



BusinessObjects™ XI Release 2 Integration Kit for SAP® Java SDK Guide

Patents

Business Objects owns the following U.S. patents, which may cover products that are offered and sold by Business Objects: 5,555,403, 6,247,008 B1, 6,578,027 B2, 6,490,593 and 6,289,352.

Trademarks

Business Objects, the Business Objects logo, Crystal Reports, and Crystal Enterprise are trademarks or registered trademarks of Business Objects SA or its affiliated companies in the United States and other countries. All other names mentioned herein may be trademarks of their respective owners.

Copyright

Copyright © 2005 Business Objects. All rights reserved.

Third-party contributors

Business Objects products in this release may contain redistributions of software licensed from third-party contributors. Some of these individual components may also be available under alternative licenses. A partial listing of third-party contributors that have requested or permitted acknowledgments, as well as required notices, can be found at:

<http://www.businessobjects.com/thirdparty>

Contents

Chapter 1	Introduction	7
	About this guide	8
	Who should read this guide	8
	Business Objects information resources	8
Chapter 2	Architecture	9
	BusinessObjects XI Release 2 Integration Kit for SAP Framework	10
	Crystal SAP Web Parameter Package	13
	ParameterUtilityFactory Class	14
	getParameterUtility Method	14
	IPParameterUtility Interface	15
	deletePersonalizedValue Method	16
	getPicklistGenerator Method	17
	getReport Method	19
	initialize Method	20
	isBWParameter Method	21
	isHierarchyNodeParameter Method	22
	loadDefaultValueToReport Method	22
	loadPersonalizedValueToReport Method	23
	savePersonalizedValue Method	24
	validateHierarchicalParameters Method	25
	IPicklistGenerator Interface	26
	get Method	26
	size Method	27
	IPicklistEntry Interface	28
	getDescription Method	29
	getValue Method	29
	isForDisplay Method	29
	setDescription Method	30

Contents

setForDisplay Method	30
setValue Method	30
ITreeNode Interface	31
getChildAt Method	31
getChildren Method	32
getChildrenCount Method	32
getParent Method	32
isChildrenAvailable Method	33
ParameterException Class	34
Subclasses	35
HierarchyMissingException Class	35
HierarchyNodeException Class	36
InvalidParameterIndexException Class	36
LoadDefaultValuesAbortedException Class	36
NotInitializedException Class	37
PersonalizationException Class	37
Methods	37
getCauseParameterIndex Method	37
getSourceParameterIndex Method	38
Crystal SAP Web RRI Package	39
RRI Class	40
getRRIReceiver Method	40
IRRIReceiver Interface	41
get Method	41
initialize Method	42
size Method	43
IRRIEntry Interface	44
getHigh Method	44
getLow Method	45
getName Method	45
getOpt Method	45
getSign Method	46
RRIException Class	47

Contents

InvalidParameterIndexException Class	47
NotInitializedException Class	47
UnableToInitializeException Class	48
Crystal SAP Web SSO Package	49
SSO Class	50
getDatasourceSSO Method	50
IDatasourceSSO Interface	51
getDatabaseName Method	52
getDataSourceName Method	52
getExternalID Method	53
getLogonCount Method	53
getPassword Method	54
getServerName Method	54
getSubReport Method	54
getUsername Method	55
isOfflineSNCName Method	55
isSNCAvailable Method	55
setBufferWidth Method	56
setCMSLogonToken Method	56
setDataSourceIndex Method	56
setEnterpriseSession Method	57
setPassword Method	57
setReportID Method	57
setScheduleTime Method	58
Appendix A Get More Help	59
Index	63

Contents



Introduction



1
chapter

About this guide

The BusinessObjects XI Release 2 Integration Kit for SAP Java Software Development Kit (SDK) provides you with the necessary tools for building BusinessObjects XI Release 2 Integration Kit for SAP web applications. With this SDK, you can build web applications that take advantage of the integration between an SAP BW system and a BusinessObjects Enterprise system. It provides you with the ability to use SAP BW variables that appear as parameters in Crystal Reports. It also allows you to build applications that use Crystal Reports as Report-Report-Interface (RRI) receivers. Moreover, the SDK allows you to build applications that use Secure Network Communication (SNC) and password-less Single Sign On (SSO).

Who should read this guide

This guide help has been written with the web developer in mind. In particular, it is aimed towards those who are responsible for building custom web applications, and those who have a clear understanding of Java and Java Server Pages.

BusinessObjects Enterprise web applications are written as Java Server Pages (JSP). If you are a web developer, you will find that Java Server Pages offer the same flexibility as Microsoft's Active Server Pages (ASP). Like ASP, JSP is a server-side scripting environment in which you can combine HTML pages, scripting languages, and Java objects to create interactive web applications. For more information on using JSP pages, consult the Sun Java website, <http://www.java.sun.com>.

BusinessObjects Enterprise web applications can be developed in any environment that supports JSP, and they can be run on multiple platforms.

Business Objects information resources

For more information and assistance, see [Get More Help](#). This appendix describes the Business Objects documentation, customer support, training, and consulting services, with links to online resources.



Architecture



2
chapter

BusinessObjects XI Release 2 Integration Kit for SAP Framework

The BusinessObjects XI Release 2 Integration Kit for SAP SDK is an extension of the BusinessObjects Enterprise framework SDK. It builds on the functionality provided by BusinessObjects Enterprise and allows you to use the BusinessObjects XI Release 2 Integration Kit for SAP SDK to communicate with, and exchange information between, BusinessObjects Enterprise and SAP systems.

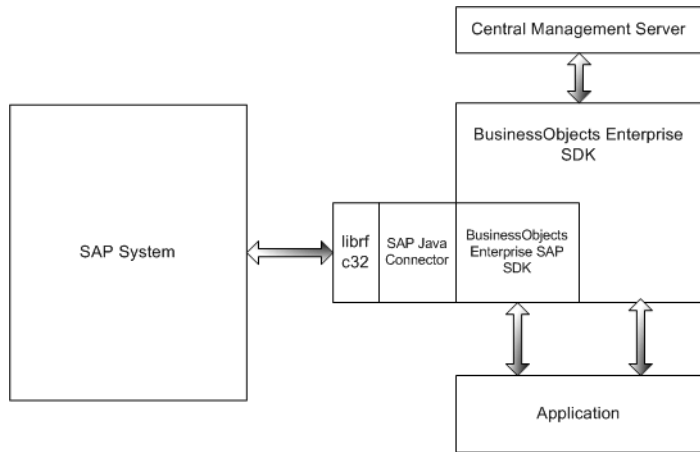
Direct communication with the SAP system is handled by the SAP connectivity library, `librfc32`. This component of the framework is provided by SAP and must be installed on top of your BusinessObjects Enterprise System. For more information on how to install the `librfc32`, see the *BusinessObjects XI Release 2 Integration Kit for SAP Installation Guide* (`installguide.pdf`), included in the `docs` directory on both Add-Ons Discs.

Another component of the framework is the Central Management Server (CMS). The CMS has four main functions:

- maintains security
- manages objects
- manages servers
- manages auditing.

However, with regard to the BusinessObjects XI Release 2 Integration Kit for SAP SDK, the CMS primarily handles security. For more information about security and authentication, see the *BusinessObjects XI Release 2 Integration Kit for SAP User's Guide* (`userguide.pdf`), included in the `docs` directory on both Add-Ons Discs.

The following diagram illustrates the information flow between the components involved in the BusinessObjects XI Release 2 Integration Kit for SAP Framework:



Crystal SAP Web Parameter Package

The Crystal SAP Web Parameter Package contains four interfaces and two classes: [IPParameterUtility Interface](#), [IPicklistEntry Interface](#), [IPicklistGenerator Interface](#), [ITreeNode Interface](#), [ParameterUtilityFactory Class](#), and [ParameterException Class](#).

This package provides you with the ability to use the SAP BW variables that appear as parameters in Crystal Reports. Because of the potentially changing and user-specific nature of pick list values for SAP BW variables, dynamic pick list functionality is supplied. However, if for any reason it is not possible to fill the pick lists dynamically, the parameter values that are stored with the InfoObject for the specified parameter are returned as an alternative.

A dynamic pick list provides you with a current list of possible values to choose from, for parameters that represent SAP BW variables. The values displayed in a dynamic pick list are retrieved from the SAP server and correspond to the user's rights associated with the query variable.

The entry point for the IPParameterUtility Interface is the ParameterUtilityFactory Class which creates an instance of the IPParameterUtility Interface. From this point you initialize the parameter helper; you can then use the associated methods within the IPParameterUtility Interface.

Furthermore, this package also allows you to load default and personalized SAP BW parameters into a report, save personalized SAP BW parameters into a report, and verify that the hierarchy and the hierarchy node relationship is valid.

Package Name

com.crystaldecisions.sap.web.parameter

ParameterUtilityFactory Class

See Also

[IPParameterUtility Interface](#)
[IPicklistEntry Interface](#)
[IPicklistGenerator Interface](#)
[ITreeNode Interface](#)
[ParameterException Class](#)

Syntax

```
public class ParameterUtilityFactory
```

Remarks

The ParameterUtilityFactory Class is the starting point for using the Parameter package. This class allows you to retrieve an instance of the IPParameterUtility object.

Methods

Method	Description
getParameterUtility Method	Returns an instance of an IPParameterUtility object.

getParameterUtility Method

Applies To

[ParameterUtilityFactory Class](#)

Description

Returns an instance of an IPParameterUtility object.

Syntax

```
public static IPParameterUtility getParameterUtility();
```

Returns

A IPParameterUtility object.

IPParameterUtility Interface

See Also

- [IPicklistEntry Interface](#)
- [IPicklistGenerator Interface](#)
- [ITreeNode Interface](#)
- [ParameterUtilityFactory Class](#)
- [ParameterException Class](#)

Syntax

```
public interface IPParameterUtility
```

Remarks

The IPParameterUtility Interface provides a number of methods that allow you to retrieve specific parameter information for a Report object. However, you must first initialize the parameter helper object. This is achieved by calling the [initialize Method](#).

Methods

Method	Description
deletePersonalizedValue Method	Deletes the personalized value for the specified parameter.
getPicklistGenerator Method	Returns the pick list generator for the specified parameter.
getReport Method	Returns the Report object that initialized the IPParameterUtility object.
initialize Method	Initializes the parameter helper with the specified report and logon credentials.
isBWParameter Method	Returns a boolean that indicates whether the parameter is a SAP BW parameter.
isHierarchyNodeParameter Method	Returns a boolean that indicates whether the specified parameter is a hierarchy node parameter.
loadDefaultValueToReport Method	Loads the BW default variable value into the report as the current value of the parameter, and then returns it.
loadPersonalizedValueToReport Method	Loads the BW personalized value into the report as the current value of the parameter, and then returns it.

Method	Description
savePersonalizedValue Method	Sets the personalized value(s) as the current value(s) for the specified parameter in the report object.
validateHierarchicalParameters Method	Raises an exception on the first invalid hierarchy and hierarchy node parameter detected.

deletePersonalizedValue Method

Applies To

[IParameterUtility Interface](#)

See Also

[ParameterException Class](#)

Description

Deletes the personalized value for the specified parameter.

Syntax

```
public void deletePersonalizedValue(  
    int paramNumber) throws ParameterException;
```

Parameters

paramNumber

An int that specifies the zero-based index of the parameter. This parameter matches the index that is used to access the parameter values returned by the `getReportParameters` Method. (The `getReportParameters` Method is a member of the `IReportProcessingInfo` Class in the *BusinessObjects Enterprise Java SDK*.)

Throws

`ParameterException`

This is thrown if the process is unsuccessful.

Remarks

Throws one of the following parameter exceptions:

- `NotInitializedException`
- `InvalidParameterIndexException`

getPicklistGenerator Method

Applies To

[IParameterUtility Interface](#)

See Also

[ParameterException Class](#)

Description

Returns the pick list generator for the specified parameter.

Syntax

```
public IPicklistGenerator getPicklistGenerator(  
    int paramNumber,  
    boolean returnAsFlattened) throws ParameterException;
```

Parameters

paramNumber

An int that specifies the zero-based index of the parameter. This parameter matches the index that is used to access the parameter values returned by the `getReportParameters` Method. (The `getReportParameters` Method is a member of the `IReportProcessingInfo` Class in the *BusinessObjects EnterpriseJava SDK*.)

returnAsFlattened

A boolean that indicates whether the nodes are returned in a flat structure or in a hierarchical tree structure. This parameter affects variables that are hierarchy nodes. Variables that are not hierarchy nodes are always returned in a flat structure.

true - the nodes are returned in a flat structure, regardless of the parent-child relationship.

false - the nodes are returned in a hierarchical tree structure and they adhere to the parent-child relationship.

Returns

A `IPicklistGenerator` object for the provided parameter.

Throws

`ParameterException`

This is thrown if the process is unsuccessful.

Remarks

A pick-list generator creates a list of values for use as parameters for the report. Furthermore, if the parameter you specify is a hierarchy node variable, and this parameter is associated with a hierarchy variable, then the pick-list generator and its values change depending on the hierarchy variable's current value. This behavior is called cascading parameters.

The following example shows a hierarchical structure for two distinct sales regions. The hierarchy variable represents a country and the hierarchy node variable represents the corresponding territorial information for that country:

Canada

- Ontario
 - Ottawa
 - Toronto
- British Columbia
 - Victoria
 - Vancouver

United States

- California
 - Los Angeles
 - San Francisco
- New York
 - Buffalo
 - New York City

If you set the hierarchy variable to Canada and call the [getPicklistGenerator Method](#) for the hierarchy node variable, you are returned the following pick list:

Canada

- Ontario
 - Ottawa
 - Toronto
- British Columbia
 - Victoria
 - Vancouver

If you switch the hierarchy variable to the United States and call the [getPicklistGenerator Method](#) for the hierarchy node variable, you are returned the following pick list:

United States

California
 Los Angeles
 San Francisco
New York
 Buffalo
 New York City

Note:

- A flat structure indicates that the nodes are represented on the same level.
- If for any reason it is not possible to fill the pick lists dynamically, the parameter values that are stored with the InfoObject for the specified parameter are returned as an alternative.
- Throws one of the following parameter exceptions:
 - `HierarchyMissingException`
 - `HierarchyNodeException`
 - `NotInitializedException`
 - `InvalidParameterException`
- Throws the exception when a pick list cannot be generated for the hierarchy node because the corresponding hierarchy has not been selected.
- If the dynamic pick list cannot be retrieved from the SAP system, the parameter values that are stored with the published Crystal Report will be returned as an alternative.

getReport Method**Applies To**

[IParameterUtility Interface](#)

See Also

[ParameterException Class](#)

Description

Returns the Report object that initialized the `IParameterUtility` object.

Syntax

```
public IReport getReport() throws ParameterException;
```

Returns

An IReport object that specifies the report that initialized the IParameterUtility object.

Throws

ParameterException

This is thrown if the process is unsuccessful.

initialize Method

Applies To

[IParameterUtility Interface](#)

See Also

[ParameterException Class](#)

Description

Initializes the parameter helper with the provided report and logon credentials.

Syntax

```
public void initialize(  
    IReport reportObject,  
    String cmsLogonToken,  
    String bwUser,  
    String bwPassword,  
    String bwTicket) throws  
    com.crystaldecisions.sdk.exception.SDKException;
```

Parameters

reportObject

An IReport that specifies the report object.

cmsLogonToken

A string that specifies the CMS logon token.

bwUser

A string that specifies the BW user.

bwPassword

A string that specifies the BW password.

bwTicket

A string that specifies the BW logon ticket.

Throws

`com.crystaldecisions.sdk.exception.SDKException`

This is thrown if the process is unsuccessful.

Remarks

To log on to the system you must provide one of the following authentications:

- BW logon ticket
- BW user and BW password

If for any reason it is not possible to fill the pick lists dynamically, the parameter values that are stored with the InfoObject for the specified parameter are returned as an alternative.

isBWParameter Method

Applies To

[IParameterUtility Interface](#)

See Also

[ParameterException Class](#)

Description

Returns a boolean that indicates whether the parameter is a SAP BW parameter.

Syntax

```
public boolean isBWParameter(  
    int paramNumber) throws ParameterException;
```

Parameters

`paramNumber`

An int that specifies the zero-based index of the parameter. This parameter matches the index that is used to access the parameter values returned by the `getReportParameters` Method. (The `getReportParameters` Method is a member of the `IReportProcessingInfo` Class in the *BusinessObjects Enterprise Java SDK*.)

Returns

`true` if the specified parameter is a SAP BW parameter; `false` if the specified parameter is not a SAP BW parameter.

Throws

`ParameterException`

This is thrown if the process is unsuccessful.

isHierarchyNodeParameter Method

Applies To

[IPicklistGenerator Interface](#)

Description

Returns a boolean that indicates whether the specified parameter is a hierarchy node parameter.

Syntax

```
public boolean isHierarchyNodeParameter(  
    int paramNumber) throws ParameterException;
```

Parameters

paramNumber

An int that specifies the zero-based index of the parameter. This parameter matches the index that is used to access the parameter values returned by the `getReportParameters` Method. (The `getReportParameters` Method is a member of the `IReportProcessingInfo` Class in the *BusinessObjects Enterprise Java SDK*.)

Returns

true if the specified parameter is a hierarchy node parameter; false if the specified parameter is not a hierarchy node parameter.

Throws

ParameterException

This is thrown if the process is unsuccessful.

loadDefaultValueToReport Method

Applies To

[IParameterUtility Interface](#)

See Also

[ParameterException Class](#)

Description

Loads the BW default variable value into the report as the current value of the parameter, and then returns it.

Syntax

```
public IReportParameterValue loadDefaultValueToReport(  
    int paramNumber) throws ParameterException;
```

Parameters

paramNumber

An int that specifies the zero-based index of the parameter that you want to load the default value for. This parameter matches the index that is used to access the parameter values returned by the `getReportParameters` Method. (The `getReportParameters` Method is a member of the `IReportProcessingInfo` Class in the *BusinessObjects Enterprise Java SDK*.)

Returns

An `IReportParameterValue` object that contains the default value.

Throws

ParameterException

This is thrown if the process is unsuccessful.

Remarks

Returns null if the BW default variable does not exist.

Note: Throws one of the following parameter exceptions:

- `HierarchyMissingException`
- `NotInitializedException`
- `LoadDefaultValuesAbortedException`
- `InvalidParameterIndexException`

loadPersonalizedValueToReport Method

Applies To

[IParameterUtility Interface](#)

See Also

[ParameterException Class](#)

Description

Loads the BW personalized value(s) into the report as the current value(s) of the parameter, and then returns it.

Syntax

```
public IReportParameterValues loadPersonalizedValueToReport(  
    int paramNumber) throws ParameterException;
```

Parameters

paramNumber

An int that specifies the zero-based index of the parameter that you want to load the personalized value(s) for. This parameter matches the index that is used to access the parameter values returned by the `getReportParameters` Method. (The `getReportParameters` Method is a member of the `IReportProcessingInfo` Class in the *BusinessObjects Enterprise Java SDK*.)

Returns

A `IReportParameterValues` object that contains the personalized value(s).

Throws

`ParameterException`

This is thrown if the process is unsuccessful.

Remarks

Returns null if no BW personalized variable was loaded.

Throws one of the following parameter exceptions:

- `HierarchyMissingException`
- `NotInitializedException`
- `InvalidParameterIndexException`.

savePersonalizedValue Method

Applies To

[IParameterUtility Interface](#)

See Also

[ParameterException Class](#)

Description

Sets the personalized value(s) as the current value(s) for the specified parameter in the report object.

Syntax

```
public void savePersonalizedValue(  
    int paramNumber) throws ParameterException;
```

Parameters

paramNumber

An int that specifies the zero-based index of the parameter. This parameter matches the index that is used to access the parameter

values returned by the `getReportParameters` Method. (The `getReportParameters` Method is a member of the `IReportProcessingInfo` Class in the *BusinessObjects Enterprise Java SDK*.)

Throws

`ParameterException`

This is thrown if the process is unsuccessful.

Remarks

Throws one of the following parameter exceptions:

- `NotInitializedException`
- `PersonalizationException`
- `InvalidParameterIndexException`

validateHierarchicalParameters Method

Applies To

[IParameterUtility Interface](#)

See Also

[ParameterException Class](#)

Description

Raises an exception on the first invalid hierarchy and hierarchy node parameter detected.

Syntax

```
public void validateHierarchicalParameters() throws  
ParameterException;
```

Throws

`ParameterException`

This is thrown if the process is unsuccessful.

Remarks

Throws one of the following parameter exceptions:

- `HierarchyMissingException`
- `HierarchyNodeException`
- `NotInitializedException`

Note:

- Throws a `HierarchyMissingException` when no hierarchy has been set as a current value for a given pair of hierarchy and hierarchy node parameters.
- Throws a `HierarchyNodeException` when the hierarchy node is not valid for the selected hierarchy.

IPicklistGenerator Interface

See Also

[IParameterUtility Interface](#)
[IPicklistEntry Interface](#)
[ITreeNode Interface](#)
[ParameterUtilityFactory Class](#)
[ParameterException Class](#)

Syntax

```
public interface IPicklistGenerator
```

Remarks

The [IPicklistGenerator Interface](#) contains a collection of [IPicklistEntry](#) objects, which belong to a particular report. This interface allows you to enumerate through the various pick lists in a report.

Methods

Method	Description
get Method	Returns the pick list entry for the provided value number.
size Method	Returns the number of pick list entries available at the top level.

get Method

Applies To

[IPicklistGenerator Interface](#)

Description

Returns the pick list entry for the provided value number.

Syntax

```
public IPicklistEntry get(  
    int valueNum);
```

Parameters

valueNum

An int that specifies the zero-based index of the parameter. This parameter matches the index that is used to access the parameter values returned by the [getReportParameters](#) Method. (The [getReportParameters](#) Method is a member of the

IReportProcessingInfo Class in the *BusinessObjects Enterprise Java SDK*.)

Returns

A IPicklistEntry object that contains the provided pick list entry.

size Method**Applies To**

[IPicklistGenerator Interface](#)

Description

Returns the number of pick list entries available at the top level.

Syntax

```
public int size();
```

Returns

An int that specifies the number of pick list entries available at the top level.

Remarks

Top level refers to nodes that are on the top level of the hierarchy structure. If no hierarchy structure exists, the top level refers to the entire collection of entries in the pick list.

IPicklistEntry Interface

See Also

[IParameterUtility Interface](#)
[IPicklistGenerator Interface](#)
[ITreeNode Interface](#)
[ParameterUtilityFactory Class](#)
[ParameterException Class](#)

Syntax

```
public interface IPicklistEntry extends ITreeNode
```

Remarks

The [IPicklistEntry Interface](#) represents a single entry within a pick list. This interface provides methods which allow you to get and set the description and value for the pick list entry, and determine if you intend to display the pick list.

This interface extends the [ITreeNode Interface](#), which allows you to examine the tree node structure associated with the parameter values.

Note: If you use the [setDescription Method](#) or the [setValue Method](#) to modify a pick list entry, the changes will not persist in your SAP system. Therefore, any changes you made to the value of a pick list entry will not be reflected in SAP.

Methods

Method	Description
getDescription Method	Returns the description of the entry.
getValue Method	Returns the value of the entry.
isForDisplay Method	Returns a boolean that indicates whether the entry is intended for display.
setDescription Method	Sets the description of the entry.
setForDisplay Method	Sets whether the pick list is intended for display.
setValue Method	Sets the value of the entry.

getDescription Method

Applies To

IPicklistEntry Interface

Description

Returns the description of the entry.

Syntax

```
public String getDescription();
```

Returns

A String that contains the description of the entry.

getValue Method

Applies To

IPicklistEntry Interface

Description

Returns the value of the entry.

Syntax

```
public String getValue();
```

Returns

A String that contains the value of the entry.

isForDisplay Method

Applies To

IPicklistEntry Interface

Description

Returns a boolean that indicates whether the entry is intended for display.

Syntax

```
public boolean isForDisplay();
```

Returns

A boolean that indicates whether the entry is intended for display.
true - the entry is intended for display.
false - the entry is not intended for display.

setDescription Method

Applies To

[IPicklistEntry Interface](#)

Description

Sets the description of the entry.

Syntax

```
public void setDescription(  
    String description);
```

Parameters

description

A String that specifies the description of the entry.

setForDisplay Method

Applies To

[IPicklistEntry Interface](#)

Sets whether the pick list should be displayed.

Syntax

```
public void setForDisplay(  
    boolean forDisplay);
```

Parameters

forDisplay

A boolean that indicates whether the entry is set for display.

true - the entry is set for display.

false - the entry is not set for display.

setValue Method

Applies To

[IPicklistEntry Interface](#)

Description

Sets the value of the entry.

Syntax

```
public void setValue(  
    String value);
```

Parameters

value

A String that specifies the value of the description.

ITreeNode Interface

See Also

[IParameterUtility Interface](#)

[IPicklistEntry Interface](#)

Syntax

```
public interface ITreeNode
```

Remarks

The [ITreeNode Interface](#) represents nodes in a tree structure, and provides methods to access these nodes.

Methods

Method	Description
getChildAt Method	Returns the child node at the given index.
getChildren Method	Returns a collection of child nodes for the current node.
getChildrenCount Method	Returns the number of children for the current node.
getParent Method	Returns the parent node of the current node.
isChildrenAvailable Method	Returns a boolean that indicates whether child nodes are available.

getChildAt Method

Applies To

[ITreeNode Interface](#)

Description

Returns the child node at the given index.

Syntax

```
public ITreeNode getChildAt(  
    int index) throws ArrayIndexOutOfBoundsException;
```

Parameters

index

An int that specifies the location of the child node.

Returns

An ITree object that represents the child node.

Remarks

Returns null if no child exists for the specified index.

getChildren Method

Applies To

[ITreeNode Interface](#)

Description

Returns a collection of child nodes for the current node.

Syntax

```
public ITreeNode[] getChildren();
```

Returns

A collection that contains the child nodes associated with current node.

Remarks

Returns null if no children are available.

getChildrenCount Method

Applies To

[ITreeNode Interface](#)

Description

Returns the number of children for the current node.

Syntax

```
public int getChildrenCount();
```

Returns

An int that contains the number of children associated with the current node.

getParent Method

Applies To

[ITreeNode Interface](#)

Description

Returns the parent node of the current node.

Syntax

```
public ITreeNode getParent();
```

Returns

An ITree object that represents the parent node of the current node.

Remarks

Returns null if no parent exists.

isChildrenAvailable Method**Applies To**

[ITreeNode Interface](#)

Description

Returns a boolean that indicates whether child nodes are available.

Syntax

```
public boolean isChildrenAvailable();
```

Returns

true if child nodes are available; false if child nodes are not available.

ParameterException Class

See Also

- [IParameterUtility Interface](#)
- [IPicklistEntry Interface](#)
- [IPicklistGenerator Interface](#)
- [ITreeNode Interface](#)
- [ParameterUtilityFactory Class](#)

Syntax

```
public class ParameterException extends  
    com.crystaldecisions.celib.exception.AbstractException
```

Remarks

The [ParameterException Class](#) is thrown when an error is encountered using the interfaces associated with this package. This class also provides two methods, [getSourceParameterIndex Method](#) and [getCauseParameterIndex Method](#), which enable you to determine which hierarchy or hierarchy node caused an exception.

Subclasses

Class	Description
HierarchyMissingException Class	Thrown when the hierarchy is missing for a dependent hierarchy node.
HierarchyNodeException Class	Thrown when a mismatch exists between the hierarchy and hierarchy node.
InvalidParameterIndexException Class	Thrown when an invalid parameter index is accessed.
LoadDefaultValuesAbortedException Class	Thrown when the default value cannot be loaded.
NotInitializedException Class	Thrown when the object is not initialized.
PersonalizationException Class	Thrown when an error is encountered loading variables with personalized values.

Methods

Method	Description
getSourceParameterIndex Method	Returns the index of the parameter that throws the exception.
getCauseParameterIndex Method	Returns the index of the parameter that throws the exception.

Subclasses

HierarchyMissingException Class

Applies To

[ParameterException Class](#)

See Also

[getSourceParameterIndex Method](#)

[getCauseParameterIndex Method](#)

Description

Thrown when the hierarchy is missing for a dependent hierarchy node.

Syntax

```
public static class HierarchyMissingException extends  
    ParameterException
```

Remarks

If you want to get more information about the cause of the exception you can use the following two methods: [getSourceParameterIndex Method](#) and [getCauseParameterIndex Method](#).

HierarchyNodeException Class

Applies To

[ParameterException Class](#)

See Also

[getSourceParameterIndex Method](#)

[getCauseParameterIndex Method](#)

Description

Thrown when a mismatch exists between the hierarchy and the hierarchy node.

Syntax

```
public static class HierarchyNodeException extends  
    ParameterException
```

Remarks

If you want to get more information about the cause of the exception you can use the following two methods: [getSourceParameterIndex Method](#) and [getCauseParameterIndex Method](#).

InvalidParameterIndexException Class

Applies To

[ParameterException Class](#)

Description

Thrown when an invalid parameter index is accessed.

Syntax

```
public static class InvalidParameterIndexException extends  
    ParameterException
```

LoadDefaultValuesAbortedException Class

Applies To

[ParameterException Class](#)

Description

Thrown when the default value cannot be loaded.

Syntax

```
public static class LoadDefaultValuesAbortedException  
    extends ParameterException
```

NotInitializedException Class

Applies To

[ParameterException Class](#)

Description

Thrown when the object is not initialized.

Syntax

```
public static class NotInitializedException extends  
    ParameterException
```

PersonalizationException Class

Applies To

[ParameterException Class](#)

Description

Thrown when a error is encountered loading variables with personalized values.

Syntax

```
public static class PersonalizationException extends  
    ParameterException
```

Methods

getCauseParameterIndex Method

Applies To

[ParameterException Class](#)

See Also

[getSourceParameterIndex Method](#)

Description

Returns the index of the parameter that throws the exception.

Syntax

```
public int getCauseParameterIndex();
```

Returns

An int that specifies the index of the parameter throwing the exception.

Remarks

This method is most often used to retrieve the index of the hierarchy that is causing the exception.

getSourceParameterIndex Method

Applies To

[ParameterException Class](#)

See Also

[getCauseParameterIndex Method](#)

Description

Returns the index of the parameter that throws the exception.

Syntax

```
public int getSourceParameterIndex();
```

Returns

An int that specifies the index of the parameter throwing the exception.

Remarks

This method is most often used to retrieve the index of the hierarchy node that is causing the exception.

Crystal SAP Web RRI Package

The Crystal SAP Web RRI (Report-Report-Interface) Package allows you to access SAP report parameter values sent from a SAP report to a Crystal Report. In RRI, the receiver functions as a bridge between the two types of reports, and interprets the SAP report parameter values for the Crystal Report. The [IRRIReceiver Interface](#) lets you view the SAP report parameter values sent by the SAP BW system. The IRRIReceiver's [initialize Method](#) takes the Crystal report ID, the CMS logon token, and the SAP BW system credentials. IRRIReceiver then populates itself with SAP parameter values from SAP BW, which are encapsulated as instances of the [IRRIEntry Interface](#). As a result, you can iterate through the IRRIReceiver for SAP web parameter values.

Package Name

com.crystaldecisions.sap.web.rri

RRI Class

See Also

[IRRIEntry Interface](#)
[IRRIReceiver Interface](#)

Syntax

```
public class RRI
```

Remarks

The [RRI Class](#) is the starting point for using the RRI package. This class allows you to retrieve an instance of the [IRRIReceiver Interface](#).

Methods

Method	Description
getRRIReceiver Method	Returns an instance of the IRRIReceiver Interface .

getRRIReceiver Method

Applies To

[RRI Class](#)

See Also

[IRRIReceiver Interface](#)

Description

Returns an instance of the [IRRIReceiver Interface](#).

Syntax

```
public static synchronized IRRIReceiver getRRIReceiver();
```

Returns

An instance of the [IRRIReceiver Interface](#).

IRRIReceiver Interface

See Also

[IRRIEntry Interface](#)

Syntax

```
public interface IRRIReceiver
```

Remarks

The [IRRIReceiver Interface](#) accesses SAP report parameter values sent from a SAP report to a Crystal Report. Before you can use this interface, you must first initialize it with the [initialize Method](#).

Methods

Method	Description
get Method	Returns the instance of IRRIEntry Interface at the provided index.
initialize Method	Initializes this interface using an RRI handle and SAP BW credentials.
size Method	Returns the number of parameter values.

get Method

Applies To

[IRRIReceiver Interface](#)

See Also

[IRRIEntry Interface](#)

[RRIException Class](#)

Description

Returns the instance of [IRRIEntry Interface](#) at the provided index.

Syntax

```
IRRIEntry get(  
    int index) throws RRIException;
```

Parameters

index

An int that specifies the zero-based index of the [IRRIEntry Interface](#) to be retrieved.

Throws

RRIException
An RRIException.

Returns

The instance of [IRRIEntry Interface](#) at the provided index.

initialize Method

Applies To

[IRRIReceiver Interface](#)

See Also

[RRIException Class](#)

Description

Initializes this interface using an RRI handle and SAP BW credentials.

Syntax

```
public void Initialize(  
    int reportId,  
    String RRIHandle,  
    String CMSLogonToken,  
    String BWPassword,  
    String BWTicket) throws RRIException;
```

Parameters

reportId
The report ID.

RRIHandle
The RRI handle.

CMSLogonToken
The CMS logon token.

BWPassword
The SAP BW password.

BWTicket
The SAP BW SSO Logon Ticket.

Throws

RRIException
An RRIException.

size Method

Applies To

[IRRIReceiver Interface](#)

See Also

[RRIException Class](#)

Description

Returns the number of parameter values.

Syntax

```
int size() throws RRIException;
```

Throws

RRIException

An RRIException.

Returns

The number of parameter values.

IRRIEntry Interface

See Also

[IRRIReceiver Interface](#)

Syntax

```
public interface IRRIEntry
```

Remarks

The [IRRIEntry Interface](#) represents a single SAP report parameter value. IRRIEntry instances are contained in an [IRRIReceiver Interface](#).

Methods

Method	Description
getHigh Method	Returns the upper bound range value of the parameter.
getLow Method	Returns the lower bound range value of the parameter.
getName Method	Returns the name of the parameter.
getOpt Method	Returns the bound operator of the parameter.
getSign Method	Returns whether the parameter is inclusive or exclusive of the range or discrete value, which is defined by the getHigh Method , the getLow Method , and the getOpt Method .

getHigh Method

Applies To

[IRRIEntry Interface](#)

Description

Returns the upper bound range value of the parameter.

Syntax

```
String getHigh();
```

Returns

The upper bound range value of the parameter.

Remarks

The only time that both the [getHigh Method](#) and the [getLow Method](#) are used together is when the [getOpt Method](#) returns BT (between). In all other cases, only the [getLow Method](#) is used.

getLow Method

Applies To

[IRRIEntry Interface](#)

Description

Returns the lower bound range value of the parameter.

Syntax

```
String getLow();
```

Returns

The lower bound range value of the parameter.

Remarks

The only time that both the [getHigh Method](#) and the [getLow Method](#) are used together is when the [getOpt Method](#) returns BT (between). In all other cases, only the [getLow Method](#) is used.

getName Method

Applies To

[IRRIEntry Interface](#)

Description

Returns the name of the parameter.

Syntax

```
String getName();
```

Returns

The name of the parameter.

getOpt Method

Applies To

[IRRIEntry Interface](#)

Description

Returns the bound operator of the parameter.

Syntax

```
String getOpt();
```

Returns

The bound operator of the parameter.

Remarks

The values are outlined below:

Value	Description
EQ	equals to
BT	between
GT	greater than
LT	less than
LE	less than or equal to
GE	greater than or equal to
CP	contains pattern

getSign Method

Applies To

[IRRIEntry Interface](#)

Description

Returns whether the parameter is inclusive or exclusive of the range or discrete value, which is defined by the [getHigh Method](#), the [getLow Method](#), and the [getOpt Method](#).

Syntax

```
String getSign();
```

Returns

A string that indicates whether the parameter is inclusive or exclusive.

Remarks

The string can be one of two values: I (includes), or E (excludes).

RRIException Class

See Also

[IRRIEntry Interface](#)
[IRRIReceiver Interface](#)

Syntax

```
public class RRIException extends  
    com.crystaldecisions.celib.exception.AbstractException
```

Remarks

The [RRIException Class](#) is thrown when an error occurs while using the [IRRIReceiver Interface](#).

Subclasses

Class	Description
InvalidParameterIndexException Class	Thrown when an invalid parameter index is accessed.
NotInitializedException Class	Thrown when the object is not initialized.
UnableToInitializeException Class	Thrown when the component cannot be initialized.

InvalidParameterIndexException Class

Applies To

[RRIException Class](#)

Description

Thrown when an invalid parameter index is accessed.

NotInitializedException Class

Applies To

[RRIException Class](#)

Description

Thrown when the object is not initialized.

Syntax

```
public static class NotInitializedException extends  
    RRIException
```

Syntax

```
public static class InvalidParameterIndexException extends  
    RRIException
```

UnableToInitializeException Class

Applies To

[RRIException Class](#)

Description

Thrown when the component cannot be initialized.

Syntax

```
public static class UnableToInitializeException extends  
    RRIException
```


Crystal SAP Web SSO Package

The Crystal SAP Web SSO Package provides SSO (Single Sign On) functionality between a SAP system and a BusinessObjects Enterprise system. The [IDataSourceSSO Interface](#) determines if Secure Network Communication (SNC) is available for a given Enterprise session and data source of a Crystal Report. To achieve this task, IDataSourceSSO determines the logical SAP system name from the database name. It then matches the system name against the available SAP aliases to make the verification.

Package Name

com.crystaldecisions.sap.web.sso

SSO Class

See Also

[IDataSourceSSO Interface](#)

Syntax

```
public class SSO
```

Remarks

The [SSO Class](#) is the starting point for using the SSO package. This class allows you to get an instance of the [IDataSourceSSO Interface](#).

Methods

Method	Description
getDataSourceSSO Method	Returns an instance of the IDataSourceSSO Interface .

getDataSourceSSO Method

Applies To

[SSO Class](#)

See Also

[IDataSourceSSO Interface](#)

Description

Returns an instance of the [IDataSourceSSO Interface](#).

Syntax

```
public static synchronized IDataSourceSSO  
getDataSourceSSO();
```

Returns

An instance of the [IDataSourceSSO Interface](#).

IDatasourceSSO Interface

See Also

[SSO Class](#)

Syntax

```
public interface IDatasourceSSO
```

Remarks

The [IDatasourceSSO Interface](#) determines if Secure Network Communication (SNC) is available for a given Enterprise session and data source of a Crystal Report. Given the CMS logon token, the Enterprise session, and the data source of a report, the [isSNCAvailable Method](#) returns a true or false value.

A Crystal Report may contain multiple data sources. In that case, the interface provides the [setDataSourceIndex Method](#) to access each specific data source. The [getLogonCount Method](#) returns the number of data sources for the report. Using these two methods, you can iterate through multiple data sources and verify SNC on each one.

In addition, IDatasourceSSO can create an offline version of the logon token. In doing so, a Crystal Report can be scheduled to run after the Enterprise session has expired. IDatasourceSSO encrypts and stores the user credentials into the Crystal Report.

This interface is particularly useful for creating web applications that schedule and view Crystal Reports based on an SAP data source. If SNC is available, the application logic can use password-less logon and offline tokens.

Methods

Method	Description
getDatabaseName Method	Returns the database name.
getDataSourceName Method	Returns the data source name.
getExternalID Method	Returns the current alias of the user.
getLogonCount Method	Returns the number of data sources for the Crystal Report.
getPassword Method	Returns the password.
getServerName Method	Returns the server name.
getSubReport Method	Returns the subreport name.
getUsername Method	Returns the offline token.

Method	Description
isOfflineSNCCName Method	Determines whether an offline token is valid for the data source.
isSNCAvailable Method	Determines whether SNC is available for the data source.
setBufferWidth Method	For internal use only.
setCMSLogonToken Method	Sets the CMS logon token.
setDataSourceIndex Method	Sets the current data source of the report using the given index.
setEnterpriseSession Method	Sets the Enterprise session.
setPassword Method	Sets the password.
setReportID Method	Sets the Crystal report ID.
setScheduleTime Method	Sets whether to schedule the report with an offline token.

getDatabaseName Method

Applies To

[IDatasourceSSO Interface](#)

Description

Returns the database name.

Syntax

```
String getDatabaseName();
```

Returns

The database name.

getDataSourceName Method

Applies To

[IDatasourceSSO Interface](#)

Description

Returns the data source name.

Syntax

```
String getDataSourceName();
```

Returns

The data source name.

getExternalID Method

Applies To

[IDatasourceSSO Interface](#)

Description

Returns the current alias of the user.

Syntax

```
String getExternalID();
```

Returns

The current alias of the user.

Remarks

The current alias corresponds to the current data source.

getLogonCount Method

Applies To

[IDatasourceSSO Interface](#)

Description

Returns the number of data sources for the Crystal Report.

Syntax

```
int getLogonCount();
```

Returns

The logon count.

Remarks

Use this method in conjunction with the [setDataSourceIndex Method](#) to iterate through multiple data sources.

getPassword Method

Applies To

[IDatasourceSSO Interface](#)

See Also

[setPassword Method](#)

[getUsername Method](#)

Description

Returns the password.

Syntax

```
String getPassword();
```

Returns

The password.

getServerName Method

Applies To

[IDatasourceSSO Interface](#)

Description

Returns the server name.

Syntax

```
String getServerName();
```

Returns

The server name.

getSubReport Method

Applies To

[IDatasourceSSO Interface](#)

Description

Returns the subreport name.

Syntax

```
String getSubReport();
```

Returns

The subreport name.

getUsername Method

Applies To

[IDatasourceSSO Interface](#)

Description

Returns the offline token.

Syntax

```
String getUsername();
```

Returns

The offline token.

isOfflineSNCName Method

Applies To

[IDatasourceSSO Interface](#)

Description

Determines whether an offline token is valid for the data source.

Syntax

```
boolean isOfflineSNCName(  
    String name);
```

Parameters

name

A String that specifies the token to verify.

Returns

true if the offline token is valid; false if the offline token is not valid.

Remarks

You can obtain the token from the [getUsername Method](#).

isSNCAvailable Method

Applies To

[IDatasourceSSO Interface](#)

Description

Determines whether Secure Network Communication (SNC) is available for the data source.

Syntax

```
boolean isSNCAvailable();
```

Returns

true if SNC is available; false if SNC is not available.

setBufferWidth Method

Description

For internal use only.

setCMSLogonToken Method

Applies To

[IDatasourceSSO Interface](#)

Description

Sets the CMS logon token.

Syntax

```
void setCMSLogonToken(  
    String logonToken);
```

Parameters

logonToken
The CMS logon token.

setDataSourceIndex Method

Applies To

[IDatasourceSSO Interface](#)

Description

Sets the current data source of the report using the given index.

Syntax

```
void setDataSourceIndex(  
    int index);
```

Parameters

index
The index of the data source that you want to access.

Remarks

Use this method together with the [getLogonCount Method](#) to iterate through multiple data sources.

setEnterpriseSession Method

Applies To

[IDatasourceSSO Interface](#)

Description

Sets the Enterprise session.

Syntax

```
void setEnterpriseSession(  
    com.crystaldecisions.sdk.framework.IEnterpriseSession  
    enterpriseSession);
```

Parameters

enterpriseSession

The Enterprise session.

setPassword Method

Applies To

[IDatasourceSSO Interface](#)

See Also

[getPassword Method](#)

[getUsername Method](#)

Description

Sets the password.

Syntax

```
void setPassword(  
    String password);
```

Parameters

password

A String that specifies the password.

setReportID Method

Applies To

[IDatasourceSSO Interface](#)

Description

Sets the Crystal report ID.

Syntax

```
void setReportID(  
    int reportId);
```

Parameters

reportId
The report ID.

setScheduleTime Method

Applies To

[IDatasourceSSO Interface](#)

Description

Sets whether to schedule the report with an offline token.

Syntax

```
void setScheduleTime(  
    boolean scheduleTime);
```

Parameters

scheduleTime
A boolean that indicates if the report is scheduled with an offline token.
true - scheduled with an offline token.
false - not scheduled with an offline token.



Get More Help





appendix

Additional developer resources

http://www.businessobjects.com/products/dev_zone/java/default.asp

Online customer support

The Business Objects Customer Support web site contains information about Customer Support programs and services. It also has links to a wide range of technical information including knowledgebase articles, downloads, and support forums.

<http://www.businessobjects.com/support/>

Looking for the best deployment solution for your company?

Business Objects consultants can accompany you from the initial analysis stage to the delivery of your deployment project. Expertise is available in relational and multidimensional databases, in connectivities, database design tools, customized embedding technology, and more.

For more information, contact your local sales office, or contact us at:

<http://www.businessobjects.com/services/consulting/>

Looking for training options?

From traditional classroom learning to targeted e-learning seminars, we can offer a training package to suit your learning needs and preferred learning style. Find more information on the Business Objects Education web site:

<http://www.businessobjects.com/services/training>

Send us your feedback

Do you have a suggestion on how we can improve our documentation? Is there something you particularly like or have found useful? Drop us a line, and we will do our best to ensure that your suggestion is included in the next release of our documentation:

documentation@businessobjects.com

Note: If your issue concerns a Business Objects product and not the documentation, please contact our Customer Support experts. For information about Customer Support visit: <http://www.businessobjects.com/support/>.

A | Get More Help
Send us your feedback

Index

A

architecture 9

B

Business Objects
 consulting services 60
 support services 60
 training services 60

C

consultants, Business Objects 60
Crystal SAP Web Parameter Package 13
Crystal SAP Web RRI Package 39
Crystal SAP Web SSO Package 49
customer support 60

D

deletePersonalizedValue Method
 (IPParameterUtility) 16
documentation
 feedback on 60

F

feedback, on documentation 60

G

get Method (IPicklistGenerator) 26
get Method (IRRIReceiver) 41
getCauseParameterIndex Method
 (ParameterException) 37
getChildAt Method (ITreeNode) 31
getChildren Method (ITreeNode) 32
getChildrenCount (ITreeNode) 32
getDatabaseName Method (IDatasourceSSO) 52
getDataSourceName Method (IDatasourceSSO)
 52

getDatasourceSSO Method (SSO) 50
getDescription Method (IPicklistEntry) 29
getExternalID Method (IDatasourceSSO) 53
getHigh Method (IRRIEntry) 44
getLogonCount Method (IDatasourceSSO) 53
getLow Method (IRRIEntry) 45
getName Method (IRRIEntry) 45
getOpt Method (IRRIEntry) 45
getParameterUtility Method
 (ParameterUtilityFactory) 14
getParent Method (ITreeNode) 32
getPassword Method (IDatasourceSSO) 54
getPicklistGenerator Method (IPParameterUtility) 17
getReport Method (IPParameterUtility) 19
getRRIReceiver Method (RRI) 40
getServerName Method (IDatasourceSSO) 54
getSign Method (IRRIEntry) 46
getSourceParameterIndex Method
 (ParameterException) 38
getSubReport Method (IDatasourceSSO) 54
getUsername Method (IDatasourceSSO) 55
getValue Method (IPicklistEntry) 29

H

HierarchyMissingException Class
 (ParameterException) 35
HierarchyNodeException Class
 (ParameterException) 36

I

IDatasourceSSO Interface 51
initialize Method (IPParameterUtility) 20
initialize Method (IRRIReceiver) 42
InvalidParameterIndexException Class
 (ParameterException) 36
InvalidParameterIndexException Class
 (RRIException) 47
IPParameterUtility Interface 15

Index

IPicklistEntry Interface 28
IPicklistGenerator Interface 26
IRRIEntry Interface 44
IRRIReceiver Interface 41
IsBWPParameter Method (IPParameterUtility) 21
isChildrenAvailable Method (ITreeNode) 33
isForDisplay Method (IPicklistEntry) 29
isHierarchyNodeParameter Method
(IPParameterUtility) 22
isOfflineSNCName Method (IDatasourceSSO) 55
isSNCAvailable Method (IDatasourceSSO) 55
ITreeNode Interface 31

L

LoadDefaultValuesAbortedException Class
(ParameterException) 36
loadDefaultValueToReport Method
(IPParameterUtility) 22
loadPersonalizedValueToReport Method
(IPParameterUtility) 23

N

NotInitializedException Class
(ParameterException) 37
NotInitializedException Class (RRIException) 47

O

Online Customer Support 60

P

ParameterUtilityFactory Class 14
PersonalizedException Class
(ParameterException) 37

R

RRI Class 40
RRIException Class 47

S

savePersonalizedValue Method
(IPParameterUtility) 24
setBufferWidth Method (IDatasourceSSO) 56

setCMSLogonToken Method
(IDatasourceSSO) 56
setDataSourceIndex Method
(IDatasourceSSO) 56
setDescription Method (IPicklistEntry) 30
setEnterpriseSession Method
(IDatasourceSSO) 57
setForDisplay Method (IPicklistEntry) 30
setPassword Method (IDatasourceSSO) 57
setReportID Method (IDatasourceSSO) 57
setScheduleTime Method (IDatasourceSSO) 58
setValue Method (IPicklistEntry) 30
size Method (IPicklistGenerator) 27
size Method (IRRIReceiver) 43
SSO Class 50
support
 customer 60
 technical 60
 web site 60

T

technical support 60
training, on Business Objects products 60

U

UnableToInitializeException Class
(RRIException) 48

V

validateHierarchicalParameters Method
(IPParameterUtility) 25

W

web
 customer support 60
web sites
 support 60
 training 60