true”, the export will use Adaptive Planning internal codes for accounts and plans, otherwise it will use their external (mapped) names.
- **includeUnmappedItems:** (required) either “true” or “false”. Default is “false”. If set to “true”, then if the useInternalCodes attribute above has been set to “false”, the export will include data for accounts and plans that do not have external (mapped) names and will export that data using their internal codes, while items which are mapped will use their external names.
- **locale:** (optional) name of the locale under which the data should be exported. Default is “US”. Note that this is really only significant to allow Adaptive Planning to format correctly dates in the RowHeader and decimal signs in currency amounts. Some locales use commas to separate decimal portions of currencies, while some use commas only for thousands separators and periods for decimal separators.
- **useIds:** (optional) either “true” or “false”. Default is “false”. If set to “true”, the export will use ids for accounts and dimensions, otherwise it will use the names.

Contents:

None

Adaptive Planning, Inc. Data Integration API

Filters tag (optional):

Contains tags for **Accounts**, **Plans**, and a **TimeSpan** whose associated values are to be included in the export. All values not associated with these filters will be excluded from the export. Filters are optional. Some or all of the enclosed tags may be left out of the export request. If no filters are included and `uselids` is false, no filtering will occur and no values will be excluded from the export based on their associations with accounts, plans, or specific time periods.

Attributes:

None

Contents:

- At most one **Accounts** tag, which would have no attributes and, as its contents one, or more **Account** tags
- At most one **Plans** tag, which would have no attributes and, as its contents, one or more **Plan** tags
- At most one **TimeSpan** tag
- At most one **AccountIds** tag
- At most one **DimensionIds** tag

Account tag (optional, many allowed):

Attributes:

- **code**: (required) indicates the Adaptive Planning account code for an account whose associated data is to be exported.
- **isAssumption**: (required) either "true" or "false". If set to "true" the account is an assumption, otherwise it is a non-assumption account.
- **includeDescendants**: (required) either "true" or "false". If set to "true", the export will include data for all descendant accounts under this account.

Contents:

None

Plan tag (optional, many allowed):

Attributes:

- **name**: (required) the name of a plan whose associated data will be exported.
- **isRollup**: (required) either "true" or "false". If set to "true", the rollup values associated with the plan will be exported. Otherwise, the non-rollup uncategorized values for the plan will be exported.
- **includeDescendants**: (required) either "true" or "false". If set to "true", the export will include data for the descendants of this plan. If `isRollup` above is "true" both rollup and uncategorized values will be exported, otherwise only uncategorized values will be exported.

Contents:

None

Timespan tag (optional):

Attributes:

- **start**: (required) the first month of a timespan whose associated data will be exported. It is in the MMM-YYYY format, for example "Jan-2010".
- **end**: (required) the last month of a timespan whose associated data will be exported. It is in the MMM-YYYY format, for example "Jan-2010".

Contents:

Adaptive Planning, Inc. Data Integration API

None

Dimensions tag (optional):

The presence of a Dimensions tag indicates that the export will be broken down by dimension values for each Dimension tag it encloses. Rows will be provided for each valid dimension combined with a particular account/plan combination. If more than one Dimension tag is included, a row will be exported for each valid combination of dimension values for a particular account/plan combination. The Dimensions section is optional. Without it, a single row will be exported for each account/plan combination.

Attributes:

None

Contents:

One or more **Dimension** tags

Dimension tag (required, if the Dimensions tag is used):

Attributes:

- **name:** (required) the name of a dimension whose dimension values will be enumerated in the export.

Contents:

None

AccountIds tag (optional): indicates the Adaptive Planning account ids for accounts whose associated data is to be exported. Valid only if useIds in Format is true.

Attributes:

None

Contents:

None

DimensionIds tag (optional): indicates the export will be broken down by dimension values for each Dimension id it contains. Valid only if useIds in Format is true.

Attributes:

None

Contents:

None

Rules tag (optional):

Attributes:

- **useCorporateCurrency:** (optional) either "true" or "false". If set to "true", the exported values will be expressed in the corporate currency. Otherwise, the local currency for the plan upon which the data resides will be used.
- **includeZeroRows:** (optional) either "true" or "false". If set to "true", the export will include those rows that are empty, or contain only zeroes. Otherwise, zero-only rows will be excluded from the export.
- **includeRollups:** (optional) either "true" or "false". If set to "true", the export will include rows for all accounts or plans, including rollup rows for rollup accounts and plans. If account or plan filters have been indicated in the Filters tag, only those accounts and plans will have their rollups exported. Note that this has no effect on dimensions specified in the Dimensions tag.

Adaptive Planning, Inc. Data Integration API

- **markInvalidValues:** (optional) either “true” or “false”. Default is “false”. If set to “true”, the export will add “I” to invalid values. Otherwise adds “=NA()” to invalid values.
- **markBlanks:** (optional) either “true” or “false”. Default is “false”. If set to “true”, the export will send blank values as “B”. Otherwise, blank values are sent as “0”.

Contents:
None

Response Format

```
<Response success="true/false">
  <Messages>
    <Message> Authorization failed for user myemail@company.com </Message>
  </Messages>
  <Result> ... csv output ... </Result> (only for successful exports)
</Response>
```

A successful export response (i.e. one with success="true") contains a **Result** tag with the comma-separated (.csv) results of the export request if the request was for data. If the request was for metadata, the contents of the Result block is an XML block which contains an XML representation of the structure requested.

All responses contain a **Messages** element, which, in turn, may contain **Message** tags indicating the causes of any differences between a request and its response. Successful export response messages indicate why portions of the request were ignored. For example, if a Timespan in a request exceeds the start and the end dates of a Version whose export was requested, two messages will be included: <Message>Ignoring start of timespan, which precedes start of version; timespan start: Nov-2003, version start date: Dec-2003</Message> <Message>Ignoring end of timespan, which is beyond end of version; timespan end: Dec-2008, version end date: Dec-2006</Message>. A failed export response message (with success="false") contains messages indicating why the export failed. Errors can include authentication and authorization failure and invalid attribute values.

Adaptive Planning, Inc. Data Integration API

7. Appendix A – Sample Session

The following is a sample of a web service client's http session conversation with the live.adaptiveplanning.com web service. Here, the client first GET's the Adaptive Planning ExportData schema wsdl. It then receives the server's response with a reference to the ExportData schema. This is followed by an actual ExportData request POST'ed by the client, conforming to the format described in the ExportData section above. Finally, the client receives the server's response with an export of the requested data. Requests and responses are made using https, and must be encrypted and decrypted as required by that protocol.

```
GET /exportData?wsdl HTTPS/1.1
User-Agent: Java/1.6.0_17
Host: live.adaptiveplanning.com
Accept: text/html, image/gif, image/jpeg, *, q=.2, */*; q=.2
Connection: keep-alive
```

```
HTTPS/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/xml;charset=utf-8
Transfer-Encoding: chunked
Date: Sun, 19 Sep 2010 18:28:22 GMT
```

68f

```
<?xml version='1.0' encoding='UTF-8'?><!-- Published by JAX-WS RI at http://jax-
ws.dev.java.net. RI's version is JAX-WS RI 2.1.4-hudson-208-. --><!-- Generated by JAX-WS RI
at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.1.4-hudson-208-. --><definitions
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://webservices.adaptiveplanning.com/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://webservices.adaptiveplanning.com/" name="ExportDataService">
<types>
<xsd:schema>
<xsd:import namespace="http://webservices.adaptiveplanning.com/"
schemaLocation="https://live.adaptiveplanning.com/exportData?xsd=1" />
</xsd:schema>
</types>
<message name="exportData">
<part name="parameters" element="tns:exportData" />
</message>
<message name="exportDataResponse">
<part name="parameters" element="tns:exportDataResponse" />
</message>
<portType name="ExportData">
<operation name="exportData">
<input message="tns:exportData" />
<output message="tns:exportDataResponse" />
</operation>
</portType>
<binding name="ExportDataPortBinding" type="tns:ExportData">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
<operation name="exportData">
<soap:operation soapAction="" />
<input>
<soap:body use="literal" />
```

Adaptive Planning, Inc. Data Integration API

```
</input>
<output>
<soap:body use="literal" />
</output>
</operation>
</binding>
<service name="ExportDataService">
<port name="ExportDataPort" binding="tns:ExportDataPortBinding">
<soap:address location="https://live.adaptiveplanning.com/exportData" />
</port>
</service>
</definitions>
0
```

POST /exportData HTTPS/1.1

SOAPAction: ""

Accept: text/xml, multipart/related, text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2

Content-Type: text/xml;charset=utf-8"

User-Agent: JAX-WS RI 2.1.4-hudson-208-

Host: live.adaptiveplanning.com

Connection: keep-alive

Content-Length: 680

```
<?xml version='1.0' encoding='UTF-8'?><S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"><S:Body><ns2:exportData
xmlns:ns2="http://webservices.adaptiveplanning.com/"><arg0><Authentication
password="mypassword" email="me@mytestcompany.com"/><Version isDefault="true"
name="Budget 2005"/><Format locale="en-US" includeUnmappedItems="true"
useInternalCodes="true"/><Filters><Accounts><Account includeDescendants="false"
isAssumption="false" code="Total_Salary"/></Accounts><Plans><Plan
includeDescendants="false" isRollup="true" name="Corporate Finance"/></Plans><TimeSpan
end="Jun-2006" start="Jan-2005"/></Filters></arg0></ns2:exportData></S:Body></S:Envelope>
```

HTTPS/1.1 200 OK

Server: Apache-Coyote/1.1

Content-Type: text/xml;charset=utf-8

Transfer-Encoding: chunked

Date: Sun, 19 Sep 2010 18:28:22 GMT

6e

```
<?xml version='1.0' encoding='UTF-8'?><S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"><S:Body>
38e
<ns2:exportDataResponse
xmlns:ns2="http://webservices.adaptiveplanning.com/"><return>&lt;Response
success="true"&gt;&lt;Messages&gt;&lt;/Messages&gt;&lt;Result&gt;&lt;![CDATA[Acc
ount Name,Account Code,Plan Name,Jan-2005,Feb-2005,Mar-2005,Apr-2005,May-2005,Jun-
2005,Jul-2005,Aug-2005,Sep-2005,Oct-2005,Nov-2005,Dec-2005,Jan-2006,Feb-2006,Mar-
2006,Apr-2006,May-2006,Jun-2006
```

Adaptive Planning, Inc. Data Integration API

"Total Modeled Salary From Personnel Sheet(s)",Total_Salary,"Corporate
Finance
(Rollup)";4833.333333333333,4833.333333333333,4833.333333333333,4833.3333333333
33,4833.333333333333,4833.333333333333,4833.333333333333,4833.333333333333,4833.33
3333333333,4833.333333333333,4833.333333333333,4833.333333333333,4833.333333333333
3,4833.333333333333,4833.333333333333,4833.333333333333,4833.333333333333,4833.333
3333333333

```
]]&gt;&lt;/Result&gt;&lt;/Response&gt;</return></ns2:exportDataResponse></S:Body></S:Envelope>
```

0