
OZ .NET Server

API Developer's Guide



Table of Contents

Part I. OZ .NET Server API	1
1. Class	2
Cache	2
ConnectionPool	4
ConnectionPoolInfo	5
ConnectionPoolStatus	8
DataBind	9
Log	10
Mail	11
MemoryStatus	15
Module	15
Monitor	21
OZAPIException	22
OZDataSetInfo	23
OZODItem	24
OZParameter	25
OZReportParameter	25
OZServerInfo	26
Service	27
2. Interface	29
IReportInfo	29
Part II. OZ Scheduler API	32
1. Class	33
DirectExportResult	33
DirectPrintResult	34
DirectTaskResult	35
FileInfo	36
Program	36
Publisher	39
Scheduler	41
ServerInfo	51
TaskHolidayGroupInfo	52
TaskHolidayInfo	54
TaskResult	55

Part III. OZ Repository API	57
1. Class	58
RepositoryEX	58
Configuring the repository	59
Creating groups and users	60
Editing groups and users	61
Deleting groups and users	63
Getting groups and users	64
Setting up group and user permissions	66
Getting group and user permissions	68
Creating categories and items	69
Editing categories and items	71
Deleting categories and items	74
Getting categories and items	76
Setting up category and item permissions	79
Getting category and item permissions	81
Using item version control	85
2. Interface	91
ICategoryInfo	91
IGroupInfo	93
IHistoryInfo	94
IItemInfo	95
ILoginInfo	97
IRepository	99
IRepositoryCategory	102
IRepositoryGroup	107
IRepositoryGroupAuthorityToCategory	113
IRepositoryGroupAuthorityToItem	115
IRepositoryHistory	117
IRepositoryItem	119
IRepositoryMultiLoginUser	125
IRepositoryUser	126
IRepositoryUserAuthorityToCategory	129
IRepositoryUserAuthorityToItem	132
IUserInfo	134
Part IV. User Defined Log API	137
1. Interface	138

IClearQuestLogTarget	138
IDMLogTarget	139
IFileStoreLogTarget	140
IHTTPStoreLogTarget	141
IMonitorLogTarget	141
IProcedureLogTarget	143
IRDBLogTarget	144
ISAPLogTarget	146
ISOAPLogTarget	147
IUDSLogTarget	148
IUserDefinedLogger	148
IUserDefinedLogTarget	149
IXMLLogTarget	150
 Part V. User Defined Resource API	 152
1. Class	153
OZUDRInfo	153
OZUDRResult	154
Resource	155
2. Interface	157
IOZUserDefinedResource	157

OZ .NET Server API

OZ .NET Server API provides [classes](#) and [interfaces](#).

Class

OZ .NET Server API provides the following classes:

- [Cache](#)
- [ConnectionPool](#)
- [ConnectionPoolInfo](#)
- [ConnectionPoolStatus](#)
- [DataBind](#)
- [Log](#)
- [Mail](#)
- [MemoryStatus](#)
- [Module](#)
- [Monitor](#)
- [OZAPIException](#)
- [OZDataSetInfo](#)
- [OZODItem](#)
- [OZParameter](#)
- [OZReportParameter](#)
- [OZServerInfo](#)
- [Service](#)

Cache

This class provides methods related to the cache function of OZ Server.

Constructor Detail

Cache

```
public Cache(string url, string id, string pw, bool autoLogin, bool useUSL)
```

Performs processing related to OZ Server Cache Manager and processes functions such as the Cache Manager configuration and delete cache item.

Parameters:

url - URL of OZ Server

id - User ID

pw - User Password

autoLogin - Whether to use the autologin

useUSL - Whether to use the USL

Method Detail

GetCacheConfiguration

```
public OZAttributeList GetCacheConfiguration() throws OZAPIException
```

Gets the cache settings of the OZ Server.

SetCacheConfiguration

```
public void SetCacheConfiguration(OZAttributeList attrs) throws OZAPIException
```

Changes the cache configurations.

Parameters:

attrs - Cache configuration property

ConnectionPool

This class provides methods related to the database connection pool of OZ Server.

Constructor Detail

ConnectionPool

```
public ConnectionPool(string url, string id, string pw, bool autoLogin, bool useUSL)
```

Performs management functions such as registering, deleting the JDBC/ODBC data of the database.

Parameters:

url - URL of OZ Server

id - User ID

pw - User Password

autoLogin - Whether to use the Autologin

useUSL - Whether to use the USL

Method Detail

AddPool

```
public void AddPool(ConnectionPoolInfo pool) throws OZAPIException
```

Adds ConnectionPool. Set a value in the declared member variable in the "ConnectionPoolInfo" class to add ConnectionPool.

Parameters:

pool - ConnctionPoolInfo object, which is the information of the ConnectionPool to add

RemovePool

public void RemovePool(string alias) throws OZAPIException

Deletes the ConnectionPool of the selected name.

Parameters:

alias - ConnectionPool alias

GetPoolInfos

public ConnectionPoolInfo[] GetPoolInfos() throws OZAPIException

Gets the information about all ConnectionPools.

GetPoolStatuses

public ConnectionPooStatus[] GetPoolStatuses() throws OZAPIException

Gets the status information of all ConnectionPools.

GetPoolInfo

public ConnectionPoolInfo GetPoolInfo(string alias) throws OZAPIException

Gets the ConnectionPoolInfo list for the corresponding ConnectionPool.

Parameters:

alias - Alias of ConnectionPool

Save

public void Save() throws OZAPIException throws OZAPIException

Save the data of all ConnectionPool in the server.

ConnectionPoolInfo

This class provides properties to get the information about database connection.

Property Detail

Alias

```
public string Alias
```

The alias of the connection.

Vendor

```
public string Vendor
```

The vendor name of the connection.

InitialConnections

```
public int InitialConnections
```

The initial number of connections.

MaxConnections

```
public int MaxConnections
```

The maximum number of connections.

TimeOut

```
public int TimeOut
```

The connection timeout in second.

InitSQLs

```
public string InitSQLs
```

The initial query to execute after connection open.

CloseSQLs

```
public string CloseSQLs
```

The query to execute before closing or freeing a connection.

SessionQuery

`public string SessionQuery`

The session query of a connection.

DoConnectioncheck

`public bool DoConnectionCheck`

The status of connection validity check.

TestQuery

`public string TestQuery`

The query to test connection.

Items

`public IDictionary Items`

The Item <key,value> of the vendor in dbconfig.xml.

Decode

`public string Decode`

The decoding set for the corresponding ConnectionPool.

Encode

`public string Encode`

The encoding set for the corresponding ConnectionPool.

FetchRow

`public int FetchRow`

The number of JDBC result set rows to get by one time.

ConnectionPoolStatus

This class provides member variables to get the state information of the connection pool.

Member Variables

Status

```
public Status Status
```

The current status.

Statusstring

```
public string Statusstring
```

The current status message.

FreeConnectionCount

```
public int FreeConnectionCount
```

The number of free connections.

CheckedOutConnectionCount

```
public int CheckedOutConnectionCount
```

The number of Checked out connections.

ErrorMessage

```
public string ErrorMessage
```

The error message occurred during ConnectionPool creation.

Error

```
public Exception Error
```

The error occurred during ConnectionPool creation.

DataBind

This class provides methods related to the data binding function of OZ Server.

Constructor Detail

DataBind

```
public DataBind(string url, string id, string pw, bool autoLogin, bool useUSL)
```

Performs data binding function.

Parameters:

url - URL of OZ Server

id - User ID

pw - User Password

autoLogin - Whether to use the AutoLogin

useUSL - Whether to use the USL

Method Detail

GetConfiguration

```
public OZAttributeList GetConfiguration() throws OZAPIException
```

Gets the values defined in the databind.properties file.

SetConfiguration

```
public void SetConfiguration(OZAttributeList config) throws OZAPIException
```

Sets the values defined in the databind.properties file.

Parameters:

config - DataBind configuration properties

Log

This class provides methods related to the log function of OZ Server.

Constructor Detail

Log

```
public Log(string url, string id, string pw, bool autoLogin, bool useUSL)
```

Performs the processing related to OZ server Log Manager and configures the settings for recording the various messages, information, and errors, etc. arising during server operation.

Parameters:

url - URL of OZ Server

id - User ID

pw - User Password

autoLogin - Whether to use the AutoLogin

useUSL - Whether to use the USL

Method Detail

GetConfiguration

```
public string GetConfiguration() throws OZAPIException
```

Gets the log configurations from the server.

DownloadLog

`public Stream DownloadLog() throws OZAPIException`

Downloads the log file from the server.

DownloadLog

`public Stream DownloadLog(string fileName) throws OZAPIException`

Saves the log file downloaded from the server to the local.

Parameters:

fileName - The name of the file to save

SetConfiguration

`public void SetConfiguration(string logs) throws OZAPIException`

Changes the log configuration values.

Parameters:

logs - The log configuration value to modify, set in the form of "key=value"

GetFileNames

`public string[] GetFileNames() throws OZAPIException`

Gets the created log file.

Mail

This class provides methods related to the email sending function of OZ Server.

Constructor Detail

Mail

```
public Mail(string url, string id, string pw, bool autoLogin, bool useUSL)
```

Performs email sending function.

Parameters:

url - URL of OZ Server

id - User ID

pw - User Password

autoLogin - Whether to use the AutoLogin

useUSL - Whether to use the USL

Method Detail

AddAlias

```
public void AddAlias(string aliasName, NameValueCollection configMap)
```

Adds mail configuration information with the alias name configured in the OZ Server.

Parameters:

aliasName - The alias

configMap - The email configuration property value

GetAliasConfig

```
public NameValueCollection GetAliasConfig(string aliasName)
```

Gets the mail configuration information of the corresponding alias.

Parameters:

aliasName - The alias

GetMailAliasNames

```
public string[] GetMailAliasNames()
```

Gets all mail aliases configured in the amil.properties in the OZ Server.

ModifyAlias

```
public void ModifyAlias(string aliasName, string newAliasName, NameValueCollection  
configMap)
```

Modifies the name and mail configuration information of the corresponding alias.

Parameters:

aliasName - The alias

newAliasName - The new alias

configMap - Mail configuration property value

RemoveAlias

```
public void RemoveAlias(string aliasName)
```

Deletes the mail configuration information of the corresponding alias.

Parameters:

aliasName - The alias

Send

```
public void Send(string aliasName, string from, string fromUserName, string to,  
string cc, string bcc, string subject, string context, bool isHTML, string  
localFileFullPath, string fileName)
```

Sends asynchronous mail according to the mail information of the corresponding alias.

Parameters:

aliasName - The alias

from - The email address of a person who sent the email

fromUserName - The name who sent the email
to - The email address of a person to receive the email
cc - The cc email address
bcc - The bcc email address
subject - The email subject
context - The email content
isHTML - Whether to set the message format to HTML
localFileFullPath - The full path of the file for attachment
fileName - The attached file name

SendSync

```
public bool SendSync(string aliasName, string from, string fromUserName, string to, string cc, string bcc, string subject, string context, bool isHTML, string localFileFullPath, string fileName)
```

Sends synchronous mail according to the mail information of the corresponding alias, and gets whether the transmission succeeded.

Parameters:

aliasName - The alias
from - The email address of a person who sent the email
fromUserName - The name who sent the email
to - The email address of a person to receive the email
cc - The cc email address
bcc - The bcc email address
subject - The email subject
context - The email content
isHTML - Whether to set the message format to HTML
localFileFullPath - The full path of the file for attachment

fileName - The attached file name

MemoryStatus

This class provides member variables to get the information about the system's memory.

Member Variables

TotalMemory

```
public long TotalMemory
```

The total memory of server VM.

FreeMemory

```
public long FreeMemory
```

The free memory of server VM.

FreeMemoryPercentage

```
public long FreeMemoryPercentage
```

The free memory percent of server VM.

UseMemory

```
public long UseMemory
```

The used memory of server VM.

Module

This class provides properties and methods related to the data module processing function of OZ Server.

Constructor Detail

Module

```
public Module(string url, string id, string pw, bool autoLogin, bool useUSL)
```

Performs OZ Data Module related functions.

Parameters:

url - URL of OZ Server

id - User ID

pw - User Password

autoLogin - Whether to use the AutoLogin

useUSL - Whether to use the USL

Property Detail

MemoAllowed

```
public bool MemoAllowed
```

Allow memo or not.

Password

```
public string Password
```

Password for saving as OZD.

Method Detail

AddODIParameter

`public void AddODIPParameter(string odiName, string key, string value) throws OZAPIException`

Configures the ODI parameter values to create SDM files. Default ODI parameter values will be used if not configured.

Parameters:

odiName - ODI name

key - ODI parameter name

value - ODI parameter value

AddODIPParameter

`public void AddODIPParameter(string odiName, string item, string category, IDictionary parameters) throws OZAPIException`

Configures the ODI parameter values to create SDM files. Default ODI parameter values will be used if not configured. Used when creating SDM generation units for each parameter and when each different SDM file is needed depending on the ODI parameter values.

Parameters:

odiName - ODI name set in the Designer

item - ODI file name

category - ODI file path

parameters - Dictionary where the parameter keys and values are set.

AddParameter

`public void AddParameter(string key, string value) throws OZAPIException`

Configures form parameter values to create SDM files. Default form parameter values will be used if not configured.

Parameters:

key - Form Parameter Name

value - Form Parameter Value

AddReport

```
public IReportInfo AddReport(string itemName, string categoryName)
```

Adds reports in the process of making multiple reports into one OZD file.

Parameters:

itemName - Item name where reports will be added

categoryName - Category name of the item where reports will be added

AddReport

```
public IReportInfo AddReport(string itemName, string categoryName, string  
displayName)
```

Adds reports in the process of making multiple reports into one OZD file.

Parameters:

itemName - Item name where reports will be added

categoryName - Category name of the item where reports will be added

displayName - Report name to appear in the report tree of the OZ Viewer

GetODIParameter

```
public OZParameter[] GetODIParameter(string category, string odiName)
```

Gets ODI parameter from ODI

Parameters:

category - Category name

odiName - ODI name

GetOZD

```
public Stream GetOZD()
```

Creates an OZD with multiple reports and gets its Input Stream.

GetOZD

```
public Stream GetOZD(string item, string category, string[] urls) throws  
OZAPIException
```

Creates an OZD with multiple reports and gets its Input Stream.

Parameters:

item - Item name(OZR file name)

category - Category name

urls - The URL of the image file to save together with the OZD

GetOZD

```
public Stream GetOZD(string itemName, string categoryName)
```

Creates an OZD with multiple reports and gets its Input Stream.

Parameters:

itemName - Item name(OZR file name)

categoryName - Category name

GetOZQReportParameter

```
public OZODIItem[] GetOZQReportParameter(string category, string ozqItemName)
```

Gets ODI parameter from OZ Query Designer.

Parameters:

category - Category name

ozqItemName - ODI name

GetOZReportParameter

```
public OZReportParameter GetOZReportParameter(string category, string reportName)
```

Gets ODI parameter and form parameter from report.

Parameters:

category - Category name

reportName - Report name

GetOZReportDataSetInfo

```
public OZDataSetInfo[] GetOZReportDataSetInfo(string category, string reportName)  
throws OZCPEException
```

Gets the dataset information in the report. Returns in the order they are added to the data tree.

Parameters:

category - Category name

reportName - Report name

RedirectODIPath

```
public void RedirectODIPath(string odiName, string path)
```

Registers an ODI file to dynamically change when creating an OZD.

Parameters:

odiName - ODI name

path - ODI file path

SaveOZD

```
public void saveOZD(String fileName) throws OZCPEException
```

Creates an OZD with multiple reports and saves it in a specific path.

Parameters:

fileName - Category path where the file will be saved and file name

SaveOZD

```
public void saveOZD(String fileName, String item, String category) throws  
OZCPEException
```

Saves a report file as OZD file.

Parameters:

fileName - OZD file name including the save path

item - Report file(.ozr) name

category - Category name of the report file(.ozr)

SaveOZD

```
public void saveOZD(String fileName, String item, String category, String[] urls)  
throws OZCPEException
```

Saves a report file as OZD file.

Parameters:

fileName - OZD file name including the save path

item - Report file(.ozr) name

category - Category name of the report file(.ozr)

urls - URL of the image file to save together with the OZD file

Monitor

This class provides methods related to the monitor log function of OZ Server.

Constructor Detail

Monitor

```
public Monitor(string url, string id, string pw, bool autoLogin, bool useUSL)
```

Performs the real-time monitoring related functions of OZ Server.

Parameters:

url - URL of OZ Server

id - User ID

pw - User Password

autoLogin - Whether to use the AutoLogin

useUSL - Whether to use the USL

Method Detail

GetServerInfo

public OZServerInfo GetServerInfo() throws OZPAIException

Gets the version information of the server and respective modules.

GetMemoryInfo

public MemoryStatus GetMemoryInfo() throws OZAPIException

Gets the current memory usage status.(total memory, used memory, free memory)

DownloadLog

public Stream DownloadLog() throws OZAPIException

Gets the monitor log file saved in the Server.

OZAPIException

This class provides property related to the exception that occurs while using API.

Property Detail

Message

```
public string Message
```

Error message.

OZDataSetInfo

This class methods methods to get the information about data set added to the OZR file.

Method Detail

GetDataSetServiceType

```
public string GetDataSetServiceType()
```

Gets the service type. Returns one of ODI, DataService, FXData.

GetDataSetName

```
public string GetDataSetName()
```

Gets the dataset name.

GetDataSetFields

```
public OZAttributeList GetDataSetFields()
```

Gets the dataset field information.

GetMaxRowCount

```
public int GetMaxRowCount()
```

Gets the value of "Max Row Count" property.

GetUse

```
public bool GetUse()
```

Gets the value of "Use" property.

GetExportable

```
public bool GetExportable()
```

Gets the value of "Exportable" property.

OZODIItem

This class provides methods to get the information about ODI file.

Method Detail

GetCategory

```
public string GetCategory()
```

Gets the category name.

GetODIName

```
public string GetODIName()
```

Gets the ODI name.

GetODIFileName

```
public string GetODIFileName()
```

Gets the ODI file name.

GetODIParameters

```
public OZParameter[] GetODIParameters()
```

Gets the ODI parameter

OZParameter

This class provides methods to get the information about FORM parameters added to the OZR file.

Method Detail

GetName

```
public string GetName()
```

Gets the parameter name.

GetType

```
public int GetType()
```

Gets the parameter type.

GetValue

```
public string GetValue()
```

Gets the parameter value.

GetDescription

```
public string GetDescription()
```

Gets the parameter description.

OZReportParameter

This class provides methods to get the information about ODI parameters and FORM parameters added to the OZR file.

Method Detail

GetODIList

```
public OZODIItem[] GetODIList()
```

Gets the ODI information object.

GetFormParameters

```
public OZParameter[] GetFormParameters()
```

Gets the form parameter information object.

OZServerInfo

This class provides member variables to get the information about the system where the OZ Server.

Member Variables

OSName

```
public string OSName
```

The OS name.

OSVersion

```
public string OSVersion
```

The OS version.

FrameworkVersion

```
public string FrameworkVersion
```

The version of .NET Framework.

FrameworkVendor

```
public string FrameworkVendor
```

The vendor name of .NET Framework.

ServerVersion

```
public string ServerVersion
```

The version of the OZ server.

Service

This class provides methods related to the service function of OZ Server.

Constructor Detail

Service

```
public Service(string url, string id, string pw, bool autoLogin, bool useUSL)
```

Performs processing functions such as OZ Server information, status, processing, memory collection.

Parameters:

url - URL of OZ Server

id - User ID

pw - User password

autoLogin - Whether to use the AutoLogin

useUSL - Whether to use the USL

Method Detail

GrabageCollect

```
public void GrabageCollect() throw OZCPEException
```

Runs garbage collection, which with the help of a separate memory collection program collects the free memory on the JVM so that it can be recycled.

Ping

```
public bool Ping() throw OZCPEException
```

Returns whether the server is operating.

GetHandlerCount

```
public int GetHandlerCount() throw OZCPEException
```

Gets the number of concurrent multithread of the server.

Interface

OZ .NET Server API provides the following interface:

- [IReportInfo](#)

IReportInfo

This interface provides a method to set additional information about a report when generating an OZD with multiple reports.

Property Detail

ItemName

string ItemName()

Gets the item name.

CategoryName

string CategoryName()

Gets the category name of the item.

Method Detail

AddURL

void AddURL(params string[] urls)

Sets the URL of the image file included in the report as array.

Parameters:

urls - Image file URL

AddParameter

```
void AddParameter(string key, string value)
```

Configures the parameter values.

Parameters:

key - Parameter name

value - Parameter value

AddODIParameter

```
public void AddODIParameter(string odiName, string key, string value)
```

Configures the ODI parameter values.

Parameters:

odiName - ODI name

key - ODI parameter name

value - ODI parameter value

AddProperties

```
public void AddProperties(string key, string value)
```

Configures report save options.

Parameters:

key - Option name

value - Option value

AddFrameworkURL

```
void AddFrameworkURL(string fxName, string url, string charset)
```

Configures the framework URL of FX data.

Parameters:

fxName - FX data name

url - Framework URL

charset - Character set for parameter of framework URL

OZ Scheduler API

OZ Scheduler API provides [classes](#).

Class

OZ Scheduler API provides the following classes:

- [DirectExportResult](#)
- [DirectPrintResult](#)
- [DirectTaskResult](#)
- [FileInfo](#)
- [Program](#)
- [Publisher](#)
- [Scheduler](#)
- [ServerInfo](#)
- [TaskHolidayGroupInfo](#)
- [TaskHolidayInfo](#)
- [TaskResult](#)

DirectExportResult

This class provides the execution result of a task whose execution type is DirectExport.

Member Variables

ExportFileList

```
public string ExportFileList
```

The list of exported file names. Multiple file names are listed with "|" as a separator.

PageCount

```
public int PageCount
```

The number of exported pages.

DirectPrintResult

This class provides the execution result of a task whose execution type is DirectPrint.

Member Variables

PageCount

```
public int PageCount
```

The number of printed pages.

PageCopy

```
public int PageCopy
```

The number of copies.

PageRange

```
public string PageRange
```

The printed page range.

PrinterName

```
public string PrinterName
```

The printer name.

PrinterDriverName

```
public string PrinterDriverName
```

The printer driver name.

DirectTaskResult

This class provides the execution result of a task whose execution type is DirectExport or DirectPrint.

Member Variables

TaskID

```
public string TaskID
```

The ID of the task.

CompletedTime

```
public string CompletedTime
```

The completion time of task.

ExecuteTime

```
public string ExecuteTime
```

The run time of task.

IsSuccessful

```
public bool IsSuccessful
```

Whether the task was successful.

FormName

```
public string FormName
```

The name of the OZR file.

ErrorMessage

```
public string ErrorMessage
```

The error message.

FileInfo

This class provides information about files and folders.

Property Detail

IsDirectory

`public bool IsDirectory`

Whether it is a directory.

Name

`public string Name`

The name of the file or directory.

Size

`public long Size`

The size of the file or directory.

LastModified

`public DateTime LastModified`

The time of the last modification of the file or directory, calculated as a date and time based on 00:00:00 GMT on January 1, 1970.

Program

This class is related to the external program registration and management functions of OZ Scheduler.

Constructor Detail

Program

```
public Program(string ip, int port)
```

Performs external program registration and management functions in OZ Scheduler.

Parameters:

ip - The IP address

port - The port number

Method Detail

CreateFolder

```
public void CreateFolder(ServerInfo s, string folder)
```

Creates an external program folder in the OZ Scheduler.

Parameters:

s - The OZ Server information

folder - The name of folder

DownloadFile

```
public Stream DownloadFile(ServerInfo s, string file)
```

Downloads the registered external program files.

Parameters:

s - The OZ Server information

file - The name of the file to download

GetExternalProgramInfos

```
public FileInfo[] GetExternalProgramInfos(ServerInfo s, string folder)
```

Gets the list of external programs in the OZ Scheduler.

Parameters:

s - The OZ Server information

folder - The folder name

RemoveFiles

```
public void RemoveFiles(ServerInfo s, string folder, string[] files)
```

Deletes a registered external program file.

Parameters:

s - The OZ Server information

folder - The folder name

files - The name of the file to delete

RemoveFolder

```
public void RemoveFolder(ServerInfo s, string folder, bool forciblyRemove)
```

Deletes an external program folder in the OZ Scheduler.

Parameters:

s - The OZ Server information

folder - The folder name

forciblyRemove - Whether to delete the folder when it has sub-folders

UploadFile

```
public void UploadFile(ServerInfo s, string file, Stream inputStream)
```

Registers an external program in the OZ Scheduler.

Parameters:

s - OZ Server information

file - The name of the file to register

inputStream - The input stream of the file to register

Publisher

This is a class that manages the result files of tasks generated by the OZ Scheduler.

Constructor Detail

Publisher

```
public Publisher(string ip, int port)
```

Manages the results file of tasks generated by the OZ Scheduler.

Parameters:

ip - The IP address

port - The port number

Method Detail

CreateFolder

```
public void CreateFolder(ServerInfo s, string folder)
```

Creates a folder for saving the OZ Scheduler export file.

Parameters:

s - The OZ Server information

folder - The folder name

DownloadFile

```
public Stream DownloadFile(ServerInfo s, string file)
```

Downloads the registered the OZ Scheduler export file.

Parameters:

s - The OZ Server information

file - The downloaded file name

GetPublishedInfos

```
public FileInfo[] GetPublishedInfos(ServerInfo s, string folder)
```

Gets the list of export files from the OZ Scheduler.

Parameters:

s - The OZ Server information

folder - The folder name

RemoveFiles

```
public void RemoveFiles(ServerInfo s, string folder, string[] files)
```

Deletes the registered the OZ Scheduler export file.

Parameters:

s - The OZ Server information

folder - The folder name

files - The file name

RemoveFolder

```
public void RemoveFolder(ServerInfo s, string folder, bool forciblyRemove)
```

Deletes the folder where the OZ Scheduler export file is saved.

Parameters:

s - The OZ Server information

folder - The folder name

forciblyRemove - Whether to delete a folder if it has sub-folders.

Scheduler

This class creates report scheduling jobs and performs the result file and scheduler management functions.

Constructor Detail

Scheduler

```
public Scheduler(string ip, int port)
```

Creates report scheduling jobs and perform result file and scheduler management functions.

Parameters:

ip - The IP address

port - The port number

Method Detail

CreateTask

```
public string CreateTask(ServerInfo s, NameValueCollection configMap,  
NameValueCollection exportMap)
```

Creates a scheduling task based on the form file and gets the task ID.

Parameters:

s - The OZ Server information

configMap - Configuration option used when creating task

exportMap - Configuration option of the files exported from the OZ Scheduler

GetTaskInfos

```
public ScheduledTask[] GetTaskInfos(ServerInfo s)
```

Gets the result of the currently scheduling task.

Parameters:

s - The OZ Server information

GetTaskResult

```
public TaskResult[] GetTaskResult(ServerInfo s, string from, string to, string  
taskId, int rollingIndex)
```

Gets the results of task execution of a rolled file.

Parameters:

s - The OZ Server information

from - The start time

to - The end time

taskId - The task ID

rollingIndex - The file index

GetTaskResults

```
public TaskResult[] GetTaskResults(ServerInfo s, string from, string to, string  
taskId)
```

Gets the scheduling task result.

Parameters:

s - The OZ Server information

from - The start time

to - The end time

taskId - The task ID

GetTaskWaitTime

```
public int GetTaskWaitTime(ServerInfo s, String taskId)
```

Gets the amount of time in seconds the task waited.

Parameters:

s - The OZ Server information

taskId - The task ID

GetTaskWaitCount

```
public int GetTaskWaitCount()
```

Gets the number of tasks that are waiting.

RemoveTask

```
public void RemoveTask(ServerInfo s, string taskId)
```

Deletes a scheduling task.

Parameters:

s - The OZ Server information

taskId - The task ID

ModifyTask

```
public string ModifyTask(ServerInfo s, string taskId, NameValueCollection  
configMap, NameValueCollection exportMap)
```

Modifies the task properties.

Parameters:

s - The OZ Server information

taskId - The task ID

configMap - The OZ Scheduler configuration option

exportMap - The export information

PauseTask

```
public void PauseTask(ServerInfo s, string taskId)
```

Pauses a task.

Parameters:

s - The OZ Server information

taskId - The task ID

ResumeTask

```
public void ResumeTask(ServerInfo s, string taskId)
```

Resumes a paused task.

Parameters:

s - The OZ Server information

taskId - The task ID

Stop

```
public void Stop(ServerInfo s, bool forciblyStop)
```

Stops the Scheduler.

Parameters:

s - The OZ Server information

forciblyStop - Whether to forcibly stop the Scheduler

Export

```
public bool Export(ServerInfo s, NameValueCollection configMap,  
NameValueCollection exportMap)
```

Exports by using the viewer parameter as it is and then returns the result.

Parameters:

s - The OZ Server information

configMap - The OZ Scheduler configuration option

exportMap - The export information

MakePDF

```
public bool MakePDF(ServerInfo s, NameValueCollection configMap,  
NameValueCollection exportMap)
```

Performs PDF exporting without registering in the OZ Scheduler's work list and then returns the result.

Parameters:

s - The OZ Server information

configMap - The form information for export

exportMap - The PDF Export information

Print

```
public bool Print(ServerInfo s, NameValueCollection configMap, NameValueCollection  
printMap)
```

Prints by applying the viewer parameter.

Parameters:

s - The OZ Server information

configMap - The OZ Scheduler configuration information

printMap - The print information

GetCurrentStatus

```
string GetCurrentStatus()
```

Gets the scheduler's used memory, total memory, number of running viewers, and number of waiting viewers.

GetConfiguration

```
public NameValueCollection GetConfiguration(ServerInfo s)
```

Gets the Scheduler configuration.

Parameters:

s - The OZ Server information

ModifyConfiguration

```
public void ModifyConfiguration(ServerInfo s, NameValueCollection configMap)
```

Updates the configuration settings of the Scheduler.

Parameters:

s - The OZ Server information

configMap - The OZ Scheduler configuration option

SetReceiveTimeout

```
public void SetReceiveTimeout(int timeOut)
```

Sets the socket timeout.

Parameters:

timeout - The socket timeout

Ping

```
public bool Ping()
```

Gets whether the server is running.

GetSchedulingInfos

```
public string[] GetSchedulingInfos(string path)
```

Gets the list of OZS files in the set path.

Parameters:

path - The path from which the OZS file list will be received

ConvertSchedulingInfos

```
public void ConvertSchedulingInfos(string oldPath, string newPath)
```

Converts the 2.5 version OZS file into the latest version and saves it.

Parameters:

oldPath - The OZS file name or path to update

newPath - The path to save the updated OZS file

AddTaskHoliday

```
public bool AddTaskHoliday(TaskHolidayInfo value)
```

Adds task holiday information.

Parameters:

value - The task holiday information

ModifyTaskHoliday

```
public bool ModifyTaskHoliday(string old_key, TaskHolidayInfo new_value)
```

Modifies task holiday information.

Parameters:

old_key - The task holiday name

new_value - The new task holiday information

DeleteTaskHoliday

```
public bool DeleteTaskHoliday(string[] keys)
```

Deletes task holiday information.

Parameters:

keys - The array of the task holiday name

AddTaskHolidayGroup

```
public bool AddTaskHolidayGroup(TaskHolidayGroupInfo value)
```

Adds task holiday group information.

Parameters:

value - The task holiday group information

ModifyTaskHolidayGroup

```
public bool ModifyTaskHolidayGroup(string old_key, TaskHolidayGroupInfo new_value)
```

Modifies task holiday group information.

Parameters:

old_key - The task holiday group name

new_value - The new task holiday group information

DeleteTaskHolidayGroup

```
public bool DeleteTaskHolidayGroup(string key)
```

Deletes task holiday group information.

Parameters:

key - The task holiday group name

GetTaskHolidayInfos

```
public TaskHolidayInfos GetTaskHolidayInfos()
```

Gets task holiday information list.

GetTaskHolidayGroupInfos

```
public TaskHolidayGroupInfos GetTaskHolidayGroupInfos()
```

Gets task holiday information list.

SaveTaskHoliday

```
public void SaveTaskHoliday()
```

Save the task holiday information as an xml file.

DirectExport

```
public DirectExportResult DirectExport(ServerInfo s, NameValueCollection  
configMap, NameValueCollection exportMap)
```

Exports by applying the export information.

Parameters:

s - The OZ Server information

configMap - The OZ Scheduler configuration option

exportMap - The export information

DirectExportFiles

```
public Hashtable[] DirectExportFiles(ServerInfo s, NameValueCollection properties,  
NameValueCollection exportProperties, string localTempPath)
```

Exports by applying the export information.

Parameters:

s - The OZ Server information

properties - The OZ Scheduler configuration option

exportProperties - The export information

localTempPath - local temporary path

DirectExportResult

```
public DirectExportResult DirectExportResult(ServerInfo s, NameValueCollection  
properties, NameValueCollection exportProperties)
```

Exports by applying the export information.

Parameters:

s - The OZ Server information

properties - The OZ Scheduler configuration option

exportProperties - The export information

DirectExportByteArray

```
public Hashtable DirectExportByteArray(ServerInfo s, NameValueCollection  
configProperties, NameValueCollection exportProperties)
```

Exports by applying the export information.

Parameters:

s - The OZ Server information

configProperties - The OZ Scheduler configuration option

exportProperties - The export information

DirectPrint

```
public DirectPrintResult DirectPrint(ServerInfo s, NameValueCollection configMap,  
NameValueCollection printMap)
```

Prints by applying the export information.

Parameters:

s - The OZ Server information

configMap - The OZ Scheduler configuration option

printMap - The print information

GetFile

```
public void GetFile(ServerInfo s, string fileName, string targetFile)
```

Gets the file.

Parameters:

s - The OZ Server information

fileName - The file name

targetFile - The target file name

GetTaskProperties

```
public void GetTaskProperties(ServerInfo s, string TaskID, NameValueCollection  
configMap, NameValueCollection exportMap)
```

Gets the task configuration options(configMap) to apply when the scheduler creates a task and the file configuration options(exportMap) to apply when the scheduler exports a file.

Parameters:

s - The OZ Server information

taskID - The task ID

configMap - The form file information

exportMap - The export information

ServerInfo

This is a class that contains information about OZ Server.

Property Detail

URL

`public string URL`

The OZ Server URL.

ID

`public string ID`

The user ID.

Password

`public string Password`

The user password.

DirectoryName

`public string DirectoryName`

The directory name.

UserType

`public UserType UserType`

The user type.

TaskHolidayGroupInfo

This class defines a task holiday group.

Constructor Detail

TaskHolidayGroupInfo

```
public TaskHolidayGroupInfo(string name)
```

Defines the group of task holidays. The task holiday means the date when the task will not be performed.

Parameters:

name - The task holiday group name

Property Detail

Name

```
public string Name
```

The task holiday group name.

References

```
public string[] References
```

The task holiday name that the task holiday group refers to.

Method Detail

AddReference

```
public void AddReference(string[] names)
```

Registers the task holiday name that the task holiday group refers to.

Parameters:

names - The task holiday name

RemoveReferences

```
public void RemoveReferences(string[] names)
```

Deletes the task holiday name that the task holiday group refers to.

Parameters:

names - The task holiday name

ClearReferences

```
public void ClearReferences()
```

Deletes all task holiday names that the task holiday group refers to.

TaskHolidayInfo

This class defines a task holiday.

Constructor Detail

TaskHolidayInfo

```
public TaskHolidayInfo(string name)
```

Defines the task holiday. The task holiday means the date when the task will not be performed..

Parameters:

name - The task holiday name

Property Detail

Name

```
public string Name
```

The task holiday name.

BaseDay

```
public string BaseDay(string pattern)
```

Base day to be designated as task holiday.

Parameters:

pattern - The pattern of the base day

Start

```
public int Start
```

Sets how many days before the "baseday" will be holiday.

End

```
public int End
```

Sets how many days after the "baseday" will be holiday.

TaskResult

This class provides the results of the task execution.

Property Detail

TaskID

```
public string TaskID
```

The ID of the task.

CompletedTime

```
public string CompletedTime
```

The completion time of task.

IsSuccessful

```
public bool IsSuccessful
```

Whether the task execution is successful.

FormFileName

```
public string FormFileName
```

The name of the OZR file.

ParamInfo

```
public string ParamInfo
```

The information of the parameters set in the task.

SchedulingType

```
public string SchedulingType
```

The schedule type.(immediately, once, periodically)

ExportedFiles

```
public string ExportedFiles
```

The list of exported file names. Multiple file names are listed with "|" as a separator.

ErrorMessage

```
public string ErrorMessage
```

An error message.

OZ Repository API

OZ Repository API provides [classes](#) and [interfaces](#).

Class

OZ Repository API provides the following class:

- [RepositoryEX](#)

RepositoryEX

This is a class related to the repository function of OZ Server.

Constructor Detail

RepositoryEx

`public RepositoryEx(string url, string id, string pw, bool useUSL) throws OZCPEException`

Performs repository functions such as group/user management function, category/item authority setting function, and item history management function.

Parameters:

url - The URL of OZ Server

id - The user ID

pw - The user password

useUSL - Whether to use the USL or not. Can be omitted. If omitted, false is applied.

Method Detail

The RepositoryEX class provides the following methods:

- [Configuring the repository](#)
- [Creating groups and users](#)
- [Editing groups and users](#)
- [Deleting groups and users](#)
- [Getting groups and users](#)
- [Setting up group and user permissions](#)
- [Getting group and user permissions](#)
- [Creating categories and items](#)
- [Editing categories and items](#)
- [Deleting categories and items](#)
- [Getting categories and items](#)
- [Setting up category and item permissions](#)
- [Getting category and item permissions](#)
- [Using item version control](#)

Configuring the repository

To configure the repository using the RepositoryEX class, use the method below:

Method Detail

GetConfiguration

```
public NameValueCollection GetConfiguration()
```

Gets the repository settings.

SetConfiguration

```
public void SetConfiguration(NameValueCollection conf)
```

Sets up Repository configuration.

Parameters:

conf - The configuration to set up

Reload

```
public void Reload()
```

Reloads the Repository.

Creating groups and users

To create groups and users using the RepositoryEX class, use the method below:

Method Detail

CreateGroup

```
public string CreateGroup(string groupName, string parentGroupID, string description)
```

Creates a new group and returns its ID.

Parameters:

groupName - The group name

parentGroupID - The parent group ID

description - The group description

CreateUser

```
public string CreateUser(string userName, string password, string description)
```

Creates a new user and returns its ID.

Parameters:

userName - The user name

password - The user password

description - The user description

CreateUserInGroup

```
public string CreateUserInGroup(string uName, string pwd, string gID, string desc)  
throws OZCPEException
```

Creates a new user in the designated group and returns its ID.

Parameters:

uName - The user name

Pwd - The user password

gID - The group ID

desc - The user description

Editing groups and users

To edit groups and users using the RepositoryEX class, use the method below:

Method Detail

ModifyGroupName

```
public string ModifyGroupName(string groupID, string groupName)
```

Modifies the name of the user corresponding to the user ID.

Parameters:

groupID - The group ID

groupName - The group name

ModifyGroupDescription

```
public bool ModifyGroupDescription(string groupID, string description)
```

Modifies the description of the user corresponding to the user ID.

Parameters:

groupID - The group ID

description - The group description

ModifyUserName

```
public string ModifyUserName(string userID, string userName)
```

Modifies the name of the user corresponding to the user ID.

Parameters:

userID - The user ID

userName - The user name

ModifyUserPassword

```
public bool ModifyUserPassword(string userID, string oldPwd, string newPwd)
```

Modifies the password of the user corresponding to the user ID.

Parameters:

userID - The user ID

oldPwd - The old password

newPwd - The new password

ModifyUserDescription

```
public bool ModifyUserDescription(string userID, string description)
```

Modifies the description of the user corresponding to the user ID.

Parameters:

userID - The user ID

description - The user description

TransferGroup

```
public bool TransferGroup(string groupID, string targetGroupID)
```

Moves the designated group to another group.

Parameters:

groupID - The group ID to transfer

targetGroupID - The target group ID

TransferUser

```
public bool TransferUser(string userID, string newGroupID)
```

Moves the group of the assigned user.

Parameters:

userID - The user ID

newGroupID - The target group ID

Deleting groups and users

To delete groups and users using the RepositoryEX class, use the method below:

Method Detail

DeleteGroup

```
public bool DeleteGroup(string groupID)
```

Deletes the group corresponding to the group ID.

Parameters:

groupID - The group ID

DeleteUser

```
public bool DeleteUser(string userID)
```

Deletes the all information of the user corresponding to the user ID.

Parameters:

userID - The user ID

Getting groups and users

To get groups and users using the RepositoryEX class, use the method below:

Method Detail

GetGroupID

```
public string GetGroupID(string userID)
```

Gets the information of the group where the designated user belongs.

Parameters:

userID - The group ID

GetGroupInfo

```
public IGroupInfo GetGroupInfo(string groupID)
```

Gets the information of the group corresponding to the group ID.

Parameters:

groupID - The group ID

GetGroupInfoListInGroup

```
public IGroupInfo[] GetGroupInfoListInGroup(string groupID, bool isRecursive)
```

Gets the group information that was added to the group.(Not supported in NONE type repository)

Parameters:

groupID - The group ID

isRecursive - Whether to include sub-group

GetSubGroupInfos

```
public IGroupInfo[] GetSubGroupInfos(string groupID)
```

Gets the information of a sub group of the designated group.

Parameters:

groupID - The group ID

GetParentGroupInfo

```
public IGroupInfo GetParentGroupInfo(string groupID)
```

Gets the information of the parent group of the designated group.

Parameters:

groupID - The group ID

GetUserInfo

```
public IUserInfo GetUserInfo(string userID)
```

Gets the information of the user corresponding to the user ID.

Parameters:

userID - The user ID

GetUserInfos

```
public IUserInfo[] GetUserInfos()
```

Gets the information of all users.

GetUserInfos

```
public IUserInfo[] GetUserInfos(string groupID)
```

Gets the information of all users registered in the designated group ID.

Parameters:

groupID - The group ID

GetUserInfoListInGroup

```
public IUserInfo[] GetUserInfoListInGroup(string groupID, bool isRecursive)
```

Gets the user information.(Not supported in NONE type repository)

Parameters:

groupID - The group ID

isRecursive - Whether to include sub-group user

CheckPassword

```
public bool CheckPassword(string userID, string password)
```

Check that the user password is correct.

Parameters:

userID - The user ID

password - The password to be checked

Setting up group and user permissions

To set group and user permissions using the RepositoryEX class, use the method below:

Method Detail

AddGroupAdministrator

```
public bool AddGroupAdministrator(string userID, string groupID)
```

Adds the group administrator to the designated group.

Parameters:

userID - The user ID

groupID - The group ID

RemoveGroupAdministrator

```
public bool RemoveGroupAdministrator(string userID, string groupID)
```

Cancel the group administrator of the designated group.

Parameters:

userID - The user ID

groupID - The group ID

AddAdministrator

```
public bool AddAdministrator(string userID)
```

Adds the administrator to the designated user ID.(Not supported in NONE type repository)

Parameters:

userID - The user ID

RemoveAdministrator

```
public bool RemoveAdministrator(string userID)
```

Cancel the administrator of the designated user ID.(Not supported in NONE type repository)

Parameters:

userID - The user ID

EnableLogin

```
public void EnableLogin(string userID)
```

Sets to allow the user corresponding to the ID to log in.

Parameters:

userID - The user ID

DisableLogin

```
public void DisableLogin(string userID)
```

Sets to prohibit the user corresponding to the ID from logging in.

Parameters:

userID - The user ID

Getting group and user permissions

To get group and user permissions using the RepositoryEX class, use the method below:

Method Detail

GetGroupAdministrators

```
public IUserInfo[] GetGroupAdministrators(string groupID)
```

Gets the group administrator information of the designated group.

Parameters:

groupID - The group ID

IsGroupAdministrator

```
public bool IsGroupAdministrator(string userID, string groupID)
```

Check that the user corresponding to the user ID is the administrator of the group.

Parameters:

userID - The user ID

groupID - The group ID

IsAdministrator

```
public bool IsAdministrator(string userID)
```

Check that the user corresponding to the user ID is the administrator.

Parameters:

userID - The user ID

IsLoginEnabled

```
public bool IsLoginEnabled(string userID)
```

Gets whether the login of the user corresponding to the ID is enabled.

Parameters:

userID - The user ID

Creating categories and items

To create categories and items using the RepositoryEX class, use the method below:

Method Detail

CreateCategories

```
public string[] CreateCategories(string[] categoryNames, string[]  
parentCategoryIDs, string comment)
```

Creates a new category and returns the ID of the new category.

Parameters:

categoryName - The category name

parentCategoryIDs - The parent category ID

comment - The comment

CreateItem

```
public string[] CreateItem(string[] itemNames, string[] descriptions, string[]  
categoryIDs, bool[] isCompressed, Stream[] items, string comment, long[]  
createdTimes)
```

Creates an item by specifying the time and returns its ID.

Parameters:

itemNames - The item Name

descriptions - The item description

categoryIDs - The category ID

isCompressed - Whether to compress the item

items - The input stream of the new item

comment - The comment

createdTimes - Time the item was created. If the value is set to 0 or negative, it is applied to the current time.

CreateItems

```
public string[] CreateItems(string[] itemNames, string[] descriptions, string[]  
categoryIDs, bool[] isCompressed, Stream[] items, string comment)
```

Creates an item and returns its ID.

Parameters:

itemNames - The item Name

descriptions - The item description

categoryIDs - The category ID

isCompressed - Whether to compress the item

items - The input stream of the new item

comment - The comment

Editing categories and items

To edit categories and items using the RepositoryEX class, use the method below:

Method Detail

ModifyCategoryName

```
public string ModifyCategoryName(string categoryID, string newCategoryName, string comment)
```

Modifies the name of the category corresponding to the category ID.

Parameters:

categoryID - The category ID

newCategoryName - The category name

comment - The comment

ModifyCategoryDesc

```
public bool ModifyCategoryDesc(string categoryID, string description)
```

Modifies the description of the category. If the repository type is NONE, it only works if REPOSITORY_NONE_BACKUP=true.

Parameters:

categoryID - The category ID

description - The category description

ModifyItemName

```
public string ModifyItemName(string itemID, string newItemName, string comment)
```

Modifies an item name and returns it ID.

Parameters:

itemID - The item ID

newItemName - The item new name

comment - The comment

ModifyItemDescription

```
public bool ModifyItemDescription(string itemID, string description)
```

Modifies the description of the item corresponding to the item ID.

Parameters:

itemID - The item ID

description - The item description

ModifyItemDatetime

```
public bool ModifyItemDatetime(string itemID, long datetime)
```

Modifies an item time.(Not supported in NONE type repository)

Parameters:

itemID - The item ID

datetime - Time the item was changed. If the value is set to 0 or negative, it is applied to the current time.

TransferCategory

```
public bool TransferCategory(string categoryID, string targetCategoryID)
```

Moves the designated category to another category.

Parameters:

categoryID - The category ID

targetCategoryID - The target category ID

TransferItems

```
public bool TransferItems(string[] itemIDs, string targetCategoryID)
```

Moves the category of the designated item.

Parameters:

itemIDs - The item ID

targetCategoryID - The target category ID

EncryptFile

```
public OZAttributeList EncryptFile(string[] itemIDs)
```

Encrypts the OZR files and returns a list of files that failed to encrypt.

Parameters:

itemIDs - The item ID

EncryptFile

```
public OZAttributeList EncryptFile(string categoryID, bool isRecursive)
```

Encrypts the OZR files and returns a list of files that failed to encrypt.

Parameters:

categoryID - The category ID

isRecursive - Whether to include sub-categories

DecryptFile

```
public OZAttributeList DecryptFile(string[] itemIDs)
```

Decrypts the OZR files and returns a list of files that failed to decrypt.

Parameters:

itemIDs - The item ID

DecryptFile

```
public OZAttributeList DecryptFile(string categoryID, bool isRecursive)
```

Decrypts the OZR files and returns a list of files that failed to decrypt.

Parameters:

categoryID - The category ID

isRecursive - Whether to include sub-categories

SetUserDefinedResourceSync

```
public int SetUserDefinedResourceSync(string alias, string[] itemIDs, string  
comment, string[] errorMsg)
```

Synchronize OZR files with the user defined resource and returns the result.(1: complete success, 2: partial success, 3: fail)

Parameters:

alias - The alias name defined in ozudr.properties

itemIDs - The full path of the report

comment - The comment

errorMsg - The error message

Deleting categories and items

To delete categories and items using the RepositoryEX class, use the method below:

Method Detail

DeleteCategories

```
public bool[] DeleteCategories(string[] categoryIDs, bool[] toBeDestroyed, string comment)
```

Deletes the category corresponding to the designated category ID.

Parameters:

categoryIDs - The category ID

toBeDestroyed - Whether to permanently delete the category

comment - The comment

UnDeleteCategories

```
public bool[] UndeleteCategories(string[] categoryIDs, string comment)
```

Restores a deleted category.

Parameters:

categoryIDs - The category ID

comment - - The comment

DeleteItems

```
public bool[] DeleteItems(string[] itemIDs, bool[] toBeDestroyed, string comment)
```

Deletes the item from the repository.

Parameters:

itemIDs - The item ID

toBeDestroyed - Whether to permanently delete the item

comment - The comment

UnDeleteItems

```
public bool[] UndeleteItems(string[] itemIDs, string comment)
```

Restores an deleted item.

Parameters:

itemIDs - The item ID

comment - The comment

Getting categories and items

To get categories and items using the RepositoryEX class, use the method below:

Method Detail

GetCategoryID

```
public string GetCategoryID(string itemID)
```

Gets the category ID.

Parameters:

itemID - The item ID

GetCategoryInfo

```
public ICategoryInfo GetCategoryInfo(string categoryID)
```

Gets the information of the designated category.

Parameters:

categoryID - The category ID

GetCategoryInfos

```
public ICategoryInfo[] GetCategoryInfos(string categoryID)
```

Gets the information of all sub categories of the designated category.

Parameters:

categoryID - The category ID

GetItem

```
public Stream GetItem(string itemID, int version, bool isCompressed)
```

Gets the item of the designated version among the items corresponding to the item ID.

Parameters:

itemID - The item ID

version - The version

isCompressed - Whether to compress the item

GetItems

```
public Stream[] GetItems(string[] itemIDs, bool[] isCompressed, bool[]  
isObjStreams)
```

Gets the item corresponding to the item ID.

Parameters:

itemIDs - The item ID

isCompressed - Whether to compress the item

isObjStreams - Whether the ODI object stream is necessary

GetItems

```
public Stream[] GetItems(string[] itemIDs, long[] modifiedTimes, bool[]  
isCompressed, bool[] isObjStreams)
```

Gets the item corresponding to the item ID.

Parameters:

itemIDs - The item ID

modifiedTimes - Time the item was last modified

isCompressed - Whether to compress the item

isObjStreams - Whether the ODI object stream is necessary

GetItemInfo

```
public IItemInfo GetItemInfo(string itemID)
```

Gets the information of the item corresponding to the item ID.

Parameters:

itemID The item ID

GetItemInfos

```
public IItemInfo[] GetItemInfos(string categoryID)
```

Gets the information of the item corresponding to the category ID.

Parameters:

categoryID - The category ID

GetItemInfoListInCategory

```
public IItemInfo[] GetItemInfoListInCategory(string categoryID, bool isRecursive)
```

Gets the information of the item.(Not supported in NONE type repository)

Parameters:

categoryID - The category ID

isRecursive - Whether to include sub-categories

SearchItem

```
public ISearchResult[] SearchItem(string categoryID, string checkOutUser, string  
startDate, string endDate, SearchOptions options, string fileFilter, string  
criteria)
```

Gets the items corresponding to the set options and inquiry criteria among the items in the designated category.

Parameters:

categoryID - The category ID

checkoutUser - The Checkout user

startDate - The start date

endDate - The end date

options - The search options

fileFilter - The file extension

criteria - The search conditions

HasItem

```
public bool HasItem(string itemID)
```

Checks that the designated item exists.

Parameters:

itemID - The item ID

GetItemCount

```
public int GetItemCount(string categoryID)
```

Gets the number of all items belonging to the designated category.

Parameters:

categoryID - The category ID

Setting up category and item permissions

To set category and item permissions using the RepositoryEX class, use the method below:

Method Detail

ModifyGroupAuthorityToCategory

```
public bool ModifyGroupAuthorityToCategory(string groupID, string categoryID,
Authority permission)
```

Modifies permission of the given category.

Parameters:

groupID - The group ID

categoryID - The category ID

permission - The permission information

ModifyUserAuthorityToCategory

```
public bool ModifyUserAuthorityToCategory(string userID, string categoryID,
Authority permission)
```

Modifies the permission of the given user ID for the given category ID.

Parameters:

userID - The user ID

categoryID - The category ID

permission - The permission information

ModifyGroupAuthorityToItem

```
public bool ModifyGroupAuthorityToItem(string groupID, string itemID, Authority
permission)
```

Modifies the permission of the given group ID for the given item ID.

Parameters:

groupID - The group ID

itemID - The item ID

permission - The permission information

ModifyUserAuthorityToItem

```
public bool ModifyUserAuthorityToItem(string userID, string itemID, Authority permission)
```

Modifies the user permission for the given item ID.

Parameters:

userID - The user ID

itemID - The item ID

permission - The permission information

Getting category and item permissions

To get category and item permissions using the RepositoryEX class, use the method below:

Method Detail

GetGroupAuthorityToCategory

```
public Authority GetGroupAuthorityToCategory(string groupID, string categoryID)
```

Gets the authorization information of the group ID for the designated category ID.

Parameters:

groupID - The group ID

categoryID - The category ID

GetGroupInfosOfCategory

```
public IGroupInfo[] GetGroupInfosOfCategory(string categoryID, Authority permission)
```

Gets all group information with an authorization level of and higher than permission for the designated category ID.

Parameters:

categoryID - The category ID

permission - The permission information

GetGroupAuthorityToItem

```
public Authority GetGroupAuthorityToItem(string groupID, string itemID)
```

Gets the authorization information of the assigned group ID for the designated item.

Parameters:

groupID - The group ID

itemID - The item ID

GetGroupInfosOfItem

```
public IGroupInfo[] GetGroupInfosOfItem(string itemID, Authority permission)
```

Gets all group information with an authorization level of and higher than permission for the designated item ID.

Parameters:

itemID - The item ID

permission - The permission information

GetUserAuthorityToCategory

```
public Authority GetUserAuthorityToCategory(string userID, string categoryID)
```

Gets the authorization information of the assigned user ID for the designated category.

Parameters:

userID - The user ID

categoryID - The category ID

GetUserInfosOfCategory

```
public IUserInfo[] GetUserInfosOfCategory(string categoryID, Authority permission)
```

Gets all user information with an authorization level of and higher than permission for the designated category ID.

Parameters:

categoryID - The category ID

permission - Authorization information

GetUserAuthorityToItem

```
public Authority GetUserAuthorityToItem(string userID, string itemID)
```

Gets the authorization information of the assigned user ID for the designated item.

Parameters:

userID - The user ID

itemID - The item ID

GetUserInfosOfItem

```
public IUserInfo[] GetUserInfosOfItem(string itemID, Authority permission)
```

Gets all user information with an authorization level of and higher than permission for the designated item ID.

Parameters:

itemID - The item ID

permission - The permission information

GetCategoryInfosOfGroup

```
public ICategoryInfo[] GetCategoryInfosOfGroup(string groupID, string categoryID,
Authority permission)
```

Gets the list of all categories for which the designated group's authorization level is and higher than permission among the sub-categories in the designated category.

Parameters:

groupID - The group ID

categoryID - The category ID

permission - The permission information

GetCategoryInfosOfUser

```
public ICategoryInfo[] GetCategoryInfosOfUser(string userID, string categoryID,
Authority permission)
```

Gets the list of all categories for which the designated user's authorization level is and higher than permission among the sub-categories in the designated category.

Parameters:

userID - The user ID

categoryID - The category ID

permission - The permission information

GetItemInfosOfGroup

```
public IItemInfo[] GetItemInfosOfGroup(string groupID, string categoryID,
Authority permission)
```

Gets the list of all items for which the assigned group's authorization level is and higher than permission among the items in the designated category.

Parameters:

groupID - The group ID

categoryID - The category ID

permission - The permission information

GetItemInfosOfUser

```
public IItemInfo[] GetItemInfosOfUser(string userID, string categoryID, Authority permission)
```

Gets the list of all items for which the assigned user's authorization level is and higher than permission among the items in the designated category.

Parameters:

userID - The user ID

categoryID - The category ID

permission - The permission information

Using item version control

To use item version control function using the RepositoryEX class, use the method below:

Method Detail

CheckOut

```
public Stream[] CheckOut(string[] itemIDs, string[] checkOutFolders, long[] localFileTimes, bool[] isCompressed, string[] checkOutCmts)
```

Check out the designated item to the folder to be checked-out.

Parameters:

itemIDs - The item ID

checkOutFolders - The local folder name

localFileTimes - The local file time

isCompressed - Whether to compress the item

checkoutCmts - The comment

CheckIn

```
public bool[] CheckIn(string[] itemIDs, bool[] isCompressed, Stream[] items,  
string comment, bool[] keepCheckOut)
```

Check in the item by specifying the time.(Not supported in NONE type repository)

Parameters:

itemIDs - The item ID

isCompressed - Whether to compress the item

items - The input stream of the item

comment - The comment

keepCheckOut - Whether to keep the checked-out status

CheckIn

```
public bool[] CheckIn(string[] itemIDs, bool[] isCompressed, Stream[] items,  
string comment, bool[] keepCheckOut, long[] datetimes)
```

Check in the item by specifying the time.(Not supported in NONE type repository)

Parameters:

itemIDs - The item ID

isCompressed - Whether to compress the item

items - The input stream of the item

comment - The comment

keepCheckOut - Whether to keep the checked-out status

datetimes - Time the item was created. If the value is set to 0 or negative, it is applied to the current time.

UndoCheckOut

```
public Stream[] UndoCheckOut(string[] itemIDs, bool[] replaceItems, bool[] isCompressed)
```

Undo checking out of the designated item.

Parameters:

itemIDs - The item ID

replaceItems - Whether to get items in the local working folder

isCompressed - Whether to compress the item

Upload

```
public bool Upload(string categoryID, Stream in)
```

Upload the item.

Parameters:

categoryID - The category ID

in - the compressed stream

IsCheckOutUser

```
public bool[] isCheckOutUser(string[] ItemIDs) throws OZCPEException
```

Check that the item designated by the current user is checked out.

Parameters:

itemIDs - The item ID

GetHistoryInfos

```
public IHistoryInfo[] GetHistoryInfos(string itemID)
```

Gets the history information of the designated item.

Parameters:

itemID - The item ID

GetHistoryItemByDatetime

```
public Stream GetHistoryItemByDatetime(string itemID, long datetime, bool  
isCompressed, bool isObjStream)
```

Gets Most Recent Items specified time.(Not supported in NONE type repository)

Parameters:

itemID - The item ID

datetime - The base time to get item history

isCompressed - Whether to compress the item

isObjStream - Whether the obj item is objstream

GetHistoryItemListByDatetime

```
public IHistoryInfo[] GetHistoryItemListByDatetime(string itemID)
```

Gets the history information of the designated item the order date.(Not supported in NONE type repository)

Parameters:

itemID - The item ID

GetDeletedItemInfos

```
public IItemInfo[] GetDeletedItemInfos(string categoryID)
```

Gets the information of deleted items in the designated category.

Parameters:

categoryID - The category ID

GetDeletedItemHistoryInfos

```
public IHistoryInfo[] GetDeletedItemHistoryInfos(string itemID)
```

Gets the information of a deleted item.

Parameters:

itemID - The item ID

ModifyHistoryItemComment

```
public bool ModifyHistoryItemComment(string itemID, int itemVersion, string  
newComment) throws OZCPEException
```

Modifies the comment of the specified version.

Parameters:

itemID - The item ID

itemVersion - The item version

newComment - The comment

ModifyHistoryItemDatetime

```
public bool ModifyHistoryItemDatetime(string itemID, int itemVersion, long  
datetime)
```

Modifies the item time of the specified version.(Not supported in NONE type repository)

Parameters:

itemID - The item ID

itemVersion - The item version

datetime - Time the item was changed. If the value is set to 0 or negative, it is applied to the current time.

RemoveHistory

```
public bool RemoveHistory(string itemID, int version)
```

Deletes the history of specific version of the designated item.

Parameters:

itemID - The item ID

version - The item version

RollBack

```
public bool RollBack(string itemID, int version, string comment)
```

Recovers the designated item to the version of choice.

Parameters:

itemID - ID of the item to recover

version - The version to recover

comment - The comment

Interface

OZ Repository API provides the following Interfaces:

- [ICategoryInfo](#)
- [IGroupInfo](#)
- [IHistoryInfo](#)
- [IItemInfo](#)
- [ILoginInfo](#)
- [IRepository](#)
- [IRepositoryCategory](#)
- [IRepositoryGroup](#)
- [IRepositoryGroupAuthorityToCategory](#)
- [IRepositoryGroupAuthorityToItem](#)
- [IRepositoryHistory](#)
- [IRepositoryItem](#)
- [IRepositoryMultiLoginUser](#)
- [IRepositoryUser](#)
- [IRepositoryUserAuthorityToCategory](#)
- [IRepositoryUserAuthorityToItem](#)
- [IUserInfo](#)

ICategoryInfo

This interface provides category information.

Property Detail

ID

string ID

Gets the ID of the current category.

Name

string Name

Gets the name of the current category.

ParentID

string ParentID

Gets the ID of the parent category of the current category.

ParentName

string ParentName

Gets the name of the parent category of the current category.

Comment

string Comment

Gets the comment of the current category.

Description

string Description

Gets the description of the current category.

UpdateTime

long UpdateTime

Gets the update time of the current category.

CreatedUserID

string CreatedUserID

Gets the ID of the user who created the current category.

CreatedUserName

string CreatedUserName

Gets the name of the user who created the current category.

Permission

Authority Permission

Gets the authorization information of the current category.

IGroupInfo

This interface provides group information.

Property Detail

Administrators

IDictionary Administrators

Gets the administrator information of the selected group.

ID

string ID

Gets the ID of the selected group.

Name

string Name

Gets the name of the selected group.

ParentID

string ParentID

Gets the ID of the parent group of the selected group.

ParentName

string ParentName

Gets the name of the parent group of the selected group.

Description

string Description

Gets the description of the selected group.

UpdateTime

long UpdateTime

Gets the update time information of the selected group.

Permission

Authority Permission

Gets the authorization information of the selected group.

IHistoryInfo

This interface provides history information.

Property Detail

ItemID

string ItemID

Gets the item ID.

ItemName

string ItemName

Gets the item name.

Version

int Version

Gets the number of item versions.

CheckInTime

long CheckInTime

Gets the check-in time for item's each version.

CheckInUser

string CheckInUser

Gets the user ID of the items' each version.

CheckInComment

string CheckInComment

Gets the check-in comment of the item's each version.

ItemInfo

This interface provides item information.

Property Detail

IsCheckedOut

bool IsCheckedOut

Gets whether the selected item is checked out.

IsDeletable

bool IsDeletable

Gets whether the selected item is deleted.

ID

string ID

Gets the ID of the selected item.

Name

string Name

Gets the name of the selected item.

CategoryID

string CategoryID

Gets the ID of the category where the selected item belongs.

CategoryName

string CategoryName

Gets the name of the category where the selected item belongs.

UpdateTime

long UpdateTime

Gets the updatetime of the selected item.

Description

string Description

Gets the description of the selected item.

Comment

string Comment

Gets the comment of the selected item.

CreatedUserID

string CreatedUserID

Gets the ID of the user who created the selected item.

CreatedUserName

string CreatedUserName

Gets the name of the user who created the selected item.

CheckOutUser

string CheckOutUser

Gets the name of the user who checked out the selected item.

CheckOutLocalPath

string CheckOutLocalPath

Gets the name of the folder of the checked-out item.

Permission

Authority Permission

Gets the authorization information of the selected item.

ILoginInfo

This interface provides information of the logged in user.

Property Detail

ClientType

ClientType ClientType

Gets the client type requested by the client.

MACAddress

string MACAddress

Gets the MAC Address from the client.

IPAddress

string IPAddress

Gets the IP Address information from the client.

HDDSerialNo

string HDDSerialNO

Gets the HDD Serial number from the client.

ProtocolVersion

int ProtocolVersion

Gets the protocol version from the client.

UserID

string UserID

Gets the ID of the user logged in by the client.

UserName

string UserName

Gets the name of the user logged in by the client.

SessionID

string SessionID

Gets the session ID created upon the client login.

ClientIP

string ClientIP

Gets the client IP requested by the client.

IRepository

This interface provides repository setup functions.

Property Detail

Property

oz.util.IStringDictionary Property

Changes Repository settings. The values in the "repository.properties" can be configured.

Version

int Version

Defines Repository version and gets it.

Category

IRepositoryCategory Category

Gets the object when the Category Interface is implemented.

Item

IRepositoryItem Item

Gets the object when the Item Interface is implemented.

ItemHistory

`IRepositoryHistory ItemHistory`

Gets the object when the Item History Interface is implemented.

Group

`IRepositoryGroup Group`

Gets the object when the Group Interface is implemented.

User

`IRepositoryUser User`

Gets the object when the User Interface is implemented.

MultiLoginUser

`IRepositoryMultiLoginUser MultiLoginUser`

Gets the object when the Multi-login User Interface is implemented.

GroupAuthorityToCategory

`IRepositoryGroupAuthorityToCategory GroupAuthorityToCategory`

Gets the class when the Group Authorization Interface for categories is implemented.

GroupAuthorityToItem

`IRepositoryGroupAuthorityToItem GroupAuthorityToItem`

Gets the implemented object when the Group Authorization Interface for Items is implemented.

UserAuthorityToCategory

`IRepositoryUserAuthorityToCategory UserAuthorityToCategory`

Gets the implemented class when the User Authorization Interface for categories is implemented.

UserAuthorityToItem

`IRepositoryUserAuthorityToItem UserAuthorityToItem`

Gets the implemented class when the User Authorization Interface for items is implemented.

Method Detail

Open

`void Open()`

Open Repository when operates the server.

Close

`void Close()`

Closes Repository when closes the server.

Login

`object Login(string username, string password, ILoginInfo loginInfo, HttpContext ctx)`

Gets the session information logged in by calling the corresponding method when calling the client.

Parameters:

UserName - Name of the logged-in user

password - Password of the logged-in user

loginInfo - Log-in information

ctx - Current context when connecting to HTTP

Logout

`bool Logout(object session, string userName, ILoginInfo loginInfo, HttpContext ctx)`

Log out the user corresponding to the ID and gets whether the log-out result.

Parameters:

session - Session ID

userName - User name

loginInfo - Log-in information

ctx - Current context when connecting to HTTP

UpdateSessionState

```
void UpdateSessionState(object session, SessionState state)
```

Gets the session information logged in by calling the corresponding method when calling the client.

Parameters:

session - Session ID

state - Server Status

IRepositoryCategory

This interface provides functions for setting category.

Property Detail

AccessType

CategoryAccess AccessType

Gets the methods with access to the Category Interface as numerals.

Repository

IRepository Repository

Gets the object where the IRepository Interface is implemented.

Method Detail

CreateItems

```
string[] CreateItems(object session, string[] itemNames, string[] descriptions,  
string[] categoryIDs, Stream[] items, string comment, OZErrorCode[] errCodes,  
string[] errMsgs)
```

Creates a new item and returns the its ID.

Parameters:

session - Session ID

itemNames - ID of the new item

descriptions - Description of the new item

categoryIDs - Category ID of the new item

items - Input stream of the new item

comment - Comments

errCodes - Error code

errMsgs - Error message

CreateCategories

```
string[] CreateCategories(object session, string[] categoryNames, string[]  
parentCategoryIDs, string comment, OZErrorCode[] errCodes, string[] errMsgs)
```

Creates a new category and returns the ID of it..

Parameters:

session - Session ID

categoryNames - Name of the new category

parentCategoryIDs - parent category ID of the new category

comment - Comments

errCodes - Error code

errMsgs - Error message

ModifyCategoryName

```
string ModifyCategoryName(object session, string categoryID, string  
newCategoryName, string comment)
```

Modifies the name of the category corresponding to the category ID.

Parameters:

session - Session ID

categoryID - Category ID

newCategoryName - Category name

comment - Comments

DeleteCategories

```
bool[] DeleteCategories(object session, string[] categoryIDs, bool[]  
toBeDestroyed, string comment, OZErrorCode[] errCodes, string[] errMsgs)
```

Deletes the category corresponding to the category ID.

Parameters:

session - Session ID

categoryIDs - Category ID

toBeDestroyed - Whether to delete the category permanently

comment - Comments

errorCodes - Error code

errorMsgs - Error message

UndeleteCategories

```
bool[] UndeleteCategories(object session, string[] categoryIDs, string comment,
OZErrorCode[] errCodes, string[] errMsgs)
```

Recover the deleted category and gets whether the result.

Parameters:

session - Session ID

CategoryIDs - ID of the category to recover

comment - Comments

errorCode - Error code

errorMsg - Error message

GetItemCount

```
int GetItemCount(object session, string categoryID)
```

Gets the number of all items belonging to the designated category.

Parameters:

session - Session ID

categoryID - Category ID

GetItemInfos

```
IItemInfo[] GetItemInfos(object session, string categoryID)
```

Gets the information of all items belonging to the designated category.

Parameters:

session - Session ID

categoryID - Category ID

GetCategoryInfos

`ICategoryInfo[] GetCategoryInfos(object session, string categoryID)`

Gets the information of the designated category.

Parameters:

session - Session ID

categoryID - Category ID

GetCategoryID

`string GetCategoryID(object session, string itemID)`

Returns the ID of the category where the designated item exists.

Parameters:

session - Session ID

itemID - The item ID

GetCategoryInfo

`ICategoryInfo GetCategoryInfo(object session, string categoryID)`

Gets the information of the designated category.

Parameters:

session - Session ID

categoryID - Category ID

GetDeletedItemInfos

`IItemInfo[] GetDeletedItemInfos(object session, string categoryID)`

Imports the information of the deleted items in the designated category.

Parameters:

session - Session ID

categoryID - The Category ID

TransferItems

`bool TransferItems(object session, string[] itemIDs, string targetCategoryID)`

Moves the category of the assigned item and gets whether the result.

Parameters:

session - Session ID

itemIDs - Item ID

targetCategoryID - Target category ID

TransferCategory

`bool TransferCategory(object session, string categoryID, string targetCategoryID)`

Moves the designated category to another category and gets whether the result.

Parameters:

session - Session ID

categoryID - New category ID

targetCategoryID - Target category ID

IRepositoryGroup

This interface provides functions for setting group.

Property Detail

AccessType

`GroupAccess AccessType`

Gets the methods with access to the Group Interface as numerals.

Repository

IRepository Repository

Gets the object where the IRepository Interface is implemented.

Method Detail

CreateGroup

```
string CreateGroup(object session, string groupName, string parentGroupID, string description)
```

Creates a new group and returns its ID.

Parameters:

session - Session ID

groupName - Name of the new group

parentGroupID - ID of the parent group for the new group

description - Group description

ModifyGroupName

```
string ModifyGroupName(object session, string groupID, string groupName)
```

Changes the name of the group corresponding to the group ID and returns the changed group ID.

Parameters:

session - Session ID

groupID - Group ID

groupName - New group name

ModifyGroupDescription

`bool ModifyGroupDescription(object session, string groupID, string description)`

Modifies the description of the group corresponding to the group ID and returns whether the result.

Parameters:

session - Session ID

groupID - Group ID

description - New group description

DeleteGroup

`bool DeleteGroup(object session, string groupID)`

Deletes the group corresponding to the group ID and returns whether the result.

Parameters:

session - Session ID

groupID - ID of the group to delete

CreateUser

`string CreateUser(object session, string userName, string password, string groupID, string description)`

Creates a new user ID in the designated group and returns the ID.

Parameters:

session - Session ID

userName - User name

password - Password

groupID - Group ID

description - User description

GetUserInfos

`IUserInfo[] GetUserInfos(object session, string groupID)`

Gets the information of all users registered in the designated group ID.

Parameters:

session - Session ID

groupID - Group ID

GetGroupInfo

`IGroupInfo GetGroupInfo(object session, string groupID)`

Gets the information of the group corresponding to the group ID.

Parameters:

session - Session ID

groupID - Group ID

GetSubGroupInfos

`IGroupInfo[] GetSubGroupInfos(object session, string groupID)`

Gets the information of the subgroup of the designated group.

Parameters:

session - Session ID

groupID - Group ID

GetParentGroupInfo

`IGroupInfo GetParentGroupInfo(object session, string groupID)`

Gets the information of the parent group of the designated group.

Parameters:

session - Session ID

groupID - Group ID

GetGroupID

`string GetGroupID(object session, string userID)`

Gets the information of the group where the designated user belongs to.

Parameters:

session - Session ID

userID - User ID

AddGroupAdministrator

`bool AddGroupAdministrator(object session, string userID, string groupID)`

Add a group administrator to the designated group and returns whether the result.

Parameters:

session - Session ID

userID - User ID to add as a group admin.

groupID - ID Group ID to add as a group admin.

RemoveGroupAdministrator

`bool RemoveGroupAdministrator(object session, string userID, string groupID)`

Cancels the group Admin for the designated group and returns whether the result.

Parameters:

session - Session ID

userID - User ID

groupID - Group ID

IsGroupAdministrator

`bool IsGroupAdministrator(object session, string userID, string groupID)`

Checks that the user corresponding to the user ID is the administrator of the group.

Parameters:

session - Session ID

userID - The User ID

groupID - The Group ID

GetAdministrators

```
IUserInfo[] GetAdministrators(object session, string groupID)
```

Gets the group administrator information of the designated group.

Parameters:

session - Session ID

groupID - The Group ID

TransferUser

```
bool TransferUser(object session, string userID, string newGroupID)
```

Moves the group of the assigned user and gets whether the result.

Parameters:

session - Session ID

userID - The User ID

newGroupID - New group ID

TransferGroup

```
bool TransferGroup(object session, string groupID, string targetGroupID)
```

Moves the designated group to another group and gets whether the result.

Parameters:

session - Session ID

userID - The Group ID

targetGroupID - Target group ID

IRepositoryGroupAuthorityToCategory

This interface provides group permissions for categories.

Property Detail

AccessType

AuthorityAccess AccessType

Gets the methods with access to the Group Authorization Interface for a category as numerals.

Repository

IRepository Repository

Gets the object where the IRepository Interface is implemented.

Method Detail

Modify

bool Modify(object session, string categoryID, string groupID, Authority permission)

Modifies the authorization information of the designated category and group and returns whether the result.

Parameters:

session - Session ID

categoryID - Category ID

groupID - Group ID

permission - Permission information

GetAuthority

Authority GetAuthority(object session, string groupID, string categoryID)

Gets the group authorization of the assigned group for the designated category.

Parameters:

session - Session ID

groupID - Group ID

categoryID - Category ID

GetGroupInfos

IGroupInfo[] GetGroupInfos(object session, string categoryID, Authority permission)

Gets the information of all users with an authorization level of and higher than Permission for the designated category.

Parameters:

session - Session ID

categoryID - Category ID

permission - Permission information

GetItemInfos

IItemInfo[] GetItemInfos(object session, string groupID, string categoryID, Authority permission)

Gets the information of all items with an authorization level of and higher than Permission among the items in the designated category.

Parameters:

session - Session ID

groupID - Group ID

categoryID - Category ID

permission - Permission information

GetCategoryInfos

ICategoryInfo[] GetCategoryInfos(object session, string groupID, string categoryID, Authority permission)

Gets the list of all categories with an authorization level of and higher than Permission among the sub-categories of the designated category.

Parameters:

session - Session ID

groupID - Group ID

categoryID - Category ID

permission - Permission information

IRepositoryGroupAuthorityToItem

This interface provides group permissions for items.

Property Detail

AccessType

AuthorityAccess AccessType

Gets the methods with access to the Group Authorization Interface for items as numerals.

Repository

IRepository Repository

Gets the object where the IRepository is implemented.

Method Detail

Modify

`bool Modify(object session, string groupID, string itemID, Authority permission)`

Modifies the authorization information of the assigned group ID for the designated item ID and returns whether the result.

Parameters:

session - Session ID

groupID - The group ID

itemID - The item ID

permission - Permission information

GetAuthority

`Authority GetAuthority(object session, string groupID, string itemID)`

Gets the authorization information of the assigned group ID for the designated item ID.

Parameters:

session - Session ID

groupID - Group ID

itemID - Item ID

GetGroupInfos

`IGroupInfo[] GetGroupInfos(object session, string itemID, Authority permission)`

Gets the information of all groups with an authorization level of or higher than Permission for the designated item ID.

Parameters:

session - Session ID

itemID - Item ID

permission - Permission information

GetItemInfos

```
IItemInfo[] GetItemInfos(object session, string groupID, Authority permission)
```

Gets the information of all items with an authorization level of or higher than Permission for the designated group ID.

Parameters:

session - Session ID

groupID - Group ID

permission - Permission information

IRepositoryHistory

This interface provides information on the item's change history.

Property Detail

AccessType

HistoryAccess AccessType

Gets the methods with access to the Item History Interface as numerals.

Repository

IRepository Repository

Gets the object where the IRepository Interface is implemented.

Method Detail

RollBack

```
bool RollBack(object session, string itemID, int version, string comment)
```

Recover the designated item to the designated version.

Parameters:

session - Session ID

itemID - Item ID

version - Version to recover

comment - Comments

GetItem

```
Stream GetItem(object session, string itemID, int version)
```

Gets the item of the assigned version among the items corresponding to the item ID.

Parameters:

session - Session ID

itemID - Item ID

version - Version of the item to get

GetHistoryInfos

```
IHistoryInfo[] GetHistoryInfos(object session, string itemID)
```

Gets the history information of the designated item.

Parameters:

session - Session ID

itemID - Item ID

GetDeleteHistoryItemInfo

IHistoryInfo[] GetDeletedItemHistoryInfos(object session, string itemID)

Gets the list of deleted histories.

Parameters:

session - Session ID

itemIDs - Item ID

RemoveHistory

bool RemoveHistory(object session, string itemID, int version)

Deletes the history of a specific version of the assigned item.

Parameters:

session - Session ID

itemID - Item ID

version - The version

IRepositoryItem

This interface provides functions for setting item.

Property Detail

AccessType

CategoryAccess AccessType

Gets the methods with access to the Item Interface as numerals.

Repository

IRepository Repository

Gets the object where the IRepository Interface is implemented.

Method Detail

CreateItems

```
string[] CreateItems(object session, string[] itemNames, string[] descriptions,  
Stream[] items, string comment, OZErrorCode[] errCodes, string[] errMsgs)
```

Creates a new item and returns it ID.

Parameters:

session - Session ID

itemNames - New item ID

descriptions - New item description

items - New item input stream

comment - Comments

errCodes - Error code

errMsgs - Error message

ModifyItemName

```
string ModifyItemName(object session, string itemID, string newItemName, string  
comment)
```

Modifies the name of the item with the assigned item ID and returns the new item ID.

Parameters:

session - Session ID

itemID - Item ID

newItemName - New item ID

comment - Comments

DeleteItems

```
bool[] DeleteItems(object session, string[] itemIDs, bool[] toBeDestroyed, string  
comment, OZErrorCode[] errCodes, string[] errMsgs)
```

Deletes the assigned item from the repository and gets whether the result.

Parameters:

session - Session ID

itemIDs - Item ID

toBeDestroyed - Whether to delete it permanently

comment - Comments

errCodes - Error code

errMsgs - Error message

UnDeleteItems

```
bool[] UndeleteItems(object session, string[] itemIDs, string comment,  
OZErrorCode[] errCodes, string[] errMsgs)
```

Recover the deleted item and gets whether the result.

Parameters:

session - Session ID

itemIDs - Item ID

comment - Comments

errCodes - Error code

errMsgs - Error message

ModifyItemDescription

```
bool ModifyItemDescription(object session, string itemID, string description)
```

Modifies the description of the item corresponding to the item ID and gets whether the result.

Parameters:

session - Session ID

itemID - Item ID

description - The new description

GetItemInfo

```
IItemInfo GetItemInfo(object session, string itemID)
```

Gets the information of the item corresponding to the item ID.

Parameters:

session - Session ID

itemID - Item ID

HasItem

```
bool HasItem(object session, string itemID)
```

Gets whether the designated item exists.

Parameters:

session - Session ID

itemID - Item ID

GetItems

```
Stream[] GetItems(object session, string[] itemIDs, OZErrorCode[] errCodes,  
string[] errMsgs)
```

Gets the assigned item and after that the item stream.

Parameters:

session - Session ID

itemIDs - Item ID

errCodes - Error code

errMsgs - Error message

GetItems

```
Stream[] GetItems(object session, string[] itemIDs, long[] modifiedTimes,  
ItemOptions[] options, OZErrorCode[] errCodes, string[] errMsgs)
```

Gets the assigned item and after that the item stream.

Parameters:

session - Session ID

itemIDs - Item ID

modifiedTimes - Time the item was last modified

options - Options for processing error message

errCodes - Error code

errMsgs - Error message

CheckOut

```
Stream[] CheckOut(object session, string[] itemIDs, string[] localCheckOutFolders,  
long[] localFileTimes, OZErrorCode[] errCodes, string[] errMsgs)
```

Checks out the assigned item to the folder corresponding to the user ID and gets the item stream if the local file time is less then the item time.

Parameters:

session - Session ID

itemIDs - ID of the item to check out

localCheckOutFolders - Name of the local folder to check out

localFileTimes - Local file time of the item to check out

errCodes - Error code

errMsgs - Error message

CheckIn

```
bool[] CheckIn(object session, string[] itemIDs, Stream[] items, string comment,  
bool[] keepCheckOut, OZErrorCode[] errCodes, string[] errMsgs)
```

Checks in the assigned item corresponding to the user ID and gets whether the check in result.

Parameters:

session - Session ID

itemIDs - ID of the item to check in

items - Stream of the item to check in

comment - Comments

keepCheckOut - Whether to keep check-out

errCodes - Error code

errMsgs - Error message

UndoCheckOut

```
Stream[] UndoCheckOut(object session, string[] itemIDs, bool[] replace,  
OZErrorCode[] errCodes, string[] errMsgs)
```

Undo checking-out of the designated item.

Parameters:

session - Session ID

itemIDs - ID of the item to undo check-out

replaces - Whether to replace local files

errCodes - Error code

errMsgs - Error message

IsCheckOutUser

```
bool[] IsCheckOutUser(object session, string[] itemIDs)
```

Checks that the assigned user checked out the designated item.

Parameters:

session - Session ID

itemIDs - Item ID

IRepositoryMultiLoginUser

This interface provides functions for setting multi-login.

Property Detail

AccessType

MultiLoginAccess AccessType

Gets the methods with access to the Multi Login Interface as numerals.

Repository

IRepository Repository

Gets the object where the IRepository Interface is implemented.

Method Detail

DisableLogin

void DisableLogin(object session, string userID)

Set to prohibit the user corresponding to the ID from log-in.

Parameters:

session - Session ID

userID - The user ID

EnableUserLogin

void EnableLogin(object session, string userID)

Set to allow the user corresponding to the session ID to log in.

Parameters:

session - Session ID

userID - The user ID

IsLoginEnabled

bool IsLoginEnabled(object session, string userID)

Gets whether the user corresponding to the ID is allowed to log in.

Parameters:

session - Session ID

userID - The user ID

IRepositoryUser

This interface provides functions for setting user.

Property Detail

AccessType

UserAccess AccessType

Gets the methods with access to the User Interface as numerals.

Repository

IRepository Repository

Gets the object where the IRepository interface is implemented.

Method Detail

CreateUser

string CreateUser(object session, string userName, string password, string description)

Creates a new user and returns its ID.

Parameters:

session - Session ID

userName - The User name

password - Password

description - User description

ModifyName

string ModifyName(object session, string userID, string newUserName)

Modifies the name of the user corresponding to the user ID and returns the new user ID.

Parameters:

session - Session ID

userID - The User ID

newUserName - New user name

ModifyPassword

bool ModifyPassword(object session, string userID, string oldPassword, string newPassword)

Modifies the password of the designated user and returns whether the result.

Parameters:

session - Session ID

userID - The User ID

oldPassword - Old password

newPassword - New password

ModifyUserDescription

bool ModifyDescription(object session, string userID, string description)

Modifies the description of the user corresponding to the user ID and returns whether the result.

Parameters:

session - Session ID

userID - The User ID

description - The description

DeleteUser

bool DeleteUser(object session, string userID)

Deletes all information of the user corresponding to the user ID and returns whether the result.

Parameters:

session - Session ID

userID - The User ID

GetUserInfo

IUserInfo GetUserInfo(object session, string userID)

Gets the information of the user corresponding to the user ID.

Parameters:

session - Session ID

userID - User ID

CheckPassword

bool CheckPassword(object session, string userID, string password)

Check whether the user password is correct.

Parameters:

session - Session ID

userID - The User ID

password - Password to check

GetUserInfos

IUserInfo[] GetUserInfos(object session)

Gets the user information of all users.

Parameters:

session - Session ID

IsAdministrator

bool IsAdministrator(object session, string userID)

Check that the user corresponding to the user ID is the administrator of the Repository.

Parameters:

session - Session ID

userID - The user ID

IRepositoryUserAuthorityToCategory

This interface provides user permissions for categories.

Property Detail

AccessType

AuthorityAccess AccessType

Gets the methods with access to the User Authorization Interface for categories as numerals.

Repository

IRepository Repository

Gets the class where the OZRepository Interface is implemented.

Method Detail

Modify

bool Modify(object session, string userID, string categoryID, Authority permission)

Modifies the authorization information of the assigned user ID for the designated category ID and returns whether the result.

Parameters:

session - Session ID

userID - The user ID

categoryID - The category ID

permission - Permission information

GetAuthority

Authority GetAuthority(object session, string userID, string categoryID)

Gets the authorization information of the assigned user ID for the designated category ID.

Parameters:

session - Session ID

userID - The User ID

categoryID - The category ID

GetUserInfos

```
IUserInfo[] GetUserInfos(object session, string categoryID, Authority permission)
```

Gets the information of all users with an authorization level of or high than Permission for the designated category ID.

Parameters:

session - Session ID

categoryID - The category ID

permission - Permission information

GetItemInfos

```
IItemInfo[] GetItemInfos(object session, string userID, string categoryID,  
Authority permission)
```

Gets the list of all items for which the assigned user's authorization level is or higher than Permission among the items in the designated category.

Parameters:

session - Session ID

userID - The user ID

categoryID - The category ID

permission - Permission information

GetCategoryInfos

ICategoryInfo[] GetCategoryInfos(object session, string userID, string categoryID, Authority permission)

Gets the list of all categories for which the user's authorization level is or higher than Permission among the sub categories of the designated category.

Parameters:

session - Session ID

userID - The user ID

categoryID - The category ID

permission - Permission information

IRepositoryUserAuthorityToItem

This interface provides user permissions for items.

Property Detail

AccessType

AuthorityAccess AccessType

Gets the methods with access to the User Authorization Interface for items as numerals.

Repository

IRepository Repository

Gets the object where the IRepository Interface is implemented.

Method Detail

Modify

bool Modify(object session, string userID, string itemID, Authority permission)

Modifies a user's authorization information for the designated item and gets the result.

Parameters:

session - Session ID

userID - User ID

itemID - Item ID

permission - Permission information

GetAuthority

Authority GetAuthority(object session, string userID, string itemID)

Imports the a user's authorization information for the designated item.

Parameters:

session - Session ID

userID - User ID

itemID - Permission information

GetUserInfos

IUserInfo[] GetUserInfos(object session, string itemID, Authority permission)

Gets the information of all items for which the assigned user has an authorization level of or higher than Permission.

Parameters:

session - Session ID

itemID - Item ID

permission - Permission information

GetItemInfos

ItemInfo[] GetItemInfos(object session, string userID, Authority permission)

Gets the information of all items for which the assigned user has an authorization level of or higher than Permission.

Parameters:

session - Session ID

userID - Item ID

permission - Permission information

IUserInfo

This interface provides user information.

Property Detail

ID

string ID

Gets the ID of the selected user.

Name

string Name

Gets the name of the selected user.

GroupID

string GroupID

Gets the ID of the group where the selected user belongs.

GroupName

string GroupName

Gets the name of the group where the selected user belongs.

Description

string Description

Gets the description of the selected user.

UpdateTime

long UpdateTime

Gets the update time of the selected user.

IsLoggedIn

bool IsLoggedIn

Gets whether the selected user is logged in.

SessionID

string SessionID

Gets the session ID created when the selected user logged in.

IsLoginEnabled

bool IsLoginEnabled

Gets whether the login of the selected user is enabled.

LoginIP

string LoginIP

Gets the IP address of the PC where the selected user is logged in.

LastLoginTime

long LastLoginTime

Gets the last login time of the selected user.

Permission

Authority Permission

Gets the authorization information of the selected user.

User Defined Log API

The UDL(User Defined Log) is an interface to allow users to record databases or information related to the report in the user-defined type in a file or DB when referring data through the OZ server.

User Defined Log API provides [interfaces](#).

Interface

User Defined Log API provides the following Interfaces:

- [IClearQuestLogTarget](#)
- [IDMLogTarget](#)
- [IFileStoreLogTarget](#)
- [IHTTPStoreLogTarget](#)
- [IMonitorLogTarget](#)
- [IProcedureLogTarget](#)
- [IRDBLogTarget](#)
- [ISAPLogTarget](#)
- [ISOAPLogTarget](#)
- [IUDSLogTarget](#)
- [IUserDefinedLogger](#)
- [IUserDefinedLogTarget](#)
- [IXMLLogTarget](#)

IClearQuestLogTarget

This interface provides ClearQuest store related information.

Method Detail

ExecuteType

```
public string ExecuteType
```

Gets the Clear Quest execution type requested by the client.

QuerySubType

`public string QuerySubType`

Gets the QuerySub type requested by the client.

Query

`public string Query`

Gets the query statement requested by the client.

QueryExecuteTime

`public long QueryExecuteTime`

Gets the time taken to execute the query statement requested by the client.

QueryState

`public string QueryState`

Gets whether the query statement requested by the client has been successfully executed.

IDMLogTarget

This interface provides information about the data module.

Method Detail

ODIName

`public string ODIName()`

Gets the ODI name that executed query command.

DataSetName

```
public string DataSetName()
```

Gets the dataset name that executed query command.

DataStoreName

```
public string DataStoreName()
```

Gets the data store name that executed query command.

Parameters

```
public System.Connection.IDictionary Parameters
```

Gets OZ user-set parameters in the form of a dictionary.

IFileStoreLogTarget

This interface provides file store related information.

Method Detail

FilePath

```
public string FilePath
```

Gets the file path requested by the client.

FileAccessTime

```
public long FileAccessTime
```

Gets the time taken to get the file requested by the client

FileAccessState

```
public string FileAccessState
```

Gets whether the file requested by the client has successfully been executed.

IHTTPStoreLogTarget

This interface provides HTTP store related information.

Method Detail

URL

```
public string URL
```

Gets the URL requested by the client.

HTTPAccessTime

```
public long HTTPAccessTime
```

Gets the time taken to access the file in the HTTP path requested by the client.

HTTPAccessState

```
public string HTTPAccessState
```

Gets whether the file in the HTTP path requested by the client has successfully been executed.

IMonitorLogTarget

This interface provides monitor log related information.

Method Detail

ThreadID

```
public int ThreadID()
```

Gets the ODI name that executed the query statement.

Mark

```
public string Mark()
```

Gets the service start and end sign.

ThreadName

```
public string ThreadName()
```

Gets the service thread name.

ServiceTime

```
public long ServiceTime()
```

Gets the service start time and end time.(unit: ms)

FreeMemory

```
public long FreeMemory()
```

Gets the available memory size.(unit: byte)

TotalMemory

```
public long TotalMemory()
```

Gets the total JVM memory size.(unit: byte)

ServiceCode

```
public int ServiceCode()
```

Gets the service classification code.

ServiceStatus

```
public int ServiceStatus()
```

Gets the service status code.

ServiceParameter

```
public string ServiceParameter()
```

Gets the parameter value transmitted when the client requests service.

DBSessionID

```
public string DBSessionID()
```

Gets DBMS session ID.

ExecuteTime

```
public string ExecuteTime()
```

Gets the time taken to process service.(unit: ms)

ErrorCode

```
public string ErrorCode()
```

Gets the error code when error occurs.

ErrorMsg

```
public string ErrorMsg()
```

Gets the error message when error occurs.

ErrorStackTrace

```
public string ErrorStackTrace()
```

Gets Stack Trace when error occurs.

DBConns

```
public string DBConns()
```

The number of DB connections for an alias

IProcedureLogTarget

This interface provides information about procedure data.

Method Detail

Alias

`public string Alias`

Gets the DB alias name of the Connection Pool used by the executed ODI.

ProcedureName

`public string ProcedureName`

Gets the procedure name requested by the client.

ParameterList

`public IList ParameterList`

Gets the procedure parameter requested by the client in the form of `IList`.

ProcedureExecuteTime

`public long ProcedureExecuteTime`

Gets the time it takes for the procedure requested by the client to execute.

ProcedureState

`public string ProcedureState`

Gets whether the procedure requested by the client has been successfully executed.

IRDBLogTarget

This interface provides information about the RDB store.

Method Detail

Alias

`public string Alias`

Gets the DB alias name of the Connection Pool used by the executed ODI.

Query

`public string Query`

Gets the query command requested by the client.

PreparedQueryValue

`public IList PreparedQueryValue`

Gets the used value if a compiled query is used.

QueryState

`public string QueryState`

Gets whether the query command requested by the client has been successfully executed.

QueryExecuteTime

`public long QueryExecuteTime`

Gets the time it takes for the query command requested by the client to execute.

IsPrepared

`public bool IsPrepared`

Gets whether the compiled query has been used.

QueryType

`public QueryType QueryType`

Gets the type of the query executed and requested by the client.

ISAPLogTarget

This interface provides SAP store related information.

Method Detail

FunctionName

```
public string FunctionName
```

Gets the function name requested by the client.

FunctionType

```
public string FunctionType
```

Gets the function type requested by the client.

InputParameters

```
public IDictionary InputParameters
```

Gets the input parameter requested by the client in the form of IDictionary.

ResultSetTypes

```
public string ResultSetTypes
```

Gets the Resultset type requested by the client.

FunctionExecuteTime

```
public long FunctionExecuteTime
```

Gets the time taken to execute the function requested by the client.

FunctionState

```
public string FunctionState
```

Gets whether the function requested by the client has been successfully executed.

ISOAPLogTarget

This interface provides SOAP store related information.

Method Detail

ServiceName

```
public string ServiceName
```

Gets the service name requested by the client.

Port

```
public string Port
```

Gets the service port.

Operation

```
public string Operation
```

Gets the operation of the service requested by the client.

EndPoint

```
public string EndPoint
```

Gets the EndPoint that called the Service requested by the client.

RequestXML

```
public string RequestXML
```

Gets the XML content requested by the client.

ServiceExecuteTime

```
public long ServiceExecuteTime
```

Gets the time taken to execute the service requested by the client.

ServiceState

`public string ServiceState`

Gets whether the SOAP service requested by the client has successfully been executed.

IUDSLogTarget

This interface provides information about the user data store.

Method Detail

ExecuteCommand

`public string ExecuteCommand`

Gets the execution command requested by the client.

QueryType

`public QueryType QueryType`

Gets the type of the query command requested and executed by the client.

CommandExecuteTime

`public long CommandExecuteTime`

Gets the time it takes for the execution command requested by the client to execute.

CommandState

`public string CommandState`

Gets whether the execution command requested by the client has been successfully executed.

IUserDefinedLogger

This interface provides methods to implement User Defined Log.

Method Detail

Init

```
public void Init(OZConnection con, OZLog log)
```

Defines the target to leave user-defined log when operating the OZ server.

Parameters:

con - The DB Connection information and type of log to record when recording user-defined log in the DB

log - The file information and type of log to record when recording user-defined log in the file

Trace

```
public void Trace(IUserDefinedLogTarget target, OZConnection con, OZLog log)
```

Defines the target content to record in the log.

Parameters:

target - User defined log target

con - The DB Connection information and type of log to record when recording user-defined log in the DB

log - The file information and type of log to record when recording user-defined log in the file

DbAlias

```
public string DbAlias
```

Alias in OZ server connection pool to log for logging user defined log into DB.

IUserDefinedLogTarget

This interface provides basic information of OZ UDL.

Method Detail

IPAddress

```
public string IPAddress()
```

Gets the OZ server IP address requested by the client.

HttpRequest

```
public HttpRequest HttpRequest()
```

Gets the HttpRequest requested by the client.

HttpContext

```
public HttpContext HttpContext()
```

Gets the HttpContext from the client.

UserID

```
public string UserID()
```

Gets the OZ server user ID that executed the query command from the client.

IXMLLogTarget

This interface provides XML store related information.

Method Detail

URL

```
public string URL
```

Gets the file path of URO requested by the client.

URLAccessTime

`public long URLAccessTime`

Gets the time taken to get the file in the URL requested by the client.

URLAccessState

`public string URLAccessState`

Gets whether the file in the URL requested by the client has been successfully executed.

User Defined Resource API

When create a resource field of a report, UDR(User Defined Resource) enables you to set the resource field values using resource information in database without setting the each resource field values using OZ Report Designer.

User Defined Resource API provides [classes](#) and [interfaces](#).

Class

User Defined Resource API provides the following classes:

- [OZUDRInfo](#)
- [OZUDRResult](#)
- [Resource](#)

OZUDRInfo

This class provides methods to get UDR information.

Constructor Detail

OZUDRInfo

```
public OZUDRInfo(string category, string reportname, string[] language, Resource[] resource)
```

Gets the UDR information.

Parameters:

category - Category name

reportname - Report name

language - Language list

resource - Resource information object

Property Detail

Category

`public string Category`

Category name.

Reportname

`public string Reportname`

Report name.

Language

`public string[] Language`

Language list.

Resource

`public Resource[] Resource`

Resource information.

OZUDRResult

This class provides methods to get the UDR execution result.

Constructor Detail

OZUDRResult

`public OZUDRResult(bool isUpdate, OZUDRInfo info)`

Gets the UDR execution result.

Parameters:

isUpdate - Whether to update or not

info - Object that have new user defined resource information

Property Detail

Update

`public bool Update`

Whether to update or not.

UDRInfo

`public OZUDRInfo UDRInfo`

Object that have new user defined resource information.

Resource

This class provides methods to get resource information.

Constructor Detail

Resource

`public Resource(string name, string desc, string[] value)`

Gets resource information.

Parameters:

name - Resource field name

desc - Resource description

value - Resource value for each language

Property Detail

Name

`public string Name`

Resource field name.

Desc

`public string Desc`

Resource description.

Value

`public string[] Value`

Resource values for each language.

DescHash

`public Hashtable DescHash`

Object with key, value in the hash for parsing resource description.

Interface

User Defined Resource API provides the following interface:

- [IOZUserDefinedResource](#)

IOZUserDefinedResource

This interface provides resource synchronization functions.

Method Detail

SyncResource

```
public OZUDRRResult SyncResource(OZConnection ozconn, OZLog cat, OZUDRInfo info)
```

Synchronize resource.

Parameters:

ozconn - DB Connection

cat - Category to be logged in server log

info - Resource information object in the report