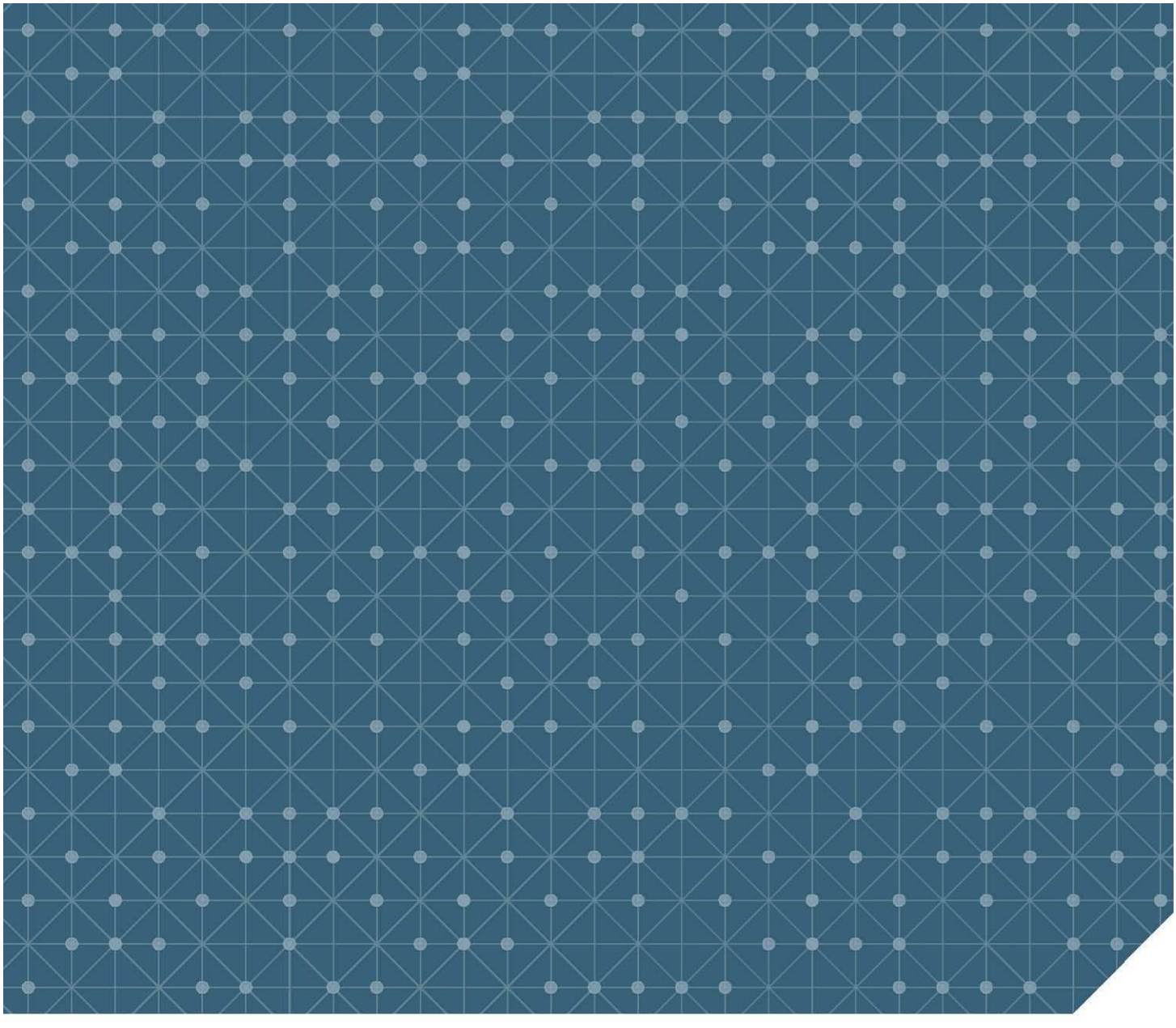
BARRAONE®

Developer's Toolkit Interactive Reference Guide



MSCI

CONTACT US

clientservice@msci.com

AMERICAS

Americas	1 888 588 4567 *
Atlanta	+ 1 404 551 3212
Boston	+ 1 617 532 0920
Chicago	+ 1 312 675 0545
Monterrey	+ 52 81 1253 4020
New York	+ 1 212 804 3901
San Francisco	+ 1 415 836 8800
Sao Paulo	+ 55 11 3706 1360
Toronto	+ 1 416 628 1007

EUROPE, MIDDLE EAST & AFRICA

Cape Town	+ 27 21 673 0100
Frankfurt	+ 49 69 133 859 00
Geneva	+ 41 22 817 9777
London	+ 44 20 7618 2222
Milan	+ 39 02 5849 0415
Paris	0800 91 59 17 *

ASIA PACIFIC

China North	10800 852 1032 *
China South	10800 152 1032 *
Hong Kong	+ 852 2844 9333
Mumbai	+ 91 22 6784 9160
Seoul	00798 8521 3392 *
Singapore	800 852 3749 *
Sydney	+ 61 2 9033 9333
Taipei	008 0112 7513 *
Thailand	0018 0015 76207 7181 *
Tokyo	+ 81 3 5290 155

* = toll free

ABOUT MSCI

For more than four decades, MSCI (NYSE: MSCI) has helped global investors build and manage better portfolios. Our research-based tools and services provide our clients with deeper insights into drivers of risk and performance, broad asset class coverage and innovative ways to bring investment strategies to market. Our offerings include indexes, data, analytical models, regulatory reporting and ESG research. MSCI's clients include 95 of the world's 100 largest money managers, as ranked by P&I. For more information, visit us at www.msci.com

NOTICE AND DISCLAIMER

This document and all of the information contained in it, including without limitation all text, data, graphs, charts (collectively, the "Information") is the property of MSCI Inc. or its subsidiaries (collectively, "MSCI"), or MSCI's licensors, direct or indirect suppliers or any third party involved in making or compiling any Information (collectively, with MSCI, the "Information Providers") and is provided for informational purposes only. The Information may not be modified, reverse-engineered, reproduced or redisseminated in whole or in part without prior written permission from MSCI.

The Information may not be used to create derivative works or to verify or correct other data or information. For example (but without limitation), the Information may not be used to create indexes, databases, risk models, analytics, software, or in connection with the issuing, offering, sponsoring, managing or marketing of any securities, portfolios, financial products or other investment vehicles utilizing or based on, linked to, tracking or otherwise derived from the Information or any other MSCI data, information, products or services.

The user of the Information assumes the entire risk of any use it may make or permit to be made of the Information. NONE OF THE INFORMATION PROVIDERS MAKES ANY EXPRESS OR IMPLIED WARRANTIES OR REPRESENTATIONS WITH RESPECT TO THE INFORMATION (OR THE RESULTS TO BE OBTAINED BY THE USE THEREOF), AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, EACH INFORMATION PROVIDER EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES (INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF ORIGINALITY, ACCURACY, TIMELINESS, NON-INFRINGEMENT, COMPLETENESS, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE) WITH RESPECT TO ANY OF THE INFORMATION.

Without limiting any of the foregoing and to the maximum extent permitted by applicable law, in no event shall any Information Provider have any liability regarding any of the Information for any direct, indirect, special, punitive, consequential (including lost profits) or any other damages even if notified of the possibility of such damages. The foregoing shall not exclude or limit any liability that may not by applicable law be excluded or limited, including without limitation (as applicable), any liability for death or personal injury to the extent that such injury results from the negligence or willful default of itself, its servants, agents or sub-contractors.

Information containing any historical information, data or analysis should not be taken as an indication or guarantee of any future performance, analysis, forecast or prediction. Past performance does not guarantee future results.

The Information should not be relied on and is not a substitute for the skill, judgment and experience of the user, its management, employees, advisors and/or clients when making investment and other business decisions. All Information is impersonal and not tailored to the needs of any person, entity or group of persons.

None of the Information constitutes an offer to sell (or a solicitation of an offer to buy), any security, financial product or other investment vehicle or any trading strategy.

It is not possible to invest directly in an index. Exposure to an asset class or trading strategy or other category represented by an index is only available through third party investable instruments (if any) based on that index. MSCI does not issue, sponsor, endorse, market, offer, review or otherwise express any opinion regarding any fund, ETF, derivative or other security, investment, financial product or trading strategy that is based on, linked to or seeks to provide an investment return related to the performance of any MSCI index (collectively, "Index Linked Investments"). MSCI makes no assurance that any Index Linked Investments will accurately track index performance or provide positive investment returns. MSCI Inc. is not an investment adviser or fiduciary and MSCI makes no representation regarding the advisability of investing in any Index Linked Investments.

Index returns do not represent the results of actual trading of investible assets/securities. MSCI maintains and calculates indexes, but does not manage actual assets. Index returns do not reflect payment of any sales charges or fees an investor may pay to purchase the securities underlying the index or Index Linked Investments. The imposition of these fees and charges would cause the performance of an Index Linked Investment to be different than the MSCI index performance.

The Information may contain back tested data. Back-tested performance is not actual performance, but is hypothetical. There are frequently material differences between back tested performance results and actual results subsequently achieved by any investment strategy.

Constituents of MSCI equity indexes are listed companies, which are included in or excluded from the indexes according to the application of the relevant index methodologies. Accordingly, constituents in MSCI equity indexes may include MSCI Inc., clients of MSCI or suppliers to MSCI. Inclusion of a security within an MSCI index is not a recommendation by MSCI to buy, sell, or hold such security, nor is it considered to be investment advice.

Data and information produced by various affiliates of MSCI Inc., including MSCI ESG Research Inc. and Barra LLC, may be used in calculating certain MSCI indexes. More information can be found in the relevant index methodologies on www.msci.com.

MSCI receives compensation in connection with licensing its indexes to third parties. MSCI Inc.'s revenue includes fees based on assets in Index Linked Investments. Information can be found in MSCI Inc.'s company filings on the Investor Relations section of www.msci.com.

MSCI ESG Research Inc. is a Registered Investment Adviser under the Investment Advisers Act of 1940 and a subsidiary of MSCI Inc. Except with respect to any applicable products or services from MSCI ESG Research, neither MSCI nor any of its products or services recommends, endorses, approves or otherwise expresses any opinion regarding any issuer, securities, financial products or instruments or trading strategies and MSCI's products or services are not intended to constitute investment advice or a recommendation to make (or refrain from making) any kind of investment decision and may not be relied on as such. Issuers mentioned or included in any MSCI ESG Research materials may include MSCI Inc., clients of MSCI or suppliers to MSCI, and may also purchase research or other products or services from MSCI ESG Research. MSCI ESG Research materials, including materials utilized in any MSCI ESG Indexes or other products, have not been submitted to, nor received approval from, the United States Securities and Exchange Commission or any other regulatory body.

Any use of or access to products, services or information of MSCI requires a license from MSCI. MSCI, Barra, RiskMetrics, IPD, InvestorForce, and other MSCI brands and product names are the trademarks, service marks, or registered trademarks of MSCI or its subsidiaries in the United States and other jurisdictions. The Global Industry Classification Standard (GICS) was developed by and is the exclusive property of MSCI and Standard & Poor's. "Global Industry Classification Standard (GICS)" is a service mark of MSCI and Standard & Poor's.

Introduction

The BarraOne Developer's Toolkit Interactive (BDTI) is a Web Service API that provides client developers granular access to BarraOne analytics. With BDTI, a developer can programmatically extract granular position plan and portfolio-level risk data from BarraOne via Web services without the need for creating report definitions in the BarraOne interface and without the use of export sets.

Web Service Overview

A Web service is a software system designed according to W3C standards that supports interoperable computer interaction over a network, often an application programming interface (API) that can be accessed over the Internet, and executed on a remote system hosting the requested services.

Most Web services, including BDTI, use SOAP-formatted XML envelopes and have their interfaces described by WSDL. SOAP is an XML-based, extensible message envelope format, with bindings to underlying protocols, such as HTTP. WSDL is an XML format that enables service interfaces to be described, along with the details of their bindings to specific protocols. WSDL technology is the basis for generating the client code that will interact with the Web service. and it frees the developer from writing much of the application code.

Developers will employ BDTI by developing a custom client application that communicates via SOAP with the BDTI Web Service. The Web service enables a client application to:

- Select, create, or modify a portfolio
- Customize a Positions Report
- Generate Positions Report data
- Generate Risk Decomposition Report data
- Generate Factor Exposure Breakdown Report data
- Generate Allocation-Selection Report data
- Generate Portfolio History Report Data
- Generate Stress Test Positions Report Data

Development Environment

Various tool vendors provide software to facilitate the development of Web service clients. Barra has used and tested two such environments:

1. Sun's JAX-WS Metro technology (http://java.sun.com/webservices/). Our Web service is built on this platform. It is our primary recommendation.

2. Microsoft's .NET development environment.

Code Generation

Although it is certainly possible to refer to the SOAP schema for each BDTI operation and directly code the XML output, a significantly less error-prone approach is to use a tool that generates code from the WSDL. Microsoft's .NET environment provides this functionality directly.

In the Java environment, Barra uses tools provided by Metro to generate the data binding. The website http://java.sun.com/webservices/ has a tutorial that is useful for becoming familiar with the product.

WSDL

One of the primary benefits of Web service technology is the ability to generate data binding classes and RMI-like function stubs from a WSDL. The WSDL is an XML schema document that defines the operations that can be performed on the Web service and describes the request and response values for those operations.

▷ **Note:** The examples in this document use the convention that any text beginning with "my," such as "myColumn," "myFilter," *etc.,* represent user variables that can be named as the user desires, so long as each variable is unique within the file.

Getting Started

This section provides a description of the steps that are required to perform the various functions supported by the BarraOne Developer's Toolkit, plus sample code fragments. It may be helpful to refer to subsequent sections and the appendices for additional information on the required inputs.

Assumptions

These instructions assume that the following statements are true:

- The client has a BarraOne license
- The client has a BarraOne Developer's Toolkit Interactive license
- The user has obtained the BDTI WSDL from Barra, available on the client support Web site: https://support.msci.com/docs/DOC-3511.
- The user has installed one of the following development environments:
  - Java SDK 5.0 and Metro libraries from Sun Microsystems (https://metro.dev.java.net/1.2/).
  - Visual Studio 2005 or 2008 from Microsoft; .Net 3.5 framework (including WCF) or higher; optional Windows SDK (http://msdn.microsoft.com/en-us/netframework/dd936009.aspx).

Data Conventions

String data types must be entered within quotation marks (" "). All other data types are entered without quotation marks.

Case Insensitivity

All inputs except Login inputs are case insensitive.

Date Format

The user must create a calendar object to be invoked when setting a date value.

▷ **Note:** All times passed to BDTI must be expressed in GMT. Likewise, all dates returned by BDTI should be interpreted in terms of GMT.

Enum Format

The enumeration format requires the user to append the name of the attribute to the enumeration value. The following example sets the value of the BaseValueType attribute of CurrentSettings to "ASSIGNED":

```
myCurrentSettings.setBaseValueType(BaseValueType.ASSIGNED);
```

In .NET:

```
myCurrentSettings.BaseValueType = BaseValueType.ASSIGNED;
```

▷ **Note:** In .NET, if an attribute for an enumerated field is not specified in the request, the first attribute in the enumerated list is sent automatically. (A default attribute cannot be designated in the wsdl.) Thus, if an enumerated attribute for a required field is not selected, the job will not fail; however, the user should not depend upon this behavior, and the value should be set explicitly.

Optional Fields in .NET

If you are programming in a .NET environment, all entries for optional fields (except for string datatype) require an additional entry to indicate that a value has been sent for the field, unless that field has a default value specified in the WSDL. The format for this entry is as follows:

```
<FieldName>Specified = true;
```

For instance, to set the value of the BaseValueType for CurrentSettings the following two entries are required:

```
myCurrentSettings.BaseValueType = BaseValueType.ASSIGNED;
myCurrentSettings.BaseValueTypeSpecified = true;
```

If the field has a default value in the WSDL, or if the field has a string datatype, this additional entry is not required.

Create the Build Environment

To create the build environment for Java, follow these steps:

1 Export the BarraOne certificate to a file (refer to "Appendix C — Exporting the BarraOne Certificate" on page 100).

2 Use the Java key tool to register the BarraOne certificate.

```
keytool -import -file [certificate file from step 1] -alias [server name]
  -keystore [pathToYourKeystore] -keypass [keyStorePassword]
```

3 Download the WSDL and sample code from the Web site to /tmp, for example.

4 Create a root directory, *e.g.,* /BDTI.

5 Create the subdirectory src under /BDTI.

6 Copy the sample code to the src directory.

7 Copy the Metro distribution from the website http://java.sun.com/webservices/, and install.

8 Generate the BDTI client-side classes by running the following command:

```
wsimport bdti.wsdl -d destfolder
```

9 Change to the /destfolder directory, and run the jar command package-generated classes. Include this jar in your IDE environment, for example:

```
jar cvf bdti.jar
```

Create BDTI Client File

Java File

The following steps are required of any Java file, regardless of the BDTI function.

1 Create a Java file using any editor (IntelliJ or Eclipse, for example).

2 In the Java file, create an import statement to import the BDTI client classes:

```
import com.barra.cp.bdti.*;
```

3 `import com.barra.cp.bdti.*;`Create a public class in the Java file. For example:

```
public class SendPortfolio;
```

.NET File

The following steps are required of any .NET file, regardless of the BDTI function.

1 Create a file using any editor (VisualStudio.NET, for example).

2 In the file, create a using statement to include the BDTInteractive.cs file. (To generate this file, run the .NET tool svcutil.exe.)

3 Create a public class in the file. For example:

```
public class SendPortfolio;
```

Configuration

In Java, set the Session Maintain property variable:

```
BDTInteractive myBrtw = new BDTInteractive();
BDTIWebServicePortType myPort = myBrtw.getBDTIWebServicePort();
((BindingProvider)myPort).getRequestContext().put(BindingProvider.ENDPOINT
  _ADDRESS_PROPERTY ,  "https://www.barraone.com/bdti/BDTInteractive");
((BindingProvider)myPort).getRequestContext().put(BindingProvider.SESSION_
  MAINTAIN_PROPERTY,true);
```

In .NET, set the following:

- In app.config, set the BDTI URL as shown below.:

```
<endpoint address = "https://www.barraone.com/bdti/BDTInteractive"
binding = "basicHttpBinding" bindingConfiguration = "BDTIWebServiceBinding"
contract = "BDTIWebServicePortType" name = "BDTIWebServicePort" />
```

- In your C# code, instantiate your BDTI port class:

```
BDTIWebServicePortTypeClient bdtiClient = new
  BDTIWebServicePortTypeClient()
```

Workgroup Permissioning

Your access to various portfolios, benchmarks, reports, and the like is controlled by the BarraOne workgroup permissioning mechanism. Refer to the BarraOne Help system for details about workgroup permissioning.

Sessions

BarraOne Developer's Toolkit Interactive is session driven. As such, the Analysis Date and any other settings are maintained for the life of the session. Developers can set up their code to maximize the benefits of having the context maintained for the life of the session. Note that an inactive session will timeout after 60 minutes.

Operations

Login

The Login element has three child elements as listed in the table below. It enables the user to log in to the BarraOne Developer's Toolkit Interactive environment, and it must be executed before any other operation may be executed.

Table 1: Login (1)

Attribute	Required?	Data Type (Length)	Description
User	Y	string	Enter your User ID.
Client	Y	string	Enter the Client ID for your firm or organization.
Password	Y	string	Enter your Password.

▷ **Note:** The ordinality of each class (*e.g.,* **Login** (**1**)), *i.e.,* the number of instances allowed per class, is shown in parentheses.

To create a Login object, send the following:

```
myPort.login("myUser", "myClient", "myPassword");
```

In .NET, send the following:

```
bdtiClient.Login("myUser", "myClient", "myPassword");
```

LoginUserAgent

The LoginUserAgent element is not used in normal BDTI processes and is invoked only for special partner operations.

GetFilters

The GetFilters element enables users to get a list of available filters associated with the specified FilterType in BarraOne. Filters are the owner names of the Portfolio Trees, Factor Trees, Models, and Scenarios in the BarraOne application. It will return a String array. The response can then be used in the element "GetPortfolioTree" on page 9, "GetFactorTree" on page 20, "GetModels" on page 36, "SetModel" on page 37, and "GetScenarioNames" on page 68.

FilterType

FilterType is the sole element of the GetFilters element.

Table 2: FilterType (0-1)

Attribute	Required?	Data Type (Length)	Description
FilterType	N	enum	The type of filter. If no value is specified, the application will default to "FilterType.PortfolioTree." Acceptable values: • FilterType.PortfolioTree • FilterType.FactorTree • FilterType.Model • FilterType.Scenario

To create a GetFilters object, send the following:

```
List<String> myFilters = myPort.getFilters(FilterType.PortfolioTree);
```

In .NET, send the following:

```
string[] myFilters = bdtiClient.GetFilters(FilterType.PortfolioTree);
```

Filter

Filter is the sole element of the String array returned by the GetFilters element.

Table 3: Filter (1-n)

Attribute	Required?	Data Type (Length)	Description
Filter	Y	string	The name of the filter. Filters are the owner names of the Portfolio Trees, Factor Trees, Models, and Scenarios in the BarraOne application.

GetPortfolioTree

The GetPortfolioTree element enables users to get a list of available portfolios and portfolio trees in BarraOne associated with the user specified as the portfolio tree filter. It will return a Folder array. This provides users with a way to locate the name of an existing portfolio or tree they want to use for their report. The output can then be used in "SetCurrentPortfolio" on page 11.

Filter

Filter is the sole child element of the GetPortfolioTree element.

Table 4: Filter (0-1)

Attribute	Required?	Data Type (Length)	Description
Filter	N	string	Enter the name of a filter. A filter is the equivalent of an item in the dropdown list (such as owner or workgroup) available in the Select Portfolio dialog in the BarraOne application. The user can obtain a list of available filters using "GetFilters" on page 8.

To create a GetPortfolioTree object, send the following:

```
List<Folder> myFolders = myPort.getPortfolioTree("myFilter");
```

In .NET, send the following:

```
Folder[] myFolders = bdtiClient.GetPortfolioTree("myFilter");
```

Folder

Folder is the root element of the Folder array returned as a response to the GetPortfolioTree element.

Table 5: Folder (0-n)

Attribute	Required?	Data Type (Length)	Description
Owner	Y	string	The User ID of the owner of the folder ("SYSTEM" unless it is a user-created folder).
Path	Y	string	The path of the folder.

PfNode

PfNode is the sole child element of the Folder element.

Table 6: Folder.PfNode (1-n)

Attribute	Required?	Data Type (Length)	Description
PfNode			PfNode is a recursive entry, in that it consists of itself plus Type and Name. This enables the entire portfolio tree to be represented in the Folder response to the GetPortfolioTree method.
Type	Y	enum	The type of node. Acceptable values: • PfNodeType.PORTFOLIO • PfNodeType.DYNAMIC • PfNodeType.SHORTCUT • PfNodeType.AGGREGATE
Name	Y	string	The name of the node.

SetCurrentPortfolio

The SetCurrentPortfolio element enables users to set the current portfolio in BDTI. It is a prerequisite to using "ModifyPortfolio" on page 42, "SavePortfolio" on page 44, "SavePortfolioAs" on page 46, "GetPositionsReport" on page 48, "SubmitPositionsReport" on page 56, "SubmitRiskDecompReport" on page 60, "SubmitFEBReport" on page 64, "GetPortfolioHistoryReport" on page 68, and "SubmitStressTestPortfolioReport" on page 69. It will return a "StatusResponse" on page 13. It has two child elements as listed in the table below.

▷ **Note:** In BarraOne, Current Settings are categorized in the following manner:

- Base Strategy (a named strategy defined in BarraOne and assigned to the portfolio)
- Delta Strategy (modifications to the Base Strategy in BarraOne—this is the strategy initially associated with the portfolio)
- Delta Session Strategy (modifications to the strategy made using the SetCurrentSettings or SetCurrentSettingsKeyVal elements during the current BDTI session)

When the SetCurrentPortfolio element is used, you will start with the Base Strategy + Delta Strategy. Any previously set Delta Session Strategy is cleared.

Table 7: SetCurrentPortfolio (1)

Attribute	Required?	Data Type (Length)	Description
PortfolioName	Y	string	Enter the short name of your portfolio. The user can obtain a list of available portfolios by using "GetPortfolioTree" on page 9. **Note:** To retrieve an aggregate portfolio when there are both aggregate and regular portfolios with the same name, preface the aggregate portfolio name with a slash (/).
PortfolioOwner	Y	string	Enter the User ID of the portfolio owner. Enter "SYSTEM" unless it is a user-created portfolio.

To create a SetCurrentPortfolio object, send the following:

```
StatusResponse myResponse = myPort.setCurrentPortfolio("myPortfolioName",
  "myPortfolioOwner");
```

In .NET, send the following:

```
StatusResponse myResponse =
  bdtiClient.SetCurrentPortfolio("myPortfolioName", "myPortfolioOwner");
```

StatusResponse

StatusResponse is retuned in response to a SetCurrentPortfolio request. It has two child elements as listed in the table below.

Table 8: StatusResponse (1)

Attribute	Required?	Data Type (Length)	Description
IsDone	Y	boolean	Indicates if the current portfolio was set successfully. Acceptable values: • true • false
RequestID	Y	string	The request ID assigned by the operation "SetCurrentPortfolio" on page 11.

CheckRequestStatus

The CheckRequestStatus element enables users to get the status of a SetCurrentPortfolio request. It has one child element, as shown below.

To create a CheckRequestStatus object, send the following:

```
StatusResponse myResponse = myPort.checkRequestStatus(“myRequestID”);
```

In .NET, send the following:

```
StatusResponse myResponse = bdtiClient.CheckRequestStatus("myRequestID");
```

RequestID

RequestID is the sole element of the CheckRequestStatus element.

Table 9: RequestID (1)

Attribute	Required?	Data Type (Length)	Description
RequestID	Y	string	Enter the request ID assigned by the operation “SetCurrentPortfolio” on page 11.

StatusResponse

StatusResponse is retuned in response to a CheckRequestStatus request. It has two child elements as listed in the table below.

Table 10: StatusResponse (1)

Attribute	Required?	Data Type (Length)	Description
IsDone	Y	boolean	Indicates if the current portfolio was set successfully. Acceptable values: • true • false
RequestID	Y	string	The request ID assigned by the operation “SetCurrentPortfolio” on page 11.

GetReportColumnName

The GetReportColumnName element retrieves the set of column names associated with an owner and specified report type. This enables users to locate the names of one or more report columns they want to use for their report. It will return a String array. The information returned can then be used in the elements "GetGroupings" on page 15, "GetPositionsReport" on page 48, "SubmitPositionsReport" on page 56, "SubmitRiskDecompReport" on page 60, Report Settings for "SubmitFEBReport" on page 64, "SubmitStressTestPortfolioReport" on page 69, "SubmitAllocationSelectionReport" on page 74, and "SubmitStressTestSummaryReport" on page 79.

Table 11: GetReportColumnName (1)

Attribute	Required?	Data Type (Length)	Description
ReportName	Y	string	Specify the report type for which you want to retrieve column names. Acceptable values: • ReportName.Positions • ReportName.RiskDecomposition • ReportName.FactorExposureBreakdown • ReportName.StressTestPositions • ReportName.AllocationSelection • ReportName.StressTestSummary
Owner	Y	string	Enter the User ID of the column owner. Enter "SYSTEM" unless it is a user-created column.

To create a GetReportColumnName object, send the following:

```
List<String> myColumns =
  myPort.getReportColumnName(ReportName.Positions,"myOwner");
```

In .NET, send the following:

```
string[] myColumns =
  bdtiClient.GetReportColumnName(ReportName.Positions,"myOwner");
```

Column

Column is the sole element of the String array returned in response to the GetReportColumnName element.

Table 12: Column (1-n)

Attribute	Required?	Data Type (Length)	Description
Column	Y	string	A column name available in the report.

GetGroupings

The GetGroupings element retrieves the grouping schemes defined in your BarraOne account for each of the column names submitted. It has one child element, Column, as indicated below. It will return a Grouping array. The results of this operation can then be used in the Report Settings for "GetPositionsReport" on page 48, "SubmitPositionsReport" on page 56, "SubmitRiskDecompReport" on page 60, "SubmitStressTestPortfolioReport" on page 69, and "SubmitAllocationSelectionReport" on page 74.

Column

Column is the sole child element of the GetGroupings element.

Table 13: Column (1-n)

Attribute	Required?	Data Type (Length)	Description
Name	Y	string	Enter the name of the column (attribute) to which the grouping applies. Column names can be retrieved using "GetReportColumnName" on page 14.
Owner	Y	string	Enter the User ID of the column owner. Enter "SYSTEM" unless it is a user-created column.

To create a GetGroupings object, send the following:

```
List<Column> myCol = new ArrayList<Column>();
Column c1 = new Column();
c1.setName("Price");
c1.setOwner("System");
myCol.add(c1);
List<Grouping> myGroups = myPort.getGroupings(myCol);
```

In .NET, send the following:

```
Column gc1 = new Column();
gc1.Name ="Price";
gc1.Owner ="System";

Column gc2 = new Column();
gc2.Name ="Main Industry";
gc2.Owner ="System";

Grouping[] groupings = bdtic.GetGroupings(new Column[] { gc1, gc2 });
```

Grouping

Grouping is the sole element of the Grouping array returned in response to the GetGroupings element.

Table 14: Grouping (0-n)

Attribute	Required?	Data Type (Length)	Description
Name	Y	string	The name of the grouping (*i.e.,* an available column or attribute on which the Positions Report may be grouped) for the Positions Report.
Owner	Y	string	The User ID of the grouping owner ("SYSTEM" unless it is a user-created column).
Bucket	Y	string	The name of the grouping scheme for the Positions Report (*e.g.,* "distinct").

GetSystemDates

The GetSystemDates element enables users to get current system dates in BarraOne. The signature of this element has no arguments. It will return a SystemDates array.

To create a GetSystemDates object, send the following:

```
SystemDates myDates = myPort.getSystemDates();
```

In .NET, send the following:

```
SystemDates myDates = bdtiClient.GetSystemDates();
```

SystemDates

SystemDates is the sole element of the SystemDates array returned in response to the GetSystemDates element.

Table 15: SystemDates (1)

Attribute	Required?	Data Type (Length)	Description
LatestFinalizedMonthEndDate	Y	date	The latest finalized month-end date.
LatestFinalizedDate	Y	date	The latest finalized date.
LatestAnalysisDate	Y	date	The latest analysis date.

GetAnalysisDate

The GetAnalysisDate element enables users to get the current analysis date for their active BDT Interactive session. The signature of this element has no arguments.

To create a GetAnalysisDate object, send the following:

```
XMLGregorianCalendar myAnalysisDate = myPort.getAnalysisDate();
```

In .NET, send the following:

```
DateTime myAnalysisDate = bdtiClient.GetAnalysisDate();
```

AnalysisDate

AnalysisDate is the sole element returned in response to the GetAnalysisDate element.

Table 16: AnalysisDate (1)

Attribute	Required?	Data Type (Length)	Description
AnalysisDate	Y	date	The current analysis date.

SetAnalysisDate

The SetAnalysisDate element is an operation that enables the user to set the analysis date for the active BDT Interactive session. It has one child element, AnalysisDate, as indicated below.

▷ **Note:** If you change the analysis date, then any unsaved changes to a portfolio (refer to "ModifyPortfolio" on page 42 or "SetAdHocPortfolio" on page 39) will be lost. As such, to preserve any portfolio changes, be certain to use either "SavePortfolio" on page 44 or "SavePortfolioAs" on page 46 before changing the analysis date.

AnalysisDate

AnalysisDate is the sole child element of the SetAnalysisDate element.

Table 17: AnalysisDate (1)

Attribute	Required?	Data Type (Length)	Description
AnalysisDate	Y	date	Enter the analysis date you would like to set.

To create a SetAnalysisDate object, send the following:

```
XMLGregorianCalendar myCalendar =
  DatatypeFactory.newInstance().newXMLGregorianCalendar();
myCalendar.setTimezone(DatatypeConstants.FIELD_UNDEFINED);
myCalendar.setTime(DatatypeConstants.FIELD_UNDEFINED,
  DatatypeConstants.FIELD_UNDEFINED, DatatypeConstants.FIELD_UNDEFINED);
myCalendar.setYear(2007);
myCalendar.setMonth(10);
myCalendar.setDay(9);
myPort.setAnalysisDate(myCalendar);
```

In .NET, send the following:

```
DateTime myDate = new DateTime(2007, 10, 9);
bdtiClient.SetAnalysisDate(myDate);
```

GetFactorTree

The GetFactorTree element takes a Filter list as input and retrieves a NameOwner array of all factor trees associated with the user specified as the factor tree filter. Factor tree filters are the names of the factor tree owners in the BarraOne application. This provides users with a way to locate the name of an existing factor tree they want to use for their Risk Decomposition report. The output can then be used for the function that uses a factor tree NameOwner as an argument (i.e., "KeyValue" on page 60 for "SubmitRiskDecompReport" on page 60).

Filter

Filter is the sole child element of the GetFactorTree element.

Table 18: Filter (0-1)

Attribute	Required?	Data Type (Length)	Description
Filter	N	string	Enter the name of a filter. The user can obtain a list of available filters using "GetFilters" on page 8.

To create a GetFactorTree object, send the following:

```
List<NameOwner> myFactorTree = myPort.getFactorTree("myFilter");
```

In .NET, send the following:

```
NameOwner[] myFactorTree = bdtiClient.GetFactorTree("myFilter");
```

NameOwner

NameOwner is the array returned as a response to the GetFactorTree element.

Table 19: NameOwner (0-n)

Attribute	Required?	Data Type (Length)	Description
Name	Y	string	The name of the factor tree.
Owner	Y	string	The User ID of the owner of the factor tree ("SYSTEM" unless it is a user-created factor tree).

GetFactorTypes

The GetFactorTypes element signature has no arguments, and it retrieves a list of all factor types in BarraOne (e.g., Currency, All Within Market, Industry, Term Structure, Emerging Market, Style, Commodity, Hedge Fund, Spread). This provides users with a way to locate the name of existing factor types they want to use for their Factor Exposure Breakdown Report. It will return a FactorTypes list. The output can then be used for the functions that use FactorTypes as an argument ("GetFactorGroupings" on page 22 and "SubmitFEBReport" on page 64).

To create a GetFactorTypes object, send the following:

```
List<String> myFactorTypes = myPort.getFactorTypes();
```

In .NET, send the following:

```
String[] myFactorTypes = bdtiClient.GetFactorTypes();
```

Table 20: FactorTypes (1-n)

Attribute	Required?	Data Type (Length)	Description
FactorType	Y	string	The text string of the factor type (e.g., "Currency").

GetFactorGroupings

The GetFactorGroupings element takes a list of FactorTypes as input and retrieves a list of all factor groupings associated with the factor types specified. This provides users with a way to locate the name of an existing factor grouping they want to use for their Factor Exposure Breakdown Report. It will return a Grouping list. The output can then be used for "SubmitFEBReport" on page 64.

Table 21: FactorTypes (0-n)

Attribute	Required?	Data Type (Length)	Description
FactorTypes	N	string	Enter the text string of the factor type (e.g., "Currency"). Refer to "GetFactorTypes" on page 21 to retrieve a list.

To create a GetFactorGroupings object, send the following:

```
List<Grouping> myGroupings = myPort.getFactorGroupings("myFactorTypes");
```

In .NET, send the following:

```
Grouping[] myGroupings = bdtiClient.GetFactorGroupings("myFactorTypes");
```

Table 22: Grouping (0-n)

Attribute	Required?	Data Type (Length)	Description
Grouping	N	string	The name of the factor grouping.

GetLocalMarketGrouping

The GetLocalMarketGrouping element signature has no arguments, and it retrieves a list of all local market groupings. This provides users with a way to locate the name of an existing local market grouping they want to use for their Factor Exposure Breakdown Report. It will return a LocalMarketGroupingName list. The output can then be used for the functions that use LocalMarketGroupingName as an argument (i.e., "GetLocalMarket" on page 24, and "KeyValue" on page 64 for "SubmitFEBReport" on page 64).

To create a GetLocalMarketGrouping object, send the following:

```
List<String> myGroupings = myPort.getLocalMarketGrouping();
```

In .NET, send the following:

```
String[] myGroupings = bdtiClient.GetLocalMarketGrouping();
```

Table 23: LocalMarketGroupingName (0-n)

Attribute	Required?	Data Type (Length)	Description
LocalMarketGroupingName	N	string	Text string of the local market grouping.

GetLocalMarket

The GetLocalMarket element takes the LocalMarketGroupingName as input and retrieves a list of all local markets associated with the local market group specified. This provides users with a way to locate the name of an existing local market group they want to use for their Factor Exposure Breakdown Report. It will return a LocalMarketGroup list. The output can then be used for the function that uses LocalMarketGroup as an argument (i.e., "KeyValue" on page 64 for "SubmitFEBReport" on page 64).

Table 24: LocalMarketGroupingName (1)

Attribute	Required?	Data Type (Length)	Description
LocalMarketGroupingName	Y	string	Enter the text string of the name of the local market grouping. Refer to "GetLocalMarketGrouping" on page 23 to retrieve a list.

To create a GetLocalMarket object, send the following:

```
List<String> myGroups = myPort.getLocalMarket("myLocalMarketGroupingName");
```

In .NET, send the following:

```
String[] myGroups = bdtiClient.GetLocalMarket("myLocalMarketGroupingName");
```

Table 25: LocalMarketGroup (0-n)

Attribute	Required?	Data Type (Length)	Description
LocalMarketGroup	N	string	The text string of the local market group.

GetCurrentSettings

The GetCurrentSettings element retrieves the set of current strategy settings for the current portfolio in a Key/Value pair format.

▷ **Note:** In BarraOne, Current Settings are categorized in the following manner:

- Base Strategy (a named strategy defined in BarraOne and assigned to the portfolio)
- Delta Strategy (modifications to the Base Strategy in BarraOne—this is the strategy initially associated with the portfolio)
- Delta Session Strategy (modifications to the strategy made using the SetCurrentSettings or SetCurrentSettingsKeyVal elements during the current BDTI session)

When the GetCurrentSettings element is used and the InclSessionOverride attribute is set to "true," then the element returns the Current Settings, which are the combined result of Base Strategy + Delta Strategy + Delta Session Strategy.

Table 26: GetCurrentSettings (1)

Attribute	Required?	Data Type (Length)	Description
InclSessionOverride	Y	boolean	Indicates if you want the retrieved settings to include changes made to the strategy during the BDTI session using the SetCurrentSettings or SetCurrentSettingsKeyVal elements. Acceptable values: • true (include Delta Session Strategy changes) • false (exclude Delta Session Strategy changes)

To create a GetCurrentSettings object, send the following:

```
List<KeyValue> myKeyValues = myPort.getCurrentSettings(false);
```

In .NET, send the following:

```
KeyValue[] myKeyValues = bdtiClient.GetCurrentSettings(false);
```

KeyValue is the sole element of the string array returned as a response to GetCurrentSettings element.

Table 27: KeyValue (0-n)

Attribute	Required?	Data Type (Length)	Description
Key	Y	string	The name of the strategy setting.
Value	Y	string	The value of the strategy setting.

SetCurrentSettings

The SetCurrentSettings element enables users to set the current portfolio strategy settings for their active BDT Interactive session. (It is equivalent to the element "SetCurrentSettingsKeyVal" on page 31.) It has one child element, CurrentSettings, as indicated below. Refer to the element "GetCurrentSettings" on page 25 to retrieve the current portfolio settings.

▷ **Note:** In BarraOne, Current Settings are categorized in the following manner:

- Base Strategy (a named strategy defined in BarraOne and assigned to the portfolio)
- Delta Strategy (modifications to the Base Strategy in BarraOne—this is the strategy initially associated with the portfolio)
- Delta Session Strategy (modifications to the strategy made using the SetCurrentSettings or SetCurrentSettingsKeyVal elements during the current BDTI session)

When the SetCurrentSettings element is used, you are modifying the Current Settings by changing the Delta Session Strategy. The Base Strategy and Delta Strategy are not changed.

▷ **Note:** Any settings that are changed using this element are volatile and may be restored to their default values by invoking elements, such as SavePortfolio.

CurrentSettings

CurrentSettings is the sole child element of the SetCurrentSettings element.

Table 28: CurrentSettings (1)

Attribute	Required?	Data Type (Length)	Description
Benchmark	N	string	Enter the portfolio (or "CASH") you want to use as the benchmark for this portfolio. **Note:** To specify an aggregate portfolio when there are both aggregate and regular portfolios with the same name, preface the aggregate portfolio name with a slash (/). If specified: · BarraOne validates that the benchmark portfolio is a valid tree or home portfolio for the specified BenchmarkOwner and that the user has access to it. If it does not exist, the SetCurrentSettings call will be rejected. If not specified: · The value defaults to the benchmark in the base strategy.
BenchmarkOwner	N*	string	Enter the User ID of the owner of the benchmark portfolio. If you want to use a BarraOne-supplied portfolio, enter "SYSTEM" as the owner. *Required if a Benchmark is specified.

Table 28: CurrentSettings (1) (Continued)

Attribute	Required?	Data Type (Length)	Description
Market	N	string	Enter the market portfolio (or "CASH") you want to use with this portfolio. **Note:** To specify an aggregate portfolio when there are both aggregate and regular portfolios with the same name, preface the aggregate portfolio name with a slash (/). If specified: • BarraOne validates that the market portfolio is a valid tree or home portfolio for the specified MarketOwner and that the user has access to it. If it does not exist, the SetCurrentSettings call will be rejected. If not specified: • The value defaults to the market in the base strategy.
MarketOwner	N*	string	Enter the User ID of the owner of the market portfolio. If you want to use a BarraOne-supplied portfolio, enter "SYSTEM" as the owner. *Required if a Market is specified.
Universe	N	string	Enter the universe portfolio (or "CASH") you want to use with this portfolio. **Note:** To specify an aggregate portfolio when there are both aggregate and regular portfolios with the same name, preface the aggregate portfolio name with a slash (/). If specified: • BarraOne validates that the universe portfolio is a valid tree or home portfolio for the specified UniverseOwner and that the user has access to it. If it does not exist, the SetCurrentSettings call will be rejected. If not specified: • The value defaults to the market in the base strategy.
UniverseOwner	N*	string	Enter the User ID of the owner of the universe portfolio. If you want to use a BarraOne-supplied portfolio, enter "SYSTEM" as the owner. *Required if a Universe is specified.
BaseCurrency	N	string (3)	Enter the currency perspective you want to use with this portfolio. If not specified, the value defaults to the currency in the base strategy. For a list of currency codes, refer to "Appendix A — Country and Currency Codes" on page 85.

Table 28: CurrentSettings (1) (Continued)

Attribute	Required?	Data Type (Length)	Description
BaseValueType	N	enum	Enter the base value type you want to use with this portfolio. Acceptable values: • BaseValueType.ASSIGNED • BaseValueType.LONG • BaseValueType.LONG_SHORT • BaseValueType.NET If specified: • The value overrides the value in the base strategy. • If set to BaseValueType.ASSIGNED, then a valid AssignedBaseValue needs to be provided. If not specified, this value defaults to the BaseValueType in the base strategy.
AssignedBaseValue	N*	double	Enter the assigned base value you want to use with the portfolio, specified in terms of the base currency. *Required if BaseValueType is set to ASSIGNED.
IsBalanceCurrency	N	boolean	If BaseValueType is "ASSIGNED," indicate if the implicit cash position (above the value of the actual holdings in the portfolio) should be included as an explicit asset in the portfolio. Acceptable values: • true • false
ParamVarHorizon	N	int	Enter the number of days for the parametric value-at-risk horizon.
ParamVarConfidenceLevel	N	double	Enter the confidence level for parametric value-at-risk in percentage terms (e.g., 95 for 95%).
SimulationVarLimitCurUnit	N	double	Enter a limit in units of the base currency that enables you to monitor the portfolio's simulation VaR (mutually exclusive of SimulationVaRLimitMultiplier).
SimulationVarLimitMultiplier	N	double	Enter a limit as a multiplier of Benchmark VaR (e.g., 2 for 2 X Benchmark VaR) that enables you to monitor the portfolio's simulation VaR (mutually exclusive of SimulationVaRLimitCurUnit).
LossTarget	N	double	Enter the loss target (e.g., 5 for 5%) for which BarraOne will calculate Below Target Probability. This is the percent chance that the portfolio will lose a specified percent of its value over a specified horizon.
CurrencyHedgingBenchmark	N	boolean	Indicate if you want to use currency hedging for the benchmark. Acceptable values: • true • false

Table 28: CurrentSettings (1) (Continued)

Attribute	Required?	Data Type (Length)	Description
CurrencyHedgingBenchmark IsNumeraire	N	boolean	If CurrencyHedgingBenchmark is true, indicate if the hedging currency is the numeraire currency (mutually exclusive of CurrencyHedgingBenchmarkCurrency). Acceptable values: · true · false
CurrencyHedgingBenchmark Currency	N	string	If CurrencyHedgingBenchmark is true, enter the hedging currency (mutually exclusive of CurrencyHedgingBenchmarkIsNumeraire).
CurrencyHedgingBenchmark Percentage	N	double	Enter the percentage of currency hedging for the benchmark (e.g., 10 for 10%).
CurrencyHedgingPortfolio	N	boolean	Indicate if you want to use currency hedging for the portfolio. Acceptable values: · true · false
CurrencyHedgingPortfolio IsNumeraire	N	boolean	If CurrencyHedgingPortfolio is true, indicate if the hedging currency is the numeraire currency (mutually exclusive of CurrencyHedgingPortfolioCurrency). Acceptable values: · true · false
CurrencyHedgingPortfolio Currency	N	string	If CurrencyHedgingPortfolio is true, enter the hedging currency (mutually exclusive of CurrencyHedgingPortfolioIsNumeraire).
CurrencyHedgingPortfolio Percentage	N	double	Enter the percentage of currency hedging for the portfolio (e.g., 10 for 10%).
NetLeverageHedgingCutOff	N	double	For each portfolio that employs netting or hedging, the user specifies, as a strategy setting, the Net Leverage Hedging Cut-Off value as a percentage. This represents the minimum correlation between the derivative instrument and its underlying asset necessary for Net Leverage to be calculated for the portfolio.

To create a SetCurrentSettings object, send the following:

```
CurrentSettings mySettings = new CurrentSettings();
mySettings.setBenchmark("ssap500d");
mySettings.setBenchmarkOwner("system");
mySettings.setBaseCurrency("jpy");
mySettings.setMarket("mmim");
mySettings.setMarketOwner("system");
mySettings.setUniverse("ife");
mySettings.setUniverseOwner("system");
mySettings.setBaseValueType(BaseValueType.ASSIGNED);
mySettings.setAssignedBaseValue(1000000000);
port.setCurrentSettings(mySettings);
```

In .NET, send the following:

```
CurrentSettings mySettings = new CurrentSettings();
mySettings.Benchmark = "ssap500d";
mySettings.BenchmarkOwner = "SYSTEM";
mySettings.BaseValueType = BaseValueType.Long_Short;
mySettings.BaseCurrency = "USD";
mySettings.BaseValueTypeSpecified = true;
bdtiClient.SetCurrentSettings(mySettings);
```

GetCurrentSettingsKeys

The GetCurrentSettingsKeys element signature has no arguments, and it enables users to get a list of available strategy setting names in BarraOne. It will return a String array. The response can then be used in the element "SetCurrentSettingsKeyVal" on page 31.

To create a GetCurrentSettingsKeys object, send the following:

```
List<String> myKeys = myPort.getCurrentSettingsKeys();
```

In .NET, send the following:

```
string[] myKeys = bdtiClient.GetCurrentSettingsKeys();
```

Keys

Keys is the sole element of the String array returned by the GetCurrentSettingsKeys element.

Table 29: Keys (1-n)

Attribute	Required?	Data Type (Length)	Description
Keys	Y	string	The name of the portfolio strategy setting.

SetCurrentSettingsKeyVal

The SetCurrentSettingsKeyVal element, similar to the SetCurrentSettings element, enables users to set the current portfolio strategy settings for their active BDT Interactive session. It differs from SetCurrentSettings in that the input is an array of Keys and Values. It has one child element, CurrentSettings, as indicated below. Refer to "GetCurrentSettingsKeys" on page 30 to retrieve the available strategy setting keys. Refer to "GetCurrentSettings" on page 25 to retrieve an array of Keys and Values.

To create a SetCurrentSettingsKeyVal object, send the following:

```
KeyValue kvBch = new KeyValue();
kvBch.setKey("Benchmark");
kvBch.setValue("SSAP500d");

KeyValue kvBchOwner = new KeyValue();
kvBchOwner.setKey("BenchmarkOwner");
kvBchOwner.setValue("system");

KeyValue kvMarket = new KeyValue();
kvMarket.setKey("Market");
kvMarket.setValue("xiim");

KeyValue kvMarketOwner = new KeyValue();
kvMarketOwner.setKey("MarketOwner");
kvMarketOwner.setValue("system");

KeyValue baseValType = new KeyValue();
baseValType.setKey("CurrencyHedgingBenchmarkPercentage");
baseValType.setValue("1");

KeyValue assignedVal = new KeyValue();
assignedVal.setKey("ParamVarHorizon");
assignedVal.setValue("772.00");

KeyValue ParamVarConfidenceLevel = new KeyValue();
ParamVarConfidenceLevel.setKey("ParamVarConfidenceLevel");
ParamVarConfidenceLevel.Value = "59.3";

KeyValue balCash = new KeyValue();
balCash.setKey("IsBalanceCurrency");
balCash.setValue("true");

KeyValue CurrencyHedgingBenchmark = new KeyValue();
CurrencyHedgingBenchmark.setKey("CurrencyHedgingBenchmark");
CurrencyHedgingBenchmark.setValue("true");
```

```
KeyValue CurrencyHedgingBenchmarkIsNumeraire = new KeyValue();
CurrencyHedgingBenchmarkIsNumeraire.setKey
  ("CurrencyHedgingBenchmarkIsNumeraire");
CurrencyHedgingBenchmarkIsNumeraire.setValue("true");


KeyValue CurrencyHedgingBenchmarkPercentage = new KeyValue();
CurrencyHedgingBenchmarkPercentage.setKey
  ("CurrencyHedgingBenchmarkPercentage";
CurrencyHedgingBenchmarkPercentage.setValue("27.5");


KeyValue SimulationVarLimitCurUnit = new KeyValue();
SimulationVarLimitCurUnit.setKey("SimulationVarLimitCurUnit");
SimulationVarLimitCurUnit.setValue("27");


List<KeyValue> myKeyValues = new ArrayList<KeyValue>();


myKeyValues.add(kvBch);
myKeyValues.add(kvBchOwner);
myKeyValues.add(kvMarket);
myKeyValues.add(kvMarketOwner);
myKeyValues.add(baseValType);
myKeyValues.add(assignedVal);
myKeyValues.add(ParamVarConfidenceLevel);
myKeyValues.add(balCash);
myKeyValues.add(CurrencyHedgingBenchmark);
myKeyValues.add(CurrencyHedgingBenchmarkIsNumeraire);
myKeyValues.add(CurrencyHedgingBenchmarkPercentage);
myKeyValues.add(SimulationVarLimitCurUnit);


port.setCurrentSettingsKeyVal(myKeyValues);
```

In .NET:

```
KeyValue kvBch = new KeyValue();
kvBch.Key = "Benchmark";
kvBch.Value = "SSAP500d";


KeyValue kvBchOwner = new KeyValue();
kvBchOwner.Key = "BenchmarkOwner";
kvBchOwner.Value = "system";


KeyValue kvMarket = new KeyValue();
kvMarket.Key = "Market";
kvMarket.Value = "xiim";
```

```
KeyValue kvMarketOwner = new KeyValue();
kvMarketOwner.Key = "MarketOwner";
kvMarketOwner.Value = "system";

KeyValue baseValType = new KeyValue();
baseValType.Key = "CurrencyHedgingBenchmarkPercentage";
baseValType.Value = "1";

KeyValue assignedVal = new KeyValue();
assignedVal.Key = "ParamVarHorizon";
assignedVal.Value = "772.00";

KeyValue ParamVarConfidenceLevel = new KeyValue();
ParamVarConfidenceLevel.Key = "ParamVarConfidenceLevel";
ParamVarConfidenceLevel.Value = "59.3";

KeyValue balCash = new KeyValue();
balCash.Key = "IsBalanceCurrency";
balCash.Value = "true";

KeyValue CurrencyHedgingBenchmark = new KeyValue();
CurrencyHedgingBenchmark.Key = "CurrencyHedgingBenchmark";
CurrencyHedgingBenchmark.Value = "true";

KeyValue CurrencyHedgingBenchmarkIsNumeraire = new KeyValue();
CurrencyHedgingBenchmarkIsNumeraire.Key =
  "CurrencyHedgingBenchmarkIsNumeraire";
CurrencyHedgingBenchmarkIsNumeraire.Value = "true";

KeyValue CurrencyHedgingBenchmarkPercentage = new KeyValue();
CurrencyHedgingBenchmarkPercentage.Key =
  "CurrencyHedgingBenchmarkPercentage";
CurrencyHedgingBenchmarkPercentage.Value = "27.5";

KeyValue SimulationVarLimitCurUnit = new KeyValue();
SimulationVarLimitCurUnit.Key = "SimulationVarLimitCurUnit";
SimulationVarLimitCurUnit.Value = "27";
```

```
bdtic.SetCurrentSettingsKeyVal(new KeyValue[] { kvBch, kvBchOwner,
  kvMarket,kvMarketOwner,baseValType,assignedVal,
  ParamVarConfidenceLevel,SimulationVarLimitCurUnit,balCash,
  CurrencyHedgingBenchmark,CurrencyHedgingBenchmarkIsNumeraire,
  CurrencyHedgingBenchmarkPercentage  });
```

Table 30: KeyValue (0-n)

Attribute	Required?	Data Type (Length)	Description
Key	Y	string	The name of the strategy setting.
Value	Y	string	The value of the strategy setting.

▷ **Note:** In BarraOne, Current Settings are categorized in the following manner:

- Base Strategy (a named strategy defined in BarraOne and assigned to the portfolio)
- Delta Strategy (modifications to the Base Strategy in BarraOne—this is the strategy initially associated with the portfolio)
- Delta Session Strategy (modifications to the strategy made using the SetCurrentSettings or SetCurrentSettingsKeyVal elements during the current BDTI session)

When the SetCurrentSettingsKeyVal element is used, you are modifying the Current Settings by changing the Delta Session Strategy. The Base Strategy and Delta Strategy are not changed.

GetReportSettings

The GetReportSettings element takes a ReportName as input and retrieves available keys and default values for report settings. It will return a String array. The response can be used in the element "GetPositionsReport" on page 48, "SubmitPositionsReport" on page 56, "SubmitRiskDecompReport" on page 60, "SubmitFEBReport" on page 64, "SubmitStressTestPortfolioReport" on page 69, and "SubmitAllocationSelectionReport" on page 74.

ReportName

Table 31: ReportName (0-1)

Attribute	Required?	Data Type (Length)	Description
ReportName	N	enum	Enter the name of the report for which to get the report settings. If not specified, the application uses "Positions" as the default. Acceptable values: • ReportName.Positions • ReportName.RiskDecomposition • ReportName.FactorExposureBreakdown • ReportName.StressTestPositions • ReportName.AllocationSelection

To create a GetReportSettings object, send the following:

```
List<KeyValue> myKeyValues =
  myPort.getReportSettings(ReportName.POSITIONS);
```

In .NET, send the following:

```
KeyValue[] myKeyValues =
  bdtiClient.GetReportSettings(ReportName.Positions);
```

KeyValue is the sole element of the string array returned as a response to GetReportSettings element.

Table 32: KeyValue (0-n)

Attribute	Required?	Data Type (Length)	Description
Key	Y	string	The name of the report setting.
Value	Y	string	The default value of the report setting.

Refer to "Report Settings for Positions Report" on page 59, "Report Settings for Risk Decomposition Report" on page 63, "Report Settings for Factor Exposure Breakdown Report" on page 66, "Report Settings for Stress Test Portfolio Report" on page 73, and "Report Settings for Allocation-Selection Report" on page 77 for descriptions of the settings for the respective reports.

GetModels

The GetModels element enables users to get a list of available models in BarraOne. Filter is the sole child element of the GetModels element. It will return a Model array that can be used in the element "SetModel" on page 37.

Table 33: Filter (0-1)

Attribute	Required?	Data Type (Length)	Description
Filter	N	string	Enter the name of a filter. A filter is the equivalent of an item in the dropdown list (such as owner or workgroup) available in the Select Model dialog in the BarraOne application. The user can obtain a list of available filters using "GetFilters" on page 8.

To create a GetModels object, send the following:

```
List<Model> myModels = myPort.getModels("myFilter");
```

In .NET, send the following:

```
Model[] myModels = bdtiClient.GetModels("myFilter");
```

Model

Model is the sole element returned as a response to the GetModels element.

Table 34: Model (0-n)

Attribute	Required?	Data Type (Length)	Description
Name	Y	string	The name of the model.
Owner	Y	string	The User ID of the owner of the model ("SYSTEM" unless it is a user-created model).
CovarianceDate	N	date	For System models only, indicates the model covariance date if different from the analysis date.
CovarianceDateSupport	N	boolean	Indicates if the model supports a covariance date different from the analysis date. Acceptable values: • true (covariance date supported) • false (covariance date not supported)

SetModel

The SetModel element will set the current model. Model is the sole element of the SetModel element.

Table 35: Model (0-1)

Attribute	Required?	Data Type (Length)	Description
Name	Y	string	Enter the name of the model. The user can obtain a list of available models using "GetModels" on page 36.
Owner	Y	string	Enter the User ID of the owner of the model ("SYSTEM" unless it is a user-created folder). The user can obtain a list of available filters using "GetFilters" on page 8.
CovarianceDate	N	date	For System models only, indicates the model covariance date if different from the analysis date.
CovarianceDateSupport	N	boolean	Indicates if the model supports a covariance date different from the analysis date. Acceptable values: • true (covariance date supported) • false (covariance date not supported)

To create a SetModel object, send the following:

```
myPort.setModel("myModelName","myModelOwner");
```

In .NET, send the following:

```
bdtiClient.SetModel("myModelName","myModelOwner");
```

GetCurrentSessionModel

The GetCurrentSessionModel element enables users to get the current Model being used for their active BDT Interactive session. The signature of this element has no arguments.

To create a GetCurrentSessionModel object, send the following:

```
Model myModel = myPort.getCurrentSessionModel();
```

In .NET, send the following:

```
Model myModel = bdtiClient.GetCurrentSessionModel();
```

Model

Model is the sole element returned as a response to the GetCurrentSessionModel element.

Table 36: Model (1)

Attribute	Required?	Data Type (Length)	Description
Name	Y	string	The name of the model.
Owner	Y	string	The User ID of the owner of the model ("SYSTEM" unless it is a user-created model).
CovarianceDate	N	date	For System models only, indicates the model covariance date if different from the analysis date.
CovarianceDateSupport	N	boolean	Indicates if the model supports a covariance date different from the analysis date. Acceptable values: • true (covariance date supported) • false (covariance date not supported)

SetAdHocPortfolio

The SetAdHocPortfolio element enables users to define a portfolio for use in BDTI. It has one child element, Portfolio, as indicated below. It will return a RejectedAsset array.

▷ **Note:** If you change the analysis date (refer to "SetAnalysisDate" on page 19), or if your session times out, then any unsaved ad hoc portfolio will be lost. As such, to preserve your ad hoc portfolio, be certain to use "SavePortfolioAs" on page 46 before changing the analysis date or before your session times out.

Portfolio

Portfolio is the sole child element of the SetAdHocPortfolio element, and it enables the user to specify basic attributes of the ad hoc portfolio. It has one child element, Position, as indicated below.

Table 37: Portfolio (1)

Attribute	Required?	Data Type (Length)	Description
Type	Y	enum	Enter the type of portfolio import. Acceptable values: • PortfolioType.HOLDINGS • PortfolioType.WEIGHT • PortfolioType.VALUE
PortfolioValue	N*	double	Enter the value of the portfolio. *BarraOne requires this value when Type is set to WEIGHT.
PortfolioCurrency	N*	string	Enter the currency of the portfolio. *BarraOne requires this value when Type is set to WEIGHT. For a list of currency codes, refer to "Appendix A — Country and Currency Codes" on page 85.

Position

Position is the sole child element of the Portfolio element. It enables the user to specify the position value and currency in the ad hoc portfolio. It has one child element, MID, as indicated below.

Table 38: Position (1-n)

Attribute	Required?	Data Type (Length)	Description
Value	Y	double	Enter the value (holdings, weight, or value) of the position in the portfolio.
Currency	N	string	Enter the currency of the position in the portfolio. For a list of currency codes, refer to "Appendix A — Country and Currency Codes" on page 85.

MID

MID is the sole child element of the Position element, and it enables the user to specify the identifier for the position in the ad hoc portfolio.

Table 39: Position.MID (1-n)

Attribute	Required?	Data Type (Length)	Description
ID	Y	string	Enter an ID for the position in your portfolio, including leading zeros. You can specify multiple IDs and let BarraOne search. BarraOne automatically adjusts CUSIPs and SEDOLs, whether you enter them with or without a check digit. Thus, BarraOne automatically supports 6- or 7-digit SEDOLs and 8- or 9-digit CUSIPs.
IDType	N	string	Enter the ID Type for the position ID. If you omit the ID type or leave it blank, BarraOne will search across all ID types in its database according to your ID priority. (To view your ID priority, go to the Asset ID attribute under the Data Admin tab of BarraOne.) For a list of supported ID types, refer to "Appendix B — Asset ID Types" on page 95.

To create a SetAdHocPortfolio object, send the following:

```
Portfolio myPortfolio = new Portfolio();
myPortfolio.setType(PortfolioType.HOLDINGS);
myPortfolio.Position myP1 = new Position();
myP1.setValue(777);
myP1.setCurrency("USD");
myP1.MID myMid = new MID();
myMid.setID("USAD281");
myMid.setIDType("BARRAID");
List<MID> myMidArray = new ArrayList<MID>();
myMidArray.add(myMid);
myMid = myMidArray;
myPortfolio.Position myP2 = new Position();
myP2.setCurrency("USD");
myP2.setValue(53);
myP2.MID myMid2 = new MID();
myMid2.setID("USAISP1");
myMid2.setIDType("BARRAID");
List<MID> myMidArray2 = new ArrayList<MID>();
myMidArray2.add(myMid2);
myMid = myMidArray2;
List<Position> myPositions = new ArrayList<Position>();
myPositions.add(myP1);
```

```
myPositions.add(myP2);
myPortfolio.pos = myPositions;
List<RejectedAsset> myRejectedAssets = port.setAdHocPortfolio(myPortfolio);
if(myRejectedAssets.size()>0)
{
for (RejectedAsset myAsset : myRejectedAssets){
System.out.println(myAsset.getID());
}
}
```

In .NET, send the following:

```
Portfolio myPortfolio = new Portfolio();
myPortfolio.Type = PortfolioType.HOLDINGS;
Position[] myPositions = new Position[1];
myPositions myPos = new Position();
MID[] myMids = new MID[1];
MID myMid = new MID();
myMid.ID = "id";
myMid.IDType = "idtype";
myMids[0] = myMid;
myPos.Value = 123;
myPos.MID = myMids;
myPos.Currency = "USD";
myPositions[0] = myPos;
bdtiClient.SetAdHocPortfolio(myPortfolio);
```

RejectedAsset

RejectedAsset is the sole element of the RejectedAsset array returned as a response to the SetAdHocPortfolio element.

Table 40: RejectedAsset (0-n)

Attribute	Required?	Data Type (Length)	Description
Reason	Y	string	The reason for the rejection.
ID	Y	string	The asset ID of the rejected asset.
IDType	Y	string	The ID type.

ModifyPortfolio

The ModifyPortfolio element enables users to add, modify, or delete positions in the current portfolio in BarraOne. This operation is dependent upon first setting a current portfolio (refer to "SetCurrentPortfolio" on page 11) or setting an ad hoc portfolio (refer to "SetAdHocPortfolio" on page 39). It has two child elements, Holding and ModType, as indicated below. It will return a RejectedAsset array.

▷ **Note:** If you change the analysis date (refer to "SetAnalysisDate" on page 19), or if your session times out, then any unsaved changes to a portfolio will be lost. As such, to preserve any portfolio changes, be certain to use either "SavePortfolio" on page 44 or "SavePortfolioAs" on page 46 before changing the analysis date or before your session times out.

Holding

Holding is a child element of the ModifyPortfolio element.

Table 41: Holding (1-n)

Attribute	Required?	Data Type (Length)	Description
ID	Y	string	Enter the asset ID of the asset to be modified.
IDType	Y	string	Enter the ID Type. To instruct BarraOne to search based in its priority settings, enter "USE PRIORITY." For a list of supported ID types, refer to "Appendix B — Asset ID Types" on page 95.
Value	N	double	Enter the value of the modification. (Note that Share or Weight may be zero.)
ModValueType	N	enum	Enter the type of modification value. Used for a ModType of Modify only. Acceptable values: • ModValueType.SHARE (indicates that the Value specifies number of shares in the portfolio) • ModValueType.WEIGHT (indicates that the Value specifies the weight of the holding in the portfolio) — the cash position in the portfolio will be adjusted to reflect percentages that do not sum to 100

ModType

ModType is a child element of the ModifyPortfolio element.

Table 42: ModType (1)

Attribute	Required?	Data Type (Length)	Description
ModType	Y	enum	Enter the type of modification. Acceptable values: • ModType.ADD (adds the specified asset holding to the portfolio, *i.e.,* a delta change) • ModType.DELETE (deletes the specified asset holding from the portfolio) • ModType.MODIFY (modifies the asset holding to the specified value, *i.e.,* an absolute change)

To create a ModifyPortfolio object, send the following:

```
List<Holding> myHolding = new ArrayList<Holding>();

Holding myH1 = new Holding();
myH1.setID("USAISP1");
myH1.setIDType("barraid");
myH1.setModValueType(ModValueType.WEIGHT);
myH1.setValue(.07);

Holding myH2 = new Holding();
myH2.setID("594918104");
myH2.setIDType("USE PRIORITY");
myH2.setValue((double)321);
myH2.setModValueType(ModValueType.SHARE);

myHolding.add(myH1);
myHolding.add(myH2);

List<RejectedAsset> myList = myPort.modifyPortfolio(myHolding,
  ModType.MODIFY);
if(myList.size()>0)
{
for (RejectedAsset myAsset : myList){
System.out.println(myAsset.getID()+myAsset.getReason());
}
}
```

In .NET, send the following:

```
Holding[] myHolding = new Holding[1];
myHolding[0] = new Holding();
myHolding[0].ID = "usAD281";
myHolding[0].IDType = "barraid";
myHolding[0].Value = 443;
myHolding[0].ValueSpecified = true;
myHolding[0].ModValueType = ModValueType.SHARE;
myHolding[0].ModValueTypeSpecified = true;
bdtiClient.ModifyPortfolio (myHolding, ModType.MODIFY);
```

RejectedAsset

RejectedAsset is the sole element of the RejectedAsset array returned as a response to the ModifyPortfolio element.

Table 43: RejectedAsset (0-n)

Attribute	Required?	Data Type (Length)	Description
Reason	Y	string	The reason for the rejection.
ID	Y	string	The asset ID of the rejected asset.
IDType	Y	string	The ID type.

SavePortfolio

The SavePortfolio element enables users to save the current portfolio in BarraOne. It is dependent upon first setting the current portfolio (refer to "SetCurrentPortfolio" on page 11). The element signature has no arguments, and it will return a SaveSummary array.

To create a SavePortfolio object, send the following:

```
SaveSummary mySaveSummary = myPort.savePortfolio();
String myPortfolioName = sm.getPortfolioName();
```

In .NET, send the following:

```
bdtiClient.SavePortfolio();
```

SaveSummary

SaveSummary is the sole element of the array returned as a response to the SavePortfolio element.

Table 44: SaveSummary (1)

Attribute	Required?	Data Type (Length)	Description
PortfolioOwner	Y	string	The User ID of the owner of the saved portfolio.
PortfolioName	Y	string	The name of the saved portfolio.
AnalysisDate	Y	string	The analysis date of the saved portfolio.
RecordsWritten	Y	int	The number of records written during the save operation.

SavePortfolioAs

The SavePortfolioAs element enables users to save the current portfolio in BarraOne. It is dependent upon first setting the current portfolio (refer to "SetCurrentPortfolio" on page 11) or setting an ad hoc portfolio (refer to"SetAdHocPortfolio" on page 39.) It has two child elements, PortfolioName and SaveOptions, as indicated below. It will return a SaveSummary array, and the saved portfolio will be set as the current portfolio.

PortfolioName

PortfolioName is a child element of the SavePortfolioAs element.

Table 45: PortfolioName (1)

Attribute	Required?	Data Type (Length)	Description
PortfolioName	Y	string	Enter the name for the saved portfolio. The full list of prohibited characters in portfolio names is as follows: **{ } [ ] ~ ` ‘ # + = : ; < > " / \ | ? * TAB**. The following non-alphanumeric characters are allowed: **. , - _ $ % @ ^ ( ) & SPACE**.

SaveOptions

SaveOptions is a child element of the SavePortfolioAs element.

Table 46: SaveOptions (0-1)

Attribute	Required?	Data Type (Length)	Description
UnpackEIFs	Y	boolean	Specify whether equity index futures should be unpacked into their constituents. Acceptable values: • true (unpack) • false (do not unpack)
UnpackCompositesAndETFs	Y	boolean	Specify whether composites and exchange-traded funds should be unpacked into their respective constituents. Acceptable values: • true (unpack) • false (do not unpack)

Table 46: SaveOptions (0-1) (Continued)

Attribute	Required?	Data Type (Length)	Description
StorageType	Y	enum	Determines the methodology used by BarraOne to store portfolio holdings. Enter the type of portfolio storage. Acceptable values: • StorageType.HOLDINGS (stores portfolio holdings by shares) • StorageType.VALUE (stores portfolio holdings using market value — applicable only if the portfolio is imported either by value or by weight) **Note:** A portfolio cannot alternate between storage by shares and storage by market value. Once the first save of the portfolio, the portfolio must persist using the current holdings storage methodology.

To create a SavePortfolioAs object, send the following:

```
SaveOptions mySaveOptions = new SaveOptions();
mySaveOptions.setStorageType(StorageType.HOLDINGS);
mySaveOptions.setUnpackCompositesAndETFs(false);
mySaveOptions.setUnpackEIFs(false);
SaveSummary mySaveSummary = myPort.savePortfolioAs("name", mySaveOptions);
String myPortfolioName = mySaveSummary.getPortfolioName();
```

In .NET, send the following:

```
SaveOptions mySaveOptions = new SaveOptions();
mySaveOptions.StorageType = StorageType.HOLDINGS;
mySaveOptions.UnpackCompositesAndETFs = false;
mySaveOptions.UnpackEIFs = false;
bdtiClient.SavePortfolioAs("name", mySaveOptions);
```

SaveSummary

The SaveSummary element is the sole element of the array returned as a response to the SavePortfolioAs element.

Table 47: SaveSummary (1)

Attribute	Required?	Data Type (Length)	Description
PortfolioOwner	Y	string	The User ID of the owner of the saved portfolio.
PortfolioName	Y	string	The name of the saved portfolio.
AnalysisDate	Y	string	The analysis date of the saved portfolio.
RecordsWritten	Y	int	The number of records written during the save operation.

GetPositionsReport

The GetPositionsReport element enables users to specify the attributes of a Positions Report in BDTI. It has four child elements, Column, Grouping, CustomizeReportSetting, and SortCriteria, as indicated below. It will return a "ReportRoot" on page 53.

▷ **Note:** This is a synchronous method that works well with most interactive report generation workflows. However, if you are working with a large Positions Report, it is recommended that you use the SubmitPositionsReport and RetrieveReport method, instead, to prevent timeout errors. Refer to "SubmitPositionsReport" on page 56 and "RetrieveReport" on page 82.

Column

Column is a child element of the GetPositionsReport element.

Table 48: Column (1-n)

Attribute	Required?	Data Type (Length)	Description
Name	Y	string	Enter the name of a column to include in the report. A list of available column names can be retrieved using "GetReportColumnName" on page 14.
Owner	Y	string	Enter the User ID of the column owner. Enter "SYSTEM" unless it is a user-created column.

CustomizeReportSetting

CustomizeReportSetting is a child element of the GetPositionsReport element.

Table 49: CustomizeReportSetting (0-1)

Attribute	Required?	Data Type (Length)	Description
InclBenchmarkNotHeld	N	boolean	Specify whether the Positions Report should include benchmark assets not held. Acceptable values: • true • false (default)
LookthruEIF	N	boolean	Specify whether the Positions Report should "look through" equity index futures. Acceptable values: • true • false (default)

Table 49: CustomizeReportSetting (0-1)

Attribute	Required?	Data Type (Length)	Description
LookthruComposites	N	boolean	Specify whether the Positions Report should "look through" composites. Acceptable values: • true • false (default)
InheritCTDBondFuture	N	boolean	Specify whether the Positions Report should inherit cheapest-to-deliver values fore reporting on bond futures. Acceptable values: • true • false (default)
InheritCTDBondFutureOption	N	boolean	Specify whether the Positions Report should inherit cheapest-to-deliver values for reporting on bond future options. Acceptable values: • true • false (default)
IsGlobalGroupingMode	N	boolean	Specify if the grouping scheme should be global (or local). Acceptable values: • true (default) • false
DataRows	N	boolean	"True" specifies that all rows are returned. "False" specifies that only the total and group row(s) are returned from the query if the user has specified grouping. If the user has no grouping specified, then only the total rows are returned. **Note:** If this setting is "False," the user will be unable to save any resulting modified portfolio using "SavePortfolio" on page 44 or "SavePortfolioAs" on page 46. Acceptable values: • true (default) • false
ResidualReturnContribution	N	boolean	"True" specifies that columns are computed based on residual returns methodology, which is also known as Marginal Contribution to Residual Risk Reporting (MCRR) or x-sigma rho analysis. Residual Returns-based analytics facilitate the attribution of risk from groups of factors to single assets and groups of assets. This helps to establish consistency in the analytics between the Positions Report, the Risk Decomposition Report, Factor Exposure Breakdown Report, and the Allocation-Selection Report when using the residual returns methodology. Acceptable values: • true • false (default)

Table 49: CustomizeReportSetting (0-1)

Attribute	Required?	Data Type (Length)	Description
LookthruComposites WithoutETF	N	boolean	Specify whether the Positions Report should “look through” composites without exchange-traded funds. Acceptable values: • true • false (default)
LookthruETF	N	boolean	Specify whether the Positions Report should “look through” exchange-traded funds. Acceptable values: • true • false (default)
LookthruFX	N	boolean	Specify whether the Positions Report should “look through” FX derivatives. Acceptable values: • true • false (default)
LookthruEquityOption	N	boolean	Specify whether the Positions Report should “look through” equity options. Acceptable values: • true • false (default)

Grouping

Grouping is a child element of the GetPositionsReport element that enables two levels of grouping.

Table 50: Grouping (0-2)

Attribute	Required?	Data Type (Length)	Description
Name	Y	string	Enter the name of the grouping (*i.e.,* an available column or attribute on which the Positions Report may be grouped) for the Positions Report. A list of available groupings can be retrieved using "GetGroupings" on page 15.
Owner	Y	string	Enter the User ID of the grouping owner. Enter "SYSTEM" unless it is a user-created grouping.
Bucket	Y	string	Enter the name of the grouping scheme for the Positions Report (*e.g.,* "distinct").

SortCriteria

SortCriteria is a child element of the GetPositionsReport element. Up to three criteria are allowed; if more than three are provided, then the fourth and beyond are ignored.

Table 51: SortCriteria (0-3)

Attribute	Required?	Data Type (Length)	Description
SortKeyName	Y	string	Specify the name of the sort key.
SortKeyOwner	Y	string	Specify the User ID of the owner of the sort key. Enter "SYSTEM" unless it is a user-created sort key.
Ascending	Y	boolean	Specify whether the sort should be in ascending order. Acceptable values: • true • false

To create a GetPositionsReport object, send the following:

```
Column myC1 = new Column();
myC1.setName("asset ID");
myC1.setOwner("system");
Column myC2 = new Column();
myC2.setName("asset name");
myC2.setOwner("System");
Column myC3 = new Column();
myC3.setName("price");
myC3.setOwner("system");
```

```
myColumns.add(myC1);
myColumns.add(myC2);
myColumns.add(myC3);

List<Column> myColumns = new ArrayList<Column>();

Grouping grouping = new Grouping();
myGrouping.setName("price");
myGrouping.setOwner("system");
myGrouping.setBucket("group_price");

List<SortCriteria> mySortlist = new ArrayList<SortCriteria>();
SortCriteria mySort2 = new SortCriteria();
mySort2.setSortKeyName("asset id");
mySort2.setSortKeyOwner("system");
mySort2.setAscending(false);
mySortlist.add(mySort2);
SortCriteria mySort3 = new SortCriteria();
mySort3.setSortKeyName("Asset Name");
mySort3.setSortKeyOwner("SYSTEM");
mySort3.setAscending(true);
mySortlist.add(mySort3);

CustomizeReportSetting myCRS = new CustomizeReportSetting();
myCRS.setInclBenchmarkNotHeld(false);
myCRS.setLookThruEIF(false);

Calendar myToday = Calendar.getInstance();
long myStartmil = myToday.getTimeInMillis();
ReportRoot myRoot = myPort.getPositionsReport(myColumns, myGrouping, myCRS,
  mySortlist);
```

In .NET, send the following:

```
Column[] myColumns = new Column[2];
myColumns[0] = new Column();
myColumns[0].Name = "Asset ID";
myColumns[0].Owner = "system";
myColumns[1] = new Column();
myColumns[1].Name = "holdings";
myColumns[1].Owner = "system";
CustomizeReportSetting myCrs = new CustomizeReportSetting();
myCrs.InclBenchmarkNotHeld = false;
myCrs.LookThruComposites = true;
Report[] myRoot = bdtiClient.GetPositionsReport(myColumns, null, myCrs,
  null);
```

ReportRoot

ReportRoot is the element returned as a response to the elements "GetPositionsReport" on page 48 and "GetPortfolioHistoryReport" on page 68, and as a child element of "ReportResponse" on page 82. It has one child element, Report, as indicated below.

Report

Report is the sole child element of the ReportRoot element. It has three child elements, Name, ReportDefinition, and ReportBody, as indicated below.

Table 52: ReportRoot.Report (1-n)

Attribute	Required?	Data Type (Length)	Description
Name	N	string	The name of the report.

ReportDefinition

ReportDefinition is a child element of the Report element. It has two child elements, GroupDefinition and ColDefinition, as indicated below.

GroupDefinition

GroupDefinition is a child element of the ReportDefinition element. It has two child elements, GroupDefinition and ColDefinition, as indicated below.

Table 53: ReportRoot.Report.ReportDefinition.GroupDefinition (0-n)

Attribute	Required?	Data Type (Length)	Description
Name	Y	string	The name of the grouping (*i.e.,* an available column or attribute on which the Positions Report may be grouped) for the Positions Report. A list of available groupings can be retrieved using "GetGroupings" on page 15.
Owner			The User ID of the grouping owner ("SYSTEM" unless it is a user-created grouping).
Bucket			The name of the grouping scheme for the Positions Report (*e.g.,* "distinct").

ColDefinition

ColDefinition is a child element of the ReportDefinition element. It has two child elements, Name and ColDefData, as indicated below.

Table 54: ReportRoot.Report.ReportDefinition.ColDefinition (1-n)

Attribute	Required?	Data Type (Length)	Description
Name	Y	string	The type of row for which the column is defined. In the Positions Report, the type will be either "totalRow" or "dataRow," depending upon whether the data in the row is summation data. Corresponds to ColDef in the Row element.

ColDefData

ColDefData is the sole child element of the ColDefinition element.

Table 55: ReportRoot.Report.ReportDefinition.ColDefinition.ColDefData (0-n)

Attribute	Required?	Data Type (Length)	Description
Name	Y	string	The column identifier, *e.g.,* "Asset ID," programmatically assigned by the BarraOne report generation engine, and used to identify the corresponding ColDef in the Row.
Type	Y	string	Type of data in the column, *e.g.,* "string."
Owner	Y	string	The User ID of the owner of the column data ("SYSTEM" unless it is user-created column data).

ReportBody

ReportBody is a child element of the Report element. It has two child elements: ReportGroup and Row.

ReportGroup

ReportGroup is child element of the ReportBody element. It has four attributes, and it has two child elements: Row and ReportGroup. In other words, it is a recursive element, *i.e.,* it contains itself as an element.

Table 56: ReportRoot.Report.ReportBody.ReportGroup (0-n)

Attribute	Required?	Data Type (Length)	Description
Name	Y	string	The name of the grouping (*i.e.,* an available column or attribute on which the Positions Report may be grouped) for the Positions Report, *e.g.,* "Country."
Bucket	Y	string	The name of the grouping scheme for the Positions Report, *e.g.,* 'My_Countries."
Owner	Y	string	The User ID of the grouping owner ("SYSTEM" unless it is a user-created grouping).
DisplayName	Y	string	The name of the individual bucket, *e.g.,* "USA."

Row

Row is a child element of the ReportGroup element. It has two attributes and one child element, Cell.

Table 57: ReportRoot.Report.ReportBody.ReportGroup.Row (0-n)

Attribute	Required?	Data Type (Length)	Description
ColDef	Y	string	The type of row for which the cell data is returned. In the Positions Report, the type will either be "totalRow" or "dataRow," depending upon whether the data is summation data. Corresponds to Name in the ColDefinition element.
IsTotalRow	N	boolean	Identifies a summation row. Acceptable values: • true • false (default)

Cell

Cell is the sole child element of the Row element.

Table 58: ReportRoot.Report.ReportBody.ReportGroup.Row.Cell (0-n)

Attribute	Required?	Data Type (Length)	Description
Val	Y	string	Value of the cell data.

SubmitPositionsReport

The SubmitPositionsReport element enables users to submit a Positions Report for the current portfolio in BDTI. It has two child elements, Column and KeyValue, as indicated below. It will return a ReportResponse (refer to"ReportResponse" on page 82).

Note: This is an asynchronous method that is recommended for use with a large Positions Report, instead of "GetPositionsReport" on page 48, to prevent timeout errors. It is used in conjunction with "RetrieveReport" on page 82.

Column

Column is a child element of the SubmitPositionsReport element.

Table 59: Column (1-n)

Attribute	Required?	Data Type (Length)	Description
Name	Y	string	Enter the name of a column to include in the report. A list of available column names can be retrieved using "GetReportColumnName" on page 14.
Owner	Y	string	Enter the User ID of the column owner. Enter "SYSTEM" unless it is a user-created column.

KeyValue

KeyValue is a child element of the SubmitPositionsReport element. Available keys and their default values for ReportSettings can be retrieved by using "GetReportSettings" on page 35.

Table 60: KeyValue (0-n)

Attribute	Required?	Data Type (Length)	Description
Key	Y	string	Enter the name of the report setting.
Value	Y	string	Enter the value to set for the report setting.

To create a SubmitPositionsReport object, send the following:

```
Column myC1 = new Column();
myC1.setName("asset ID");
myC1.setOwner("system");
Column myC2 = new Column();
myC2.setName("asset name");
myC2.setOwner("System");
Column myC3 = new Column();
myC3.setName("price");
myC3.setOwner("system");

myColumns.add(myC1);
myColumns.add(myC2);
myColumns.add(myC3);

List<Column> myColumns = new ArrayList<Column>();

List<KeyValue> myReportSettings = new ArrayList<KeyValue>();
  KeyValue myKVGrouping = new KeyValue();
  myKVGrouping.setKey("Grouping");
  myKVGrouping.setValue("Price:system:gr1_SYSTEM");
  myReportSettings.add(myKVGrouping);

  KeyValue myKVInclBM = new KeyValue();
  myKVInclBM.setKey("InclBenchmarkNotHeld");
  myKVInclBM.setValue("false");
  myReportSettings.add(myKVInclBM);

ReportResponse myResponse = port.submitPositionsReport(myColumns,
  myReportSettings);

if(myResponse.isDone){
  ReportRoot myReport = myResponse.getReportRoot();
}else{
  String myRequestId = myResponse.getRequestId();
  myResponse = port.retrieveReport(myRequestId);

  while(!myResponse.isDone){
    Thread.sleep(2000);
    myResponse = port.retrieveReport(myRequestId);
  }
}
```

In .NET, send the following:

```
List<Column> myColumns = new List<Column>();
myColumns.Add(new Column() { Name = "Asset ID", Owner = "SYSTEM" });
myColumns.Add(new Column() { Name = "Active Beta", Owner = "SYSTEM" })

KeyValue myRSGlobalGroup = new KeyValue();
myRSGlobalGroup.Key = "IsGlobalGroupingMode";
myRSGlobalGroup.Value = "false";

KeyValue myRSKvGrp = new KeyValue();
myRSKvGrp.Key = "Grouping";
myRSKvGrp.Value = "Price:system:gr1_SYSTEM";
KeyValue mySort = new KeyValue();
 mySort.Key = "Sort";
 mySort.Value = "Asset ID:system:y ";
 ReportResponse myResponse =
   bdtic.SubmitPositionsReport(onecolumn.ToArray(),new KeyValue[]
  {myRSKvGrp, myRSGlobalGroup, mySort });
 bool isDone=false;
 string myRequestId;
 Report[] root = null;
 if (myResponse.IsDone)
 {
   root = myResponse.ReportRoot;
   printReport(root);
 }
 else
 {
   myRequestId = myResponse.RequestId;
   try
   {
     myResponse= bdtic.RetrieveReport(myRequestId);
     while (!myResponse.IsDone)
     {
       Thread.Sleep(2000);
       myResponse = bdtic.RetrieveReport(myRequestId);
     }
     root =myResponse.ReportRoot;
     printReport(root);
   }
   catch (Exception e)
   {
     throw e;
   }
 }
```

Report Settings for Positions Report

- **InclBenchmarkNotHeld:** Indicates if the Positions Report should include benchmark assets not held in the portfolio, entered as true/false.
- **LookThruEIF:** Indicates if the Positions Report should "look through" to the constituents of equity index futures, entered as true/false.
- **LookThruComposites:** Indicates if the Positions Report should "look through" to the constituents of composite assets, entered as true/false.
- **InheritCTDBondFuture:** Indicates if the Positions Report should inherit "cheapest-to-deliver" bond futures, entered as true/false.
- **InheritCTDBondFutureOption:** Indicates if the Positions Report should inherit "cheapest-to-deliver" bond future options, entered as true/false.
- **IsGlobalGroupingMode:** Indicates if the grouping scheme in the Positions Report should be global (or local), entered as true/false.
- **ResidualReturnContribution:** Indicates if certain columns in the Positions Report are computed based on residual returns methodology, which is also known as Marginal Contribution to Residual Risk Reporting (MCRR) or x-sigma rho analysis, entered as true/false.
- **DataRows:** "True" specifies that all rows are returned. "False" specifies that only the total and group row(s) are returned from the query if the user has specified grouping. If the user has no grouping specified, then only the total rows are returned.

  ▻ **Note:** If this setting is "False," the user will be unable to save any resulting modified portfolio using "SavePortfolio" on page 44 or "SavePortfolioAs" on page 46.

- **Sort:** The name of the sort key (i.e., an available column on which the Positions Report may be sorted); the User ID of the sort key owner (e.g., "SYSTEM"); and whether the sort should be ascending (or descending).

SubmitRiskDecompReport

The SubmitRiskDecompReport element enables users to submit a Risk Decomposition Report for the current portfolio in BDTI. It has two child elements, Column and KeyValue, as indicated below. It is used in conjunction with "RetrieveReport" on page 82. It will return a ReportResponse (refer to "ReportResponse" on page 82).

Column

Column is a child element of the SubmitRiskDecompReport element.

Table 61: Column (1-n)

Attribute	Required?	Data Type (Length)	Description
Name	Y	string	Enter the name of a column to include in the report. A list of available column names can be retrieved using "GetReportColumnName" on page 14.
Owner	Y	string	Enter the User ID of the column owner. Enter "SYSTEM" unless it is a user-created column.

KeyValue

KeyValue is a child element of the SubmitRiskDecompReport element. Available keys and their default values for ReportSettings can be retrieved by using "GetReportSettings" on page 35.

Table 62: KeyValue (0-n)

Attribute	Required?	Data Type (Length)	Description
Key	Y	string	Enter the name of the report setting.
Value	Y	string	Enter the value to set for the report setting.

To create a SubmitRiskDecompReport object, send the following:

```
Column myC1 = new Column();
myC1.setName("Active Risk");
myC1.setOwner("system");
Column myC2 = new Column();
myC2.setName("Benchmark Risk");
myC2.setOwner("System");
Column myC3 = new Column();
myC3.setName("Portfolio Risk");
myC3.setOwner("system");
```

```
List<Column> myColumns = new ArrayList<Column>();

myColumns.add(myC1);
myColumns.add(myC2);
myColumns.add(myC3);

List<KeyValue> myReportSettings = new ArrayList<KeyValue>();

  KeyValue myHNR = new KeyValue();
  myHNR.setKey("HideNullRows");
  myKHNR.setValue("true");
  myReportSettings.add(myHNR);

  KeyValue mySGI = new KeyValue();
  mySGI.setKey("ShowGroupInteraction");
  mySGI.setValue("false");
  myReportSettings.add(mySGI);

ReportResponse myResponse = port.submitRiskDecompReport(myColumns,
  myReportSettings);

if(myResponse.isDone){
  ReportRoot myReport = myResponse.getReportRoot();
}else{
  String myRequestId = myResponse.getRequestId();
  myResponse = port.retrieveReport(myRequestId);

  while(!myResponse.isDone){
    Thread.sleep(2000);
    myResponse = port.retrieveReport(myRequestId);
  }
}
```

In .NET, send the following:

```
List<Column> myColumns = new List<Column>();
myColumns.Add(new Column() { Name = "Benchmark Risk", Owner = "SYSTEM" });
myColumns.Add(new Column() { Name = "Portfolio Risk", Owner = "SYSTEM" })

KeyValue myHNR = new KeyValue();
myHNR.Key = "HideNullRows";
myHNR.Value = "true";

KeyValue mySGI = new KeyValue();
mySGI.Key = "ShowGroupInteraction";
mySGI.Value = "false";

ReportResponse myResponse =
   bdtic.SubmitRiskDecompReport(onecolumn.ToArray(),new KeyValue[]
  {myHNR, mySGI });
 bool isDone=false;
 string myRequestId;
 Report[] root = null;
 if (myResponse.IsDone)
 {
   root = myResponse.ReportRoot;
   printReport(root);
 }
 else
 {
   myRequestId = myResponse.RequestId;
   try
   {
    myResponse= bdtic.RetrieveReport(myRequestId);
    while (!myResponse.IsDone)
    {
     Thread.Sleep(2000);
     myResponse = bdtic.RetrieveReport(myRequestId);
    }
    root =myResponse.ReportRoot;
    printReport(root);

   }
   catch (Exception e)
   {
    throw e;
   }
 }
```

Report Settings for Risk Decomposition Report

- **HideNullRows:** Hides rows that have null values *in all cells* of a Risk Decomposition Report. This enables you to exclude risk sources that have no bearing on your analysis, such as Term Structure and Spread in an equity portfolio. Entered as true/false.
- **ShowGroupInteraction:** Specifies showing figures for the interaction term within each factor group (correlation among factor bets) in a Risk Decomposition Report. The interaction values appear in the Variance and %Risk columns of the report. The Risk and VaR columns show N/A. Entered as true/false.
- **FactorTree:** This setting enables you to see your preferred breakdown of common factor risk in a Risk Decomposition Report by selecting a factor tree that your or someone in your firm has created. To retrieve a list of available factor trees, refer to ""GetFactorTree" on page 20.
- **ResidualReturnCalculation:** When set to "False," then the Local Market Risk contribution is decomposed into a Market Timing Risk contribution and a residual risk contribution. The residual risk contribution is reported as a Common Factor Risk contribution and a Selection Risk contribution based on residual factor exposures and asset weights.

  When set to "True," then Marginal Contribution to Residual Risk Reporting is used as an alternative methodology for attributing the residual risk contribution, and it provides a different view of the balance of the residual risk contribution attributable to Common Factor Risk and Selection Risk sources. In this methodology, residual risk is attributed on the basis of residual returns (*i.e.,* the portion of returns uncorrelated with benchmark returns), rather than that on the basis of residual exposures to non-residual returns. This approach is applicable only when the user has specified both a market and a benchmark for a portfolio, and when the user would like to drill into the both the Market Timing Risk Contribution and Residual Risk Contribution on the basis of the market timing return and the residual return components of each source of risk.

SubmitFEBReport

The SubmitFEBReport element enables users to submit a Factor Exposure Breakdown Report for the current portfolio in BDTI. It has one child element, KeyValue, as indicated below. It is used in conjunction with "RetrieveReport" on page 82. It will return a ReportResponse (refer to"ReportResponse" on page 82).

KeyValue

KeyValue is a child element of the SubmitFEBReport element. Available keys and their default values for ReportSettings can be retrieved by using "GetReportSettings" on page 35.

Table 63: KeyValue (0-n)

Attribute	Required?	Data Type (Length)	Description
Key	Y	string	Enter the name of the report setting.
Value	Y	string	Enter the value to set for the report setting.

To create a SubmitFEBReport object, send the following:

```
List<KeyValue> myReportSettings = new ArrayList<KeyValue>();

  KeyValue myHNR = new KeyValue();
  myHNR.setKey("HideNullRows");
  myHNR.setValue("false");
  myReportSettings.add(myHNR);

ReportResponse myResponse = port.submitFEBReport(myReportSettings);

if(myResponse.isDone){
  ReportRoot myReport = myResponse.getReportRoot();
}else{
  String myRequestId = myResponse.getRequestId();
  myResponse = port.retrieveReport(myRequestId);

  while(!myResponse.isDone){
    Thread.sleep(2000);
    myResponse = port.retrieveReport(myRequestId);
  }
}
```

In .NET, send the following:

```
KeyValue myHNR = new KeyValue();
myHNR.Key = "HideNullRows";
myHNR.Value = "false";


ReportResponse myResponse =
   bdtic.SubmitFEBReport(new KeyValue[]
  {myHNR});
 bool isDone=false;
 string myRequestId;
 Report[] root = null;
 if (myResponse.IsDone)
 {
  root = myResponse.ReportRoot;
  printReport(root);
 }
 else
 {
  myRequestId = myResponse.RequestId;
  try
  {
   myResponse= bdtic.RetrieveReport(myRequestId);
   while (!myResponse.IsDone)
   {
    Thread.Sleep(2000);
    myResponse = bdtic.RetrieveReport(myRequestId);
   }
   root =myResponse.ReportRoot;
   printReport(root);
  }
  catch (Exception e)
  {
   throw e;
  }
 }
```

Report Settings for Factor Exposure Breakdown Report

- **HideNullRows:** Hides rows that have null values in *all cells* of a Factor Exposure Breakdown Report. This enables you to exclude factors that have no bearing on your analysis. Entered as true/false.
- **HideNullCol:** Hide columns that have null values in *all cells* of a Factor Exposure Breakdown Report. For example, if you use CASH as a benchmark, benchmark values will be zero. Entered as true/false. **Note:** If this option is selected along with a factor grouping scheme, then the MCTR and MCAR columns will not appear in the report.
- **DisplayContributionToTR:** Add a column in a Factor Exposure Breakdown Report displaying each factor's percent contribution to total risk. This enables you to see which factors contribute most to your portfolio's total risk. Entered as true/false.
- **DisplayContributionToAR:** Add a column in a Factor Exposure Breakdown Report displaying each factor's percent contribution to active risk. This enables you to see which factors contribute most to your portfolio's active risk. Entered as true/false.
- **ShowDetail:** If you choose a factor type grouping scheme, you will show values for each item within a group, in addition to the aggregate group values in a Factor Exposure Breakdown Report. Entered as true/false.
- **LocalMarketGrouping:** Enter the local market grouping scheme you want in a Factor Exposure Breakdown Report. This enables you to cluster local markets into regions and view aggregate values for each region. Entered as NAME:OWNER. To retrieve a list of available local market grouping schemes, refer to "GetLocalMarketGrouping" on page 23.
- **LocalMarket:** If you specify All Within Market or Emerging Market as the Factor Type for the report, you can choose whether you want to view factors for a particular local market or all markets in a Factor Exposure Breakdown Report. Enter "All," or refer to "GetLocalMarket" on page 24 to retrieve a list of available local markets.
- **FactorType:** Enter "All," or refer to "GetFactorTypes" on page 21 to enter the factor type you would like to display in a Factor Exposure Breakdown Report.
- **FactorGroupingScheme:** Choose the factor grouping scheme you want, if any, in a Factor Exposure Breakdown Report. This enables you to organize the factors into categories so you can view aggregate values for each category. Refer to "GetFactorGroupings" on page 22 to retrieve a list of available factor grouping schemes.
- **CurrencyExposure:** If you choose Currency as the factor type in a Factor Exposure Breakdown Report, you can choose to display "Implicit," "Explicit," or "Total" exposure.
- **ShowLongShort:** Displays factor exposures and risk for the long and short legs of your portfolio separately in a Factor Exposure Breakdown Report. Entered as true/false.
- **DisplaySTB:** If "true," fixed income exposures and risk are displayed in terms of Barra's Shift/Twist/Butterfly factors (STB) in the Factor Exposure Breakdown Report; if "false," fixed income exposures and risk are displayed in terms of Key Rate Durations (KRD). (This is the equivalent of the "Term Structure Factors" option available in the **Display Settings** area on the **My Profile** page of the **Accounts** tab in the BarraOne interface.) Entered as true/false.

- **ShowCorrelation:** Displays the following columns in the Factor Exposure Breakdown Report (entered as true/false):
  - Portfolio Correlation
  - Active Portfolio Correlation
  - Mkt Timing Correlation
  - Residual Correlation
  - Mkt Timing Active Correlation
  - Active Residual Correlation
  - Active Currency Correlation.
- **DisplayResidualReturns:** Displays the following columns in the Factor Exposure Breakdown Report (entered as true/false):
  - Local Beta (Mkt)
  - Mkt Timing Volatility
  - Mkt Timing MCTR
  - Mkt Timing Risk Contribution
  - Residual Volatility
  - Residual MCTR
  - Residual Contribution
  - Mkt Timing MCAR
  - Active Mkt Timing Risk Contribution
  - Residual MCAR
  - Active Residual Contribution
  - Active Currency Risk
  - Active Currency MCTE
  - Active Currency Correlation
- **DisplayResidualExposures:** Displays the following columns in the Factor Exposure Breakdown Report (entered as true/false):
  - Cont. to TR (Residual)
  - Cont. to AR (Residual)
- **Sort:** The name of the Sort By and Sort Order key (i.e., an available column on which the Factor Exposure Breakdown Report may be sorted); the User ID of the sort key owner (e.g., "SYSTEM"); and whether the sort should be ascending (or descending).

GetPortfolioHistoryReport

The GetPortfolioHistoryReport element enables users to retrieve the history of the current portfolio in BarraOne. The signature of this element has no arguments. It will return a ReportRoot (refer to "ReportRoot" on page 53).

To create a GetPortfolioHistoryReport object, send the following:

```
ReportRoot myRoot = myPort.getPortfolioHistoryReport();
```

In .NET, send the following:

```
Report[] myRoot = bdtiClient.GetPortfolioHistoryReport();
```

GetScenarioNames

The GetScenarioNames element enables users to get a list of available stress testing market scenarios in BarraOne associated with the user specified as the scenario filter. It will return an array of NameOwner objects. This provides users with a way to locate the name of an existing scenario they want to use for their stress test portfolio report. The output can then be used in "SubmitStressTestPortfolioReport" on page 69.

Filter

Filter is the sole child element of the GetScenarioNames element.

Table 64: Filter (0-1)

Attribute	Required?	Data Type (Length)	Description
Filter	N	string	Enter the name of a filter. A filter is the equivalent of an item in the dropdown list (such as owner or workgroup) available in the Select Scenario dialog in the BarraOne application. The user can obtain a list of available filters using "GetFilters" on page 8.

To create a GetScenarioNames object, send the following:

```
List<NameOwner> myScenarios = myPort.getScenarioNames("myFilter");
```

In .NET, send the following:

```
NameOwner[] myScenarios = bdtiClient.GetScenarioNames("myFilter");
```

NameOwner

An array of NameOwner objects is returned as a response to the GetScenarioNames element.

Table 65: NameOwner (0-n)

Attribute	Required?	Data Type (Length)	Description
Name	Y	string	The name of a scenario. The retrieved scenario can be used in "SubmitStressTestPortfolioReport" on page 69.
Owner	Y	string	The User ID of the scenario owner. ("SYSTEM" unless it is a user-created scenario.)

SubmitStressTestPortfolioReport

The SubmitStressTestPortfolioReport element enables users to submit a Stress Test Report for the current portfolio and analysis date in BDTI. It has three child elements, Column, NameOwner, and KeyValue, as indicated below. It is used in conjunction with "RetrieveReport" on page 82. It will return a ReportResponse (refer to"ReportResponse" on page 82).

Column

Column is a child element of the SubmitStressTestPortfolioReport element.

Table 66: Column (1-n)

Attribute	Required?	Data Type (Length)	Description
Name	Y	string	Enter the name of a column to include in the report. A list of available column names can be retrieved using "GetReportColumnName" on page 14.
Owner	Y	string	Enter the User ID of the column owner. Enter "SYSTEM" unless it is a user-created column.

NameOwner

NameOwner is a child element of the SubmitStressTestPortfolioReport element.

Table 67: NameOwner (1-n)

Attribute	Required?	Data Type (Length)	Description
Name	Y	string	Enter the name of a scenario to include in the report. A list of available scenarios can be retrieved using "GetScenarioNames" on page 68.
Owner	Y	string	Enter the User ID of the scenario owner. Enter "SYSTEM" unless it is a user-created scenario.

KeyValue

KeyValue is a child element of the SubmitStressTestPortfolioReport element. Available keys and their default values for ReportSettings can be retrieved by using "GetReportSettings" on page 35.

Table 68: KeyValue (0-n)

Attribute	Required?	Data Type (Length)	Description
Key	Y	string	Enter the name of the report setting.
Value	Y	string	Enter the value to set for the report setting.

To create a SubmitStressTestPortfolioReport object, send the following:

```
Column myC1 = new Column();
myC1.setName("Modified Duration");
myC1.setOwner("system");
Column myC2 = new Column();
myC2.setName("Macaulay Duration");
myC2.setOwner("System");
Column myC3 = new Column();
myC3.setName("price");
myC3.setOwner("system");

List<Column> myColumns = new ArrayList<Column>();

myColumns.add(myC1);
myColumns.add(myC2);
myColumns.add(myC3);
```

```
NameOwner myS1 = new NameOwner();
myS1.setName("Scenario1");
myS1.setOwner("myUser");
NameOwner myS2 = new NameOwner();
myS2.setName("Scenario2");
myS2.setOwner("myUser");
NameOwner myS3 = new NameOwner();
myS3.setName("Scenario3");
myS3.setOwner("myUser");

List<NameOwner> myScenarios = new ArrayList<NameOwner>();

myScenarios.add(myS1);
myScenarios.add(myS2);
myScenarios.add(myS3);

List<KeyValue> myReportSettings = new ArrayList<KeyValue>();
  KeyValue myKVGrouping = new KeyValue();
  myKVGrouping.setKey("Grouping");
  myKVGrouping.setValue("Price:system:gr1_SYSTEM");
  myReportSettings.add(myKVGrouping);

  KeyValue myKVInclBM = new KeyValue();
  myKVInclBM.setKey("InclBenchmarkNotHeld");
  myKVInclBM.setValue("false");
  myReportSettings.add(myKVInclBM);
ReportResponse myResponse = port.submitStressTestPortfolioReport(myColumns,
  myScenarios, myReportSettings);

if(myResponse.isDone){
  ReportRoot myReport = myResponse.getReportRoot();
}else{
  String myRequestId = myResponse.getRequestId();
  myResponse = port.retrieveReport(myRequestId);

  while(!myResponse.isDone){
    Thread.sleep(2000);
    myResponse = port.retrieveReport(myRequestId);
  }
}
```

In .NET, send the following:

```
Column[] myColumns = new Column[3];
myColumns[0] = new Column();
myColumns[0].Name = "Initial Market Value";
myColumns[0].Owner = "SYSTEM";
myColumns[1] = new Column();
myColumns[1].Name = "Final Market Value";
myColumns[1].Owner = "SYSTEM";
myColumns[2] = new Column();
myColumns[2].Name = "Beta";
myColumns[2].Owner = "SYSTEM";

KeyValue[] myStressTestPortfolioSettings = new KeyValue[5];

myStressTestPortfolioSettings[0] = new KeyValue();
myStressTestPortfolioSettings[0].Key = "InheritCTDBondFutureOption";
myStressTestPortfolioSettings[0].Value = "false";

myStressTestPortfolioSettings[1] = new KeyValue();
myStressTestPortfolioSettings[1].Key = "LookThruEIF";
myStressTestPortfolioSettings[1].Value = "false";

myStressTestPortfolioSettings[2] = new KeyValue();
myStressTestPortfolioSettings[2].Key = "LookThruComposites";
myStressTestPortfolioSettings[2].Value = "false";
myStressTestPortfolioSettings[3] = new KeyValue();
myStressTestPortfolioSettings[3].Key = "InheritCTDBondFuture";
myStressTestPortfolioSettings[3].Value = "false";

myStressTestPortfolioSettings[4] = new KeyValue();
myStressTestPortfolioSettings[4].Key = "Sort";
myStressTestPortfolioSettings[4].Value = "name1:owner1:y";

NameOwner[] scenarios = new NameOwner[1];
scenarios[0] = new NameOwner();
scenarios[0].Name = "2001 Sept 11";
scenarios[0].Owner = "system";

ReportResponse myStressTestPortfolioReportResponse =
  BDTiClient.SubmitStressTestPortfolioReport(myColumns, scenarios,
  myStressTestPortfolioSettings);
myStressTestPortfolioReportResponse =
  BDTiClient.RetrieveReport(myStressTestPortfolioReportResponse.RequestId);
```

Report Settings for Stress Test Portfolio Report

- **LookThruEIF:** Indicates if the Stress Test Portfolio Report should "look through" to the constituents of equity index futures, entered as true/false.
- **LookThruComposites:** Indicates if the Stress Test Portfolio Report should "look through" to the constituents of composite assets, entered as true/false.
- **InheritCTDBondFuture:** Indicates if the Stress Test Portfolio Report should inherit "cheapest-to-deliver" bond futures, entered as true/false.
- **InheritCTDBondFutureOption:** Indicates if the Stress Test Portfolio Report should inherit "cheapest-to-deliver" bond future options, entered as true/false.
- **Grouping:** The name of the grouping (i.e., an available column or attribute on which the Stress Test Portfolio Report may be grouped); the User ID of the grouping owner (e.g., "SYSTEM"); and the name of the grouping scheme bucket (e.g., "distinct").
- **Sort:** The name of the scenario (only if the column is associated with a specific scenario); the name of the sort key (i.e., an available column or attribute on which the Stress Test Portfolio Report may be sorted); the User ID of the sort key owner (e.g., "SYSTEM"); and whether the sort should be ascending (or descending).
- **UseDollarPL:** If true, adds an additional report column ("$P&L") in which P/L is expressed in currency, entered as true/false.

SubmitAllocationSelectionReport

The SubmitAllocationSelectionReport element enables users to submit a Allocation-Selection Report for the current portfolio in BDTI. It has two child elements, Column and KeyValue, as indicated below. It is used in conjunction with "RetrieveReport" on page 82. It will return a ReportResponse (refer to "ReportResponse" on page 82).

Column

Column is a child element of the SubmitAllocationSelectionReport element.

Table 69: Column (1-n)

Attribute	Required?	Data Type (Length)	Description
Name	Y	string	Enter the name of a column to include in the report. A list of available column names can be retrieved using "GetReportColumnName" on page 14.
Owner	Y	string	Enter the User ID of the column owner. Enter "SYSTEM" unless it is a user-created column.

KeyValue

KeyValue is a child element of the SubmitAllocationSelectionReport element. Available keys and their default values for ReportSettings can be retrieved by using "GetReportSettings" on page 35.

Table 70: KeyValue (0-n)

Attribute	Required?	Data Type (Length)	Description
Key	Y	string	Enter the name of the report setting.
Value	Y	string	Enter the value to set for the report setting.

To create a SubmitAllocationSelectionReport object, send the following:

```
Column myC1 = new Column();
myC1.setName("Active Risk");
myC1.setOwner("system");
Column myC2 = new Column();
myC2.setName("Allocation Risk");
myC2.setOwner("System");
Column myC3 = new Column();
myC3.setName("Selection Risk");
myC3.setOwner("system");
```

```
List<Column> myColumns = new ArrayList<Column>();

myColumns.add(myC1);
myColumns.add(myC2);
myColumns.add(myC3);

List<KeyValue> myReportSettings = new ArrayList<KeyValue>();

  KeyValue myXSR = new KeyValue();
  myXSR.setKey("IsXsigmaRho");
  myXSR.setValue("false");
  myReportSettings.add(myXSR);

  KeyValue mySTR = new KeyValue();
  mySTR.setKey("ShowSubTotalRows");
  mySTR.setValue("false");
  myReportSettings.add(mySTR);

ReportResponse myResponse = port.submitAllocationSelectionReport(myColumns,
  myReportSettings);

if(myResponse.isDone){
  ReportRoot myReport = myResponse.getReportRoot();
}else{
  String myRequestId = myResponse.getRequestId();
  myResponse = port.retrieveReport(myRequestId);

  while(!myResponse.isDone){
    Thread.sleep(2000);
    myResponse = port.retrieveReport(myRequestId);
  }
}
```

In .NET, send the following:

```
List<Column> myColumns = new List<Column>();
myColumns.Add(new Column() { Name = "Active Risk", Owner = "SYSTEM" });
myColumns.Add(new Column() { Name = "Allocation Risk", Owner = "SYSTEM" });
myColumns.Add(new Column() { Name = "Selection Risk", Owner = "SYSTEM" })

KeyValue myXSR = new KeyValue();
myXSR.Key = "IsXsigmaRho";
myXSR.Value = "false";

KeyValue mySTR = new KeyValue();
mySTR.Key = "ShowSubTotalRows";
mySTR.Value = "false";

ReportResponse myResponse =
   bdtic.SubmitAllocationSelectionReport(onecolumn.ToArray(),new
 KeyValue[]
  {myXSR, mySTR });
 bool isDone=false;
 string myRequestId;
 Report[] root = null;
 if (myResponse.IsDone)
 {
   root = myResponse.ReportRoot;
   printReport(root);
 }
 else
 {
   myRequestId = myResponse.RequestId;
   try
   {
     myResponse= bdtic.RetrieveReport(myRequestId);
     while (!myResponse.IsDone)
     {
       Thread.Sleep(2000);
       myResponse = bdtic.RetrieveReport(myRequestId);
     }
     root =myResponse.ReportRoot;
     printReport(root);

   }
   catch (Exception e)
   {
     throw e;
```

```
    }
}
```

Report Settings for Allocation-Selection Report

- **IsXsigmaRho:** "True" specifies that the decomposition methodology is X-Sigma-Rho. With this methodology, risk is additive. The interaction term is included in the calculation of risk, and correlation among groups is attributed to the groups, themselves. You can view the contribution of each group's interaction term to active total risk.

  "False" specifies that the decomposition methodology is Variance. With this methodology, the interaction term is separated from the calculation of risk, and you can view the interaction term separately to see the correlation among allocation groups. In this framework, variance (the square of the standard deviation) is additive.

  Entered as true/false.

- **ShowSubTotalRows:** Applicable only for Variance decomposition methodology.

  "True" specifies that four total rows are displayed at the bottom of each column: **Active Common Factor Risk**, **Active Asset Selection Risk**, **Active Total Risk**, and **Interaction**.

  "False" specifies that only one total row is displayed: **Active Total Risk**.

  Entered as true/false.

- **CombineSelInt:** Available only for X-Sigma-Rho decomposition methodology. "True" specifies that the interaction term is combined with the selection term. Combining the selection and interaction terms means that the values in the **Interaction Risk Contribution** column are added to the values in the **Selection Risk Contribution** column, and the **Interaction Risk Contribution** column shows zeroes. Entered as true/false.
- **ApplyEmptyRule:** Available only for X-Sigma-Rho decomposition methodology. "True" specifies that a benchmark with zero weight will not be treated as cash in the decomposition of the Allocation-Selection Report. Rather, all of the risk is attributed to Allocation, instead of Selection/Interaction decisions, much like the way in which Variance decomposition is handled. Entered as true/false.

The values of the following columns are affected when this option is selected for X-Sigma-Rho:

- Allocation Risk
- Selection Risk - Changed only when combined with Interaction (because Interaction is changing)
- Interaction Risk
- Allocation Relative Sector Volatility
- Allocation Relative Sector Correlation
- Allocation MCAR
- Allocation Risk Contribution
- Allocation Relative Sector VaR
- Selection Relative Sector Volatility
- Selection Relative Sector Correlation
- Selection MCAR
- Selection Risk Contribution
- Selection Relative Sector VaR
- Interaction Risk Contribution

The same logic applies for ex. Currency columns.

Currency Risk and Local Market Risk do not change.

- **Grouping:** The name of the grouping (i.e., an available column or attribute on which the Allocation-Selection Report may be grouped); the User ID of the grouping owner (e.g., "SYSTEM"); and the name of the grouping scheme bucket (e.g., "distinct"), entered as NAME1:OWNER1:BUCKET1, NAME2:OWNER2:BUCKET2.
- **PrintRowLevel:** When set to "All," the row-level column is printed for each requested date in horizontal layout and once for all dates in vertical and point-in-time reports. When set to "Once," the behavior is the same as "All" for vertical and point-in- time reports; for horizontal time series reports, the row-level column is printed for the first date in the output and omitted from the remaining dates. When set to "None," these rows are omitted entirely.
- **PrintColumnHeaders:** When set to "All," the header is printed for each requested date in vertical layout and once for all dates in horizontal and point-in-time reports. When set to "Once," the behavior is the same as "All" for horizontal and point-in-time reports; for vertical time series reports, the headers are printed for the first date in the output and omitted from the remaining dates. When set to "None," the column headers are omitted entirely.

SubmitStressTestSummaryReport

The SubmitStressTestSummaryReport element enables users to submit a Stress Test Summary Report for the current portfolio and analysis date in BDTI. It has two child elements, Column and NameOwner, as indicated below. It is used in conjunction with "RetrieveReport" on page 82. It will return a ReportResponse (refer to"ReportResponse" on page 82).

Column

Column is a child element of the SubmitStressTestSummaryReport element.

Table 71: Column (1-n)

Attribute	Required?	Data Type (Length)	Description
Name	Y	string	Enter the name of a column to include in the report. A list of available column names can be retrieved using "GetReportColumnName" on page 14.
Owner	Y	string	Enter the User ID of the column owner. Enter "SYSTEM" unless it is a user-created column.

NameOwner

NameOwner is a child element of the SubmitStressTestSummaryReport element.

Table 72: NameOwner (1-n)

Attribute	Required?	Data Type (Length)	Description
Name	Y	string	Enter the name of a scenario to include in the report. A list of available scenarios can be retrieved using "GetScenarioNames" on page 68.
Owner	Y	string	Enter the User ID of the scenario owner. Enter "SYSTEM" unless it is a user-created scenario.

To create a SubmitStressTestSummaryReport object, send the following:

```
Column myC1 = new Column();
myC1.setName("Active P&L");
myC1.setOwner("System");
Column myC2 = new Column();
myC2.setName("P&L (Mkt Val)");
myC2.setOwner("System");
Column myC3 = new Column();

List<Column> myColumns = new ArrayList<Column>();
```

```
myColumns.add(myC1);
myColumns.add(myC2);

NameOwner myS1 = new NameOwner();
myS1.setName("Scenario1");
myS1.setOwner("myUser");
NameOwner myS2 = new NameOwner();
myS2.setName("Scenario2");
myS2.setOwner("myUser");
NameOwner myS3 = new NameOwner();
myS3.setName("Scenario3");
myS3.setOwner("myUser");

List<NameOwner> myScenarios = new ArrayList<NameOwner>();

myScenarios.add(myS1);
myScenarios.add(myS2);
myScenarios.add(myS3);

ReportResponse myResponse = port.submitStressTestSummaryReport(myColumns,
  myScenarios);

if(myResponse.isDone){
  ReportRoot myReport = myResponse.getReportRoot();
}else{
  String myRequestId = myResponse.getRequestId();
  myResponse = port.retrieveReport(myRequestId);

  while(!myResponse.isDone){
    Thread.sleep(2000);
    myResponse = port.retrieveReport(myRequestId);
  }
}
```

In .NET, send the following:

```
Column[] myColumns = new Column[3];
myColumns[0] = new Column();
myColumns[0].Name = "Active P&L";
myColumns[0].Owner = "SYSTEM";
myColumns[1] = new Column();
myColumns[1].Name = "P&L (Mkt Val)";
myColumns[1].Owner = "SYSTEM";
```

```
NameOwner[] scenarios = new NameOwner[1];
scenarios[0] = new NameOwner();
scenarios[0].Name = "2001 Sept 11";
scenarios[0].Owner = "system";

ReportResponse myStressTestSummaryReportResponse =
  BDTiClient.SubmitStressTestSummaryReport(myColumns, scenarios);
myStressTestSummaryReportResponse =
  BDTiClient.RetrieveReport(myStressTestSummaryReportResponse.RequestId);
```

RetrieveReport

The RetrieveReport element takes a RequestID as input and enables users to retrieve a Positions Report, Risk Decomposition Report, Factor Exposure Breakdown Report, Stress Test Portfolio Report, Allocation-Selection Report, or Stress Test Summary Report in BDTI. It will return a ReportResponse (refer to"ReportResponse" on page 82).

Table 73: RequestID (1)

Attribute	Required?	Data Type (Length)	Description
RequestID	Y	string	Enter the RequestID for the report to be retrieved. Refer to "SubmitPositionsReport" on page 56, "SubmitRiskDecompReport" on page 60, "SubmitFEBReport" on page 64, "SubmitStressTestPortfolioReport" on page 69, "SubmitAllocationSelectionReport" on page 74, or "SubmitStressTestSummaryReport" on page 79.

To create a RetrieveReport object, send the following:

```
ReportResponse myResponse = port.retrieveReport("myRequestID");
```

In .NET, send the following:

```
ReportResponse myResponse = bdtiClient.RetrieveReport("myRequestID");
```

ReportResponse

ReportResponse is returned as a response to the "SubmitPositionsReport" on page 56, "SubmitRiskDecompReport" on page 60, "SubmitFEBReport" on page 64, "SubmitStressTestPortfolioReport" on page 69, "SubmitAllocationSelectionReport" on page 74, "RetrieveReport" on page 82, and "SubmitStressTestSummaryReport" on page 79 elements. It has three child elements as listed in the table below.

Table 74: ReportResponse (1)

Attribute	Required?	Data Type (Length)	Description
RequestID	Y	string	The RequestID generated for the submitted report.
IsDone	Y	boolean	Indicates if the report has been submitted successfully. Acceptable values: • true • false
ReportRoot			Refer to "ReportRoot" on page 53.

GetExportJobStream

The GetExportJobStream element enables users to retrieve an export set, in streaming fashion, based upon the parameters specified using other operations in BarraOne Developer's Toolkit Interactive. It is included in BDTI primarily for .NET clients who would otherwise have difficulty downloading large export sets. This element has four child elements as listed in the table below. It will return a DataHandler in Java or a Stream in .NET.

Table 75: GetExportJobStream (1)

Attribute	Required?	Data Type (Length)	Description
User	Y	string	Enter your User ID.
Client	Y	string	Enter the Client ID for your firm or organization.
Password	Y	string	Enter your Password.
JobID	Y	string	Enter an ID for the export job.

To create a GetExportJobStream object, send the following:

```
DataHandler myHandler = port.getExportJobStream("myUserID", "myClientID",
  "myJobID", "myJobID");
InputStream in;
File myOutFile = new File("c:/outfromWS.zip");
byte[] buf = new byte[256];
try {
  FileOutputStream fos = new FileOutputStream(myOutFile);
  if (myHandler instanceof StreamingDataHandler) {
    in = ((StreamingDataHandler) myHandler).readOnce();
    int leng;
    while( (leng = in.read(buf,0,256))>0){
      fos.write(buf,0,leng);
    }
    fos.close();
    in.close();
  } else {
in = myHandler.getInputStream();
}
} catch (IOException e) {
  e.printStackTrace();
}
```

In .NET, send the following:

```
Stream myOutstream = bdtiClient.GetExportJobStream("myUserID",
  "myClientID", "myJobID", "myJobID");
```

GetCurrentVersion

The GetCurrentVersion element enables users to get the current release version number of BarraOne. The signature of this element has no arguments. It will return a "Version" string.

To create a GetSystemDates object, send the following:

```
Version myVersion = myPort.getCurrentVersion();
```

In .NET, send the following:

```
Version myVersion = bdtiClient.GetCurrentVersion();
```

Version

Version is the sole element returned in response to the GetCurrentVersion element.

Table 76: Version (1)

Attribute	Required?	Data Type (Length)	Description
Version	Y	string	The version of BarraOne.

Logout

The Logout element enables the user to log out of the BarraOne Developer's Toolkit Interactive environment. The signature of this element has no arguments.

To create a Logout object, send the following:

```
myPort.logout();
```

In .NET, send the following:

```
bdtiClient.Logout;
```

Appendices

Appendix A — Country and Currency Codes

The following table lists the ISO country codes used in BarraOne and their corresponding ISO currency codes.

Country Currency	3-letter Country Code	2-letter Country Code	Currency Code	Supported For		
				Currency of Issue	Issuer Country	Price Currency
Afghanistan Afghan Afghani	AFG	AF	AFN			
Albania Albanian Lek	ALB	AL	ALL			BIM
Algeria Algerian Dinar	DZA	DZ	DZD			BIM
American Samoa US Dollar	ASM	AS	USD			BIM
Andorra Euro	AND	AD	EUR			BIM
Angola Angola Kwanza	AGO	AO	AOA			BIM
Anguilla East Caribbean Dollar	AIA	AI	XCD			BIM
Antarctica	ATA	AQ	XXX			
Antigua and Barbuda East Caribbean Dollar	ATG	AG	XCD			BIM
Argentina Argentine Peso	ARG	AR	ARS		BIM	BIM
Armenia Armenian Dram	ARM	AM	AMD			
Aruba Aruban Florin	ABW	AW	AWG			BIM
Australia Australian Dollar	AUS	AU	AUD†$	BIM	BIM	BIM
Austria* Austrian Schilling	AUT	AT	ATS EUR		BIM	BIM
Azerbaijan Azerbaijanian Manat	AZE	AZ	AZN			BIM
Bahamas Bahamian Dollar	BHS	BS	BSD			BIM
Bahrain Bahraini Dinar	BHR	BH	BHD			BIM
Bangladesh Bangladeshi Taka	BGD	BD	BDT			BIM303 only
Barbados Barbados Dollar	BRB	BB	BBD USD			BIM
Belarus Belarus Ruble	BLR	BY	BYR/BYN			BIM
Belgium* Belgian Franc	BEL	BE	BEF EUR		BIM	BIM
Belize Belize Dollar	BLZ	BZ	BZD			BIM

Country Currency	3-letter Country Code	2-letter Country Code	Currency Code	Supported For		
				Currency of Issue	Issuer Country	Price Currency
Benin CFA Franc BCEAO	BEN	BJ	XOF			BIM
Bermuda Bermudian Dollar	BMU	BM	BMD USD			BIM
Bhutan Bhutanese Ngultrum	BTN	BT	BTN			BIM
Bolivia Boliviano	BOL	BO	BOB			BIM
Bosnia and Herzegovina Bosnia Herzegovina Mark	BIH	BA	BAM			BIM
Botswana Botswana Pula	BWA	BW	BWP			BIM303 only
Bouvet Island Norwegian Krone	BVT	BV	NOK			BIM
Brazil Brazilian Real	BRA	BR	BRL$	BIM	BIM	BIM
British Indian Ocean Territory US Dollar	IOT	IO	USD GBP			BIM
Brunei Darussalam Brunei Dollar	BRN	BN	BND			BIM
Bulgaria Bulgarian Lev	BGR	BG	BGN	BIM	BIM	BIM
Burkina Faso CFA Franc BCEAO	BFA	BF	XOF			BIM
Burundi Burundian Franc	BDI	BI	BIF			BIM
Cape Verde Cape Verde Escudo	CPV	CV	CVE			
Cambodia Cambodian Riel	KHM	KH	KHR			
Cameroon CFA Franc BEAC	CMR	CM	XAF			BIM
Canada Canadian Dollar	CAN	CA	CAD†$	BIM	BIM	BIM
Cayman Islands Cayman Islands Dollar	CYM	KY	KYD			BIM
Central African Republic CFA Franc BEAC	CAF	CF	XAF			BIM
Chad CFA Franc BEAC	TCD	TD	XAF			BIM
Chile Chilean Peso	CHL	CL	CLP	BIM	BIM	BIM
China Chinese Yuan	CHN	CN	CNY†	BIM	BIM	BIM
China International	CHX		CNH		BIM	
Christmas Island Australian Dollar	CXR	CX	AUD			BIM
Cocos (Keeling) Islands Australian Dollar	CCK	CC	AUD			BIM
Colombia Colombian Peso	COL	CO	COP	BIM	BIM	BIM

Country Currency	3-letter Country Code	2-letter Country Code	Currency Code	Supported For		
				Currency of Issue	Issuer Country	Price Currency
Comoros Comoro Franc	COM	KM	KMF			BIM
Congo CFA Franc BEAC	COG	CG	XAF			BIM
Congo, Democratic Republic of Congolese Franc	COD	CD	CDF			BIM
Cook Islands New Zealand Dollar	COK	CK	NZD#			BIM
Costa Rica Costa Rican Colon	CRI	CR	CRC			BIM
Croatia Croatian Kuna	HRV	HR	HRK	BIM	BIM	BIM
Cuba Cuban Peso	CUB	CU	CUP			BIM
Cyprus* Cyprus Pound	CYP	CY	CYP EUR			BIM
Czech Republic Czech Koruna	CZE	CZ	CZK†	BIM	BIM	BIM
Denmark Danish Krone	DNK	DK	DKK†	BIM	BIM	BIM
Djibouti Djiboutian Franc	DJI	DJ	DJF			
Dominica East Caribbean Dollar	DMA	DM	XCD			BIM
Dominican Republic Dominican Peso	DOM	DO	DOP		BIM	BIM
Ecuador Ecuadorian Sucre	ECU	EC	ECD USD		BIM	BIM
Egypt Egyptian Pound	EGY	EG	EGP		BIM	BIM
El Salvador El Salvador Colon	SLV	SV	SVC USD		BIM	BIM
Equatorial Guinea CFA Franc BEAC	GNQ	GQ	XAF			BIM
Eritrea Eritrean Nafka	ERI	ER	ERN			
Estonia* Estonian Kroon	EST	EE	EEK EUR			BIM
Ethiopia Ethiopian Birr	ETH	ET	ETB			BIM
European Monetary Union Euro	EMU	EU	EUR†$	BIM	BIM	BIM
Falkland Islands Falkland Islands Pound	FLK	FK	FKP GBP#			
Faroe Islands Danish Krone	FRO	FO	DKK#			BIM
Fiji Fiji Dollar	FJI	FJ	FJD			BIM
Finland* Finnish Markka	FIN	FI	FIM EUR		BIM	BIM
France* French Franc	FRA	FR	FRF EUR		BIM	BIM

Country Currency	3-letter Country Code	2-letter Country Code	Currency Code	Supported For		
				Currency of Issue	Issuer Country	Price Currency
French Guiana Euro	GUF	GF	EUR			BIM
French Polynesia CFP Franc	PYF	PF	XPF			BIM
French Southern Territories Euro	ATF	TF	EUR			BIM
Gabon CFA Franc BEAC	GAB	GA	XAF			BIM
Gambia Gambian Dalasi	GMB	GM	GMD			BIM
Georgia Georgian Lari	GEO	GE	GEL			BIM
Germany* Deutsche Mark	DEU	DE	DEM EUR		BIM	BIM
Ghana Ghanaian Cedi	GHA	GH	GHS			BIM303 only
Gibraltar Gibraltar Pound	GIB	GI	GIP			
Greece* Greek Drachma	GRC	GR	GRD EUR		BIM	BIM
Greenland Danish Krone	GRL	GL	DKK			BIM
Grenada East Caribbean Dollar	GRD	GD	XCD			BIM
Guadeloupe Euro	GLP	GP	EUR			BIM
Guam US Dollar	GUM	GU	USD			BIM
Guatemala Guatemalan Quetzel	GTM	GT	GTQ			BIM
Guinea Guinean Franc	GIN	GN	GNF			BIM
Guinea-Bissau CFA Franc BCEAO	GNB	GW	XOF			BIM
Guyana Guyanese Dollar	GUY	GY	GYD			
Haiti Haitian Gourde	HTI	HT	HTG USD			BIM
Heard and McDonald Islands Australian Dollar	HMD	HM	AUD			BIM
Honduras Honduran Lempira	HND	HN	HNL			BIM
Hong Kong Hong Kong Dollar	HKG	HK	HKD†	BIM	BIM	BIM
Hungary Hungarian Forint	HUN	HU	HUF	BIM	BIM	BIM
Iceland Icelandic Krona	ISL	IS	ISK	BIM	BIM	BIM
India Indian Rupee	IND	IN	INR	BIM	BIM	BIM
Indonesia Indonesian Rupiah	IDN	ID	IDR	BIM	BIM	BIM

Country Currency	3-letter Country Code	2-letter Country Code	Currency Code	Supported For		
				Currency of Issue	Issuer Country	Price Currency
International Monetary Fund Special Drawing Rights			XDR			BIM
Iran Iranian Rial	IRN	IR	IRR			BIM
Iraq Iraqi Dinar	IRQ	IQ	IQD			
Ireland* Irish Pound	IRL	IE	IEP EUR		BIM	BIM
Israel Israeli New Sheqel	ISR	IL	ILS	BIM	BIM	BIM
Italy* Italian Lira	ITA	IT	ITL EUR		BIM	BIM
Cote D'Ivoire CFA Franc BCEAO	CIV	CI	XOF			BIM303 only
Jamaica Jamaican Dollar	JAM	JM	JMD			BIM303 only
Japan Japanese Yen	JPN	JP	JPY†$	BIM	BIM	BIM
Jersey British Pound Sterling	JEY	JE	GBP#			BIM
Jordan Jordanian Dinar	JOR	JO	JOD			BIM
Kazakhstan Kazakhstani Tenge	KAZ	KZ	KZT			BIM
Kenya Kenyan Shilling	KEN	KE	KES			BIM303 only
Kiribati Australian Dollar	KIR	KI	AUD#			BIM
Korea Korean Won	KOR	KR	KRW†	BIM	BIM	BIM
Kuwait Kuwaiti Dinar	KWT	KW	KWD			BIM
Kyrgyzstan Kyrgyzstani Som	KGZ	KG	KGS			BIM
Laos Lao Kip	LAO	LA	LAK			
Latvia* Latvia Lat	LVA	LV	LVL EUR		BIM	BIM
Lebanon Lebanese Pound	LBN	LB	LBP		BIM303 only	BIM303 only
Lesotho Lesotho Loti	LSO	LS	LSL			BIM
Liberia Liberian Dollar	LBR	LR	LRD			
Libya Libyan Dinar	LBY	LY	LYD			
Liechtenstein Swiss Franc	LIE	LI	CHF			BIM
Lithuania* Lithuania Litas	LTU	LT	LTL EUR		BIM	BIM
Luxembourg* Luxembourg Franc	LUX	LU	LUF EUR			BIM

Country Currency	3-letter Country Code	2-letter Country Code	Currency Code	Supported For		
				Currency of Issue	Issuer Country	Price Currency
Macao Macanese Pataca	MAC	MO	MOP HKD			BIM
Macedonia Macedonian Denar	MKD	MK	MKD			BIM303 only
Madagascar Malagasy Ariary	MDG	MG	MGA			BIM
Malawi Malawian Kwacha	MWI	MW	MWK			BIM
Malaysia Malaysian Ringgit	MYS	MY	MYR†	BIM	BIM	BIM
Maldives Maldivian Rufiyaa	MDV	MV	MVR			BIM
Mali CFA Franc BCEAO	MLI	ML	XOF			BIM
Malta* Maltese Lira	MLT	MT	MTL EUR#			BIM303 only
Marshall Islands US Dollar	MHL	MH	USD			BIM
Martinique Euro	MTQ	MQ	EUR			BIM
Mauritania Mauritanian Ouguiya	MRT	MR	MRO			BIM
Mauritius Mauritian Rupee	MUS	MU	MUR			BIM303 only
Mayotte Euro	MYT	YT	EUR			BIM
Mexico Mexican Peso	MEX	MX	MXN	BIM	BIM	BIM
Micronesia US Dollar	FSM	FM	USD			BIM
Moldova Moldovan Leu	MDA	MD	MDL			BIM
Monaco Euro	MCO	MC	EUR			BIM
Mongolia Mongolian Tugrik	MNG	MN	MNT			BIM
Montenego Euro	MNE	ME	EUR			BIM303 only
Montserrat East Caribbean Dollar	MSR	MS	XCD			BIM
Morocco Moroccan Dirham	MAR	MA	MAD			BIM
Mozambique Mozambican Metical	MOZ	MZ	MZN			BIM
Myanmar Myanma Kyat	MMR	MM	MMK			
Namibia Namibian Dollar	NAM	NA	NAD			BIM303 only
Nauru Australian Dollar	NRU	NR	AUD			BIM
Nepal Nepalese Rupee	NPL	NP	NPR			BIM

Country Currency	3-letter Country Code	2-letter Country Code	Currency Code	Supported For		
				Currency of Issue	Issuer Country	Price Currency
Netherlands* Netherlands Guilder	NLD	NL	NLG EUR		BIM	BIM
Netherlands Antilles Netherlands Antillian Guilder	ANT	AN	ANG			BIM
New Caledonia CFP Franc	NCL	NC	XPF			BIM
New Zealand New Zealand Dollar	NZL	NZ	NZD†$	BIM	BIM	BIM
Nicaragua Nicaragua Cordoba	NIC	NI	NIO			BIM
Niger CFA Franc BCEAO	NER	NE	XOF			BIM
Nigeria Nigerian Naira	NGA	NG	NGN			BIM
Nieu New Zealand Dollar	NIU	NU	NZD			BIM
Norfolk Island Australian Dollar	NFK	NF	AUD			BIM
North Korea North Korean Won	PRK	KP	KPW			
Northern Mariana Islands US Dollar	MNP	MP	USD			BIM
Norway Norwegian Krone	NOR	NO	NOK†	BIM	BIM	BIM
Oman Omani Rial	OMN	OM	OMR			BIM
Pakistan Pakistani Rupee	PAK	PK	PKR			BIM
Palau US Dollar	PLW	PW	USD			BIM
Palestine Israeli New Sheqel	PSE	PS	ILS			BIM303 only
Panama Panamanian Balboa	PAN	PA	PAB USD		BIM	BIM
Papua New Guinea Papua New Guinean Kina	PNG	PG	PGK			BIM
Paraguay Paraguayan Guarani	PRY	PY	PYG			BIM
Peru Peruvian Nuevo Sol	PER	PE	PEN	BIM	BIM	BIM
Philippines Philippine Peso	PHL	PH	PHP†	BIM	BIM	BIM
Pitcairn New Zealand Dollar	PCN	PN	NZD			BIM
Poland Polish Zloty	POL	PL	PLN†	BIM	BIM	BIM
Portugal* Portuguese Escudo	PRT	PT	PTE EUR		BIM	BIM
Puerto Rico US Dollar	PRI	PR	USD			BIM
Qatar Qatari Rial	QAT	QA	QAR			BIM

Country Currency	3-letter Country Code	2-letter Country Code	Currency Code	Supported For		
				Currency of Issue	Issuer Country	Price Currency
Reunion Euro	REU	RE	EUR			BIM
Romania Romanian New Leu	ROU	RO	RON		BIM	BIM
Russia Russian Ruble	RUS	RU	RUB		BIM	BIM
Rwanda Rwandan Franc	RWA	RW	RWF			BIM
Saint Helena, Ascenion and Tristan da Cunha Saint Helena Pound	SHN	SH	SHP GBP#			
Saint Kitts and Nevis East Caribbean Dollar	KNA	KN	XCD			BIM
Saint Lucia East Caribbean Dollar	LCA	LC	XCD			BIM
Saint Pierre and Miquelon Euro	SPM	PM	EUR			BIM
Saint Vincent and the Grenadines East Caribbean Dollar	VCT	VC	XCD			BIM
Samoa Samoan Tala	WSM	WS	WST			BIM
San Marino Euro	SMR	SM	EUR			BIM
Sao Tome and Principe Sao Tome and Principe Dobra	STP	ST	STD			
Saudi Arabia Saudi Riyal	SAU	SA	SAR			BIM
Senegal CFA Franc BCEAO	SEN	SN	XOF			BIM
Serbia Serbian Dinar	SRB	RS	RSD		BIM	BIM
Seychelles Seychelles Rupee	SYC	SC	SCR			BIM
Sierra Leone Sierra Leonean Leone	SLE	SL	SLL			BIM
Singapore Singapore Dollar	SGP	SG	SGD†	BIM	BIM	BIM
Slovakia* Slovak Koruna	SVK	SK	SKK EUR		BIM	BIM
Slovenia* Slovenian Tolar	SVN	SI	SIT EUR		BIM	BIM
Solomon Islands Solomon Islands Dollar	SLB	SB	SBD			BIM
Somalia Somali Shilling	SOM	SO	SOS			
South Africa South African Rand	ZAF	ZA	ZAR†$	BIM	BIM	BIM
Spain* Spanish Peseta	ESP	ES	ESP EUR		BIM	BIM
Sri Lanka Sri Lankan Rupee	LKA	LK	LKR			BIM
Sudan Sudanese Pound	SDN	SD	SDG			BIM

Country Currency	3-letter Country Code	2-letter Country Code	Currency Code	Supported For		
				Currency of Issue	Issuer Country	Price Currency
Suriname Surinamese Dollar	SUR	SR	SRD			BIM
Svalbard and Jan Mayen Norwegian Krone	SJM	SJ	NOK			BIM
Swaziland Swazi Lilangeni	SWZ	SZ	SZL			BIM
Sweden Swedish Krona	SWE	SE	SEK†$	BIM	BIM	BIM
Switzerland Swiss Franc	CHE	CH	CHF†	BIM	BIM	BIM
Syria Syrian Pound	SYR	SY	SYP			
Taiwan New Taiwan Dollar	TWN	TW	TWD†	BIM	BIM	BIM
Tajikistan Tajikistani Somoni	TJK	TJ	TJS			
Tanzania Tanzanian Shilling	TZA	TZ	TZS			BIM
Thailand Thai Baht	THA	TH	THB	BIM	BIM	BIM
Togo CFA Franc BCEAO	TGO	TG	XOF			BIM
Tokelau New Zealand Dollar	TKL	TK	NZD			BIM
Tonga Tongan Pa'anga	TON	TO	TOP			BIM
Trinidad and Tobago Trinidad & Tobago Dollar	TTO	TT	TTD			BIM303 only
Tunisia Tunisian Dinar	TUN	TN	TND		BIM303 only	BIM303 only
Turkey Turkish Lira	TUR	TR	TRL	BIM	BIM	BIM
Turkmenistan Turkmenistani Manat	TKM	TM	TMT			
Turks and Caicos Islands US Dollar	TCA	TC	USD			BIM
Tuvalu Australian Dollar	TUV	TV	AUD#			BIM
Uganda Ugandan Shilling	UGA	UG	UGX			BIM
Ukraine Ukrainian Hryvnia	UKR	UA	UAH		BIM	BIM
UAE United Arab Emirates Dirham	ARE	AE	AED			BIM
United Kingdom British Pound Sterling	GBR	GB	GBP†$	BIM	BIM	BIM
United States US Dollar	USA	US	USD†$	BIM	BIM	BIM
United States Minor Outlying Islands US Dollar	UMI	UM	USD			BIM
Uruguay Uruguayan Peso	URY	UY	UYU		BIM	BIM

Country Currency	3-letter Country Code	2-letter Country Code	Currency Code	Supported For		
				Currency of Issue	Issuer Country	Price Currency
Uzbekistan Uzbekistan Som	UZB	UZ	UZS			BIM
Vanuatu Vanuatu Vatu	VUT	VU	VUV			BIM
Vatican City State Euro	VAT	VA	EUR			BIM
Venezuela Venezuelan Bolivar	VEN	VE	VEF		BIM303 only	BIM303 only
Viet Nam Vietnamese Dong	VNM	VN	VND			BIM303 only
Virgin Islands, British US Dollar	VGB	VG	USD			BIM
Virgin Islands, U.S. US Dollar	VIR	VI	USD			BIM
Wallis and Fortuna CFP Franc	WLF	WF	XPF			BIM
Western Sahara	ESH	EH	XXX			
Yemen Yemeni Rial	YEM	YE	YER			BIM
Zambia Zambian Kwacha	ZMB	ZM	ZMK			BIM303 only
Zimbabwe Zimbabwe Dollar	ZWE	ZW	ZWD USD			
World Organization	WLD		XXX			
Gold Currency			XAU			BIM
Palladium Currency			XPD			BIM
Platinum Currency			XPT			BIM
Silver Currency			XAG			BIM

▻ **Key:**

* Member of the European Monetary Union

† Supported for convertible bonds

$ Supported for inflation-protected bonds

Pegged currency at 1:1

Note: "XXX" in a Currency or Country field indicates that an asset has exposures to multiple currencies or countries, respectively.

Appendix B — Asset ID Types

BarraOne supports a wide range of asset IDs.

Note: To use local exchange tickers in BarraOne, you must prepend the two-character ISO country code. For example, IBM would become USIBM. This longer ticker is called a LOCALID in BarraOne. Country codes are listed in "Appendix A — Country and Currency Codes" on page 85.

Table 77: Asset ID Types

Asset ID Type	Notes
ALP_CODE	
ASX	
AUSTRALIAN	for ASX
BARRAID	
BELGIAN	
BloombergID	
BRIDGE	
CBOT	
CEDEL	
CFR_AssetID	
CHINA_INT	China Interbank Market. This ID Type is also used for China Private Placement Bonds (PPN) with Shanghai Clearing House codes (SHC).
CIDEID	
CINS	
CLIP27	
CLIP9	CLIP code that maps to the 5Y version of a Markit-supplied CDS index
CLIPEXT	Extension to the CLIP code that includes the tenor of the contract (e.g., "-10Y" for the 10-year version of the Markit-supplied CDS index, or "-5Y" for the 5-year version of the index
CODE	
CODE_CN	
CODE_GB	
CODE_HK	
CODE_JP	
CODE_KR	
CODE_TW	
COMMON_CODE	for COMMON

Table 77: Asset ID Types (Continued)

Asset ID Type	Notes
COMPOSITEID	for composite assets
CUSIP	8 and 9 digits
DEFAULT	
EUROCLEAR	
EXTEL	for IDC Exbond ID
FONDSKODE	
GERMAN	for WPKN - Wertpapier — fixed income only
HKSE	
ISIN	
JAPAN	for Nikko ID
JAPAN_SICC	
JPM	for JP Morgan
LIN	Reuters loan identification number for syndicated loans
LOANXID	Markit identifier for syndicated loans
LOCALID	local ticker prefixed by two-character country code— equity only
MGCUSIP	
MTBCUSIP	for TBA mortgage-backed securities (MBS) — refer to "Specifying the Underlying ID for TBA MBS" on page 97
NZSE	
RIC	
RSA	for Bond Exchange of South Africa
SEDOL	6 and 7 digits
SHANGHAI	
SHENZEN	
SSB	for Salomon Smith Barney
SWISS	for Valoren
TICKER_AT	
TICKER_BE	
TICKER_BR	
TICKER_CA	
TICKER_DE	
TICKER_ES	
TICKER_EU	
TICKER_FI	

Table 77: Asset ID Types (Continued)

Asset ID Type	Notes
TICKER_FR	
TICKER_GR	
TICKER_ID	
TICKER_IE	
TICKER_IT	
TICKER_LU	
TICKER_MX	
TICKER_NL	
TICKER_NZ	
TICKER_PT	
TICKER_SG	
TICKER_SI	
TICKER_SK	
TICKER_TH	
TICKER_US	
Ticker	
TRADEID	

Specifying the Underlying ID for TBA MBS

Table 78: Underlying ID for TBA MBS

Format:	character 1: MBS Type character 2–3: Pool Code character 4–5: TB (for TBA MBS) character 6–8: Coupon Code
MBS Type	=MBS: Mortgage Backed Security +PO: Principal Only –IO: Interest Only
Pool Code	For choices, see MBS Pool Codes below.
Coupon Code	Coupon expressed as follows: character 6–7: integer portion of coupon character 8: decimal portion of coupon expressed in eighths; e.g., 4 = 4/8 = 0.5
Example:	"=01TB064" means TBA MBS GNMA I 30-yr Single Family, 6.5% net coupon.

MBS Pool Codes

▷ **Note:** Only a subset of these codes is supported on any given analysis date.

Table 79: MBS Pool Codes

Pool Code	Description
01	GNMA I 30-yr Single Family
02	Graduated Payment Plan 1
03	Graduated Payment Plan 2
04	Graduated Payment Plan 3
05	Mobile Home Type 1 – single width
06	Mobile Home Type B – single width
07	Mobile Home Type C
08	Mobile Home Type D
09	Project Loans
10	FNMA 30-yr Pass-through
11	Privately Issued 30-yr Pass-through
12	GNMA 15-yr Single Family
13	FHLMC 30-yr Swap PC
14	FHLMC 15-yr Swap PC
15	FNMA 15-yr Pass-through
16	Privately Issued 15-yr Pass-through
17	FHA/VA FNMA 30-year Pass-through
18	FHLMC 30-yr non-Gold
19	FHLMC 15-yr non-Gold
20	FHA/VA FHLMC 30-yr non-Gold
21	FHA/VA FHLMC 30-yr Cash
22	GNMA II 30-yr Single Family
23	Graduated Payment GNMA II
24	Mobile Home GNMA II
25	Buy Down
26	Construction Loan
27	FNMA 20-yr
28	FHLMC 20-yr
29	FNMA 7-yr Balloon MEGA
30	FNMA 15-yr Balloon MEGA
31	FNMA 7-yr Balloon
32	FHLMC 5-yr Balloon

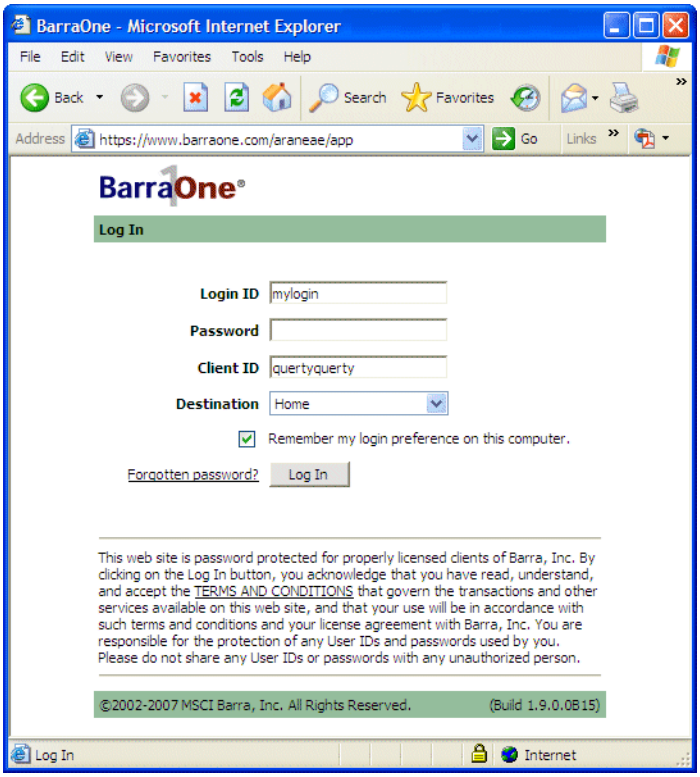
Table 79: MBS Pool Codes

Pool Code	Description
33	FHLMC 7-yr Balloon
34	FNMA 10-yr Balloon
35	FNMA 15-yr Balloon
36	FNMA 15-yr MEGA
37	FNMA 20-yr MEGA
38	FNMA 30-yr MEGA
39	GNMA I 30-yr Single Family Platinum
40	FHLMC 15-yr Gold
41	FHLMC 30-yr Gold
42	FHLMC 5-yr Balloon Gold
43	FHLMC 7-yr Balloon Gold
44	FHLMC 15-yr Gold Giant
45	FHLMC 20-yr Gold Giant
46	FHLMC 30-yr Gold Giant
47	FHLMC 5-yr Balloon Giant
48	FHLMC 7-yr Balloon Giant
49	GNMA 15-yr Single Family Platinum
50	FNMA 5-yr Balloon
51	FHA/VA FHLMC 30-yr Gold
52	FHA/VA FHLMC 30-yr Gold (Mini)
53	GNMA II 15-yr Single Family
60	GNMA II 30-yr Single Family Platinum

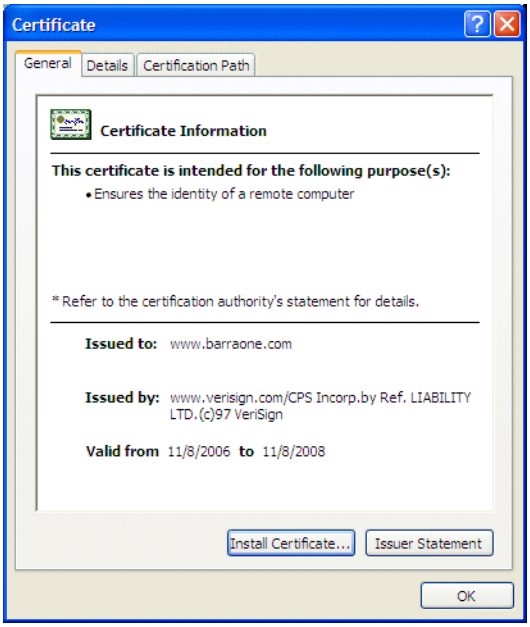
Appendix C — Exporting the BarraOne Certificate

To export the BarraOne certificate, follow these steps:

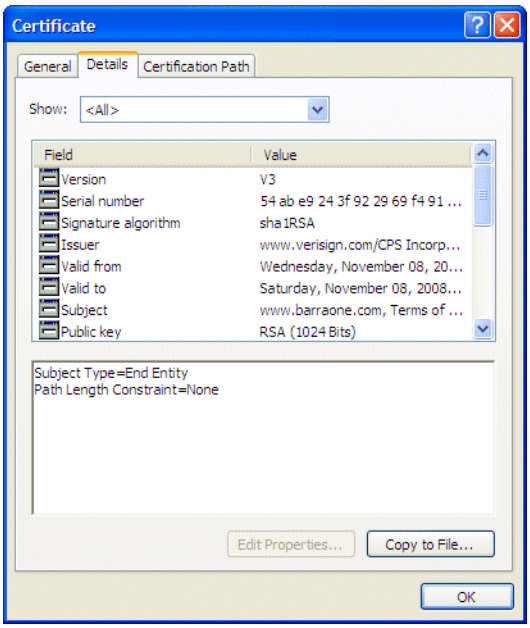
1 Launch Internet Explorer, and navigate to the BarraOne login page.



2 On the BarraOne login page, double-click the yellow lock icon ( ) in the bottom panel of the browser window. The **Certificate** window appears.

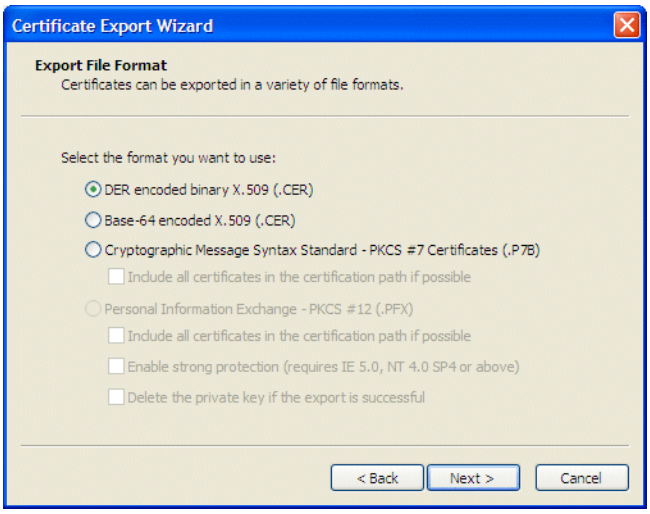


3 In the **Certificate** window, click the **Details** tab.

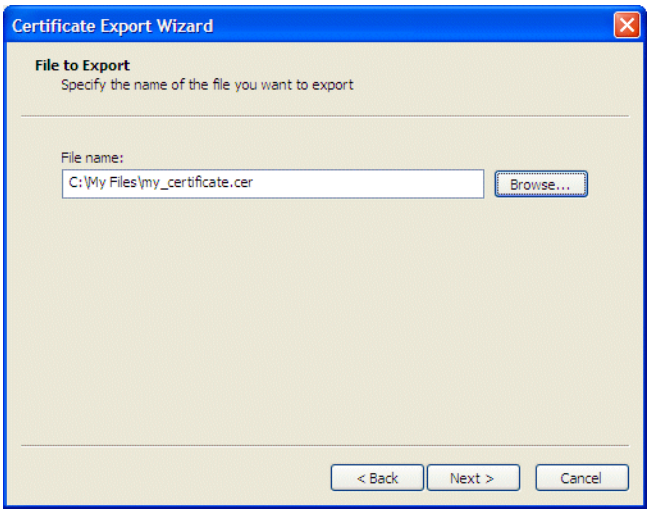


4 On the **Details** tab of the **Certificate** window, click the **Copy to File...** button. The **Certificate Export Wizard** launches.

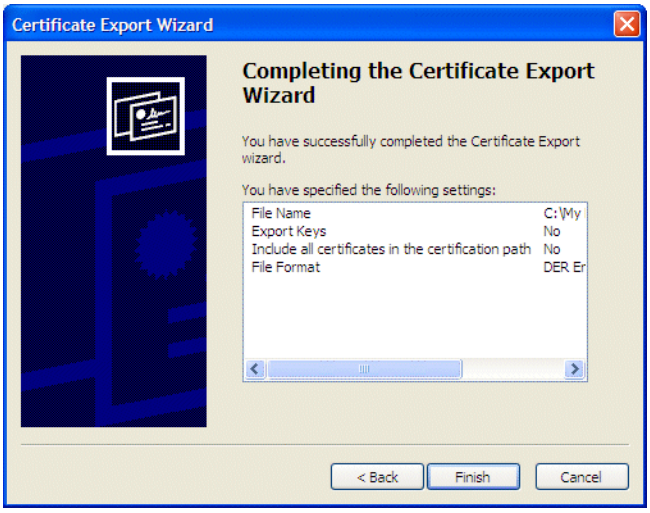
5 Click the **Next** button to open the next page of the **Certificate Export Wizard**.



6 Accept the default selection, and click the **Next** button to open the next page of the **Certificate Export Wizard**.



7 Enter the path and file to which you would like to export the BarraOne certificate, and then click the **Next** button to open the next page of the **Certificate Export Wizard**.



8 Click the **Finish** button to complete the export of the certificate. A dialog appears indicating success.

9 Click the **OK** button to exit the **Certificate Export Wizard** and return to the **Certificate** window.

10 Click the **OK** button in the **Certificate** window to close it.

Appendix D — Legacy Elements

The following elements have been replaced with other elements with more flexibility, but they have bee retained to support backward compatibility. All of these elements will be discontinued in the next release.

GetPortfolioTreeFilters

The GetPortfolioTreeFilters is a legacy element enables users to get a list of available portfolio tree filters in BarraOne (a filter is the equivalent of an item (such as owner or workgroup) in the dropdown list available in the Select Portfolio dialog in the BarraOne application). The signature of this element has no arguments. It will return a String array. The response can then be used in the element "GetPortfolioTree" on page 9. This element has been supplanted with "GetFilters" on page 8.

To create a GetPortfolioTreeFilters object, send the following:

```
List<String> myFilters = myPort.getPortfolioTreeFilters();
```

In .NET, send the following:

```
string[] myFilters = bdtiClient.GetPortfolioTreeFilters();
```

Filter

Filter is the sole element of the String array returned by the GetPortfolioTreeFilters element.

Table 80: Filter (1-n)

Attribute	Required?	Data Type (Length)	Description
Filter	Y	string	The name of the filter. A filter is the equivalent of an item (such as owner or workgroup) in the dropdown list available in the Select Portfolio dialog in the BarraOne application.

GetColumnNames

The GetColumnNames element is a legacy element that enables users to get a list of column names available in the BarraOne Positions Report. It has one child element, Owner, as indicated below. It will return a String array. The information returned can then be used in the elements "GetGroupings" on page 15, "GetPositionsReport" on page 48, and "SubmitPositionsReport" on page 56. This element has been supplanted with "GetReportColumnName" on page 14.

Owner

Owner is the sole child element of the GetColumnNames element.

Table 81: Owner (1)

Attribute	Required?	Data Type (Length)	Description
Owner	Y	string	Enter the User ID of the column owner. Enter “SYSTEM unless it is a user-created column.

To create a GetColumnNames object, send the following:

```
List<String> myColumns = myPort.getColumnNames("myOwner");
```

In .NET, send the following:

```
string[] myColumns = bdtiClient.GetColumnNames("myOwner");
```

ColumnName

ColumnName is the sole element of the String array returned in response to the GetColumnNames element.

Table 82: ColumnName (1-n)

Attribute	Required?	Data Type (Length)	Description
ColumnName	Y	string	A column name available in the report.

GetModelFilters

GetModelFilters is a legacy element that enables users to get a list of available model filters in BarraOne (a filter is the equivalent of an item (such as owner or workgroup) in the dropdown list available in the Select Model dialog in the BarraOne application). The signature of this element has no arguments. It will return a String array. The results of this operation can then be used in “GetModels” on page 36. This element has been supplanted with “GetFilters” on page 8.

To create a GetModelFilters object, send the following:

```
List<String> myFilters = myPort.getModelFilters();
```

In .NET, send the following:

```
string[] myFilters = bdtiClient.GetModelFilters();
```

Filter

Filter is the sole element of the String array returned by the GetModelFilters element.

Table 83: Filter (1-n)

Attribute	Required?	Data Type (Length)	Description
Filter	Y	string	The name of the filter. A filter is the equivalent of an item (such as owner or workgroup) in the dropdown list available in the Select Model dialog in the BarraOne application.

RetrievePositionsReport

RetrievePositionsReport is a legacy element that enables users to retrieve a Positions Report in BDTI. It has one child element, RequestID. It will return a ReportResponse (refer to"ReportResponse" on page 82). This element has been supplanted with "RetrieveReport" on page 82.

Table 84: RequestID (1)

Attribute	Required?	Data Type (Length)	Description
RequestID	Y	string	Enter the RequestID for the portfolio to be retrieved. Refer to "SubmitPositionsReport" on page 56.

To create a RetrievePositionsReport object, send the following:

```
ReportResponse myResponse = port.retrievePositionsReport("myRequestID");
```

In .NET, send the following:

```
ReportResponse myResponse =
  bdtiClient.RetrievePositionsReport("myRequestID");
```