

.	<b>API .....</b>	<b>4</b>
	Class Cache .....	6
	Class ConnectionPool .....	11
	Class DataBind .....	23
	Class Log .....	27
	Class Mail .....	31
	Class Module .....	38
	Class Monitor .....	50
	Class Service .....	55
	Class Viewer .....	59
	Class Servlet .....	67
.	<b>API .....</b>	<b>80</b>
	Class Program .....	82
	Class Publisher .....	88
	Class Scheduler .....	92
	Class TaskHolidayInfo .....	139
	Class TaskHolidayGroupInfo .....	143
.	<b>API .....</b>	<b>147</b>
	Interface Repository .....	148
	Class RepositoryEX .....	197
	.....	222

- . **API** ..... **229**
  - OZLauncherDll ..... 230
  
- . **RDB Store DataAction** ..... **232**
  - RDB Store DataAction ..... 233
  - RDB Store DataAction ..... 234
  
- . **User Data Store** ..... **236**
  - UDS ..... 237
  
- . **User Security Logic** ..... **238**
  - USL ..... 239
  - USL ..... 241
  - USL ..... 244
  
- . **User Defined Log** ..... **255**
  - UDL ..... 256
  - UDL ..... 257
  - UDL ..... 271
  
- . ..... **303**
  - ..... 304
  - ..... 304

C ..... 306

**Appendix 1. SchedulerCom ..... 323**

## API

- Class Cache
- Class ConnectionPool
- Class DataBind
- Class Log
- Class Mail
- Class Module
- Class Monitor
- Class Service
- Class Viewer
- Class Servlet

API

API

Cache	
Connection Pool	JDBC/ODBC Pool
DataBind	
Log	
Mail	
Module	
Monitor	
Service	
Viewer	
Servlet	OZD

API

가

ozsfw50.jar	Scheduler server
log4.jar	Server (API) Log classpath "log4.jar"

## Class Cache

### Constructor Summary

- `Cache(String ip, int port, String id, String pw, boolean bAutoLogin, boolean useUSL)`
- `Cache(String url, String id, String pw, boolean bAutoLogin, boolean useUSL)`

### Method Summary

- `SortProperties getCacheConfiguration()`
- `void setCacheConfiguration(SortProperties p)`

### Constructor Detail

---

	<code>//Daemon</code>	-	<code>TCP Server</code>
<b>Prototype</b>	<code>public Cache(String ip, int port, String id, String pw, boolean bAutoLogin, boolean useUSL)</code>		
	<code>//Servlet</code>	-	<code>HTTP Server</code>
	<code>public Cache(String url, String id, String pw, boolean bAutoLogin, boolean useUSL)</code>		
<b>Argument</b>	<i>url</i>	Servlet	URL ex) <code>String url = "http://127.0.0.1/oz/server";</code>
	<i>ip</i>	Daemon	IP ex) <code>String ip = "127.0.0.1";</code>
	<i>port</i>	Daemon	<code>ex) int port = 8003;</code>
	<i>id</i>		<code>ex) String id = "admin";</code>

---

<i>pw</i>	ex) String pw = "admin";
<i>bAutoLogin</i>	ex) boolean bAutoLogin = true;
<i>useUSL</i>	USL ex) boolean useUSL = false;

## Method Detail

### ■ **getCacheConfiguration**

**Prototype** public SortProperties getCacheConfiguration() throws OZCPEException

**Definition** "SortProperties" key 가 . 가

### ■ **setCacheConfiguration**

**Prototype** public void setCacheConfiguration(SortProperties p) throws OZCPEException

**Definition** key "SortProperties"

**Argument** *p*

## Class

### ■ **OZCPEException(oz.framework.cp.OZCPEException)**

API Exception . API OZCPEException

#### ▪ **getMessage**

**Prototype** public String getMessage()

**Definition** 가 .

- `getErrorCode`

<b>Prototype</b>	<code>public int getErrorCode()</code>
------------------	--

<b>Definition</b>	가
-------------------	---

- **SortProperties(oz.util.SortProperties.java)**

`getCacheConfiguration(), setCacheConfiguration()`

- `getproperty / get`

<b>Prototype</b>	<code>public synchronized String getProperty(String key)</code> <code>public synchronized Object get(String key)</code>
------------------	--

<b>Definition</b>	key 가
-------------------	-------

- `setproperty / put`

<b>Prototype</b>	<code>public synchronized Object setProperty(String key, String value)</code> <code>public synchronized Object put(Object key, Object value)</code>
------------------	--

<b>Definition</b>	key (value)
-------------------	-------------

- Key

`getProperty() setProperty() key`

Key	Value	
<b>Active</b>	"true" "false"	ex) <code>p.setProperty("datamodule.active", "false");</code>
<b>CACHE_FILE_PATH</b>		ex) <code>p.setProperty("CACHE_FILE_PATH", "%OZ_HOME%/cache");</code>
<b>DM_CACHE_FILE_PATH</b>		Data Module ex) <code>p.setProperty("DM_CACHE_FILE_PATH", "%OZ_HOME%/cache_dm/");</code>

<b>memoryCacheValidTime</b>		( : ) ex) p.setProperty("datamodule.memoryCacheValidTime", "100");
<b>diskCacheValidTime</b>		( : ) ex) p.setProperty("datamodule.diskCacheValidTime", "100");
<b>FreeMemoryPercentage</b>		ex) p.setProperty("datamodule.freeMemoryPercentage", "20");

:"  
cachemngr.properties"

### Sample : CacheSample.java

```

package sample;

import oz.framework.api.Cache;
import org.apache.log4j.*;

import oz.util.SortProperties;

public class CacheSample {
    public static void main(String[] args) {
        // 가 가
        BasicConfigurator.configure();

        // OZServer Info.
        /**
        // Daemon
        String IP = "127.0.0.1"; // 가 IP
        int PORT = 8003; // 가 TCP
        /**
        // Servlet
        String URL = "http://www.oz.com/oz/server"; //Servlet 가 URL
        /**/
        // User Info.
        String ID = "admin"; //default
        String PWD = "admin"; //default

        Cache cache = null;
        try {

```

```
    /**
     * Daemon
     * cache = new Cache(IP, PORT, ID, PWD, false, false);
     */
    /**
     * Servlet
     * cache = new Cache(URL, ID, PWD, false, false);
     */

    SortProperties p = new SortProperties();

    /**
     *
     * (setCacheConfigurati on)
     * p.setProperty("CACHE_FILE_PATH", "%OZ_HOME%/cache");
     *
     * p.setProperty("DM_CACHE_FILE_PATH", "%OZ_HOME%/cache_dm");
     * DataModule
     * p.setProperty("datamodule.active", "true");
     *
     * p.setProperty("datamodule.memoryCacheVal idTime", "1000");
     *
     * p.setProperty("datamodule.diskCacheVal idTime", "1000");
     *
     * p.setProperty("datamodule.freeMemoryPercentage", "21");
     *
     * cache.setCacheConfigurati on(p);

    /**
     *
     * 가 (getCacheConfigurati on)
     * p = cache.getCacheConfigurati on();
     */
    p.list(System.out);
}
catch(Excepti on e) {
    e.printStackTrace();
}
}
```

## Class ConnectionPool

### Constructor Summary

- `ConnectionPool(String ip, int port, String id, String pw, boolean bAutoLogin, boolean useUSL)`
- `ConnectionPool(String url, String id, String pw, boolean bAutoLogin, boolean useUSL)`

### Method Summary

- `void addPool(ConnectionPoolInfo pool)`
- `void removePool(String pool)`
- `ConnectionPoolInfo[] getPoolInfoList()`
- `ConnectionPoolStatus[] getPoolStatusList()`
- `ConnectionPoolInfo getPoolInfo(String alias)`
- `void save()`

### Constructor Detail

	<code>//Daemon</code>	-	<code>TCP Server</code>
<b>Prototype</b>	<code>public ConnectionPool (String ip, int port, String id, String pw, boolean bAutoLogin, boolean useUSL)</code>		
	<code>//Servlet</code>	-	<code>HTTP Server</code>
	<code>public ConnectionPool (String url, String id, String pw, boolean bAutoLogin, boolean useUSL)</code>		
<b>Argument</b>	<i>url</i>	Servlet	URL ex) <code>String url = "http://127.0.0.1/oz/server";</code>
	<i>ip</i>	Daemon	IP ex) <code>String ip = "127.0.0.1";</code>

<i>port</i>	Daemon ex) int port = 8003;
<i>id</i>	ex) String id = "admin";
<i>pw</i>	ex) String pw = "admin";
<i>bAutoLogin</i>	ex) boolean bAutoLogin = true;
<i>useUSL</i>	USL ex) boolean useUSL = false;

## Method Detail

### ■ addPool

<b>Prototype</b>	public void addPool (ConnectionPoolInfo pool) throws OZCPEXception
<b>Definition</b>	ConnectionPool 가 . 가 ConnectionPool "ConnectionPoolInfo" 가 .
<b>Argument</b>	<i>pool</i> 가 ConnectionPool ConnectionPoolInfo

### ■ removePool

<b>Prototype</b>	public void removePool (String pool) throws OZCPEXception
<b>Definition</b>	ConnectionPool .
<b>Argument</b>	<i>pool</i> ConnectionPool

### ■ getPoolInfoList

<b>Prototype</b>	public ConnectionPoolInfo[] getPoolInfoList() throws OZCPEXception
<b>Definition</b>	ConnectionPool ConnectionPoolInfo 가 .

### ■ getPoolStatusList

<b>Prototype</b>	public ConnectionPoolStatus[] getPoolStatusList() throws OZCPEXception
------------------	--

<b>Definition</b>	ConnectionPool	가	.
-------------------	----------------	---	---

■ **getPoolInfo**

<b>Prototype</b>	public ConnectionPoolInfo getPoolInfo(String alias) throws OZCPEXception		
<b>Definition</b>	ConnectionPool	ConnectionPoolInfo	가 .
<b>Argument</b>	<i>alias</i>	ConnectionPool	

■ **save**

<b>Prototype</b>	public void save() throws OZCPEXception		
<b>Definition</b>	ConnectionPool		.

**Class**

■ **ConnectionPoolInfo(oz.framework.db.ConnectionPoolInfo.class)**

가

▪ **getAlias**

<b>Prototype</b>	public String getAlias()		
<b>Definition</b>	Pool	alias	가 .

▪ **setAlias**

<b>Prototype</b>	public void setAlias(String _alias)		
<b>Definition</b>	Pool	alias	.
<b>Argument</b>	<i>_alias</i>	Alias	

▪ **getVendor**

<b>Prototype</b>	public String getVendor()		
<b>Definition</b>	Pool	Vendor	가 .

▪ **setVendor**

<b>Prototype</b>	public void setVendor(String vendorName)		
------------------	--	--	--

<b>Definition</b>	Pool	DB Vendor	.
<b>Argument</b>	<i>vendorName</i>	DB Vendor	

▪ **getDriver**

<b>Prototype</b>	public String getDriver()		
<b>Definition</b>	Pool	Driver	가 . ) oracle.jdbc.driver.OracleDriver

▪ **setDriver**

<b>Prototype</b>	public void setDriver(String _driver)		
<b>Definition</b>	Pool	Driver	.
<b>Argument</b>	<i>_driver</i>	Driver	

▪ **getURL**

<b>Prototype</b>	public String getURL()		
<b>Definition</b>	Pool	URL	가 .

▪ **setURL**

<b>Prototype</b>	public void setURL(String _url)		
<b>Definition</b>	Pool	URL	.
<b>Argument</b>	<i>_url</i>	URL	

▪ **getInitConns**

<b>Prototype</b>	public int getInitConns()		
<b>Definition</b>	Pool	가	.

▪ **setInitConns**

<b>Prototype</b>	public void setInitConns(int _initConns)		
<b>Definition</b>	Pool		
<b>Argument</b>	<i>_initConns</i>		

▪ **getMaxConns**

<b>Prototype</b>	public int getMaxConns()		
------------------	--------------------------	--	--

**Definition** Pool 가 .

▪ **setMaxConns**

**Prototype** public void setMaxConns(int \_maxConns)

**Definition** Pool

**Argument** \_maxConns

▪ **getTimeout**

**Prototype** public int getTimeout()

**Definition** 가 timeout 가 . ( : sec)

▪ **setTimeout**

**Prototype** public void setTimeout(int \_timeout)

**Definition** 가 timeout .

**Argument** *\_timeout* timeout

▪ **getDoConnectionCheck**

**Prototype** public boolean getDoConnectionCheck()

**Definition** 가

▪ **setDoConnectionCheck**

**Prototype** public void setDoConnectionCheck(boolean \_check)

**Definition** 가 .

**Argument** *\_check*

▪ **getTestQueryString**

**Prototype** public String getTestQueryString()

**Definition** 가 .

▪ **setTestQueryString**

**Prototype** public void setTestQueryString(String \_testQuery)

**Definition** .

**Argument** *\_testQuery*



<b>Prototype</b>	<code>public void setAutoCommit(String _commit)</code>
<b>Definition</b>	AutoCommit
<b>Argument</b>	<code>_commit</code> AutoCommit : "true", "false"

▪ `getProperties`

<b>Prototype</b>	<code>public Properties getProperties()</code>
<b>Definition</b>	가 Properties 가

▪ `setProperties`

<b>Prototype</b>	<code>public void setProperties(Properties _props)</code>
	가 Propeties )
<b>Definition</b>	<pre> ConnectionPoolInfo cpi = new ConnectionPoolInfo(); ... Driver driver = loadDriver(cpi.getDriver()); Connection conn = driver.connect(cpi.getURL(), cpi.getProperties()); </pre>
<b>Argument</b>	<code>_props</code> properties

▪ `getKey`

<b>Prototype</b>	<code>public String getKey()</code>
	Pool unique Key 가
<b>Definition</b>	connection pool key unique dbconfig.xml Vendor Items Key

▪ `setKey`

<b>Prototype</b>	<code>public void setKey(String _key)</code>
	unique Key
<b>Definition</b>	dbconfig.xml Vendor Items Key
<b>Argument</b>	<code>_key</code> key

▪ `getItems`

<b>Prototype</b>	<code>public HashMap getItems()</code>
<b>Definition</b>	dbconfig.xml Vendor Items key, Value 가

- setItems

<b>Prototype</b>	<code>public void setItems(String _map)</code>
<b>Definition</b>	<pre> dbconfig.xml      Vendor      Items      key      value          )         HashMap infos = new HashMap();         infos.put("serverAddress", "127.0.0.1");         infos.put("portNo", "1433");         infos.put("dbName", "master");         infos.put("user", "sa");         infos.put("password", "1588");         poolInfo.setItems(infos);                     </pre>
<b>Argument</b>	<i>_map</i> key

- getDecodeCharSet

<b>Prototype</b>	<code>public void getDecodeCharSet()</code>
<b>Definition</b>	ConnectionPool                  가                  .

- setDecodeCharSet

<b>Prototype</b>	<code>public void setDecodeCharSet(String decode)</code>
<b>Definition</b>	ConnectionPool                  .
<b>Argument</b>	<i>decode</i> ConnectionPool

- getEncodeCharSet

<b>Prototype</b>	<code>public String getEncodeCharSet()</code>
<b>Definition</b>	ConnectionPool                  가                  .

- setEncodeCharSet

<b>Prototype</b>	<code>public void setEncodeCharSet(String encode)</code>
<b>Definition</b>	ConnectionPool                  .
<b>Argument</b>	<i>encode</i> ConnectionPool

- **ConnectionPoolStatus(oz.framework.db.ConnectionPoolStatus.class)**

ConnectionPool                  가                  .

- `public final static int OK = 1;`



- `getErrorMessage`

<b>Prototype</b>	<code>public String getErrorMessage()</code>
------------------	--

<b>Definition</b>	Pool Error Message 가 .
-------------------	------------------------

### Sample : ConnectionPoolSample.java

```

package sample;

import java.util.HashMap;
import oz.framework.api.ConnectionPool;
import oz.framework.db.ConnectionPoolInfo;
import oz.framework.db.ConnectionPoolStatus;
import org.apache.log4j.*;

public class ConnectionPoolSample {
    public static void main(String[] args) {
        // 가 가 .
        BasicConfigurator.configure();

        // OZServer Info.

        /**
         * // Daemon
         * String IP = "127.0.0.1"; // 가 IP
         * int PORT = 8003; // 가 TCP

         */
        // Servlet
        String URL = "http://www.oz.com/oz/server"; //Servlet 가 URL

        /**/
        // User Info.
        String ID = "admin"; //default
        String PWD = "admin"; //default
        ConnectionPool conPool = null;

        try {
            /**
             * // Daemon
             * conPool = new ConnectionPool(IP, PORT, ID, PWD, false, false);

             */
            // Servlet
            conPool = new ConnectionPool(URL, ID, PWD, false, false);
        }
    }
}

```

```
/**/  
//      가(addPool)  
Connecti onPool Info pool Info = new Connecti onPool Info();  
pool Info.setAl ias("forcs"); //  
pool Info.setVendor("mssql"); //      ( MSSQL)  
  
//  
//      map      dbconfi g.xml      db      jdbc url  
HashMap infos = new HashMap();  
infos.put("serverAddress", "127.0.0.1");  
infos.put("portNo", "1433");  
infos.put("dbName", "master");  
infos.put("user", "sa");  
infos.put("password", "1588");  
pool Info.setI tems(infos);  
  
pool Info.setMaxConns(20); //  
pool Info.setIni tConns(1); //  
//  
//pool Info.setEncodeCharSet("8859_1");  
//pool Info.setDecodeCharSet("KSC5601");  
  
conPool .addPool (pool Info);  
  
//      (removePool)  
String conPool Name = "forcs"; //  
conPool .removePool (conPool Name);  
// Connecti onPool Status      가      (getPool StatusLi st)  
Connecti onPool Status[]  
pool StatusLi st=conPool .getPool StatusLi st();  
for (int i = 0; i < pool StatusLi st.length; i++) {  
    Connecti onPool Status cps = pool StatusLi st[i];  
  
    //  
    System.out.pri ntl n(i);  
    System.out.pri ntl n("Statusstri ng=" + cps.getStatusStri ng  
());  
  
    //  
    System.out.pri ntl n("free=" + new  
Integer(cps.getFreeConnecti onCount()));  
  
    //  
    System.out.pri ntl n("checkedout=" + new  
Integer(cps.getCheckedOutConnecti onCount()));  
    System.out.pri ntl n();  
}
```

```
        // (save)
        conPool . save();
    }
    catch(Excepti on e) {
        e. pri ntStackTrace();
    }
}
}
```

## Class DataBind

### Constructor Summary

- DataBind(String ip, int port, String id, String pw, boolean bAutoLogin, boolean useUSL)
- DataBind(String url, String id, String pw, boolean bAutoLogin, boolean useUSL)

### Method Summary

- void setDataBindConfiguration(SortProperties config)
- SortProperties getDataBindConfiguration()

### Constructor Detail

	<i>//Daemon</i>	-	TCP Server
<b>Prototype</b>	public DataBind(String ip, int port, String id, String pw, boolean bAutoLogin, boolean useUSL)		
	<i>//Servlet</i>	-	HTTP Server
	public DataBind(String url, String id, String pw, boolean bAutoLogin, boolean useUSL)		
<b>Argument</b>	<i>url</i>	Servlet	URL ex) String url = "http://127.0.0.1/oz/server";
	<i>ip</i>	Daemon	IP ex) String ip = "127.0.0.1";
	<i>port</i>	Daemon	ex) int port = 8003;
	<i>id</i>		ex) String id = "admin";

<i>pw</i>	ex) String pw = "admin";
<i>bAutoLogin</i>	ex) boolean bAutoLogin = true;
<i>useUSL</i>	USL ex) boolean useUSL = false;

## Method Detail

### ■ setDataBindConfiguration

**Prototype** public void setDataBindConfiguration(SortProperties config) throws OZCPEException

**Definition** DataBind , "databind.properties"

**Argument** *config* DataBind

### ■ getDataBindConfiguration

**Prototype** public SortProperties getDataBindConfiguration() throws OZCPEException

**Definition** DataBind , "databind.properties" 가 .

- Key

setDataBindConfiguration() getDataBindConfiguration() key

Key	Value
ConcurrentFetch Size	FetchType "Concurrent" Stream byte, 4096, 256

<b>ConcurrentFirstRow</b>	FetchType "Concurrent"
---------------------------	------------------------

### Sample : DataBindSample.java

```

package sample;

import oz.framework.api.DataBind;
import org.apache.log4j.*;
import oz.util.SortProperties;

public class DataBindSample {
    public static void main(String[] args) {
        //          가          가
        BasicConfigurator.configure();

        // OZServer Info.
        String IP = "127.0.0.1"; //   가          IP
        int PORT = 8003; //   가          TCP
        // User Info.
        String ID = "admin"; //default
        String PWD = "admin"; //default

        DataBind dataBind = null;
        try {
            dataBind = new DataBind(IP, PORT, ID, PWD, false, false);
            SortProperties p = new SortProperties();

            //          (setDataBindConfigurati on)
            //Stream
            p.setProperty("ConcurrentFetchSize", "4096");

            //   row
            p.setProperty("ConcurrentFirstRow", "0");
            dataBind.setDataBindConfigurati on(p);

            //          가   (getDataBindConfigurati on)
            p = dataBind.getDataBindConfigurati on();
            java.util.Vector vec = p.propertyNames();
            for(int i=0; i<vec.size(); i++) {

```

```
        String name = (String)vec.elementAt(i);
        System.out.println(name + "=" + p.getProperty(name));
    }
}
catch (Exception e) {
    e.printStackTrace();
}
}
```

## Class Log

### Constructor Summary

- `Log(String ip, int port, String id, String pw, boolean bAutoLogin, boolean useUSL)`
- `Log(String url, String id, String pw, boolean bAutoLogin, boolean useUSL)`

### Method Summary

- `String getConfiguration()`
- `byte[] downloadLog()`
- `void downloadLog(String fileName)`
- `void setConfiguration(String logs)`
- `void setPriority(String p)`

### Constructor Detail

	<code>//Daemon</code>	-	<code>TCP Server</code>
<b>Prototype</b>	<code>public Log(String ip, int port, String id, String pw, boolean bAutoLogin, boolean useUSL)</code>		
	<code>//Servlet</code>	-	<code>HTTP Server</code>
	<code>public Log(String url, String id, String pw, boolean bAutoLogin, boolean useUSL)</code>		
<b>Argument</b>	<i>url</i>	Servlet	URL ex) <code>String url = "http://127.0.0.1/oz/server";</code>
	<i>ip</i>	Daemon	IP ex) <code>String ip = "127.0.0.1";</code>
	<i>port</i>	Daemon	IP ex) <code>int port = 8003;</code>

<i>id</i>	ex) String id = "admin";
<i>pw</i>	ex) String pw = "admin";
<i>bAutoLogin</i>	ex) boolean bAutoLogin = true;
<i>useUSL</i>	USL ex) boolean useUSL = false;

## Method Detail

### ■ getConfigure

**Prototype** public String getConfigure() throws OZCPEXception

**Definition** 가 .

### ■ downloadLog

**Prototype** public byte[] downloadLog() throws OZCPEXception

**Definition** .

### ■ downloadLog

**Prototype** public void downloadLog(String fileName) throws OZCPEXception, IOEXception

**Definition** .

**Argument** *fileName*

### ■ setConfigure

**Prototype** public void setConfigure(String logs) throws OZCPEXception

**Definition** .

**Argument** *logs* , "key=value"

ex) String logs="Priority=DEBUG"

ex) String logs="CONSOLE.Layout=%r[%t]%p%c{1}%X-%m%n"

- **setPriority**

<b>Prototype</b>	public void setPriority(String p) throws OZCPEException
<b>Definition</b>	.(INFO, DEBUG, ERROR)
<b>Argument</b>	p

### Sample : LogSample.java

```

package sample;

import oz.framework.api.Log;
import org.apache.log4j.*;

public class LogSample {
    public static void main(String[] args) {
        // 가 가
        BasicConfigurator.configure();

        // OZServer Info.
        // Daemon
        String IP = "127.0.0.1"; // 가 IP
        int PORT = 8003; // 가 TCP
        // Servlet
        String URL = "http://www.oz.com/oz/server"; //Servlet 가 URL
        // User Info.
        String ID = "admin"; //default
        String PWD = "admin"; //default
        Log log = null;
        try {
            // Daemon
            log = new Log(IP, PORT, ID, PWD, false, false);
            // Servlet
            //log = new Log(URL, ID, PWD, false, false); //comment
            // 가 (getConfigure)
            String conf = null;
            conf = log.getConfigure();
            System.out.println(conf);
            // (setConfigure, setPriority)
            String logs = "Priority=INFO";
            //log.setConfigure(logs);
            log.setPriority("DEBUG");
            // (downloadLog)
            byte b[] = log.downloadLog();
            String fileName = "server.log";
            //

```

```
        log.downloadLog(fileName);
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
}
}
```

## Class Mail

### Constructor Summary

- Mail(String ip, int port, String id, String pw, boolean bAutoLogin, boolean useUSL)
- Mail(String url, String id, String pw, boolean bAutoLogin, boolean useUSL)

### Method Summary

- void addAlias(String aliasName, SortProperties p) throws OZCPEException
- SortProperties getAliasConfig(String aliasName) throws OZCPEException
- String[] getMailAliasNames()
- void modifyAlias(String aliasName, String newAliasName, SortProperties p) throws OZCPEException
- void removeAlias(String aliasName) throws OZCPEException
- void send(String aliasName, String from, String fromUserName, String to, String cc, String bcc, String subject, String context, boolean isHTML, String localFileFullPath, String fileName) throws OZCPEException
- boolean sendSync(String aliasName, String from, String fromUserName, String to, String cc, String bcc, String subject, String context, boolean isHTML, String localFileFullPath, String fileName) throws OZCPEException

### Constructor Detail

---

Prototype	<pre>//Daemon - TCP Server public Mail (String ip, int port, String id, String pw, boolean bAutoLogin, boolean useUSL)</pre>
Prototype	<pre>//Servlet - HTTP Server public Mail (String url, String id, String pw, boolean bAutoLogin, boolean useUSL)</pre>

---

	<i>url</i>	Servlet ex) String url = "http://127.0.0.1/oz/server";	URL
	<i>ip</i>	Daemon ex) String ip = "127.0.0.1";	IP
	<i>port</i>	Daemon ex) int port = 8003;	
Argument	<i>id</i>	ex) String id = "admin";	
	<i>pw</i>	ex) String pw = "admin";	
	<i>bAutoLogin</i>	ex) boolean bAutoLogin = true;	
	<i>useUSL</i>	USL ex) boolean useUSL = false;	

## Method Detail

### ■ addAlias

Prototype	public void addAlias(String aliasName, SortProperties p) throws OZCPEException		
Definition	가 .		
Argument	<i>aliasName</i>		
	<i>p</i>	key	"Option"

### ■ getAliasConfig

Prototype	public SortProperties getAliasConfig(String aliasName) throws OZCPEException		
Definition	가 .		
Argument	<i>aliasName</i>		

### ■ getMailAliasNames

<b>Prototype</b>	public String[] getMailAliases()
<b>Definition</b>	mail.properties : active 가 가

■ **modifyAlias**

<b>Prototype</b>	public void modifyAlias(String aliasName, String newAliasName, SortProperties p) throws OZCPEException
<b>Definition</b>	<i>aliasName</i> <i>newAliasName</i>
<b>Argument</b>	<i>p</i> key "Option"

■ **removeAlias**

<b>Prototype</b>	public void removeAlias(String aliasName) throws OZCPEException
<b>Definition</b>	
<b>Argument</b>	<i>aliasName</i>

■ **send**

<b>Prototype</b>	public void send(String aliasName, String from, String fromUserName, String to, String cc, String bcc, String subject, String context, boolean isHTML, String localFileFullPath, String fileName) throws OZCPEException
<b>Definition</b>	
<b>Argument</b>	<i>aliasName</i> <i>from</i> <i>fromUserName</i> null "" <i>to</i> <i>cc</i> null "" " ," "

<i>bcc</i>	null	""	","	","
<i>subject</i>				
<i>context</i>				
<i>isHTML</i>		HTML		
<i>localFileFullPath</i>			" "	
<i>fileName</i>		:	" "	

■ **sendSync**

**Prototype** public boolean sendSync(String aliasName, String from, String fromUserName, String to, String cc, String bcc, String subject, String context, boolean isHTML, String localFileFullPath, String fileName) throws OZCPEException

**Definition** 가 .

**Argument** *aliasName*

*from*

*fromUserName* null ""

*to*

*cc* null ""  
","

*bcc* null ""  
","

*subject*

*context*

*isHTML* HTML

*localFileFullPath*

```

"|"
-----
fileName : "|"
    
```

## Option

■

Key	Value	
active	true/false	ex) setProperty("active", "true")
fromSend		ex) setProperty("fromSend", "mail@forcs.com")
toSend		ex) setProperty("toSend", "mail@forcs.com")
SMTPServer	SMTP URL	SMTP URL ex) setProperty("SMTPServer", "mail.forcs.com")
SMTPServerPort	SMTP	SMTP ex) setProperty("SMTPServerPort", "25")
SMTPUserID	SMTP ID	SMTP ID ex) setProperty("SMTPUserID", "UserID")
SMTPUserPassword	SMTP ID	SMTP ID ex) setProperty("SMTPUserPassword", "Password")
SMTPUserID_encrypted	SMTP ID	SMTP ID ex) setProperty("SMTPUserID_encrypted", "ScPdRcRgFgHdEbJaJbPcFc")
SMTPUserPassword_encrypted	SMTP ID	SMTP ID ex) setProperty("SMTPUserPassword_encrypted", "ScPdRcRgFgEbGaDbKbFbMaIbPa")

<b>EnableSSL</b>	true/false	SSL ex) setProperty("EnableSSL", "true")
<b>SendRetryCount</b>		ex) setProperty("SendRetryCount", "3")
<b>SendRetryPeriodTime</b>	( : )	ex) setProperty("SendRetryPeriodTime", "10")
<b>PrefixSubjectMessage</b>		ex) setProperty("PrefixSubjectMessage", "[Forcs]")

### Sample : MailSample.java

```

package sample;

import oz.framework.api.Mail;
import oz.util.SortProperties;

public class MailSample {

    public static void main(String[] args) throws Exception {

        Mail mail = new Mail("127.0.0.1", 8003, "admin", "admin", true, false);

        SortProperties props = new SortProperties();
        props.setProperty("active", "true");
        props.setProperty("fromSend", "forcs@forcs.com");
        props.setProperty("toSend", "oz@forcs.com");
        props.setProperty("SMTPServer", "smtp.gmail.com");
        props.setProperty("SMTPServerPort", "465");
        props.setProperty("SMTPUserID", "forcs@forcs.com");
        props.setProperty("SMTPUserPassword", "UserPassword");
        props.setProperty("EnableSSL", "true");

        mail.addAlias("forcs", props);

        mail.sendSync("forcs", "forcs@forcs.com", "User Name", "oz@gmail.com",
            null, null, "Mail Subject", "Mail Contextt", false, "d:\\parameter_test.ozd",
            "test.ozd");

        props = mail.getAliasConfig("forcs");
        mail.modifyAlias("forcs", "forcs2", props);
    }
}

```

```
mail.addAlias("forcs", props);
mail.addAlias("forcs4", props);
mail.addAlias("forcs3", props);

mail.removeAlias("forcs3");

String[] names = mail.getMailAliasesNames();
for(int i = 0; i < names.length; i++) {
    SortProperties aliasProps = mail.getAliasConfig(names[i]);
    java.util.Iterator propNameNames = aliasProps.propertyNames().iterator();
    while (propNameNames.hasNext()) {
        String propName = (String) propNameNames.next();
        System.out.println(propName + "=" + aliasProps.getProperty(propName));
    }
}
}
```

## Class Module

### Constructor Summary

- `Module(String ip, int port, String id, String pw, boolean bAutoLogin, boolean useUSL)`
- `Module(String url, String id, String pw, boolean bAutoLogin, boolean useUSL)`

### Method Summary

- `Inputstring getOZD(String item, String category, String[] urls)`
- `InputStream getOZU(String item, String category, String[] urls)`
- `void addODIParameter(String odiName, String key, String value)`
- `void addODIParameter(String odiName, String item, String category, Hashtable paramHash)`
- `void addParameter(String key, String value)`
- `void addApplicationParameter(String key, String value)`
- `void registODIPath (String odiName, String path)`
- `void saveOZD(String fileName, String item, String category, String[] urls)`
- `void saveOZU(String fileName, String item, String category, String[] urls)`
- `public IReportInfo addReport(String itemName, String categoryName) throws OZCPEException`
- `public IReportInfo addReport(String itemName, String categoryName, String displayName) throws OZCPEException`
- `public InputStream getOZD() throws OZCPEException`
- `public void saveOZD(String fileName) throws OZCPEException`
- `public void setMemoAllowed(boolean allowed)`
- `public void setPassword(String password)`

## Constructor Detail

<b>Prototype</b>	<i>//Daemon</i> - TCP Server
	<code>public Module(String ip, int port, String id, String pw, boolean bAutoLogin, boolean useUSL)</code>
	<i>//Servlet</i> - HTTP Server
	<code>public Module(String url, String id, String pw, boolean bAutoLogin, boolean useUSL)</code>
<b>Argument</b>	<i>url</i> Servlet URL ex) String url = "http://127.0.0.1/oz/server";
	<i>ip</i> Daemon IP ex) String ip = "127.0.0.1";
	<i>port</i> Daemon ex) int port = 8003;
	<i>id</i> ex) String id = "admin";
	<i>pw</i> ex) String pw = "admin";
	<i>bAutoLogin</i> ex) boolean bAutoLogin = true;
	<i>useUSL</i> USL ex) boolean useUSL = false;

## Method Detail

### ■ getOZD

**Prototype** `public InputStream getOZD(String item, String category, String[] urls) throws OZCPEXception`

<b>Definition</b>	가 , OZD urls 가 OZD serverDMType serverDMType DM_TYPE="Memory", FetchType="Batch"												
<b>Argument</b>	<table border="1"> <tr> <td><i>item</i></td> <td>(</td> <td>OZR</td> <td>)</td> </tr> <tr> <td><i>category</i></td> <td></td> <td></td> <td></td> </tr> <tr> <td><i>urls</i></td> <td>OZD</td> <td></td> <td>URL</td> </tr> </table>	<i>item</i>	(	OZR	)	<i>category</i>				<i>urls</i>	OZD		URL
<i>item</i>	(	OZR	)										
<i>category</i>													
<i>urls</i>	OZD		URL										

■ **getOZU**

<b>Prototype</b>	public InputStream getOZU(String item, String category, String[] urls) throws OZCPEXCEPTION												
<b>Definition</b>	SDM 가 OZU 가 : API DM_TYPE="Memory", FetchType="Batch" : "FetchUnit" "DM_PER_DATAMODULE"												
<b>Argument</b>	<table border="1"> <tr> <td><i>item</i></td> <td>(</td> <td>OZA</td> <td>)</td> </tr> <tr> <td><i>category</i></td> <td></td> <td></td> <td></td> </tr> <tr> <td><i>urls</i></td> <td>OZU</td> <td></td> <td>URL</td> </tr> </table>	<i>item</i>	(	OZA	)	<i>category</i>				<i>urls</i>	OZU		URL
<i>item</i>	(	OZA	)										
<i>category</i>													
<i>urls</i>	OZU		URL										

■ **addODIParameter**

<b>Prototype</b>	public void addODIParameter(String odiName, String key, String value)						
<b>Definition</b>	SDM ODI ODI . ODI ODI						
<b>Argument</b>	<table border="1"> <tr> <td><i>odiName</i></td> <td>ODI</td> </tr> <tr> <td><i>key</i></td> <td>ODI</td> </tr> <tr> <td><i>value</i></td> <td>ODI</td> </tr> </table>	<i>odiName</i>	ODI	<i>key</i>	ODI	<i>value</i>	ODI
<i>odiName</i>	ODI						
<i>key</i>	ODI						
<i>value</i>	ODI						

■ **addODIParameter**

**Prototype**     public void addODIParameter(String odiName, String item, String category, Hashtable paramHash) throws IllegalArgumentException

**Definition**

SDM	ODI	ODI
. ODI		ODI
ODI	SDM	
SDM		

**Argument**

<i>odiName</i>	ODI
<i>item</i>	ODI
<i>category</i>	ODI
<i>paramHash</i>	Key, Value 가 Hashtable

```

: OZU     paramHash
OZU     addODIParameter()     paramHash     null
         addApplicationParameter(Key,Value)     ODI
    
```

ex) addApplicationParameter

```

module.addApplicationParameter("odi.odinames", "sample");
module.addApplicationParameter("odi.sample.pcount", "1");
module.addApplicationParameter("odi.sample.args1", "deptid=501");
    
```

■ **addParameter**

**Prototype**     public void addParameter(String key, String value)

**Definition**     SDM

**Argument**

<i>key</i>
<i>value</i>

■ **setMemoAllowed**

**Prototype**     public void setMemoAllowed(boolean allowed)

**Definition**

**Argument**     *allowed*

■ **setPassword**

<b>Prototype</b>	<code>public void setPassword(String password)</code>
<b>Definition</b>	
<b>Argument</b>	<i>password</i>

■ **addApplicationParameter**

<b>Prototype</b>	<code>public void addApplicationParameter(String key, String value)</code>
<b>Definition</b>	SDM ODI ODI
<b>Argument</b>	<i>key</i> ODI <i>value</i> ODI

■ **registODIPath**

<b>Prototype</b>	<code>public void registODIPath (String odiName, String path) throws IllegalArgumentException</code>
<b>Definition</b>	OZD ODI
<b>Argument</b>	<i>odiName</i> ODI <i>path</i> ODI

■ **saveOZD**

<b>Prototype</b>	<code>public void saveOZD(String fileName, String item, String category, String[] urls) throws OZCPEException</code>
<b>Definition</b>	OZD : API DM_TYPE="Momory", FetchType="Batch"
<b>Argument</b>	<i>fileName</i> OZD 가 <i>item</i> (.ozr) <i>category</i> (.ozr) <i>urls</i> OZD URL

■ **saveOZU**

<b>Prototype</b>	public void saveOZU(String filename, String item, String category, String[] urls) throws OZCPEXception		
<b>Definition</b>	OZU	:	API
			DM_TYPE="Momory", FetchType="Batch"
		:	"FetchUnit" "DM_PER_DATAMODULE"
<b>Argument</b>	<i>filename</i>	OZU	가
	<i>item</i>	(.oza)	
	<i>category</i>	(.oza)	
	<i>urls</i>	OZU	URL

■ **addReport**

<b>Prototype</b>	public IReportInfo addReport(String itemName, String categoryName) throws OZCPEXception		
<b>Definition</b>	OZD	:	가
<b>Argument</b>	<i>itemName</i>	가	
	<i>categoryName</i>	가	

<b>Prototype</b>	public IReportInfo addReport(String itemName, String categoryName, String displayName) throws OZCPEXception		
<b>Definition</b>	OZD	:	가
<b>Argument</b>	<i>itemName</i>	가	
	<i>categoryName</i>	가	
	<i>displayname</i>		

■ **getOZD**

<b>Prototype</b>	public InputStream getOZD() throws OZCPEXception		
<b>Definition</b>	OZD	:	가

■ **saveOZD**

<b>Prototype</b>	public void saveOZD(String fileName) throws OZCPEXception		
------------------	---	--	--

Definition	OZD	OZD
Argument	<i>filename</i>	

## Interface

### ■ IReportInfo(oz.framework.api.IReportInfo)

OZD 가

#### - getItemName

**Prototype** public String getItemName()

**Definition** 가

#### - getCategoryName

**Prototype** public String getCategoryName()

**Definition** 가

#### - addURL

**Prototype** public void addURL(String[] urls)

**Definition** URL

**Argument** *urls* URL

**Prototype** public void addURL(String url)

**Definition** URL

**Argument** *url* URL

#### - addParameter

**Prototype** public void addParameter(String key, String value)

**Definition**

**Argument** *key*

*value*

#### - addODIParameter

**Prototype** public void addODIParameter(String odiName, String key, String value)

<b>Definition</b>	ODI	
	<i>odi Name</i>	ODI
<b>Argument</b>	<i>key</i>	ODI
	<i>value</i>	ODI

- **addProperties**

<b>Prototype</b>	public void addProperties(String key, String value)		
<b>Definition</b>	<i>key</i>		
<b>Argument</b>	<i>value</i>	key	"Option"

- **addFrameworkURL**

<b>Prototype</b>	public void addFrameworkURL(String fxName, String url, String charset)		
<b>Definition</b>	FX	URL	
	<i>fxName</i>	FX	
<b>Argument</b>	<i>url</i>	URL	
	<i>charset</i>	<ul style="list-style-type: none"> <li>• null</li> <li>• base64( <ul style="list-style-type: none"> <li>base64/ksc5601</li> <li>base64/utf-8</li> <li>ksc5601</li> <li>utf-8</li> </ul> </li> </ul>	)
		base64	

**Sample : ModuleSample.java**

```
package sample;

import java.io.*;
import oz.framework.api.Module;
import org.apache.log4j.*;

public class ModuleSample {
```

```

public static void main(String[] args) {
//      가      가      .
    BasicConfigurator.configure();

    // OZServer Info.
    /**
    // Daemon
    String IP = "127.0.0.1"; // 가      IP
    int PORT = 8003; // 가      TCP
    /**/
    // User Info.
    String ID = "admin"; //default
    String PWD = "admin"; //default
    Module module = null;
    InputStream stream = null;
    try {
        /**
        // Daemon
        module = new Module(IP, PORT, ID, PWD, false, false);
        // -----
        // OZD
        // set form parameters
        module.addParameter("formparam1", "ABCD1");
        module.addParameter("formparam2", "EFGH2");
        // set odi parameters
        module.addODIParameter("parameter_test", "odi param1", "IJKL1");
        module.addODIParameter("parameter_test", "odi param2", "LMN02");
        module.setPassword("1234");
        module.setMemoAllowed(true);
        String[] url = new String[0];
        module.saveOZD("C:/parameter_test.ozd", "parameter_test.ozr", "/",
url);
    }
    catch(Exception e) {
        e.printStackTrace();
    }
}
}

```

### Sample : ModuleOZUSample.java

```

package sample;

import oz.framework.api.Module;
import org.apache.log4j.*;

```

```
public class ModuleOZUSample{
    public static void main(String[] args){
        // 가 가 .
        BasicConfigurator.configure();

        // OZServer Info.
        String IP = "127.0.0.1"; // 가 IP
        int PORT = 8003; // 가 TCP
        // User Info.
        String ID = "admin"; //default
        String PWD = "admin"; //default
        Module module = null;
        try{
            module = new Module(IP, PORT, ID, PWD, false, /*usl*/ false);
            //ozu
            module.addApplicationParameter("odi.fetchunit", "DM_PER_DATAMODULE");
            module.addApplicationParameter("odi.odi names", "parameter_test");
            module.addApplicationParameter("odi.parameter_test.pcount", "2");
            module.addApplicationParameter("odi.parameter_test.args1",
            "odi param1=ABCD");
            module.addApplicationParameter("odi.parameter_test.args2",
            "odi param2=1234");

            String[] urls = new String[0];

            module.addODIParameter("parameter_test", "parameter_test.odi", "/",
            null);

            //ozu , oza , URL
            module.saveOZU("C:/parameter_test.ozu", "parameter_test.oza", "/",
            urls);
        }
        catch(Exception e){
            e.printStackTrace();
        }
    }
}
```

### Sample : MultiReportSample.java

```
package sample;

import org.apache.log4j.*;
```

```
import oz.framework.api.*;

public class MultiReportSample{

    public static void main(String[] args) throws Exception{

        BasicConfigurator.configure();

        Module maker = new Module("127.0.0.1", 8003, "admin", "admin", true,
false);

        String[] urls = new String[0];

        oz.framework.api.IReportInfo reportInfo =
maker.addReport("parameter_test.ozr", "/", "Report1");
        reportInfo.addURL(urls);
        reportInfo.addODIParameter("parameter_test.odi", "odi param1", "form
number 11");
        reportInfo.addODIParameter("parameter_test.odi", "odi param2", "form
number 12");
        reportInfo.addParameter("formparam1", "form parameters 11");
        reportInfo.addParameter("formparam2", "form parameters 12");

        reportInfo = maker.addReport("parameter_test.ozr", "/", "Report2");
        reportInfo.addURL(urls);
        reportInfo.addODIParameter("parameter_test.odi", "odi param1", "form
number 21");
        reportInfo.addODIParameter("parameter_test.odi", "odi param2", "form
number 22");
        reportInfo.addParameter("formparam1", "form parameters 21");
        reportInfo.addParameter("formparam2", "form parameters 22");

        reportInfo = maker.addReport("parameter_test.ozr", "/", "Report3");
        reportInfo.addURL(urls);
        reportInfo.addODIParameter("parameter_test.odi", "odi param1", "form
number 31");
        reportInfo.addODIParameter("parameter_test.odi", "odi param2", "form
number 32");
        reportInfo.addParameter("formparam1", "form parameters 31");
        reportInfo.addParameter("formparam2", "form parameters 32");

        maker.saveOZD("c:\\MultiReport.ozd");
    }
}
```

## UseOzdParameterSample.java

```
package sample;

import oz.framework.api.*;

public class UseOzdParameterSample {

    public static void main(String[] args) throws Exception{

        oz.framework.api.Module module = new
        oz.framework.api.Module("127.0.0.1", 8003, "admin", "admin");

        oz.framework.api.IReportInfo reportInfo =
        module.addReport("parameter_test.ozr", "/", "1page");
        reportInfo.addODIParameter("parameter_test", "odi param1", "A1");
        reportInfo.addODIParameter("parameter_test", "odi param2", "B2");

        reportInfo.addParameter("formparam1", "C3");
        reportInfo.addParameter("formparam2", "D4");

        reportInfo.addProperties("use_ozd_parameter", "false");

        reportInfo = module.addReport("parameter_test.ozr", "/",
        "2page");
        reportInfo.addODIParameter("parameter_test", "odi param1", "E5");
        reportInfo.addODIParameter("parameter_test", "odi param2", "F6");

        reportInfo.addParameter("formparam1", "G7");
        reportInfo.addParameter("formparam2", "H8");

        reportInfo.addProperties("use_ozd_parameter", "false");

        module.saveOZD("c:/daemon_api_OZD.ozd");
        System.out.println("ok");
    }
}
```

## Class Monitor

### Constructor Summary

- `Monitor(String ip, int port, String id, String pw, boolean bAutoLogin, boolean useUSL)`
- `Monitor (String url, String id, String pw, boolean bAutoLogin, boolean useUSL)`

### Method Summary

- `Versions getVersions()`
- `MemoryStatus getMemoryInfo()`
- `byte[] downloadMonitorLog()`
- `void downloadMonitorLog(String fileName)`

### Constructor Detail

	<code>//Daemon</code>	-	<code>TCP Server</code>
<b>Prototype</b>	<code>public Monitor(String ip, int port, String id, String pw, boolean bAutoLogin, boolean useUSL)</code>		
	<code>//Servlet</code>	-	<code>HTTP Server</code>
	<code>public Monitor(String url, String id, String pw, boolean bAutoLogin, boolean useUSL)</code>		
<b>Argument</b>	<i>url</i>	Servlet	URL ex) <code>String url = "http://127.0.0.1/oz/server";</code>
	<i>ip</i>	Daemon	IP ex) <code>String ip = "127.0.0.1";</code>
	<i>port</i>	Daemon	IP ex) <code>int port = 8003;</code>

<i>id</i>	ex) String id = "admin";
<i>pw</i>	ex) String pw = "admin";
<i>bAutoLogin</i>	ex) boolean bAutoLogin = true;
<i>useUSL</i>	USL ex) boolean useUSL = false;

## Method Detail

### ■ **getVersions**

**Prototype** public Versions getVersions() throws OZCPEException

**Definition** 가 .

### ■ **getMemoryInfo**

**Prototype** public MemoryStatus getMemoryInfo() throws OZCPEException

**Definition** ( , , ) 가 .

### ■ **downloadMonitorLog**

**Prototype** public byte[] downloadMonitorLog() throws OZCPEException

**Definition** 가 .

### ■ **downloadMonitorLog(String fileName)**

**Prototype** public void downloadMonitorLog(String fileName) throws OZCPEException, IOException

**Definition** 가 fileName

## Class

### ■ MemoryStatus(oz.server.monitor.MemoryStatus)

Server가 System

-

- public static void garbageCollect() : garbage collection
- public long getTotalMemory() : VM
- public long getFreememory() : VM Free

### ■ Versions(oz.server.monitor.Versions)

Server Server가 System

-

- public String osName : Server가 OS
- public String osVersion : Server가 OS
- public String javaVendor : Server가 JVM
- public String javaVersion : Server가 JVM Version
- public String OZServerVersion :
- public String CPRelease : OZ Common Protocol
- public int CPProtocol : OZ Common Protocol
- public String DMRelease : OZ Data Module
- public int DMStreaming : OZ Data Module Streaming

## Sample : MonitorSample.java

```
package sample;

import oz.framework.api.Monitor;
import oz.server.monitor.Versions;
import oz.server.monitor.MemoryStatus;
import org.apache.log4j.*;

public class MonitorSample {
    public static void main(String[] args) {
        // Server가 System
        BasicConfigurator.configure();

        // OZServer Info.
```

```
/**
// Daemon
String IP = "127.0.0.1"; // 가 IP
int PORT = 8003; // 가 TCP
/**
// Servlet
String URL = "http://www.oz.com/oz/server"; //Servlet 가 URL
/**/
// User Info.
String ID = "admin"; //default
String PWD = "admin"; //default

Monitor monitor = null;
try {
/**
// Daemon
monitor = new Monitor(IP, PORT, ID, PWD, false, false);
/**/
// Servlet
monitor = new Monitor(URL, ID, PWD, false, false);
/**/

// (getServerInformation)
Versions v = monitor.getVersions();
v._printOut();

// (getServerStatus)
MemoryStatus ms = monitor.getMemoryInfo();

// = -
long usedMemory = ms.getTotalMemory() - ms.getFreeMemory();

// 가
System.out.println("Total Memory="+ms.getTotalMemory());
System.out.println("Used Memory="+usedMemory);
System.out.println("Free Memory="+ms.getFreeMemory());
System.out.println("");

//
byte [] logBytes = monitor.downloadMonitorLog();

//
String logFileName = "monitor.log";
monitor.downloadMonitorLog(logFileName);

}
catch(Exception e)
{
```

```
        e.printStackTrace();
    }
}
```

## Class Service

### Constructor Summary

- Service(String ip, int port, String id, String pw, boolean bAutoLogin, boolean useUSL)
- Service(String url, String id, String pw, boolean bAutoLogin, boolean useUSL)

### Method Summary

- void garbageCollect()
- void stop(boolean check)
- void restart()
- boolean ping()
- int getHandlerCount()

### Constructor Detail

<b>Prototype</b>	<code>//Daemon</code>	-	<b>TCP Server</b>
	<code>public Service(String ip, int port, String id, String pw, boolean bAutoLogin, boolean useUSL)</code>		
<b>Argument</b>	<code>//Servlet</code>	-	<b>HTTP Server</b>
	<code>public Service(String url, String id, String pw, boolean bAutoLogin, boolean useUSL)</code>		
	<i>url</i>	Servlet	URL ex) String url = "http://127.0.0.1/oz/server";
	<i>ip</i>	Daemon	IP ex) String ip = "127.0.0.1";

<i>port</i>	Daemon ex) int port = 8003;
<i>id</i>	ex) String id = "admin";
<i>pw</i>	ex) String pw = "admin";
<i>bAutoLogin</i>	ex) boolean bAutoLogin = true;
<i>useUSL</i>	USL ex) boolean useUSL = false;

## Method Detail

### ■ garbageCollect

**Prototype** `public void garbageCollect() throws OZCPEException`

**Definition** JVM  
가 garbage collection

### ■ stop

**Prototype** `public void stop(boolean check) throws OZCPEException`

**Definition**

**Argument** *check*

- true :
- false :

### ■ restart

**Prototype** `public void restart() throws OZCPEException`

**Definition**

### ■ ping

**Prototype** `public boolean ping()`

**Definition**

■ **getHandlerCount**

**Prototype**    public int getHandlerCount()

**Definition**    .

**Sample : ServiceSample.java**

```

package sample;

import oz.framework.api.Service;
import org.apache.log4j.*;

public class ServiceSample {
    public static void main(String[] args) {
        //          가          가

        BasicConfigurator.configure();
        // OZServer Info.
        // *
        // Daemon
        String IP = "127.0.0.1"; //          가          IP
        int PORT = 8003; //          가          TCP
        String URL = "http://localhost:8080/OZServlet/server";

        /**
        * // Servlet String URL = "http://www.oz.com/oz/server";
        //Servlet
        *          가          URL /

        *****/

        // User Info.
        String ID = "admin"; // default
        String PWD = "admin"; // default
        Service service = null;
        try {
            // *
            // Daemon
            service = new Service(IP, PORT, ID, PWD, false,
false);

            //service = new Service(URL, ID, PWD, false, false);

```

```

/*****
* // Servlet service = new Service(URL, ID, PWD,
false, false); /
*****/

// ping
boolean isAlive = service.ping();

//
if(isAlive == true) {
    System.out.println("Server is Running");

    //

System.out.println(service.getHandlerCount());

// garbageCollection (garbageCollects)
service.garbageCollect();

// (restart)
// service.restart();

// (serverStop)
boolean check = true; //
service.stop(check);

// 가
isAlive = true;
while (isAlive) {
    isAlive = service.ping();
}
if(isAlive == false) {
    System.out.println("Server is
stopped");
}
//
else {
    System.out.println("Server is dead");
}

} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

## Class Viewer

### Constructor Summary

- Viewer(String ip, int port, String id, String pw, boolean bAutoLogin, boolean useUSL)
- Viewer(String url, String id, String pw, boolean bAutoLogin, boolean useUSL)

### Method Summary

- byte[] getByteArrayForm(String reportName, String categoryName)
- InputStram getForm(String reportName, String categoryName, boolean isCompress)
- Parameter[] getUserParametersWithDefaultValue(String itemName, int itemType, String categoryName)
- HCDataModule getHCUSDM(String fileName, String categoryName)
- HCDataModule getDataModule(InputStream sdmlInput)
- HCDataModule getDataModule(String odiname, String categoryName, Parameter[] parameters, boolean doCompress, boolean forceRefresh)
- HCDataModule getDataModule(String odiName, String categoryName, Parameter[] parameters, boolean doCompress, boolean forceRefresh, String[] invalidDataset, MaxRowsOfSet[] maxRows)
- HCDataModule getDataModule(String odiName, String categoryName, Parameter[] parameters, boolean doCompress, boolean forceRefresh, boolean isPushServer)

### Constructor Detail

<b>Prototype</b>	<i>//Daemon</i> - TCP Server						
	<code>public Viewer(String ip, int port, String id, String pw, boolean bAutoLogin, boolean useUSL)</code>						
<b>Argument</b>	<i>//Servlet</i> - HTTP Server						
	<code>public Viewer(String url, String id, String pw, boolean bAutoLogin, boolean useUSL)</code>						
	<table border="1"> <tr> <td><i>url</i></td> <td>Servlet</td> <td>URL</td> </tr> <tr> <td></td> <td colspan="2">ex) String url = "http://127.0.0.1/oz/server";</td> </tr> </table>	<i>url</i>	Servlet	URL		ex) String url = "http://127.0.0.1/oz/server";	
<i>url</i>	Servlet	URL					
	ex) String url = "http://127.0.0.1/oz/server";						
	<table border="1"> <tr> <td><i>ip</i></td> <td>Daemon</td> <td>IP</td> </tr> <tr> <td></td> <td colspan="2">ex) String ip = "127.0.0.1";</td> </tr> </table>	<i>ip</i>	Daemon	IP		ex) String ip = "127.0.0.1";	
<i>ip</i>	Daemon	IP					
	ex) String ip = "127.0.0.1";						
	<table border="1"> <tr> <td><i>port</i></td> <td>Daemon</td> <td></td> </tr> <tr> <td></td> <td colspan="2">ex) int port = 8003;</td> </tr> </table>	<i>port</i>	Daemon			ex) int port = 8003;	
<i>port</i>	Daemon						
	ex) int port = 8003;						
	<table border="1"> <tr> <td><i>id</i></td> <td></td> <td></td> </tr> <tr> <td></td> <td colspan="2">ex) String id = "admin";</td> </tr> </table>	<i>id</i>				ex) String id = "admin";	
<i>id</i>							
	ex) String id = "admin";						
	<table border="1"> <tr> <td><i>pw</i></td> <td></td> <td></td> </tr> <tr> <td></td> <td colspan="2">ex) String pw = "admin";</td> </tr> </table>	<i>pw</i>				ex) String pw = "admin";	
<i>pw</i>							
	ex) String pw = "admin";						
	<table border="1"> <tr> <td><i>bAutoLogin</i></td> <td></td> <td></td> </tr> <tr> <td></td> <td colspan="2">ex) boolean bAutoLogin = true;</td> </tr> </table>	<i>bAutoLogin</i>				ex) boolean bAutoLogin = true;	
<i>bAutoLogin</i>							
	ex) boolean bAutoLogin = true;						
	<table border="1"> <tr> <td><i>useUSL</i></td> <td>USL</td> <td></td> </tr> <tr> <td></td> <td colspan="2">ex) boolean useUSL = false;</td> </tr> </table>	<i>useUSL</i>	USL			ex) boolean useUSL = false;	
<i>useUSL</i>	USL						
	ex) boolean useUSL = false;						

## Method Detail

### ■ `getByteArrayForm`

<b>Prototype</b>	<code>public byte[] getByteArrayForm(String reportName, String categoryName) throws OZCPEException</code>		
<b>Definition</b>	가 .		
<b>Argument</b>	<table border="1"> <tr> <td><i>reportName</i></td> </tr> <tr> <td><i>categoryName</i></td> </tr> </table>	<i>reportName</i>	<i>categoryName</i>
<i>reportName</i>			
<i>categoryName</i>			

### ■ `getForm`

<b>Prototype</b>	<code>public InputStream getForm(String reportName, String categoryName, boolean isCompress) throws OZCPEException</code>
------------------	---

<b>Definition</b>	가 .
	<i>reportName</i>
<b>Argument</b>	<i>categoryName</i>
	<i>isCompress</i>

■ **getUserParametersWithDefaultValue**

<b>Prototype</b>	public Parameter[] getUserParametersWithDefaultValue(String itemName, int itemType, String categoryName) throws OZCPEXception
<b>Definition</b>	.
	<i>itemName</i>
<b>Argument</b>	<i>itemName</i> (OZR : 20001 / ODI : 10000)
	<i>categoryName</i>
	<i>isCompress</i>

■ **getHCUSDM**

<b>Prototype</b>	public HCDataModule getHCUSDM(String fileName, String categoryName) throws OZCPEXception
<b>Definition</b>	SDM 가 .
	<i>fileName</i>
<b>Argument</b>	<i>categoryName</i>

■ **getDataModule**

<b>Prototype</b>	public HCDataModule getDataModule(InputStream sdmInput) throws OZCPEXception
<b>Definition</b>	SDM 가 . FETCH (batch) , DataModule 2.5 Normal .
<b>Argument</b>	<i>sdmInput</i> SDM

■ **getDataModule**

	<pre>public HCDataModule getDataModule(String odiName, String categoryName, Parameter[] parameters, boolean doCompress, boolean forceRefresh) throws OZCPEXception</pre>																
<b>Prototype</b>	<pre>public HCDataModule getDataModule(String odiName, String categoryName, Parameter[] parameters, boolean doCompress, boolean forceRefresh, String[] invalidDataset, MaxRowsOfSet[] maxRows) throws OZCPEXception</pre>																
<b>Definition</b>	<pre>public HCDataModule getDataModule(String odiName, String categoryName, Parameter[] parameters, boolean doCompress, boolean forceRefresh, boolean isPushServer) throws OZCPEXception</pre>																
<b>Argument</b>	<table border="1"> <tr> <td><i>odiName</i></td> <td>.</td> </tr> <tr> <td><i>categoryName</i></td> <td>.</td> </tr> <tr> <td><i>parameters</i></td> <td>가</td> </tr> <tr> <td><i>doCompress</i></td> <td>.</td> </tr> <tr> <td><i>forceRefresh</i></td> <td>.</td> </tr> <tr> <td><i>isPushServer</i></td> <td>.</td> </tr> <tr> <td><i>invalidDataset</i></td> <td>.</td> </tr> <tr> <td><i>maxRows</i></td> <td>.</td> </tr> </table>	<i>odiName</i>	.	<i>categoryName</i>	.	<i>parameters</i>	가	<i>doCompress</i>	.	<i>forceRefresh</i>	.	<i>isPushServer</i>	.	<i>invalidDataset</i>	.	<i>maxRows</i>	.
<i>odiName</i>	.																
<i>categoryName</i>	.																
<i>parameters</i>	가																
<i>doCompress</i>	.																
<i>forceRefresh</i>	.																
<i>isPushServer</i>	.																
<i>invalidDataset</i>	.																
<i>maxRows</i>	.																

## Class

### ■ Parameter(oz.dm.Parameter)

가

- public String name :
- public String value :

### ■ MaxRowsOfSet (oz.dm. MaxRowsOfSet)

가

- SetSetName

**Prototype** `public void SetSetName(String v)`

**Definition**

▪ GetSetName

**Prototype** public String GetSetName()

**Definition** 가 가

▪ SetMaxRow

**Prototype** public void SetMaxRow(int v)

**Definition**

▪ GetMaxRow

**Prototype** public int GetMaxRow()

**Definition** 가

■ **HCDDataModule(oz.dm.hc.HCDDataModule)**

HCDDataSet HCDDataSet BCDDataSet 가 OZ  
DataModule Streaming

▪ public final static String PARAMETERSET\_NAME = "OZParam"; :

▪ getDataSetNames

**Prototype** public String[] getDataSetNames()

**Definition**

▪ getDataSets

**Prototype** public HCDDataSet[] getDataSets(String name)

가

**Definition**

가

(length)가 0

**Argument** name



```
/**
 * Daemon
 * String IP = "127.0.0.1"; // 가 IP
 * int PORT = 8003; // 가 TCP
 */
// Servlet
String URL = "http://www.oz.com/oz/server"; //Servlet
가 URL
/**/
// User Info.
String ID = "admin"; //default
String PWD = "admin"; //default

Viewer viewer = null;
try {
    /**
     * Daemon
     * viewer = new Viewer(IP, PORT, ID, PWD, false, false);
     */
    // Servlet
    viewer = new Viewer(URL, ID, PWD, false, false);
    /**/

    int ODI_FILE_TYPE = 10000; // ODI
    int OZR_FILE_TYPE = 20001; // OZR
    String ozrName = "parameter_test.ozr";
    String odiName = "parameter_test.odi"; //ODI
    String sdmName = "parameter_test.sdm";
    String categoryName = "/"; //
    Parameter[] param = new Parameter[0]; //
    boolean force = false; //
    boolean compress = false; // DM
    String[] invalidSet = new String[0]; //

    MaxRowsOfSet[] mx = new MaxRowsOfSet[0]; //

    // 가 SDM 가 (getHCUSDM)
    HCDataModule usdm = viewer.getHCUSDM(sdmName,
categoryName);

    // 가 . (getByteArrayForm)
    byte[] formBytes = viewer.getByteArrayForm(ozrName,
categoryName);

    // 가 (getForm)
    InputStream in = viewer.getForm(ozrName, categoryName,
compress);
```

```
//
//(getUserParametersWithDefaultVal ue)
param = viewer.getUserParametersWithDefaultVal ue(odi Name,
    ODI_FILE_TYPE, categoryName);

// Input Stream      SDM
(getDataModule)
HCDataModule sdm = viewer.getDataModule(
    new FileInputStream(sdmName));

//                                (getDataModules)
HCDataModule dModule = viewer.getDataModule(
    odi Name, categoryName, param, compress, force);

//                                (getDataModules)
HCDataModule dModule2 = viewer.getDataModule(
    odi Name, categoryName, param, compress, force, invalidSet,
mx);
}
catch (Exception e) {
    e.printStackTrace();
}
}
```

## Class Servlet

### Method Summary

- `InputStream getOZD(String item, String category, Hashtable formparam, Hashtable odiparam, boolean memoallowed, String password, String id, String pw, HttpServletRequest request)`
- `InputStream getOZD(String item, String category, String serverDMType, Hashtable formparam, Hashtable odiparam, Hashtable odipath, boolean memoallowed, String password, String id, String pwd, HttpServletRequest request)`
- `InputStream getOZD(String[] item, String[] category, Hashtable[] formparam, Hashtable[] odiparam, String id, String pwd, HttpServletRequest request, String[] displayName, String[][] urls, Hashtable[] options) throws Exception`
- `InputStream getOZD(String[] item, String[] category, Hashtable[] formparam, Hashtable[] odiparam, boolean memoallowed, String password, String id, String pwd, HttpServletRequest request, String[] displayName, String[][] urls, Hashtable[] options) throws Exception`
- `InputStream getOZD(String[] item, String[] category, String serverDMType, Hashtable[] formparam, Hashtable[] odiparam, boolean memoallowed, String password, String id, String pwd, HttpServletRequest request, String[] displayName, String[][] urls, Hashtable[] options) throws Exception`
- `InputStream getOZD(String[] item, String[] category, String serverDMType, Hashtable[] formparam, Hashtable[] odiparam, Hashtable[] odiPaths, boolean memoallowed, String password, String id, String pwd, HttpServletRequest request, String[] displayName, String[][] urls, Hashtable[] options, Hashtable[] fxURLs) throws Exception`
- `InputStream getOZDWithSDM(String item, String category, Hashtable sdm, Hashtable formParam, boolean memoallowed, String password, String id, String pwd, HttpServletRequest request)`
- `getOZDWithSDM(String item, String category, Hashtable sdm, Hashtable`

formParam, boolean memoallowed, String password, String id, String pwd, HttpServletRequest request)

## Method Detail

### ■ getOZD

---

```
public final InputStream getOZD(String item, String category, Hashtable formparam, Hashtable odi param, boolean memoallowed, String password, String id, String pwd, HttpServletRequest request) throws Exception
```

```
public final InputStream getOZD(String item, String category, String serverDMType, Hashtable formparam, Hashtable odi param, Hashtable odi path, boolean memoallowed, String password, String id, String pwd, HttpServletRequest request)
```

```
public final InputStream getOZD(String[] item, String[] category, Hashtable[] formparam, Hashtable[] odi param, String id, String pwd, HttpServletRequest request, String[] displayName, String[][] urls, Hashtable[] options) throws Exception
```

**Prototype**

```
public final InputStream getOZD(String[] item, String[] category, Hashtable[] formparam, Hashtable[] odi param, boolean memoallowed, String password, String id, String pwd, HttpServletRequest request, String[] displayName, String[][] urls, Hashtable[] options) throws Exception
```

```
public final InputStream getOZD(String[] item, String[] category, String serverDMType, Hashtable[] formparam, Hashtable[] odi param, boolean memoallowed, String password, String id, String pwd, HttpServletRequest request, String[] displayName, String[][] urls, Hashtable[] options) throws Exception
```

```
public final InputStream getOZD(String[] item, String[] category, String serverDMType, Hashtable[] formparam, Hashtable[] odi param, Hashtable[] odi Paths, boolean memoallowed, String password, String id, String pwd, HttpServletRequest request, String[] displayName, String[][]
```

	urls, Hashtable[] options, Hashtable[] fxURLs) throws		
	Exception		
	OZD		
		OZD	
<b>Definition</b>	: API		serverDMType serverDMType DM_TYPE="Memory", FetchType="Batch"
	<i>item</i>	(	OZR )
	<i>category</i>		
	<i>urls</i>	OZD	URL
	<i>serverDMType</i>	Memory :Memory)	File (
	<i>formparam</i>	:	Hashtable (String), (string )
		ODI	
<b>Argument</b>	<i>odi param</i>	: ODI	Hashtable ODI (String), (Hashtable) Hashtable (String), (String)
	<i>odi path</i>	ODI	
	<i>memoallowed</i>	OZD	
	<i>password</i>	OZD	
	<i>id</i>		
	<i>pwd</i>		
	<i>request</i>	HttpServletRequest	
	<i>displayName</i>		
	<i>urls</i>	URL	
	<i>options</i>	OZD	key "Option"

	FX	URL
		URL
		<ul style="list-style-type: none"> <li>fxURL fxName value가 String url</li> <li>, String[] String[0]= URL,</li> <li>String[1]=</li> </ul>
<i>fxURLs</i>		ex) <pre> Hashtable[] fxURLs = new Hashtable[2]; fxURLs[0] = new Hashtable(); String[] url = new String[2]; url[0] = "http://127.0.0.1:8080/ozcar.jsp?param1=" "; url[1] = "base64/ksc5601"; fxURLs[0].put("sample",url);                     </pre>

■ **getOZDWithSDM**

<b>Prototype</b>	<pre> public final InputStream getOZDWithSDM(String item, String category, Hashtable sdm, Hashtable formParam, boolean memoAllowed, String password, String id, String pwd, HttpServletRequest request) throws Exception                     </pre>		
<b>Definition</b>	- Servlet API SDM "SDM → gzip → Base64 Encode → Encode URI" - SDM "Base64 Decode → ungzip → SDM" - getOZDWithSDM doGet doPost doPost getOZDWithSDM		
<b>Argument</b>	<i>item</i>	(	OZR )
	<i>category</i>		
	<i>sdm</i>	OZD	SDM
	<i>formparam</i>	:	Hashtable (String), (string )
	<i>memoAllowed</i>	OZD	

---

<i>password</i>	OZD
<i>id</i>	
<i>pwd</i>	
<i>request</i>	HttpServletRequest

---

### Sample : RequestOZDSample.java

```
package sample;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.ServletConfig;
import oz.server.OZServlet;
import javax.servlet.ServletException;

import java.io.*;
import java.util.*;

import oz.framework.cp.io.OZDataOutputStream;

public class RequestOZDSample extends OZServlet {
    private static final int PROTOCOL_VER = 2005;
    private static final String _ROOT_PATH = "D:/";
    private byte[] _buf = new byte[1024];
    public void init(ServletConfig config) throws ServletException {
        super.init(config);
    }

    public void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException
    {
        doPost(request, response);
    }

    public void doPost(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException
    {
        try {
            System.out.println("Start to get OZD");
            long _JOB_ID = System.currentTimeMillis();
            String tempOZDFile = _ROOT_PATH + "sample" + _JOB_ID + ".ozd";
            String tempWMFile = _ROOT_PATH + "sample" + _JOB_ID + ".mtx";
            // Make OZD
            makeOZD(tempOZDFile, request);
        }
    }
}
```

```
    }
    catch(Exception e) {
        ByteArrayOutputStream bout = new ByteArrayOutputStream();
        e.printStackTrace(new PrintStream(bout));
        bout.flush();
        byte[] b = bout.toByteArray();
        String error = new String(bout.toByteArray());
        bout.close();
        System.out.println(error);
        PrintWriter writer = response.getWriter();
        writer.println("RequestOZDSample : Error");
        writer.println();
        writer.println(error);
        writer.flush();
    }
}

private void writeFile(String filename, OZDataOutputStream out)
    throws IOException
{
    BufferedInputStream bin = null;
    try {
        ByteArrayOutputStream bout = new ByteArrayOutputStream();
        int len;
        bin = new BufferedInputStream(new FileInputStream(filename));
        while((len = bin.read(_buf)) >= 0) {
            bout.write(_buf, 0, len);
        }
        bout.flush();
        byte[] buf = bout.toByteArray();
        bout.close();
        out.writeInt(buf.length);
        out.write(buf, 0, buf.length);
    }
    finally {
        if(bin != null) {
            try {
                bin.close();
            }
            catch(Exception ex) {
            }
        }
    }
}

private void makeOZD(String filename, HttpServletRequest request)
    throws Exception
{
```

```
InputStream stream = null;
try {
    // Info.
    String _ITEM = "parameter_test.ozr";
    String _CATEGORY = "/";
    boolean _MEMOALLOW = true;
    String _PASSWORD = "1234";
    String _UID = "admin";
    String _PWD = "admin";

    //Form parameter
    Hashtable _FORM_PARAM = new Hashtable();
    _FORM_PARAM.put("formparam1", "FORM 1");
    _FORM_PARAM.put("formparam2", "FORM 2");
    //ODI parameter
    Hashtable _ODI_PARAM = new Hashtable();
    Hashtable _odi_param = new Hashtable();
    _odi_param.put("odi param1", "ODI 1");
    _odi_param.put("odi param2", "ODI 2");
    _ODI_PARAM.put("parameter_test", _odi_param);
    // call make ozd
    stream = getOZD(_ITEM, _CATEGORY,
        _FORM_PARAM, _ODI_PARAM, _MEMOALLOW, _PASSWORD,
        _UID, _PWD, request);
    //ODI parameter
    //Hashtable _ODI_PARAM = new Hashtable();
    //Hashtable _ODI_PATH = new Hashtable();
    //_ODI_PARAM.put("odi param1", "ODI 1");
    //_ODI_PARAM.put("odi param2", "ODI 2");
    //_ODI_PARAM.put("parameter_test", _odi_param);
    //_ODI_PATH.put("parameter_test", "/test/parameter_test.odi");
    //call make ozd
    //stream = getOZD(_ITEM, _CATEGORY,
    //    _FORM_PARAM, _ODI_PARAM, _ODI_PATH, _MEMOALLOW,
    //    _PASSWORD, _UID, _PWD, request);
    FileOutputStream out = new FileOutputStream(filename);
    copy(stream, out);
    out.flush();
    out.close();
}
catch(Exception e) {
    e.printStackTrace();
    throw e;
}
finally {
    if(stream != null) {
        try {
```

```
        stream.close();
    }
    catch(Exception e) {}
    }
}

// Util method
public static int copy(InputStream is, OutputStream os) throws IOException
{
    byte[] buf = new byte[1024];
    int rt = 0;
    int len;
    while((len = is.read(buf)) >= 0) {
        os.write(buf, 0, len);
        rt += len;
    }
    return rt;
}
}
```

### Sample : RequestMultiOZDSample.java

```
package oz.server;

import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class RequestMultiOZDSample extends OZServlet {

    public String getServletName() {
        return "[ RequestMultiOZDSample ]";
    }

    public void init(ServletConfig config) throws ServletException {
    }

    public void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException
    {
        makeOZD(request);
    }
}
```

```
}

public void doPost(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException
{
    makeOZD(request);
}

private void makeOZD(HttpServletRequest request) throws ServletException,
IOException
{
    InputStream stream = null;
    try {

        //
int size = 2;
        String[] item = new String[size];
        String[] category = new String[size];
        Hashtable[] formparam = new Hashtable[size];
        Hashtable[] odi params = new Hashtable[size];
        boolean memoAllowed = false;
        String ozdPassword = "";
        String serverID = "admin";
        String serverPWD = "admin";

        String[] displayName = new String[size];
        String[][] urls = {
            {"http://localhost:8080/tomcat.gif", "ozp:///REQ/b.gif"}
            ,
            {"http://localhost:8080/tomcat.gif", "ozp:///REQ/b.gif"}
        };

        Hashtable[] options = new Hashtable[size];

        for(int i=0; i<size; i++) {
            item[i] = "parameter_test.ozr";
            category[i] = "/";

            formparam[i] = new Hashtable();
            formparam[i].put("formparam1", "FORM1="+i);
            formparam[i].put("formparam2", "FORM2="+i);

            Hashtable odi param = new Hashtable();
```

```
odi param.put("odi param1", "ODI PARAM1=odi "+i);
odi param.put("odi param2", "ODI PARAM2=odi "+i);

odi params[i] = new Hashtable();
odi params[i].put("parameter_test", odi param);

displayName[i] = (i+1)+"="+item[i];

options[i] = new Hashtable();
if (i == 0)
    options[i].put("use_ozd_parameter", "true");
else
    options[i].put("use_ozd_parameter", "false");
}

stream = getOZD(item, category, formparam, odi params, memoAllowed,
ozdPassword, serverID, serverPWD, request, displayName, urls, options);
byte[] ozd = getBytes(stream);

FileOutputStream out = new FileOutputStream("D:/MultiImage.ozd");
out.write(ozd, 0, ozd.length);
out.flush();
out.close();
}
catch(Exception e) {
    e.printStackTrace();
    throw new ServletException(makeStackTrace(e));
}
finally {
    if(stream != null) {
        try {
            stream.close();
        }
        catch(Exception e) {}
    }
}

private byte[] getBytes(InputStream is) {

    BufferedInputStream bis = new BufferedInputStream(is);

    int totalRead = 0;
```

```
int bytesRead = 0;
byte[] buffer = new byte[10000];
byte[] ret = new byte[0];
byte[] tempBuff;

try {
    while((bytesRead = bis.read(buffer)) >=0) {
        totalRead += bytesRead;
        tempBuff = new byte[totalRead];
        if(ret.length > 0 ) {
            System.arraycopy(ret, 0, tempBuff, 0, ret.length);
        }
        if(bytesRead > 0) {
            System.arraycopy(buffer, 0, tempBuff, ret.length, bytesRead);
        }
        ret = tempBuff;
    }
    bis.close();
} catch(Exception e) {
    return null;
}

return ret;
}

private String makeStackTrace(Throwable t) throws IOException {
    ByteArrayOutputStream bout = new ByteArrayOutputStream();
    t.printStackTrace(new PrintStream(bout));
    bout.flush();
    byte[] b = bout.toByteArray();
    String error = new String(bout.toByteArray());

    return getServletName() + "\n" + error;
}
}
```

### Sample : OZOZMakerServlet.java

```
package sample;

import java.io.*;
```

```
import java.io.IOException;
import java.util.Hashtable;

import javax.servlet.*;
import javax.servlet.http.*;

import oz.server.OZServlet;

public class OZ0ZMakerServlet extends OZServlet {
    private static final Long serialVersionUID = 1L;

    public void init(ServletConfig config) throws ServletException{
        super.init(config);
    }

    public void doGet(HttpServletRequest req, HttpServletResponse res)
throws ServletException, IOException {
        doPost(req, res);
    }

    public void doPost(HttpServletRequest req, HttpServletResponse res)
throws ServletException, IOException {
        //getOZDWithSDM      Argument      .
        String inOutPath = "D:\\ServerRepository\\";
        String ozrName = "SDM_OZD_MAKE_TEST.ozr";
        String category = "/";
        boolean isMemo = false;
        String reportPassword = null;
        String serverID = "admin";
        String serverPwd = "admin";

        Hashtable param = new Hashtable();
        param.put("param1", "OZ PARAMETER ONE");
        param.put("param2", "OZ PARAMETER TWO");

        Hashtable sdm = new Hashtable();
        FileInputStream fis1 = new FileInputStream(inOutPath +
                                                    "SDM_OZD_MAKE_TEST.sdm");
        FileInputStream fis2 = new FileInputStream(inOutPath +
                                                    "SDM_OZD_MAKE_TEST1.sdm");
        sdm.put("SDM_OZD_MAKE_TEST", fis1);
        sdm.put("SDM_OZD_MAKE_TEST1", fis2);

        InputStream is = null;
        FileOutputStream fos = null;
        try {
            //getOZDWithSDM      Argument      SDM      .
            is = getOZDWithSDM(ozrName, category, sdm, param,
```

```
        isMemo, reportPassword, serverID,
        serverPwd, req);
        fos = new FileOutputStream(inOutPath +
"RESULT.ozd");

        byte[] buffer = new byte[1024];
        int len = 0;
        while((len = is.read(buffer)) > 0) {
            fos.write(buffer, 0, len);
        }
    } catch(Exception e) {
        e.printStackTrace();
    } finally {
        if(fos != null) {
            fos.close();
        }

        if(is != null) {
            is.close();
        }
        fis1.close();
        fis2.close();
    }
}

public void destroy() {
    super.destroy();
}
}
```

## API

- Class Program
- Class Publisher
- Class Scheduler
- Class TaskHolidayInfo
- Class TaskHolidayGroupInfo

API

API

<b>Program</b>	
<b>Publisher</b>	
<b>Scheduler</b>	
<b>TaskHolidayInfo</b>	
<b>TaskHolidayGroupInfo</b>	

API

가

<b>ozsfw50.jar</b>	Scheduler server
<b>log4.jar</b>	Server (API Log classpath "log4.jar" )



	<i>folder</i>
--	---------------

■ **downloadFile**

<b>Prototype</b>	<code>public byte[] downloadFile(ServerInfo s, String file)</code>
<b>Definition</b>	.
<b>Argument</b>	<i>s</i> <i>file</i>

■ **getExternalProgramList**

<b>Prototype</b>	<code>public FileInfo[] getExternalProgramList(ServerInfo s, String folder)</code>
<b>Definition</b>	가 .
<b>Argument</b>	<i>s</i> <i>folder</i>

■ **removeFiles**

<b>Prototype</b>	<code>public void removeFiles(ServerInfo s, String folder, String[] files)</code>
<b>Definition</b>	.
<b>Argument</b>	<i>s</i> <i>folder</i> <i>files</i>

■ **removeFolder**

<b>Prototype</b>	<code>public void removeFolder(ServerInfo s, String folder, boolean isAll)</code>
<b>Definition</b>	.
<b>Argument</b>	<i>s</i> <i>folder</i> <i>isAll</i> true : 가 ( ) false : 가 ( )

■ **uploadFile**

<b>Prototype</b>	<code>public void uploadFile(ServerInfo s, String file, byte[] b)</code>
<b>Definition</b>	<code>( )</code>
<b>Argument</b>	<code>s</code> <code>file</code> <code>b</code>

## Class

- **SchedulerException(oz.scheduler.SchedulerException)**

Exception

- **ServerInfo(oz.scheduler.ServerInfo)**

가

- **setIsDaemon**

<b>Prototype</b>	<code>public final void setIsDaemon(boolean isDaemon)</code>
<b>Definition</b>	<code>.</code> 가 Daemon
<b>Argument</b>	<code>isDaemon</code> <ul style="list-style-type: none"> <li>• true : Daemon</li> <li>• false : Servlet</li> </ul>

- **setIP**

<b>Prototype</b>	<code>public final void setIP(String ip)</code>
<b>Definition</b>	IP . Server가 Daemon
<b>Argument</b>	<code>ip</code> IP

- **setPortNo**

<b>Prototype</b>	<code>public final void setPortNo(int portNo)</code>
<b>Definition</b>	Port . Server가 Daemon
<b>Argument</b>	<code>portNo</code> Port

- setURL

<b>Prototype</b>	public final void setURL(String url) throws Illegal ArgumentExcepti on
------------------	---

<b>Definition</b>	URL . Server가 Servlet
-------------------	-----------------------

<b>Argument</b>	<i>url</i> URL , 'http://'
-----------------	----------------------------

- setID

<b>Prototype</b>	public final void setID(String id)
------------------	------------------------------------

<b>Definition</b>	ID
-------------------	----

<b>Argument</b>	<i>id</i> ID
-----------------	--------------

- setPWD

<b>Prototype</b>	public final void setPWD(String pwd)
------------------	--------------------------------------

<b>Definition</b>	.
-------------------	---

<b>Argument</b>	<i>pwd</i>
-----------------	------------

- FileInfo(oz.scheduler.FileInfo.class)

( / , , , ) 가

- public boolean isDirectory : ( )
- public String name :
- public long size :
- public long lastModified : , 1970 1 1  
00:00:00 GMT

### Sample : ProgramSample.java

```
package sampl e;

import oz. schedul er. ServerI nfo;
import oz. schedul er. Fi l eI nfo;
```

```

import oz.framework.api.Program;
import org.apache.log4j.*;

public class ProgramSample {
    public static void main(String[] args) {
        //          가          가          .
        BasicConfigurator.configure();

        //
        String IP = "127.0.0.1"; //          가          IP
        int PORT = 9521; //          가          TCP

        Program program = null;
        try {
            program = new Program(IP, PORT);

            //
            ServerInfo serverInfo = new ServerInfo();

            serverInfo.setID("admin"); //
            serverInfo.setPWD("admin"); //
            serverInfo.setIP(IP); //OZ Server IP          . daemon
            serverInfo.setIsDaemon(true);
            //OZ Server          daemon, servlet
            serverInfo.setPortNo(8003);
            /* OZ Server Port          . daemon
             * s.setURL(null); //OZ Server URL          .
             * URL          ,          /* http: //
             *          . Server 가 Servlet Type          . */

            //          (createFolder)
            String folderName = "testFolder"; //
            program.createFolder(serverInfo, folderName);

            //          (uploadFile)
            String fileName = "testProgram.bat"; //
            byte [] fileBytes = fileName.getBytes();
            program.uploadFile(serverInfo, folderName+"/"+fileName, fileBytes);

            //          (downloadFile)
            String downFileName = "testProgram.bat"; //
            byte byteArray[] =
                program.downloadFile(serverInfo, folderName+"/"+downFileName);

            //          (getExternalProgramList)
            FileInfo fileInfo[] =
                program.getExternalProgramList(serverInfo, folderName);
            for(int i=0; i<fileInfo.length; i++){

```

```
        FileInfo fi = fileInfo[i];
        System.out.println(i);
        System.out.println(" isDirectory " + fi.isDirectory());
        System.out.println(" name " + fi.name);
        System.out.println(" size " + fi.size);
        System.out.println(" lastModified " + fi.lastModified);
        System.out.println();
    }

    //          (removeFiles)
    //          ( )          String
    String file1 = "testProgram.bat";
    String file2 = "testProgram2.bat";
    String [] files = new String[]{file1 }; //
    program.removeFiles(serverInfo, folderName, files);

    //          (removeFolder)
    boolean isAll = true; //
    String folderNameToRemove = "testFolder";
    //
    program.removeFolder(serverInfo, folderNameToRemove, isAll);

}
catch(Excepti on e)
{
    e.printStackTrace();
}
}
```



■ **downloadFile**

**Prototype** public byte[] downloadFile(ServerInfo s, String file)

**Definition**

**Argument** s  
file

■ **getPublishedFiles**

**Prototype** public FileInfo[] getPublishedFiles(ServerInfo s, String folder)

**Definition** 가

**Argument** s  
folder

■ **removeFiles**

**Prototype** public void removeFiles(ServerInfo s, String folder, String[] files)

**Definition**

**Argument** s  
folder  
files

■ **removeFolder**

**Prototype** public void removeFolder(ServerInfo s, String folder, boolean isAll)

**Definition**

**Argument** s  
folder  
isAll true : 가 ( )  
false : 가 ( )

**Class**

### ■ ServerInfo(oz.scheduler.ServerInfo)

Program class " class"

### ■ FileInfo(oz.scheduler.FileInfo.class)

Program class " class"

## Sample : PublisherSample.java

```
package sample;

import oz.framework.api.Publisher;
import oz.scheduler.FileInfo;
import oz.scheduler.ServerInfo;
import org.apache.log4j.*;

public class PublisherSample {
    public static void main(String[] args) {
        // 가 가
        BasicConfigurator.configure();

        //
        String IP = "127.0.0.1"; // 가 IP
        int PORT = 9521; // 가 TCP

        Publisher publisher = null;
        try {
            publisher = new Publisher(IP, PORT);

            //
            ServerInfo serverInfo = new ServerInfo();

            serverInfo.setID("admin"); //
            serverInfo.setPWD("admin"); //
            serverInfo.setIP(IP); //OZ Server IP . daemon
            serverInfo.setIsDaemon(true);
            //OZ Server . daemon, servlet
            serverInfo.setPortNo(8003);
            /* OZ Server Port . daemon
            * s.setURL(null); //OZ Server URL
            * URL , /*' http: //'
            * . Server 가 Servlet Type . */

            // (createFolder)
```

```
String folderName = "excel"; //
//publisher.createFolder(serverInfo, folderName);

//                                (downloadFile)
String fileName = "customer.xls"; //
byte byteArray[] = publisher.downloadFile(serverInfo, fileName);

//                                (getPublishedFiles)
File[] fileInfoList[] =
    publisher.getPublishedFiles(serverInfo, folderName);
for(int i=0; i<fileInfoList.length; i++) {
    File fi = fileInfoList[i];
    System.out.println(i);
    System.out.println(" isDirectory="+fi.isDirectory());
    System.out.println(" name="+fi.name());
    System.out.println(" size="+fi.size());
    System.out.println(" lastModified="+fi.lastModified());
    System.out.println();
}
//                                (removeFiles)
//                                ( ) String
// customer.xls orderInfo.xls
String file1 = "customer.xls";
String file2 = "orderInfo.xls";
String testFolderName = "excel";
String [] filesToRemove = new String[]{file1, file2};
//
publisher.removeFiles(serverInfo, testFolderName, filesToRemove);

//
boolean isAll = true; //
String folderNameToRemove = "excel"; //
publisher.removeFolder(serverInfo, folderNameToRemove, isAll);

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

## Class Scheduler

### Constructor Summary

- `scheduler(String ip, int port)`

### Method Summary

- `String createTask(ServerInfo s, NameValueCollection configMap, NameValueCollection exportMap)`
- `Vector getTask(ServerInfo s)`
- `public TaskResult[] getTaskResult(ServerInfo s, String from, String to, String taskId)`
- `SortProperties[] getTaskProperties(ServerInfo info, String taskId)`
- `void removeTask(ServerInfo s, String task)`
- `String modifyTask(ServerInfo s, String taskId, SortProperties p, SortProperties exportMap)`
- `boolean taskPause(ServerInfo s, String task)`
- `boolean taskResume(ServerInfo s, String task)`
- `void stop(ServerInfo s, boolean waitTask)`
- `public boolean export(ServerInfo s, SortProperties configMap, SortProperties exportMap)`
- `boolean makePDF(ServerInfo s, SortProperties configMap, SortProperties exportMap)`
- `public boolean print(ServerInfo s, SortProperties configMap, SortProperties printMap)`
- `SortProperties getConfiguration(ServerInfo s)`
- `void modifyConfiguration(ServerInfo s, SortProperties configMap, SortProperties exportMap)`
- `boolean ping()`
- `String[] getOZSList(String path)`
- `void ozsVerConv(String oldPath, String newPath)`

- `boolean addTaskHolidayInfo(TaskHolidayInfo value)`
- `boolean modifyTaskHolidayInfo(String old_key, TaskHolidayInfo new_value)`
- `boolean deleteTaskHolidayInfo(String key)`
- `boolean deleteTaskHolidayInfo(String[] keys)`
- `boolean addTaskHolidayGroupInfo(TaskHolidayGroupInfo value)`
- `boolean modifyTaskHolidayGroupInfo(String old_key, TaskHolidayInfo new_value)`
- `boolean deleteTaskHolidayGroupInfo(String key)`
- `OZMap getTaskHolidayInfoList()`
- `OZMap getTaskHolidayGroupInfoList()`
- `void saveTaskHoliday()`
- `DirectExportResult directExport(ServerInfo s, SortProperties configMap, SortProperties exportMap) throws SchedulerException`
- `DirectPrintResult directPrint(ServerInfo s, SortProperties configMap, SortProperties printMap) throws SchedulerException`

## Constructor Detail

<b>Prototype</b>	<code>public scheduler(String ip, int port)</code>												
<b>Argument</b>	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;"><i>ip</i></td> <td style="width: 40%; text-align: center;">가</td> <td style="width: 30%; text-align: right;">IP</td> </tr> <tr> <td></td> <td colspan="2" style="text-align: center;">ex) String ip = "127.0.0.1";</td> </tr> <tr> <td><i>port</i></td> <td style="text-align: center;">( :9521)</td> <td></td> </tr> <tr> <td></td> <td colspan="2" style="text-align: center;">ex) int port = 9521;</td> </tr> </table>	<i>ip</i>	가	IP		ex) String ip = "127.0.0.1";		<i>port</i>	( :9521)			ex) int port = 9521;	
<i>ip</i>	가	IP											
	ex) String ip = "127.0.0.1";												
<i>port</i>	( :9521)												
	ex) int port = 9521;												

## Method Detail

- **createTask**

---

**Prototype**    `public String createTask(ServerInfo s, NameValueCollection configMap, NameValueCollection exportMap)`

---

			ID 가
<b>Definition</b>		: CreateTask	Thread param
			s
<b>Argument</b>	<i>configMap</i>	key	"Option"
	<i>exportMap</i>	key	"Option"

■ **getTask**

<b>Prototype</b>	public Vector getTask(ServerInfo s)
<b>Definition</b>	가
<b>Argument</b>	s

■ **getTaskResult**

<b>Prototype</b>	public TaskResult[] getTaskResult(ServerInfo s, String from, String to, String taskId)
<b>Definition</b>	가
	s
<b>Argument</b>	<i>from</i> 가
	<i>to</i> 가
	<i>taskId</i> 가

■ **getTaskProperties**

<b>Prototype</b>	public SortProperties[] getTaskProperties(ServerInfo s, String taskId)
<b>Definition</b>	(configMap) (exportMap) 가
	s
<b>Argument</b>	<i>taskId</i> 가

■ **removeTask**

**Prototype** `public void removeTask(ServerInfo s, String taskId)`

**Definition**

**Argument** `s`  
`taskId`

■ **modifyTask**

**Prototype** `public String modifyTask(ServerInfo s, String taskId, SortProperties configMap, SortProperties exportMap)`

**Definition**

**Argument** `s`  
`taskId`  
`configMap` key "Option"  
`exportMap` key "Option"

■ **taskPause**

**Prototype** `public boolean taskPause(ServerInfo s, String taskId)`

**Definition**

**Argument** `s`  
`taskId`

■ **taskResume**

**Prototype** `public boolean taskResume(ServerInfo s, String taskId)`

**Definition**

**Argument** `s`  
`taskId`

■ **stop**

**Prototype** `public void stop(ServerInfo s, boolean waitTaskId)`

**Definition**

*S*  
*waitTaskId*

■ **export**

**Prototype** `public boolean export(ServerInfo s, SortProperties configMap, SortProperties exportMap)`

**Definition**  `: "ViewType=None"`  
 `: *.pdf, *.ozd, *.html, *.jpg, *.xls, *.doc, *.svg, *.txt, *.ppt, *.tif, *.csv 가 .`  
*S*

**Argument** `configMap` key "Option"  
`exportMap` key "Option"

`: export` API 가 COM ASP  
"Appendix 1. SchedulerCOM "

■ **makePDF**

**Prototype** `public boolean makePDF(ServerInfo s, SortProperties configMap, SortProperties exportMap)`

**Definition** `PDF`  
 `: "ViewType=None"`

**Argument** *S*  
`configMap` key `creatTask()` "Option"  
PDF

	PDF	key	"Option"
<i>exportMap</i>		. creatTask()	PDF
: makePDF	API	가	COM ASP
	"Appendix 1. SchedulerCOM"		

■ **print**

<b>Prototype</b>	public boolean print(ServerInfo s, SortProperties configMap, SortProperties printMap)									
<b>Definition</b>	<pre>         가         : "task_type=viewerTag"         "ViewType=None"         s     </pre>									
<b>Argument</b>	<table border="1"> <tr> <td><i>configMap</i></td> <td>key</td> <td>"Option"</td> </tr> <tr> <td><i>printMap</i></td> <td>key</td> <td>"Option"</td> </tr> <tr> <td></td> <td>, print.mode = silent</td> <td></td> </tr> </table>	<i>configMap</i>	key	"Option"	<i>printMap</i>	key	"Option"		, print.mode = silent	
<i>configMap</i>	key	"Option"								
<i>printMap</i>	key	"Option"								
	, print.mode = silent									

- print API

, print API

- print API

```

viewer.allowmultiframe=true
viewer.mode=print
viewer.printcommand=true
viewer.showerrorMessage=false
viewer.useprogressbar=false
print.ingnoreerror=false
    
```

```
print.mode=silent
export.confirmsave=false
export.format=""
information.debug=debug
```

■ **getConfiguration**

**Prototype** public SortProperties getConfiguration(ServerInfos)

**Definition** .

**Argument** s

■ **modifyConfiguration**

**Prototype** public void modifyConfiguration(ServerInfos, SortProperties configMap)

**Definition** .

s

**Argument** configMap key "Option"

■ **ping**

**Prototype** public boolean ping()

**Definition** .

■ **getOZSList**

**Prototype** public String[] getOZSList(String path)

**Definition** OZS 가 . (path

OZS 가 , path  
OZS 가 .  
)

**Argument** path : OZS 가  
/%SCH\_HOME%/[path]/  
path sample  
/%SCH\_HOME%/sample/  
OZS 가 .

■ **ozsVerConv**

<b>Prototype</b>	public void ozsVerConv(String oldPath, String newPath)												
<b>Definition</b>	2.5 OZS												
<b>Argument</b>	<table border="0"> <tr> <td><i>oldPath</i></td> <td>OZS</td> <td>:</td> <td>/%SCH_HOME%/[oldPath]/</td> </tr> <tr> <td><i>newPath</i></td> <td>OZS</td> <td>:</td> <td>/%SCH_HOME%/[newPath]/</td> </tr> <tr> <td></td> <td></td> <td></td> <td>newPath oldPath</td> </tr> </table>	<i>oldPath</i>	OZS	:	/%SCH_HOME%/[oldPath]/	<i>newPath</i>	OZS	:	/%SCH_HOME%/[newPath]/				newPath oldPath
<i>oldPath</i>	OZS	:	/%SCH_HOME%/[oldPath]/										
<i>newPath</i>	OZS	:	/%SCH_HOME%/[newPath]/										
			newPath oldPath										

■ **addTaskHolidayInfo**

<b>Prototype</b>	public boolean addTaskHolidayInfo(TaskHolidayInfo value)
<b>Definition</b>	가
<b>Argument</b>	<i>value</i>

■ **modifyTaskHolidayInfo**

<b>Prototype</b>	public boolean modifyTaskHolidayInfo(String old_key, TaskHolidayInfo new_value)				
<b>Definition</b>	.				
<b>Argument</b>	<table border="0"> <tr> <td><i>old_key</i></td> <td></td> </tr> <tr> <td><i>new_value</i></td> <td></td> </tr> </table>	<i>old_key</i>		<i>new_value</i>	
<i>old_key</i>					
<i>new_value</i>					

■ **deleteTaskHolidayInfo**

<b>Prototype</b>	<pre>// public boolean deleteTaskHolidayInfo(String[] keys) //</pre>
<b>Definition</b>	public boolean deleteTaskHolidayInfo(String key)
<b>Argument</b>	<i>key</i>

■ **addTaskHolidayGroupInfo**

<b>Prototype</b>	public boolean addTaskHolidayGroupInfo(TaskHolidayGroupInfo value)
<b>Definition</b>	가

Argument	<i>value</i>
----------	--------------

■ **modifyTaskHolidayGroupInfo**

Prototype	<code>public boolean modifyTaskHolidayGroupInfo(String old_key, TaskHolidayGroupInfo new_value)</code>
-----------	--

Definition	.
------------	---

Argument	<i>old_key</i>
----------	----------------

	<i>new_value</i>
--	------------------

■ **deleteTaskHolidayGroup**

Prototype	<code>public boolean deleteTaskHolidayGroupInfo(String key)</code>
-----------	--

Definition	.
------------	---

Argument	<i>key</i>
----------	------------

■ **getTaskHolidayInfoList**

Prototype	<code>public OZMap getTaskHolidayInfoList()</code>
-----------	--

Definition	가 .
------------	-----

■ **getTaskHolidayGroupInfoList**

Prototype	<code>public OZMap getTaskHolidayGroupInfoList()</code>
-----------	---

Definition	가 .
------------	-----

■ **saveTaskHoliday**

Prototype	<code>public void saveTaskHoliday()</code>
-----------	--

Definition	xml .
------------	-------

■ **directExport**

Prototype	<code>public DirectExportResult directExport(ServerInfos, SortProperties configMap, SortProperties exportMap) throws SchedulerException</code>
-----------	--

**Definition** : "ViewType=None"  
 : \*.pdf, \*.ozd, \*.html, \*.jpg,  
 \*.xls, \*.doc, \*.svg, \*.txt, \*.ppt, \*.tif, \*.csv 가

*S*

**Argument**

<i>configMap</i>	key	"Option"
<i>exportMap</i>	key	"Option"

■ **directPrint**

**Prototype** public DirectPrintResult directPrint(ServerInfo s, SortProperties configMap, SortProperties printMap) throws SchedulerException

**Definition** : "task\_type=viewerTag"  
 "ViewType=None"

*S*

**Argument**

<i>configMap</i>	key	"Option"
<i>printMap</i>	key	"Option"

, print.mode = silent

**Class**

■ **ServerInfo(oz.scheduler.ServerInfo)**

Program class " class"

### ■ **SortProperties (oz.util.SortProperties)**

Cache class " class"

### ■ **TaskHolidayInfo(oz.scheduler.Holiday)**

TaskHolidayInfo class " class"

### ■ **TaskHolidayGroupInfo(oz.scheduler.Holiday)**

TaskHolidayInfo class " class"

### ■ **TaskResult(oz.scheduler.TaskResult)**

-

- public String taskID: ID
- public String completedTime :
- public int isSuccessfuCode:
- public String isSuccessfu:
- public String formFileName:
- public String Parameter:
- public String schedulingType:
- public String exportFileList:
- public String errorMsg:

### ■ **DirectTaskResult(oz.scheduler.DirectTaskResult)**

-

- public String taskID: ID
- public String completedTime :
- public String executeTime: ( : )
- public boolean isSuccessfu:
- public String formName:

### ■ **oz.scheduler.DirectExportResult extends DirectTaskResult**

-

- public String exportFileList:

- public int pageCount:

```

        ,
        가 4
        xls, doc
        pageCount 8
        "viewer.largebundle=true"
        pageCount
    1
    
```

■ **oz.scheduler.DirectPrintResult extends DirectTaskResult**

- public int pageCount:
- public int pageCopy:
- public String pageRange:
- public String printerName :
- public String printerDriverName :

**Option**

```

        "ViewType" "None"
        API
        , OZD
        viewer.mode "export" "print"
        가 "export" viewer.mode
        가
    
```

<b>viewer.useprogressbar</b>	false
<b>viewer.allowmultiframe</b>	true
<b>export.mode</b>	silent

<b>export.confirmsave</b>	false
<b>print.mode</b>	silent
<b>print.ignoreerror</b>	false
<b>information.debug</b>	debug
<b>viewer.showerrormessage</b>	false

- reportname
- displayname
- external\_program\_check
- external\_program\_command
- image\_url\_size
- image\_url\_n
- parameter\_count
- parameter\_name\_n
- parameter\_value\_n
- option\_properties\_size
- option\_property\_name\_n
- option\_property\_value\_n

Key	Value	
reportname		ex) setProperty("reportname", "/category/sample.ozr")

Key	Value	
displayname		ex) setProperty("displayname", "sample")

Key	Value	
-----	-------	--

<b>reportcount</b>		( : 1) ex) setProperty("reportcount", "2")
--------------------	--	---

Key	Value	
<b>dm_server_check</b>	"check" "null"	ex) setProperty("dm_server_check", "check")
<b>dm_server_name</b>	SDM	(Repository odi가 category SDM ) ex) setProperty("dm_server_name", test1.sdm")
<b>odi_name</b>	ODI	ODI ex) setProperty("odi_name", "testodi.odi")
<b>odi_category_name</b>		ODI가 ( "/" ) ex) setProperty("odi_category_name", "/test")

Key	Value	
<b>task_type</b>	"viewerTag" "none" "sdm"	viewerTag : OZU g , none : (ViewType None 가 ) sdm : SDM ex) setProperty("task_type ", "viewerTag")

Key	Value	
<b>external_program_check</b>	"check" "null"	ex) setProperty("external_program_check", "check")

<b>external_program_command</b>		("SCH_HOME/External ) ex) SetProperty("external_program_command", "notepad.bat")
---------------------------------	--	---

- URL

Key	Value	
<b>image_url_size</b>	url	OZD url ex) SetProperty("image_url_size", "1")
<b>image_url_1</b> ... <b>image_url_n</b>	url	url (n : url ) ex) SetProperty("image_url_1", "http://127.0.0.1/sample.jpg")

■

Key	Value	
<b>mail_check</b>	"check" "null"	ex) SetProperty("mail_check", "check")
<b>mail_notify_error_check</b>	"check" "null"	ex) SetProperty("mail_notify_error_check", "null")
<b>mail_recipient_to</b>		ex) SetProperty("mail_recipient_to", "gil_dong@forcs.com")
<b>mail_recipient_cc</b>		ex) SetProperty("mail_recipient_cc", "aaa@forcs.com")
<b>mail_recipient_bcc</b>		ex) SetProperty("mail_recipient_bcc", "bbb@forcs.com")
<b>mail_subject</b>		ex) SetProperty("mail_subject", " ")
<b>mail_text_message</b>		ex) SetProperty("mail_text_message", " ")
<b>mail_html_comment</b>	"check" "null"	HTML ex) SetProperty("mail_html_comment", "check")

<b>html_mail_content</b>	"check" "null"	HTML ex) setProperty("html_mail_content", "check")
<b>mail_attach_list</b>		( '/' ) ex) setProperty("mail_attach_list", "excel/pdf/word")

■

Key	Value	
<b>file_export_list</b>		( 가 '/' ) ex) setProperty("file_export_list ", "xls/pdf/tif")

■

Key	Value	
<b>parameter_count</b>		ex) setProperty("parameter_count", "1")
<b>parameter_name_1</b> ... <b>parameter_name_n</b>		(n : ) ex) setProperty("parameter_name_1", "[FORM].empNo")
<b>parameter_value_1</b> ... <b>parameter_value_n</b>		(n : ) ex) setProperty("parameter_value_1", " 10")

- ODI
  - ODI
      - ODI
      - ODI
- "[FORM]."  
 "[FORM].empNo"  
 "testodi"  
 "id"  
 "testodi.id"  
 empNo

■ ODI

Key	Value	
<b>odi_parameter_count</b>		ex) setProperty("odi_parameter_count", "1")
<b>odi_parameter_name_1</b> ... <b>odi_parameter_name_n</b>		(n : ) ex) setProperty("odi_parameter_name_1", "id")
<b>odi_parameter_value_1</b> ... <b>odi_parameter_value_n</b>		(n : ) ex) setProperty("odi_parameter_value_1", " 10")

: ODI

ODI

ODI

■

Key	Value	
<b>use_ozd_parameter</b>	true/false	OZD OZD ex) setProperty("use_ozd_parameter ", "false")

■

Key	Value	
<b>launch_type</b>	"once" "immediately" "periodically"	once : immediately : periodically : ex) setProperty("launch_type", "once")

■ launch\_type = once

Key	Value	
<b>execution_year</b>		- ex) setProperty("execution_year", "2005")

<b>execution_month</b>		- ex) setProperty("execution_month", "12")
<b>execution_day</b>		- ex) setProperty("execution_day", "30")
<b>execution_hour</b>		- ex) setProperty("execution_hour", "10")
<b>execution_min</b>		- ex) setProperty("execution_minute", "30")

- launch\_type = periodically

Key	Value	
<b>start_year</b>		- ex) setProperty("start_year", "2002")
<b>start_month</b>		- ex) setProperty("start_month", "11")
<b>start_day</b>		- ex) setProperty("start_day", "16")
<b>periodically_execution_day_type</b>	"daily" "weekly" "monthly"	ex) setProperty("periodically_execution_day_type", "daily")

- periodically\_execution\_day\_type = daily

Key	Value	
<b>daily_type</b>	weekday	ex) setProperty("daily_type", "weekday")
<b>daily_every_days</b>		( ) ex) 2 : setProperty ("daily_every_days", "2")

- periodically\_execution\_day\_type = weekly

Key	Value	
<b>weekly_every_weeks</b>		ex) setProperty("weekly_every_weeks", "2")
<b>weekly_monday_check</b>	"check" "null"	ex) setProperty("weekly_monday_check", "check")

<b>weekly_tuesday_check</b>	"check" "null"	ex) setProperty("weekly_tuesday_check", "check")
<b>weekly_wednesday_check</b>	"check" "null"	ex) setProperty("weekly_wednesday_check", "check")
<b>weekly_thursday_check</b>	"check" "null"	ex) setProperty("weekly_thursday_check", "check")
<b>weekly_friday_check</b>	"check" "null"	ex) setProperty("weekly_friday_check", "check")
<b>weekly_saturday_check</b>	"check" "null"	ex) setProperty("weekly_saturday_check", "check")
<b>weekly_sunday_check</b>	"check" "null"	ex) setProperty("weekly_sunday_check", "check")

- periodically\_execution\_day\_type = monthly

Key	Value	
<b>monthly_every_months</b>		( ) ex) setProperty("monthly_every_months", "2")
<b>monthly_type</b>	"specific_day", "day_of_week", "user_defined"	ex) setProperty("monthly_type", "specific_day")
<b>monthly_days</b>		ex) setProperty("monthly_days", "2")
<b>monthly_which_week</b>	"T1" "T2" "T3" "T4" "T5"	T1 : T2 : T3 : T4 : T5 : ex) setProperty("monthly_which_week", "T2")
<b>monthly_which_week_day</b>	"sunday" "monday" "tuesday" "wednesday" "thursday" "friday" "saturday"	ex) : setProperty("monthly_which_week_day", "monday")

<b>monthly_user_defined_days</b>		(,) ex) 1,15 : setProperty("monthly_user_defined_days", "1,15")
----------------------------------	--	---

Key	Value	
<b>periodically_execution_time_type</b>	"once" "repeat" "user_defined"	( , , ) ex) setProperty ("periodically_execution_time_type", "once")

- periodically\_execution\_time\_type = once

Key	Value	
<b>once_hour</b>		- ex) setProperty("once_hour", "01")
<b>once_min</b>		- ex) setProperty("once_min", "00")

- periodically\_execution\_time\_type = repeat

Key	Value	
<b>repeat_every_hours</b>		- ex) setProperty("repeat_every_hours", "01")
<b>repeat_every_minutes</b>		- ex) setProperty("repeat_every_minutes", "04")
<b>repeat_start_hour</b>		- ex) setProperty("repeat_start_hour", "02")
<b>repeat_start_minute</b>		- ex) setProperty("repeat_start_minute", "05")
<b>repeat_end_hour</b>		- ex) setProperty("repeat_end_hour", "09")
<b>repeat_end_minute</b>		- ex) setProperty("repeat_end_minute", "35")

: 6가 2 5 9 35 1 4

- periodically\_execution\_time\_type = user\_defined

Key	Value	
user_defined_time		( : ) (.) ex) setProperty("user_defined_time", "01:30,13:30")



" .mailattach"

. true

, false

ex) setProperty("hdm.mailattach", "true")

key

" "

: Value " / "

"% %\Repository/"

, "FORCS"

CSV

setProperty("csv.filename",

"FORCS/test.csv")

"% %\Repository/FORCS"

"test.csv"

- modifyConfiguration ConfigMap

Key	Value	
SchedulerPort		( : "9521") ex) p.setProperty("SchedulerPort", "9521");
SchedulingInfoFile ePath		ex) p.setProperty("SchedulingInfoFilePath", "%SCH_HOME%/ScheduledTask");
SMTPServer		SMTP ex) p.setProperty("SMTPServer", "mail.forcs.com");

<b>SMTPServerProt</b>		SMTP ex) p.setProperty("SMTPServerProt", "25");
<b>MailFrom</b>		ex) p.setProperty("MailFrom", "mail@forcs.com");
<b>TempRepositoryFilePath</b>		ex) p.setProperty("TempRepositoryFilePath", "%SCH_HOME%/Temp Repository");
<b>RepositoryFileRootPath</b>		ex) p.setProperty("RepositoryFileRootPath", "%SCH_HOME%/Repository");
<b>ExternalProgramFilePath</b>		ex) p.setProperty("ExternalProgramFilePath", "%SCH_HOME%/External");
<b>ErrorNotifyToSender</b>	"true" "false"	<ul style="list-style-type: none"> <li>• true :</li> <li>• false :</li> </ul> ex) p.setProperty("ErrorNotifyToSender", "false");

### Sample : SchedulerSample.java

```

package sample;

import oz.framework.api.*;
import oz.scheduler.ServerInfo;
import oz.util.SortProperties;
import org.apache.log4j.*;

public class SchedulerSample {
    private static String[][] schedulerInfoProperties =
    {
        // Basic Configuration
        {"launch_type", "Immediately"}, // Task Execute Type
        {"launch_type", "Periodically"},
        {"start_year", "2004"},
        {"start_month", "12"},
        {"start_day", "29"},
        {"periodically_execution_day_type", "daily"},
        {"daily_type", "specific_day"},
        {"daily_every_days", "1"},
    }
}
    
```

```

//      {"periodical ly_executi on_ti me_type", "repeat"}, // "once", "repeat",
"user_defi ned"
//      {"repeat_every_hours", "00"},
//      {"repeat_every_m i nutes", "01"},
//      {"repeat_start_hour", "01"},
//      {"repeat_start_m i nute", "01"},
//      {"repeat_end_hour", "23"},
//      {"repeat_end_m i nute", "59"},

//      {"dai ly_type", "speci fi c_day"},
//      {"dai ly_every_days", "1"},
//      {"periodical ly_executi on_ti me_type", "user_defi ned"},
//      {"user_defi ned_ti me", "16: 20"},

//      {"dai ly_type", "speci fi c_day"},
//      {"dai ly_every_days", "1"},
//      {"periodical ly_executi on_ti me_type", "user_defi ned"},
//      {"user_defi ned_ti me", "16: 20"},

//task hol i day
//      {"task_hol i day", "common"},

//      {"dm_server_name", "testtesttest.sdm"},
{"cfg.type", "new"}, // Type of Task -> new, edit
//      {"cfg.task_id", "parameter_test_060823103556234_296956235.ozs"},
{"report_name", "parameter_test.ozr"},
{"category_name", "/"},
{"group_name", ""},

// ODI
{"odi_name", "parameter_test.odi"},
{"odi_category_name", "/"},

//image URL properti es
//      {"image_url_si ze", "1"},
//      {"image_url_1", "http://l ocal host: 8088/tomcat.gi f"},
//      {"image_url_2", "http://l ocal host: 8088/b.gi f"},

//      {"opti on_properti es_si ze", "1"},
//      {"opti on_property_name_1", "use_ozd_parameter"},
//      {"opti on_property_val ue_1", "true"},

// Server Data Moudle
{"dm_server_check", "check"},
{"dm_server_name", "testtesttest.sdm"},

```

```

{"parameter_count", "4"},

{"parameter_name_1", "[FORM].formparam1"},
{"parameter_value_1", "AAAA"},
{"parameter_name_2", "[FORM].formparam2"},
{"parameter_value_2", "BBBB"},
{"parameter_name_3", "parameter_test.odi param1"},
{"parameter_value_3", "CCCC"},
{"parameter_name_4", "parameter_test.odi param2"},
{"parameter_value_4", "DDDD"},

//      // It must input for ODI parameter sdm create.
{"odi_parameter_count", "2"},
{"odi_parameter_name_1", "odi param1"},
{"odi_parameter_value_1", "111"},
{"odi_parameter_name_2", "odi param2"},
{"odi_parameter_value_2", "222"},
//      {"odi_parameter_name_1", "odi param1"},
//      {"odi_parameter_name_1", "odi param2"},

// e-mail send.
//      {"mail_check", "check"},
//      {"html_mail_content", "check"},
//      {"mail_notify_error_check", "null"},
//      {"mail_recipient_to", ""},
//      {"mail_recipient_cc", ""},
//      {"mail_recipient_bcc", ""},
//      {"mail_subject", ""},
//      {"mail_text_message", "context"},
//      {"mail_attach_list", "csv/xls/html/ozd/pdf/txt/tif/doc/ppt/jpg/svg"},
{"file_export_list", "csv/xls/html/ozd/pdf/txt/tif/doc/ppt/jpg/svg"}

};

private static String[][] exportedProperties =
{

    {"csv.filename", "test.csv"}, // csv file name
    {"xls.filename", "test.xls"}, // excel file name
    {"html.filename", "test.html"}, // html file name
    {"ozd.filename", "newimage.ozd"}, //OZD file name
    {"pdf.filename", "test.pdf"}, //pdf file name
    {"txt.filename", "test.txt"}, //text file name
    {"tif.filename", "test.tif"}, //tif file name
    {"doc.filename", "test.doc"}, //word file name
    {"ppt.filename", "test.ppt"}, //ppt file name

```

```
        {"jpg.filename ", "test.jpg"}, //jpg file name
        {"svg.filename ", "test.svg"} //svg file name

    };

    public static void main(String[] args)
    {

        BasicConfigurator.configure();
        String IP = "127.0.0.1"; //scheduler server ip
        int PORT = 9521; //scheduler server port
        String[][] values = null;
        Scheduler scheduler = null;
        String taskID = null;

        try {
            scheduler = new Scheduler(IP, PORT);
            // scheduler Server Info
            ServerInfo serverInfo = new ServerInfo();

            serverInfo.setIP("127.0.0.1"); //OZ Server IP conf. In case use of
daemon
            serverInfo.setPortNo(8003);
            serverInfo.setIsDaemon(true); //OZ Server Type. -> daemon, servlet

            serverInfo.setID("admin"); //id
            serverInfo.setPWD("admin"); //pw

            SortProperties p = new SortProperties();
            setProperties(p, schedulerInfoProperties);
            // p.setProperty("task_type", "sdm");

            SortProperties exportMap = new SortProperties();
            setProperties(exportMap, exportedProperties);

            taskID = scheduler.createTask(serverInfo, p, exportMap);

            System.out.println("taskid="+taskID);

        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }

    private static void setProperties(SortProperties p, String[][] values)
    {
```

```

        for(int i=0; i<values.length; i++)
        {
            p.setProperty(values[i][0], values[i][1]);
        }
    }
}

```

### Sample : SchedulerSampleTaskExportMakePrintExport.java

```

package sample;

import oz.framework.api.*;
import oz.scheduler.ServerInfo;
import oz.scheduler.ScheduledTask;
import oz.util.SortProperties;
import org.apache.log4j.*;
import oz.scheduler.SchedulerException;

public class SchedulerSampleTaskExportMakePrintExport {
    public SchedulerSampleTaskExportMakePrintExport() {
    }

    public static void main(String[] args) throws SchedulerException {
        BasicConfigurator.configure();

        Scheduler scheduler = null;
        scheduler = new Scheduler("127.0.0.1", 9521);

        //
        ServerInfo serverInfo = new ServerInfo();

        serverInfo.setID("admin"); //
        serverInfo.setPWD("admin"); //
        serverInfo.setIsDaemon(true);

        serverInfo.setIP("127.0.0.1"); //OZ Server IP . daemon
        serverInfo.setPortNo(8003);

        SortProperties config = new SortProperties();
        //
        config.setProperty("task_type", "viewerTag"); //
        config.setProperty("cfg.type", "new"); // Task new, edit
    }
}

```

```
SortProperties printexport = new SortProperties();
//
printexport.setProperty("connecti on. server", "127. 0. 0. 1");
printexport.setProperty("connecti on. port", "8003");
printexport.setProperty("connecti on. reportName",
"/parameter_test. ozt");
printexport.setProperty("appl et. useprogressbar", "fal se");
printexport.setProperty("appl et. al lowmul ti frame", "true");

printexport.setProperty("vi ewer. mode", "pri nt");
printexport.setProperty("pri nt. mode", "si l ent");
printexport.setProperty("pri nt. i gnoreerror", "fal se");

printexport.setProperty("connecti on. fetchtype", "BATCH");
printexport.setProperty("i nformati on. debug", "debug");
printexport.setProperty("appl et. showerrorMessage", "fal se");

//          가
printexport.setProperty("vi ewer. chi ldcount", "1");
printexport.setProperty("chi ld1. connecti on. server", "127. 0. 0. 1");
printexport.setProperty("chi ld1. connecti on. port", "8003");
printexport.setProperty("chi ld1. connecti on. reportName",
"/parameter_test. ozt");
printexport.setProperty("chi ld1. vi ewer. appl et. mode", "pri nt");
printexport.setProperty("chi ld1. pri nt. mode", "si l ent");
printexport.setProperty("chi ld1. pri nt. i gnoreerror", "fal se");
printexport.setProperty("chi ld1. pri nt. pagerange", "range");
printexport.setProperty("chi ld1. pri nt. pages", "2");

//   confi g. setProperty("l aunch_ type", "i mmedi atel y"); //
(   "i mmedi atel y",   "peri odi cal l y")
confi g. setProperty("l aunch_ type", "peri odi cal l y");
confi g. setProperty("start_ year", "2009");
confi g. setProperty("start_ month", "03");
confi g. setProperty("start_ day", "01");
confi g. setProperty("start_ hour", "06");
confi g. setProperty("start_ mi n", "42");
confi g. setProperty("peri odi cal l y_ executi on_ day_ type", "dai l y");
confi g. setProperty("dai l y_ type", "speci fi c_ day");
confi g. setProperty("dai l y_ every_ days", "1");
//   confi g. setProperty("peri odi cal l y_ executi on_ ti me_ type", "once");
//   confi g. setProperty("once_ hour", "09");
//   confi g. setProperty("once_ mi n", "41");

confi g. setProperty("peri odi cal l y_ executi on_ ti me_ type", "repeat");
confi g. setProperty("repeat_ every_ hours", "00");
confi g. setProperty("repeat_ every_ mi nutes", "01");
```

```
config.setProperty("repeat_start_hour", "09");
config.setProperty("repeat_start_minute", "01");
config.setProperty("repeat_end_hour", "23");
config.setProperty("repeat_end_minute", "01");

String taskID = scheduler.createTask(serverInfo, config, printexport);
}
}
```

### Sample : SchedulerViewerTagSample.java

```
package sample;

import oz.framework.api.*;
import oz.scheduler.TaskResult;
import oz.scheduler.ServerInfo;
import oz.scheduler.ScheduledTask;
import oz.util.SortProperties;
import java.util.Vector;
import java.io.*;
import org.apache.log4j.*;
import oz.scheduler.SchedulerException;

public class SchedulerViewerTagSample {

    public static void main(String[] args) {
        // 가 가
        BasicConfigurator.configure();

        Scheduler scheduler = null;
        scheduler = new Scheduler("127.0.0.1", 9521);

        //
        ServerInfo serverInfo = new ServerInfo();
        serverInfo.setID("admin"); //
        serverInfo.setPWD("admin"); //
        serverInfo.setIsDaemon(true);
        serverInfo.setIP("127.0.0.1"); //OZ Server IP . daemon
        serverInfo.setPortNo(8003);

        SortProperties config = new SortProperties();

        //
        config.setProperty("task_type", "viewerTag");
    }
}
```

```
// Task      new, edit
config.setProperty("cfg.type", "new");

//          (      : "immediate",      : "periodical")
config.setProperty("launch_type", "immediate");

SortProperties export = new SortProperties();

//
export.setProperty("connection.server", "127.0.0.1");
export.setProperty("connection.port", "8003");
export.setProperty("connection.reportName", "/parameter_test.ozr");
export.setProperty("applet.mode", "export");
export.setProperty("applet.useprogressbar", "false");
export.setProperty("applet.allowmultiframe", "true");
export.setProperty("connection.pcount", "2");
export.setProperty("connection.args1", "formparam1= ");
export.setProperty("connection.args2", "formparam2= ");
export.setProperty("export.mode", "silent");

//
export.setProperty("export.saveonefile", "true");
export.setProperty("connection.fetchtype", "BATCH");
export.setProperty("export.confirmsave", "false");
export.setProperty("information.debug", "debug");
export.setProperty("applet.showerrorMessage", "false");
export.setProperty("odi.parameter_test.pcount", "2");
export.setProperty("odi.parameter_test.args1", "odi param1= ");
export.setProperty("odi.parameter_test.args2", "odi param2= ");
export.setProperty("odi.odi names", "parameter_test");
export.setProperty("export.format",
    "csv/xls/html/ozd/pdf/txt/tif/doc/ppt/jpg/svg/mht");
export.setProperty("csv.filename", "1.csv");
export.setProperty("excel.filename", "1.xls");
export.setProperty("html.filename", "1.html");
export.setProperty("ozd.filename", "1.ozd");
export.setProperty("pdf.filename", "1.pdf");
export.setProperty("text.filename", "1.txt");
export.setProperty("tif.filename", "1.tif");
export.setProperty("word.filename", "1.doc");
export.setProperty("ppt.filename", "1.ppt");
export.setProperty("jpg.filename", "1.jpg");
export.setProperty("svg.filename", "1.svg");
export.setProperty("mht.filename", "1.mht");
export.setProperty("hdm.filename", "1.hdm");

// 가
export.setProperty("viewer.chilcount", "1");
```

```

export.setProperty("child1.connection.server", "127.0.0.1");
export.setProperty("child1.connection.port", "8003");
export.setProperty("child1.connection.reportName",
                    "/parameter_test.ozr");
export.setProperty("child1.applet.mode", "export");
export.setProperty("child1.applet.useprogressbar", "false");
export.setProperty("child1.applet.allowmultiframe", "true");
export.setProperty("child1.export.mode", "silent");
export.setProperty("child1.connection.fetchtype", "BATCH");
export.setProperty("child1.export.confirmsave", "false");
export.setProperty("child1.information.debug", "debug");
export.setProperty("child1.applet.showerrormessage", "false");

export.setProperty("child1.connection.pcount", "2");
export.setProperty("child1.connection.args1", "formparam1= 1");
export.setProperty("child1.connection.args2", "formparam2= 2");

export.setProperty("child1.odi.odi names", "parameter_test");
export.setProperty("child1.odi.parameter_test.pcount", "2");
export.setProperty("child1.odi.parameter_test.args1", "odi param1= 3");
export.setProperty("child1.odi.parameter_test.args2", "odi param2= 4");

//
export.setProperty("child1.export.format",
                    "csv/xls/html/ozd/pdf/txt/tif/doc/ppt/jpg/svg/mht");
export.setProperty("child1.csv.filename", "child_1.csv");
export.setProperty("child1.excel.filename", "child_1.xls");
export.setProperty("child1.html.filename", "child_1.html");
export.setProperty("child1.ozd.filename", "child_1.ozd");
export.setProperty("child1.pdf.filename", "child_1.pdf");
export.setProperty("child1.text.filename", "child_1.txt");
export.setProperty("child1.tiff.filename", "child_1.tif");
export.setProperty("child1.word.filename", "child_1.doc");
export.setProperty("child1.ppt.filename", "child_1.ppt");
export.setProperty("child1.jpg.filename", "child_1.jpg");
export.setProperty("child1.svg.filename", "child_1.svg");
export.setProperty("child1.mht.filename", "child_1.mht");
export.setProperty("child1.hdm.filename", "child_1.hdm");

//      API      가      EM      가
String taskID = scheduler.createTask(serverInfo, config, export);
}
}

```

**Sample : SchedulerExportOZDsample.java**

```
package sample;

import oz.framework.api.*;
import oz.scheduler.ServerInfo;
import oz.util.SortProperties;
import org.apache.log4j.*;

public class SchedulerExportOZDsample {
    private static String[][] schedelInfoProperties = {
{"task_type", "none"},
        {"cfg.type", "new"}, // Type of Task -> new, edit
{"reportname", "/sample.ozr"},
//      {"category_name", "/"},
//      {"group_name", ""},
//      {"odi_name", "sample"},
//      {"odi_category_name", "/"},
        {"file_export_list", "ozd"},

        {"export.confirmsave", "false"},
        {"TempRepositoryFilePath", "%SCH_HOME%/TempRepository"}, //

        {"RepositoryFileRootPath", "%SCH_HOME%/Repository"},

{"parameter_count", "2"},
        {"parameter_name_1", "[FORM].formparam"},
        {"parameter_value_1", "value1"},
        {"parameter_name_2", "sample.odi param"},
        {"parameter_value_2", "value2"},

        {"image_url_size", "2"},
        {"image_url_1", "file://D:\\\\Repository\\img1.jpg"},
        {"image_url_2", "file://D:\\\\pictures\\img2.jpg"},

//
        {"external_program_check", "check"},
        {"external_program_command", "external.bat"},

        {"reportcount", "2"},
        {"child1.reportname", "/orders.ozr"},
        {"child1.displayname", "orders"},
        {"child1.odi_name", "/orders"},
        {"child1.odi_category_name", "/"},

        {"child1.parameter_count", "2"},
        {"child1.parameter_name_1", "[FORM].formparam"},
```

```
        {"child1.parameter_value_1", "value1"},
        {"child1.parameter_name_2", "sampleparam"},
        {"child1.parameter_value_2", "value2"},

        //
        {"child1.external_program_check", "check"},
        {"child1.external_program_command", "external.bat"},

//
{"child1.option_properties_size", "1"},
        {"child1.option_property_name_1", "use_ozd_parameter"},
        {"child1.option_property_value_1", "true"},

        //
        {"launch_type", "periodically"},
//      (    "immediately",    "periodically")

        {"start_year", "2009"},
        {"start_month", "03"},
        {"start_day", "01"},
        {"start_hour", "06"},
        {"start_min", "42"},
        {"periodically_execution_day_type", "daily"},
        {"daily_type", "specific_day"},
        {"daily_every_days", "1"},
        {"periodically_execution_time_type", "repeat"},
        {"repeat_every_hours", "00"},
        {"repeat_every_minutes", "01"},
        {"repeat_start_hour", "09"},
        {"repeat_start_minute", "01"},
        {"repeat_end_hour", "23"},
        {"repeat_end_minute", "01"},

        {"mail_check", "check"},
        {"mail_notify_error_check", "null"},
        {"mail_recipient_to", "sblee@forcs.com"},
        {"mail_recipient_cc", ""},
        {"mail_recipient_bcc", ""},
        {"mail_subject", "aaa"},
        {"mail_text_message", "message"},
        {"mail_html_comment", "comment"},
        {"html_mail_content", "content"},
        {"mail_attach_list", "ozd"}
};

private static String[][] exportedProperties = {
    {"ozd.filename", "test11.ozd"}, //pdf file name
    {"ozd.mailattach", "true"}, //
```

```

        {"ozd.password", "aaaa"}, //
        {"ozd.memoallowed", "false"}, //
        {"ozd.includeimage", "false"} //
    };

    public static void main(String[] args) {

        BasicConfigurator.configure();
        String IP = "127.0.0.1"; //scheduler server ip
        int PORT = 9521; //scheduler server port
        Scheduler scheduler = null;
        boolean isDaemon = true;
        try {
            scheduler = new Scheduler(IP, PORT);

            // scheduler Server Info
            ServerInfo serverInfo = new ServerInfo();
            if(isDaemon){
                serverInfo.setIP("127.0.0.1"); //OZ Server IP conf. In case use of
daemon
                serverInfo.setPortNo(8003);
                serverInfo.setIsDaemon(true); //OZ Server Type. -> daemon, servlet
            }else{
                serverInfo.setIsDaemon(false);
                serverInfo.setURL("http://127.0.0.1:8080/oz50/server");
            }

            serverInfo.setID("admin"); //id
            serverInfo.setPWD("admin"); //pw

            SortProperties props = new SortProperties();
            setProperties(props, schedulerInfoProperties);

            SortProperties exportMap = new SortProperties();
            setProperties(exportMap, exportedProperties);

            scheduler.createTask(serverInfo, props, exportMap);
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }

    private static void setProperties(SortProperties p, String[][] values) {
        for(int i=0; i<values.length; i++) {
            p.setProperty(values[i][0], values[i][1]);
        }
    }

```

```
}  
}
```

### Sample : SchedulerTaskResult.java

```
package sample;  
import oz.framework.api.*;  
import oz.scheduler.TaskResult;  
import oz.scheduler.ServerInfo;  
import oz.scheduler.ScheduledTask;  
  
public class SchedulerTaskResult {  
  
    public static void main(String[] args)  
    {  
  
        String IP = "127.0.0.1"; //scheduler server ip  
        int PORT = 9521; //scheduler server port  
        Scheduler scheduler = null;  
  
        try {  
            scheduler = new Scheduler(IP, PORT);  
  
            // scheduler Server Info  
            ServerInfo serverInfo = new ServerInfo();  
  
            serverInfo.setIP("127.0.0.1"); //OZ Server IP conf. In case use of  
daemon  
            serverInfo.setPortNo(8003);  
            serverInfo.setIsDaemon(true); //OZ Server Type. -> daemon, servlet  
  
            serverInfo.setID("admin"); //id  
            serverInfo.setPWD("admin"); //pw  
  
            ScheduledTask[] taskList = scheduler.getTask(serverInfo);  
            for (int k = 0; k < taskList.length; k++) {  
                showTask(taskList[k]);  
  
                System.out.println("\n\n");  
  
                TaskResult tr = new TaskResult();  
                tr.taskID = taskList[k].taskID;  
  
                String from = ""; //Task Result Start time  
                String to = ""; //Task Result Endt time
```

```
        boolean isComplete = false;

        TaskResult[] trList = scheduler.getTaskResult(serverInfo,
            from, to, tr.taskID);
        if (trList.length > 0) {
            for (int i = 0; i < trList.length; i++) {
                showTaskResult(trList[i]);
            }
            isComplete = true;
        }
    }
}
catch(Exception e)
{
    e.printStackTrace();
}
}

private static void showTask(ScheduledTask t)
{
    System.out.println("TASK ID : " + t.taskID);
    System.out.println("Report Name : " + t.reportName);
    System.out.println("Type : " + t.schedulingTypeStr);
    System.out.println("Finish Execute Time : " + t.lastRunTimeStr);
    System.out.println("Next Execute Time : " + t.nextRunTimeStr);
    System.out.println("Status : " + t.status);
    System.out.println("");
}

private static void showTaskResult(TaskResult tr)
{
    System.out.println("TASK ID : " + tr.taskID);
    System.out.println("TASK FINISH TIME : " + tr.completedTime);
    System.out.println("Is Succeeded Code : " + tr.isSuccessCode);
    System.out.println("Is Succeeded? : " + tr.isSuccessful);
    System.out.println("Report Name : " + tr.formFileName);
    System.out.println("Parameter : " + tr.paramInfo);
    System.out.println("Type : " + tr.schedulingType);
    System.out.println("Error Message : " + tr.errorMessage);
    System.out.println("");
}
}
```

### Sample : SchedulerTaskModify.java

```
package sample;

import org.apache.log4j.*;
import oz.framework.api.*;
import oz.scheduler.ServerInfo;
import oz.scheduler.ScheduledTask;
import oz.util.SortProperties;

public class SchedulerTaskModify {

    public static void main(String[] args) {
        // 가 가
        BasicConfigurator.configure();

        //
        String IP = "127.0.0.1"; // IP
        int PORT = 9521; // Port

        Scheduler scheduler = null;
        String taskID = null;

        try {
            scheduler = new Scheduler(IP, PORT);

            // OZ Server
            ServerInfo serverInfo = new ServerInfo();

            serverInfo.setIP("127.0.0.1"); // IP
            serverInfo.setPortNo(8003); // Port
            serverInfo.setIsDaemon(true); //

            serverInfo.setID("admin"); // ID
            serverInfo.setPWD("admin"); // Password

            // taskID
            // task task_type viewerTag
            taskID = "parameter_test_061031115221500_263141500.ozs";

            // 가
            SortProperties[] props = scheduler.getTaskProperties(serverInfo,
            taskID);

            // txt Export Task csv export
            props[0].setProperty("user_defined_time", "17:53");
            props[0].setProperty("file_export_list", "csv/txt");
            props[1].setProperty("csv.filename", "modify_task.csv");
        }
    }
}
```

```

//
System.out.println("config.....");
props[0].list(System.out);

System.out.println("export.....");
props[1].list(System.out);

//
ScheduledTask[] taskList = scheduler.getTask(serverInfo);
for (int i = 0; i < taskList.length; i++) {
    if (taskID.endsWith(taskList[i].taskID)) {
        while(true) {
            if (taskList[i].status != 'R') { //
                //
                scheduler.modifyTask(serverInfo, taskID, props[0],
props[1]);

                System.out.println("modify ok...");
                break;
            }
            Thread.sleep(1000);
        }
    }
} catch(Excepti on e) {
    e.printStackTrace();
}
}
}
}

```

### Sample : SchedulerPrint.java

```

package sample;

import org.apache.log4j.BasicConfigurator;
import oz.framework.api.Scheduler;
import oz.scheduler.SchedulerException;
import oz.scheduler.ServerInfo;
import oz.util.SortProperties;

public class SchedulerPrint {

    public static void main(String[] args) {

```

```
//          가          가
BasicConfigurator.configure();

//
String IP = "127.0.0.1"; //          IP
int PORT = 9521; //          Port

try {
    Scheduler scheduler = new Scheduler(IP, PORT);

    // OZ Server
    ServerInfo serverInfo = new ServerInfo();
    serverInfo.setIP("127.0.0.1"); //          IP
    serverInfo.setPortNo(8003); //          Port
    serverInfo.setIsDaemon(true); //
    serverInfo.setID("admin"); //          ID
    serverInfo.setPWD("admin"); //          Password

    SortProperties config = new SortProperties();

    // print          task_type   viewerTag
    config.setProperty("task_type", "viewerTag");
    config.setProperty("launch_type", "immediately");

    SortProperties printMap = new SortProperties();
    printMap.setProperty("connection.server", "127.0.0.1");
    printMap.setProperty("connection.port", "8003");
    printMap.setProperty("connection.reportName",
"/parameter_test.ozr");
    printMap.setProperty("repository_agent.try_license_check", "true");
    printMap.setProperty("connection.pcount", "2");
    printMap.setProperty("connection.args1", "formparam1= ");
    printMap.setProperty("connection.args2", "formparam2= ");
    printMap.setProperty("connection.fetchtype", "BATCH");
    printMap.setProperty("odi.parameter_test.pcount", "2");
    printMap.setProperty("odi.parameter_test.args1", "odi param1= ");
    printMap.setProperty("odi.parameter_test.args2", "odi param2= ");
    printMap.setProperty("odi.odi names", "parameter_test");

    //print          (          )
    printMap.setProperty("print.adjust", "true");
    printMap.setProperty("print.alldocument", "false");
    printMap.setProperty("print.close", "true");
    printMap.setProperty("print.collate", "true");
    printMap.setProperty("print.copies", "1");
    printMap.setProperty("print.duplex", "none");
    printMap.setProperty("print.gray", "true");
    printMap.setProperty("print.ignoreerror", "true");
```

```

printMap.setProperty("print.lockopt", "true");
printMap.setProperty("print.once", "false");
printMap.setProperty("print.orientation", "default");
printMap.setProperty("print.pageorient", "horizontal");
printMap.setProperty("print.pagerange", "all");
printMap.setProperty("print.printbypage", "true");
printMap.setProperty("print.printername", "default");
printMap.setProperty("print.scaling", "100");
printMap.setProperty("print.size", "A4");
printMap.setProperty("print.spool pages", "100");
printMap.setProperty("print.style", "normal");
printMap.setProperty("print.usedefaultpaper", "false");
printMap.setProperty("print.spool pages", "100");
printMap.setProperty("print.usedi al oopt", "true");

//
printMap.setProperty("viewer.childcount", "1");
printMap.setProperty("child1.connection.server", "127.0.0.1");
printMap.setProperty("child1.connection.port", "8003");
printMap.setProperty("child1.connection.reportName",
"/module_sample.ozr");
printMap.setProperty("child1.connection.fetchtype", "BATCH");

//          API          가          EM          가          .
boolean result = scheduler.print(serverInfo, config, printMap);
if(result == true) {
    System.out.println("Success Print");
} else {
    System.out.println("Failed Print");
}
} catch (SchedulerException se) {
    se.printStackTrace();
}
}
}

```

### Sample : SchedulerTaskConverter.java

```

package sample;
import org.apache.log4j.BasicConfigurator;
import oz.framework.api.Scheduler;

public class SchedulerTaskConverter {
    public static void main(String[] args) {

```

```
BasicConfigurator.configure();
String IP = "127.0.0.1"; //scheduler server ip
int PORT = 9521; //scheduler server port

Scheduler scheduler = null;

try {
    scheduler = new Scheduler(IP, PORT);

    // /%SCH_HOME%/ScheduledTask          ozs          가
(   OZS          가   )
    String[] ozsList = scheduler.getOZSList("/ScheduledTask");

    for(int i=0 ; i < ozsList.length ; i++) {
        System.out.println(ozsList[i]);
    }

    // /%SCH_HOME%/ScheduledTask          ozs
/%SCH_HOME%/ConvertedTask
    scheduler.ozsVerConv("/ScheduledTask", "/ConvertTask");

    // /%SCH_HOME%/ ConvertTask          ozs          가
(   OZS          가   )
    ozsList = scheduler.getOZSList("/ConvertTask");
    for(int i=0 ; i < ozsList.length ; i++) {
        System.out.println(ozsList[i]);
    }
}
catch(Excepti on e) {
    e.printStackTrace();
}
}
```

## UseSCOzdParameterSample.java

```
package sample;

import oz.framework.api.*;
import oz.scheduler.TaskResult;
import oz.scheduler.ServerInfo;
import oz.scheduler.ScheduledTask;
import oz.util.SortProperties;
import java.util.Vector;
import java.io.*;
```

```
import org.apache.log4j.*;

public class UseSCOzdParameterSample {
    private static String[][] scheduleInfoProperties =
    {
        // Basic Configuration
        {"launch_type", "Immediately"}, // Task Execute Type

        {"cfg.type", "new"}, // Type of Task -> new, edit
        {"report_name", "parameter_test.ozr"},
        {"category_name", "/"},
        {"group_name", ""},

        // ODI
        {"odi_name", "parameter_test.odi"},
        {"odi_category_name", "/"},

        {"option_properties_size", "1"},
        {"option_property_name_1", "use_ozd_parameter"},
        {"option_property_value_1", "true"},

        {"parameter_count", "4"},

        {"parameter_name_1", "[FORM].formparam1"},
        {"parameter_value_1", "1"},
        {"parameter_name_2", "[FORM].formparam2"},
        {"parameter_value_2", "2"},
        {"parameter_name_3", "parameter_test.odi param1"},
        {"parameter_value_3", "3"},
        {"parameter_name_4", "parameter_test.odi param2"},
        {"parameter_value_4", "4"},
        {"file_export_list", "ozd"}
    };

    private static String[][] exportedProperties =
    {
        {"ozd.filename", "newimage.ozd"}, //OZD file name
    };

    public static void main(String[] args)
```

```
{

    BasicConfigurator.configure();
    String IP = "127.0.0.1"; //scheduler server ip
    int PORT = 9521; //scheduler server port
    String[][] values = null;
    Scheduler scheduler = null;
    String taskID = null;

    try {
        scheduler = new Scheduler(IP, PORT);
        for (int i = 0; i < 1; i++) {
            // scheduler Server Info
            ServerInfo serverInfo = new ServerInfo();

            serverInfo.setIP("127.0.0.1"); //OZ Server IP conf. In
case use of daemon
            serverInfo.setPortNo(8003);
            serverInfo.setIsDaemon(true); //OZ Server Type. -> daemon,
servlet

            serverInfo.setID("admin"); //id
            serverInfo.setPWD("admin"); //pw

            SortProperties p = new SortProperties();
            setProperties(p, schedulerInfoProperties);

            SortProperties exportMap = new SortProperties();
            setProperties(exportMap, exportedProperties);

            taskID = scheduler.createTask(serverInfo, p, exportMap);

            System.out.println("i i==" + i + "      taskid==" + taskID);
        }
    }
    catch (Exception e) {
        e.printStackTrace();
    }
}

private static void setProperties(SortProperties p, String[][]
values)
{
```

```
for(int i=0; i<values.length; i++)
{
    p.setProperty(values[i][0], values[i][1]);
}
}
```

### Sample : DirectExportResultSample.java

```
package sample;

import oz.framework.api.*;
import oz.scheduler.DirectExportResult;
import oz.scheduler.ServerInfo;
import oz.util.SortProperties;

public class DirectExportResultSample {
    public DirectExportResultSample() {
    }

    static final String IP = "127.0.0.1";
    static final int PORT = 8003;
    static final String URL = "http://localhost:8088/oz/server";
    static final boolean isDaemon = true;

    static final String ID = "admin";
    static final String PWD = "admin";

    static final String schedulerIP = "127.0.0.1";
    static final int schedulerPORT = 9521;

    public static void main(String[] args) {

        Scheduler scheduler = null;
        try {

            scheduler = new Scheduler(schedulerIP, schedulerPORT);

            //
            ServerInfo serverInfo = new ServerInfo();
            serverInfo.setID(ID); //
            serverInfo.setPWD(PWD); //
            serverInfo.setIsDaemon(isDaemon);

            if (serverInfo.getIsDaemon()) {
```

```
serverInfo.setIP(IP); //OZ Server IP      . daemon
serverInfo.setPortNo(PORT);
}
else {
    serverInfo.setURL(URL);
}
SortProperties props = new SortProperties();
props.setProperty("task_type", "viewerTag");
props.setProperty("cfg.type", "new");
props.setProperty("launch_type", "immediate");

SortProperties export = new SortProperties();

export.setProperty("connection.server", "127.0.0.1");
export.setProperty("connection.port", "8003");
export.setProperty("connection.reportName", "/parameter_test.ozr");
export.setProperty("viewer.mode", "export");
export.setProperty("export.mode", "silent");
export.setProperty("viewer.isframe", "true");
export.setProperty("viewer.mode", "silent");
export.setProperty("export.confirmsave", "false");
export.setProperty("export.format", "xls");
export.setProperty("xls.filename", "sample.xls");

DirectExportResult result = scheduler.directExport(serverInfo,
props, export);
showDirectExportResult(result);

}
catch (Exception e) {
    e.printStackTrace();
}
}

private static void showDirectExportResult(DirectExportResult t) {
    System.out.println("          : " + t.taskID);
    System.out.println("          : " + t.completedTime);
    System.out.println("          : " + t.executeTime + "sec");
    System.out.println("          : " + t.isSuccess);
    System.out.println("          : " + t.formName);
    System.out.println("export          : " + t.exportFileList);
System.out.println("          : " + t.pageCount);

    System.out.println("");
}
}
```

## Sample : DirectPrintResultSample.java

```
package sample;

import oz.framework.api.*;
import oz.scheduler.DirectPrintResult;
import oz.scheduler.ServerInfo;
import oz.util.SortProperties;

public class DirectPrintResultSample {
    public DirectPrintResultSample() {
    }

    static final String IP = "127.0.0.1";
    static final int PORT = 8003;
    static final String URL = "http://localhost:8088/oz35/server";
    static final boolean isDaemon = true;

    static final String ID = "admin";
    static final String PWD = "admin";

    static final String schedulerIP = "127.0.0.1";
    static final int schedulerPORT = 9521;

    public static void main(String[] args) {

        Scheduler scheduler = null;
        try {

            scheduler = new Scheduler(schedulerIP, schedulerPORT);

            //
            ServerInfo serverInfo = new ServerInfo();
            serverInfo.setID(ID); //
            serverInfo.setPWD(PWD); //
            serverInfo.setIsDaemon(isDaemon);

            if (serverInfo.getIsDaemon()) {
                serverInfo.setIP(IP); //OZ Server IP . daemon
                serverInfo.setPortNo(PORT);
            }
            else {
                serverInfo.setURL(URL);
            }
            SortProperties props = new SortProperties();
            props.setProperty("task_type", "viewerTag");
            props.setProperty("cfg.type", "new");
        }
    }
}
```

```

props.setProperty("launch_type", "immediate");

SortProperties print = new SortProperties();

print.setProperty("connection.server", "127.0.0.1");
print.setProperty("connection.port", "8003");
print.setProperty("connection.id", "admin");
print.setProperty("connection.password", "admin");

print.setProperty("connection.reportname", "/parameter_test.ozr");
print.setProperty("viewer.configmode", "html");
print.setProperty("odi.odinames", "parameter_test");
print.setProperty("connection.pcount", "2");
print.setProperty("connection.args1", "formparam2=");
print.setProperty("connection.args2", "formparam1=");
print.setProperty("odi.parameter_test.pcount", "2");

print.setProperty("odi.parameter_test.args1", "odi param1=odi Value1111111");

print.setProperty("odi.parameter_test.args2", "odi param2=odi Value2222222");
//      print.setProperty("application.isframe", "false");
print.setProperty("connection.clientdmttype", "Memory");
print.setProperty("connection.serverdmttype", "Memory");
print.setProperty("connection.fetchtype", "Concurrent");
print.setProperty("viewer.mode", "print");
print.setProperty("print.mode", "silent");
print.setProperty("viewer.printcommand", "true");

DirectPrintResult result = scheduler.directPrint(serverInfo, props,
print);
showDirectExportResult(result);

}
catch (Exception e) {
    e.printStackTrace();
}
}

private static void showDirectExportResult(DirectPrintResult t) {
    System.out.println("          : " + t.taskID);
    System.out.println("          : " + t.completedTime);
    System.out.println("          : " + t.executeTime + "sec");
    System.out.println("          : " + t.isSuccess);
    System.out.println("          : " + t.formName);

    System.out.println("          : " + t.pageCount);
    System.out.println("          : " + t.pageCopy);

```

```
System.out.println("      : " + t.pageRange);
System.out.println("      : " + t.printerName);
System.out.println("      : " + t.printerDriverName);

System.out.println("");
}
}
```



**Definition** `가` .

■ **setBaseDay**

**Prototype** `public void setbaseDay(String pattern)`

**Definition** `가` .

**Argument** *pattern*

- yyyy-MM-dd :
- yyyy-MM-01 : 1
- yyyy-01-01 : 1 1
- 2007-01-01 : 2007 1 1

■ **getBaseDay**

**Prototype** `public String getBaseDay()`

**Definition** `가` .

■ **setType**

**Prototype** `public void setType(String type)`

**Definition** `가` .

**Argument** *type*

- solar :
- lunar :

`setType("solar, lunar")` : solar, lunar solar  
`setType("solar, lunar")` : (solar, lunar) 가 )

■ **getType**

**Prototype** `public String getType()`

**Definition** `( , ) 가` .

■ **isLunar**

**Prototype** `public boolean isLunar()`

**Definition** `가` .

■ **setStart**

**Prototype** `public void setStart(int start)`

**Definition** "baseday"

**Argument** *start*

■ **getStart**

**Prototype** public int getStart()

**Definition** "baseday"

가

■ **setEnd**

**Prototype** public void setEnd(int end)

**Definition** "baseday"

**Argument** *end*

■ **getEnd**

**Prototype** public void getEnd()

**Definition** "baseday"

가

**Sample : SchedulerTaskHoliday.java**

```

package sample;

import java.util.*;

import oz.util.OZMap;
import oz.framework.api.*;
import oz.scheduler.holiday.*;

public class SchedulerTaskHoliday {

    public static void main(String[] args)
    {

        String IP = "127.0.0.1"; //scheduler server ip
        int PORT = 9521; //scheduler server port
        Scheduler scheduler = null;

        try {
            scheduler = new Scheduler(IP, PORT);

            //TaskHolidayInfo 가
    
```

```

OZMap infoMap = scheduler.getTaskHolidayInfoList();

Iterator set = infoMap.keys();
while(set.hasNext()) {
    String key = (String)set.next();
    TaskHolidayInfo info = (TaskHolidayInfo)infoMap.get(key);
    System.out.println(info.toString());
}

//TaskHolidayInfo 가
TaskHolidayInfo info = new TaskHolidayInfo(" ");
info.setBaseDay("yyyy-07-01");
info.setType("sol ar");
info.setStart(0);
info.setEnd(0);
scheduler.addTaskHolidayInfo(info);

//TaskHolidayInfo
info.setBaseDay("yyyy-07-02");
scheduler.modifyTaskHolidayInfo(" ", info);

//TaskHolidayInfo
scheduler.deleteTaskHolidayInfo(" ");

// TaskHoliday
//(
scheduler.saveTaskHoliday();
}
catch(Excepti on e)
{
    e.printStackTrace();
}
}
}

```

```

:
Scheduler Info Properties {"task_holiday", " ";
    " } 가 . (ex : {"task_holiday",
" ;Group1"})

```





```
import oz.util.OZMap;
import oz.framework.api.*;
import oz.scheduler.holiday.*;

public class SchedulerTaskHolidayGroup {

    public static void main(String[] args)
    {

        String IP = "127.0.0.1"; //scheduler server ip
        int PORT = 9521; //scheduler server port
        Scheduler scheduler = null;

        try {
            scheduler = new Scheduler(IP, PORT);

            //TaskHolidayInfoGroup 가 .
            OZMap<gl nfoMap> = scheduler.getTaskHolidayGroupInfoList();
            Iterator set1 = gl nfoMap.keys();
            while(set1.hasNext()) {
                String key = (String)set1.next();
                TaskHolidayGroupInfo gl nfo =
                (TaskHolidayGroupInfo)gl nfoMap.get(key);
                System.out.println(gl nfo.toString());
            }

            //TaskHolidayGroupInfo 가 .
            TaskHolidayGroupInfo gl nfo = new TaskHolidayGroupInfo("group1");
            gl nfo.addReference(" ");
            gl nfo.addReference(" ");
            gl nfo.addReference("31 ");
            scheduler.addTaskHolidayGroupInfo(gl nfo);

            //TaskHolidayGroupInfo .
            gl nfo.addReference(" ");
            scheduler.modifyTaskHolidayGroupInfo("group1", gl nfo);
            //TaskHolidayGroupInfo .
            scheduler.deleteTaskHolidayGroupInfo("group1");

            // TaskHoliday
            //( .)
            scheduler.saveTaskHoliday();

        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

```
    }  
  }  
}
```

:  
Scheduler Info Properties {"task\_holiday", "  
"} 가 . (ex : {"task\_holiday",  
" ;Group1"})



- Interface Repository
- Class RepositoryEx
-

## Interface Repository

### Interface Summary

// `IOZRepository` : `oz.framework.repositoryex`

- - `public interface IOZRepository`
  
- - `public interface IOZRepositoryCategory`
  - `public interface IOZRepositoryItem`
  - `public interface IOZRepositoryItemHistory`
  
- - `public interface IOZRepositoryGroup`
  - `public interface IOZRepositoryUser`
  - `public interface IOZRepositoryMultiLoginUser`
  
- **Exception**
  - `public class OZRepositoryException extends Exception`

// `IOZRepositoryAuth` : `oz.framework.repositoryex.auth`

- - `public interface IOZRepositoryAuthGroupCategory`
  - `public interface IOZRepositoryAuthGroupItem`
  - `public interface IOZRepositoryAuthUserCategory`
  - `public interface IOZRepositoryAuthUserItem`

// `IOZRepositoryInfo` : `oz.framework.repositoryex.info`

- `IOZRepositoryInfo` (bean)

- public interface IOZCategoryInfo
  - public interface IOZItemInfo
  - public interface IOZHistoryInfo
  - public interface IOZGroupInfo
  - public interface IOZUserInfo
  - public interface IOZLoginInfo
- (bean)
    - public interface IOZCategoryInfoList
    - public interface IOZItemInfoList
    - public interface IOZHistoryInfoList
    - public interface IOZGroupInfoList
    - public interface IOZUserInfoList

## Interface Method Summary

### // IOZRepository

- void setProperty(SortProperties prop) throws OZRepositoryException
- int getRepositoryVersion()
- void openRepository() throws OZRepositoryException
- void closeRepository() throws OZRepositoryException
- Object checkRepositoryLogin(String userName, String password, IOZLoginInfo loginInfo, HttpServletRequest request, HttpServletRequest servlet) throws OZRepositoryException
- void updateSessionState(Object sessionID, int type) throws OZRepositoryException
- boolean repositoryLogout(Object sessionID, String userName, IOZLoginInfo loginInfo, HttpServletRequest request, HttpServletRequest servlet) throws OZRepositoryException
- IOZRepositoryCategory getRepositoryCategory() throws OZRepositoryException
- IOZRepositoryItem getRepositoryItem() throws OZRepositoryException
- IOZRepositoryItemHistory getRepositoryItemHistory() throws OZRepositoryException

- IOZRepositoryGroup getRepositoryGroup() throws OZRepositoryException
- IOZRepositoryUser getRepositoryUser() throws OZRepositoryException
- IOZRepositoryMultiLoginUser getRepositoryMultiLoginUser() throws OZRepositoryException
- IOZRepositoryAuthGroupCategory getRepositoryAuthGroupCategory() throws OZRepositoryException
- IOZRepositoryAuthGroupItem getRepositoryAuthGroupItem() throws OZRepositoryException
- IOZRepositoryAuthUserCategory getRepositoryAuthUserCategory() throws OZRepositoryException
- IOZRepositoryAuthUserItem getRepositoryAuthUserItem() throws OZRepositoryException

// IOZRepositoryCategory

- int getAccessTypeCategory()
- IOZRepository getOZRepository() throws OZRepositoryException
- String[] createItemInCategory(Object sessionId, String[] itemNames, String[] desc, String[] categoryIDs, InputStream[] item\_ins, String comment, int[] errorCode, String[] errorMsg) throws OZRepositoryException
- String[] createCategory(Object sessionId, String[] categoryNames, String[] pCategoryIDs, String comment, int[] errorCode, String[] errorMsg) throws OZRepositoryException
- String modifyCategoryName(Object sessionId, String categoryID, String new\_CategoryName, String comment) throws OZRepositoryException
- boolean[] deleteCategory(Object sessionId, String[] categoryIDs, boolean[] isDestroys, String comment, int[] errorCode, String[] errorMsg) throws OZRepositoryException
- boolean[] undeleteCategory(Object sessionId, String[] categoryIDs, String comment, int[] errorCode, String[] errorMsg) throws OZRepositoryException
- boolean hasTheItemInCategory(Object sessionId, String itemID) throws OZRepositoryException
- int getItemCountInCategory(Object sessionId, String categoryID) throws OZRepositoryException

- IOZItemInfoList getItemListInCategory(Object sessionID, String categoryID) throws OZRepositoryException
- IOZCategoryInfoList getCategoryListInCategory(Object sessionID, String categoryID) throws OZRepositoryException
- String getCategoryIDofItem(Object sessionID, String itemID) throws OZRepositoryException
- IOZCategoryInfo getCategoryInfo(Object sessionID, String categoryID) throws OZRepositoryException
- IOZItemInfoList getDeletedItemListInCategory(Object sessionID, String categoryID) throws OZRepositoryException
- boolean transferItem(Object sessionID, String[] itemIDs, String target\_CategoryID) throws OZRepositoryException
- boolean transferCategory(Object sessionID, String categoryID, String target\_CategoryID) throws OZRepositoryException

#### // IOZRepositoryItem

- public int getAccessTypeItem()
- public IOZRepository getOZRepository() throws OZRepositoryException
- public String[] createItem(Object sessionID, String[] itemNames, String[] desc, InputStream[] item\_ins, String comment, int[] errorCode, String[] errorMsg) throws
- public String modifyItemName(Object sessionID, String itemID, String new\_itemName, String comment) throws OZRepositoryException
- public boolean[] deleteItem(Object sessionID, String[] itemIDs, boolean[] isDestroys, String comment, int[] errorCode, String[] errorMsg) throws
- public boolean[] undeleteItem(Object sessionID, String[] itemIDs, String comment, int[] errorCode, String[] errorMsg) throws OZRepositoryException
- public boolean modifyItemDesc(Object sessionID, String itemID, String desc) throws OZRepositoryException
- public IOZItemInfo getItemInfo(Object sessionID, String itemID) throws OZRepositoryException
- public boolean hasTheItem(Object sessionID, String itemID) throws OZRepositoryException
- public InputStream[] getItems(Object sessionID, String itemIDs[], int[]

`errorCode, String[] errorMsg)` throws `OZRepositoryException`

- `public InputStream[] getItems(Object sessionId, String itemIDs[], long[] modifiedTimes, int[] errorCode, String[] errorMsg)` throws `OZRepositoryException`
- `public InputStream[] checkoutItem(Object sessionId, String itemIDs[], String[] localCheckoutFolders, long[] localFileTimes, int[] errorCode, String[] errorMsg)` throws
- `public boolean[] checkinItem(Object sessionId, String[] itemIDs, InputStream[] item_ins, String comment, boolean[] keepCheckout, int[] errorCode, String[] errorMsg)` throws
- `public InputStream[] undoCheckoutItem(Object sessionId, String[] itemIDs, boolean[] isReplaces, int[] errorCode, String[] errorMsg)` throws `OZRepositoryException`
- `public boolean[] isCheckedOutUser(Object sessionId, String[] itemIDs)` throws `OZRepositoryException`

#### // IOZRepositoryItemHistory

- `public int getAccessTypeItemHistory()`
- `public IOZRepository getOZRepository()` throws `OZRepositoryException`
- `public boolean rollbackItem(Object sessionId, String itemID, int itemVersion, String comment)` throws `OZRepositoryException`
- `public InputStream getItemByVersion(Object sessionId, String itemID, int version)` throws `OZRepositoryException`
- `public IOZHistoryInfoList getHistoryItemList(Object sessionId, String itemID)` throws `OZRepositoryException`
- `public IOZHistoryInfoList getDeleteHistoryItemInfo(Object sessionId, String itemID)` throws `OZRepositoryException`
- `public boolean removeHistoryItem(Object sessionId, String itemID, int itemVersion)` throws `OZRepositoryException`

#### // IOZRepositoryGroup

- `public int getAccessTypeGroup()`
- `public IOZRepository getOZRepository()` throws `OZRepositoryException`
- `public String createGroup(Object sessionId, String gName, String pGroupID, String desc)` throws `OZRepositoryException`

- `public String modifyGroupName(Object sessionID, String groupID, String gName) throws OZRepositoryException`
- `public boolean modifyGroupDesc(Object sessionID, String groupID, String desc) throws OZRepositoryException`
- `public boolean deleteGroup(Object sessionID, String groupID) throws OZRepositoryException`
- `public String createUserInGroup(Object sessionID, String userName, String password, String groupID, String desc) throws OZRepositoryException`
- `public IOZUserInfoList getUserInfoListInGroup(Object sessionID, String groupID) throws OZRepositoryException`
- `public IOZGroupInfo getGroupInfo(Object sessionID, String groupID) throws OZRepositoryException`
- `public IOZGroupInfoList getSubGroupInfoList(Object sessionID, String groupID) throws OZRepositoryException`
- `public IOZGroupInfo getParentGroupInfo(Object sessionID, String groupID) throws OZRepositoryException`
- `public IOZGroupInfo getGroupIdOfUser(Object sessionID, String userID) throws OZRepositoryException`
- `public boolean addGroupAdmin(Object sessionID, String userID, String groupID) throws OZRepositoryException`
- `public boolean removeGroupAdmin(Object sessionID, String userID, String groupID) throws OZRepositoryException`
- `public boolean isUserGroupAdmin(Object sessionID, String userID, String groupID) throws OZRepositoryException`
- `public IOZUserInfoList getUserAdminListInGroup(Object sessionID, String groupID) throws OZRepositoryException`
- `public boolean transferUser(Object sessionID, String userID, String target_GroupID) throws OZRepositoryException`
- `public boolean transferGroup(Object sessionID, String groupID, String target_GroupID) throws OZRepositoryException`

**// IOZRepositoryUser**

- `public int getAccessTypeUser()`
- `public IOZRepository getOZRepository() throws OZRepositoryException`
- `public String createUser(Object sessionID, String userName, String`

password, String desc) throws OZRepositoryException

- **public String modifyUserName(Object sessionID, String userID, String userName) throws OZRepositoryException**
- **public boolean modifyUserPassword(Object sessionID, String userID, String old\_password, String new\_password) throws OZRepositoryException**
- **public boolean modifyUserDesc(Object sessionID, String userID, String desc) throws OZRepositoryException**
- **public boolean deleteUser(Object sessionID, String userID) throws OZRepositoryException**
- **public IOZUserInfo getUserInfo(Object sessionID, String userID) throws OZRepositoryException**
- **public boolean checkUserPassword(Object sessionID, String userID, String password) throws OZRepositoryException**
- **public IOZUserInfoList getUserInfoList(Object sessionID) throws OZRepositoryException**
- **public boolean isCheckAdmin(Object sessionID, String userID) throws OZRepositoryException**

**// IOZRepositoryMultiLoginUser**

- **public int getAccessTypeUser()**
- **public IOZRepository getOZRepository() throws OZRepositoryException**
- **public String createUser(Object sessionID, String userName, String password, String desc) throws OZRepositoryException**
- **public String modifyUserName(Object sessionID, String userID, String userName) throws OZRepositoryException**
- **public boolean modifyUserPassword(Object sessionID, String userID, String old\_password, String new\_password) throws OZRepositoryException**
- **public boolean modifyUserDesc(Object sessionID, String userID, String desc) throws OZRepositoryException**
- **public boolean deleteUser(Object sessionID, String userID) throws OZRepositoryException**
- **public IOZUserInfo getUserInfo(Object sessionID, String userID) throws OZRepositoryException**

- **public boolean checkUserPassword(Object sessionID, String userID, String password) throws OZRepositoryException**
- **public IOZUserInfoList getUserInfoList(Object sessionID) throws OZRepositoryException**
- **public boolean isCheckAdmin(Object sessionID, String userID) throws OZRepositoryException**

#### // IOZRepositoryAuthGroupCategory

- **public int getAccessTypeAuthGroupCategory()**
- **public IOZRepository getOZRepository() throws OZRepositoryException**
- **public boolean modifyCategoryGroupAuth(Object sessionID, String categoryID, String groupID, int perm) throws OZRepositoryException**
- **public int getGroupAuthToCategory(Object sessionID, String categoryID, String groupID) throws OZRepositoryException**
- **public IOZGroupInfoList getGroupListAuthToCategory(Object sessionID, String categoryID, int perm) throws OZRepositoryException**
- **public IOZItemInfoList getItemListAuthToGroupInCategory(Object sessionID, String groupID, String categoryID, int perm) throws OZRepositoryException**
- **public IOZCategoryInfoList getCategoryListAuthToGroupInCategory(Object sessionID, String groupID, String categoryID, int perm) throws OZRepositoryException**

#### // IOZRepositoryAuthGroupItem

- **public int getAccessTypeAuthGroupItem()**
- **public IOZRepository getOZRepository() throws OZRepositoryException**
- **public boolean modifyItemGroupAuth(Object sessionID, String groupID, String itemID, int perm) throws OZRepositoryException**
- **public int getGroupAuthToItem(Object sessionID, String groupID, String itemID) throws OZRepositoryException**
- **public IOZGroupInfoList getGroupListAuthToItem(Object sessionID, String itemID, int perm) throws OZRepositoryException**
- **public IOZItemInfoList getItemListAuthToGroup(Object sessionID, String groupID, int perm) throws OZRepositoryException**

// IOZRepositoryAuthUserCategory

- public int getAccessTypeAuthUserCategory()
- public IOZRepository getOZRepository() throws OZRepositoryException
- public boolean modifyCategoryUserAuth(Object sessionID, String userID, String categoryID, int perm) throws OZRepositoryException
- public int getUserAuthToCategory(Object sessionID, String userID, String categoryID) throws OZRepositoryException
- public IOZUserInfoList getUserListAuthToCategory(Object sessionID, String categoryID, int perm) throws OZRepositoryException
- public IOZItemInfoList getItemListAuthToUserInCategory(Object sessionID, String userID, String categoryID, int perm) throws OZRepositoryException
- public IOZCategoryInfoList getCategoryListAuthToUserInCategory(Object sessionID, String userID, String categoryID, int perm) throws OZRepositoryException

// IOZRepositoryAuthUserItem

- public int getAccessTypeAuthUserItem()
- public IOZRepository getOZRepository() throws OZRepositoryException
- public boolean modifyItemUserAuth(Object sessionID, String userID, String itemID, int perm) throws OZRepositoryException
- public int getUserAuthToItem(Object sessionID, String userID, String itemID) throws OZRepositoryException
- public IOZUserInfoList getUserListAuthToItem(Object sessionID, String itemID, int perm) throws OZRepositoryException
- public IOZItemInfoList getItemListAuthToUser(Object sessionID, String userID, int perm) throws OZRepositoryException

## Interface Method Detail

// IOZRepository Interface Method Summary

- **setProperty**

---

Prototype	public void setProperty(SortProperties prop) throws OZRepositoryException
-----------	---

---

Definition	repository.properties Key
Argument	<i>prop</i>

■ **getRepositoryVersion**

Prototype	public int getRepositoryVersion()
Definition	가

■ **openRepository**

Prototype	public void openRepository() throws OZRepositoryException
Definition	

■ **closeRepository**

Prototype	public void closeRepository() throws OZRepositoryException
Definition	

■ **checkRepositoryLogin**

Prototype	public Object checkRepositoryLogin(String userName, String password, IOZLoginInfo loginInfo, HttpServletRequest request, HttpServletResponse servlet) throws OZRepositoryException												
Definition	가												
Argument	<table border="0"> <tr> <td><i>userName</i></td> <td></td> </tr> <tr> <td><i>password</i></td> <td></td> </tr> <tr> <td><i>isCallServer</i></td> <td></td> </tr> <tr> <td><i>loginInfo</i></td> <td>: IOZLoginInfo</td> </tr> <tr> <td><i>request</i></td> <td>HTTP Servlet Request</td> </tr> <tr> <td><i>servlet</i></td> <td>HTTP Servlet</td> </tr> </table>	<i>userName</i>		<i>password</i>		<i>isCallServer</i>		<i>loginInfo</i>	: IOZLoginInfo	<i>request</i>	HTTP Servlet Request	<i>servlet</i>	HTTP Servlet
<i>userName</i>													
<i>password</i>													
<i>isCallServer</i>													
<i>loginInfo</i>	: IOZLoginInfo												
<i>request</i>	HTTP Servlet Request												
<i>servlet</i>	HTTP Servlet												

■ **updateSessionState**

Prototype	public void updateSessionState(Object sessionId, int type) throws OZRepositoryException
-----------	---

Definition	가	
	<i>sessionId</i>	ID
Argument	<i>type</i>	<ul style="list-style-type: none"> <li>• 0 : DEFAULT</li> <li>• 1 : BARE_SERVER</li> <li>• 2 : FROM_SERVER</li> </ul>

■ repositoryLogout

Prototype	public boolean repositoryLogout(Object sessionId, String userName, IOZLogInInfo logInInfo, HttpServletRequest request, HttpServletRequest servlet) throws OZRepositoryException	
Definition	ID	가
	<i>sessionId</i>	ID
	<i>userName</i>	
Argument	<i>logInInfo</i>	( )
	<i>request</i>	HTTP Servlet Request
	<i>servlet</i>	HTTP Servlet

■ getRepositoryCategory

Prototype	public IOZRepositoryCategory getRepositoryCategory() throws OZRepositoryException	
Definition	가	

■ getRepositoryItem

Prototype	public IOZRepositoryItem getRepositoryItem() throws OZRepositoryException	
Definition	가	

■ getRepositoryItemHistory

Prototype	public IOZRepositoryItemHistory getRepositoryItemHistory() throws OZRepositoryException	
Definition	가	

■ getRepositoryGroup

<b>Prototype</b>	<code>public IOZRepositoryGroup getRepositoryGroup() throws OZRepositoryException</code>
<b>Definition</b>	가 .

■ **getRepositoryUser**

<b>Prototype</b>	<code>public IOZRepositoryUser getRepositoryUser() throws OZRepositoryException</code>
<b>Definition</b>	가 .

■ **getRepositoryMultiLoginUser**

<b>Prototype</b>	<code>public IOZRepositoryMultiLoginUser getRepositoryMultiLoginUser() throws OZRepositoryException</code>
<b>Definition</b>	가 .

■ **getRepositoryAuthGroupCategory**

<b>Prototype</b>	<code>public IOZRepositoryAuthGroupCategory getRepositoryAuthGroupCategory() throws OZRepositoryException</code>
<b>Definition</b>	가 .

■ **getRepositoryAuthGroupItem**

<b>Prototype</b>	<code>public IOZRepositoryAuthGroupItem getRepositoryAuthGroupItem() throws OZRepositoryException</code>
<b>Definition</b>	가 .

■ **getRepositoryAuthUserCategory**

<b>Prototype</b>	<code>public IOZRepositoryAuthUserCategory getRepositoryAuthUserCategory() throws OZRepositoryException</code>
<b>Definition</b>	가 .

■ **getRepositoryAuthUserItem**

<b>Prototype</b>	<code>public IOZRepositoryAuthUserItem getRepositoryAuthUserItem() throws OZRepositoryException</code>
<b>Definition</b>	가 .

- Key

setRepositoryConfig()    getRepositoryConfig()    key

Key	Value	
REPOSITORY_TYPE	"RDB" "BUILTIN"	ex) prop.setProperty("REPOSITORY_TYPE", "RDB");
REPOSITORY_FILE_PATH		ex) prop.setProperty("REPOSITORY_FILE_PATH", "c:/temp_repository");
REPOSITORY_ITEM_NUMBER_PER_DIRECTORY		가 (        : "500") ex) prop.setProperty("REPOSITORY_ITEM_NUMBER_PER_DIRECTORY", "100");
REPOSITORY_HISTORY_ITEM_VALID_DAYS		ex) prop.setProperty("REPOSITORY_HISTORY_ITEM_VALID_DAYS", "20");
REPOSITORY_ADD_COMPRESSED_ITEM	"true" "false"	ex) prop.setProperty("REPOSITORY_ADD_COMPRESSED_ITEM", "false");

// IOZRepositoryCategory Interface Method Summary

■ **getAccessTypeCategory**

---

Prototype    public int getAccessTypeCategory()

---

	가	가
	int ACCESS_CATEGORY_NOT = 0x00000000	
	int ACCESS_CREATEITEM_IN_CATEGORY = 0x00000001	
	int ACCESS_CREATE_CATEGORY = 0x00000002	
	int ACCESS_MODIFY_CATEGORY_NAME = 0x00000004	
	int ACCESS_DELETE_CATEGORY = 0x00000008	
	int ACCESS_UN_DELETE_CATEGORY = 0x00000010	
	int ACCESS_HAS_THE_ITEM_IN_CATEGORY = 0x00000020	
<b>Definition</b>	int ACCESS_GET_ITEMCOUNT_IN_CATEGORY = 0x00000040	
	int ACCESS_GET_ITEMLIST_IN_CATEGORY = 0x00000080	
	int ACCESS_GET_CATEGORYID_OF_ITEM = 0x00000100	
	int ACCESS_GET_CATEGORY_INFO = 0x00000200	
	int ACCESS_GET_DELETED_ITEMLIST_IN_CATEGORY = 0x00000400	
	int ACCESS_GET_SEARCH_ITEMLIST_IN_CATEGORY = 0x00000800	
	int ACCESS_GET_CATEGORYLIST_IN_CATEGORY = 0x00001000	
	int ACCESS_TRANSFER_ITEM = 0x00002000;	
	int ACCESS_TRANSFER_CATEGORY = 0x00004000;	

■ **getOZRepository**

<b>Prototype</b>	public IOZRepository getOZRepository() throws OZRepositoryException
<b>Definition</b>	OZRepositroy 가

■ **createItemInCategory**

<b>Prototype</b>	public String[] createItemInCategory(Object sessionID, String[] itemNames, String[] desc, String[] categoryIDs, InputStream[] item_ins, String comment, int[] errorCode, String[] errorMsg) throws OZRepositoryException
<b>Definition</b>	ID
	<i>sessionID</i> ID
	<i>itemName</i> ID
	<i>descs</i>
<b>Argument</b>	<i>categoryIDs</i> ID
	<i>item_ins</i>
	<i>comment</i>
	<i>errorCode</i>
	<i>errorMsg</i>

■ **createCategory**

<b>Prototype</b>	public String[] createCategory(Object sessionId, String[] categoryNames, String[] pCategoryIDs, String comment, int[] errorCode, String[] errorMsg) throws OZRepositoryException	
<b>Definition</b>	ID	
	<i>sessionId</i>	ID
	<i>categoryName</i>	
<b>Argument</b>	<i>pCategoryIDs</i>	ID
	<i>comment</i>	
	<i>errorCode</i>	
	<i>errorMsg</i>	

■ **modifyCategoryName**

<b>Prototype</b>	public String modifyCategoryName(Object sessionId, String categoryID, String new_CategoryName, String comment) throws OZRepositoryException	
<b>Definition</b>	ID	
	<i>sessionId</i>	ID
	<i>categoryID</i>	ID
<b>Argument</b>	<i>new_categoryName</i>	
	<i>comment</i>	

■ **deleteCategory**

<b>Prototype</b>	public boolean[] deleteCategory(Object sessionId, String[] categoryIDs, boolean[] isDestroys, String comment, int[] errorCode, String[] errorMsg) throws OZRepositoryException	
<b>Definition</b>	ID	
	<i>sessionId</i>	ID
	<i>categoryIDs</i>	ID
<b>Argument</b>	<i>isDestroys</i>	
	<i>comment</i>	
	<i>errorCode</i>	
	<i>errorMsg</i>	

■ **unDeleteCategory**

<b>Prototype</b>	<code>public boolean[] unDeleteCategory(Object sessionId, String[] categoryIDs, String comment, int[] errorCode, String[] errorMsg) throws OZRepositoryException</code>
<b>Definition</b>	가 : "isDestroys=false" 가
<b>Argument</b>	<i>sessionId</i> ID <i>categoryIDs</i> ID <i>comment</i> <i>errorCode</i> <i>errorMsg</i>

■ **hasTheItemInCategory**

<b>Prototype</b>	<code>public boolean hasTheItemInCategory(Object sessionId, String itemID) throws OZRepositoryException</code>
<b>Definition</b>	가
<b>Argument</b>	<i>sessionId</i> ID <i>itemID</i> ID

■ **getItemCountInCategory**

<b>Prototype</b>	<code>public int getItemCountInCategory(Object sessionId, String categoryID) throws OZRepositoryException</code>
<b>Definition</b>	가
<b>Argument</b>	<i>sessionId</i> ID <i>categoryID</i> ID

■ **getItemListInCategory**

<b>Prototype</b>	<code>public IOZItemInfoList getItemListInCategory(Object sessionId, String categoryID) throws OZRepositoryException</code>
<b>Definition</b>	가
<b>Argument</b>	<i>sessionId</i> ID <i>categoryID</i> ID

■ **getCategoryIdOfItem**

<b>Prototype</b>	public String getCategoryIDofItem(Object sessionId, String itemID) throws OZRepositoryException		
<b>Definition</b>	ID		
<b>Argument</b>	<i>sessionId</i>	ID	
	<i>itemID</i>	ID	ID

■ **getCategoryInfo**

<b>Prototype</b>	public IOZCategoryInfo getCategoryInfo(Object sessionId, String categoryID) throws OZRepositoryException		
<b>Definition</b>	가		
<b>Argument</b>	<i>sessionId</i>	ID	
	<i>categoryID</i>	가	ID

■ **getDeletedItemListInCategory**

<b>Prototype</b>	public IOZItemInfoList getDeletedItemListInCategory(Object sessionId, String categoryID) throws OZRepositoryException		
<b>Definition</b>	가 : "isDestroys=false" 가		
<b>Argument</b>	<i>sessionId</i>	ID	
	<i>categoryID</i>	가	ID

■ **transferItem**

<b>Prototype</b>	public boolean transferItem(Object sessionId, String[] itemIDs, String target_CategoryID) throws OZRepositoryException		
<b>Definition</b>	가		
<b>Argument</b>	<i>sessionId</i>	ID	
	<i>itemIDs</i>		ID
	<i>target_CategoryID</i>		ID

■ **transferCategory**

<b>Prototype</b>	public boolean transferCategory(Object sessionId, String categoryID, String target_CategoryID) throws OZRepositoryException		
<b>Definition</b>	가		

	<i>sessi onID</i>	ID
<b>Argument</b>	<i>i temIDs</i>	ID
	<i>target_CategoryID</i>	ID

// IOZRepositoryItem Interface Method Summary

■ **getAccessTypeItem**

<b>Prototype</b>	public int getAccessTypeCategory()
	가 가 .
<b>Definition</b>	int ACCESS_ITEM_NOT = 0x00000000 int ACCESS_CREATE_ITEM = 0x00000001 int ACCESS_MODIFY_ITEMNAME = 0x00000002 int ACCESS_DELETE_ITEM = 0x00000004 int ACCESS_UN_DELETE_ITEM = 0x00000008 int ACCESS_MODIFY_ITEM_DESC = 0x00000010 int ACCESS_HAS_THE_ITEM = 0x00000020 int ACCESS_GET_ITEM_INFO = 0x00000040 int ACCESS_GET_ITEMS_UNCONDITION = 0x00000080 int ACCESS_GET_ITEMS = 0x00000100 int ACCESS_CHECKOUT_ITEM = 0x00000200 int ACCESS_CHECKIN_ITEM = 0x00000400 int ACCESS_UNDO_CHECKOUT_ITEM = 0x00000800 int ACCESS_IS_CHECKOUT_USER = 0x00001000

■ **getOZRepository**

<b>Prototype</b>	public IOZRepository getOZRepository() throws OZRepositoryException
<b>Definition</b>	OZRepostiroy 가 .

■ **createItem**

<b>Prototype</b>	public String[] createItem(Object sessionID, String[] itemNames, String[] desc, InputStream[] item_ins, String comment, int[] errorCode, String[] errorMsg) throws OZRepositoryException
<b>Definition</b>	ID .
<b>Argument</b>	<i>sessi onID</i> ID
	<i>i temNames</i>
	<i>descs</i>

<i>itemIns</i>
<i>comment</i>
<i>errCodes</i>
<i>errMsgs</i>

■ **modifyItemName**

<b>Prototype</b>	public String modifyItemName(Object sessionId, String itemID, String newItemName, String comment) throws OZRepositoryException	
<b>Definition</b>	ID	ID
	<i>sessionId</i>	ID
<b>Argument</b>	<i>itemID</i>	ID
	<i>newItemName</i>	
	<i>comment</i>	

■ **deleteItem**

<b>Prototype</b>	public boolean[] deleteItem(Object sessionId, String[] itemIDs, boolean[] isDestroys, String comment, int[] errorCode, String[] errorMsg) throws	
<b>Definition</b>		가
	<i>sessionId</i>	ID
	<i>itemIDs</i>	ID
<b>Argument</b>	<i>isDestroys</i>	
	<i>comment</i>	
	<i>errCodes</i>	
	<i>errMsgs</i>	

■ **unDeleteItem**

<b>Prototype</b>	public boolean[] unDeleteItem(Object sessionId, String[] itemIDs, String comment, int[] errorCode, String[] errorMsg) throws OZRepositoryException	
<b>Definition</b>	가	"isDestroys=false" 가
<b>Argument</b>	<i>sessionId</i>	ID

<i>itemIDs</i>	ID
<i>comment</i>	
<i>errCodes</i>	
<i>errMsgs</i>	

■ **modifyItemDesc**

<b>Prototype</b>	public boolean modifyItemDesc(Object sessionId, String itemID, String desc) throws OZRepositoryException	
<b>Definition</b>	ID	가
	<i>sessionId</i>	ID
<b>Argument</b>	<i>itemID</i>	ID
	<i>desc</i>	

■ **getItemInfo**

<b>Prototype</b>	public IOZItemInfo getItemInfo(Object sessionId, String itemID) throws OZRepositoryException	
<b>Definition</b>	ID	가
	<i>sessionId</i>	ID
<b>Argument</b>	<i>itemID</i>	가 ID

■ **hasTheItem**

<b>Prototype</b>	public boolean hasTheItem(Object sessionId, String itemID) throws OZRepositoryException	
<b>Definition</b>	가	
	<i>sessionId</i>	ID
<b>Argument</b>	<i>itemID</i>	ID

■ **getItems**

<b>Prototype</b>	public InputStream[] getItems(Object sessionId, String itemIDs[], int[] errorCode, String[] errorMsg) throws OZRepositoryException	
<b>Prototype</b>	public InputStream[] getItems(Object sessionId, String itemIDs[], long[] modifiedTimes, int[] errorCode, String[] errorMsg) throws OZRepositoryException	

<b>Definition</b>	가	가	.
	<i>sessi onID</i>	ID	
	<i>i temIDs</i>	가	ID
<b>Argument</b>	<i>modi fi edTimes</i>		
	<i>errCodes</i>		
	<i>errMsgs</i>		

■ **checkOutItem**

<b>Prototype</b>	public InputStream[] checkOutItem(Object sessi onID, String i temIDs[], String[] local CheckOutFol ders, long[] local FileTimes, int[] errorCode, String[] errorMsg) throws OZReposi toryExcepti on		
<b>Definition</b>		ID	가
	<i>sessi onID</i>	ID	
	<i>i temIDs</i>		ID
<b>Argument</b>	<i>local CheckOut Fol ders</i>		
	<i>local FileTimes</i>		
	<i>errCodes</i>		
	<i>errMsgs</i>		

■ **checkInItem**

<b>Prototype</b>	public boolean[] checkInItem(Object sessi onID, String[] i temIDs, InputStream[] i tem_ins, String comment, boolean[] keepCheckOut, int[] errorCode, String[] errorMsg) throws OZReposi toryExcepti on		
<b>Definition</b>		ID	가
	<i>sessi onID</i>	ID	
	<i>i temIDs</i>		ID
	<i>i tem Streams</i>		
<b>Argument</b>	<i>comment</i>		
	<i>keepCheckOut</i>		
	<i>errCodes</i>		
	<i>errMsgs</i>		

■ **undoCheckOutItem**

**Prototype** public InputStream[] undoCheckOutItem(Object sessonID, String[] itemIDs, boolean[] isReplaces, int[] errorCode, String[] errorMsg) throws OZRepositoryException

**Definition**

<i>sessonID</i>	ID
<i>itemIDs</i>	ID

**Argument** *isReplaces*

*errCodes*

*errMsgs*

■ **isCheckOutUser**

**Prototype** public boolean[] isCheckOutUser(Object sessonID, String[] itemIDs) throws OZRepositoryException

**Definition** 가

**Argument** *sessonID* ID

*itemIDs* ID

// IOZRepositoryItemHistory

■ **getAccessTypeItemHistory**

**Prototype** public int getAccessTypeItemHistory()

가 가

**Definition**  
 int ACCESS\_ITEM\_HISTORY\_NOT = 0x00000000  
 int ACCESS\_ROLLBACK\_ITEM = 0x00000001  
 int ACCESS\_GET\_ITEM\_BY\_VERSION = 0x00000002  
 int ACCESS\_GET\_HISTORY\_ITEMLIST = 0x00000004  
 int ACCESS\_GET\_DELETE\_HISTORY\_ITEMLIST = 0x00000008  
 int ACCESS\_REMOVE\_HISTORY\_ITEM = 0x00000010

■ **getOZRepository**

**Prototype** public IOZRepository getOZRepository() throws OZRepositoryException

**Definition** OZRepositroy 가

■ **rollBackItem**

<b>Prototype</b>	<code>public boolean rollBackItem(Object sessionId, String itemID, int itemVersion, String comment) throws OZRepositoryException</code>								
<b>Definition</b>	.								
<b>Argument</b>	<table border="0"> <tr> <td><i>sessionId</i></td> <td>ID</td> </tr> <tr> <td><i>itemID</i></td> <td>ID</td> </tr> <tr> <td><i>itemVersion</i></td> <td></td> </tr> <tr> <td><i>comment</i></td> <td></td> </tr> </table>	<i>sessionId</i>	ID	<i>itemID</i>	ID	<i>itemVersion</i>		<i>comment</i>	
<i>sessionId</i>	ID								
<i>itemID</i>	ID								
<i>itemVersion</i>									
<i>comment</i>									

■ **getItemByVersion**

<b>Prototype</b>	<code>public InputStream getItemByVersion(Object sessionId, String itemID, int version) throws OZRepositoryException</code>						
<b>Definition</b>	ID 가 .						
<b>Argument</b>	<table border="0"> <tr> <td><i>sessionId</i></td> <td>ID</td> </tr> <tr> <td><i>itemID</i></td> <td>가 ID</td> </tr> <tr> <td><i>version</i></td> <td>가</td> </tr> </table>	<i>sessionId</i>	ID	<i>itemID</i>	가 ID	<i>version</i>	가
<i>sessionId</i>	ID						
<i>itemID</i>	가 ID						
<i>version</i>	가						

■ **getHistoryItemList**

<b>Prototype</b>	<code>public IOZHistoryInfoList getHistoryItemList(Object sessionId, String itemID) throws OZRepositoryException</code>				
<b>Definition</b>	가 .				
<b>Argument</b>	<table border="0"> <tr> <td><i>sessionId</i></td> <td>ID</td> </tr> <tr> <td><i>itemIDs</i></td> <td>가 ID</td> </tr> </table>	<i>sessionId</i>	ID	<i>itemIDs</i>	가 ID
<i>sessionId</i>	ID				
<i>itemIDs</i>	가 ID				

■ **getDeleteHistoryItemInfo**

<b>Prototype</b>	<code>public IOZHistoryInfoList getDeleteHistoryItemInfo(Object sessionId, String itemID) throws OZRepositoryException</code>				
<b>Definition</b>	가 : "isDestroys=false" 가				
<b>Argument</b>	<table border="0"> <tr> <td><i>sessionId</i></td> <td>ID</td> </tr> <tr> <td><i>itemIDs</i></td> <td>가 ID</td> </tr> </table>	<i>sessionId</i>	ID	<i>itemIDs</i>	가 ID
<i>sessionId</i>	ID				
<i>itemIDs</i>	가 ID				

■ **removeHistoryItem**

<b>Prototype</b>	public boolean removeHistoryItem(Object sessionId, String itemID, int itemVersion) throws OZRepositoryException
<b>Definition</b>	<i>sessionId</i> ID
<b>Argument</b>	<i>itemID</i> ID <i>item version</i>

// IOZRepositoryGroup

■ **getAccessTypeGroup**

<b>Prototype</b>	public int getAccessTypeGroup() 가 가
<b>Definition</b>	int ACCESS_GROUP_NOT = 0x00000000 int ACCESS_CREATE_GROUP = 0x00000001 int ACCESS_MODIFY_GROUP_NAME = 0x00000002 int ACCESS_MODIFY_GROUP_DESC = 0x00000004 int ACCESS_DELETE_GROUP = 0x00000008 int ACCESS_CREATE_USER_IN_GROUP = 0x00000010 ACCESS_TRANSFER_USER = 0x00000020 int ACCESS_GET_USERINFLIST_IN_GROUP = 0x00000040 int ACCESS_GET_GROUPINFO = 0x00000080 int ACCESS_GET_SUB_GROUPINFO_LIST = 0x00000100 int ACCESS_GET_PARENT_GROUPINFO = 0x00000200 int ACCESS_GET_GROUPID_OF_USER = 0x00000400 int ACCESS_ADD_GROUP_ADMIN = 0x00000800 int ACCESS_REMOVE_GROUP_ADMIN = 0x00001000 int ACCESS_IS_USER_GROUP_ADMIN = 0x00002000 int ACCESS_GET_USER_ADMINLIST_IN_GROUP = 0x00004000 int ACCESS_TRANSFER_GROUP = 0x00008000

■ **getOZRepository**

<b>Prototype</b>	public IOZRepository getOZRepository() throws OZRepositoryException
<b>Definition</b>	OZRepository 가

■ **createGroup**

<b>Prototype</b>	public String createGroup(Object sessionId, String groupName, String groupId, String desc) throws OZRepositoryException
<b>Definition</b>	ID

	<i>sessi onID</i>	ID
Argument	<i>gName</i>	
	<i>pGroupID</i>	ID
	<i>desc</i>	

■ **modifyGroupName**

Prototype	public String modifyGroupName(Object sessi onID, String groupID, String gName) throws OZRepositoryException	
Definition	ID	ID
	<i>sessi onID</i>	ID
Argument	<i>groupID</i>	ID
	<i>gName</i>	

■ **modifyGroupDesc**

Prototype	public boolean modifyGroupDesc(Object sessi onID, String groupID, String desc) throws OZRepositoryException	
Definition	ID	
	<i>sessi onID</i>	ID
Argument	<i>groupID</i>	ID
	<i>desc</i>	

■ **deleteGroup**

Prototype	public boolean deleteGroup(Object sessi onID, String groupID) throws OZRepositoryException	
Definition	ID	
	<i>sessi onID</i>	ID
Argument	<i>groupID</i>	ID

■ **createUserInGroup**

Prototype	public String createUserInGroup(Object sessi onID, String userName, String password, String groupID, String desc) throws OZRepositoryException	
Definition	ID	ID

	<i>sessi onI D</i>	ID
	<i>user Name</i>	
<b>Argument</b>	<i>password</i>	
	<i>groupI D</i>	ID
	<i>desc</i>	

■ **getUserInfoListInGroup**

<b>Prototype</b>	public IOZUserInfoList getUserInfoListInGroup(Object sessi onI D, String groupI D) throws OZRepositoryExcepti on	
<b>Definition</b>	ID	가 .
<b>Argument</b>	<i>sessi onI D</i>	ID
	<i>groupI D</i>	가 ID

■ **getGroupInfo**

<b>Prototype</b>	public IOZGroupInfo getGroupInfo(Object sessi onI D, String groupI D) throws OZRepositoryExcepti on	
<b>Definition</b>	ID	가 .
<b>Argument</b>	<i>sessi onI D</i>	ID
	<i>groupI D</i>	ID

■ **getSubGroupInfoList**

<b>Prototype</b>	public IOZGroupInfoList getSubGroupInfoList(Object sessi onI D, String groupI D) throws OZRepositoryExcepti on	
<b>Definition</b>		가 .
<b>Argument</b>	<i>sessi onI D</i>	ID
	<i>groupI D</i>	가 ID

■ **getParentGroupInfo**

<b>Prototype</b>	public IOZGroupInfo getParentGroupInfo(Object sessi onI D, String groupI D) throws OZRepositoryExcepti on	
<b>Definition</b>		가 .
<b>Argument</b>	<i>sessi onI D</i>	ID
	<i>groupI D</i>	가 ID

■ **getGroupIdOfUser**

<b>Prototype</b>	public IOZGroupInfo getGroupIdOfUser(Object sessionID, String userID) throws OZRepositoryException		
<b>Definition</b>	가	가	.
<b>Argument</b>	<i>sessionID</i>	ID	
	<i>userID</i>	가	ID

■ **addGroupAdmin**

<b>Prototype</b>	public boolean addGroupAdmin(Object sessionID, String userID, String groupId) throws OZRepositoryException		
<b>Definition</b>	가	가	.
	<i>sessionID</i>	ID	
<b>Argument</b>	<i>userID</i>	가	ID
	<i>groupId</i>	가	ID

■ **removeGroupAdmin**

<b>Prototype</b>	public boolean removeGroupAdmin(Object sessionID, String userID, String groupId) throws OZRepositoryException		
<b>Definition</b>	.		
	<i>sessionID</i>	ID	
<b>Argument</b>	<i>userID</i>		ID
	<i>groupId</i>		ID

■ **isUserGroupAdmin**

<b>Prototype</b>	public boolean isUserGroupAdmin(Object sessionID, String userID, String groupId) throws OZRepositoryException		
<b>Definition</b>	ID	가	.
	<i>sessionID</i>	ID	
<b>Argument</b>	<i>userID</i>		ID
	<i>groupId</i>	ID	

■ **getUserAdminListInGroup**

<b>Prototype</b>	<code>public IOZUserInfoList getUserAdminListInGroup(Object sessionID, String groupID) throws OZRepositoryException</code>
<b>Definition</b>	가 .
<b>Argument</b>	<i>sessionID</i> ID <i>groupID</i> 가 ID

■ **transferUser**

<b>Prototype</b>	<code>public boolean transferUser(Object sessionID, String userID, String target_GroupID) throws OZRepositoryException</code>
<b>Definition</b>	가 .
<b>Argument</b>	<i>sessionID</i> ID <i>userID</i> ID <i>target_groupID</i> ID

■ **transferGroup**

<b>Prototype</b>	<code>public boolean transferGroup(Object sessionID, String groupID, String target_GroupID) throws OZRepositoryException</code>
<b>Definition</b>	가 .
<b>Argument</b>	<i>sessionID</i> ID <i>userID</i> ID <i>target_groupID</i> ID

// **IOZRepositoryUser**

■ **getAccessTypeUser**

<b>Prototype</b>	<code>public int getAccessTypeUser()</code>
<b>Definition</b>	가 가 . <code>int ACCESS_USER_NOT = 0x00000000</code> <code>int ACCESS_CREATE_USER = 0x00000001</code> <code>int ACCESS_MODIFY_USER_NAME = 0x00000002</code> <code>int ACCESS_MODIFY_USER_PASSWORD = 0x00000004</code> <code>int ACCESS_MODIFY_USER_DESC = 0x00000008</code> <code>int ACCESS_DELETE_USER = 0x00000010</code> <code>int ACCESS_GET_USERINFO = 0x00000020</code> <code>int ACCESS_CHECK_USER_PASSWORD = 0x00000040</code> <code>int ACCESS_GET_USERINFO_LIST = 0x00000080</code> <code>int ACCESS_GET_IS_CHECK_ADMIN = 0x00000100</code>



<b>Prototype</b>	public boolean modifyUserDesc(Object sessionID, String userID, String desc) throws OZRepositoryException
<b>Definition</b>	ID
<b>Argument</b>	<i>sessionID</i> ID <i>userID</i> ID <i>desc</i>

■ **deleteUser**

<b>Prototype</b>	public boolean deleteUser(Object sessionID, String userID) throws OZRepositoryException
<b>Definition</b>	ID
<b>Argument</b>	<i>sessionID</i> ID <i>userID</i> ID

■ **getUserInfo**

<b>Prototype</b>	public IOZUserInfo getUserInfo(Object sessionID, String userID) throws OZRepositoryException
<b>Definition</b>	ID 가
<b>Argument</b>	<i>sessionID</i> ID <i>userID</i> 가 ID

■ **checkUserPassword**

<b>Prototype</b>	public boolean checkUserPassword(Object sessionID, String userID, String password) throws OZRepositoryException
<b>Definition</b>	가
<b>Argument</b>	<i>sessionID</i> ID <i>userID</i> ID <i>password</i>

■ **getUserInfoList**

<b>Prototype</b>	public IOZUserInfoList getUserInfoList(Object sessionID) throws OZRepositoryException
<b>Definition</b>	가

Argument	<i>sessionID</i>	ID
<b>isCheckAdmin</b>		
Prototype	public boolean isCheckAdmin(Object sessionID, String userID) throws OZRepositoryException	
Definition	ID	가
Argument	<i>sessionID</i>	ID
	<i>userID</i>	ID

// IOZRepositoryMultiLoginUser

<b>getAccessTypeMultiLoginUser</b>		
Prototype	public int getAccessTypeMultiLoginUser() 가	
Definition	int ACCESS_MULTILOGINUSER_NOT = 0x00000000 int ACCESS_DISABLE_USER_LOGIN = 0x00000001 int ACCESS_ENABLE_USER_LOGIN = 0x00000002 int ACCESS_IS_LOGIN_ENABLED = 0x00000004	
<b>getOZRepository</b>		
Prototype	public IOZRepository getOZRepository() throws OZRepositoryException	
Definition	OZRepository	가
<b>disableUserLogin</b>		
Prototype	public void disableUserLogin(Object sessionID) throws OZRepositoryException	
Definition	ID	
Argument	<i>sessionID</i>	ID
<b>enableUserLogin</b>		
Prototype	public void enableUserLogin(Object sessionID) throws OZRepositoryException	
Definition	ID	가
Argument	<i>sessionID</i>	가 ID





■ **getCategoryListAuthToGroupInCategory**

<b>Prototype</b>	public IOZCategoryInfoList getCategoryListAuthToGroupInCategory(Object sessionID, String groupID, String categoryID, int perm) throws OZRepositoryException	
<b>Definition</b>		perm
	가	.
	<i>sessionID</i>	ID
<b>Argument</b>	<i>groupID</i>	ID
	<i>categoryID</i>	ID
	<i>perm</i>	

// IOZRepositoryAuthGroupItem

■ **getAccessTypeAuthGroupItem**

<b>Prototype</b>	public int getAccessTypeAuthGroupItem()	
	가	가
<b>Definition</b>	int ACCESS_TYPE_AUTH_GROUP_ITEM_NOT = 0x00000000 int ACCESS_TYPE_MODIFY_ITEM_GROUPAUTH = 0x00000010 int ACCESS_TYPE_GET_GROUPAUTH_TO_ITEM = 0x00000020 int ACCESS_TYPE_GET_GROUPLIST_AUTH_TO_ITEM = 0x00000040 int ACCESS_TYPE_GET_ITEMLIST_AUTH_TO_GROUP = 0x00000080	

■ **getOZRepository**

<b>Prototype</b>	public IOZRepository getOZRepository() throws OZRepositoryException	
<b>Definition</b>	OZRepostiroy	가

■ **modifyItemGroupAuth**

<b>Prototype</b>	public boolean modifyItemGroupAuth(Object sessionID, String groupID, String itemID, int perm) throws OZRepositoryException	
<b>Definition</b>	ID	ID
<b>Argument</b>	<i>sessionID</i>	ID
	<i>groupID</i>	ID
	<i>itemID</i>	ID

*perm*

■ **getGroupAuthToItem**

**Prototype** public int getGroupAuthToItem(Object sessionId, String groupId, String itemId) throws OZRepositoryException

**Definition** ID ID 가 .  
*sessionId* ID

**Argument** *groupId* ID  
*itemId* ID

■ **getGroupListAuthToItem**

**Prototype** public IOZGroupInfoList getGroupListAuthToItem(Object sessionId, String itemId, int perm) throws OZRepositoryException

**Definition** ID perm 가 .  
*sessionId* ID

**Argument** *itemId* ID  
*perm*

■ **getItemListAuthToGroup**

**Prototype** public IOZItemInfoList getItemListAuthToGroup(Object sessionId, String groupId, int perm) throws OZRepositoryException

**Definition** ID perm 가 .  
*sessionId* ID

**Argument** *groupId* ID  
*perm*

// IOZRepositoryAuthUserCategory

■ **getAccessTypeAuthUserCategory**

**Prototype** public int getAccessTypeAuthUserCategory()

	가
	가
<b>Definition</b>	int ACCESS_TYPE_AUTH_USER_CATEGORY_NOT = 0x00000000 int ACCESS_TYPE_MODIFY_CATEGORY_USERAUTH = 0x00000100 int ACCESS_TYPE_GET_USERAUTH_TO_CATEGORY = 0x00000200 int ACCESS_TYPE_GET_USERLIST_AUTH_TO_CATEGORY = 0x00000400 int ACCESS_TYPE_GET_ITEMLIST_AUTH_TO_USER_IN_CATEGORY = 0x00000800 int ACCESS_TYPE_GET_CATEGORYLIST_AUTH_TO_USER_IN_CATEGORY = 0x00020000

■ **getOZRepository**

<b>Prototype</b>	public IOZRepository getOZRepository() throws OZRepositoryException
<b>Definition</b>	OZRepository 가

■ **modifyCategoryUserAuth**

<b>Prototype</b>	public boolean modifyCategoryUserAuth(Object sessionId, String userID, String categoryID, int perm) throws OZRepositoryException
<b>Definition</b>	ID ID
	<i>sessionId</i> ID
<b>Argument</b>	<i>userID</i> ID
	<i>categoryID</i> ID
	<i>perm</i>

■ **getUserAuthToCategory**

<b>Prototype</b>	public int getUserAuthToCategory(Object sessionId, String userID, String categoryID) throws OZRepositoryException
<b>Definition</b>	ID ID 가
	<i>sessionId</i> ID
<b>Argument</b>	<i>userID</i> ID
	<i>categoryID</i> ID

■ **getUserListAuthToCategory**

<b>Prototype</b>	public IOZUserInfoList getUserListAuthToCategory(Object sessionId, String categoryID, int perm) throws OZRepositoryException
------------------	--

Definition	가 .	ID	perm
	<i>sessi onID</i>	ID	
Argument	<i>categoryID</i>	ID	
	<i>perm</i>		

■ **getI temListAuthToUserInCategory**

Prototype	public IOZItemInfoList getI temListAuthToUserInCategory(Object sessi onID, String userID, String categoryID, int perm) throws OZRepositoryExcepti on		
Definition	가 .		perm
	<i>sessi onID</i>	ID	
	<i>userID</i>	ID	
Argument	<i>categoryID</i>	ID	
	<i>perm</i>		

■ **getCategoryListAuthToUserInCategory**

Prototype	public IOZCategoryInfoList getCategoryLi stAuthToUserInCategory(Object sessi onID, String userID, String categoryID, int perm) throws OZRepositoryExcepti on		
Definition	가 .		perm
	<i>sessi onID</i>	ID	
	<i>userID</i>	ID	
Argument	<i>categoryID</i>	ID	
	<i>perm</i>		

// IOZRepositoryAuthUserI tem

■ **getAccessTypeAuthUserI tem**

Prototype	public int getAccessTypeAuthUserI tem()
-----------	---

가

가  
**Definition** int ACCESS\_TYPE\_AUTH\_USER\_ITEM\_NOT = 0x00000000  
 int ACCESS\_TYPE\_MODIFY\_ITEM\_USERAUTH = 0x00001000  
 int ACCESS\_TYPE\_GET\_USERAUTH\_TO\_ITEM = 0x00002000  
 int ACCESS\_TYPE\_GET\_USERLIST\_AUTH\_TO\_ITEM = 0x00004000  
 int ACCESS\_TYPE\_GET\_ITEMLIST\_AUTH\_TO\_USER = 0x00008000

■ **getOZRepository**

**Prototype** public IOZRepository getOZRepository() throws  
 OZRepositoryException

**Definition** OZRepostiroy 가

■ **modifyItemUserAuth**

**Prototype** public boolean modifyItemUserAuth(Object sessionId, String  
 userID, String itemID, int perm) throws  
 OZRepositoryException

**Definition** 가

*sessionId* ID

*userID* ID

**Argument** *itemID* ID

*perm*

■ **getUserAuthToItem**

**Prototype** public int getUserAuthToItem(Object sessionId, String  
 userID, String itemID) throws OZRepositoryException

**Definition** 가

*sessionId* ID

**Argument** *userID* ID

*itemID*

■ **getUserListAuthToItem**

**Prototype** public IOZUserInfoList getUserListAuthToItem(Object  
 sessionId, String itemID, int perm) throws  
 OZRepositoryException

Definition	perm	가
Argument	<i>sessionID</i>	ID
Argument	<i>itemId</i>	ID
Argument	<i>perm</i>	

■ **getItemListAuthToUser**

Prototype	public IOZItemInfoList getItemListAuthToUser(Object sessionID, String userID, int perm) throws OZRepositoryException	
Definition	perm	가
Argument	<i>sessionID</i>	ID
Argument	<i>userID</i>	ID
Argument	<i>perm</i>	

// IOZCategoryInfo

■ **getCategoryID**

Prototype	public String getCategoryID()
Definition	ID 가

■ **getCategoryName**

Prototype	public String getCategoryName()
Definition	가

■ **getParentCategoryID**

Prototype	public String getParentCategoryID()
Definition	ID 가

■ **getParentCategoryName**

Prototype	public String getParentCategoryName()
Definition	가

■ **getComment**

Prototype	public String getComment()
-----------	----------------------------



---

**Prototype** `public String getItemID()`

**Definition** ID 가 .

---

■ **getItemName**

---

**Prototype** `public String getItemName()`

**Definition** 가 .

---

■ **getCategoryID**

---

**Prototype** `public String getCategoryID()`

**Definition** ID 가 .

---

■ **getCategoryName**

---

**Prototype** `public String getCategoryName()`

**Definition** 가 .

---

■ **getUpdateTime**

---

**Prototype** `public Long getUpdateTime()`

**Definition** 가 .

---

■ **getDesc**

---

**Prototype** `public String getDesc()`

**Definition** 가 .

---

■ **getComment**

---

**Prototype** `public String getComment()`

**Definition** 가 .

---

■ **getCreateUserID**

---

**Prototype** `public String getCreateUserID()`

**Definition** ID 가 .

---

■ **getCreateUserName**

---

**Prototype** `public String getCreateUserName()`

---

Definition 가 .

■ **getCheckoutUser**

Prototype public String getCheckoutUser()

Definition 가 .

■ **getCheckoutLocalPath**

Prototype public String getCheckoutLocalPath()

Definition 가 .

■ **getPerm**

Prototype public int getPerm()

Definition 가 .

// IOZHistoryInfo

■ **getItemID**

Prototype public String getItemID()

Definition ID 가 .

■ **getItemName**

Prototype public String getItemName()

Definition 가 .

■ **getHistoryItemVersion**

Prototype public int getHistoryItemVersion()

Definition 가 .

■ **getHistoryCheckInTime**

Prototype public long getHistoryCheckInTime()

Definition 가 .

■ **getHistoryCheckInUser**

Prototype public String getHistoryCheckInUser()

**Definition** ID 가 .

■ **getHistoryCheckInComment**

**Prototype** public String getHistoryCheckInComment()

**Definition** 가 .

// IOZGroupInfo

■ **isGroupAdmin**

**Prototype** public boolean isGroupAdmin(String userID)

**Definition** ID가 가 .

**Argument** *userID* ID

■ **getGroupAdminList**

**Prototype** public HashMap getGroupAdminList()

**Definition** : "key= ID, value= " HashMap .

■ **getGroupID**

**Prototype** public String getGroupID()

**Definition** ID 가 .

■ **getGroupName**

**Prototype** public String getGroupName()

**Definition** 가 .

■ **getParentGroupID**

**Prototype** public String getParentGroupID()

**Definition** ID 가 .

■ **getParentGroupName**

**Prototype** public String getParentGroupName()

**Definition** 가 .

■ **getDesc**

Prototype `public String getDesc()`

Definition `가 .`

■ **getUpdateTime**

Prototype `public long getUpdateTime()`

Definition `가 .`

// IOZUserInfo

■ **getUserID**

Prototype `public String getUserID()`

Definition `ID 가 .`

■ **getPassword**

Prototype `public String getPassword()`

Definition `가 .`

■ **getUserName**

Prototype `public String getUserName()`

Definition `가 .`

■ **getGroupID**

Prototype `public String getGroupID()`

Definition `가 ID 가 .`

■ **getGroupName**

Prototype `public String getGroupName()`

Definition `가 가 .`

■ **getDesc**

Prototype `public String getDesc()`

Definition `가 .`



■ **getMACAddress**

Prototype `public String getMACAddress()`

Definition MAC Address 가 .

■ **getIPAddress**

Prototype `public String getIPAddress()`

Definition IP Address 가 .

■ **getHDDSerialNo**

Prototype `public String getHDDSerialNo()`

Definition HDD Serial 가 .

■ **getUserID**

Prototype `public String getUserID()`

Definition 가 ID 가 .

■ **getUserName**

Prototype `public String getUserName()`

Definition 가 가 .

■ **getSessionID**

Prototype `public String getSessionID()`

Definition 가 ID 가 .

■ **getClientIP**

Prototype `public String getClientIP()`

Definition 가 IP 가 .

// IOZCategoryInfoList

■ **getCategoryInfo**

Prototype `public IOZCategoryInfo getCategoryInfo(int index)`

Definition 가 .

Argument *index*



Prototype	<code>public IOZHistoryInfo[] getHistoryInfos()</code>
-----------	--

Definition	가 .
------------	-----

// IOZGroupInfoList

■ **getGroupInfo**

Prototype	<code>public IOZGroupInfo getGroupInfo(int index)</code>
-----------	--

Definition	가 .
------------	-----

Argument	<i>index</i>
----------	--------------

■ **getSize**

Prototype	<code>public String getSize()</code>
-----------	--------------------------------------

Definition	가 .
------------	-----

■ **getGroupInfos**

Prototype	<code>public IOZGroupInfo[] getGroupInfos()</code>
-----------	--

Definition	가 .
------------	-----

// IOZUserInfoList

■ **getGroupInfo**

Prototype	<code>public IOZUserInfoList getUserInfo(int index)</code>
-----------	--

Definition	가 .
------------	-----

Argument	<i>index</i>
----------	--------------

■ **getSize**

Prototype	<code>public String getSize()</code>
-----------	--------------------------------------

Definition	가 .
------------	-----

■ **getGroupInfos**

Prototype	<code>public IOZUserInfo[] getUserInfos()</code>
-----------	--

Definition	가 .
------------	-----

## Class OZRepositoryException

// : oz.framework.repositoryex

■ **public class OZRepositoryException extends Exception**

■ **Method Summary**

- **public OZRepositoryException(String \_msg)**
- **public OZRepositoryException(int code, String \_msg)**
- **public OZErrorCode ErrorCode{get;}**
- **public String Message{get;}**

■ **Method Detail**

- OZRepositoryException

	<code>public OZRepositoryException(String _msg)</code>
<b>Prototype</b>	
	<code>public OZRepositoryException(int code, String _msg)</code>
<b>Definition</b>	<code>OZRepositoryException</code>
<b>Argument</b>	<i>code</i>
	<i>_msg</i>

- ErrorCode

<b>Prototype</b>	<code>public OZErrorCode ErrorCode{get;}</code>
<b>Definition</b>	

- Message

<b>Prototype</b>	<code>public String Message{get;}</code>
<b>Definition</b>	

## Class RepositoryEX

oz.framework.api.RepositoryEx

Constructor

### Constructor Summary

- **public RepositoryEx(String ip, int port, String id, String pw) throws OZCPEException**
- **public RepositoryEx(String ip, int port, String id, String pw, boolean useUSL) throws OZCPEException**
- **public RepositoryEx(String iurl, String id, String pw) throws OZCPEException**
- **public RepositoryEx(String iurl, String id, String pw, boolean useUSL) throws OZCPEException**

### Method Summary

- **public void setRepositoryConfig(SortProperties prop) throws OZCPEException**
- **public SortProperties getRepositoryConfig() throws OZCPEException**
- **public void reloadRepository() throws OZCPEException**
- **public String[] createItem(String[] itemNames, String[] iDescs, String[] categoryIDs, boolean[] isComp, InputStream[] itemStreams, String comment, int[] errorCodes, String[] errorMsg) throws OZCPEException**
- **public String modifyItemName(String itemID, String itemName, String comment) throws OZCPEException**
- **public boolean[] deleteItem(String[] itemIDs, boolean[] isDestorys, String comment, int[] errorCodes, String[] errorMsg) throws OZCPEException**
- **public boolean[] undeleteItem(String[] itemIDs, String comment, int[] errorCodes, String[] errorMsg) throws OZCPEException**
- **public boolean modifyItemDesc(String itemID, String iDesc) throws OZCPEException**

- `public IOZItemInfo getItemInfo(String itemID) throws OZCPEException`
- `public boolean hasTheItem(String itemID) throws OZCPEException`
- `public InputStream[] getItemsUnCondition(String[] itemIDs, boolean[] isComp, boolean[] isObjStream, String[] categoryIDs, int[] errorCodes, String[] errorMsg) throws OZCPEException`
- `public InputStream[] getItems(String[] itemIDs, long[] modifiedTimes, boolean[] isComp, boolean[] isObjStream, String[] categoryIDs, int[] errorCodes, String[] errorMsg) throws OZCPEException`
- `public InputStream[] checkOutItem(String itemIDs[], String[] localCheckOutFolders, long[] localFileTimes, boolean[] isComp, int[] errorCodes, String[] errorMsg) throws OZCPEException`
- `public boolean[] checkInItem(String[] itemIDs, boolean[] isComp, InputStream[] iStreams, String comment, boolean[] keepCheckOut, int[] errorCodes, String[] errorMsg) throws OZCPEException`
- `public InputStream[] undoCheckOutItem(String[] itemIDs, boolean[] isReplaces, boolean[] isComp, int[] errorCodes, String[] errorMsg) throws OZCPEException`
- `public boolean[] isCheckOutUser(String[] itemIDs) throws OZCPEException`
- `public boolean rollBackItem(String itemID, int itemVersion, String comment) throws OZCPEException`
- `public InputStream getItemByVersion(String itemID, int version, boolean isComp) throws OZCPEException`
- `public IOZHistoryInfoList getHistoryItemList(String itemID) throws OZCPEException`
- `public IOZHistoryInfoList getDeleteHistoryItemInfo(String itemID) throws OZCPEException`
- `public boolean removeHistoryItem(String itemID, int itemVersion) throws OZCPEException`
- `public String[] createCategory(String[] categoryNames, String[] pCategoryIDs, String comment, int[] errorCodes, String[] errorMsg) throws OZCPEException`
- `public String modifyCategoryName(String categoryID, String new_CategoryName, String comment) throws OZCPEException`
- `public boolean[] deleteCategory(String[] categoryIDs, boolean[] isDestroys, String comment, int[] errorCodes, String[] errorMsg) throws`

**OZCPEException**

- **public boolean[] unDeleteCategory(String[] categoryIDs, String comment, int[] errorCodes, String[] errorMsg) throws OZCPEException**
- **public int getItemCountInCategory(String categoryID) throws OZCPEException**
- **public IOZItemInfoList getItemListInCategory(String categoryID) throws OZCPEException**
- **public IOZCategoryInfoList getCategoryListInCategory(String categoryID) throws OZCPEException**
- **public String getCategoryIDOfItem(String itemID) throws OZCPEException**
- **public IOZCategoryInfo getCategoryInfo(String categoryID) throws OZCPEException**
- **public IOZItemInfoList getDeletedItemListInCategory(String categoryID) throws OZCPEException**
- **public boolean transferCategory(String categoryID, String target\_CategoryID) throws OZCPEException**
- **public boolean transferItem(String[] itemIDs, String target\_CategoryID) throws OZCPEException**
- **public String createUser(String userName, String password, String desc) throws OZCPEException**
- **public String modifyUserName(String userID, String userName) throws OZCPEException**
- **public boolean modifyUserPassword(String userID, String old\_password, String new\_password) throws OZCPEException**
- **public boolean modifyUserDesc(String userID, String desc) throws OZCPEException**
- **public boolean deleteUser(String userID) throws OZCPEException**
- **public IOZUserInfo getUserInfo(String userID) throws OZCPEException**
- **public boolean checkUserPassword(String userID, String password) throws OZCPEException**
- **public IOZUserInfoList getUserInfoList() throws OZCPEException**
- **public boolean isCheckAdmin(String userID) throws OZCPEException**
- **public void disableUserLogin(String userID) throws OZCPEException**
- **public void enableUserLogin(String userID) throws OZCPEException**
- **public boolean isLoginEnabled(String userID) throws OZCPEException**

- `public String createGroup(String gName, String pGroupID, String desc)` throws `OZCPEException`
- `public String modifyGroupName(String groupID, String gName)` throws `OZCPEException`
- `public boolean modifyGroupDesc(String groupID, String desc)` throws `OZCPEException`
- `public boolean deleteGroup(String groupID)` throws `OZCPEException`
- `public String createUserInGroup(String userName, String password, String groupID, String desc)` throws `OZCPEException`
- `public boolean transferGroup(String groupID, String target_GroupID)` throws `OZCPEException`
- `public boolean transferUser(String userID, String target_GroupID)` throws `OZCPEException`
- `public IOZUserInfoList getUserInfoListInGroup(String groupID)` throws `OZCPEException`
- `public IOZGroupInfo getGroupInfo(String groupID)` throws `OZCPEException`
- `public IOZGroupInfoList getSubGroupInfoList(String groupID)` throws `OZCPEException`
- `public IOZGroupInfo getParentGroupInfo(String groupID)` throws `OZCPEException`
- `public String getGroupIdOfUser(String userID)` throws `OZCPEException`
- `public boolean addGroupAdmin(String userID, String groupID)` throws `OZCPEException`
- `public boolean removeGroupAdmin(String userID, String groupID)` throws `OZCPEException`
- `public boolean isUserGroupAdmin(String userID, String groupID)` throws `OZCPEException`
- `public IOZUserInfoList getUserAdminListInGroup(String groupID)` throws `OZCPEException`
- `public InputStream distributeRepository(String categoryID, boolean isRecursive)` throws `OZCPEException`
- `public boolean uploadZipItem(String categoryID, InputStream stream)` throws `OZCPEException`
- `public boolean modifyCategoryGroupAuth(String categoryID, String groupID, int perm)` throws `OZCPEException`

- `public int getGroupAuthToCategory(String categoryID, String groupID) throws OZCPEException`
- `public IOZGroupInfoList getGroupListAuthToCategory(String categoryID, int perm) throws OZCPEException`
- `public IOZItemInfoList getItemListAuthToGroupInCategory(String groupID, String categoryID, int perm) throws OZCPEException`
- `public IOZCategoryInfoList getCategoryListAuthToGroupInCategory(String groupID, String categoryID, int perm) throws OZCPEException`
- `public boolean modifyItemGroupAuth(String groupID, String itemID, int perm) throws OZCPEException`
- `public int getGroupAuthToItem(String groupID, String itemID) throws OZCPEException`
- `public IOZGroupInfoList getGroupListAuthToItem(String itemID, int perm) throws OZCPEException`
- `public IOZItemInfoList getItemListAuthToGroup(String groupID, int perm) throws OZCPEException`
- `public boolean modifyCategoryUserAuth(String userID, String categoryID, int perm) throws OZCPEException`
- `public int getUserAuthToCategory(String userID, String categoryID) throws OZCPEException`
- `public IOZUserInfoList getUserListAuthToCategory(String categoryID, int perm) throws OZCPEException`
- `public IOZItemInfoList getItemListAuthToUserInCategory(String userID, String categoryID, int perm) throws OZCPEException`
- `public IOZCategoryInfoList getCategoryListAuthToUserInCategory(String userID, String categoryID, int perm) throws OZCPEException`
- `public boolean modifyItemUserAuth(String userID, String itemID, int perm) throws OZCPEException`
- `public int getUserAuthToItem(String userID, String itemID) throws OZCPEException`
- `public IOZUserInfoList getUserListAuthToItem(String itemID, int perm) throws OZCPEException`
- `public IOZItemInfoList getItemListAuthToUser(String userID, int perm) throws OZCPEException`

## Constructor Detail

### ■ RepositoryEx

	<b>//Daemon</b>	-	<b>TCP Server</b>
	public RepositoryEx(String ip, int port, String id, String pw) throws OZCPEXception		
	<b>// Daemon</b>	-	<b>TCP Server ( )</b>
	public RepositoryEx(String ip, int port, String id, String pw, boolean useUSL) throws OZCPEXception		
Prototype	<b>//Servlet</b>	-	<b>HTTP Server</b>
	public RepositoryEx(String iurl, String id, String pw) throws OZCPEXception		
	<b>//Servlet</b>	-	<b>HTTP Server ( )</b>
	public RepositoryEx(String iurl, String id, String pw, boolean useUSL) throws OZCPEXception		
	<i>ip</i>	Servlet	URL
		ex) String url = "http://127.0.0.1/oz/server";	
	<i>port</i>	Daemon	IP
		ex) String ip = "127.0.0.1";	
Argument	<i>id</i>	Daemon	
		ex) int port = 8003;	
	<i>pw</i>		
		ex) String id = "admin";	
	<i>useUSL</i>	USL	
		ex) boolean useUSL = false;	

## Method Detail

### ■ setRepositoryConfig

Prototype	public void setRepositoryConfig(SortProperties prop) throws OZCPEXception
Definition	.
Argument	<i>SortProperties</i>

### ■ getRepositoryConfig

<b>Prototype</b>	<code>public SortProperties getRepositoryConfig() throws OZCPEException</code>
------------------	--

<b>Definition</b>	가 .
-------------------	-----

■ **reloadRepository**

<b>Prototype</b>	<code>public void reloadRepository() throws OZCPEException</code>
------------------	---

<b>Definition</b>	.
-------------------	---

■ **createItem**

<b>Prototype</b>	<code>public String[] createItem(String[] itemNames, String[] iDescs, String[] categoryIDs, boolean[] isComp, InputStream[] itemStreams, String comment, int[] errorCodes, String[] errorMsg) throws OZCPEException</code>
------------------	--

<b>Definition</b>	ID .
-------------------	------

<b>Argument</b>	<i>itemNames</i>
-----------------	------------------

<b>Argument</b>	<i>descs</i>
-----------------	--------------

<b>Argument</b>	<i>categoryIDs</i>
-----------------	--------------------

<b>Argument</b>	<i>isComp</i>
-----------------	---------------

<b>Argument</b>	<i>item_ins</i>
-----------------	-----------------

<b>Argument</b>	<i>comment</i>
-----------------	----------------

<b>Argument</b>	<i>errCodes</i>
-----------------	-----------------

<b>Argument</b>	<i>errMsgs</i>
-----------------	----------------

■ **modifyItemName**

<b>Prototype</b>	<code>public String modifyItemName(String itemID, String itemName, String comment) throws OZCPEException</code>
------------------	---

<b>Definition</b>	ID .
-------------------	------

<b>Argument</b>	<i>itemID</i> ID
-----------------	------------------

<b>Argument</b>	<i>new_itemName</i>
-----------------	---------------------

<b>Argument</b>	<i>comment</i>
-----------------	----------------

■ **deleteItem**

<b>Prototype</b>	<code>public boolean[] deleteItem(String[] itemIDs, boolean[] isDestroys, String comment, int[] errorCodes, String[] errorMsg) throws OZCPEException</code>
------------------	---

<b>Definition</b>	.	가
	<i>itemIDs</i>	ID
	<i>isDestroys</i>	
<b>Argument</b>	<i>comment</i>	
	<i>errCodes</i>	
	<i>errMsgs</i>	

■ **unDeleteItem**

<b>Prototype</b>	public boolean unDeleteItem(String[] itemIDs, String comment, int[] errorCodes, String[] errorMsg) throws OZCPEXception	
<b>Definition</b>	:	"isDestroys=false" 가
	<i>itemIDs</i>	ID
<b>Argument</b>	<i>comment</i>	
	<i>errCodes</i>	
	<i>errMsgs</i>	

■ **modifyItemDesc**

<b>Prototype</b>	public boolean modifyItemDesc(String itemID, String iDesc) throws OZCPEXception	
<b>Definition</b>	ID	.
<b>Argument</b>	<i>itemID</i>	ID
	<i>desc</i>	

■ **getItemInfo**

<b>Prototype</b>	public IOZItemInfo getItemInfo(String itemID) throws OZCPEXception	
<b>Definition</b>	ID	가
<b>Argument</b>	<i>itemID</i>	가 ID

■ **hasTheItem**

<b>Prototype</b>	public boolean hasTheItem(String itemID) throws	
------------------	---	--

	OZCPException
<b>Definition</b>	.
<b>Argument</b>	<i>itemID</i>

■ **getItemsUnCondition**

<b>Prototype</b>	public InputStream[] getItemsUnCondition(String[] itemIDs, boolean[] isComp, boolean[] isObjStream, String[] categoryIDs, int[] errorCodes, String[] errorMsg) throws OZCPException
<b>Definition</b>	ID                    가                    .
	<i>itemIDs</i> 가                    ID
	<i>isComp</i>
<b>Argument</b>	<i>isObjStream</i> ODI
	<i>errCodes</i>
	<i>errMsgs</i>

■ **getItems**

<b>Prototype</b>	public InputStream[] getItems(String[] itemIDs, long[] modifiedTimes, boolean[] isComp, boolean[] isObjStream, String[] categoryIDs, int[] errorCodes, String[] errorMsg) throws OZCPException
<b>Definition</b>	ID                    가                    .
	<i>itemIDs</i> 가                    ID
	<i>modifiedTimes</i>
	<i>isComp</i>
<b>Argument</b>	<i>isObjStream</i> ODI
	<i>errCodes</i>
	<i>errMsgs</i>

■ **checkoutItem**

<b>Prototype</b>	public InputStream[] checkoutItem(String itemIDs[], String[] localCheckoutFolders, long[] localFileTimes, boolean[] isComp, int[] errorCodes, String[] errorMsg) throws OZCPException
<b>Definition</b>	.
<b>Argument</b>	<i>itemIDs</i> ID

<i>localCheckout</i>
<i>Folders</i>
<i>localFileTimes</i>
<i>isComp</i>
<i>errCodes</i>
<i>errMsgs</i>

■ **checkInItem**

<b>Prototype</b>	public boolean[] checkInItem(String[] itemIds, boolean[] isComp, InputStream[] iStreams, String comment, boolean[] keepCheckout, int[] errorCodes, String[] errorMsg) throws OZCPEXception
<b>Definition</b>	.
	<i>itemIds</i> ID
	<i>isComp</i>
	<i>itemStreams</i>
<b>Argument</b>	<i>comment</i>
	<i>keepCheckout</i>
	<i>errCodes</i>
	<i>errMsgs</i>

■ **undoCheckoutItem**

<b>Prototype</b>	public InputStream[] undoCheckoutItem(String[] itemIds, boolean[] isReplaces, boolean[] isComp, int[] errorCodes, String[] errorMsg) throws OZCPEXception
<b>Definition</b>	.
	<i>itemIds</i> ID
	<i>isReplaces</i> 가
<b>Argument</b>	<i>errCodes</i>
	<i>errMsgs</i>

■ **isCheckoutUser**

<b>Prototype</b>	public boolean[] isCheckoutUser(String[] itemIds) throws OZCPEXception
<b>Definition</b>	가
<b>Argument</b>	<i>itemIds</i> ID

■ **rollbackItem**

<b>Prototype</b>	public boolean rollbackItem(String itemID, int itemVersion, String comment) throws OZCPEXception
<b>Definition</b>	
	<i>itemID</i> ID
<b>Argument</b>	<i>itemVersion</i> <i>comment</i>

■ **getItemByVersion**

<b>Prototype</b>	public InputStream getItemByVersion(String itemID, int version, boolean isComp) throws OZCPEXception
<b>Definition</b>	ID 가
	<i>itemID</i> 가 ID
<b>Argument</b>	<i>version</i> 가 <i>isComp</i>

■ **getHistoryItemList**

<b>Prototype</b>	public IOZHistoryInfoList getHistoryItemList(String itemID) throws OZCPEXception
<b>Definition</b>	가
<b>Argument</b>	<i>itemID</i> 가 ID

■ **getDeleteHistoryItemInfo**

<b>Prototype</b>	public IOZHistoryInfoList getDeleteHistoryItemInfo(String itemID) throws OZCPEXception
<b>Definition</b>	가 : "isDestroys=false" 가
<b>Argument</b>	<i>itemID</i> 가 ID

■ **removeHistoryItem**

<b>Prototype</b>	public boolean removeHistoryItem(String itemID, int itemVersion) throws OZCPEXception
<b>Definition</b>	
<b>Argument</b>	<i>itemID</i> ID

*itemversion*

---

■ **createCategory**

**Prototype**     public String[] createCategory(String[] categoryNames,  
String[] pCategoryIDs, String comment, int[] errorCodes,  
String[] errorMsg) throws OZCPEXception

**Definition**     ID     .

*categoryName*

*pCategoryIDs*     ID

**Argument**     *comment*

*errorCode*

*errorMsg*

---

■ **modifyCategoryName**

**Prototype**     public String modifyCategoryName(String categoryID, String  
new\_CategoryName, String comment) throws OZCPEXception

**Definition**     ID     .

*categoryID*     ID

**Argument**     *new\_categoryName*

*comment*

---

■ **deleteCategory**

**Prototype**     public boolean[] deleteCategory(String[] categoryIDs,  
boolean[] isDestroys, String comment, int[] errorCodes,  
String[] errorMsg) throws OZCPEXception

**Definition**     ID     .

*categoryIDs*     ID

*isDestroys*

**Argument**     *comment*

*errorCode*

*errorMsg*

---

■ **unDeleteCategory**

**Prototype**     public boolean[] unDeleteCategory(String[] categoryIDs,  
String comment, int[] errorCodes, String[] errorMsg) throws  
OZCPEXception

---

<b>Definition</b>	:	가 "isDestroys=false" 가 가
<b>Argument</b>		<i>categoryIDs</i> ID <i>comment</i> <i>errorCode</i> <i>errorMsg</i>

■ **getItemCountInCategory**

<b>Prototype</b>	public int getItemCountInCategory(String categoryId) throws OZCPEXception	
<b>Definition</b>		가
<b>Argument</b>	<i>categoryId</i>	ID

■ **getItemListInCategory**

<b>Prototype</b>	public IOZItemInfoList getItemListInCategory(String categoryId) throws OZCPEXception	
<b>Definition</b>		가
<b>Argument</b>	<i>categoryId</i>	ID

■ **getCategoryListInCategory**

<b>Prototype</b>	public IOZCategoryInfoList getCategoryListInCategory(String categoryId) throws OZCPEXception	
<b>Definition</b>		가
<b>Argument</b>	<i>categoryId</i>	ID

■ **getCategoryIdOfItem**

<b>Prototype</b>	public String getCategoryIdOfItem(String itemId) throws OZCPEXception	
<b>Definition</b>		ID
<b>Argument</b>	<i>itemId</i>	ID ID

■ **getCategoryInfo**

<b>Prototype</b>	public IOZCategoryInfo getCategoryInfo(String categoryID) throws OZCPEXception
<b>Definition</b>	가
<b>Argument</b>	<i>categoryID</i> 가      ID

■ **getDeletedItemListInCategory**

<b>Prototype</b>	public IOZItemInfoList getDeletedItemListInCategory(String categoryID) throws OZCPEXception
<b>Definition</b>	가 : "isDestroys=false"      가
<b>Argument</b>	<i>categoryID</i> 가      ID

■ **transferCategory**

<b>Prototype</b>	public boolean transferCategory(String categoryID, String target_CategoryID) throws OZCPEXception
<b>Definition</b>	가
<b>Argument</b>	<i>categoryID</i> ID <i>target_CategoryID</i> ID

■ **transferItem**

<b>Prototype</b>	public boolean transferItem(String[] itemIDs, String target_CategoryID) throws OZCPEXception
<b>Definition</b>	가
<b>Argument</b>	<i>itemIDs</i> ID <i>target_CategoryID</i> ID

■ **createUser**

<b>Prototype</b>	public String createUser(String userName, String password, String desc) throws OZCPEXception
<b>Definition</b>	ID
<b>Argument</b>	<i>userName</i> <i>password</i> <i>desc</i>

■ **modifyUserName**

<b>Prototype</b>	public String modifyUserName(String userID, String userName) throws OZCPEXception	
<b>Definition</b>	ID	
<b>Argument</b>	<i>userID</i>	ID
	<i>userName</i>	

■ **modifyUserPassword**

<b>Prototype</b>	public boolean modifyUserPassword(String userID, String old_password, String new_password) throws OZCPEXception	
<b>Definition</b>	ID	
<b>Argument</b>	<i>userID</i>	ID
	<i>old_password</i>	
	<i>new_password</i>	

■ **modifyUserDesc**

<b>Prototype</b>	public boolean modifyUserDesc(String userID, String desc) throws OZCPEXception	
<b>Definition</b>	ID	
<b>Argument</b>	<i>userID</i>	ID
	<i>desc</i>	

■ **deleteUser**

<b>Prototype</b>	public boolean deleteUser(String userID) throws OZCPEXception	
<b>Definition</b>	ID	
<b>Argument</b>	<i>userID</i>	ID

■ **getUserInfo**

<b>Prototype</b>	public IOZUserInfo getUserInfo(String userID) throws OZCPEXception	
<b>Definition</b>	ID	가
<b>Argument</b>	<i>userID</i>	가 ID

■ **checkUserPassword**

Prototype	public boolean checkUserPassword(String userID, String password) throws OZCPEXception
Definition	가
Argument	<i>userID</i> ID
	<i>password</i>

■ **getUserInfoList**

Prototype	public IOZUserInfoList getUserInfoList() throws OZCPEXception
Definition	가

■ **DisableLogin**

Prototype	public void DisableLogin(String userID)
Definition	ID
Argument	<i>userID</i> ID

■ **EnableLogin**

Prototype	public void EnableLogin(String userID)
Definition	ID 가
Argument	<i>userID</i> 가 ID

■ **isCheckAdmin**

Prototype	public boolean isCheckAdmin(String userID) throws OZCPEXception
Definition	ID 가
Argument	<i>userID</i> ID

■ **disableUserLogin**

Prototype	public void disableUserLogin(String userID) throws OZCPEXception
Definition	ID
Argument	<i>userID</i> ID

■ **enableUserLogin**

<b>Prototype</b>	public void enableUserLogin(String userID) throws OZCPEXception		
<b>Definition</b>	ID	가	.
<b>Argument</b>	<i>userID</i>	가	ID

■ **isLoginEnabled**

<b>Prototype</b>	public boolean isLoginEnabled(String userID) throws OZCPEXception			
<b>Definition</b>	ID	가	가	가
<b>Argument</b>	<i>userID</i>	가	가	ID

■ **createGroup**

<b>Prototype</b>	public String createGroup(String gName, String pGroupID, String desc) throws OZCPEXception		
<b>Definition</b>	ID	.	
	<i>gName</i>		
<b>Argument</b>	<i>pGroupID</i>	ID	
	<i>desc</i>		

■ **modifyGroupName**

<b>Prototype</b>	public String modifyGroupName(String groupID, String gName) throws OZCPEXception	
<b>Definition</b>	ID	.
	<i>groupID</i>	ID
<b>Argument</b>	<i>gName</i>	

■ **modifyGroupDesc**

<b>Prototype</b>	public boolean modifyGroupDesc(String groupID, String desc) throws OZCPEXception	
<b>Definition</b>	ID	.
	<i>groupID</i>	ID
<b>Argument</b>	<i>desc</i>	

■ **deleteGroup**

Prototype	public boolean deleteGroup(String groupId) throws OZCPEXception
Definition	ID
Argument	<i>groupId</i> ID

■ **createUserInGroup**

Prototype	public String createUserInGroup(String userName, String password, String groupId, String desc) throws OZCPEXception
Definition	ID
Argument	<i>userName</i> <i>password</i> <i>groupId</i> ID <i>desc</i>

■ **transferUser**

Prototype	public boolean transferUser(String userID, String target_groupID) throws OZCPEXception
Definition	가
Argument	<i>userID</i> ID <i>target_groupID</i> ID

■ **transferGroup**

Prototype	public boolean transferGroup(String groupId, String target_groupID) throws OZCPEXception
Definition	가
Argument	<i>groupId</i> ID <i>target_groupID</i> ID

■ **getUserInfoListInGroup**

Prototype	public IOZUserInfoList getUserInfoListInGroup(String groupId) throws OZCPEXception
Definition	ID 가
Argument	<i>groupId</i> 가 ID

■ **getGroupInfo**

Prototype	public IOZGroupInfo getGroupInfo(String groupId) throws OZCPEXception
Definition	ID 가 .
Argument	groupId 가 ID

■ **getSubGroupInfoList**

Prototype	public IOZGroupInfoList getSubGroupInfoList(String groupId) throws OZCPEXception
Definition	가 .
Argument	groupId 가 ID

■ **getParentGroupInfo**

Prototype	public IOZGroupInfo getParentGroupInfo(String groupId) throws OZCPEXception
Definition	가 .
Argument	groupId 가 ID

■ **getGroupIdOfUser**

Prototype	public String getGroupIdOfUser(String userID) throws OZCPEXception
Definition	가 가 .
Argument	userID 가 ID

■ **addGroupAdmin**

Prototype	public boolean addGroupAdmin(Object sessonID, String userID, String groupId) throws OZRepositoryException
Definition	가 가 .
Argument	userID 가 ID
Argument	groupId 가 ID

■ **removeGroupAdmin**

Prototype	public boolean removeGroupAdmin(String userID, String groupId) throws OZCPEXception
Definition	.

Argument	<i>userID</i>	ID
	<i>groupID</i>	ID

■ **isUserGroupAdmin**

Prototype	public boolean isUserGroupAdmin(String userID, String groupID) throws OZCPEXception	
Definition	ID	가
Argument	<i>userID</i>	ID
	<i>groupID</i>	ID

■ **getUserAdminListInGroup**

Prototype	public IOZUserInfoList getUserAdminListInGroup(String groupID) throws OZCPEXception	
Definition	가	.
Argument	<i>groupID</i>	가 ID

■ **distributeRepository**

Prototype	public InputStream distributeRepository(String categoryID, boolean isRecursive) throws OZCPEXception	
Definition	.	
Argument	<i>categoryID</i>	ID
	<i>isRecursive</i>	

■ **uploadZipItem**

Prototype	public boolean uploadZipItem(String categoryID, InputStream stream) throws OZCPEXception	
Definition	.	
Argument	<i>categoryID</i>	ID
	<i>Stream</i>	

■ **modifyCategoryGroupAuth**

<b>Prototype</b>	public boolean modifyCategoryGroupAuth(String categoryID, String groupID, int perm) throws OZCPEXception						
<b>Definition</b>	가						
<b>Argument</b>	<table border="1"> <tr> <td><i>categoryID</i></td> <td>ID</td> </tr> <tr> <td><i>groupID</i></td> <td>ID</td> </tr> <tr> <td><i>perm</i></td> <td></td> </tr> </table>	<i>categoryID</i>	ID	<i>groupID</i>	ID	<i>perm</i>	
<i>categoryID</i>	ID						
<i>groupID</i>	ID						
<i>perm</i>							

■ **getGroupAuthToCategory**

<b>Prototype</b>	public int getGroupAuthToCategory(String categoryID, String groupID) throws OZCPEXception						
<b>Definition</b>	가						
<b>Argument</b>	<table border="1"> <tr> <td><i>categoryID</i></td> <td>가</td> <td>ID</td> </tr> <tr> <td><i>groupID</i></td> <td>ID</td> <td></td> </tr> </table>	<i>categoryID</i>	가	ID	<i>groupID</i>	ID	
<i>categoryID</i>	가	ID					
<i>groupID</i>	ID						

■ **getGroupListAuthToCategory**

<b>Prototype</b>	public IOZGroupInfoList getGroupListAuthToCategory(String categoryID, int perm) throws OZCPEXception						
<b>Definition</b>	가 ID perm 가						
<b>Argument</b>	<table border="1"> <tr> <td><i>categoryID</i></td> <td>가</td> <td>ID</td> </tr> <tr> <td><i>perm</i></td> <td>ID</td> <td></td> </tr> </table>	<i>categoryID</i>	가	ID	<i>perm</i>	ID	
<i>categoryID</i>	가	ID					
<i>perm</i>	ID						

■ **getItemListAuthToGroupInCategory**

<b>Prototype</b>	public IOZItemInfoList getItemListAuthToGroupInCategory(String groupID, String categoryID, int perm) throws OZCPEXception						
<b>Definition</b>	가 ID perm 가						
<b>Argument</b>	<table border="1"> <tr> <td><i>groupID</i></td> <td>ID</td> </tr> <tr> <td><i>categoryID</i></td> <td>ID</td> </tr> <tr> <td><i>perm</i></td> <td></td> </tr> </table>	<i>groupID</i>	ID	<i>categoryID</i>	ID	<i>perm</i>	
<i>groupID</i>	ID						
<i>categoryID</i>	ID						
<i>perm</i>							

■ **getCategoryListAuthToGroupInCategory**

<b>Prototype</b>	public IOZCategoryInfoList getCategoryListAuthToGroupInCategory(String groupId, String categoryId, int perm) throws OZCPEXception		
<b>Definition</b>	ID	perm	가
<b>Argument</b>	<i>groupId</i>	ID	
	<i>categoryId</i>	ID	
	<i>perm</i>		

■ **modifyItemGroupAuth**

<b>Prototype</b>	public boolean modifyItemGroupAuth(String groupId, String itemId, int perm) throws OZCPEXception		
<b>Definition</b>	ID	ID	
<b>Argument</b>	<i>groupId</i>	ID	
	<i>itemId</i>	ID	
	<i>perm</i>		

■ **getGroupAuthToItem**

<b>Prototype</b>	public int getGroupAuthToItem(String groupId, String itemId) throws OZCPEXception		
<b>Definition</b>	ID	ID	가
<b>Argument</b>	<i>groupId</i>	ID	
	<i>itemId</i>	ID	

■ **getGroupListAuthToItem**

<b>Prototype</b>	public IOZGroupInfoList getGroupListAuthToItem(String itemId, int perm) throws OZCPEXception		
<b>Definition</b>	ID	perm	가
<b>Argument</b>	<i>itemId</i>	ID	
	<i>perm</i>		

■ **getItemListAuthToGroup**

<b>Prototype</b>	public IOZItemInfoList getItemListAuthToGroup(String groupID, int perm) throws OZCPEXception				
<b>Definition</b>	<table border="0"> <tr> <td style="text-align: center;">ID</td> <td style="text-align: center;">perm</td> </tr> <tr> <td style="text-align: center;">가</td> <td style="text-align: center;">.</td> </tr> </table>	ID	perm	가	.
ID	perm				
가	.				
<b>Argument</b>	<table border="0"> <tr> <td style="text-align: center;"><i>groupID</i></td> <td style="text-align: center;">ID</td> </tr> <tr> <td style="text-align: center;"><i>perm</i></td> <td></td> </tr> </table>	<i>groupID</i>	ID	<i>perm</i>	
<i>groupID</i>	ID				
<i>perm</i>					

■ **modifyCategoryUserAuth**

<b>Prototype</b>	public boolean modifyCategoryUserAuth(String userID, String cID, int perm) throws OZCPEXception						
<b>Definition</b>	<table border="0"> <tr> <td style="text-align: center;">ID</td> <td style="text-align: center;">ID</td> </tr> <tr> <td style="text-align: center;">.</td> <td></td> </tr> </table>	ID	ID	.			
ID	ID						
.							
<b>Argument</b>	<table border="0"> <tr> <td style="text-align: center;"><i>userID</i></td> <td style="text-align: center;">ID</td> </tr> <tr> <td style="text-align: center;"><i>categoryID</i></td> <td style="text-align: center;">ID</td> </tr> <tr> <td style="text-align: center;"><i>perm</i></td> <td></td> </tr> </table>	<i>userID</i>	ID	<i>categoryID</i>	ID	<i>perm</i>	
<i>userID</i>	ID						
<i>categoryID</i>	ID						
<i>perm</i>							

■ **getUserAuthToCategory**

<b>Prototype</b>	public int getUserAuthToCategory(String userID, String categoryID) throws OZCPEXception				
<b>Definition</b>	<table border="0"> <tr> <td style="text-align: center;">ID</td> <td style="text-align: center;">ID</td> <td style="text-align: center;">가</td> <td style="text-align: center;">.</td> </tr> </table>	ID	ID	가	.
ID	ID	가	.		
<b>Argument</b>	<table border="0"> <tr> <td style="text-align: center;"><i>userID</i></td> <td style="text-align: center;">ID</td> </tr> <tr> <td style="text-align: center;"><i>categoryID</i></td> <td style="text-align: center;">ID</td> </tr> </table>	<i>userID</i>	ID	<i>categoryID</i>	ID
<i>userID</i>	ID				
<i>categoryID</i>	ID				

■ **getUserListAuthToCategory**

<b>Prototype</b>	public IOZUserInfoList getUserListAuthToCategory(String categoryID, int perm) throws OZCPEXception				
<b>Definition</b>	<table border="0"> <tr> <td style="text-align: center;">ID</td> <td style="text-align: center;">perm</td> </tr> <tr> <td style="text-align: center;">가</td> <td style="text-align: center;">.</td> </tr> </table>	ID	perm	가	.
ID	perm				
가	.				
<b>Argument</b>	<table border="0"> <tr> <td style="text-align: center;"><i>categoryID</i></td> <td style="text-align: center;">ID</td> </tr> <tr> <td style="text-align: center;"><i>perm</i></td> <td></td> </tr> </table>	<i>categoryID</i>	ID	<i>perm</i>	
<i>categoryID</i>	ID				
<i>perm</i>					

■ **getItemListAuthToUserInCategory**

<b>Prototype</b>	public IOZItemInfoList getItemListAuthToUserInCategory(Object sessionID, String userID, String categoryID, int perm) throws OZRepositoryException
------------------	---

Definition	가	perm
Argument	<i>userID</i>	ID
Argument	<i>categoryID</i>	ID
Argument	<i>perm</i>	

■ getCategoryListAuthToUserInCategory

Prototype	public IOZCategoryInfoList getCategoryListAuthToUserInCategory(String userID, String categoryID, int perm) throws OZCPEException	
Definition	가	perm
Argument	<i>userID</i>	ID
Argument	<i>categoryID</i>	ID
Argument	<i>perm</i>	

■ modifyItemUserAuth

Prototype	public boolean modifyItemUserAuth(String userID, String itemID, int perm) throws OZCPEException	
Definition	가	
Argument	<i>userID</i>	ID
Argument	<i>itemID</i>	ID
Argument	<i>perm</i>	

■ getUserAuthToItem

Prototype	public int getUserAuthToItem(String userID, String itemID) throws OZCPEException	
Definition	가	
Argument	<i>userID</i>	ID
Argument	<i>itemID</i>	ID
Argument	<i>perm</i>	

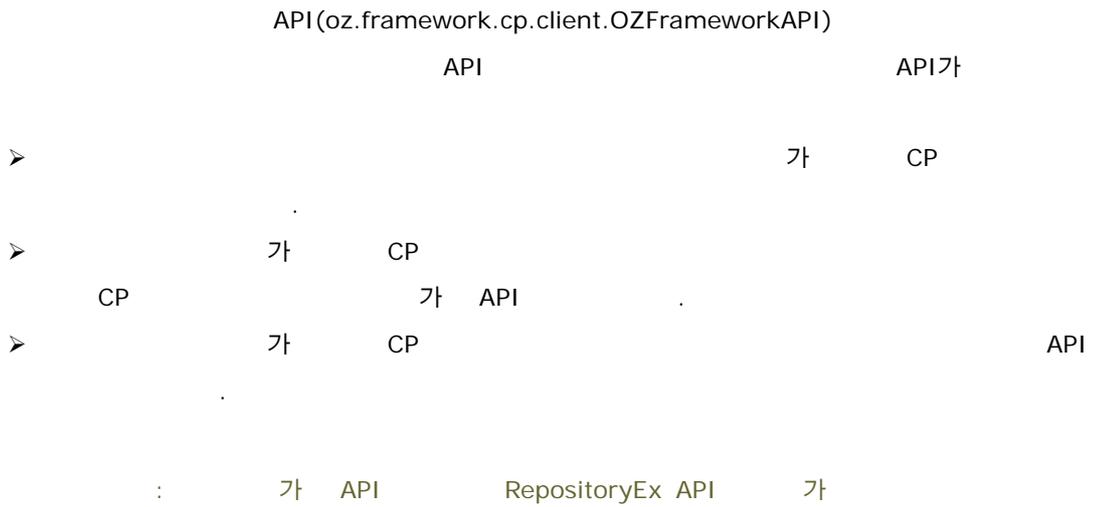
■ getUserListAuthToItem

Prototype	public IOZUserInfoList getUserListAuthToItem(String itemID, int perm) throws OZCPEException	
-----------	---	--

<b>Definition</b>	.	perm	가
<b>Argument</b>	<i>itemID</i>	ID	
	<i>perm</i>		

■ **getItemListAuthToUser**

<b>Prototype</b>	public IOZItemInfoList getItemListAuthToUser(String userID, int perm) throws OZCPEException		
<b>Definition</b>	.	perm	가
<b>Argument</b>	<i>userID</i>	ID	
	<i>perm</i>		



- Adapter 가 IOZRepository
- Adapter 가 Category Item Group  
User
- 

**Sample : RepositoryExSample.java**

```

package sample;

import oz.framework.api.RepositoryEx;
  
```

```
import oz.util.SortProperties;
import oz.framework.repositoryex.info.*;
import oz.framework.repositoryex.adapter.impl.*;
import java.io.*;

public class RepositoryExSample {

    public RepositoryExSample() {

    }

    private static RepositoryEx rep = null;
    public static void main(String[] args) {
        String IP = "127.0.0.1"; // 가 IP
        int PORT = 8003; // 가 TCP
        String ID = "admin"; //default
        String PASSWORD = "admin"; //default
        try {
            rep = new RepositoryEx(IP, PORT, ID, PASSWORD, /*usi*/ false);
            //getConf();
            //setConf();
            //reloadConf();
            itemTest();
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }

    private static void getConf() throws Exception {
        //Repository configuration 가 .
        System.out.println("Repository.getRepositoryConfig()");
        SortProperties props = rep.getRepositoryConfig();
        props.list(System.out);
    }

    private static void setConf() throws Exception {
        //Repository Configuration
        SortProperties prop = new SortProperties();
        prop.setProperty("REPOSITORY_TYPE", "RDB"); //RDB, FILESYSTEM, USER
        prop.setProperty("REPOSITORY_FILE_PATH", "c:/temp_repository");
        prop.setProperty("REPOSITORY_ITEM_NUMBER_PER_DIRECTORY", "100");
        prop.setProperty("REPOSITORY_HISTORY_ITEM_VALID_DAYS", "20");
        rep.setRepositoryConfig(prop);
    }

    private static void reloadConf() throws Exception {
        rep.reloadRepository();
    }
}
```

```
}

private static void itemTest() throws Exception {
    String[] categoryNames = {
        "182",
        "183",
        "184",
        "185",
        "186"
    };
    String[] parentCategoryNames = {
        "/REQ",
        "/REQ1",
        "/REQ2",
        "/REQ3",
        "/REQ4"
    };
    int[] errCodes = new int[categoryNames.length];
    String[] errMsgs = new String[categoryNames.length];
    //
    String[] categoryIDs = rep.createCategory(categoryNames,
                                             parentCategoryNames,
                                             "comment", errCodes, errMsgs);
    // ID
    for (int i = 0; i < categoryIDs.length; i++) {
        if (categoryIDs[i] != null) {
            System.out.println("categoryID=" + categoryIDs[i]);
        }
        else {
            System.out.println("error=[code=" + errCodes[i] + "] " +
                               errMsgs[i]);
        }
    }
}
//Item
String[] itemNames = {
    "car.odi",
    "car.ozr",
    "car.oza",
    "cap.txt"
};
String[] itemDesc = {
    "odi",
    "ozr",
    "oza",
    "0 byte text"
};
FileInputStream[] fis = new FileInputStream[itemNames.length];
fis[0] = new FileInputStream("d:/REQ/182/car.odi");
```

```
fis[1] = new FileInputStream("d:/REQ/182/car.ozr");
fis[2] = new FileInputStream("d:/REQ/182/car.oza");
fis[3] = new FileInputStream("d:/cap.txt");
errCodes = new int[itemNames.length];
errMsgs = new String[itemNames.length];
String[] itemIDs = rep.createItem(itemNames, itemDesc, categoryIDs,
                                new boolean[] {false, false, false, false}
                                ,
                                //
                                fis, "comment", errCodes, errMsgs);

//ItemID ID
for (int i = 0; i < itemIDs.length; i++) {
    if (itemIDs[i] != null) {
        System.out.println("itemID=" + itemIDs[i]);
    }
    else {
        System.out.println("error=[code=" + errCodes[i] + "] " +
                            errMsgs[i]);
    }
}
errCodes = new int[itemNames.length];
errMsgs = new String[itemNames.length];

//Item 가
download(rep.getItems(itemIDs, new long[] {0, 0, 0, 0}
                    ,
                    new boolean[] {true, false, false, false}
                    , //
                    new boolean[] {false, false, false, false}
                    , //odi
                    categoryIDs, errCodes, errMsgs),
        new
        String[] {"d:/car.odi", "d:/car.ozr", "d:/car.oza",
                 "d:/cap1.txt"});

//
String newItemID =
    rep.modifyItemName(itemIDs[3],
                      "modify" + System.currentTimeMillis() + ".txt",
                      "comment");
System.out.println("new Item ID =" + newItemID);
itemIDs[3] = newItemID;

//ItemD 가
OZItemInfo info = new OZItemInfo(rep.getItemInfo(newItemID));
System.out.println(info.toString());

// ID ItemList
```

```

OZItemInfoListItemImpl itemList =
    new OZItemInfoListItemImpl (rep.getItemListInCategory("/REQ1/182/"));

//ItemList ItemInfo
OZItemInfoImpl itemInfo = null;
for (int i = 0; i < itemList.getSize(); i++) {
    itemInfo = new OZItemInfoImpl (itemList.getItemInfo(i));
    System.out.println(itemInfo.toString());
}

//
System.out.println(rep.hasTheItem("/REQ1/182/car.odi"));

//
Item checkIn(update)
errCodes = new int[itemDs.length];
errMsgs = new String[itemDs.length];
fis[0] = new FileInputStream("d:/REQ/182/car.odi");
fis[1] = new FileInputStream("d:/REQ/182/car.ozr");
fis[2] = new FileInputStream("d:/REQ/182/car.oza");
fis[3] = new FileInputStream("d:/cap.txt");
boolean[] isResult =
    rep.checkItem(itemDs, new boolean[] {false, false, false, false}
        , fis,
        "comment",
        new boolean[] {false, false, false, false}
        , errCodes,
        errMsgs);
for (int i = 0; i < isResult.length; i++) {
    System.out.println("check in success = " + isResult[i]);
    if (!isResult[i]) {
        System.out.println("error [" + errCodes[i] + "] =" + errMsgs[i]);
    }
}
//
errCodes = new int[itemDs.length];
errMsgs = new String[itemDs.length];
isResult = rep.deleteItem(itemDs, new boolean[] {true, true, true, true}
    ,
    "comment",
    errCodes, errMsgs);
for (int i = 0; i < isResult.length; i++) {
    System.out.println("delete item success = " + isResult[i]);
    if (!isResult[i]) {
        System.out.println("error [" + errCodes[i] + "] =" + errMsgs[i]);
    }
}
//
errCodes = new int[itemDs.length];

```

```
errMsgs = new String[itemsDs.length];
isResult = rep.deleteCategory(categoryIDs, new boolean[] {true, true,
true, true, true}

        ,
        "comment",
        errCodes, errMsgs);
for (int i = 0; i < isResult.length; i++) {
    System.out.println("delete category success = " + isResult[i]);
    if (!isResult[i]) {
        System.out.println("error [" + errCodes[i] + "] =" + errMsgs[i]);
    }
}
//
rep.transferCategory("/REQ1", "/REQ");
//
rep.transferItem(new String[] {"/test1.odi", "/test1.ozr"}
        , "/REQ");
//
rep.transferCategory("/REQ/REQ1", "/REQ2");
//
rep.transferItem(new String[] {"/REQ/test1.odi", "/REQ/test1.ozr"}
        , "/");

IOZCategoryInfo[]          infos          =
rep.getCategoryListInCategory("/", true).getCategoryInfos();
String[] d = new String[infos.length-1];
int index = 0;
for(int i=0; i<d.length; i++) {
    if (!"/".equals(infos[i].getCategoryID())){

System.out.println("infos[i].getCategoryID()="+infos[i].getCategoryID());
        d[index++] = infos[i].getCategoryID();
    }
}
errCodes = new int[d.length];
errMsgs = new String[d.length];
boolean[] b = new boolean[d.length];
java.util.Arrays.fill(b, true);
isResult = rep.deleteCategory(d, b

        ,
        "comment",
        errCodes, errMsgs);
for (int i = 0; i < isResult.length; i++) {
    System.out.println("final delete category success = " + isResult[i]);
    if (!isResult[i]) {
        System.out.println("error [" + errCodes[i] + "] =" + errMsgs[i]);
    }
}
}
```

```
    }

    private static void download(InputStream[] in, String[] fileName) throws
        Exception {
        for (int i = 0; i < in.length; i++) {
            if (in[i] != null) {
                byte[] buf = new byte[1024];
                int len;
                FileOutputStream fos = new FileOutputStream(fileName[i]);
                while ( (len = in[i].read(buf)) >= 0) {
                    fos.write(buf, 0, len);
                }
                in[i].close();
                fos.flush();
                fos.close();
                System.out.println("ok download = " + fileName[i]);
            }
            else {
                System.out.println(fileName[i] + " stream is null");
            }
        }
    }
}
```

## API

 OZLauncherDll



<b>Argument</b>	<i>str_param</i>	" n"
	<i>n_type</i>	2
<b>Example</b>	<pre>CreateOZViewer("connecti on. server=127. 0. 0. 1\n connecti on. port=8003\n connecti on. reportname=/ozsampl e. oZR\n", 2);</pre>	

■ **Release()**

<b>Prototype</b>	<code>void __stdcall Release()</code>
<b>Definition</b>	
<b>Example</b>	<pre>SetCommand("/String"); SetPath("./"); CreateOZViewer("connecti on. server=127. 0. 0. 1\n tool bar. all=true\n i nformati on. debug=debug\n i nformati on. bmt=true\n connecti on. port=8003\n connecti on. reportname=/ozsampl e. oZR\n connecti on. compressedForm=true", 1); resul t = GetResul t(); Rel ease();</pre>

## . RDB Store DataAction

- RDB Store DataAction
- RDB Store DataAction

## RDB Store DataAction

RDB DataAction

```
public interface IRDBDelegate extends IDelegate
{
    public void init(String param) throws Exception;
    public void close();
    public String insert(oz.framework.dac.OZDACItem dac,
        java.util.Dictionary parameters)
        throws Exception;
    public String update(oz.framework.dac.OZDACItem dac,
        java.util.Dictionary parameters)
        throws Exception;
    public String delete(oz.framework.dac.OZDACItem dac,
        java.util.Dictionary parameters)
        throws Exception;
    public String commit() throws Exception;
    public void rollback() throws Exception;

    public void setConnection(java.sql.Connection con);
}
```

1. setConnection()
2. init()
3. insert(), update(), delete()
4. commit(), rollback()
5. close()

## RDB Store DataAction

### DataAction

#### DataAction

```
package com.forcs.sample.dataaction.rdb;

import com.forcs.log4oz.OZLog;

public class DelegateSample
    implements oz.udd.IRDBDelegate, oz.udd.ILoggerRef {

    private OZLog cat;
    private java.sql.Connection con;

    public DelegateSample() {
    }

    public void init(String param) throws Exception {
        //
    }

    public void close() {
        //close
    }

    public String insert(oz.framework.dac.OZDACItem dac,
        java.util.Dictionary parameters) throws Exception {
        //insert
        return null;
    }

    public String update(oz.framework.dac.OZDACItem dac,
        java.util.Dictionary parameters) throws Exception {
        //Update
        return null;
    }

    public String delete(oz.framework.dac.OZDACItem dac,
        java.util.Dictionary parameters) throws Exception {
        //Delete
        return null;
    }
}
```

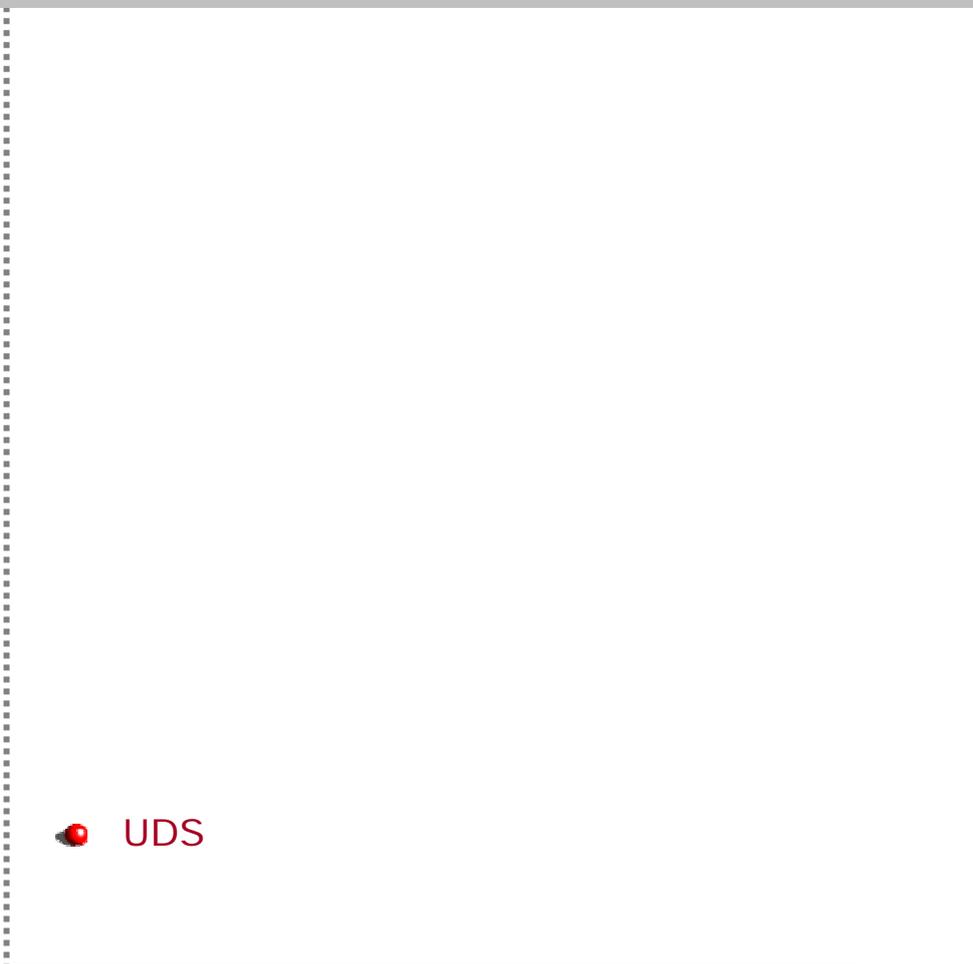
```
public String commit() throws Exception {
    //commit
    return null;
}

public void rollback() throws Exception {
    //rollback
}

public void setConnection(java.sql.Connection con) {
    this.con = con;
}

public void setLogger(OZLog cat) {
    this.cat = cat;
}
}
```

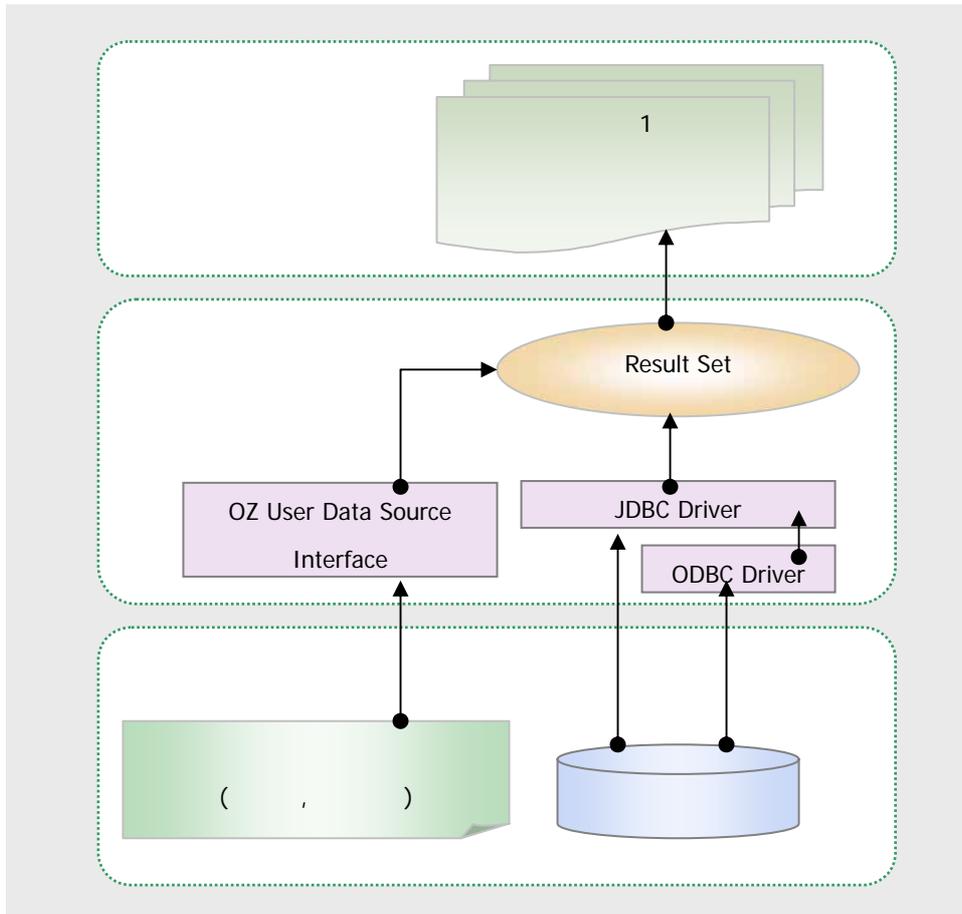
## . User Data Store



UDS

# UDS

UDS(User Data Store) 가 Non-DB Java Interface  
 , CSV, XML EJB,  
 Servlet, ASP, JSP, Applet



UDS  
 가 ResultSet  
 . UDS  
 SQL  
 : UDS

## . User Security Logic

● USL

● USL

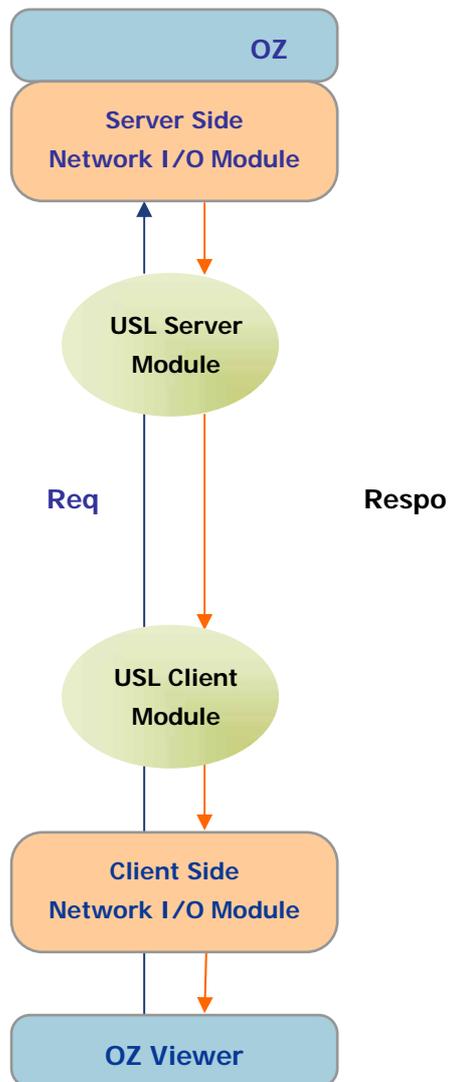
● USL

## USL

USL(User Security Logic)

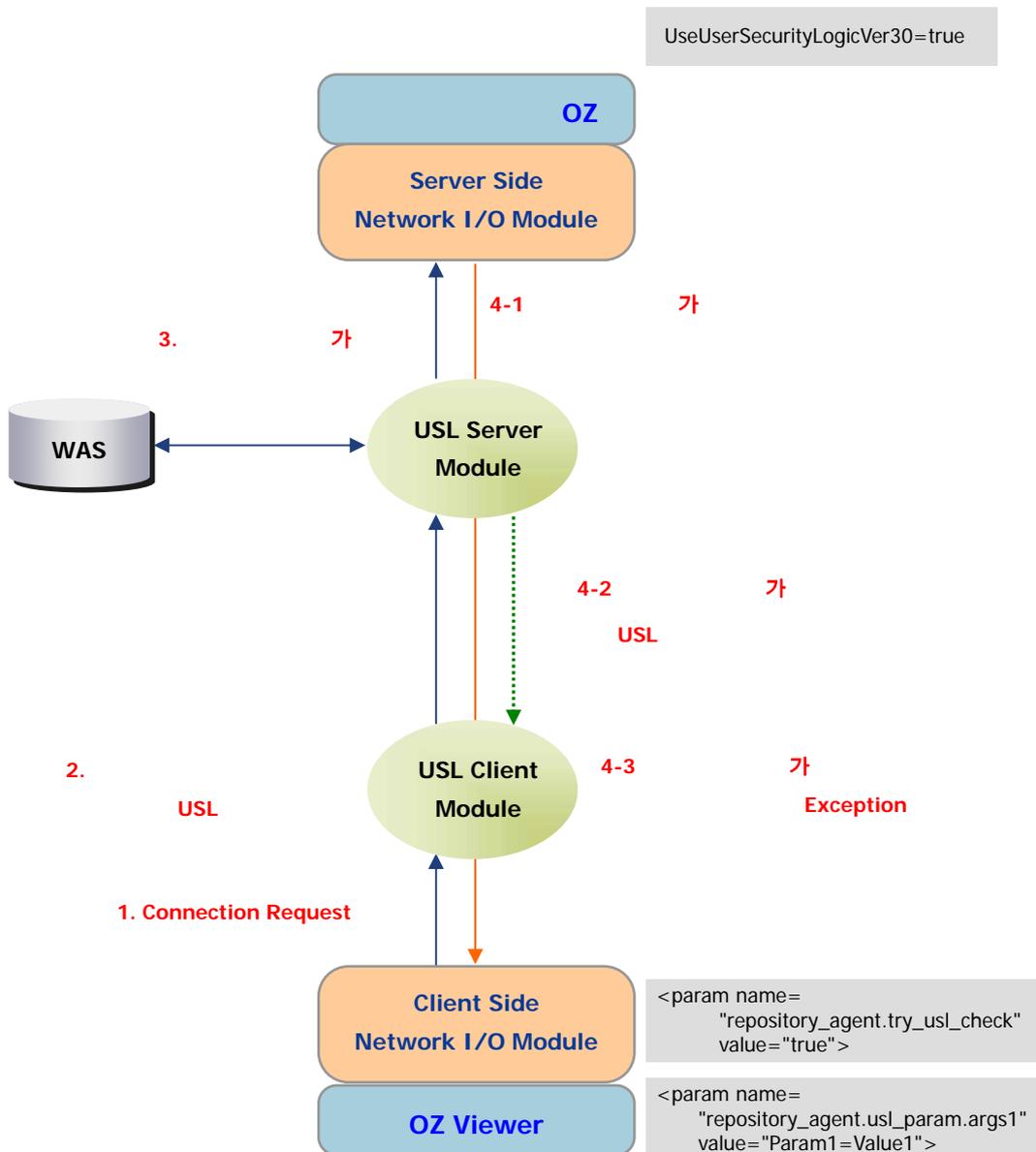
. USL    USL Server    USL Client  
I/O

### USL Flow



USL Session Cookie ( , PKI ) , Network I/O Stream

Session ( Session USL )



## USL

USL

- - USL : OZ\_HOME/lib/ozsfw50.jar
  - USL : OZ\_HOME/conf/uslmngr.properties

- - ozsfw50.jar CLASSPATH
  - uslmngr.properties USL

```
#
# use user security logic apply
#
UseUserSecurityLogicVer30=true
#
# default USL(Server&Client) class name (with package name)
#
OZDefault_SERVER=oz.usl.USL
OZDefault_CLIENT=oz.usl.USL
```

- uslmngr.properties UseUserSecurityLogicVer30=true , USL
- USL
  - OZDefault\_SERVER= oz.usl.USL
  - OZDefault\_CLIENT= oz.usl.USL
 ex)
  - OZDefault\_SERVER=oz.usl.OZUSLServerSession
  - OZDefault\_CLIENT=oz.usl.OZUSLClientSession

## OZ Report Viewer

- USL

```

(OZR)          USL
"repository_agent.try_usl_check"  "true"
<param name="repository_agent.try_usl_check" value="true">
:          "UseUserSecurityLogicVer30"  "true"
          "repository_agent.try_usl_check"  "true"

```

- USL

```

USL          "repository_agent.usl_param.pcount"  USL
, "repository_agent.usl_param.args#"
:          "UseUserSecurityLogicVer30"  "true"
, "repository_agent.try_usl_check"  "true"
USL
가 USL

```

- HTML

```

<html>
<body>
<object id="OZReportViewer" width="100%" height="100%" classid="CLSID:64DA633F-E73B-
4344-83BF-48483346CD53">
  <param name="connection.servlet" value="http://127.0.0.1:8087/ozservlet/server">
  <param name="viewer.configmode" value="html">

  <param name="repository_agent.try_usl_check" value="true">
  <param name="repository_agent.usl_param.pcount" value="2">
  <param name="repository_agent.usl_param.args1" value="Param1=Value1">
  <param name="repository_agent.usl_param.args2" value="Param2=Value2">

  <param name="connection.reportname" value="sample.ozr">
</object>
</body>
</html>

```

## OZ Application Viewer

- USL

```

(OZA)          USL
"repository_agent.try_usl_check"  "true"
:
"UseUserSecurityLogicVer30"      "true"
"repository_agent.try_usl_check"  "true"

```

- USL

```

USL          "repository_agent.usl_param.pcount"  USL
, "repository_agent.usl_param.args#"

```

- HTML

```

<html>
<body>
<object id="OZApplicationViewer" width="100%" height="100%" classid="CLSID:907A00F3-
7390-4EF2-931B-360546587804">
  <param name="repository_agent.type" value="FROM_OZSERVER">
  <param name="repository_agent.ozserver.servlet"
    value="http://127.0.0.1:8087/ozervlet/server">

  <param name="repository_agent.try_usl_check" value="true">
  <param name="repository_agent.usl_param.pcount" value="2">
  <param name="repository_agent.usl_param.args1" value="Param1=Value1">
  <param name="repository_agent.usl_param.args2" value="Param2=Value2">

  <param name="ozadoc.path" value="sample.oz">
</object>
</body>
</html>

```



DataOutputStream flush

## USL

USL Request Stream USL  
Response Stream

### ■ USL

OZ USL DLL USL 가

createSecureOutputStream(), createSecureInputStream()

Request Stream

createSecureOutputStream()

가 Response Stream

createSecureInputStream()

USL

createSecureOutputStream()

OutputStream

writeUTF() writeINT()

### ■ InputStream

( )

CJDataInputStream

가

USL

CJDataInputStream

read()가

CJDataInputStream

가

read가

### ■ OutputStream

( )

CJDataOutputStream

CJDataOutputStream

```

        USL                                CjDataOutputStream
        CjDataOutputStream
        write                                flush
        CjDataOutputStream
        flush
    
```

## USL 1 -

```

        USL                                WAS
        Valid
        OZUSLServerSession.jar
        OZUSLClientSession.dll
    
```

- USL OZ\_HOME/conf/uslMgr.properties

```

#
# use user security logic apply
#
UseUserSecurityLogicVer30=true

#
# default USL(Server&Client) class name (with package name)
#
OZDefault_SERVER=oz.usl.OZUSLServerSession
OZDefault_CLIENT=oz.usl.OZUSLClientSession
    
```

- USL
 

```

                OZUSLServerSession                                OZUSLServer
                OZUSLServerSession
                , WAS                                OZ Servlet
            
```

```
<<OZUSLServerSession.java>>
```

```
package oz.usl;

import java.io.*;
import java.util.Enumeration;
import javax.servlet.http.HttpSession;
import oz.cp.message.OzcmException;
import oz.framework.cp.io.OZDataInputStream;
import oz.framework.cp.io.OZDataOutputStream;

import org.apache.log4j.*;

public class OZUSLServerSession extends OZUSLServer
{
    ////////////////////////////////////////////////////////////////////
    // Non-custom section. Leave below as is.
    static protected Category cat =
    Category.getInstance(oz.version.OZSERVERFRAMEWORKVER.getLogVer()+".server");

    public OZUSLServerSession(){
        super();
    }

    ////////////////////////////////////////////////////////////////////
    // custom section. add your codes & modify

    // invoked after set HttpSession & clientIP
    public void initialize(){

    }

    // invoked once per a transaction when prepare to read from client
    public InputStream createSecureInputStream(DataInputStream raw_in)
        throws OzcmException
    {
        cat.debug("OZUSLServerSession: CreateSecureInputStream");
        try{
            params_from_client.read(new OZDataInputStream(raw_in));
            for(int i=params_from_client.getLength()-1; i>=0; i--){
                cat.debug("OZUSLServerSession: CreateSecureInputStream: "
                    + params_from_client.getKey(i) + ", " +
params_from_client.getValue(i));
            }
        }catch(Exception e){
            cat.error("OZUSLServerSession: read params from client:" , e);
            throw new OzcmException(e.getMessage());
        }
        return null;
    }
}
```

```

}

// invoked once per a transaction when prepare to write to client
public OutputStream createSecureOutputStream(DataOutputStream raw_out)
    throws OzcmException
{
    cat.debug("OZUSLServerSession: CreateSecureOutputStream");
    try{
        if(this.http_request != null){
            HttpSession http_session = http_request.getSession(false);

            if(http_session == null){
                cat.debug("http_session is null");
                http_session = http_request.getSession(true);

                http_session.setAttribute("user_id",
params_from_client.get("user_id"));
                http_session.setAttribute("user_pw",
params_from_client.get("user_pw"));
            }else {
                cat.debug("http_session is not null");
            }

            for(int i = params_from_client.getLength()-1; i >=0; i--){
                http_session.setAttribute(params_from_client.getName(i),
params_from_client.getValue(i));
            }

            Enumeration e = http_session.getAttributeNames();
            for (; e.hasMoreElements(); ) {
                String key = (String) e.nextElement();

                Object value = (Object)http_session.getAttribute(key);

                if(value instanceof String){
                    params_to_client.put(key, (String)value);
                    cat.debug("OZUSLServerSession: http_session: "+key+", "+value);
                }
            }

        }
        params_to_client.write(new OZDataOutputStream(raw_out));
    }catch(Exception e){
        e.printStackTrace();
        cat.error("OZUSLServerSession: write params to client: " , e);
        throw new OzcmException(e.getMessage());
    }
}

```



```

#include <common/io/OZFuncOutputStream.h>
#include <oz/usl/OZUSLClientWrapper.h>

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

OZUSLClientSession::OZUSLClientSession() : OZUSLClient()
{
}

OZUSLClientSession::~OZUSLClientSession()
{
}

CJInputStream* OZUSLClientSession::createSecureInputStream(CJDataInputStream*
raw_in, CJOZAttributeList& params_from_server_to_fillup, bool blnit)
{
    __super::createSecureInputStream(raw_in, params_from_server_to_fillup, blnit);
    return NULL;
}

CJOutputStream
OZUSLClientSession::createSecureOutputStream(CJDataOutputStream* raw_out,
CJOZAttributeList& params_from_client_to_send)
{
    __super::createSecureOutputStream(raw_out, params_from_client_to_send);
    return NULL;
}

void OZUSLClientSession::destory()
{
    delete this;
}
/*
 * OZUSLClient_create: initialize User Secure Logic
 *
 * this function is called every time a trasaction with OZServer occured
 *
 * @param [in] tags          null terminated UTF8 OZParameter tag string array
to send to server
 * @param [in] values       null terminated UTF8 OZParameter value string array
to send to server
 * @param [in] env_raw      environment variable used in OZClient. just forget
about what is it.
 *
 *                          however you must care about to pass env_raw to

```

```

rawRead(Wri te)Byte(BArray)
*
*           functions.
*
* @param [in] rawReadByte    function to get the encrypted raw byte from
server side USL to decrypt
* @param [in] rawReadBArray  function to get the encrypted raw byte array from
server side USL to decrypt
* @param [in] rawAvailable   function to get the tempory readable input buffer
size
* @param [in] rawReadClose   function to close raw inputstream
*
* @param [in] rawWriteByte   function to get the encrypted raw byte to send to
server side USL
* @param [in] rawWriteBArray function to get the encrypted raw byte array to
send to server side USL
* @param [in] rawFlush       function to flush raw outputstream
* @param [in] rawWriteClose  function to close raw outputstream
*
* @param [in] rawGetLastErrorMessage function to get error message from raw
in/out stream
*
* @param [out] do_read_hook   whether do input stream wrappering with
OZUSLClient_read or not
* @param [out] do_write_hook  whether do output stream wrappering with
OZUSLClient_write or not
* @param [out] env           the state variable if needs.
*
*           state variable is passed to all USL functions
*           to cover stateless function call interface.
*
* @return                    returns >= 0 if success returns returns < 0 if error.
*
*           if got error return, OZClient will call
OZUSLClient_getLastErrorMessage
*
*           to get detailed error message.
*/

extern "C" __declspec(dllexport) int OZUSLClient_create
(
    byte* params_bytes, int params_bytes_len,

    void* env_raw,

    int (__cdecl *rawReadByte)(void* env_raw, byte* p_value, int *read_len),
int (__cdecl *rawReadBArray)(void* env_raw, byte* array_value, int len, int
*read_len),
    int (__cdecl *rawAvai lable)(void* env_raw, int *avai lable_len),
    int (__cdecl *rawReadClose)(void* env_raw, BOOL do_not_close_raw_in),

```

```

        int (__cdecl *rawWriteByte)(void* env_raw, byte value),
        int (__cdecl *rawWriteByteArray)(void* env_raw, byte* array_value, int len),
            int (__cdecl *rawFlush)(void* env_raw),
        int (__cdecl *rawWriteClose)(void* env_raw),

        byte* (__cdecl *rawGetLastErrorMessage)(void* env_raw),

        void** env
    )
{
    TRACE(_T("OZUSLClient_create\n"));

    OZUSLClientWrapper *uslw = NULL;

    uslw = new OZUSLClientWrapper();
    uslw->raw_in = new CJDataInputStream
        (new OZFuncInputStream(env_raw, uslw, rawReadByte, rawReadByteArray));

    uslw->raw_out = new CJDataOutputStream
        (new OZFuncOutputStream(env_raw, rawWriteByte, rawWriteByteArray,
        rawWriteClose));

    uslw->ozusl = new OZUSLClientSession();

    CJOZAttributeList attrs;
    CJByteArrayInputStream bin((char*)params_bytes, params_bytes_len, FALSE);
    CJDataInputStream din(&bin, FALSE);

    attrs.read(din);

    uslw->ozusl->setServerIP(attrs.get(_T("OZ_SERVER_IP")));
    uslw->ozusl->setServerURL(attrs.get(_T("OZ_SERVER_URL")));

    *env = uslw;
    return OZUSLCLIENT_INTERFACE_VERSION; // all ok.
}

```

■ **USL** - USL 가

(getUSLServerParam) USL 가

[ ]-[ ]

()

- 가 [ 가]-[ ]-[ ]-[ ]



가 .

- [ 가]-[ ]-[ ]-[ ]



가 .

- [ ]

가 .

```
String str;
str = getUSLServerParam("OZUSL_SERVER_URL");
// name USL OZUSL_SERVER_URL
setattr("caption", str);
write("str"); // ( CTRL+Z)
```

- [ ] [ ] 

'getUSLServerParam.ozr'

'D:\Repository'

- html .

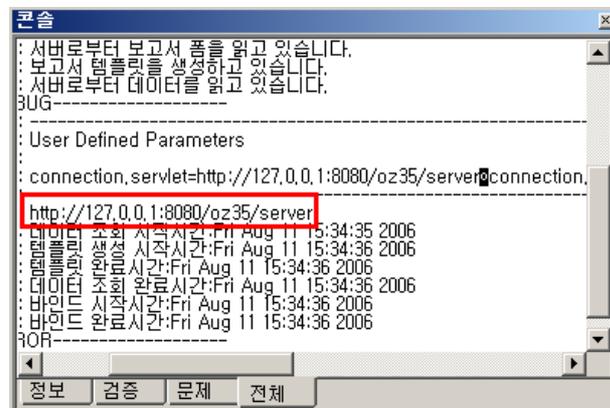
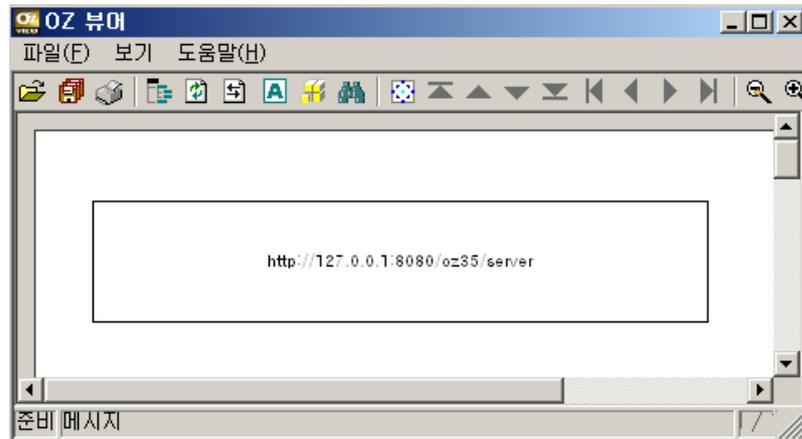
```
<HTML>
<BODY>
<OBJECT id = "ozviewer" CLASSID="CLSID: ODEF32F8-170F-46f8-B1FF-4BF7443F5F25" width="100%" height="100%">

<param name="connection.servlet" value="http://127.0.0.1:8080/oz50/server">
<param name="connection.reportname" value="/getUSLServerParam.ozr">
<param name="connection.usingusl" value="true">
<param name="viewer.isframe" value="true">
<param name="information.debug" value="true">

</OBJECT>
</body>
</HTML>
```

connection.usingusl true .

- caption USL



## . User Defined Log

 UDL

 UDL

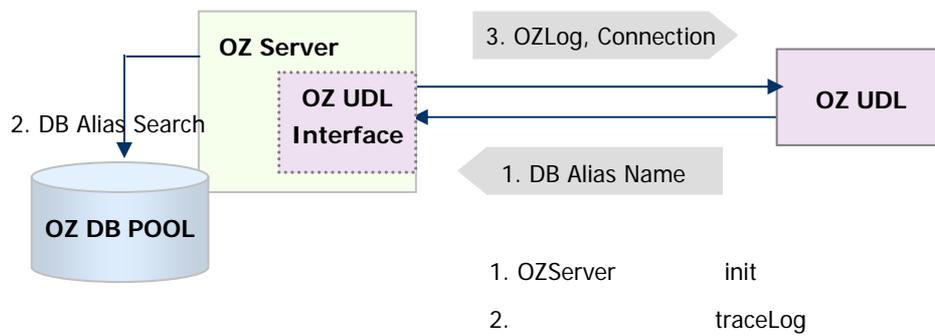
 UDL

## UDL

UDL(User Defined Log)

DB 가

UDL



## UDL

UDL

### OZUserDefinedLogger

OZ UDL `oz.udl.OZUserDefinedLogger`  
`oz.udl.OZUserDefinedLogger` `init, tracelog, getOZDBAlias`

- `init`

---

<b>Prototype</b>	<code>public void init(oz.framework.db.OZConnection ozconn, com.forcs.log4oz.OZLog cat) throws OZUserDefinedLogException</code>
<b>Definition</b>	
<b>Argument</b>	<code>ozconn</code> DB DB <code>cat</code>

---

- `tracelog`

---

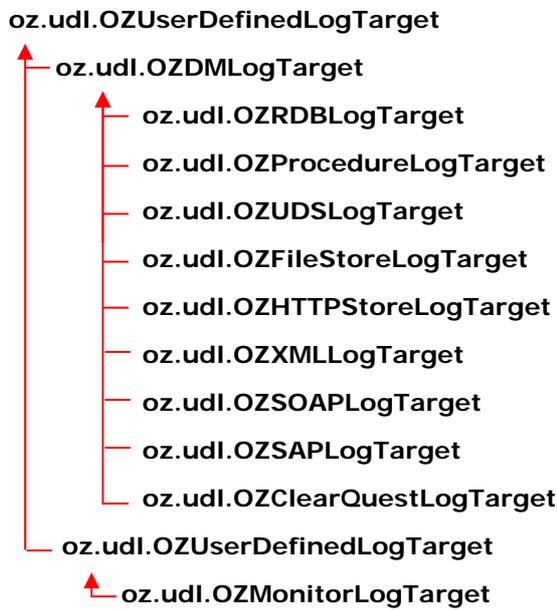
<b>Prototype</b>	<code>public void tracelog(oz.udl.OZUserDefinedLogTarget target, oz.framework.db.OZConnection ozconn, com.forcs.log4oz.OZLog cat) throws OZUserDefinedLogException</code>
<b>Definition</b>	
<b>Argument</b>	<code>target</code> OZUserDefinedLogTarget <code>ozconn</code> DB DB <code>cat</code>

---

- getOZDBAlias

<b>Prototype</b>	public java.lang.String getOZDBAlias()
<b>Definition</b>	DB
<b>return</b>	DB

## OZUserDefinedLogTarget



### ■ OZUserDefinedLogTarget

OZ UDL

- getIPAddress

<b>Prototype</b>	public String getIPAddress()
<b>Definition</b>	가 IP 가
<b>return</b>	IP

- getHttpRequest

<b>Prototype</b>	public HttpServletRequest getHttpRequest()
<b>Definition</b>	가 HttpRequest 가

<b>return</b>	HttpServletRequest	:	가	:	null	.
---------------	--------------------	---	---	---	------	---

- getHttpServlet

<b>Prototype</b>	public HttpServlet getHttpServlet()
<b>Definition</b>	가 HttpServlet 가 .
<b>return</b>	HttpServlet : 가 null .

- getUserID

<b>Prototype</b>	public String getUserID()
<b>Definition</b>	: ID 가 .
<b>return</b>	ID : UserID (" ") .

■ **OZDMLogTarget**

OZ UDL

- getODIName

<b>Prototype</b>	public java.lang.String getODIName()
<b>Definition</b>	ODI 가 .
<b>return</b>	ODI

- getDataSetName

<b>Prototype</b>	public java.lang.String getDataSetName()
<b>Definition</b>	가 .
<b>return</b>	

- getDataStoreName

<b>Prototype</b>	public java.lang.String getDataStoreName()
<b>Definition</b>	가 .
<b>return</b>	

- getOZParam

<b>Prototype</b>	public java.util.HashMap getOZParam()
------------------	---------------------------------------

<b>Definition</b>	HashMap	가	.
<b>return</b>	key	Value	

■ **OZRDBLogTarget**

OZ UDL RDB

가 .

- getDBAlias

<b>Prototype</b>	public String	getDBAlias()	
<b>Definition</b>	ODI	DB	가 .
<b>return</b>	DB	:	가 .

- getDBExecuteQuery

<b>Prototype</b>	public String	getDBExecuteQuery()
<b>Definition</b>		가 .
<b>return</b>		

- getPreparedQueryValue

<b>Prototype</b>	public Vector	getPreparedQueryValue()
<b>Definition</b>		가 .
<b>return</b>	(	)

- getExecuteQueryState

<b>Prototype</b>	public String	getExecuteQueryState()
<b>Definition</b>		가 .
<b>return</b>	true	:"false:
	false: <i>ErrMsg</i>	..

- getQueryExecuteTime

<b>Prototype</b>	public Long	getQueryExecuteTime()
<b>Definition</b>		가 .
<b>return</b>	(	: msec)



- getProcedureExecuteTime

<b>Prototype</b>	public Long getProcedureExecuteTime()
<b>Definition</b>	가 가 .
<b>return</b>	( : msec)

- getProcedureState

<b>Prototype</b>	public String getProcedureState()
<b>Definition</b>	가 가 .
<b>return</b>	true 가 false: <i>ErrMsg</i> 가 "false: ..

■ OZUDSLogTarget

OZ UDL

가 .

- getExecuteCommand

<b>Prototype</b>	public String getExecuteCommand()
<b>Definition</b>	가 .
<b>return</b>	

- getCRUDFlagName

<b>Prototype</b>	public String getCRUDFlagName()
<b>Definition</b>	가 .
<b>return</b>	select Select insert Insert update Update delete Delete

- getCommandExecuteTime

<b>Prototype</b>	public Long getCommandExecuteTime()
<b>Definition</b>	가 .
<b>return</b>	( : msec)

- getCommandState

<b>Prototype</b>	public String getCommandState()
<b>Definition</b>	가
<b>return</b>	true false: <i>ErrMsg</i> "false:"

■ **OZFileStoreLogTarget**

OZ UDL

가

- getFilePath

<b>Prototype</b>	public String getFilePath()
<b>Definition</b>	가
<b>return</b>	

- getFileAccessTime

<b>Prototype</b>	public Long getFileAccessTime()
<b>Definition</b>	가 가
<b>return</b>	( : msec)

- getFileAccessState

<b>Prototype</b>	public String getFileAccessState()
<b>Definition</b>	가
<b>return</b>	true false: <i>ErrMsg</i> "false:"

■ **OZHTTPStoreLogTarget**

OZ UDL

HTTP

HTTP

가

- getURL

<b>Prototype</b>	public String getURL()
<b>Definition</b>	URL 가
<b>return</b>	URL

- getHTTPAccessTime

<b>Prototype</b>	public Long getHTTPAccessTime()
<b>Definition</b>	HTTP 가 가 .
<b>return</b>	( : mesc)

- getHTTPAccessState

<b>Prototype</b>	public String getHTTPAccessState()
<b>Definition</b>	HTTP 가 .
<b>return</b>	true false: <i>ErrMsg</i> .. "false:

■ **OZXMLLogTarget**

OZ UDL XML XML  
가 .

- getURL

<b>Prototype</b>	public String getURL()
<b>Definition</b>	URL 가 .
<b>return</b>	URL

- getURLAccessTime

<b>Prototype</b>	public Long getURLAccessTime()
<b>Definition</b>	URL 가 가 .
<b>return</b>	( : mesc)

- getURLAccessState

<b>Prototype</b>	public String getURLAccessState()
<b>Definition</b>	URL 가 .
<b>return</b>	true false: <i>ErrMsg</i> .. "false:

■ **OZSOAPLogTarget**

OZ UDL SOAP Service Response

가 .

- getServiceName

**Prototype** public String getServiceName()

**Definition** 가 .

**return**

- getPort

**Prototype** public String getPort()

**Definition** 가 .

**return**

- getOperation

**Prototype** public String getOperation()

**Definition** 가 .

**return**

- getEndPoint

**Prototype** public String getEndPoint()

**Definition** EndPoint 가 .

**return** EndPoint

- getRequestXML

**Prototype** public String getRequestXML()

**Definition** XML 가 .

**return** XML

- getServiceExecuteTime

**Prototype** public Long getServiceExecuteTime()

**Definition** 가 가 .

**return** ( : msec)

- getServiceState

<b>Prototype</b>	public String getServiceState()
<b>Definition</b>	가 SOAP 가
<b>return</b>	true 가 false: ErrMsg 가 "false:"

■ **OZSAPLogTarget**

OZ UDL SAP 가 가

- **getFunctionName**

<b>Prototype</b>	public String getFunctionName()
<b>Definition</b>	가
<b>return</b>	

- **getFunctionType**

<b>Prototype</b>	public String getFunctionType()
<b>Definition</b>	가
<b>return</b>	

- **getInputParameters**

<b>Prototype</b>	public HashMap getInputParameters()
<b>Definition</b>	HashMap 가
<b>return</b>	key value

- **getResultSetTypes**

<b>Prototype</b>	public String getResultSetTypes()
<b>Definition</b>	ResultSet 가
<b>return</b>	Structure Structure Table Table SimpleFields SimpleFields

- **getFunctionExecuteTime**

<b>Prototype</b>	public Long getFunctionExecuteTime()
------------------	--------------------------------------

<b>Definition</b>	가 가 .
<b>return</b>	( : mesc)

- getFunctionState

<b>Prototype</b>	public String getFunctionState()
<b>Definition</b>	가 가 .
	true 가
<b>return</b>	false: ErrMsg 가 "false:

■ **OZClearQuestLogTarget**

OZ UDL Clear Quest Clear Quest  
가 .

- getExecuteType

<b>Prototype</b>	public String getExecuteType()
<b>Definition</b>	Clear Quest 가 .
	ExecuteSQL SQL
<b>return</b>	ExecuteQuery
	ExecuteDynamicQuery

- getQuerySubType

<b>Prototype</b>	public String getQuerySubType()
<b>Definition</b>	QuerySub 가 .
<b>return</b>	Query Query
	Chart Chart

- getQuery

<b>Prototype</b>	public String getQuery()
<b>Definition</b>	가 .
<b>return</b>	

- getQueryExecuteTime

<b>Prototype</b>	public Long getQueryExecuteTime()
<b>Definition</b>	가 .

**return** ( : msec)

---

- getQueryState

**Prototype** public String getQueryState()

---

**Definition** 가

---

true

---

**return** false: *ErrMsg* "false:"

---

■ OZMonitorLogTarget

OZ UDL

- getMark

**Prototype** public String getMark()

---

**Definition** 가

---

start

---

**return** end

---

- getThreadName

**Prototype** public String getThreadName()

---

**Definition** Name 가

---

**return**

---

- getServiceTime

**Prototype** public Long getServiceTime()

---

**Definition** 가 . ( : ms)  
: 1970 1 1 Millisecond

---

**return**

---

- getFreeMemory

**Prototype** public Long getFreeMemory()

---

**Definition** 가 가 . ( : byte)

---

**return**

---

- getTotalMemory

<b>Prototype</b>	public long getTotalMemory()
<b>Definition</b>	JVM 가 . ( : byte)
<b>return</b>	

- getServiceCode

<b>Prototype</b>	public integer getServiceCode()
<b>Definition</b>	가 . <ul style="list-style-type: none"> <li>MARK가 "end" "start" -1</li> <li>가</li> </ul>
<b>return</b>	

- getServiceStatus

<b>Prototype</b>	public integer getServiceStatus()
<b>Definition</b>	가 . <ul style="list-style-type: none"> <li>MARK가 "end" "start" -1</li> <li>"9001", "9002"</li> </ul>
<b>return</b>	9001 9002

- getServiceParameter

<b>Prototype</b>	public String getServiceParameter()
<b>Definition</b>	가 . : MARK가 "end" "start" -1 .
<b>return</b>	

- getDBSessionID

<b>Prototype</b>	public String getDBSessionID()
<b>Definition</b>	DBMS ID 가 . : MARK가 "end" "start"

	-1	.
<b>return</b>	DBMS ID	

- getExecuteTime

<b>Prototype</b>	public String getExecuteTime()
<b>Definition</b>	: MARK가 "end" "start" -1 .
<b>return</b>	

- getErrorCode

<b>Prototype</b>	public String getErrorCode()
<b>Definition</b>	: conf/server_error_msg_ _ 가 .xml .
<b>return</b>	

- getErrorMsg

<b>Prototype</b>	public String getErrorMsg()
<b>Definition</b>	가
<b>return</b>	

- getErrorStackTrace

<b>Prototype</b>	public String getErrorStackTrace()
<b>Definition</b>	Stack Trace 가 .
<b>return</b>	Stack Trace

- getDBConns

<b>Prototype</b>	public String getDBConns()
<b>Definition</b>	Alias DB
<b>return</b>	Alias DB 가 .

## UDL

### UDL

- - UDL : OZ\_HOME/lib/ozudl.jar
  - UDL : OZ\_HOME/conf/ozudl.properties
  
- - ozudl.jar CLASSPATH
  - ozudl.properties UDL

```
#-----
# configuraion of OZ User Defined log
#-----
#

OZ_USER_DEFINED_LOG.Active=true
OZ_USER_DEFINED_LOG.Class=oz.udl.UDL

OZ_UDL_MONITOR.Active= true
OZ_UDL_MONITOR.Class=oz.udl.UDL
```

- ozudl.properties UDL
  - OZ\_USER\_DEFINED\_LOG.Active=true
  - OZ\_UDL\_MONITOR.Active= true
  
- UDL
  - OZ\_USER\_DEFINED\_LOG.Class=oz.udl.UDL

➤ OZ\_UDL\_MONITOR.Class=oz.udl.UDL

ex)

OZ\_USER\_DEFINED\_LOG.Class=oz.udl.file.OZUserDefinedLogger

OZ\_UDL\_MONITOR.Class=oz.udl.db.OZMonitorLogForDB

## UDL 1 -

UDL

가



UDL

java

oz.udl.file.OZUserDefinedLoggerForFile

"ozudl.jar"

```
package oz.udl.file;

import java.io.*;
import java.util.*;
import org.apache.log4j.*;
import javax.servlet.http.*;

import oz.udl.*;
import oz.framework.db.*;
import com.forcs.log4oz.OZLog;
import oz.framework.dac.OZDACData;
import oz.dm.meta.OZProcedureParameter;

public class OZUserDefinedLoggerForFile implements OZUserDefinedLogger{

    Category logger = null;

    private final static String ROLLING_FILE_APPENDER = "RollingAppender";
    private final static String DAILY_ROLLING_FILE_APPENDER =
"DailyRollingAppender";

    private String logpath;
    private String layouts;
    private String appenderType;
    private String filesize;
```

```
private String backupindex;
private String dailytype;

public OZUserDefinedLoggerForFile() {

}

/**
 *      가
 * @param ozconn OZConnection      Connection
 * @param cat OZLog      trace
 * @throws OZUserDefinedLogException
 */
public void init(OZConnection ozconn, OZLog cat) throws
OZUserDefinedLogException {
    try {
        load(cat);
        setAppender(cat);

    }catch(Exception ex){
        if (logger != null) {
            logger.error(ex.getMessage(), ex);
            cat.error(ex.getMessage(), ex);
        }
        throw new OZUserDefinedLogException(ex.getMessage());
    }
}

private void load(OZLog cat) {
    Properties props = new Properties();
    FileInputStream fis = null;
    try {
        fis = new FileInputStream("udl_file.properties");
        props.load(fis);

        logpath=props.getProperty("logpath");
        layoutS=props.getProperty("layout");
        AppenderType=props.getProperty("AppenderType");
        fileSize=props.getProperty("fileSize");
        backupindex=props.getProperty("backupindex");
        dailytype=props.getProperty("dailytype");

    }catch(Exception ex){
        cat.error(ex.getMessage(), ex);
    }finally{
        if (fis != null) try { fis.close(); }catch(Exception ex1){}
    }
}
```

```

}

private void setAppender(OZLog cat) throws Exception {
    PatternLayout layout = new PatternLayout(layoutS);

    if (ROLLING_FILE_APPENDER.equals(AppenderType)) {
        RollingFileAppender appender = new RollingFileAppender(layout,
logpath, true);

        appender.setMaxFileSize(fileSize);
        try {
            appender.setMaxBackupIndex(Integer.parseInt(backupIndex));
        } catch (Exception ex) {
            appender.setMaxBackupIndex(5);
        }

        logger = Category.getInstance("oz.udl.file");
        logger.setPriority(Priority.DEBUG);

        logger.addAppender(appender);

        logger.debug("start...");
        cat.debug("user db logger init..."); // trace
    }

    else if (DAILY_ROLLING_FILE_APPENDER.equals(AppenderType)) {

        String datePattern = "." + dailyType;
        DailyRollingFileAppender appender = new DailyRollingFileAppender();

        appender.setDatePattern(datePattern);
        appender.setLayout(layout);
        appender.setFile(logpath, true);
    }
}

/**
 * trace target 가 가
 * @param target OZUserDefinedLogTarget
 * @param ozconn OZConnection
 * @param cat OZLog ( trace param )cat.debug(str);)
 * @throws OZUserDBLogException
 */
public void traceLog(OZUserDefinedLogTarget target, OZConnection ozconn,
OZLog cat) throws OZUserDefinedLogException {

    StringBuffer sb = new StringBuffer();

```

```
try {

    if(target instanceof OZProcedureLogTarget) {
        sb.append(procedureStore(target));
    }
    else if(target instanceof OZRDBLogTarget){
        sb.append(rdbstore(target));
    }
    else if(target instanceof OZUDSLogTarget) {
        sb.append(udsStore(target));
    }
    else if(target instanceof OZFileStoreLogTarget) {
        sb.append(filestore(target));
    }
    else if(target instanceof OZHTTPStoreLogTarget) {
        sb.append(httpStore(target));
    }
    else if(target instanceof OZXMLLogTarget) {
        sb.append(xmlStore(target));
    }
    else if(target instanceof OZSOAPLogTarget) {
        sb.append(soapStore(target));
    }
    else if(target instanceof OZSAPLogTarget) {
        sb.append(sapStore(target));
    }
    else if(target instanceof OZClearQuestLogTarget) {
        sb.append(clearQuestStore(target));
    }
    sb.append(commonfactor(target));

    logger.info(sb.toString());
} catch(Exception ex){
    logger.error(ex.getMessage(), ex);
    cat.error(ex.getMessage(), ex);
    throw new OZUserDefinedLogException(ex.getMessage());
}

}

public String rdbstore(OZUserDefinedLogTarget target){
    OZRDBLogTarget ref = (OZRDBLogTarget)target;
    StringBuffer sb = new StringBuffer();
    String dbAlias = (ref.getDBAlias() == null)? "" : ref.getDBAlias();
    sb.append("[DB Alias=");
    sb.append(dbAlias);
    sb.append("]");
}
```

```

        String crudFlagName = (ref.getCRUDFlagName() == null) ? "" :
ref.getCRUDFlagName();
        sb.append("[CRUDType=");
        sb.append(crudFlagName);
        sb.append("]");
        sb.append("[isPrepared=");
        sb.append(ref.isPrepared());
        sb.append("]");
        if(ref.getPreparedQueryValue() != null){
            sb.append("[QueryPreparedValue=");
            Vector vec = ref.getPreparedQueryValue();
            if (crudFlagName.equals("select")) {
                for (int i = 0; i < vec.size(); i++) {
                    sb.append(vec.elementAt(i));
                }
            }else {
                for (int i = 0; i < vec.size(); i++) {
                    OZDACData data = (OZDACData)vec.elementAt(i);
                    sb.append(data.toStringData());
                    sb.append(";");
                }
            }
            sb.append("]");
        }
        String query = (ref.getDBExecuteQuery() == null) ? "" :
ref.getDBExecuteQuery();
        sb.append("[Query=");
        sb.append(query);
        sb.append("]");
        Long queryExec = ref.getQueryExecuteTime();
        sb.append("[Query exec=");
        sb.append(queryExec);
        sb.append("]");
        String queryState = (ref.getExecuteQueryState() == null) ? "" :
ref.getExecuteQueryState();
        sb.append("[QueryState=");
        sb.append(queryState);
        sb.append("]");

        return sb.toString();
    }

    public String udsStore(OZUserDefinedLogTarget target){
        OZUDSLogTarget ref = (OZUDSLogTarget)target;
        StringBuffer sb = new StringBuffer();

        sb.append("[ExecuteCommand=");
        sb.append(ref.getExecuteCommand());

```

```
sb.append("]");
sb.append("[CRUDFI agName=");
sb.append(ref.getCRUDFI agName());
sb.append("]");
sb.append("[CommandExecuteTi me=");
sb.append(ref.getCommandExecuteTi me());
sb.append("]");
sb.append("[CommandState=");
sb.append(ref.getCommandState());
sb.append("]");

return sb.toString();
}

public String procedureStore(OZUserDefi nedLogTarget target){
    OZProcedureLogTarget ref = (OZProcedureLogTarget)target;
    StringBuffer sb = new StringBuffer();

    sb.append("[DBAI i as=");
    sb.append(ref.getDBAI i as());
    sb.append("]");
    sb.append("[ProcedureName=");
    sb.append(ref.getProcedureName());
    sb.append("]");
    sb.append("[ParameterLi st=");
    ArrayLi st l i s t = ref.getParameterLi st();
    for(int i=0; i<l i s t. si ze(); i++){
        OZProcedureParameter param = (OZProcedureParameter)l i s t. get(i);
        sb.append(param. getVal ue());
        sb.append(", ");
        sb.append(param. getDataTypeNa me());
        sb.append("; ");
    }
    sb.append(ref.getParameterLi st());
    sb.append("]");
    sb.append("[ProcedureExecuteTi me=");
    sb.append(ref.getProcedureExecuteTi me());
    sb.append("]");
    sb.append("[ProcedureState=");
    sb.append(ref.getProcedureState());
    sb.append("]");

    return sb.toString();
}

public String xml Store(OZUserDefi nedLogTarget target){
    StringBuffer sb = new StringBuffer();
    OZXMLLogTarget ref = (OZXMLLogTarget)target;
```

```
        sb.append(" [URL=");
        sb.append(ref.getURL());
        sb.append("]");
        sb.append(" [URLAccessTime=");
        sb.append(ref.getURLAccessTime());
        sb.append("]");
        sb.append(" [URLAccessState=");
        sb.append(ref.getURLAccessState());
        sb.append("]");

        return sb.toString();
    }

    public String soapStore(OZUserDefinedLogTarget target){
        StringBuffer sb = new StringBuffer();
        OZSOAPLogTarget ref = (OZSOAPLogTarget)target;

        sb.append(" [ServiceName=");
        sb.append(ref.getServiceName());
        sb.append("]");
        sb.append(" [Port=");
        sb.append(ref.getPort());
        sb.append("]");
        sb.append(" [EndPoint=");
        sb.append(ref.getEndPoint());
        sb.append("]");
        sb.append(" [ServiceExecuteTime=");
        sb.append(ref.getServiceExecuteTime());
        sb.append("]");
        sb.append(" [ServiceState=");
        sb.append(ref.getServiceState());
        sb.append("]");
        sb.append(" [RequestXML=");
        sb.append(ref.getRequestXML());
        sb.append("]");

        return sb.toString();
    }

    public String sapStore(OZUserDefinedLogTarget target){
        StringBuffer sb = new StringBuffer();
        OZSAPLogTarget ref = (OZSAPLogTarget)target;

        sb.append(" [FunctionName=");
        sb.append(ref.getFunctionName());
        sb.append("]");
        sb.append(" [FunctionType=");
```

```
sb.append(ref.getFunctionType());
sb.append("]");
sb.append("[ResultSetTypes=");
sb.append(ref.getResultSetTypes());
sb.append("]");
sb.append("[InputParameters=");
sb.append(ref.getInputParameters());
sb.append("]");
sb.append("[FunctionExecuteTime=");
sb.append(ref.getFunctionExecuteTime());
sb.append("]");
sb.append("[FunctionState=");
sb.append(ref.getFunctionState());
sb.append("]");

return sb.toString();
}

public String clearQuestStore(OZUserDefinedLogTarget target){
    StringBuffer sb = new StringBuffer();
    OZClearQuestLogTarget ref = (OZClearQuestLogTarget)target;

    sb.append("[ExecuteType=");
    sb.append(ref.getExecuteType());
    sb.append("]");
    sb.append("[QuerySubType=");
    sb.append(ref.getQuerySubType());
    sb.append("]");
    sb.append("[Query=");
    sb.append(ref.getQuery());
    sb.append("]");
    sb.append("[QueryExecuteTime=");
    sb.append(ref.getQueryExecuteTime());
    sb.append("]");
    sb.append("[QueryState=");
    sb.append(ref.getQueryState());
    sb.append("]");

    return sb.toString();
}

public String httpStore(OZUserDefinedLogTarget target){
    OZHTTPStoreLogTarget ref = (OZHTTPStoreLogTarget)target;
    StringBuffer sb = new StringBuffer();

    sb.append("[URL=");
    sb.append(ref.getURL());
    sb.append("]");
```

```
sb.append("[HTTPAccessTime=");
sb.append(ref.getHTTPAccessTime());
sb.append("]");
sb.append("[HTTPAccessState=");
sb.append(ref.getHTTPAccessState());
sb.append("]");

return sb.toString();
}

public String filestore(OZUserDefinedLogTarget target){
    StringBuffer sb = new StringBuffer();

    OZFileStoreLogTarget ref = (OZFileStoreLogTarget)target;
    if(ref!=null){
        sb.append("[FilePath=");
        sb.append(ref.getFilePath());
        sb.append("]");
        sb.append("[FileAccessTime=");
        sb.append(ref.getFileAccessTime());
        sb.append("]");
        sb.append("[FileAccessState=");
        sb.append(ref.getFileAccessState());
        sb.append("]");
    }else{
        sb.append("filestore is null");
    }

    return sb.toString();
}

public String commonfactor(OZUserDefinedLogTarget target){

    OZDMLogTarget ref = (OZDMLogTarget)target;
    StringBuffer sb = new StringBuffer();

    HttpServletRequest request = ref.getHttpRequest();
    if(request != null){
        sb.append("[Request Parameter=");
        Enumeration en = request.getParameterNames();
        while(en.hasMoreElements()){
            String key = (String)en.nextElement();
            String value = request.getParameter(key);
            sb.append("(");
            sb.append(key + ", " + value);
            sb.append(")");
        }
        sb.append("]");
    }
```

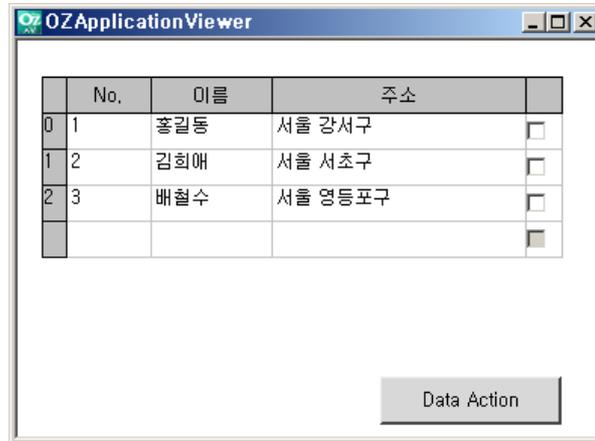
```
HttpSession session = request.getSession(false);
if(session != null){
    sb.append("[Session Attribute=");
    en = session.getAttributeNames();
    while(en.hasMoreElements()){
        String key = (String)en.nextElement();
        sb.append("(");
        sb.append(key);
        sb.append(", ");
        sb.append(session.getAttribute(key));
        sb.append(")");
    }
    sb.append("]");
}

String ipAddress = (ref.getIPAddress() == null) ? "" :
ref.getIPAddress();
sb.append("[IP Address=");
sb.append(ipAddress);
sb.append("]");
String userId = (ref.getUserID() == null) ? "" : ref.getUserID();
sb.append("[UserID=");
sb.append(userId);
sb.append("]");
String odiName = (ref.getODIName() == null) ? "" : ref.getODIName();
sb.append("[ODI Name=");
sb.append(odiName);
sb.append("]");
String dataSetName = (ref.getDataSetName() == null) ? "" :
ref.getDataSetName();
sb.append("[DataSetName=");
sb.append(dataSetName);
sb.append("]");
String dataStoreName = (ref.getDataStoreName() == null) ? "" :
ref.getDataStoreName();
sb.append("[DataStoreName=");
sb.append(dataStoreName);
sb.append("]");

return sb.toString();
}

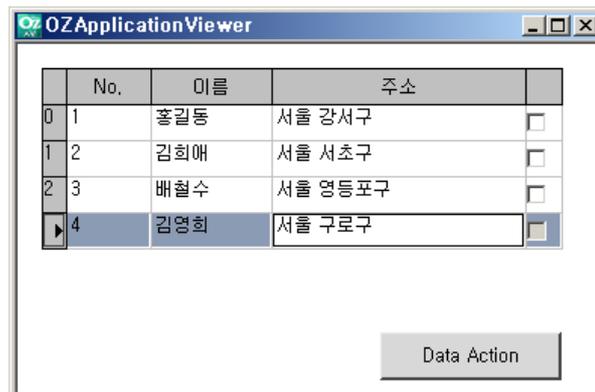
/**
 * UDL DB OZ Server Pool alias
 * @return String
 */
public String getOZDBAlias() {
```





가 [Data Action]

가



가 /log "UDL.log"

UDL.log

ODI

```
[INFO ] OZUserDefinedLoggerForFile.traceLog - [DB Alias=qa_ms][IP Address=127.0.0.1][UserID=admin][CRUDType=select][isPrepared=true][QueryPreparedValue=][Query=select * from REQ397][Query exec=16][QueryState=true][ODI Name=/DataAction.odi][DataSetName=Sample][DataStoreName=DB]
```

```
[INFO ] OZUserDefinedLoggerForFile.traceLog - [DB Alias=qa_ms][IP Address=127.0.0.1][UserID=admin][CRUDType=insert][isPrepared=true][QueryP
```

```

reparedValue=4; ; ][Query=insert into REQ397
(id, name, address)
values (?, ?, ?)][Query exec=16][QueryState=true]
[ODI Name=/DataAction.odi ][DataSetName=Sample][DataStoreName=DB]
    
```

## UDL 2 -

UDL 가



UDL  
 java  
 oz.udl.file.OZUserDefinedLoggerForDB "ozudl.jar"  
 ( db.properties DB  
 "Sample" "OZDBHISTORY" .)

```

package oz.udl.db;

import oz.udl.*;
import oz.framework.db.*;
import com.forcs.log4oz.OZLog;
import oz.framework.dac.OZDACData;

import java.sql.*;
import java.util.*;
import java.text.*;
import javax.servlet.http.*;
import oz.dm.meta.OZProcedureParameter;

public class OZUserDefinedLoggerForDB implements OZUserDefinedLogger{

    private final static String TABLENAME = "OZDBHISTORY";
    private final static String QUERY = "insert into "+TABLENAME +
        " values (?, ?, ?, ?, ?, ?, ?) ";

    public OZUserDefinedLoggerForDB(){}
    
```

```
/**
 *      가
 * @throws OZUserDefi nedLogExcepti on
 */
public void ini t(OZConnecti on ozconn, OZLog cat) throws
OZUserDefi nedLogExcepti on{
    try{
        Connecti on conn = ozconn.getConnecti on();

        if(!isExi stsTable(conn, TABLENAME)){
            String createQuery =
                "create table "+TABLENAME+" ( " +
                "property text NOT NULL, "+
                "Sessi onInfo text NULL, "+
                "IP varchar(27) NULL, "+ //
                "USERID varchar(255) NULL, "+ //
                "ODI NAME varchar(255) NOT NULL, "+ //
                "DataSetName varchar(255) NOT NULL, "+ //
                "DataStoreName varchar(255) NOT NULL "+ //
                " ) ";

            /*
                "create table "+TABLENAME+" ( " +
                    "TIME_ID varchar(255) NOT NULL, "+
                    "DB_ALIAS varchar(128) NOT NULL, "+
                    "IP varchar(27) NULL, "+
                    "USERID varchar(255) NULL, "+
                    "CRUDNAME varchar(6) NULL, "+
                    "ISPrepared varchar(5) NULL, "+
                    "Sessi onInfo text NULL, "+
                    "preParedValue text NULL, "+
                    "Query text NULL, "+
                    "exe_time int NOT NULL, "+
                    "queryState varchar(255) NOT NULL, "+
                    "ODI NAME varchar(255) NOT NULL, "+
                    "DataSetName varchar(255) NOT NULL, "+
                    "DataStoreName varchar(255) NOT NULL "+
                    " ) ";
            */

            PreparedStatement pstmt = conn.prepareStatement(createQuery);

            pstmt.executeUpdate();

            createQuery = "CREATE INDEX " + TABLENAME + "_i dx_ti me_i d ON "
                + TABLENAME + " (TIME_ID)";
        }
    }
}
```

```

        pstmt.close();
        pstmt = conn.prepareStatement(createQuery);
        pstmt.executeUpdate();
        pstmt.close();
    }
}
catch(Exception ex){
    cat.error(ex.getMessage(), ex);
    throw new OZUserDefinedLogException(ex.getMessage());
}
}

/**
 *      trace      target   가      가
 * @param target OZUserDefinedLogTarget
 * @param ozconn OZConnection
 * @param cat OZLog (      trace      param      )cat.debug(str);)
 * @throws OZUserDefinedLogException
 */
public void traceLog(OZUserDefinedLogTarget target, OZConnection ozconn,
OZLog cat) throws OZUserDefinedLogException{
    try{
//      OZRDBLogTarget ref = (OZRDBLogTarget)target;
        Connection conn = ozconn.getConnection();
        PreparedStatement pstmt = conn.prepareStatement(QUERY);
        StringBuffer sb2 = new StringBuffer();
        if(target instanceof OZProcedureLogTarget) {
            sb2.append(procedureStore(target));
        }else if(target instanceof OZRDBLogTarget){
            sb2.append(rdbstore(target));
        }else if(target instanceof OZUDSLogTarget) {
            sb2.append(udsStore(target));
        }else if(target instanceof OZFileStoreLogTarget) {
            sb2.append(filestore(target));
        }else if(target instanceof OZHTTPStoreLogTarget) {
            sb2.append(httpStore(target));
        }else if(target instanceof OZXMLLogTarget) {
            sb2.append(xmlStore(target));
        }else if(target instanceof OZSOAPLogTarget) {
            sb2.append(soapStore(target));
        }else if(target instanceof OZSAPLogTarget) {
            sb2.append(sapStore(target));
        }else if(target instanceof OZClearQuestLogTarget) {
            sb2.append(clearQuestStore(target));
        }
    }

    ////

```

```
OZDMLogTarget ref = (OZDMLogTarget)target;
StringBuffer sb = new StringBuffer();

HttpServletRequest request = ref.getRequest();
if(request != null){
    sb.append("[Request Parameter=");
    Enumeration en = request.getParameterNames();
    while(en.hasMoreElements()){
        String key = (String)en.nextElement();
        String value = request.getParameter(key);
        sb.append("(");
        sb.append(key + ", " + value);
        sb.append(")");
    }
    sb.append("]");
    HttpSession session = request.getSession(false);
    if(session != null){
        sb.append("[Session Attribute=");
        en = session.getAttributeNames();
        while(en.hasMoreElements()){
            String key = (String)en.nextElement();
            sb.append("(");
            sb.append(key);
            sb.append(", ");
            sb.append(session.getAttribute(key));
            sb.append(")");
        }
        sb.append("]");
    }
}

String ipAddress = (ref.getIPAddress() == null) ? "" :
ref.getIPAddress();
String userId = (ref.getUserID() == null) ? "" : ref.getUserID();
String odiName = (ref.getODIName() == null) ? "" : ref.getODIName();
String dataSetName = (ref.getDataSetName() == null) ? "" :
ref.getDataSetName();
String dataStoreName = (ref.getDataStoreName() == null) ? "" :
ref.getDataStoreName();

pstmt.setString(1, sb.toString());
pstmt.setString(2, sb.toString());
pstmt.setString(3, ipAddress);
pstmt.setString(4, userId);
pstmt.setString(5, odiName);
pstmt.setString(6, dataSetName);
pstmt.setString(7, dataStoreName);
```

```

        pstmt.executeUpdate();
        pstmt.close();
    }
    catch(Exception ex){
        cat.error(ex.getMessage(), ex);
        throw new OZUserDefinedLogException(ex.getMessage());
    }
}

/**
 * UDL DB OZ Server Pool alias
 * @return String
 */
public String getOZDBAlias(){
    return "qa_ms";
}

private SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss.SSS");

/**
 * 가
 * @return String
 */
private String getDateTime() {
    return sdf.format(new java.util.Date(System.currentTimeMillis()));
}

/**
 *
 * @param tableName String
 * @throws SQLException
 * @return boolean
 */
private boolean isExistsTable(Connection conn, String tableName) throws
SQLException
{
    DatabaseMetaData meta = null;
    ResultSet rs = null;

    try { //////////////////////////////////////

```

```
    if (conn != null) {
        meta = conn.getMetaData();
    } else {
        throw new SQLException("Connection is null");
    }

    String[] tmpType = {"TABLE", "VIEW", "SYSTEM TABLE", "GLOBAL
TEMPORARY",
                        "LOCAL TEMPORARY", "ALIAS", "SYNONYM"};
    rs = meta.getTables(null, null, null, tmpType);

    while (rs.next()) {
        if (rs.getString(3).equals(tableName)) {
            return true;
        }
    }
} catch (SQLException sqlEx) {
    throw sqlEx;
} finally {
    if (rs != null)
        rs.close();
    rs = null;
    meta = null;
}
return false;
}

public String rdbstroe(OZUserDefinedLogTarget target){
    OZRDBLogTarget ref = (OZRDBLogTarget)target;
    StringBuffer sb = new StringBuffer();
    String dbAlias = (ref.getDBAlias() == null)? "" : ref.getDBAlias();
    sb.append("[DB Alias=");
    sb.append(dbAlias);
    sb.append("]");

    String crudFlagName = (ref.getCRUDFlagName() == null) ? "" :
ref.getCRUDFlagName();
    sb.append("[CRUDType=");
    sb.append(crudFlagName);
    sb.append("]");
    sb.append("[isPrepared=");
    sb.append(ref.isPrepared());
    sb.append("]");
    if(ref.getPreparedQueryValue() != null){
        sb.append("[QueryPreparedValue=");
        Vector vec = ref.getPreparedQueryValue();
        if (crudFlagName.equals("select")) {
```

```

        for (int i = 0; i < vec.size(); i++) {
            sb.append(vec.elementAt(i));
        }
    }else {
        for (int i = 0; i < vec.size(); i++) {
            OZDACData data = (OZDACData)vec.elementAt(i);
            sb.append(data.toStringData());
            sb.append(";");
        }
    }
    sb.append("]");
}
String query = (ref.getDBExecuteQuery() == null) ? "" :
ref.getDBExecuteQuery();
sb.append("[Query=");
sb.append(query);
sb.append("]");
Long queryExec = ref.getQueryExecuteTime();
sb.append("[Query exec=");
sb.append(queryExec);
sb.append("]");
String queryState = (ref.getExecuteQueryState() == null) ? "" :
ref.getExecuteQueryState();
sb.append("[QueryState=");
sb.append(queryState);
sb.append("]");

return sb.toString();
}

public String udsStore(OZUserDefinedLogTarget target){
    OZUDSLogTarget ref = (OZUDSLogTarget)target;
    StringBuffer sb = new StringBuffer();

    sb.append("[ExecuteCommand=");
    sb.append(ref.getExecuteCommand());
    sb.append("]");
    sb.append("[CRUDFI agName=");
    sb.append(ref.getCRUDFI agName());
    sb.append("]");
    sb.append("[CommandExecuteTi me=");
    sb.append(ref.getCommandExecuteTi me());
    sb.append("]");
    sb.append("[CommandState=");
    sb.append(ref.getCommandState());
    sb.append("]");

    return sb.toString();
}

```

```
}

public String procedureStore(OZUserDefinedLogTarget target){
    OZProcedureLogTarget ref = (OZProcedureLogTarget)target;
    StringBuffer sb = new StringBuffer();

    sb.append("[DBAliases=");
    sb.append(ref.getDBAliases());
    sb.append("]");
    sb.append("[ProcedureName=");
    sb.append(ref.getProcedureName());
    sb.append("]");
    sb.append("[ParameterList=");
    ArrayList list = ref.getParameterList();
    for(int i=0; i<list.size(); i++){
        OZProcedureParameter param = (OZProcedureParameter)list.get(i);
        sb.append(param.getValue());
        sb.append(", ");
        sb.append(param.getDataTypeName());
        sb.append("; ");
    }
    sb.append(ref.getParameterList());
    sb.append("]");
    sb.append("[ProcedureExecuteTime=");
    sb.append(ref.getProcedureExecuteTime());
    sb.append("]");
    sb.append("[ProcedureState=");
    sb.append(ref.getProcedureState());
    sb.append("]");

    return sb.toString();
}

public String xmlStore(OZUserDefinedLogTarget target){
    StringBuffer sb = new StringBuffer();
    OZXMLLogTarget ref = (OZXMLLogTarget)target;

    sb.append("[URL=");
    sb.append(ref.getURL());
    sb.append("]");
    sb.append("[URLAccessTime=");
    sb.append(ref.getURLAccessTime());
    sb.append("]");
    sb.append("[URLAccessState=");
    sb.append(ref.getURLAccessState());
    sb.append("]");

    return sb.toString();
}
```

```
}

public String soapStore(OZUserDefinedLogTarget target){
    StringBuffer sb = new StringBuffer();
    OZSOAPLogTarget ref = (OZSOAPLogTarget)target;

    sb.append("[ServiceName=");
    sb.append(ref.getServiceName());
    sb.append("]");
    sb.append("[Port=");
    sb.append(ref.getPort());
    sb.append("]");
    sb.append("[EndPoint=");
    sb.append(ref.getEndPoint());
    sb.append("]");
    sb.append("[ServiceExecuteTime=");
    sb.append(ref.getServiceExecuteTime());
    sb.append("]");
    sb.append("[ServiceState=");
    sb.append(ref.getServiceState());
    sb.append("]");
    sb.append("[RequestXML=");
    sb.append(ref.getRequestXML());
    sb.append("]");

    return sb.toString();
}

public String sapStore(OZUserDefinedLogTarget target){
    StringBuffer sb = new StringBuffer();
    OZSAPLogTarget ref = (OZSAPLogTarget)target;

    sb.append("[FunctionName=");
    sb.append(ref.getFunctionName());
    sb.append("]");
    sb.append("[FunctionType=");
    sb.append(ref.getFunctionType());
    sb.append("]");
    sb.append("[ResultSetTypes=");
    sb.append(ref.getResultSetTypes());
    sb.append("]");
    sb.append("[InputParameters=");
    sb.append(ref.getInputParameters());
    sb.append("]");
    sb.append("[FunctionExecuteTime=");
    sb.append(ref.getFunctionExecuteTime());
    sb.append("]");
    sb.append("[FunctionState=");
```

```
sb.append(ref.getFunctionState());
sb.append("]");

return sb.toString();
}

public String clearQuestStore(OZUserDefinedLogTarget target){
    StringBuffer sb = new StringBuffer();
    OZClearQuestLogTarget ref = (OZClearQuestLogTarget)target;

    sb.append("[ExecuteType=");
    sb.append(ref.getExecuteType());
    sb.append("]");
    sb.append("[QuerySubType=");
    sb.append(ref.getQuerySubType());
    sb.append("]");
    sb.append("[Query=");
    sb.append(ref.getQuery());
    sb.append("]");
    sb.append("[QueryExecuteTime=");
    sb.append(ref.getQueryExecuteTime());
    sb.append("]");
    sb.append("[QueryState=");
    sb.append(ref.getQueryState());
    sb.append("]");

    return sb.toString();
}

public String httpStore(OZUserDefinedLogTarget target){
    OZHTTPStoreLogTarget ref = (OZHTTPStoreLogTarget)target;
    StringBuffer sb = new StringBuffer();

    sb.append("[URL=");
    sb.append(ref.getURL());
    sb.append("]");
    sb.append("[HTTPAccessTime=");
    sb.append(ref.getHTTPAccessTime());
    sb.append("]");
    sb.append("[HTTPAccessState=");
    sb.append(ref.getHTTPAccessState());
    sb.append("]");

    return sb.toString();
}

public String fileStore(OZUserDefinedLogTarget target){
    StringBuffer sb = new StringBuffer();
```

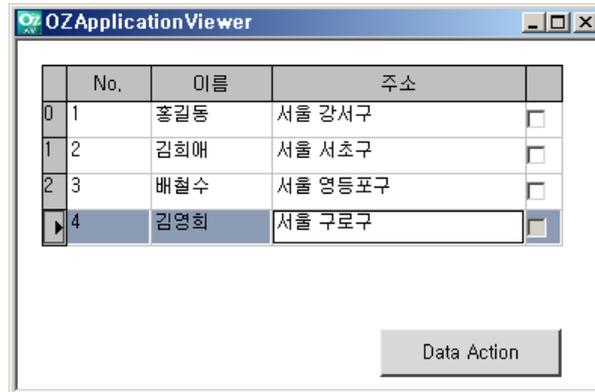


```
#
Sample.vendor=MSSQL
Sample.serverAddress=127.0.0.1
Sample.user=user
Sample.password=userpw
Sample.portNo=1433
Sample.dbName=DBName
Sample.maxconns=20
Sample.initconns=5
Sample.timeout=5
Sample.doConnecti onCheck=fal se
```



가 [Data Action]

가



ODI

T...	DB...	IP	USERID	CR...	preParedValue	Query	e...	q...	ODINAME	DataS...	DataStoreName
20...	qa_ms	12...	admin	insert	.[QueryPreparedValue=4;김영희];...	insert into REQ397 (id,...	109	true	/DataAction.odi	Sample	DB
20...	qa_ms	12...	admin	select	.[QueryPreparedValue=]	select * from REQ397	0	true	/DataAction.odi	Sample	DB
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

**UDL 3 -**

OZ UDL monitor.log

■

UDL monitor.log

java

oz.udl.db.OZUserDefinedMonitorLogger

"ozudl.jar"

( db.properties

DB

"L

Sample"

"OZ\_MONITOR\_JAVA"

.)

```
package oz.udl.db;

import oz.udl.*;
import oz.framework.db.*;
import com.forcs.log4oz.OZLog;
```

```
import oz.framework.dac.OZDACData;

import java.sql.*;
import java.util.*;
import java.text.*;
import javax.servlet.http.*;

//          가
public class OZUserDefinedMonitorLogger implements OZUserDefinedLogger {
    private final static String TABLENAME = "OZ_MONITOR_JAVA";
    private final static String QUERY = "insert into "+TABLENAME +
        " values (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
};

public OZUserDefinedMonitorLogger(){}

/**
 *          가
 * @throws OZUserDefinedLogException
 */
public void init(OZConnection ozconn, OZLog cat) throws
OZUserDefinedLogException{
    try{
        Connection conn = ozconn.getConnection();

        if(!isExistsTable(conn, TABLENAME)){
            String createQuery =
                "create table "+TABLENAME+" ( " +
                "MARK varchar(5) NOT NULL, " +
                "THREAD_Name varchar(255) NOT NULL, " +
                "SERVICE_TIME bigint NOT NULL, " +
                "FREE_MEMORY bigint NULL, " +
                "TOTAL_MEMORY bigint NULL, " +
                "SERVICE_CODE smallint NULL, " +
                "SERVICE_STATUS smallint NULL, " +
                "SERVICE_PARAMETERS text NULL, " +
                "DB_SESSION_ID varchar(255) NULL, " +
                "EXECUTION_TIME bigint NULL, " +
                "CLIENT_IP varchar(50) NULL, " +
                "SESSION_INFO text NULL ) ";

            PreparedStatement pstmt = conn.prepareStatement(createQuery);

            pstmt.executeUpdate();

            createQuery = "CREATE INDEX " + TABLENAME + "_idx_service_time_id
ON "
                + TABLENAME + " (SERVICE_TIME)";
```

```

        pstmt.close();
        pstmt = conn.prepareStatement(createQuery);
        pstmt.executeUpdate();
        pstmt.close();
    }
}
catch(Exception ex){
    cat.error(ex.getMessage(), ex);
    throw new OZUserDefinedLogException(ex.getMessage());
}
}

/**
 *
 * @param target OZUserDefinedLogTarget
 * @param ozconn OZConnection
 * @param cat OZLog (      trace      param      )cat.debug(str);)
 * @throws OZUserDefinedLogException
 */
public void traceLog(OZUserDefinedLogTarget target, OZConnection ozconn,
OZLog cat) throws OZUserDefinedLogException{
    try{

        OZMonitorLogTarget ref = (OZMonitorLogTarget)target;
        Connection conn = ozconn.getConnection();
        PreparedStatement pstmt = conn.prepareStatement(QUERY);

        pstmt.setString(1, ref.getMark());
        pstmt.setString(2, ref.getThreadName());
        pstmt.setLong(3, ref.getServiceTime());
        pstmt.setLong(4, ref.getFreeMemory());
        pstmt.setLong(5, ref.getTotalMemory());
        pstmt.setInt(6, ref.getServiceCode());
        pstmt.setInt(7, ref.getServiceStatus());
        pstmt.setString(8, ref.getServiceParameter());
        pstmt.setString(9, ref.getDBSessionID());
        pstmt.setLong(10, ref.getExecuteTime());
        pstmt.setString(11, ref.getIPAddress());

        StringBuffer sb = new StringBuffer();
        HttpServletRequest request = ref.getHttpRequest();
        if (request != null) {
            sb.append("[Request Parameter=");

            Enumeration en = request.getParameterNames();
            while(en.hasMoreElements()) {
                String key = (String)en.nextElement();

```

```

        String value = request.getParameter(key);
        sb.append("("); sb.append(key+", "+value); sb.append(")");
    }
    sb.append("]");

    HttpSession session = request.getSession(false);
    if (session != null) {

        sb.append("[Session Attribute=");

        en = session.getAttributeNames();
        while(en.hasMoreElements()) {
            String key = (String)en.nextElement();
            sb.append("(");
            sb.append(key);
            sb.append(", ");
            sb.append(session.getAttribute(key));
            sb.append(")");
        }
        sb.append("]");
    }
    pstmt.setString(12, sb.toString());
} else {
    pstmt.setString(12, null);
}

pstmt.executeUpdate();
pstmt.close();
}
catch(Exception ex){
    cat.error(ex.getMessage(), ex);
    throw new OZUserDefinedLogException(ex.getMessage());
}
}

/**
 * UDL DB          OZ Server Pool alias
 * @return String
 */
public String getOZDBAlias(){
    return "Sample";
}

private SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss.SSS");

/**

```

```

*          가
* @return String
*/
private String getDateTIme() {
    return sdf.format(new java.util.Date(System.currentTimeMillis()));
}

/**
*
* @param tableName String
* @throws SQLException
* @return boolean
*/
private boolean isExistsTable(Connection conn, String tableName) throws
SQLException
{
    DatabaseMetaData meta = null;
    ResultSet rs = null;

    try {
        if (conn != null) {
            meta = conn.getMetaData();
        } else {
            throw new SQLException("Connection is null");
        }

        String[] tmpType = {"TABLE", "VIEW", "SYSTEM TABLE", "GLOBAL
TEMPORARY",
                            "LOCAL TEMPORARY", "ALIAS", "SYNONYM"};
        rs = meta.getTables(null, null, null, tmpType);

        while (rs.next()) {
            if (rs.getString(3).equals(tableName)) {
                return true;
            }
        }
    } catch (SQLException sqlEx) {
        throw sqlEx;
    } finally {
        if (rs != null)
            rs.close();
        rs = null;
        meta = null;
    }
    return false;
}
}

```



- 가 /log/monitor.log IP

```

start 120224 1184045287342130799136 133955584 null null null null null null
end 120224 1184045287358130664648 133955584 142 9001 item name; Car.odi , item
type; 10000, category name; /127.0.0.1 16
start 4523271 1184045782686130517528 133955584 null null null null null null
end 4523271 1184045782717130399088 133955584 122 9001 item id; 333960544
127.0.0.1 31
    
```

- IP

MARK	THREA...	SERVICE_TIME	FREE_MEMORY	TOTAL_MEMORY	SE...	SERV...	SERVICE_PARAMET...	D...	EXE...	CLIENT_IP	SESSION_INFO
start	120224	1184045287342	130799136	133955584	-1	-1			-1	127.0.0.1	NULL
end	120224	1184045287358	130664648	133955584	142	9001	car.odi,item type;10000!		16	127.0.0.1	NULL
end	4523271	1184045782717	130399088	133955584	122	9001	item id;333960544		31	127.0.0.1	NULL
end	3981922	1184045782748	130241848	133955584	174	9001	item name;DataActio...		0	127.0.0.1	NULL



C

가/



"PDF417" 가



### DLL

```
: ozc_ .dll  
" PDF 417" "ozc_pdf417.dll"
```

launch.cfg classpath 가

ozc\_pdf417.dll WIN32 API LoadLibrary

DLL ozuser.zip dll zip  
( : ozviewer.idf ) ZTransfer  
(\* .idf) "

: PDF SVG 가  
. Excel, Word, Powerpoint, HTML

## C

■

- GetNewInstance

<b>Prototype</b>	DWORD GetNewInstance()
<b>Definition</b>	
<b>Argument</b>	
<b>Return</b>	0(NULL)

- GetCopyInstance

<b>Prototype</b>	DWORD GetCopyInstance(DWORD src)
<b>Definition</b>	
<b>Argument</b>	<i>src</i>
<b>Return</b>	0(NULL)

- DeleteInstance

<b>Prototype</b>	void DeleteInstance(DWORD src)
<b>Definition</b>	
<b>Argument</b>	<i>src</i>
<b>Return</b>	

■

**Serialize**

- getAttrListLength

<b>Prototype</b>	int getAttrListLength(DWORD src)
<b>Definition</b>	가
<b>Argument</b>	<i>src</i>

<b>Return</b>	, -1
---------------	------

- getAttrList

<b>Prototype</b>	BOOL getAttrList(DWORD src, char** attrs, const int length)
<b>Definition</b>	attrs
<b>Argument</b>	<i>src</i>
	<i>attrs</i> Pointer
	<i>length</i> getAttrListLength
<b>Return</b>	true, false

- getAttrLength

<b>Prototype</b>	int getAttrLength(DWORD src, const char* name)
<b>Definition</b>	.
<b>Argument</b>	<i>src</i>
	<i>name</i>
<b>Return</b>	, ( ) -1

- getAttr

<b>Prototype</b>	BOOL getAttr(DWORD src, const char* name, char* value, const int value_length)
<b>Definition</b>	name value .
	<i>src</i>
<b>Argument</b>	<i>name</i>
	<i>value</i>
	<i>value_length</i> getAttrLength
<b>Return</b>	true, ( ) false

- setAttr

<b>Prototype</b>	BOOL setAttr(DWORD src, const char* name, const char* value)
<b>Definition</b>	.
<b>Argument</b>	<i>src</i>

	<i>name</i>
	<i>value</i>
<b>Return</b>	, ( ) -1

- writeMe

<b>Prototype</b>	char* writeMe(DWORD src, int * length)
<b>Definition</b>	serialize .
	<i>src</i>
<b>Argument</b>	<i>length</i> serialize binary data int pointer
<b>Return</b>	serialize binary data pointer , NULL

- readMe

<b>Prototype</b>	void readMe(DWORD src, const char* pData, const int length)
<b>Definition</b>	serialize binary data .
	<i>src</i>
<b>Argument</b>	<i>pData</i> binary data <i>length</i>
<b>Return</b>	



- paint

<b>Prototype</b>	void paint(DWORD src, HDC hDC, LPCTSTR data, const float x, const float y, const float w, const float h, const float scale)
<b>Definition</b>	.
<b>Argument</b>	<i>src</i>
	<i>hDC</i> (WIN32 API )
	<i>data</i> Caption
	<i>x</i> x
	<i>y</i> y
	<i>w</i>
	<i>h</i>

*scale*

**Return**

- paintEX

**Prototype** void paintEX(\_\_cdecl \*paintEX)(DWORD src, HDC hDC, LPCTSTR data, const float x, const float y, const float w, const float h, const float scale const float x\_offset, const float y\_offset)

**Definition** . 가 offset

*src*

*hDC* (WIN32 API )

*data* Caption

*x* x

*y* y

**Argument**

*w*

*h*

*scale*

*x\_offset* 가 x offset

*y\_offset* 가 y offset

**Return**

- print

**Prototype** void print(DWORD src, HDC hDC, LPCTSTR data, const float x, const float y, const float w, const float h, const float scale, const float x\_offset, const float y\_offset)

**Definition** .

**Argument** *src*

*hDC* (WIN32 API )

*data* Caption

*x* x

*y* y

*w*

*h*

*scale*

	<i>x_offset</i>	x Offset
	<i>y_offset</i>	x Offset
<b>Return</b>		

- getAutoSize

<b>Prototype</b>	void getAutoSize(DWORD src, HDC hDC, LPCTSTR data, float * w, float * h)	
<b>Definition</b>	가 ,	
<b>Argument</b>	<i>src</i>	
	<i>hDC</i>	(WIN32 API )
	<i>data</i>	Caption
	<i>w</i>	pointer
	<i>h</i>	pointer
<b>Return</b>	w, h	가 ,

C++ dll PDF 417

■ PDF417.h

```
// hanja.h : main header file for the HANJA DLL
//
//
#if !defined(AFX_CPdf417App_H__7FDF11DA_5BF8_4C07_9BCF_BB1C08A763FA__INCLUDED_)
#define AFX_CPdf417App_H__7FDF11DA_5BF8_4C07_9BCF_BB1C08A763FA__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

#ifndef __AFXWIN_H__
    #error include 'stdafx.h' before including this file for PCH
#endif

#include <resource.h> // main symbols
////////////////////////////////////
```

```

////////////////////////////////////
////////////////////////////////////

////////////////////////////////////

////////////////////////////////////
// CPdf417App
// See CPdf417App.cpp for the implementation of this class
//

class CPdf417App : public CWinApp
{
public:
    CPdf417App();

// Overrides
// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CPdf417App)
//}}AFX_VIRTUAL
//{{AFX_MSG(CPdf417App)
// NOTE - the ClassWizard will add and remove member functions here.
// DO NOT EDIT what you see in these blocks of generated code !
//}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

////////////////////////////////////

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before
the previous line.

#endif
// !defined(AFX_CPdf417App_H__7FDF11DA_5BF8_4C07_9BCF_BB1C08A763FA__INCLUDED_)

```

■ **PDF417.cpp**

```

// OZPdf417Comp.cpp : Defines the initialization routines for the DLL.
//

#include <stdafx.h>
#include <pdf417.h>
#include <CString.h>

#include "pdfapi.h"

```

```

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

//
// Note!
//
// If this DLL is dynamically linked against the MFC
// DLLs, any functions exported from this DLL which
// call into MFC must have the AFX_MANAGE_STATE macro
// added at the very beginning of the function.
//
// For example:
//
// extern "C" BOOL PASCAL EXPORT ExportedFunction()
// {
//     AFX_MANAGE_STATE(AfxGetStaticModuleState());
//     normal function body here
// }
//
// It is very important that this macro appear in each
// function, prior to any calls into MFC. This means that
// it must appear as the first statement within the
// function, even before any object variable declarations
// as their constructors may generate calls into the MFC
// DLL.
//
// Please see MFC Technical Notes 33 and 58 for additional
// details.
//

////////////////////////////////////
// CHanjaApp

BEGIN_MESSAGE_MAP(CPdf417App, CWinApp)
//{{AFX_MSG_MAP(CPdf417App)
// NOTE - the ClassWizard will add and remove mapping macros here.
// DO NOT EDIT what you see in these blocks of generated code!
//}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
// CPdf417App construction

```

```
CPdf417App : CPdf417App()
{
    // TODO: add construction code here,
    // Place all significant initialization in InitInstance
}

////////////////////////////////////
// The one and only CPdf417App object

CPdf417App theApp;

class OZPdf417Comp{
public:
    OZPdf417Comp();
    OZPdf417Comp(OZPdf417Comp & cp);
    virtual ~OZPdf417Comp();
    int getAttrListLength();
    BOOL getAttrList(WCHAR** attrs, const int length);
    int getAttrLength(CString name);
    BOOL getAttr(CString name, WCHAR* value, const int value_length);
    BOOL setAttr(CString name, CString value);
    char* writeMe(int * length);
    void readMe(const char* pData, const int length);
    void paint(HDC hDC, CString data, const float x, const float y, const
float w, const float h, const float scale);
    void print(HDC hDC, CString data, const float x, const float y, const float
w, const float h, const float scale, const float x_offset, const float
y_offset);
    void getAutoSize(HDC hDC, CString data, float * w, float * h);
private:
    int rowNum;
    int colNum;
    int ecc;
    int xScale;
    int yScale;
};

////////////////////////////////////
#ifdef __cplusplus
extern "C"
{
#endif

_declspec(dllexport) DWORD __cdecl GetNewInstance()
{
    return (DWORD)(void*)new OZPdf417Comp();
}
}
```

```

__declspec( dll export ) DWORD __cdecl GetCopyInstance(DWORD src)
{
    return (DWORD)(void*)new OZPdf417Comp*((OZPdf417Comp *) (void *)src);
}
__declspec( dll export ) void __cdecl DeleteInstance(DWORD src)
{
    delete ((OZPdf417Comp *) (void *)src);
}
/////////////////////////////////////////////////////////////////
__declspec( dll export ) int __cdecl GetAttrListLength(DWORD src)
{
    return ((OZPdf417Comp *) (void *)src)->GetAttrListLength();
}
__declspec( dll export ) BOOL __cdecl GetAttrList(DWORD src, WCHAR** attrs,
const int length){
    return ((OZPdf417Comp *) (void *)src)->GetAttrList(attrs, length);
}
__declspec( dll export ) int __cdecl GetAttrLength(DWORD src, const WCHAR* name)
{
    return ((OZPdf417Comp *) (void *)src)->GetAttrLength(CString(name));
}
__declspec( dll export ) BOOL __cdecl GetAttr(DWORD src, const WCHAR* name,
WCHAR* value, const int value_length){
return ((OZPdf417Comp *) (void *)src)->GetAttr(CString(name), value,
value_length);
}
__declspec( dll export ) BOOL __cdecl SetAttr(DWORD src, const WCHAR* name,
const WCHAR* value)
{
return ((OZPdf417Comp *) (void *)src)->SetAttr(CString(name), CString(value));
}
__declspec( dll export ) char* __cdecl WriteMe(DWORD src, int * length)
{
    return ((OZPdf417Comp *) (void *)src)->WriteMe(length);
}
__declspec( dll export ) void __cdecl ReadMe(DWORD src, const char* pData, const
int length)
{
    ((OZPdf417Comp *) (void *)src)->ReadMe(pData, length);
}
__declspec( dll export ) void __cdecl Paint(DWORD src, HDC hDC, LPCWSTR data,
const float x, const float y, const float w, const float h, const float scale)
{
    ((OZPdf417Comp *) (void *)src)->Paint(hDC, data, x, y, w, h, scale);
}
__declspec( dll export ) void __cdecl Print(DWORD src, HDC hDC, LPCWSTR data,
const float x, const float y, const float w, const float h, const float scale,
const float x_offset, const float y_offset)

```

```
{
((OZPdf417Comp *) (void *)src)->print(hDC, data, x, y, w, h, scale, x_offset,
y_offset);
}
__declspec( dll export ) void __cdecl getAutoSize(DWORD src, HDC hDC, LPCWSTR
data, float * w, float * h)
{
((OZPdf417Comp *) (void *)src)->getAutoSize(hDC, data, w, h);
}
#ifdef __cplusplus
}
#endif
static PDFObject g_MacroObjIn, g_MacroObjOut;

PDFSI ZET Encode(CMBString dataStr, unsigned short rowNum, unsigned short colNum,
unsigned short ecc, UINT xScale, UINT yScale)
{
PDFSI ZET nInputLen = 0;
UINT nDPI = 300;

PDFSetDefaults();
PDFBinaryMode( false );
PDFSetSymbolStyle( STYLE_NORMAL );
PDFSetECCLevel( ecc );
if(rowNum == 0 && colNum == 0){
PDFSetAspect( "1:2" );
}else{
PDFSetDimType( USE_FIXED );
PDFSetRowCol( rowNum, colNum );
}

CString szAspect;
szAspect.Format(_T("%d:%d"), xScale+yScale-1, xScale );
PDFSetRowHeight((LPCSTR)(CMBString)szAspect );

MPDFDi sable();
nInputLen = (PDFSI ZET)dataStr.GetLength();
MakeMemoryObject( &g_MacroObjIn, PDFINPUT, (LPSTR)(LPCSTR)dataStr );
uint16 wParam = 0;
uint32 lParam = MakeLParam( nDPI, xScale );
wParam |= DIB_ADDFILEHEADER;
/*
if ( pSettings->nWShave )
wParam |= ShaveWidthOption( pSettings->nWShave );
if ( pSettings->nHShave )
wParam |= ShaveHeightOption( pSettings->nHShave );
*/
PDFOutputAsDIB( lParam, wParam );
return nInputLen;
}
```

```

}
int EncodeData(CMBString dataStr, BYTE * rt, unsigned short rowNum, unsigned
short colNum, unsigned short ecc, UINT xScale, UINT yScale)
{
PDFSI ZET nInputLen = Encode(dataStr, rowNum, colNum, ecc, xScale, yScale);
BOOL fStatus;
PDFSI ZET rtsize = -1;
// MakeFilenameObject( &g_MacroObjOut, PDFOUTPUT, fileStr);
  MakeMemoryObject( &g_MacroObjOut, PDFOUTPUT, (LPVOID) rt);

  if ( PDFEncode( &g_MacroObjIn, nInputLen ) >= 0 ) {
    fStatus = (PDFMakeImage( &g_MacroObjOut, &rtsize ) == 0);
  }else{
    fStatus = FALSE;
  }
  if(!fStatus){
    rtsize = -1;
    PDFAbort( &g_MacroObjIn );
    PDFAbort( &g_MacroObjOut);
  }
// printf(dataStr);
  return rtsize;
}

int EncodeFile(CMBString dataStr, CMBString fileStr, unsigned short rowNum,
unsigned short colNum, unsigned short ecc, UINT xScale, UINT yScale)
{
PDFSI ZET nInputLen = Encode(dataStr, rowNum, colNum, ecc, xScale, yScale);
BOOL fStatus;
PDFSI ZET rtsize = -1;
MakeFilenameObject( &g_MacroObjOut, PDFOUTPUT, fileStr);
// MakeMemoryObject( &g_MacroObjOut, PDFOUTPUT, (LPVOID) rt);

  if ( PDFEncode( &g_MacroObjIn, nInputLen ) >= 0 ) {
    fStatus = (PDFMakeImage( &g_MacroObjOut, &rtsize ) == 0);
  }else{
    fStatus = FALSE;
  }
  if(!fStatus){
    rtsize = -1;
    PDFAbort( &g_MacroObjIn );
    PDFAbort( &g_MacroObjOut);
  }
// printf(dataStr);
// printf(fileStr);
  return rtsize;
}

```

```
LPPICTURE EncodePrint(CMBString dataStr, unsigned short rowNum, unsigned short
colNum, unsigned short ecc, UINT xScale, UINT yScale)
{
PDFSI ZET nInputLen = Encode(dataStr, rowNum, colNum, ecc, xScale, yScale);
BOOL fStatus;
PDFSI ZET rtsize = -1;

HGLOBAL hGlobal = NULL;
DWORD dwFileSize = 1024*10;

LPVOID pvData = NULL;
// alloc memory based on file size
hGlobal = GlobalAlloc(GMEM_MOVEABLE, dwFileSize);
if(NULL == hGlobal){
    PDFAbort( &g_MacroObjIn );
    PDFAbort( &g_MacroObjOut);
    return NULL;
}

pvData = GlobalLock(hGlobal);
if(NULL == pvData){
    PDFAbort( &g_MacroObjIn );
    PDFAbort( &g_MacroObjOut);
    return NULL;
}

    MakeMemoryObject( &g_MacroObjOut, PDFOUTPUT, pvData);

if ( PDFEncode( &g_MacroObjIn, nInputLen ) >= 0 ) {
    fStatus = (PDFMakeImage( &g_MacroObjOut, &rtsize ) == 0);
}else{
    fStatus = FALSE;
}
if(!fStatus){
    PDFAbort( &g_MacroObjIn );
    PDFAbort( &g_MacroObjOut);
    GlobalUnlock(hGlobal);
    return NULL;
}

GlobalUnlock(hGlobal);

LPSTREAM pstm = NULL;
// create IStream* from global memory
HRESULT hr = CreateStreamOnHGlobal(hGlobal, TRUE, &pstm);
if(!SUCCEEDED(hr) || pstm == NULL){
    PDFAbort( &g_MacroObjIn );
    PDFAbort( &g_MacroObjOut);
}
```

```

        return NULL;
    }

    // Create IPicture from image file
    LPPICTURE gpPicture = NULL;
    hr = ::OleLoadPicture(pstm, dwFileSize, FALSE, IID_IPicture, (LPVOID *)&gpPicture);

    if(!SUCCEEDED(hr) || gpPicture == NULL){
        BOOL b1 = hr == E_POINTER;
        BOOL b2 = hr == E_NOINTERFACE;
        BOOL b3 = hr == E_OUTOFMEMORY;
        BOOL b4 = hr == E_UNEXPECTED;
        pstm->Release();
        PDFAbort(&g_MacroObjIn);
        PDFAbort(&g_MacroObjOut);
        return NULL;
    }
    pstm->Release();

    return gpPicture;
}

CString _toString(int i){
    CString rst;
    rst.Format(_T("%d"), i);
    return (CString)rst;
}

////////////////////////////////////
OZPdf417Comp::OZPdf417Comp(){
    rowNum = 0;
    colNum = 8;
    ecc = 0;
    xScale = 1;
    yScale = 1;
}

OZPdf417Comp::OZPdf417Comp(OZPdf417Comp & comp){
    rowNum = comp.rowNum;
    colNum = comp.colNum;
    ecc = comp.ecc;
    xScale = comp.xScale;
    yScale = comp.yScale;
}

OZPdf417Comp::~OZPdf417Comp(){
}

int OZPdf417Comp::getAttrListLength(){
    return 5;
}

```

```

BOOL OZPdf417Comp::getAttrList(WCHAR** attrs, const int length){
    if(length < 0)return FALSE;
    int nIndex = 0;
    if(length == nIndex)return TRUE;
    attrs[nIndex++] = L"PDF417_ROWNUM";
    if(length == nIndex)return TRUE;
    attrs[nIndex++] = L"PDF417_COLNUM";
    if(length == nIndex)return TRUE;
    attrs[nIndex++] = L"PDF417_ECC";
    if(length == nIndex)return TRUE;
    attrs[nIndex++] = L"PDF417_XSCALE";
    if(length == nIndex)return TRUE;
    attrs[nIndex++] = L"PDF417_YSCALE";
    return TRUE;
}

int OZPdf417Comp::getAttrLength(CString name){
    CString value;
    if(name == _T("PDF417_ROWNUM")){
        value = _toString(rowNum);
    }else if(name == _T("PDF417_COLNUM")){
        value = _toString(colNum);
    }else if(name == _T("PDF417_ECC")){
        value = _toString(ecc);
    }else if(name == _T("PDF417_XSCALE")){
        value = _toString(xScale);
    }else if(name == _T("PDF417_YSCALE")){
        value = _toString(yScale);
    }else{
        return -1;
    }
    return value.GetLength()+1;
}

BOOL OZPdf417Comp::getAttr(CString name, WCHAR* valueBuffer, const int
valueLength){
    CString value;
    if(name == _T("PDF417_ROWNUM")){
        value = _toString(rowNum);
    }else if(name == _T("PDF417_COLNUM")){
        value = _toString(colNum);
    }else if(name == _T("PDF417_ECC")){
        value = _toString(ecc);
    }else if(name == _T("PDF417_XSCALE")){
        value = _toString(xScale);
    }else if(name == _T("PDF417_YSCALE")){
        value = _toString(yScale);
    }
}

```

```

////////////////////////////////////
    }else{
        return FALSE;
    }
    if(value.GetLength() >= value_Length){
        return FALSE;
    }
    wcsncpy(valueBuffer, value);
    return TRUE;
}
BOOL OZPdf417Comp::setAttr(CString name, CString value){
    if(name == _T("PDF417_ROWNUM")){
        rowNum = _ttoi(value);
    }else if(name == _T("PDF417_COLNUM")){
        colNum = _ttoi(value);
    }else if(name == _T("PDF417_ECC")){
        ecc = _ttoi(value);
    }else if(name == _T("PDF417_XSCALE")){
        xScale = _ttoi(value);
    }else if(name == _T("PDF417_YSCALE")){
        yScale = _ttoi(value);
    }else{
        return FALSE;
    }
    return TRUE;
}
char* OZPdf417Comp::writeMe(int * length){
    *length = 20;
    char* pData = new char[*length];
    int i = 0;
    pData[i++] = (char)((rowNum & 0xFF000000) >> 24);
    pData[i++] = (char)((rowNum & 0x00FF0000) >> 16);
    pData[i++] = (char)((rowNum & 0x0000FF00) >> 8);
    pData[i++] = (char)((rowNum & 0x000000FF));

    pData[i++] = (char)((colNum & 0xFF000000) >> 24);
    pData[i++] = (char)((colNum & 0x00FF0000) >> 16);
    pData[i++] = (char)((colNum & 0x0000FF00) >> 8);
    pData[i++] = (char)((colNum & 0x000000FF));

    pData[i++] = (char)((ecc & 0xFF000000) >> 24);
    pData[i++] = (char)((ecc & 0x00FF0000) >> 16);
    pData[i++] = (char)((ecc & 0x0000FF00) >> 8);
    pData[i++] = (char)((ecc & 0x000000FF));

    pData[i++] = (char)((xScale & 0xFF000000) >> 24);
    pData[i++] = (char)((xScale & 0x00FF0000) >> 16);
    pData[i++] = (char)((xScale & 0x0000FF00) >> 8);
}

```

```
pData[i++] = (char)((xScale & 0x000000FF));

pData[i++] = (char)((yScale & 0xFF000000) >> 24);
pData[i++] = (char)((yScale & 0x00FF0000) >> 16);
pData[i++] = (char)((yScale & 0x0000FF00) >> 8);
pData[i++] = (char)((yScale & 0x000000FF));

return pData;
}

void OZPdf417Comp::readMe(const char* pData, const int length){
    if(length != 20){
        AfxThrowUserException();
    }
    int i = 0;
    rowNum = ((pData[i++] << 24) + (pData[i++] << 16) + (pData[i++] << 8) +
(pData[i++] << 0));
    colNum = ((pData[i++] << 24) + (pData[i++] << 16) + (pData[i++] << 8) +
(pData[i++] << 0));
    ecc = ((pData[i++] << 24) + (pData[i++] << 16) + (pData[i++] << 8) +
(pData[i++] << 0));
    xScale = ((pData[i++] << 24) + (pData[i++] << 16) + (pData[i++] << 8) +
(pData[i++] << 0));
    yScale = ((pData[i++] << 24) + (pData[i++] << 16) + (pData[i++] << 8) +
(pData[i++] << 0));
}

void OZPdf417Comp::paint(HDC hDC, CUniString data, const float x, const float y,
const float w, const float h, const float scale){
    LPPICTURE gpPicture = EncodePrint((CMBString)data,
rowNum, colNum, ecc, xScale, yScale); //
    if(gpPicture == NULL) return;
    long hmWidth;
    long hmHeight;
    gpPicture->get_Width(&hmWidth);
    gpPicture->get_Height(&hmHeight);
    // convert metric to pixelsA
    int imageWidth;
    int imageHeight;
    imageWidth= MulDiv(hmWidth*72, GetDeviceCaps(hDC, LOGPIXELSX), 2540*96);
    imageHeight= MulDiv(hmHeight*72, GetDeviceCaps(hDC, LOGPIXELSY), 2540*96);
    imageWidth = (int)(imageWidth*scale);
    imageHeight = (int)(imageHeight*scale);
    RECT rc = { 0, 0, imageWidth, imageHeight };
    HRESULT result = gpPicture->Render(hDC, (int)(x*scale), (int)(y*scale) ,
imageWidth, imageHeight, 0, hmHeight, hmWidth, -hmHeight, &rc);

    gpPicture->Release();
}

void OZPdf417Comp::print(HDC hDC, CUniString data, const float x, const float y,
```

```

const float w, const float h, const float scale, const float x_offset, const
float y_offset){
    LPPICTURE gpPicture = EncodePrint((CMBString)data,
        rowNum, colNum, ecc, xScale, yScale); //
    if(gpPicture == NULL) return;
    long hmWidth;
    long hmHeight;
    gpPicture->get_Width(&hmWidth);
    gpPicture->get_Height(&hmHeight);
    // convert hi metric to pixelsA
    int imageWidth;
    int imageHeight;
    imageWidth= MulDiv(hmWidth, GetDeviceCaps(hdc, LOGPIXELSX), 2540);
    imageHeight= MulDiv(hmHeight, GetDeviceCaps(hdc, LOGPIXELSY), 2540);

    RECT rc = { 0, 0, imageWidth, imageHeight };
    HRESULT result = gpPicture->Render(hdc, (int)(x*scale+x_offset),
        (int)(y*scale+y_offset), imageWidth, imageHeight, 0, hmHeight, hmWidth, -
        hmHeight, &rc);

    gpPicture->Release();
}
void OZPdf417Comp::getAutoSize(HDC hdc, CString data, float * w, float * h){
    LPPICTURE gpPicture = EncodePrint((CMBString)data,
        rowNum, colNum, ecc, xScale, yScale); //
    if(gpPicture == NULL) return;
    long hmWidth;
    long hmHeight;
    gpPicture->get_Width(&hmWidth);
    gpPicture->get_Height(&hmHeight);
    // convert hi metric to pixelsA
    *w= (float)MulDiv(hmWidth*72, GetDeviceCaps(hdc, LOGPIXELSX), 2540*96);
    *h= (float)MulDiv(hmHeight*72, GetDeviceCaps(hdc, LOGPIXELSY), 2540*96);
    gpPicture->Release();
}

```

## Appendix 1. SchedulerCom

---



<b>Scheduler API</b>	<b>makePDF</b>	<b>export</b>	<b>ASP</b>
<b>SchedulerCom</b>	<b>makePDF</b>	<b>export</b>	<b>ASP</b>

## SchedulerCom

### ■ Com

- .Init

<b>Prototype</b>	.Init()
<b>Definition</b>	COM
<b>Argument</b>	

- .Clean

<b>Prototype</b>	.Clean()
<b>Definition</b>	COM
<b>Argument</b>	

### ■

- .SetServerType

<b>Prototype</b>	.SetServerType String ServerType
<b>Definition</b>	
<b>Argument</b>	<i>ServerType</i> "TCP" 가 "TCP" "Servlet" 가 "Servlet" ( )

- .SetServerIP

<b>Prototype</b>	.SetServerIP String IP
<b>Definition</b>	가 IP
<b>Argument</b>	<i>IP</i> IP ex) "127.0.0.1"

- .SetServerPort

<b>Prototype</b>	.SetServerPort String Port		
<b>Definition</b>	가	Port	.
<b>Argument</b>	<i>Port</i>	Port	ex) "8003"

- .SetServerURL

<b>Prototype</b>	.SetServerURL String URL		
<b>Definition</b>	가	URL	.
<b>Argument</b>	<i>URL</i>	URL	ex) "http://localhost:7001/oz/server"

■

- .SetSchedulerIP

<b>Prototype</b>	.SetSchedulerIP String IP		
<b>Definition</b>	가	IP	.
<b>Argument</b>	<i>IP</i>	IP	ex) "127.0.0.1"

- .SetSchedulerPort

<b>Prototype</b>	.SetSchedulerPort String Port		
<b>Definition</b>	가	Port	.
<b>Argument</b>	<i>Port</i>	Port	ex) "9521"

■

- .SetUser

<b>Prototype</b>	.SetUser String UserID		
<b>Definition</b>	가		.
<b>Argument</b>	<i>UserID</i>		ex) "admin"

- .SetPassword

<b>Prototype</b>	.SetPassword String Password		
<b>Definition</b>	가		.
<b>Argument</b>	<i>Password</i>		ex) "admin"

■ Key

- .SetProperty





"<ozd/html/jpg/xls/doc/svg/ txt/ppt/tif/csv>.filename"	" "	ex).setExportProperty "ozd.filename", "test.ozd"
"viewer.childcount"	" "	

Key  
 "child<index>."  
 ex) "child1.connection.server"  
 ex) "child1.connection.port"



- .MakePDF

<b>Prototype</b>	. MakePDF(String ExportType)		
<b>Definition</b>	pdf	:	"ViewType"
		"None"	
<b>Argument</b>	ExportType	"ATTACH"	가 pdf
		"NONE"	pdf

- .Export

<b>Prototype</b>	. Export()		
<b>Definition</b>	pdf, ozd, html, jpg, xls, doc, svg, txt, ppt, tif, csv	:	"ViewType"
		"None"	
<b>Argument</b>			

- .IsExportSucceeded

<b>Prototype</b>	. IsExportSucceeded()		
<b>Definition</b>			

Return	<i>true</i>
	<i>false</i>

■

- .ShowMessage

Prototype	. ShowMessage(String Msg)
Definition	.
Argument	<i>Msg</i>

## SchedulerCom

■

가

"SchedulerCom" MS

■

**SchedulerCOM.dll**

"SchedulerCom" "SchedulerCOM.dll"

"SchedulerCOM.dll" "C:\OZServer\Scheduler\"

```
regsvr32 "C:\OZServer\Scheduler\SchedulerCOM.dll"
```

■

MakePDF Export

"scheduler\_server.properties" "RepositoryFileRootPath"

### MakePDF

### ASP

```
<%
Dim Com
Set Com = Server.CreateObject("SchedulerCOM.CSchedulerCall.1")

If Not IsObject (Com) Then

    Response.Write("PDF File  
Response.End

Else

    With Com

        ' Init. COM
        .Init()
        ' TCP-Daemon Type Server
        .SetServerType "TCP"
        .SetServerIP "127.0.0.1"
        .SetServerPort "8003"

        ' Servlet Type Server
        ' .SetServerType "Servlet"
        ' .SetServerURL "http://localhost:7001/oz/server"

        ' Set Scheduler Info.
        .SetSchedulerIP "127.0.0.1"
        .SetSchedulerPort "9521"

        ' set User Info.
        .SetUser "admin"
        .SetPassword "admin"

        ' set Launch Type
        .SetProperty "launch_type", "Immediately"

        ' set Report Info.
        .SetProperty "report_name", "parameter_test.ozr"
        .SetProperty "category_name", "/"
        .SetProperty "export.confirmsave", "false"

        ' with NO parameter.
        .setProperty "parameter_count", "0"
```

```
' with parameters.  
' .setProperty "parameter_count", "4"  
' .setProperty "parameter_name_1", "[FORM].formparam1"  
' .setProperty "parameter_value_1", "PARAM 1"  
' .setProperty "parameter_name_2", "[FORM].formparam2"  
' .setProperty "parameter_value_2", "PARAM 2"  
' .setProperty "parameter_name_3", "parameter_test.odi param1"  
' .setProperty "parameter_value_3", "PARAM 3"  
' .setProperty "parameter_name_4", "parameter_test.odi param2"  
' .setProperty "parameter_value_4", "PARAM 4"  
  
' set Export Info.  
    .setExportProperty "pdf.filename", "PDF_TEST.pdf"  
  
.MakePDF("NONE")  
  
res = .IsExportSucceeded()  
if res = "true" then  
    .ShowMessage ("succeeded...")  
else  
    .ShowMessage ("failed...")  
end if  
  
.Clean()  
  
End With  
  
Set Com = Nothing  
  
End If  
</%>
```

## Export ASP

```
<%  
Dim Com  
Set Com = Server.CreateObject("SchedulerCOM.CSchedulerCall.1")  
  
If Not IsObject (Com) Then  
    Response.Write("File .")  
Response.End
```

```
Else
  With Com
    .Init()
    .SetServerType "TCP"
    .SetServerIP "127.0.0.1"
    .SetServerPort "8003"
    .SetSchedulerIP "127.0.0.1"
    .SetSchedulerPort "9521"
    .SetUser "admin"
    .SetPassword "admin"
    .SetProperty "launch_type", "Immediately"
    .setProperty "cfg.type", "new"

    .setExportProperty "connection.server", "127.0.0.1"
    .setExportProperty "connection.port", "8003"
    .setExportProperty "connection.reportName", "/parameter_test.ozr"
    .setExportProperty "connection.fetchtype", "BATCH"
    .setExportProperty "connection.pcount", "2"
    .setExportProperty "connection.args1=formparam1", "form1"
    .setExportProperty "connection.args2=formparam2", "form2"

    .setExportProperty "odi.parameter_test.args1", "odi param1=odi 1"
    .setExportProperty "odi.parameter_test.args2", "odi param2=odi 2"
    .setExportProperty "odi.parameter_test.pcount", "2"
    .setExportProperty "odi.odi names", "parameter_test"

    .setExportProperty "export.format",
      "ozd/html/jpg/xls/doc/svg/txt/ppt/tiff/csv"
    .setExportProperty "ozd.filename", "test.ozd"
    .setExportProperty "html.filename", "test.html"
    .setExportProperty "jpg.filename", "test.jpg"
    .setExportProperty "excel.filename", "test.xls"
    .setExportProperty "word.filename", "test.doc"
    .setExportProperty "svg.filename", "test.svg"
    .setExportProperty "text.filename", "test.txt"
    .setExportProperty "ppt.filename", "test.ppt"
    .setExportProperty "tiff.filename", "test.tiff"
    .setExportProperty "csv.filename", "test.csv"

    .setExportProperty "viewer.childcount", "1"
    .setExportProperty "child1.connection.server", "127.0.0.1"
    .setExportProperty "child1.connection.port", "8003"
    .setExportProperty "child1.connection.reportName",
```

```

"/parameter_test.ozr"
.setExportProperty "child1.connection.fetchtype", "BATCH"
.setExportProperty "child1.connection.pcount", "2"
.setExportProperty "child1.connection.args1=formparam1", "form1"
.setExportProperty "child1.connection.args2=formparam2", "form2"

.setExportProperty "child1.odi.parameter_test.args1", "odi param1=odi 1"
.setExportProperty "child1.odi.parameter_test.args2", "odi param2=odi 2"
.setExportProperty "child1.odi.parameter_test.pcount", "2"
.setExportProperty "child1.odi.odi names", "parameter_test"

.setExportProperty "child1.export.format",
                    "ozd/html/jpg/xls/doc/svg/txt/ppt/tif/csv"
.setExportProperty "child1.ozd.filename", "child_test.ozd"
.setExportProperty "child1.html.filename", "child_test.html"
.setExportProperty "child1.jpg.filename", "child_test.jpg"
.setExportProperty "child1.excel.filename", "child_test.xls"
.setExportProperty "child1.word.filename", "child_test.doc"
.setExportProperty "child1.svg.filename", "child_test.svg"
.setExportProperty "child1.text.filename", "child_test.txt"
.setExportProperty "child1.ppt.filename", "child_test.ppt"
.setExportProperty "child1.tif.filename", "child_test.tif"
.setExportProperty "child1.csv.filename", "child_test.csv"

.Export()

Response.Write(IfSExportSucceeded())

.Clean()

End With

Set Com = Nothing

End If
%>

```