

.	API	3
	Class Cache	5
	Class ConnectionPool	10
	Class DataBind	15
	Class Log	19
	Class Module	23
	Class Monitor	38
	Class Repository	42
	Class Service	88
	Class Viewer	91
.	API	99
	Class Program	101
	Class Publisher	107
	Class Scheduler	111
.	API	147
	OZLauncherDll	148

. User Data Store	151
UDS	152
UDS	153
UDS	155
. User Security Logic	167
USL	168
USL	170
USL	173
.	193
.....	194
.....	194
.....	196
C	204
Appendix 1. SchedulerCom	223
Appendix 2. Servlet API	235

API

- Class Cache
- Class ConnectionPool
- Class DataBind
- Class Log
- Class Module
- Class Monitor
- Class Repository
- Class Service
- Class Viewer

API

API

Cache	
Connection Pool	JDBC/ODBC Pool
DataBind	
Log	
Module	
Monitor	
Repository	
Service	
Viewer	

API

가

ozsfw40.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

Prototype	//Daemon - TCP Server	
	<code>public Cache(String ip, int port, String id, String pw, boolean bAutoLogin, boolean useUSL)</code>	
Argument	//Servlet - HTTP Server	
	<code>public Cache(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

■	getCacheConfiguration
Prototype	public SortProperties getCacheConfiguration() throws OZCPEException
Definition	가 . 가 "SortProperties" key .
■	setCacheConfiguration
Prototype	public void setCacheConfiguration(SortProperties p) throws OZCPEException
Definition	. "SortProperties" key .
Argument	<i>p</i>

Class

■	OZCPEException(oz.framework.cp.OZCPEException)
API	Exception . API OZCPEException
.	.
-	-
■	getMessage
Prototype	public String getMessage()
Definition	가 .

- `getErrorCode`

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

Definition	가 .
-------------------	-----

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

`getCacheConfiguration(), setCacheConfiguration()`

- `getproperty`

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

Definition	key 가 .
-------------------	---------

- `setproperty`

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

Definition	key (value) .
-------------------	---------------

- Key

`getProperty()` `setProperty()` key .

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

memoryCacheValidTime		(:) ex) p.setProperty("datamodule.memoryCacheValidTime", "100");
diskCacheValidTime		(:) ex) p.setProperty("datamodule.diskCacheValidTime", "100");
FreeMemoryPercentage		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();

    /**
     *
     * (setCacheConfiguration)
     */
    p.setProperty("CACHE_FILE_PATH", "%OZ_HOME%/cache");
    /**
     *
     * (DM_CACHE_FILE_PATH, "%OZ_HOME%/cache_dm");
     * DataModule
     */
    p.setProperty("datamodule.active", "true");
    /**
     *
     * (datamodule.memoryCacheValidTime, "1000");
     *
     */
    p.setProperty("datamodule.diskCacheValidTime", "1000");
    /**
     *
     */
    p.setProperty("datamodule.freeMemoryPercentage", "21");
    /**

    cache.setCacheConfiguration(p);

    /**
     *
     * 가 (getCacheConfiguration)
     */
    p = cache.getCacheConfiguration();
    /**
     *
     */
    p.list(System.out);
}
catch(Exception 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

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

■ addPool

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

■ removePool

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

■ getPoolInfoList

Prototype	public ConnectionPoolInfo[] getPoolInfoList() throws OZCPEException
Definition	ConnectionPool ConnectionPoolInfo 가 .

■ getPoolStatusList

Prototype	public ConnectionPoolStatus[] getPoolStatusList() throws OZCPEException
------------------	---

Definition `ConnectionPool` 가 .

■ `getPoolInfo`

Prototype `public ConnectionPoolInfo getPoolInfo(String alias) throws OZCPEException`

Definition `ConnectionPool` `ConnectionPoolInfo` 가 .

Argument *alias* `ConnectionPool`

■ `save`

Prototype `public void save() throws OZCPEException`

Definition `ConnectionPool` .

Class

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

가 .

■ `ConnectionPoolStatus(oz.framework.db.ConnectionPoolStatus.class)`

`ConnectionPool` 가 .

-

- `public final static int OK = 1;`

`ConnectionPool` .

Status	
1	OK <code>ConnectionPool</code>
-1	DRIVER_ERROR <code>ConnectionPool</code> JDBC
-2	CONNECTION_ERROR <code>ConnectionPool</code> DBMS

- `public final static int DRIVER_ERROR = -1;`
- `public final static int CONNECTION_ERROR = -2;`

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) {
        // 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)
             * ConnectionPoolInfo poolInfo = new ConnectionPoolInfo();
             * poolInfo.setAlias("forcs"); //
             * poolInfo.setVendor("mssql"); // ( MSSQL)
             */
        }
    }
}
```

```

//
//      map      dbconfig.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");
poolInfo.setItems(infos);

poolInfo.setMaxConns(20); //
poolInfo.setInitConns(1); //
conPool.addPool(poolInfo);

//      (removePool)
String conPoolName = "forcs"; //
conPool.removePool(conPoolName);
// ConnectionPoolStatus      가      (getPoolStatusList)
ConnectionPoolStatus[]
poolStatusList=conPool.getPoolStatusList();
for (int i = 0; i < poolStatusList.length; i++) {
    ConnectionPoolStatus cps = poolStatusList[i];

    //
    System.out.println(i);
    System.out.println("StatusString=" + cps.getStatusString());

    //
    System.out.println("free=" + new
Integer(cps.getFreeConnectionCount()));

    //
    System.out.println("checkedout=" + new
Integer(cps.getCheckedOutConnectionCount()));
    System.out.println();
}
//      (save)
conPool.save();
}
catch(Exception e) {
    e.printStackTrace();
}
}
}

```

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

Prototype	//Daemon - TCP Server	
	<code>public DataBind(String ip, int port, String id, String pw, boolean bAutoLogin, boolean useUSL)</code>	
Argument	//Servlet - HTTP Server	
	<code>public DataBind(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

■ setDataBindConfiguration

Prototype	public void setDataBindConfiguration(SortProperties config) throws OZCPEException		
Definition	DataBind	, "databind.properties"	.
Argument	<i>config</i>	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 : , .

ConcurrentFirstRow		FetchType "Concurrent" , 0 : 0
---------------------------	--	--

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();

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

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

            //                     가     (getDataBindConfiguration)
            p = dataBind.getDataBindConfiguration();
            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

Prototype	//Daemon - TCP Server	
	<code>public Log(String ip, int port, String id, String pw, boolean bAutoLogin, boolean useUSL)</code>	
Argument	//Servlet - HTTP Server	
	<code>public Log(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;

■ **setPriority**

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

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);
            // 가 (getConfigure)
            String conf = null;
            conf = log.getConfigure();
            System.out.println(conf);
            // (setConfigure, setProirity)
            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 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

- `InputStream getOZD(String item, String category, String[] urls)`
- `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 getOZDWithSDM(String item, String category, Hashtable sdm, Hashtable formParam, boolean memoallowed, String password, String id, String pwd, HttpServletRequest request)`
- `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)`

Constructor Detail

Prototype	//Daemon - TCP Server	
	public Module(String ip, int port, String id, String pw, boolean bAutoLogin, boolean useUSL)	
	//Servlet - HTTP Server	
	public Module(String url, String id, String pw, boolean bAutoLogin, boolean useUSL)	
Argument	<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 OZCPEException
	public final InputStream getOZD(String item, String category, Hashtable formparam, Hashtable odiparam, 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 odiparam, Hashtable odipath, boolean memoallowed, String password, String id, String pwd, HttpServletRequest request)		
	가 OZD		
	가	, OZD	urls
	OZD		
Definition	OZD	: API serverDMType serverDMType DM_TYPE="Memory", FetchType="Batch"	
Argument	item	(	OZR)
	category		
	urls	OZD	URL
	serverDMType	Memory File (:Memory)	
	formparam	:	Hashtable (String), (String)
	odiparam	ODI : ODI Hashtable ODI (String), (Hashtable) , Hashtable (String), (String)	
	odipath	ODI	
	memoallowed	OZD	
	password	OZD	
	id		
	pwd		

	<i>request</i>	HttpServletRequest
--	----------------	--------------------

■ getOZDWithSDM

Prototype	<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>	
Definition	Servlet API SDM "SDM → gzip → Base64 Encode → Encode URI" SDM "Base64 Decode → ungzip → SDM getOZDWithSDM doGet doPost doPost getOZDWithSDM	
Argument	<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

■ getOZU

Prototype	<pre>public InputStream getOZU(String item, String category, String[] urls) throws OZCPEException</pre>
------------------	---

	SDM	가	OZU	가
Definition	: API DM_TYPE="Momory", FetchType="Batch"			
	: "FetchUnit" "DM_PER_DATAMODULE"			
Argument	<i>item</i>	(	OZA	)
	<i>category</i>			
	<i>urls</i>	OZU	URL	

■ addODIPParameter

Prototype	public void addODIPParameter(String odiName, String key, String value)			
Definition	SDM	ODI	ODI	ODI
	. ODI			
Argument	<i>odiName</i>	ODI		
	<i>key</i>	ODI		
	<i>value</i>	ODI		

■ addODIPParameter

Prototype	public void addODIPParameter(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>

■ addApplicationParameter

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

■ registODIPath

Prototype	public void registODIPath (String odiName, String path) throws IllegalArgumentException
Definition	OZD ODI
Argument	odiName ODI path ODI

■ saveOZD

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

Definition	OZD		
	: API DM_TYPE="Momory", FetchType="Batch"		
Argument	<i>fileName</i>	OZD 가	
	<i>item</i>	(.ozr)	
	<i>category</i>	(.ozr)	
	<i>urls</i>	OZD	URL

■ saveOZU

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

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
*/
// Servlet
String URL = "http://www.oz.com/oz/server";
/**/
// 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);
    */
    // Servlet
    module = new Module(URL, ID, PWD, false, false);
    /**/
    //module.addODIPParameter("          100  ", "rowcount", "40000");
    //replace ODI
    //module.registODIPath("          100  ", "/"          /          100
    // .odi");
    // module.addParameter("server", "127.0.0.1");
    // module.addParameter("port", "8003");
    // module.addParameter("reportname", "/"          /          100  .ozr");
    //          ozd
    //stream = module.getOZD("          100  .ozr", "/"          );
    //FileOutputStream out = new FileOutputStream("D:/TEST_OZD.ozd");
    // saveOZD(file name, item name, category name)
    //module.saveOZD("D:/TEST_OZD.ozd", "          100  .ozr", "/"          );
    //out.flush();
    //out.close();
    // -----
    // OZD
    // set form parameters
    module.addParameter("FORM_PARAM1", "COCO          1");
    module.addParameter("FORM_PARAM2", "COCO          2");
    // set odi parameters
    module.addODIPParameter("OZP_PARAM", "ODI_PARAM1", "          ODI
1");

```

```
module.addODIPParameter("OZP_PARAM", "ODI_PARAM2", "ODI
2");
module.addODIPParameter("OZP_DB", "condition", "KIA");
String[] urls = {
    "http://211.116.251.20/img/ban_center.gif",
    "http://211.116.251.20/img/img_interview.gif",
    "ozp://img/netiq.gif", // ozp:// image
    "ozp://img/credos.gif", // ozp:// image - table
    "ozp://img/enterprise.gif", // ozp:// image - table
    "ozp://img/sephia.gif" // ozp:// image - table
};
module.setPassword("1234");
module.setMemoAllowed(true);
module.saveOZD("D:/OZP_TEST01.ozd", "OZP_TEST.ozr", "/", urls);
}
catch(Exception e) {
    e.printStackTrace();
}
}
}
```

Sample : RequestOZDSample.java

```
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("odiparam1", "ODI 1");
        _odi_param.put("odiparam2", "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_PARAM = new Hashtable();
        //Hashtable _ODI_PATH = new Hashtable();
        //_ODI_PARAM.put("odiparam1", "ODI 1");
        //_ODI_PARAM.put("odiparam2", "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 : OZOZDMakerServlet.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 OZOZMakerServlet 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();
}
}
```

Sample : ModuleOZUSample.java

```
package sample;

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

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;
        InputStream stream = null;
    }
}
```

```
try{
    module = new Module(IP, PORT, ID, PWD, false, /*usl*/ false);
    //ozu
    module.addApplicationParameter("odi.fetchunit", "DM_PER_DATAMODULE");
    module.addApplicationParameter("odi.odinames", "sample");
    module.addApplicationParameter("odi.sample.pcount", "1");
    module.addApplicationParameter("odi.sample.args1", "deptid=501");
    Hashtable hash = new Hashtable();
    hash.put("deptid", "501");
    String[] urls = {
        "http://www.anykiki.com/custom/casting/ /Dana/ .gif");
    // paramHash  NULL   가   addApplication Parameter
    //
    module.addODIPParameter("sample", "sample.odi", "/sample", null);
    module.addODIPParameter("sample", "sample.odi", "/sample", hash);
    //ozu           , oza           ,           , URL
    module.saveOZU("D:/ozu.ozu", "sample.oza", "/sample", urls);
}
catch(Exception e){
    e.printStackTrace();
}
}
```

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

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

■ **getVersions**

Prototype public Versions getVersions() throws OZCPEException

Definition .

■ **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 .

■ 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) {
        // 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

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.getFreeMememoy();

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

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

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

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

Class Repository

Constructor Summary

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

Method Summary

// Configuration

- `public void setRepositoryConfig(SortProperties prop)`
- `public SortProperties getRepositoryConfig()`

// User

- `public int createUser(String userName, String pwd, int gid, String description)`
- `public void deleteUser(int uid)`
- `public void updateUserName(int uid, String userName)`
- `public String getUserNameByID(int uid)`

// UserLogin

- `public void disableUserLogin(String userName)`
- `public void updateLoginDefault(int loginDVal)`
- `public void enableUserLogin(String userName)`
- `public void userLogout(int uid)`
- `public boolean isUserLogin(int uid)`
- `public boolean loginToServer()`

// UserPwd

- `public boolean checkUserPwd(int uid, String pwd)`

```

■ public void updateUserPwd(int uid, String pwd)

// UserDesc
■ public void updateUserDescription(int uid, String description)
■ public String getUserDescription(int uid)

// UserID
■ public int getGroupIdOfUser(int uid)
■ public int getUserIdByName(String userName)
■ public void updateGroupIdOfUser(int gid, int uid)

// UserList
■ public OZRepositoryUser[] getUserList()
■ public OZRepositoryUser[] getUserListInGroup(int gid)
■ public OZRepositoryUser[] getUserListAuthToItem(int itemid, byte perm)
■ public OZRepositoryUser[] getUserListAuthToCategory(int categoryId, byte
    perm)

// Group
■ public int createGroup(String groupName, int upperGid)
■ public void deleteGroup(int gid)
■ public void updateUpperGroupId(int gid, int upperGid)
■ public void updateGroupName(int groupId, String groupName)

// GroupAdmin
■ public void updateUserGroupAdmin(int uid, int gid)
■ public boolean isUserGroupAdmin(int uid, int gid)

// GroupList
■ public OZRepositoryGroup[] getGroupListInGroup(int gid)
■ public OZRepositoryGroup getGroupInfo(int gid)
■ public OZRepositoryGroup[] getGroupListAuthToItem(int itemid, byte
    perm)
■ public OZRepositoryGroup[] getGroupListAuthToCategory(int categoryId,
    byte perm)

```

// Item

- **public int createItem(String itemName, int itemType, String itemDescription, int cid, InputStream itemIn)**
- **public int createItem(String itemName, int itemType, String itemDescription, String categoryName, InputStream itemIn)**
- **public int getItemId(String itemName, int itemType, int cid)**
- **public int getItemId(String itemName, int itemType, String cName)**
- **public void deleteItem(int itemid)**
- **public InputStream getItem(int itemid)**
- **public void updateItemName(int itemId, String itemName)**
- **public InputStream getDirectItem(String itemName, int itemType, String categoryName)**
- **public InputStream getDirectItem(String itemName, int itemType, String categoryName, boolean compressedItem)**
- **public void updateItem(int itemId, InputStream input)**
- **public void updateDirectItem(String itemName, int itemType, String categoryName, InputStream input)**
- **public boolean hasItemInRepository(String itemName, int itemType, String categoryName)**

// InfoByItem

- **public int getCategoryIdOfItem(int itemid)**
- **public void updateCategoryIdOfItem(int cid, int itemid)**

// ItemList

- **public OZRepositoryItem[] getItemList()**
- **public OZRepositoryItem getItemInfo(int itemid)**
- **public OZRepositoryItem[] getItemListInCategory(int cid)**
- **public OZRepositoryItem[] getItemListInCategory(String cName)**
- **public OZRepositoryItem[] getItemListInCategory(int cid, int uid, byte perm)**
- **public OZRepositoryItem[] getItemListInCategory(String categoryName, int uid, byte perm)**
- **public OZRepositoryItem[] getItemListInCategoryAuthGroup(int cid, int gid, byte perm)**

- `public OZRepositoryItem[] getItemListInCategoryAuthGroup(String categoryName, int gid, byte perm)`
- `public OZRepositoryItem[] getItemListAuthToUser(int uid, byte perm)`
- `public OZRepositoryItem[] getItemListAuthToGroup(int gid, byte perm)`

// Category

- `public int createCategory(String categoryName, int upperCid)`
- `public int createCategory(String categoryPath)`
- `public void deleteCategory(int cid)`
- `public int getCategoryID(String fullPath)`
- `public void updateUpperCategoryId(int cid, int upperCid)`
- `public void updateCategoryName(int cid, String categoryName)`
- `public int getItemCountInCategory(int cid)`
- `public OZRepositoryCategory[] getCategoryListInCategory(int cid)`
- `public OZRepositoryCategory[] getCategoryListInCategory(int cid, int uid, byte perm)`
- `public OZRepositoryCategory[] getCategoryListInCategoryAuthGroup(int cid, int gid, byte perm)`
- `public OZRepositoryCategory getCategoryInfo(int cid)`
- `public OZRepositoryCategory[] getCategoryListAuthToUser(int uid, int cid, byte perm)`
- `public OZRepositoryCategory[] getCategoryListAuthToGroup(int gid, int cid, byte perm)`

// CheckInOut

- `public void checkOutItem(int itemId, int uid, String checkoutFolder)`
- `public void undoCheckOutItem(int itemId, int uid)`
- `public void checkInItem(boolean keepChkOut, int itemId, int uid, InputStream itemIn)`
- `public boolean isCheckOutUser(int itemId, int uid)`

// History

- `public void rollBackItem(int itemId, int itemVersion)`
- `public InputStream getSpecifiedVersionItem(int itemId, int itemVersion)`
- `public OZRepositoryHistory[] getHistoryItemList(int itemId)`

- `public void clearHistoryItem(int itemId, int itemVersion)`

Constructor Detail

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

// Configuration

- `setRepositoryConfig`

Prototype	public void setRepositoryConfig(SortProperties prop) throws OZCPEException
-----------	--

Definition . "repository.properties"
Key

Argument *prop*

■ **getRepositoryConfig**

Prototype public SortProperties getRepositoryConfig() throws
OZCPEException

Definition 가 , 가 . "repository.properties"
Key

- Key

getRepositoryConfig() 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");

// User

- **createUser**

Prototype	public int createUser(String userName, String pwd, int gid, String description) throws OZCPEException
Definition	, ID .
	<i>userName</i>
	<i>pwd</i>
Argument	<i>gid</i> ID
	<i>description</i>

- **deleteUser**

Prototype	public void deleteUser(int uid) throws OZCPEException
Definition	ID .
Argument	<i>uid</i> ID

- `updateUserName`

Prototype	public void updateUserName(int uid, String userName) throws OZCPEException
Definition	ID .
Argument	<i>uid</i> ID <i>userName</i>

- `getUserNameByID`

Prototype	public String getUsernameByID(int uid) throws OZCPEException
Definition	ID가 .
Argument	<i>uid</i> ID

```
// UserLogin
```

- **disableUserLogin**

Prototype	public void disableUserLogin(String userName) throws OZCPEException
Definition	.
Argument	<i>userName</i>

■ **updateLoginDefault**

Prototype	public void updateLoginDefault(int loginDVal) throws OZCPEException
------------------	---

Definition	ID	.
-------------------	----	---

Argument	<i>loginDVal</i>	ID
-----------------	------------------	----

■ **enableUserLogin**

Prototype	public void enableUserLogin(String userName) throws OZCPEException
------------------	--

Definition	가	.
-------------------	---	---

Argument	<i>userName</i>	가
-----------------	-----------------	---

■ **userLogout**

Prototype	public void userLogout(int uid) throws OZCPEException
------------------	---

Definition	ID	,	.
-------------------	----	---	---

Argument	<i>uid</i>	ID
-----------------	------------	----

■ **isUserLogin**

Prototype	public boolean isUserLogin(int uid) throws OZCPEException
------------------	---

Definition	ID	가	,	.
-------------------	----	---	---	---

Argument	<i>uid</i>	ID
-----------------	------------	----

■ **loginToServer**

Prototype	public boolean loginToServer() throws OZCPEException
------------------	--

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

// **UserPwd**■ **checkUserPwd**

Prototype	public boolean checkUserPwd(int uid, String pwd) throws OZCPEException
------------------	--

Definition	가	.
-------------------	---	---

Argument	<i>uid</i>	ID
	<i>pwd</i>	

■ updateUserPwd

Prototype	public void updateUserPwd(int uid, String pwd) throws OZCPEException		
Definition	ID		.
Argument	<i>uid</i>	ID	
	<i>pwd</i>		

// UserDesc

■ updateUserDescription

Prototype	public void updateUserDescription(int uid, String description) throws OZCPEException		
Definition	ID		.
Argument	<i>uid</i>	ID	
	<i>description</i>		

■ getUserDescription

Prototype	public String getUserDescription(int uid) throws OZCPEException		
Definition	ID	가	.
Argument	<i>uid</i>	가	ID

// UserID

■ getGroupIdOfUser

Prototype	public int getGroupIdOfUser(int uid) throws OZCPEException		
Definition	ID	ID	가 .
Argument	<i>uid</i>	ID	가 ID

■ getUserIdByName

Prototype	public int getUserIdByName(String userName) throws OZCPEException		
Definition		ID	가 .
Argument	<i>userName</i>	ID	가

■ **updateGroupIdOfUser**

Prototype	public void updateGroupIdOfUser(int gid, int uid) throws OZCPEException		
Definition	ID	.	
Argument	<i>gid</i>	ID	
	<i>uid</i>	ID	ID

// **UserList**■ **getUserList**

Prototype	public OZRepositoryUser[] getUserList() throws OZCPEException		
Definition	가	가	.

■ **getUserListInGroup**

Prototype	public OZRepositoryUser[] getUserListInGroup(int gid) throws OZCPEException		
Definition	ID	가	.
Argument	<i>gid</i>	가	ID

■ **getUserListAuthToItem**

Prototype	public OZRepositoryUser[] getUserListAuthToItem(int itemid, byte perm) throws OZCPEException		
Definition	ID	perm	가
	가	.	
Argument	<i>itemId</i>	ID	
	<i>perm</i>		

■ **getUserListAuthToCategory**

Prototype	public OZRepositoryUser[] getUserListAuthToCategory(int categoryId, byte perm) throws OZCPEException		
Definition	ID	perm	가
	가	.	
Argument	<i>categoryId</i>	ID	
	<i>perm</i>		

// Group

■ createGroup

Prototype	public int createGroup(String groupName, int upperGid) throws OZCPEException		
Definition		ID	.
Argument	<i>groupName</i>		
	<i>upperGid</i>	ID	

■ deleteGroup

Prototype	public void deleteGroup(int gid) throws OZCPEException		
Definition		ID	.
Argument	<i>gid</i>	ID	

■ updateGroupName

Prototype	public void updateGroupName(int gid, String groupName) throws OZCPEException		
Definition		ID	.
Argument	<i>gid</i>	ID	
	<i>groupName</i>		

// GroupAdmin

■ updateUserGroupAdmin

Prototype	public void updateUserGroupAdmin(int uid, int gid) throws OZCPEException		
Definition		ID	.
Argument	<i>uid</i>	ID	
	<i>gid</i>	ID	

■ isUserGroupAdmin

Prototype	public boolean isUserGroupAdmin(int uid, int gid) throws OZCPEException		
Definition		ID	가
Argument	<i>uid</i>	ID	

	<i>gid</i>	ID
--	------------	----

// GroupList

■ **getGroupListInGroup**

Prototype	public OZRepositoryGroup[] getGroupListInGroup(int gid) throws OZCPEException	
Definition	ID	가 .
Argument	<i>gid</i>	가 ID

■ **getGroupInfo**

Prototype	public OZRepositoryGroup getGroupInfo(int gid) throws OZCPEException	
Definition	ID	가 .
Argument	<i>gid</i>	ID

■ **getGroupListAuthToItem**

Prototype	public OZRepositoryGroup[] getGroupListAuthToItem(int itemid, byte perm) throws OZCPEException	
Definition	ID perm	가 가 .
Argument	<i>itemId</i>	ID <i>perm</i>

■ **getGroupListAuthToCategory**

Prototype	public OZRepositoryGroup[] getGroupListAuthToCategory(int categoryId, byte perm) throws OZCPEException	
Definition	ID perm	가 가 .
Argument	<i>categoryId</i>	ID <i>perm</i>

■ **getSubGroupList**

Prototype	public OZRepositoryGroup[] getSubGroupList(int gid) throws OZCPEException	
-----------	--	--

Definition	getGroupListInGroup()	가 .	getSubGroupList()
	가 Recursive	가 . ,	...
Argument	<i>gid</i>	ID	
	<i>groupName</i>		

// Item

■ createItem

Prototype	public int createItem(String itemName, int itemType, String itemDescription, int cid, InputStream itemIn) throws OZCPEException		
	public int createItem(String itemName, int itemType, String itemDescription, String categoryName, InputStream itemIn) throws OZCPEException		
Definition		ID	.
	<i>itemName</i>		
Argument	<i>itemType</i>		
	<i>itemDescription</i>		
	<i>cid</i>	ID	
	<i>itemIn</i>		
	<i>categoryName</i>		

■ getItemId

Prototype	public int getItemId(String itemName, int itemType, int cid) throws OZCPEException		
	public int getItemId(String itemName, int itemType, String cName) throws OZCPEException		
Definition	ID	가 .	
	<i>itemName</i>	가	
Argument	<i>itemType</i>	가	
	<i>cid</i>	가	ID
	<i>cName</i>	가	

■ deleteItem

Prototype	public void deleteItem(int itemId) throws OZCPEException		
Definition	.		
Argument	<i>itemId</i>	ID	

■ getItem

Prototype	public InputStream getItem(int itemId) throws OZCPEException		
Definition	ID	가	.
Argument	<i>itemId</i>	가	ID

■ updateItemName

Prototype	public void updateItemName(int itemId, String itemName) throws OZCPEException		
Definition	ID		.
Argument	<i>itemId</i>	ID	
	<i>itemName</i>		

■ getDirectItem

	public InputStream getDirectItem(String itemName, int itemType, String categoryName) throws OZCPEException		
Prototype	public InputStream getDirectItem(String itemName, int itemType, String categoryName, boolean compressedItem) throws OZCPEException		
Definition	가		.
	<i>itemName</i>	가	
Argument	<i>itemType</i>	가	
	<i>categoryName</i>	가	
	<i>compressedItem</i>	가	

■ updateItem

Prototype	public void updateItem(int itemId, InputStream input) throws OZCPEException		
Definition	ID		.
Argument	<i>itemId</i>	ID	

	<i>input</i>
--	--------------

■ updateDirectItem

Prototype	public void updateDirectItem(String itemName, int itemType, String categoryName, InputStream input) throws OZCPEException
Definition	ID .
	<i>itemName</i>
	<i>itemType</i>
Argument	<i>categoryName</i>
	<i>input</i>

■ hasItemInRepository

Prototype	public boolean hasItemInRepository(String itemName, int itemType, String categoryName) throws OZCPEException
Definition	.
	<i>itemName</i>
Argument	<i>itemType</i>
	<i>categoryName</i>

// InfoByItem

■ getCategoryIdOfItem

Prototype	public int getCategoryIdOfItem(int itemId) throws OZCPEException
Definition	ID ID 가 .
Argument	<i>itemId</i> ID

■ updateCategoryIdOfItem

Prototype	public void updateCategoryIdOfItem(int cid, int itemId) throws OZCPEException
Definition	ID .
	<i>cid</i> ID
Argument	<i>itemId</i> ID

// ItemList

■ getItemList

Prototype	public OZRepositoryItem[] getItemList() throws OZCPEException
-----------	---

Definition	가 .
------------	-----

■ getItemInfo

Prototype	public OZRepositoryItem getItemInfo(int itemId) throws OZCPEException
-----------	---

Definition	ID 가 .
------------	--------

Argument	itemId 가 ID
----------	-------------

■ getItemListInCategory

Prototype	public OZRepositoryItem[] getItemListInCategory(int cid) throws OZCPEException
-----------	--

Prototype	public OZRepositoryItem[] getItemListInCategory(String categoryFullPath) throws OZCPEException
-----------	--

Prototype	public OZRepositoryItem[] getItemListInCategory(int cid, int uid, byte perm) throws OZCPEException
-----------	--

Prototype	public OZRepositoryItem[] getItemListInCategory(String categoryFullPath, int uid, byte perm) throws OZCPEException
-----------	--

Definition	가 .
------------	-----

Argument	cid 가 ID
----------	----------

Argument	categoryFullPath 가
----------	--------------------

Argument	uid 가 ID
----------	----------

Argument	perm
----------	------

■ getItemListInCategoryAuthGroup

Prototype	public OZRepositoryItem[] getItemListInCategoryAuthGroup(int cid, int gid, byte perm) throws OZCPEException
-----------	---

Prototype	public OZRepositoryItem[] getItemListInCategoryAuthGroup(String categoryFullPath, int gid, byte perm) throws OZCPEException
-----------	---

Definition	가 .
------------	-----

Argument	<i>cid</i>	가	ID
	<i>gid</i>	가	ID
	<i>perm</i>		
	<i>categoryFullPath</i>	가	

■ getItemListAuthToUser

Prototype	public OZRepositoryItem[] getItemListAuthToUser(int uid, byte perm) throws OZCPEException		
Definition	ID	perm	
	가	.	
Argument	<i>uid</i>	ID	
	<i>perm</i>		

■ getItemListAuthToGroup

Prototype	public OZRepositoryItem[] getItemListAuthToGroup(int gid, byte perm) throws OZCPEException		
Definition	ID	perm	
	가	.	
Argument	<i>gid</i>	ID	
	<i>perm</i>		

// Category

■ createCategory

Prototype	public int createCategory(String categoryName, int upperCid) throws OZCPEException		
Definition	,	ID	.
Argument	<i>categoryName</i>		
	<i>upperCid</i>	ID	

■ createCategory

Prototype	public int createCategory(String categoryPath) throws OZCPEException		
Definition	,	ID	.

Argument	<i>categoryPath</i>
----------	---------------------

■ deleteCategory

Prototype	public void deleteCategory(int cid) throws OZCPEException
-----------	---

Definition	ID	.
------------	----	---

Argument	<i>cid</i>	ID
----------	------------	----

■ getCategoryID

Prototype	public int getCategoryID(String fullPath) throws OZCPEException
-----------	---

Definition	ID	가	.
------------	----	---	---

Argument	<i>fullPath</i>	ID	가
----------	-----------------	----	---

■ updateUpperCategoryId

Prototype	public void updateUpperCategoryId(int cid, int upperCid) throws OZCPEException
-----------	--

Definition	ID	.
------------	----	---

Argument	<i>cid</i>	ID
----------	------------	----

	<i>upperCid</i>	ID
--	-----------------	----

■ updateCategoryName

Prototype	public void updateCategoryName(int cid, String categoryName) throws OZCPEException
-----------	--

Definition	ID	.
------------	----	---

Argument	<i>cid</i>	ID
----------	------------	----

	<i>categoryName</i>
--	---------------------

■ getItemCountInCategory

Prototype	public int getItemCountInCategory(int cid) throws OZCPEException
-----------	--

Definition	가	.
------------	---	---

Argument	<i>cid</i>	ID
----------	------------	----

■ getCategoryListInCategory

Prototype	public OZRepositoryCategory[] getCategoryListInCategory(int cid) throws OZCPEException	
	public OZRepositoryCategory[] getCategoryListInCategory(int cid, int uid, byte perm) throws OZCPEException	
Definition	가 .	
Argument	<i>cid</i>	ID
	<i>uid</i>	ID
	<i>perm</i>	

■ getCategoryInfo

Prototype	public OZRepositoryCategory getCategoryInfo(int cid) throws OZCPEException	
Definition	가 .	
Argument	<i>cid</i>	가 ID

■ getCategoryListAuthToUser

Prototype	public OZRepositoryCategory[] getCategoryListAuthToUser(int uid, int cid, byte perm) throws OZCPEException	
Definition	ID 가 .	perm
Argument	<i>uid</i>	ID
	<i>cid</i>	ID
	<i>perm</i>	

■ getCategoryListAuthToGroup

Prototype	public OZRepositoryCategory[] getCategoryListAuthToGroup(int gid, int cid, byte perm) throws OZCPEException	
Definition	ID 가 .	perm
Argument	<i>gid</i>	ID
	<i>cid</i>	ID
	<i>perm</i>	

// CheckInOut

■ checkOutItem

Prototype	public void checkOutItem(int itemid, int uid, String checkOutFolder) throws OZCPEException		
Definition	ID .		
	<i>itemid</i>	ID	
Argument	<i>uid</i>	ID	
	<i>checkOutFolder</i>		

■ undoCheckOutItem

Prototype	public void undoCheckOutItem(int itemId, int uid) throws OZCPEException		
Definition	ID .		
	<i>itemId</i>	ID	
Argument	<i>uid</i>	ID	

■ checkInItem

Prototype	public void checkInItem(boolean keepChkOut, int itemId, int uid, InputStream itemIn) throws OZCPEException		
Definition	ID .		
	<i>keepChkOut</i>		
	<i>itemId</i>	ID	
Argument	<i>uid</i>	ID	
	<i>itemIn</i>		

■ isCheckOutUser

Prototype	public boolean isCheckOutUser(int itemid, int uid) throws OZCPEException		
Definition	가 .		
	<i>itemId</i>	ID	
Argument	<i>uid</i>		ID

// History

■ getSpecifiedVersionItem

Prototype	public InputStream getSpecifiedVersionItem(int itemId, int itemVersion) throws OZCPEException		
Definition	ID	가	.
Argument	<i>itemId</i>	가	ID
	<i>itemVersion</i>	가	

■ **getHistoryItemList**

Prototype	public OZRepositoryHistory[] getHistoryItemList(int itemId) throws OZCPEException		
Definition		가	.
Argument	<i>itemId</i>	가	ID

■ **clearHistoryItem**

Prototype	public void clearHistoryItem(int itemId, int itemVersion) throws OZCPEException		
Definition			.
Argument	<i>itemId</i>	ID	
	<i>itemVersion</i>		

■ **rollBackItem**

Prototype	public void rollBackItem(int itemId, int itemVersion) throws OZCPEException		
Definition			.
Argument	<i>itemId</i>	ID	
	<i>itemVersion</i>		

Class

■ **OZRepositoryUser(oz.framework.repository.OZRepositoryUser)**

	,	,	,	,
가	.			
-				
▪	getUserName			

Prototype	public String getUsername()
Definition	가 .

▪ getUserID

Prototype	public int getUserID()
Definition	ID 가 .

▪ getGroupList

Prototype	public java.util.Vector getGroupList()
Definition	가 .

▪ getDescription

Prototype	public String getDescription()
Definition	가 .

▪ getPassWord

Prototype	public String getPassWord()
Definition	가 .

▪ getPermission

Prototype	public byte getPermission()
	가 .
Definition	• 0 : None() • 1 : View(가) • 3 : Read(가) • 7 : Write(가)

▪ getDirectPermission

Prototype	public byte getDirectPermission()
Definition	가 .

▪ getInDirectPermission

Prototype	public byte getInDirectPermission()
Definition	가 .

- `getIsLoggedIn`

Prototype public boolean getIsLoggedIn()

Definition 가

- getSessionID

```
Prototype    public int getSessionID()
```

Definition ID 가 .

- `getIsLoginEnabled`

```
Prototype    public boolean getIsLoginEnabled()
```

Definition	가	가	
-------------------	---	---	--

- `OZRepositoryUser[](oz.framework.repository.OZRepositoryUser[])`

가

- `OZRepositoryGroup(oz.framework.repository.OZRepositoryGroup)`

- `getGroupName`

```

Prototype    public String getGroupName()

```

Definition $\mathcal{G}$ 가 .

- `getGroupID`

```

Prototype    public int getGroupID()

```

Definition ID 가 .

- getParentGroupID

Prototype `public int getParentGroupID()`

Definition	ID	가	.
------------	----	---	---

- `getGroupAdminList`

```
Prototype    public java.util.Vector getGroupAdminList()
```

Definition 가 .

- `getDirectPermission`

```
Prototype public byte getDirectPermission()
```

Definition	가 .
------------	-----

- `getIndirectPermission`

```
Prototype public byte getIndirectPermission()
```

Definition	가 .
------------	-----

- `getPermission`

```
Prototype public byte getPermission()
```

Definition	가 .
------------	-----

- getDescription

```
Prototype public String getDescription()
```

Definition	가 .
------------	-----

- `getFullPath`

```
Prototype    public String getFullPath()
```

Definition	가 .
------------	-----

- ```
■ OZRepositoryGroup[] (oz.framework.repository.OZRepositoryGroup[])
```

- `OZRepositoryItem(oz.framework.repository.OZRepositoryItem)`

- getItemName

```
Prototype public String getItemName()
```

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

## ▪ getItemId

---

**Prototype**    `public int getItemId()`

---

**Definition**                    ID 가 .

---

## ▪ getItemType

---

**Prototype**    `public int getItemType()`

---

가 .

- 10000 : ODI\_FILE
  - 20001 : OZR\_FILE
  - 20002 : SDM\_FILE
  - 20003 : USDM\_FILE
  - 20004 : OZD\_FILE
  - 30001 : UIMG\_FILE
- 

**Definition**

## ▪ getDescription

---

**Prototype**    `public String getDescription()`

---

**Definition**                    가 .

---

## ▪ getCheckOutUserId

---

**Prototype**    `public int getCheckOutUserId()`

---

**Definition**                    ID 가 .

---

## ▪ getCheckOutUserName

---

**Prototype**    `public String getCheckOutUserName()`

---

**Definition**                    가 .

---

## ▪ getCheckOutFolder

---

**Prototype**    `public String getCheckOutFolder()`

---

**Definition**                    가 .

---

## ▪ getUpdateTime

---

**Prototype**    `public java.util.Date getUpdateTime()`

---

**Definition**                    가 .

---

▪ **getIsCheckedOut**

**Prototype** `public boolean getIsCheckedOut()`

**Definition** `getIsCheckedOut()` .

▪ **getDirectPermission**

**Prototype** `public byte getDirectPermission()`

**Definition** `getDirectPermission()` 가 .

▪ **getInDirectPermission**

**Prototype** `public byte getInDirectPermission()`

**Definition** `getInDirectPermission()` 가 .

▪ **getAdminList**

**Prototype** `public java.util.Vector getAdminList()`

**Definition** `getAdminList()` 가 .

▪ **getCategoryList**

**Prototype** `public java.util.Vector getCategoryList()`

**Definition** `getCategoryList()` 가 .

■ **OZRepositoryItem[](oz.framework.repository.OZRepositoryItem[])**

`OZRepositoryItem[]` 가 .

■ **OZRepositoryCategory(oz.framework.repository.OZRepositoryCategory)**

`OZRepositoryCategory` 가 .

▪ **getCategoryName**

**Prototype** `public String getCategoryName()`

**Definition** `getCategoryName()` 가 .

▪ **getCategoryID**

**Prototype** `public int getCategoryID()`

**Definition** `getCategoryID()` ID 가 .

- getCategoryID

|                  |                                         |
|------------------|-----------------------------------------|
| <b>Prototype</b> | <code>public int getCategoryID()</code> |
|------------------|-----------------------------------------|

|                   |        |
|-------------------|--------|
| <b>Definition</b> | ID 가 . |
|-------------------|--------|

- getCategoryAdminList

|                  |                                                             |
|------------------|-------------------------------------------------------------|
| <b>Prototype</b> | <code>public java.util.Vector getCategoryAdminList()</code> |
|------------------|-------------------------------------------------------------|

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

- getDirectPermission

|                  |                                                |
|------------------|------------------------------------------------|
| <b>Prototype</b> | <code>public byte getDirectPermission()</code> |
|------------------|------------------------------------------------|

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

- getInDirectPermission

|                  |                                                  |
|------------------|--------------------------------------------------|
| <b>Prototype</b> | <code>public byte getInDirectPermission()</code> |
|------------------|--------------------------------------------------|

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

- getPermission

|                  |                                          |
|------------------|------------------------------------------|
| <b>Prototype</b> | <code>public byte getPermission()</code> |
|------------------|------------------------------------------|

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

- getDescription

|                  |                                             |
|------------------|---------------------------------------------|
| <b>Prototype</b> | <code>public String getDescription()</code> |
|------------------|---------------------------------------------|

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

- getFullPath

|                  |                                          |
|------------------|------------------------------------------|
| <b>Prototype</b> | <code>public String getFullPath()</code> |
|------------------|------------------------------------------|

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

- `OZRepositoryCategory[](oz.framework.repository.OZRepositoryCategory[] )`

가 .

## ■ OZRepositoryHistory(oz.framework.repository.OZRepositoryHistory)

가 .

-

### ▪ getHistoryItemPath

|                  |                                    |
|------------------|------------------------------------|
| <b>Prototype</b> | public String getHistoryItemPath() |
|------------------|------------------------------------|

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

### ▪ getHistoryItemVersion

|                  |                                    |
|------------------|------------------------------------|
| <b>Prototype</b> | public int getHistoryItemVersion() |
|------------------|------------------------------------|

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

### ▪ getHistoryDate

|                  |                                |
|------------------|--------------------------------|
| <b>Prototype</b> | public String getHistoryDate() |
|------------------|--------------------------------|

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

### ▪ getHistoryCheckInUser

|                  |                                       |
|------------------|---------------------------------------|
| <b>Prototype</b> | public String getHistoryCheckInUser() |
|------------------|---------------------------------------|

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

## ■ OZRepositoryHistory[](oz.framework.repository.OZRepositoryHistory[])

가 .

## Sample : RepositorySample.java

```
package sample;

import oz.framework.api.Repository;
import oz.util.SortProperties;
import oz.framework.repository.OZRepositoryItem;
import oz.framework.repository.OZRepositoryUser;
import oz.framework.repository.OZRepositoryHistory;
import oz.framework.repository.OZRepositoryGroup;
import oz.framework.repository.OZRepositoryCategory;

import oz.framework.cp.OZCPEException;
import oz.dm.hc.HCDataModule;
```

```

import oz.dm.DMConst;
import java.io.*;
import java.net.*;
import org.apache.log4j.*;
import java.io.IOException;
import java.util.*;

public class RepositorySample {
 private static Repository repository = null;

 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

 try {
 // Daemon
 repository = new Repository(IP, PORT, ID, PWD, false, false);
 /**/
 // Servlet
 repository = new Repository(URL, ID, PWD, false, false);
 /**/

 repositoryConfiguration();
 userTest();
 groupTest();
 itemTest();
 }
 catch (Exception e)
 {
 e.printStackTrace();
 }
 }

 private static void repositoryConfiguration() throws Exception {
 //Repository configuration 가 .
 }
}

```

```
System.out.println("Repository.getRepositoryConfig()");
SortProperties props = repository.getRepositoryConfig();
props.list(System.out);

//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");
//repository.setRepositoryConfig(prop);
}

private static void historyTest(int itemid) throws Exception {
 final int itemVersion = 0;

 // itemVersion, itemid 가
 System.out.println("Repository.getSpecifiedVersionItem(
 int itemid, int itemVersion)");
 InputStream in = repository.getSpecifiedVersionItem(
 itemid, itemVersion);
 download(repository.getSpecifiedVersionItem(itemid, itemVersion),
 "getSpecifiedVersionItem" +
 repository.getItemInfo(itemid).getItemName());

 // itemid 가
 System.out.println("Repository.getHistoryItemList(int itemid)");
 OZRepositoryHistory[] historyInfoList =
 repository.getHistoryItemList(itemid);
 if (historyInfoList != null) {
 for (int i = 0; i < historyInfoList.length; i++) {
 OZRepositoryHistory historyInfo = historyInfoList[i];
 System.out.println("HistoryDate : " +
 historyInfo.getHistoryDate());
 System.out.println("HistoryItemPath : " +
 historyInfo.getHistoryItemPath());
 System.out.println("HistoryItemVersion : " +
 historyInfo.getHistoryItemVersion());
 }
 }

 // Item version history .
 // history 가
 // System.out.println("Repository.clearHistoryItem(
 // int itemid, int itemVersion)");
 // repository.clearHistoryItem(itemid, itemVersion);

 // Item itemVersion .
```

```

 System.out.println("Repository.rollbackItem(
 int itemid, int itemVersion)");
 repository.rollbackItem(itemid, itemVersion);
 }

 private static void checkInOutTest(int itemid) throws Exception{
 final int uid = repository.getUserIdByName("admin");
 String fileName = "check in out test.txt";
 createFile(fileName);

 // User(uid)가 Item(itemid) checkout
 System.out.println("Repository.checkOutItem(
 int itemid, int uid, String checkoutFolder)");
 String checkoutFolder = "..";
 repository.checkOutItem(itemid, uid, checkoutFolder);
 // uid 가 itemid checkout , checkout cancel
 // System.out.println("Repository.undoCheckOutItem(
 // int itemid, int uid)");
 // repository.undoCheckOutItem(itemid, uid);

 // item checkin . keepChkOut check-in check-out
 System.out.println(
 "Repository.checkInItem(boolean keepChkOut, int itemid, int uid,
 InputStream item_in)");
 boolean keepChkOut = false;
 FileInputStream in = new FileInputStream(fileName);
 repository.checkInItem(keepChkOut, itemid, uid, in);
 in.close();

 // uid user 가 itemid check out
 System.out.println("Repository.isCheckOutUser(int itemid, int uid)");
 System.out.println("Does UID[" + uid + "] check out the item[" +
 itemid + "] ? " + repository.isCheckOutUser(itemid, uid));
 removeFile(fileName);
 }

 private static void categoryTest() throws Exception {
 final int uid = repository.getUserIdByName("admin");
 int gid = repository.getGroupIdOfUser(uid);
 int cid, parent_cid;
 final byte authREAD = 0x02, authVIEW = 0x01;
 String categoryName = null;
 OZRepositoryCategory[] categoryInfoList = null;

 //
 System.out.println("Repository.createCategory(String categoryPath)");
 categoryName = "/Poultry";
 cid = repository.createCategory(categoryName);
 }

```

```
System.out.println(
 "categoryPath [categoryID] : " + categoryName + "[" + cid + "]"");

// upper_cid category_name .
System.out.println("Repository.createCategory(String category_name,
 int upper_cid)");
String childCategoryName = "Chickens";
parent_cid = cid;
cid = repository.createCategory(childCategoryName, parent_cid);
System.out.println("New new_category_name [new_categoryID] : " +
 childCategoryName + "[" + cid + "]"");

// cid .
System.out.println("Repository.deleteCategory(int cid)");
repository.deleteCategory(cid);
cid = parent_cid;

// Full Path 가 .
System.out.println("Repository.getCategoryID(String fullPath)");
System.out.println("the full Path [" + categoryName +
 "]"'s category ID : " + repository.getCategoryID(categoryName));

// .
System.out.println("Repository.updateCategoryName(int cid,
 String categoryName)");
categoryName = "Fishes";
System.out.println("Before Update CategoryName : " +
 (repository.getCategoryInfo(cid)).getCategoryName());
repository.updateCategoryName(cid, categoryName);
System.out.println("After Update CategoryName : " +
 (repository.getCategoryInfo(cid)).getCategoryName());
// .
System.out.println("Repository.updateUpperCategoryId(
 int cid, int uppercid)");
int new_cid = repository.createCategory("/Category Test");
System.out.println("Before Update UpperCategoryId : " +
 repository.getCategoryInfo(cid).getParentCategoryId());
repository.updateUpperCategoryId(cid, new_cid);
System.out.println("After Update UpperCategoryId : " +
 (repository.getCategoryInfo(cid)).getParentCategoryId());

// cid 가 .
System.out.println("Repository.getItemCountInCategory(String cid)");
System.out.println("Item count in the category id [" + cid + "] : " +
 repository.getItemCountInCategory(cid) + "");

// cid 가 .
System.out.println("Repository.getCategoryListInCategory(int cid) ");
```

```

categoryInfoList = repository.getCategoryListInCategory(0);
showCategoryInfoList(categoryInfoList);

// id 가 가
// 가 .
System.out.println("Repository.getCategoryListInCategory(
 int categoryId, int userId, byte perm)");
categoryInfoList = repository.getCategoryListInCategory(cid, uid,
 authVIEW);
showCategoryInfoList(categoryInfoList);

// gid 가
// 가 .
System.out.println(
 "Repository.getCategoryListInCategoryAuthGroup(int categoryId,
 int groupId, byte perm)");
categoryInfoList =
 repository.getCategoryListInCategoryAuthGroup(cid, gid, authVIEW);
showCategoryInfoList(categoryInfoList);

// id 가 .
System.out.println("Repository.getCategoryInfo(int cid)");
showCategory(repository.getCategoryInfo(cid));

//
// 가 .
System.out.println(
 "Repository.getCategoryListAuthToUser(int userId,int categoryId,
 byte perm) ");
categoryInfoList = repository.getCategoryListAuthToUser(uid, cid,
 authREAD);
showCategoryInfoList(categoryInfoList);

//
// 가
System.out.println("Repository.getCategoryListAuthToGroup(int groupId,
 int categoryId, byte perm) ");
categoryInfoList = repository.getCategoryListAuthToGroup(gid, cid,
 authREAD);
showCategoryInfoList(categoryInfoList);

repository.deleteCategory(cid);
repository.deleteCategory(new_cid);
}

private static void showCategoryInfoList(
 OZRepositoryCategory[] categoryList) {
 if(categoryList == null)

```

```
 return;
 for(int i=0; i < categoryList.length; i++) {
 showCategory(categoryList[i]);
 }
}

private static void showCategory(OZRepositoryCategory c) {
 System.out.println("CategoryId : " + c.getCategoryId());
 System.out.println("CategoryName : " + c.getCategoryName());
 System.out.println("CategoryUpperId : " + c.getParentCategoryId());
 System.out.println("Permission : " + c.getPermission());
}

private static void itemListTest(int itemid) throws Exception {
 FileOutputStream fos = null;
 InputStream in = null;
 OZRepositoryItem[] itemInfoList = null;
 OZRepositoryUser[] userInfoList = null;
 OZRepositoryCategory[] categoryInfoList = null;
 OZRepositoryGroup[] groupList = null;
 final String userName = "admin";
 String itemName = repository.getItemInfo(itemid).getItemName();
 int uid = repository.getUserIdByName(userName);
 int cid = repository.getCategoryIdOfItem(itemid);
 int gid = repository.getGroupIdOfItem(itemid);
 String categoryName = "/" +
repository.getCategoryInfo(cid).getCategoryName();
 final byte authRW = 0x07, authREAD = 0x02;
 byte[] buf = null;

 /*****

 .
 가
 .

 System.out.println("Repository.getItemList()");
 itemInfoList = repository.getItemList();
 showItemInfoList(itemInfoList);
 *****/

 //
 가
 .

 System.out.println("Repository.getItemInfo(int itemid)");
 OZRepositoryItem itemInfo = repository.getItemInfo(itemid);
 showItem(itemInfo);

 //
 id
 가
 .

 System.out.println("Repository.getItemListInCategory(int cid)");
 itemInfoList = repository.getItemListInCategory(cid);
 showItemInfoList(itemInfoList);
```

```

// 가 .
System.out.println(
 "Repository.getItemListInCategory(String categoryName)");
itemInfoList = repository.getItemListInCategory(categoryName);
showItemInfoList(itemInfoList);

// userId
// 가 .
System.out.println(
 "Repository.getItemListInCategory(String cName,int userid,
 byte perm)");
itemInfoList = repository.getItemListInCategory(categoryName, uid,
 authRW);
showItemInfoList(itemInfoList);

// categoryId userId
// 가 .
System.out.println("Repository.getItemListInCategory(int cid, int
 userid, byte perm)");
itemInfoList = repository.getItemListInCategory(cid, uid, authREAD);
showItemInfoList(itemInfoList);

// ID GroupID
// 가 .
System.out.println("Repository.getItemListInCategoryAuthGroup(int cid,
 int groupid, byte perm)");
itemInfoList = repository.getItemListInCategoryAuthGroup(cid, gid,
 authREAD);
showItemInfoList(itemInfoList);

// GroupID
// 가 .
System.out.println("Repository.getItemListInCategoryAuthGroup" +
 "(String categoryName, int groupid, byte perm)");
itemInfoList = repository.getItemListInCategoryAuthGroup(categoryName,
 gid, authREAD);
showItemInfoList(itemInfoList);

// uid 가 .
System.out.println("Repository.getItemListAuthToUser(int uid,
 byte perm)");
itemInfoList = repository.getItemListAuthToUser(uid, authREAD);
showItemInfoList(itemInfoList);

// gid 가 .
System.out.println("Repository.getItemListAuthToGroup(int gid,
 byte perm)");
itemInfoList = repository.getItemListAuthToGroup(gid, authREAD);

```

```
showItemInfoList(itemInfoList);

/*****

// 가 .
System.out.println(
 "Repository.findItemByItemIndex(String[] itemIndex, int[] oper)");
String[] itemIndex = {"index test 1", "index test 2", "index test 3"};
int[] oper = {2, 2}; //1 => AND, 2 => OR
itemInfoList = repository.findItemByItemIndex(itemIndex, oper);
showItemInfoList(itemInfoList);
*****/

}

private static void showItemInfoList(OZRepositoryItem[] itemInfoList) {
 if(itemInfoList == null)
 return;
 System.out.println(
 "[i]CategoryName:ItemName:ItemID-----");
 OZRepositoryItem item;
 for(int i=0; i < itemInfoList.length; i++) {
 item = itemInfoList[i];
 Vector v = item.getCategoryList();
 Iterator iter = v.iterator();
 while(iter.hasNext()) {
 System.out.println("[" + i + "]" + iter.next() + ":" +
 item.getItemName() + ":" + item.getItemId());
 }
 }
}

private static void showItem(OZRepositoryItem item) {
 Vector v = item.getCategoryList();
 Iterator iter = v.iterator();
 while(iter.hasNext()) {
 System.out.println(" " + iter.next() + ":" +
 item.getItemName() + ":" +
 item.getItemId());
 }
}

private static void infoByItemTest(int itemid) throws Exception {
 //
 System.out.println("Repository.getCategoryIdOfItem(int itemid)");
 System.out.println("the ItemID[" + itemid + "]'s category ID : " +
 repository.getCategoryIdOfItem(itemid));
 System.out.println(
 "Repository.updateCategoryIdOfItem(int cid, int itemid)");
}
```

```

 int cid = repository.getCategoryIdOfItem(itemid);
 System.out.println("Before Update CategoryID : " +
 repository.getCategoryIdOfItem(itemid));
 repository.updateCategoryIdOfItem(cid, itemid);
 System.out.println("After Update CategoryID : " +
 repository.getCategoryIdOfItem(itemid));
 }

 private static void itemTest() throws Exception {
 final String itemFileName = " 100 .odi";
 String itemName = itemFileName;
 String categoryName = "/Item Test";
 String itemDescription = "Test of item creation";
 final int itemType = 10000; // OZItemInfo.ODI_FILE

 int cid = repository.createCategory(categoryName);
 int itemid;

 // return

 System.out.println(
 "Repository.createItem(name, type, desc, cid, input_stream)");
 itemid = repository.createItem(itemFileName, itemType,
 itemDescription, cid,
 new FileInputStream(itemFileName));
 System.out.println("item_name [iid] : " + itemName + "[" +
 itemid + "]");

 // return
 // System.out.println(
 // "Repository.createItem(name, type, desc, category_name,
 // input_stream)");
 // itemid = repository.createItem(itemFileName, itemType,
 // itemDescription, categoryName,
 // new FileInputStream(itemFileName));
 // System.out.println("item_name [iid] : " + itemFileName + "[" +
 // itemid + "]");

 // , ,
 // ID 가
 System.out.println("Repository.getItemId(String item_name,
 int item_type, Sring categoryName)");
 itemid = repository.getItemId(itemName, itemType, categoryName);
 System.out.println("Item ID [" + itemid + "]");

 // , , id
 // ID 가
 System.out.println(
 "Repository.getItemId(String item_name, int item_type, int cid)");

```

```
itemid = repository.getItemId(itemName, itemType, cid);
System.out.println("Item ID [" + itemid + "]");

// 가 .
System.out.println("Repository.getItem(int itemid, int cid)");
System.out.println("Read the item from server and create a new file.");
download(repository.getItem(itemid, cid), "getItem" + itemName);

// .
System.out.println("Repository.updateItemName(int itemId,
 String itemName) ");
itemName = "Changed item name.odi";
System.out.println("Before Update ItemName : " +
 (repository.getItemInfo(itemid)).getItemName());
repository.updateItemName(itemid, itemName);
System.out.println("After Update ItemName : " +
 (repository.getItemInfo(itemid)).getItemName());

// 가 .
System.out.println("Repository.getDirectItem" +
 "(String itemName, int itemType, String categoryName,
 boolean compressedItem)");
System.out.println("Read the item from server and create a new file.");
boolean compress = true;
download(repository.getDirectItem(itemName, itemType, categoryName,
 compress), "getDirectItem_compressed" + itemName);

// 가 .
System.out.println("Repository.getDirectItem(String itemName,
 int itemType, String categoryName)");
System.out.println("Read the item from server and create a new file.");
download(repository.getDirectItem(itemName, itemType, categoryName),
 "getDirectItem_uncompressed" + itemName);
FileInputStream fis;

/*****

// .
System.out.println("Repository.updateItem(int itemId,
 InputStream input)");
fis = new FileInputStream(itemFileName);
repository.updateItem(itemid, fis);
fis.close();
// .
System.out.println(
 "Repository.updateDirectItem(String itemName, int itemType,
 String categoryName, InputStream input)");
fis = new FileInputStream(itemFileName);
```

```

 repository.updateDirectItem(itemName, itemType, categoryName, fis);
 fis.close();
 *****/

 //
 System.out.println("Repository.hasItemInRepository(
 String itemName, int itemType, String categoryName)");
 System.out.println("Does item[" + itemName + "] exist in category[" +
 categoryName + "] ? " +
 repository.hasItemInRepository(itemName, itemType,
 categoryName));

 infoByItemTest(itemid);
 itemListTest(itemid);
 categoryTest();
 checkInOutTest(itemid);
 historyTest(itemid);

 //
 System.out.println("Repository.deleteItem(int itemid)");
 repository.deleteItem(itemid);

 repository.deleteCategory(cid);
 }

 private static void download(InputStream in, String fileName)
 throws Exception {
 byte[] buf = new byte[in.available()];
 in.read(buf);
 in.close();
 FileOutputStream fos = new FileOutputStream(fileName);
 fos.write(buf);
 fos.flush();
 fos.close();
 }

 private static void groupListTest(int gid) throws Exception {
 // gid 가 .
 System.out.println("Repository.getGroupListInGroup(int gid)");
 OZRepositoryGroup[] groupInfoList = repository.getGroupListInGroup(0);
 showGroupList(groupInfoList);

 // gid 가 .
 System.out.println("Repository.getGroupInfo(int gid)");
 OZRepositoryGroup groupInfo = repository.getGroupInfo(gid);
 showGroup(groupInfo);

 // itemid perm 가 가 .

```

```
System.out.println("Repository.getGroupListAuthToItem(int itemid,
 byte perm)");
String sdmName = "SDM for test.sdm";
final int itemType = 20002; //OZFrameworkConst.SDM_FILE;
createFile(sdmName);
int cid = repository.createCategory("/User List Test");
int itemid = repository.createItem(sdmName, itemType, "", cid,
 new FileInputStream(sdmName));

final byte authREAD = 0x02;
groupInfoList = repository.getGroupListAuthToItem(itemid, authREAD);
showGroupList(groupInfoList);

// categoryId 가 가 .
System.out.println("Repository.getGroupListAuthToCategory(
 int categoryId, byte perm) ");
groupInfoList = repository.getGroupListAuthToCategory(cid, authREAD);
showGroupList(groupInfoList);

repository.deleteItem(itemid);
repository.deleteCategory(cid);
removeFile(sdmName);
}

private static void showGroupList(OZRepositoryGroup[] groupList) {
 if(groupList == null)
 return;
 for(int i=0; i < groupList.length; i++) {
 showGroup(groupList[i]);
 }
}

private static void showGroup(OZRepositoryGroup group) {
 System.out.println("GroupId : " + group.getGroupID());
 System.out.println("GroupName : " + group.getGroupName());
 System.out.println("GroupUpperId : " + group.getParentGroupID());
 Vector v = group.getGroupAdminList();
 showGroupAdminUserName(v);
 System.out.println("Permission : " + group.getPermission());
}

private static void showGroupAdminUserName(Vector v) {
 Iterator it = v.iterator();
 while (it.hasNext()) {
 System.out.println("GroupAdminUserName : " + it.next());
 }
}

}
```

```
private static void groupAdminTest(int gid) throws Exception{
 // Group Admin uid .
 System.out.println("Repository.updateUserGroupAdmin(int uid,
 int gid)");
 int uid = repository.createUser("test id", "1234567", gid, "");
 System.out.println("Before : ");

 showGroupAdminUserName(repository.getGroupInfo(gid).getGroupAdminList(
));

 repository.updateUserGroupAdmin(uid, gid);
 System.out.println("After : ");

 showGroupAdminUserName(repository.getGroupInfo(gid).getGroupAdminList(
));

 // uid 가 gid group admin .
 System.out.println("Repository.isUserGroupAdmin(int uid, int gid)");
 System.out.println("Is this user[" + uid +
 "] a group admin in Group[" + gid +
 "] ? " + repository.isUserGroupAdmin(uid, gid));
 repository.deleteUser(uid);
}

private static void groupTest() throws Exception {
 // upper_gid 가
 System.out.println("Repository.createGroup(String group_name,
 int upper_gid)");
 String groupName = "forcs";
 int root_gid = 0;
 int parent_gid = root_gid;
 int gid = repository.createGroup(groupName, parent_gid);
 System.out.println("group_name [upper_gid] : " + groupName +
 "[" + gid + "]");

 // 가 gid
 System.out.println("Repository.deleteGroup(int gid)");
 int temp_gid = repository.createGroup("Temporary group", root_gid);
 repository.deleteGroup(temp_gid);

 // .
 System.out.println("Repository.updateUpperGroupId(int gid,
 int uppergid) ");
 parent_gid = repository.createGroup("Temporary group", root_gid);
 System.out.println("Before Update UpperGroupID : " +
 (repository.getGroupInfo(gid)).getParentGroupID());
 repository.updateUpperGroupId(gid, parent_gid);
 System.out.println("After Update UpperGroupID : " +
```

```
 (repository.getGroupInfo(gid)).getParentGroupID());

//
System.out.println("Repository.updateGroupName(int groupId,
 String groupName)");
String newGroupName = "() ";
System.out.println("Before Update GroupIDofUser : " +
 (repository.getGroupInfo(gid)).getGroupName());
repository.updateGroupName(gid, newGroupName);
System.out.println("After Update GroupIDofUser : " +
 (repository.getGroupInfo(gid)).getGroupName());

groupAdminTest(gid);
groupListTest(gid);

repository.deleteGroup(parent_gid);
}

private static void createFile(String fileName) throws Exception {
 FileOutputStream fos = new FileOutputStream(new File(fileName));
 fos.write("dummy data".getBytes());
 fos.flush();
 fos.close();
}

private static void removeFile(String fileName) {
 try {
 File f = new File(fileName);
 f.delete();
 }
 catch(Exception e)
 {
 }
}

private static void userListTest(int uid) throws Exception {
 OZRepositoryUser userInfo = null;
 OZRepositoryUser[] userInfoList = null;
 int gid = repository.getGroupIdOfUser(uid);
 final String sdmName = "SDM for test.sdm";
 final int itemType = 20002; //OZRepositoryItem.SDM_FILE;
 createFile(sdmName);
 int cid = repository.createCategory("/User List Test");
 int itemid = repository.createItem(sdmName, itemType, "", cid,
 new FileInputStream(sdmName));

 final byte authREAD = 0x02;

//
 가
}
```

```

 System.out.println("Repository.getUserList()");
 userInfoList = repository.getUserList();
 showUserInfoList(userInfoList);

 // 가 .
 System.out.println("Repository.getUserListInGroup(int gid)");
 userInfoList = repository.getUserListInGroup(gid);
 showUserInfoList(userInfoList);

 // 가 가 .
 System.out.println(
 "Repository.getUserListAuthToItem(int itemid, byte perm)");
 userInfoList = repository.getUserListAuthToItem(itemid, authREAD);
 showUserInfoList(userInfoList);

 // 가 가 .
 System.out.println(
 "Repository.getUserListAuthToCategory(int categoryid, byte perm)");
 userInfoList = repository.getUserListAuthToCategory(cid, authREAD);
 showUserInfoList(userInfoList);

 repository.deleteItem(itemid);
 repository.deleteCategory(cid);
 removeFile(sdmName);
 }

 private static void showUserInfoList(OZRepositoryUser[] userInfoList) {
 if (userInfoList == null)
 return;
 OZRepositoryUser user;
 Iterator iter;
 System.out.println(
 "[i]GroupName:UserName:UserId-----");
 for (int i = 0; i < userInfoList.length; i++) {
 user = userInfoList[i];
 iter = user.getGroupList().iterator();
 while (iter.hasNext()) {
 System.out.println "[" + i + "]" + iter.next() + ":"
 + user.getUserName() + ":" +
 user.getUserID());
 }
 }
 }

 private static void userIDTest(int uid) throws Exception {
 // 가 Group ID .
 System.out.println(
 "Repository.updateGroupIdOfUser(int gid, int uid)");
 }

```

```
int old_gid = repository.getGroupIdOfUser(uid);
int new_gid = repository.createGroup("group_for_test", old_gid);
System.out.println("Before Update GroupIDofUser : " +
 repository.getGroupIdOfUser(uid));
repository.updateGroupIdOfUser(new_gid, uid);
System.out.println("After Update GroupIDofUser : " +
 repository.getGroupIdOfUser(uid));

//
System.out.println("Repository.getUserIdByName(String user_name)");
String userName = repository.getUserNameByID(uid);
uid = repository.getUserIdByName(userName);
System.out.println("uid[" + uid + "]" + userName);
repository.updateGroupIdOfUser(old_gid, uid);
repository.deleteGroup(new_gid);
}

private static void userDescTest(int uid) throws Exception {
 //
 System.out.println(
 "Repository.updateUserDescription(int uid, String description) ");
 String description = " ";
 System.out.println("Before Update Description : " +
 repository.getUserDescription(uid));
 repository.updateUserDescription(uid, description);
 System.out.println("After Update Description : " +
 repository.getUserDescription(uid));
}

private static void userPwdTest(int uid) throws Exception {
 // uid 가 pwd 가 check
 System.out.println("Repository.checkUserPwd(int uid, String pwd)");
 String pwd = "admin";

 // uid pwd
 System.out.println("Repository.updateUserPwd(int uid, String pwd)");
 System.out.println("Before : UserName : "
 + repository.checkUserPwd(uid, pwd));
 repository.updateUserPwd(uid, pwd);
 System.out.println("After : UserName : "
 + repository.checkUserPwd(uid, pwd));
}

private static void userLoginTest(int uid) throws Exception {
 // default login user loginVal
 System.out.println("Repository.updateLoginDefault(int loginDVal)");
 int loginDVal = uid; // uid user 가 default login setting
 repository.updateLoginDefault(loginDVal);
}
```

```
// userName login disable .
System.out.println("Repository.disableUserLogin(String userName)");
String userName = repository.getUserNameByID(uid);
repository.disableUserLogin("'" + userName + "'");

// userName login enable .
System.out.println("Repository.enableUserLogin(String userName)");
repository.enableUserLogin("'" + userName + "'");

//
System.out.println("Repository.loginToServer()");
System.out.println("loginToServer ? " + repository.loginToServer());

// uid logout .
System.out.println("Repository.userLogout(int uid)");
System.out.println("Before: Is this user[" + uid +
 "] logged on now ? " + repository.isUserLogin(uid));
//repository.userLogout(uid);
System.out.println("After: Is this user[" + uid +
 "] logged on now ? " + repository.isUserLogin(uid));
}

private static void userTest() throws Exception {

 //
 System.out.println(
 "Repository.createUser(String user_name, String pwd, int gid,
 String description)");
 String userName = "forcs";
 String pwd = "111111";
 int gid = 0; //
 String description = "() ";
 int uid = repository.createUser(userName, pwd, gid, description);
 System.out.println("user_name [uid] : " + userName + "[" + uid + "]");

 //
 System.out.println(
 "Repository.updateUserName(int userId, String userName)");
 String newUserName = " ";
 System.out.println("Before : UserName : "
 + repository.getUserNameByID(uid));
 repository.updateUserName(uid, newUserName);
 System.out.println("After : UserName : " +
 repository.getUserNameByID(uid));

 userLoginTest(uid);
}
```

```
 userDescTest(uid);
 userIDTest(uid);
 userListTest(uid);
 userPwdTest(uid);

 //
 System.out.println("Repository.deleteUser(int uid)");
 repository.deleteUser(uid);
 }
}
```

# 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()`

## Constructor Detail

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

## Method Detail

- **garbageCollect**

|            |                                                    |
|------------|----------------------------------------------------|
| Prototype  | public void garbageCollect() throws OZCPEException |
| Definition | JVM<br>가 garbage collection                        |

- stop

|                   |                                                                                         |
|-------------------|-----------------------------------------------------------------------------------------|
| <b>Prototype</b>  | public void stop(boolean check) throws OZCPEException                                   |
| <b>Definition</b> | .                                                                                       |
| <b>Argument</b>   | <i>check</i> <ul style="list-style-type: none"><li>• true :</li><li>• false :</li></ul> |

- **restart**

|                   |                                                          |
|-------------------|----------------------------------------------------------|
| <b>Prototype</b>  | <code>public void restart() throws OZCPEException</code> |
| <b>Definition</b> | .                                                        |

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

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

 // garbageCollection (garbageCollects)
 service.garbageCollect();
 // (restart)
 //service.restart();
 // (serverStop)
 boolean check = true;
 service.stop(check);

 }
 catch(Exception e)
 {
 }
 }
}

```

## 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)`
- `HCDDataModule getHCUSDM(String fileName, String categoryName)`
- `HCDDataModule getDataModule(InputStream sdmInput)`
- `HCDDataModule getDataModules(String odiname, String CategoryName, Parameter[] parameters, boolean doCompress, boolean forceRefresh)`
- `HCDDataModule getDataModule(String odiName, String categoryName, Parameter[] parameters, boolean doCompress, boolean forceRefresh, String[] invalidDataset, MaxRowsOfSet[] maxRows)`

### Constructor Detail

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

## Method Detail

### ■ getByteArrayForm

|            |                                                                                              |
|------------|----------------------------------------------------------------------------------------------|
| Prototype  | public byte[] getByteArrayForm(String reportName, String categoryName) throws OZCPEException |
| Definition | 가 .                                                                                          |
| Argument   | <i>reportName</i>                                                                            |
|            | <i>categoryName</i>                                                                          |

### ■ getForm

|           |                                                                                                              |
|-----------|--------------------------------------------------------------------------------------------------------------|
| Prototype | public InputStream getForm(String reportName, String categoryName, boolean isCompress) throws OZCPEException |
|-----------|--------------------------------------------------------------------------------------------------------------|

|            |                     |
|------------|---------------------|
| Definition | 가 .                 |
|            | <i>reportName</i>   |
| Argument   | <i>categoryName</i> |
|            | <i>isCompress</i>   |

#### ■ getUserParametersWithDefaultValue

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

#### ■ getHCUSDM

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

#### ■ getDataModules

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

#### ■ getDataModules

|            |                                                                                                                                                                                                                             |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            | <pre>public HCDDataModule getDataModule(String odiName, String categoryName, Parameter[] parameters, boolean doCompress,boolean forceRefresh) throws OZCPEException</pre>                                                   |
| Prototype  | <pre>public HCDDataModule getDataModule(String odiName, String categoryName, Parameter[] parameters, boolean doCompress, boolean forceRefresh, String[] invalidDataset, MaxRowsOfSet[] maxRows) throws OZCPEException</pre> |
| Definition | 가 .                                                                                                                                                                                                                         |
|            | <i>odiName</i>                                                                                                                                                                                                              |
|            | <i>categoryName</i>                                                                                                                                                                                                         |
|            | <i>parameters</i> 가                                                                                                                                                                                                         |
| Argument   | <i>doCompress</i>                                                                                                                                                                                                           |
|            | <i>forceRefresh</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 | . |
|------------|---|---|

---

- HCDataModule(oz.dm.hc.HCDataModule)

HCDataSet . HCDataSet BCDataSet 가 OZ  
DataModule Streaming .

- public final static String PARAMETERSET\_NAME = "OZParam"; ;

- getDataSetNames

---

|           |                                   |
|-----------|-----------------------------------|
| Prototype | public String[] getDataSetNames() |
|-----------|-----------------------------------|

---

|            |   |
|------------|---|
| Definition | . |
|------------|---|

---

- getDataSets

---

|           |                                             |
|-----------|---------------------------------------------|
| Prototype | public HCDataSet[] getDataSets(String name) |
|-----------|---------------------------------------------|

---

가 .

|            |   |   |   |
|------------|---|---|---|
| Definition | . | 가 | . |
|------------|---|---|---|

|             |   |
|-------------|---|
| (length)가 0 | . |
|-------------|---|

---

|          |      |
|----------|------|
| Argument | name |
|----------|------|

---

- PrintInfo

---

|           |                                        |
|-----------|----------------------------------------|
| Prototype | public void PrintInfo(PrintStream out) |
|-----------|----------------------------------------|

---

|            |             |   |
|------------|-------------|---|
| Definition | PrintStream | . |
|------------|-------------|---|

---

|          |     |             |
|----------|-----|-------------|
| Argument | out | PrintStream |
|----------|-----|-------------|

---

- `HCDataset(oz.dm.hc)`

```
HCDataset 가 OZ DataModule
Streaming .
```

가 .

—

- final public static int BYTEARRAY\_DATA\_SET = 0x1;
- final public static int UTF\_DATA\_SET = 0x2;
- final public static String BYTEARRAY\_PROTOCOL\_NAME = "ByteArraySet";
- final public static String UTF\_PROTOCOL\_NAME = "UTFSet";

—

- `getResultSet`

```
Prototype public ResultSet getResultSet()
```

**Definition** 가 .

## Sample : ViewerSample.java

```
package sample;

import oz.framework.api.Viewer;
import oz.dm.Parameter;
import oz.dm.hc.HCDataModule;
import oz.framework.cp.OZCPEException;
import oz.dm.MaxRowsOfSet;
import java.io.*;
import org.apache.log4j.*;

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

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 = "test.ozr";
 String odiName = "test.odi"; //ODI
 String sdmName = "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);

 //
 //(getUserParametersWithDefaultValue)
 param = viewer.getUserParametersWithDefaultValue(odiName,
 ODI_FILE_TYPE, categoryName);

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

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

```
 // (getDataModules)
 HCDataModule dModule2 = viewer.getDataModule(
 odiName, categoryName, param,compress, force, invalidSet, mx);

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

API

- Class Program
- Class Publisher
- Class Scheduler

API

API

| Program   |  |
|-----------|--|
| Publisher |  |
| Scheduler |  |

API가

| ozsfw40.jar | Scheduler server                          |
|-------------|-------------------------------------------|
| log4.jar    | Server (API Log classpath<br>"log4.jar" ) |

## Class Program

### Constructor Summary

- `Program(String ip, int port)`

### Method Summary

- `void createFolder(ServerInfo s, String folder)`
- `byte[] downloadFile(ServerInfo s, String file)`
- `FileInfo[] getExternalProgramList(ServerInfo s, String folder)`
- `void removeFiles(ServerInfo s, String folder, String[] files)`
- `void removeFolder(ServerInfo s, String folder, boolean isAll)`
- `void uploadFile(ServerInfo s, String file, byte[] b)`

### Constructor Detail

|                  |                                                                            |          |    |
|------------------|----------------------------------------------------------------------------|----------|----|
| <b>Prototype</b> | <code>public Program(String ip, int port) throws SchedulerException</code> |          |    |
|                  | <i>ip</i>                                                                  | 가        | IP |
| <b>Argument</b>  | ex) String ip = "127.0.0.1";                                               |          |    |
|                  | <i>port</i>                                                                | ( :9521) |    |
|                  | ex) int port = 9521;                                                       |          |    |

### Method Detail

- `createFolder`

|                  |                                                                                              |
|------------------|----------------------------------------------------------------------------------------------|
| <b>Prototype</b> | <code>public void createFolder(ServerInfo s, String folder) throws SchedulerException</code> |
|------------------|----------------------------------------------------------------------------------------------|

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

|          |               |
|----------|---------------|
| Argument | $s$           |
|          | <i>folder</i> |

- **downloadFile**

|            |                                                                                 |
|------------|---------------------------------------------------------------------------------|
| Prototype  | public byte[] downloadFile(ServerInfo s, String file) throws SchedulerException |
| Definition | .                                                                               |
| Argument   | <i>s</i><br><i>file</i>                                                         |

- `getExternalProgramList`

|            |                                                                                                              |
|------------|--------------------------------------------------------------------------------------------------------------|
| Prototype  | <code>public FileInfo[] getExternalProgramList(ServerInfo s, String folder) throws SchedulerException</code> |
| Definition | <code>가</code> .                                                                                             |
| Argument   | <code>s</code><br><code>folder</code>                                                                        |

- **removeFiles**

|            |                                                                                                             |
|------------|-------------------------------------------------------------------------------------------------------------|
| Prototype  | <code>public void removeFiles(ServerInfo s, String folder, String[] files) throws SchedulerException</code> |
| Definition | <code>.</code>                                                                                              |
|            | <code>s</code>                                                                                              |
| Argument   | <code>folder</code>                                                                                         |
|            | <code>files</code>                                                                                          |

- **removeFolder**

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

- **uploadFile**

|            |                                                                                          |
|------------|------------------------------------------------------------------------------------------|
| Prototype  | public void uploadFile(ServerInfo s, String file, byte[] b)<br>throws SchedulerException |
| Definition | ( ) .                                                                                    |
|            | <i>s</i>                                                                                 |
| Argument   | <i>file</i>                                                                              |
|            | <i>b</i>                                                                                 |

## Class

- `SchedulerException(oz.scheduler.SchedulerException)`

Exception .

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

가 .

- `setIsDaemon`

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

- setIP

|            |                                    |
|------------|------------------------------------|
| Prototype  | public final void setIP(String ip) |
| Definition | IP . Server가 Daemon<br>.           |
| Argument   | <i>ip</i> IP                       |

- setPortNo

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

|          |               |      |
|----------|---------------|------|
| Argument | <i>portNo</i> | Port |
|----------|---------------|------|

#### ▪ setURL

|           |                                                                         |
|-----------|-------------------------------------------------------------------------|
| Prototype | public final void setURL(String url) throws<br>IllegalArgumentException |
|-----------|-------------------------------------------------------------------------|

|            |                       |
|------------|-----------------------|
| Definition | URL . Server가 Servlet |
|------------|-----------------------|

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

#### ▪ setID

|           |                                    |
|-----------|------------------------------------|
| Prototype | public final void setID(String id) |
|-----------|------------------------------------|

|            |      |
|------------|------|
| Definition | ID . |
|------------|------|

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

#### ▪ setPWD

|           |                                      |
|-----------|--------------------------------------|
| Prototype | public final void setPWD(String pwd) |
|-----------|--------------------------------------|

|            |   |
|------------|---|
| Definition | . |
|------------|---|

|          |            |
|----------|------------|
| Argument | <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 sample;

import oz.scheduler.ServerInfo;
import oz.scheduler.FileInfo;
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(Exception e)
{
 e.printStackTrace();
}
}
```

## Class Publisher

### Constructor Summary

- Publisher(String ip, int port)

### Method Summary

- void createFolder(ServerInfo s, String folder)
- byte[] downloadFile(ServerInfo s, String file)
- FileInfo[] getPublishedFiles(ServerInfo s, String folder)
- void removeFiles(ServerInfo s, String folder, String[] files)
- void removeFolder(ServerInfo s, String folder, boolean isAll)

### Constructor Detail

|                  |                                                                 |                                   |    |
|------------------|-----------------------------------------------------------------|-----------------------------------|----|
| <b>Prototype</b> | public Publisher(String ip, int port) throws SchedulerException |                                   |    |
| <b>Argument</b>  | <i>ip</i>                                                       | 가<br>ex) String ip = "127.0.0.1"; | IP |
|                  | <i>port</i>                                                     | ( :9521)<br>ex) int port = 9521;  |    |

### Method Detail

- createFolder

|                   |                                                                                 |
|-------------------|---------------------------------------------------------------------------------|
| <b>Prototype</b>  | public void createFolder(ServerInfo s, String folder) throws SchedulerException |
| <b>Definition</b> | .                                                                               |

|          |               |
|----------|---------------|
| Argument | <i>s</i>      |
|          | <i>folder</i> |

---

#### ■ downloadFile

|            |                                                                                 |
|------------|---------------------------------------------------------------------------------|
| Prototype  | public byte[] downloadFile(ServerInfo s, String file) throws SchedulerException |
| Definition | .                                                                               |
| Argument   | <i>s</i>                                                                        |
|            | <i>file</i>                                                                     |

---

#### ■ getPublishedFiles

|            |                                                                                            |
|------------|--------------------------------------------------------------------------------------------|
| Prototype  | public FileInfo[] getPublishedFiles(ServerInfo s, String folder) throws SchedulerException |
| Definition | 가 .                                                                                        |
| Argument   | <i>s</i>                                                                                   |
|            | <i>folder</i>                                                                              |

---

#### ■ removeFiles

|            |                                                                                                |
|------------|------------------------------------------------------------------------------------------------|
| Prototype  | public void removeFiles(ServerInfo s, String folder, String[] files) throws SchedulerException |
| Definition | .                                                                                              |
| Argument   | <i>s</i>                                                                                       |
|            | <i>folder</i>                                                                                  |
|            | <i>fiels</i>                                                                                   |

---

#### ■ removeFolder

|            |                                                                                                |
|------------|------------------------------------------------------------------------------------------------|
| Prototype  | public void removeFolder(ServerInfo s, String folder, boolean isAll) throws SchedulerException |
| Definition | .                                                                                              |
| Argument   | <i>s</i>                                                                                       |
|            | <i>folder</i>                                                                                  |
|            | <i>isAll</i> 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 .*/
 } catch (Exception e) {
 e.printStackTrace();
 }
 }
}
```

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

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

// (getPublishedFiles)
FileInfo fileInfoList[] =
 publisher.getPublishedFiles(serverInfo, folderName);
for(int i=0; i<fileInfoList.length; i++) {
 FileInfo 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

- `void createTask(ServerInfo s, SortProperties configMap, SortProperties exportMap)`
- `Vector getTask(ServerInfo s)`
- `public TaskResult[] getTaskResult(ServerInfo s, String from, String to, String taskId) throws SchedulerException`
- `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) throws SchedulerException`
- `boolean makePDF(ServerInfo s, SortProperties configMap, SortProperties exportMap)`
- `public boolean print(ServerInfo s, SortProperties configMap, SortProperties printMap) throws SchedulerException`
- `SortProperties getConfiguration(ServerInfo s)`
- `void modifyConfiguration(ServerInfo s, SortProperties configMap, SortProperties exportMap)`

## Constructor Detail

|                  |                                        |                                   |    |
|------------------|----------------------------------------|-----------------------------------|----|
| <b>Prototype</b> | ppublic scheduler(String ip, int port) |                                   |    |
|                  | <i>ip</i>                              | 가<br>ex) String ip = "127.0.0.1"; | IP |
| <b>Argument</b>  | <i>port</i>                            | ( :9521)<br>ex) int port = 9521;  |    |

## Method Detail

### ■ createTask

|                   |                                                                                                                    |     |          |
|-------------------|--------------------------------------------------------------------------------------------------------------------|-----|----------|
| <b>Prototype</b>  | public void createTask(ServerInfo s, SortProperties configMap, SortProperties exportMap) throws SchedulerException |     |          |
| <b>Definition</b> | : CreateTask<br>Thread param                                                                                       |     |          |
|                   | <i>s</i>                                                                                                           |     |          |
| <b>Argument</b>   | <i>configMap</i>                                                                                                   | key | "Option" |
|                   | <i>exportMap</i>                                                                                                   | key | "Option" |

### ■ getTask

|                   |                                                               |  |  |
|-------------------|---------------------------------------------------------------|--|--|
| <b>Prototype</b>  | public Vector getTask(ServerInfo s) throws SchedulerException |  |  |
| <b>Definition</b> | 가                                                             |  |  |
| <b>Argument</b>   | <i>s</i>                                                      |  |  |

### ■ getTaskResult

|            |                                                                                                                  |   |  |
|------------|------------------------------------------------------------------------------------------------------------------|---|--|
| Prototype  | public TaskResult[] getTaskResult(ServerInfo s, String from, String to, String taskId) throws SchedulerException |   |  |
| Definition | 가 .                                                                                                              |   |  |
| Argument   | s                                                                                                                |   |  |
|            | from                                                                                                             | 가 |  |
|            | to                                                                                                               | 가 |  |
|            | taskId                                                                                                           | 가 |  |

#### ■ getTaskProperties

|            |                                                                                                  |             |   |
|------------|--------------------------------------------------------------------------------------------------|-------------|---|
| Prototype  | public SortProperties[] getTaskProperties(ServerInfo s, String taskId) throws SchedulerException |             |   |
| Definition | (configMap)                                                                                      | (exportMap) | 가 |
| Argument   | s                                                                                                |             |   |
|            | taskId                                                                                           | 가           |   |

#### ■ removeTask

|            |                                                                               |  |  |
|------------|-------------------------------------------------------------------------------|--|--|
| Prototype  | public void removeTask(ServerInfo s, String taskId) throws SchedulerException |  |  |
| Definition | .                                                                             |  |  |
| Argument   | s                                                                             |  |  |
|            | taskId                                                                        |  |  |

#### ■ modifyTask

|            |                                                                                                                                     |     |          |
|------------|-------------------------------------------------------------------------------------------------------------------------------------|-----|----------|
| Prototype  | public String modifyTask(ServerInfo s, String taskId, SortProperties configMap, SortProperties exportMap) throws SchedulerException |     |          |
| Definition | .                                                                                                                                   |     |          |
| Argument   | s                                                                                                                                   |     |          |
|            | taskId                                                                                                                              |     |          |
|            | configMap                                                                                                                           | key | "Option" |

|  |                  |     |          |
|--|------------------|-----|----------|
|  | <i>exportMap</i> | key | "Option" |
|  |                  | .   |          |

■ taskPause

|            |                                                                                 |
|------------|---------------------------------------------------------------------------------|
| Prototype  | public boolean taskPause(ServerInfo s, String taskId) throws SchedulerException |
| Definition |                                                                                 |
| Argument   | <i>s</i><br><i>taskId</i>                                                       |

■ taskResume

|            |                                                                                  |
|------------|----------------------------------------------------------------------------------|
| Prototype  | public boolean taskResume(ServerInfo s, String taskId) throws SchedulerException |
| Definition |                                                                                  |
| Argument   | <i>s</i><br><i>taskId</i>                                                        |

■ stop

|            |                                                                              |
|------------|------------------------------------------------------------------------------|
| Prototype  | public void stop(ServerInfo s, boolean waitTaskId) throws SchedulerException |
| Definition |                                                                              |
| Argument   | <i>s</i><br><i>waitTaskId</i>                                                |

■ export

|            |                                                                                                                                       |
|------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Prototype  | public boolean export(ServerInfo s, SortProperties configMap, SortProperties exportMap) throws SchedulerException                     |
| Definition | <pre>        : "ViewType=None"         : *.pdf, *.ozd, *.html, *.jpg,         *.xls, *.doc, *.svg, *.txt, *.ppt, *.tif, *csv 가 </pre> |

|          |                  |     |          |
|----------|------------------|-----|----------|
| Argument | <i>s</i>         |     |          |
|          | <i>configMap</i> | key | "Option" |
|          | <i>exportMap</i> | key | "Option" |

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

#### ■ makePDF

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

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

|          |                  |                      |                 |
|----------|------------------|----------------------|-----------------|
| Argument | <i>s</i>         |                      |                 |
|          | <i>configMap</i> | key<br>. creatTask() | "Option"<br>PDF |
|          | <i>exportMap</i> | key<br>. creatTask() | "Option"<br>PDF |

: makePDF API 가 COM ASP  
"Appendix 1. SchedulerCOM "

#### ■ print

**Prototype** public boolean print(ServerInfo s, SortProperties configMap, SortProperties printMap) throws SchedulerException

|            |                         |     |          |
|------------|-------------------------|-----|----------|
| Definition | 가                       |     |          |
|            | : "task_type=viewerTag" |     |          |
| Argument   | "ViewType=None"         |     |          |
|            | s                       |     |          |
| Argument   | configMap               | key | "Option" |
|            | exportMap               | key | "Option" |
| Argument   | , print.mode = silent   |     |          |
|            | : print API             |     |          |
| Argument   | , print API             |     |          |
|            |                         |     |          |

#### ■ getConfiguration

|            |                                                                                |  |  |
|------------|--------------------------------------------------------------------------------|--|--|
| Prototype  | public SortProperties getConfiguration(ServerInfo s) throws SchedulerException |  |  |
| Definition | 가                                                                              |  |  |
| Argument   | s                                                                              |  |  |

#### ■ modifyConfiguration

|            |                                                                                                   |     |          |
|------------|---------------------------------------------------------------------------------------------------|-----|----------|
| Prototype  | public void modifyConfiguration(ServerInfo s, SortProperties configMap) throws SchedulerException |     |          |
| Definition |                                                                                                   |     |          |
| Argument   | s                                                                                                 |     |          |
| Argument   | configMap                                                                                         | key | "Option" |
|            |                                                                                                   |     |          |

## Class

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

Program class " class"

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

Cache class " class"

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

- 
- public String taskID: ID
- public String completedTime :
- public int isSuccessfulCode:
- public String isSuccessful:
- public String formFileName:
- public String Parameter:
- public String schedulingType:
- public String exportFileList:
- public String errorMsg:

## Option

- "ViewType" "None"  
API , OZD
- 가 ,

| applet.mode           | export |
|-----------------------|--------|
| viewer.mode           | export |
| applet.useprogressbar | false  |

|                                |        |
|--------------------------------|--------|
| <b>viewer.useprogressbar</b>   | false  |
| <b>applet.allowmultiframe</b>  | true   |
| <b>viewer.allowmultiframe</b>  | true   |
| <b>export.mode</b>             | silent |
| <b>export.confirmsave</b>      | false  |
| <b>information.debug</b>       | debug  |
| <b>applet.showerrormessage</b> | false  |
| <b>viewer.showerrormessage</b> | false  |

■

-

| Key                  | Value |                                            |
|----------------------|-------|--------------------------------------------|
| <b>report_name</b>   |       | ex) setProperty("report_name", "crosstab") |
| <b>category_name</b> |       | ex) setProperty("category_name", "temp")   |

-

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

-

| Key | Value |  |
|-----|-------|--|
|-----|-------|--|

|                  |             |                                                                                           |
|------------------|-------------|-------------------------------------------------------------------------------------------|
| <b>task_type</b> | "viewerTag" | <br><br>"SchedulerViewerTagSample.java"<br><br>ex) setProperty("task_type ", "viewerTag") |
|------------------|-------------|-------------------------------------------------------------------------------------------|

| Key                             | Value             |                                                                                             |
|---------------------------------|-------------------|---------------------------------------------------------------------------------------------|
| <b>external_program_check</b>   | "check"<br>"null" | ex) setProperty("external_program_check", "check")                                          |
| <b>external_program_command</b> |                   | ( "SCH_HOME/External"<br>)<br>ex) SetProperty("external_program_command",<br>"notepad.bat") |

| Key                            | Value             |                                                               |
|--------------------------------|-------------------|---------------------------------------------------------------|
| <b>mail_check</b>              | "check"<br>"null" | ex) setProperty("mail_check", "check")                        |
| <b>mail_notify_error_check</b> | "check"<br>"null" | ex) setProperty("mail_notify_error_check", "null")            |
| <b>mail_recipient_to</b>       |                   | ex) setProperty("mail_recipient_to",<br>"gil_dong@forcs.com") |
| <b>mail_recipient_cc</b>       |                   | ex) setProperty("mail_recipient_cc",<br>"aaa@forcs.com")      |
| <b>mail_recipient_bcc</b>      |                   | ex) setProperty("mail_recipient_bcc",<br>"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"<br>"null" | HTML<br>ex) setProperty("mail_html_comment", "check")                  |
| <b>html_mail_content</b> | "check"<br>"null" | HTML<br>ex) setProperty("html_mail_content", "check")                  |
| <b>mail_attach_list</b>  |                   | (<br>'/'<br>)<br>ex) setProperty("mail_attach_list", "excel/pdf/word") |

■

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

■

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

■ : ODI

,

.

```

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

```

■ ODI

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

```

 : ODI
 ODI
 ODI

```

■

| Key         | Value                                     |                                                                                     |
|-------------|-------------------------------------------|-------------------------------------------------------------------------------------|
| launch_type | "once"<br>"immediately"<br>"periodically" | once :<br>immediately :<br>periodically :<br>ex) setProperty("launch_type", "once") |

■ launch\_type = once

| Key | Value |  |
|-----|-------|--|
|-----|-------|--|

|                        |  |                                                |
|------------------------|--|------------------------------------------------|
| <b>execution_year</b>  |  | —<br>ex) setProperty("execution_year", "2005") |
| <b>execution_month</b> |  | —<br>ex) setProperty("execution_month", "12")  |
| <b>execution_day</b>   |  | —<br>ex) setProperty("execution_day", "30")    |
| <b>execution_hour</b>  |  | —<br>ex) setProperty("execution_hour", "10")   |
| <b>execution_min</b>   |  | —<br>ex) setProperty("execution_minute", "30") |

- launch\_type = periodically

| Key                                    | Value                            |                                                             |
|----------------------------------------|----------------------------------|-------------------------------------------------------------|
| <b>start_year</b>                      |                                  | —<br>ex) setProperty("start_year", "2002")                  |
| <b>start_month</b>                     |                                  | —<br>ex) setProperty("start_month", "11")                   |
| <b>start_day</b>                       |                                  | —<br>ex) setProperty("start_day", "16")                     |
| <b>periodically_execution_day_type</b> | "daily"<br>"weekly"<br>"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> |         | ( )<br>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"<br>"null" | ex) setProperty("weekly_monday_check", "check")    |
| <b>weekly_tuesday_check</b>   | "check"<br>"null" | ex) setProperty("weekly_tuesday_check", "check")   |
| <b>weekly_wednesday_check</b> | "check"<br>"null" | ex) setProperty("weekly_wednesday_check", "check") |
| <b>weekly_thursday_check</b>  | "check"<br>"null" | ex) setProperty("weekly_thursday_check", "check")  |
| <b>weekly_friday_check</b>    | "check"<br>"null" | ex) setProperty("weekly_friday_check", "check")    |
| <b>weekly_saturday_check</b>  | "check"<br>"null" | ex) setProperty("weekly_saturday_check", "check")  |
| <b>weekly_sunday_check</b>    | "check"<br>"null" | ex) setProperty("weekly_sunday_check", "check")    |

- periodically\_execution\_day\_type = monthly

| Key                         | Value                                                |                                                                                     |
|-----------------------------|------------------------------------------------------|-------------------------------------------------------------------------------------|
| <b>monthly_every_months</b> |                                                      | ( )<br>ex) setProperty("monthly_every_months", "2")                                 |
| <b>monthly_type</b>         | "specific_day" ,<br>"day_of_week",<br>"user_defined" | ex) setProperty("monthly_type", "specific_day")                                     |
| <b>monthly_days</b>         |                                                      | ex) setProperty("monthly_days", "2")                                                |
| <b>monthly_which_week</b>   | "T1"<br>"T2"<br>"T3"<br>"T4"<br>"T5"                 | T1 :<br>T2 :<br>T3 :<br>T4 :<br>T5 :<br>ex) setProperty("monthly_which_week", "T2") |

|                                  |                                                                                          |                                                                       |
|----------------------------------|------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| <b>monthly_which_week_day</b>    | "sunday"<br>"monday"<br>"tuesday"<br>"wednesday"<br>"thursday"<br>"friday"<br>"saturday" | ex) :<br>setProperty("monthly_which_week_day", "monday")              |
| <b>monthly_user_defined_days</b> |                                                                                          | (.)<br>ex) 1,15 :<br>setProperty("monthly_user_defined_days", "1,15") |

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

- periodically\_execution\_time\_type = once

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

- periodically\_execution\_time\_type = repeat

| Key                         | Value |                                                    |
|-----------------------------|-------|----------------------------------------------------|
| <b>repeat_every_hours</b>   |       | —<br>ex) setProperty("repeat_every_hours", "01")   |
| <b>repeat_every_minutes</b> |       | —<br>ex) setProperty("repeat_every_minutes", "04") |
| <b>repeat_start_hour</b>    |       | —<br>ex) setProperty("repeat_start_hour", "02")    |
| <b>repeat_start_minute</b>  |       | —<br>ex) setProperty("repeat_start_minute", "05")  |

|                          |  |                                                 |
|--------------------------|--|-------------------------------------------------|
| <b>repeat_end_hour</b>   |  | —<br>ex) setProperty("repeat_end_hour", "09")   |
| <b>repeat_end_minute</b> |  | —<br>ex) setProperty("repeat_end_minute", "35") |

: 6가 2 5 9 35 1 4

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

| Key                      | Value |                                                                  |
|--------------------------|-------|------------------------------------------------------------------|
| <b>user_defined_time</b> |       | ( : ) (.)<br>ex) setProperty("user_defined_time", "01:30,13:30") |

#### ■ CSV

| Key                  | Value                                                                             |                                                         |
|----------------------|-----------------------------------------------------------------------------------|---------------------------------------------------------|
| <b>csv.filename</b>  |                                                                                   | CSV<br>ex) setProperty("csv.filename", "test.csv")      |
| <b>csv.pagetitle</b> | "page"                                                                            | ex) setProperty("csv.pagetitle", "<Page>")              |
| <b>csv.pageline</b>  |                                                                                   | ex) setProperty("csv.pageline ", "7")                   |
| <b>csv.pagestyle</b> | "none"<br>"# <page>"<br>"# <page> --"<br>"-"<br>"---"<br>"# <page>"<br>"<page> #" | ( :none - )<br>ex) setProperty("csv.pagestyle", "none") |
| <b>csv.separator</b> | "Tab"<br>"Space"<br>"Comma"                                                       | CSV<br>ex) setProperty("csv.separator", "Tab")          |

|                            |                   |                                                                                                                                  |
|----------------------------|-------------------|----------------------------------------------------------------------------------------------------------------------------------|
| <b>csv.removeorange</b>    |                   | (<br>ex) setProperty("csv.removeorange", "1, 3")                                                                                 |
| <b>csv.exceptfirstpage</b> | "true"<br>"false" | <ul style="list-style-type: none"> <li>• true :</li> <li>• false : ( )</li> </ul> ex) setProperty("csv.exceptfirstpage", "true") |
| <b>csv.savetointeger</b>   | "true"<br>"false" | <ul style="list-style-type: none"> <li>• true :</li> <li>• false : ( )</li> </ul> ex) setProperty("csv.savetointeger", "true")   |

#### ■ Excel

| Key                       | Value                               |                                                                                                                                                                         |
|---------------------------|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Excel.filename</b>     |                                     | ex) setProperty("excle. filename ", "test.xls")                                                                                                                         |
| <b>Excel.numberformat</b> |                                     | ex) setProperty("excel.numberformat ", "#,##0.00")                                                                                                                      |
| <b>Excel.savefont</b>     |                                     | ex) setProperty("excel.savefont", "Arial, Courier")                                                                                                                     |
| <b>Excel.matchmode</b>    | "columnpersheet"<br>"paperpersheet" | <ul style="list-style-type: none"> <li>• columnpersheet : Sheet ( )</li> <li>• paperpersheet : Sheet</li> </ul> ex)<br>setProperty("excel.matchmode", "columnpersheet") |
| <b>Excel.matchsubmode</b> | "RowFirst"<br>"ColumnFirst"         | ( :RowFirst)<br>ex) setProperty("excel.matchsubmode", "rowfirst")                                                                                                       |
| <b>excel.removeorange</b> |                                     | (<br>ex) setProperty("excel.removeorange", "1,3")                                                                                                                       |

|                           |                                                   |                                                                                                                                          |
|---------------------------|---------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <b>excel.removeoption</b> | "FirstPageOnly"<br>"FirstPageExcept"<br>"AllPage" | excel.removerange가<br><br>• FirstPageOnly :<br>• FirstPageExcept:<br>• AllPage : ( )<br>ex) setProperty("excel.removeoption", "AllPage") |
| <b>excel.removeblank</b>  | "true"<br>"false"                                 | ( :false)<br>ex) setProperty("excel.removeblank", "Yes")                                                                                 |

#### ■ HTML

| Key                    | Value             |                                                                                 |
|------------------------|-------------------|---------------------------------------------------------------------------------|
| <b>html.filename</b>   |                   | HTML<br>ex) setProperty("html.filename", "test.html")                           |
| <b>html.imagepath</b>  |                   | Html<br>(URL )<br>ex) setProperty("html.imagepath", "file://c:/image");         |
| <b>html.vertical</b>   |                   | ( : )<br>ex) setProperty("html.vertical", "1")                                  |
| <b>html.horizontal</b> |                   | ( : )<br>ex) setProperty("html.horizontal", "1")                                |
| <b>html.savebypage</b> | "true"<br>"false" | HTML<br>• true :<br>• false : ( )<br>ex) setProperty("html.savebypage", "true") |
| <b>html.offsetx</b>    |                   | x ( : )<br>ex) setProperty("html.offsetx", "1")                                 |
| <b>html.offsety</b>    |                   | y ( : )<br>ex) setProperty("html.offsety", "1")                                 |

## ■ Jpeg

| Key          | Value |                                                     |
|--------------|-------|-----------------------------------------------------|
| jpg.filename |       | JPEG<br>ex) setProperty("jpg.filename", "test.jpg") |

## ■ OZD

| Key             | Value             |                                                                                                                                                               |
|-----------------|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ozd.filename    |                   | OZD<br>ex) setProperty("ozd. filename ", "test.ozd")                                                                                                          |
| ozd.memoallowed | "true"<br>"false" | <ul style="list-style-type: none"> <li>• true :           가       (       )</li> <li>• false :</li> </ul> ex) setProperty("ozd. memoallowed ", "true")        |
| ozd.saveall     | "true"<br>"false" | <ul style="list-style-type: none"> <li>• true :                               (       )</li> <li>• false :</li> </ul> ex) setProperty("ozd.saveall ", "true") |
| ozd.password    |                   | ex) setProperty("ozd.password", "admin")                                                                                                                      |

## ■ PDF

| Key           | Value |                                                                                                                                                       |
|---------------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| pdf.filename  |       | PDF<br>ex) setProperty("pdf.filename", "test.pdf")                                                                                                    |
| pdf.saverange |       | ("                               ",                               )<br>"                               ")<br>ex) setProperty("pdf.saverange ", "1.3") |
| pdf.title     |       | PDF<br>ex) setProperty("pdf.title", "Report")                                                                                                         |
| pdf.subject   |       | PDF<br>ex) setProperty("pdf.title", "Report")                                                                                                         |
| pdf.creator   |       | ex) setProperty("pdf.creator", "Forks")                                                                                                               |

|                           |   |                                                            |
|---------------------------|---|------------------------------------------------------------|
| <b>pdf.author</b>         |   | ex) setProperty("pdf.author", "Forcs")                     |
| <b>pdf.keyword</b>        |   | PDF<br>ex) setProperty("pdf.keyword", "oz")                |
| <b>pdf.userpassword</b>   |   | PDF<br>ex) setProperty("pdf.userpassword", "user")         |
| <b>pdf.masterpassword</b> |   | PDF<br>ex) setProperty("pdf.masterpassword", "admin")      |
| <b>pdf.printable</b>      | 가 | PDF 가 ( :true)<br>ex) setProperty("pdf.printable", "true") |

#### ■ PPT

| Key                 | Value |                                                    |
|---------------------|-------|----------------------------------------------------|
| <b>ppt.filename</b> |       | PPT<br>ex) setProperty("ppt.filename", "test.ppt") |

#### ■ SVG

| Key                 | Value |                                                    |
|---------------------|-------|----------------------------------------------------|
| <b>svg.filename</b> |       | SVG<br>ex) setProperty("svg.filename", "test.svg") |

#### ■ Text

| Key                   | Value  |                                                      |
|-----------------------|--------|------------------------------------------------------|
| <b>text.filename</b>  |        | Text<br>ex) setProperty("text.filename", "test.txt") |
| <b>text.pagetitle</b> | "page" | ex) setProperty("csv.pagetitle", "<<Page>>")         |
| <b>text.pageline</b>  |        | ex) setProperty("text.pageline ", "7")               |

|                             |                                                                       |                                                                                                                                   |
|-----------------------------|-----------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <b>text.pagestyle</b>       | "none"<br>"# <page>"<br>"# <page> ---"<br>"--- #<page>"<br>"<page> #" | ( :none - )<br>ex) setProperty("text.pagestyle", "none")                                                                          |
| <b>text.separator</b>       | "Tab"<br>"Space"<br>"Comma"                                           | Text<br>ex) setProperty("text.separator", "Tab")                                                                                  |
| <b>text.removepage</b>      |                                                                       | ( , )<br>ex) setProperty("text.removepage", "1, 3")                                                                               |
| <b>text.exceptfirstpage</b> | "true"<br>"false"                                                     | <ul style="list-style-type: none"> <li>• true :</li> <li>• false : ( )</li> </ul> ex) setProperty("text.exceptfirstpage", "true") |
| <b>text.savetointeger</b>   | "true"<br>"false"                                                     | <ul style="list-style-type: none"> <li>• true :</li> <li>• false : ( )</li> </ul> ex) setProperty("text.savetointeger", "true")   |

#### ■ Tiff

| Key                  | Value        |                                                                                                                                                       |
|----------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>tiff.filename</b> |              | Tiff<br>ex) setProperty("tiff.filename", "test.tif")                                                                                                  |
| <b>tiff.encode</b>   | "G3"<br>"G4" | Tiff <ul style="list-style-type: none"> <li>• G3: fax G3<br/>Tiff ( )</li> <li>• G4 : fax G4<br/>Tiff</li> </ul> ex) setProperty("tiff.encode", "G3") |

#### ■ Word

| Key                  | Value |                                              |
|----------------------|-------|----------------------------------------------|
| <b>word.filename</b> |       | ex) setProperty("word.filename", "test.doc") |

```

Value " / . "
"% %\Repository/ "
 , "FORCS" CSV
setProperty("csv.filename", "FORCS/test.csv")
"% %\Repository/FORCS" "test.csv"

```

#### ■ modifyConfiguration ConfigMap

| Key                     | Value             |                                                                                                                                  |
|-------------------------|-------------------|----------------------------------------------------------------------------------------------------------------------------------|
| SchedulerPort           |                   | ( : "9521")<br>ex) p.setProperty("SchedulerPort", "9521");                                                                       |
| SchedulingInfoFilePath  |                   | ex)<br>p.setProperty("SchedulingInfoFilePath", "%SCH_HOME%/ScheduledTask");                                                      |
| SMTPServer              |                   | SMTP<br>ex) p.setProperty("SMTPServer", "mail.forcs.com");                                                                       |
| SMTPServerProt          |                   | SMTP<br>ex) p.setProperty("SMTPServerProt", "25");                                                                               |
| MailFrom                |                   | ex) p.setProperty("MailFrom", "mail@forcs.com");                                                                                 |
| TempRepositoryFilePath  |                   | ex) p.setProperty("TempRepositoryFilePath", "%SCH_HOME%/Temp Repository");                                                       |
| RepositoryFileRootPath  |                   | ex) p.setProperty("RepositoryFileRootPath", "%SCH_HOME%/Repository");                                                            |
| ExternalProgramFilePath |                   | ex) p.setProperty("ExternalProgramFilePath", "%SCH_HOME%/External");                                                             |
| ErrorNotifyToSender     | "true"<br>"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.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 SchedulerSample {
 private static String[][] mailRelatedProperties =
 {
 //
 {"report_name", "test.ozr"}, //
 {"category_name", "/"}, //
 {"cfg.type", "new"}, // Task new, edit

 //
 {"odi_name", "testodi.odi"},
 {"odi_category_name", "/test"},
 {"dm_server_check", "check"}, //
 {"dm_server_name", "test1.sdm"}, //

 //
 {"odi_name", "testodi.odi"}, // odi
 {"odi_category_name", "/test"}, // odi 가
 {"external_program_check", "check"},
 {"external_program_command",
 "notepad.bat"}, // .

 //
 {"parameter_count", "1"}, //
 {"parameter_name_1",
 "[FORM].empNo"}, //
 {"parameter_value_1", "10"}, //

 // ODI
 {"odi_parameter_count", "1"}, //
 {"odi_parameter_name_1", "id"}, //
 {"odi_parameter_value_1", "12"}, //

 //
 }
```

```
 {"launch_type", "once"},
 // ("immediately", "periodically")
 {"execution_year", "2003"},// -
 {"execution_month", "12"}, // -
 {"execution_day", "17"}, // -
 {"execution_hour", "10"}, // -
 {"execution_min", "30"}, // -
 {"periodically_execution_day_type", "daily"},
 // ("weekly", "monthly")
 {"daily_type", "weekday"}, // ("specific_day")
 {"periodically_execution_time_type", "once"},
 // ("repeat", "user_defined")
 {"once_hour", "01"}, // -
 {"once_min", "00"}, // -

 //
 {"mail_check", "check"}, // -
 {"html_mail_content", "check"}, // html
 {"mail_notify_error_check", "null"}, // -
 {"mail_recipient_to",
 "gil_dong@forcs.com"}, //
 {"mail_recipient_cc", "abc@forcs.com"}, //
 {"mail_recipient_bcc", "lan@forcs.com"}, //
 {"mail_subject", " "}, //
 {"mail_text_message", " "}, //
 {"mail_attach_list",
 "csv/excel/html/ozd/pdf/text/tiff/word/ppt/jpg/svg"},
 //
 {"file_export_list", "csv/xls/html/ozd/pdf/txt/tif/doc/ppt/jpg/svg"}
 // Export
 };

 private static String[][] attachRelatedProperties =
 {
 // CSV
 {"csv.filename ", "test.csv"}, // csv
 {"csv.pagetitle ", "page"}, //
 {"csv.pageline ", "7"}, //
 {"csv.pagestyle", "none"}, //
 {"csv.separator ", "Tab"}, // CSV
 {"csv.removerange ", "1,3"}, // CSV
 {"csv.exceptfirstpage ", "true"}, //
 {"csv.savetointeger ", "true"}, //

 //
 {"excel.filename ", "test.xls"}, // excel
 {"excel.numberformat ", "#,##0.00"}, //
 {"excel.savefont", "Arial"}, //
 }
```

```

{"excel.matchmode",
 "columnpersheet"}, //
{"excel.matchsubmode", "rowfirst"}, //
{"excel.removerange", "1,3"}, //
{"excel.removeoption ",
 " firstpageonly"}, //
{"excel.removeblank ", " true"}, //

// html
{"html.filename ", "test.html"}, // html
{"html.imagepath ",
 "file://c:/image"}, // html
{"html.vertical ", "1"}, // html
{"html.horizontal ", "1"}, // html
{"html.savebypage ", "true"}, // html
{"html.offsetx ", "1"}, // x offset
{"html.offsety ", "1"}, // y offset

// ozd
{"ozd.filename ", "test.ozd"}, //OZD
{"ozd.memoallowed ", "true"}, //
{"ozd.password", "admin"}, // ozd

// pdf
{"pdf.filename ", "test.pdf"}, //pdf
{"pdf.saverange ", "1.3"}, //pdf
{"pdf.title ", "Report"}, //pdf
{"pdf.subject ", "OZ"}, //pdf
{"pdf.creator ", "hong"}, //pdf
{"pdf.author ", "hong"}, //pdf
{"pdf.keyword ", "oz"}, //pdf
{"pdf.userpassword", "user"}, //pdf
{"pdf.masterpassword ", "admin"},//pdf
{"pdf.printprotected ", "true"}, //pdf

// text
{"text.filename ", "test.txt"}, //text
{"text.pagetitle ", "page"}, //
{"text.pageline ", "7"}, //
{"text.pagestyle", "none"}, //
{"text.separator ", "Tab"}, //text
{"text.removerange ", "1,3"}, //text
{"text.exceptfirstpage ", "true"}, //
{"text.savetointeger ", "true"}, //

// Tiff
{"tiff.filename ", "test.tif"},//tiff
{"tiff.encode ", "G3"}, //tiff

```

```
// word
{"word.filename ", "test.doc"}, //word

// ppt
{"ppt.filename ", "test.ppt "}, //ppt

// jpg
{"jpg.filename ", "test.jpg "}, //jpg

// svg
{"svg.filename ", "test.svg"} //svg
};

private static String[][] schedulerRelatedProperties =
{
 {"SchedulerPort", "9521"}, //
 {"schedulerserver",
 "%SCH_HOME%/ScheduledTask"}, //
 {"SMTPServer", "mail.forcs.com"}, //SMTP
 {"SMTPServerPort", "25"}, //SMTP
 {"MailFrom", "kil_dong@forcs.com"}, //
 {"TempRepositoryFilePath",
 "%SCH_HOME%/TempRepository"}, //
 {"RepositoryFileRootPath",
 "%SCH_HOME%/Repository"}, //PDF
 {"ExternalProgramFilePath",
 "%SCH_HOME%/External"}, //
 {"ErrorNotifyToSender", "false"} //
};

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

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

 String[][] values = null;
 Scheduler scheduler = null;
 try {
 scheduler = new Scheduler(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 .*/

// (createTask)

//
//
//check, uncheck 가 "check", "null" .
SortProperties props = new SortProperties();
setProperties(props, mailRelatedProperties);

SortProperties exportMap = new SortProperties();
setProperties(exportMap, attachRelatedProperties);
scheduler.createTask(serverInfo, props, exportMap);

scheduler.makePDF(serverInfo, props, exportMap);
scheduler.makePDFByPooling(serverInfo, props, exportMap);

// (modifyConfiguration)
SortProperties pro = new SortProperties();
setProperties(pro, schedulerRelatedProperties);
scheduler.modifyConfiguration(serverInfo, pro);

// (getConfiguration)
pro = scheduler.getConfiguration(serverInfo);
pro.list(System.out);

/*
 * (getTask)
 * →
 *
 * ,
 *
 *
 */
ScheduledTask sTask = new ScheduledTask();

// setting
sTask.taskID = "parameter_test.ozs"; //
sTask.reportName = "/parameter_test"; //
sTask.schedulingTypeStr = "Periodically"; //
sTask.lastRunTimeStr = ""; //
sTask.nextRunTimeStr = "2005-06-08 19:00";//
sTask.status = 'W'; // (WAITING, RUNNING, PAUSE 가 .)

```

```
ScheduledTask[] taskList = scheduler.getTask(serverInfo);
for (int i = 0; i < taskList.length; i++) {
 showTask(taskList[i]);
}

/*
 * (getTaskResult)
 * →
 * 가
 * .
 */
//
TaskResult tr = new TaskResult();
tr.taskID = "parameter_test.ozs"; //
tr.completedTime = "2005-06-08 19:00"; //
tr.isSuccessful = "true"; //
tr.formFileName = "/parameter_test"; //
tr.paramInfo = "0{}"; //
tr.schedulingType = "Periodically"; //
// - 가
//
tr.errorMsg = "Fail to execute scheduling task with OZServer. :";
String from = "2005-06-08"; //
String to = "2005-06-30"; //
TaskResult[] trList =
 scheduler.getTaskResult(serverInfo, from, to, tr.taskID);

for (int i = 0; i < trList.length; i++) {
 showTaskResult(trList[i]);
}

String task = sTask.taskID; //
// (taskPause)
boolean b = scheduler.taskPause(serverInfo, task);
// (taskResume)
boolean bool = scheduler.taskResume(serverInfo, task);
// (removeTask)
scheduler.removeTask(serverInfo, task);
// .(stop)
boolean waitTask = true; // boolean
scheduler.stop(serverInfo, waitTask);
}
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]);
 }
}

private static void showTask(ScheduledTask t) {
 System.out.println(" " + t.taskID);
 System.out.println(" " + t.reportName);
 System.out.println(" " + t.schedulingTypeStr);
 System.out.println(" " + t.lastRunTimeStr);
 System.out.println(" " + t.nextRunTimeStr);
 System.out.println(" " + t.status);
 System.out.println("");
}

private static void showTaskResult(TaskResult tr) {
 System.out.println(" " + tr.taskID);
 System.out.println(" " + tr.completedTime);
 System.out.println(" " + tr.isSuccessful);
 System.out.println(" " + tr.formFileName);
 System.out.println(" " + tr.paramInfo);
 System.out.println(" " + tr.schedulingType);
 System.out.println(" : " + tr.exportFileList);
 System.out.println(" " + tr.errorMsg);
 System.out.println("");
}
}
```

### 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) throws SchedulerException {
 // 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");

 // (: "immediately", : "periodically")
 config.setProperty("launch_type", "immediately");

 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", "odiparam1= ");
export.setProperty("odi.parameter_test.args2", "odiparam2= ");
export.setProperty("odi.odinames", "parameter_test");
export.setProperty("export.format",
 "csv/xls/html/ozd/pdf/txt/tif/doc/ppt/jpg/svg");
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("tiff.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");

// 가
export.setProperty("viewer.childcount", "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.odinames", "parameter_test");
export.setProperty("child1.odi.parameter_test.pcount", "2");
export.setProperty("child1.odi.parameter_test.args1", "odiparam1= 3");
export.setProperty("child1.odi.parameter_test.args2", "odiparam2= 4");

//
export.setProperty("child1.export.format",
 "csv/xls/html/ozd/pdf/txt/tif/doc/ppt/jpg/svg");
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");

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

}
}
```

### 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 Succeded Code : " + tr.isSuccessfulCode);
 System.out.println("Is Succeded? : " + 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.errorMsg);
 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
가
 .
 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 (Exception 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");
```

```

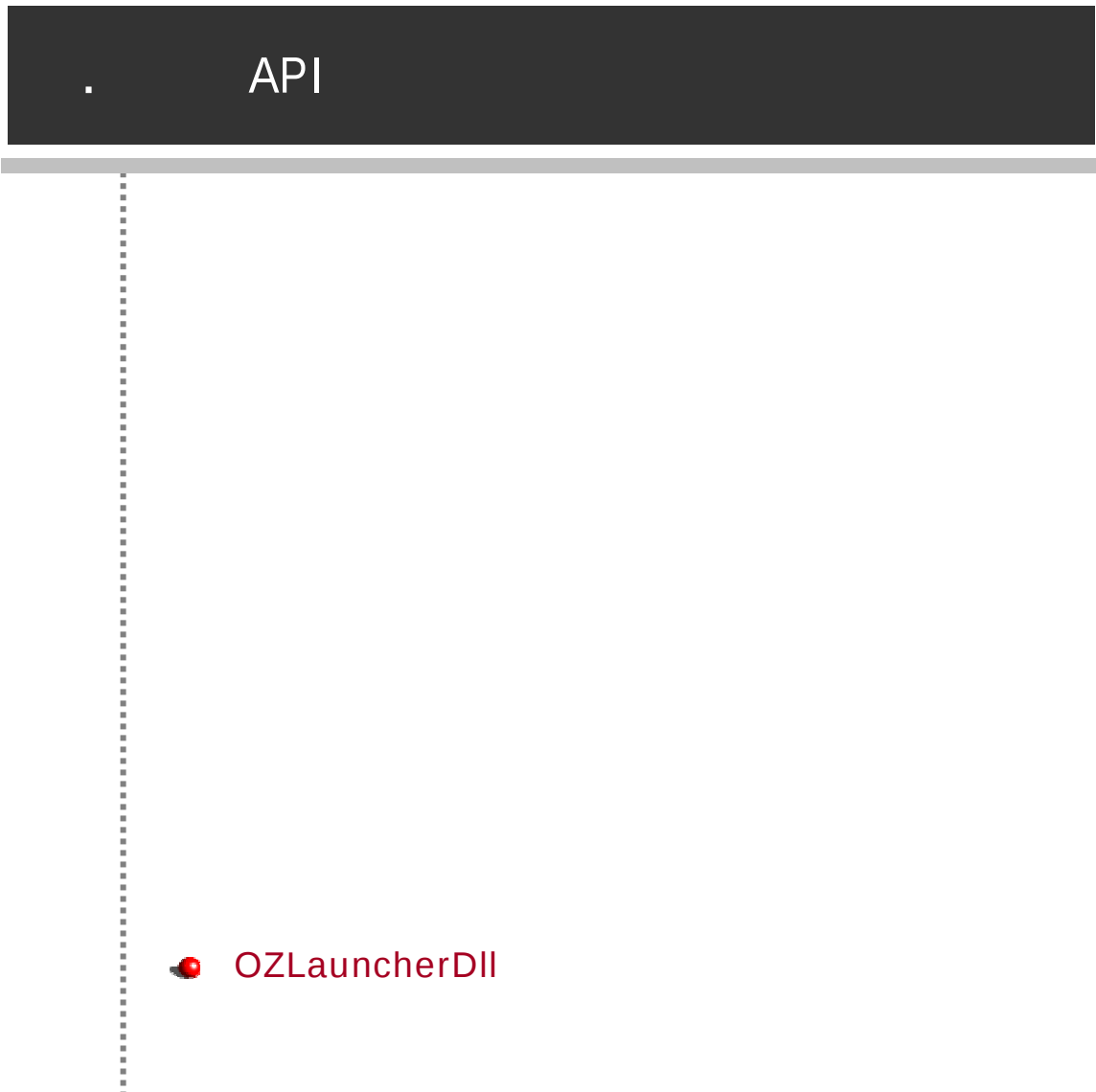
 printMap.setProperty("odi.parameter_test.pcount", "2");
 printMap.setProperty("odi.parameter_test.args1", "odiparam1= ");
 printMap.setProperty("odi.parameter_test.args2", "odiparam2= ");
 printMap.setProperty("odi.odinames", "parameter_test");

 //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.spoolpages", "100");
 printMap.setProperty("print.style", "normal");
 printMap.setProperty("print.usedefaultpaper", "false");
 printMap.setProperty("print.spoolpages", "100");
 printMap.setProperty("print.usedialogopt", "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 .
 scheduler.print(serverInfo, config, printMap);
 } catch (SchedulerException se) {
 se.printStackTrace();
 }
}
}
}

```



OZLauncherDll

OZLauncherDll

Function Summary

- void\_stdcall SetPath(LPCTSTR strpath)
- void\_stdcall SetCommand(LPCTSTR strcommand)
- BOOL\_stdcall CreateOZViewer(LPCTSTR str\_param, int n\_type)
- void\_stdcall Release()

Function Detail

|            |                                                                  |
|------------|------------------------------------------------------------------|
| ■          | SetPath                                                          |
| Prototype  | void_stdcall SetPath(LPCTSTR strpath)                            |
| Definition | .                                                                |
| Argument   | <i>strpath</i>                                                   |
| ■          | SetCommand                                                       |
| Prototype  | void_stdcall SetCommand(LPCTSTR strcommand)                      |
| Definition | .                                                                |
| Argument   | <i>strcommand</i>                                                |
| Example    | SetCommand("/locale ko/kr /mode alone /slp true /launchstring"); |
| ■          | CreateOZViewer                                                   |
| Prototype  | BOOL_stdcall CreateOZViewer(LPCTSTR str_param, int n_type)       |
| Definition | .true<br>false.                                                  |

|          |                                                                                                                             |      |
|----------|-----------------------------------------------------------------------------------------------------------------------------|------|
| Argument | <i>str_param</i>                                                                                                            | " n" |
|          | <i>n_type</i>                                                                                                               | 2    |
| Example  | <pre>CreateOZViewer("connection.server=127.0.0.1\n connection.port=8003\n connection.reportname=/ozsample.ozr\n", 2);</pre> |      |

#### ■ Release()

|            |                                                                                                                                                                                                                                                                                                      |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Prototype  | void __stdcall Release()                                                                                                                                                                                                                                                                             |
| Definition |                                                                                                                                                                                                                                                                                                      |
| Example    | <pre>SetCommand("/string"); SetPath("./"); CreateOZViewer("connection.server=127.0.0.1\n toolbar.all=true\n information.debug=debug\n information.bmt=true\n connection.port=8003\n connection.reportname=/ozsample.ozr\n connection.compressedForm=true",1); result = GetResult(); Release();</pre> |

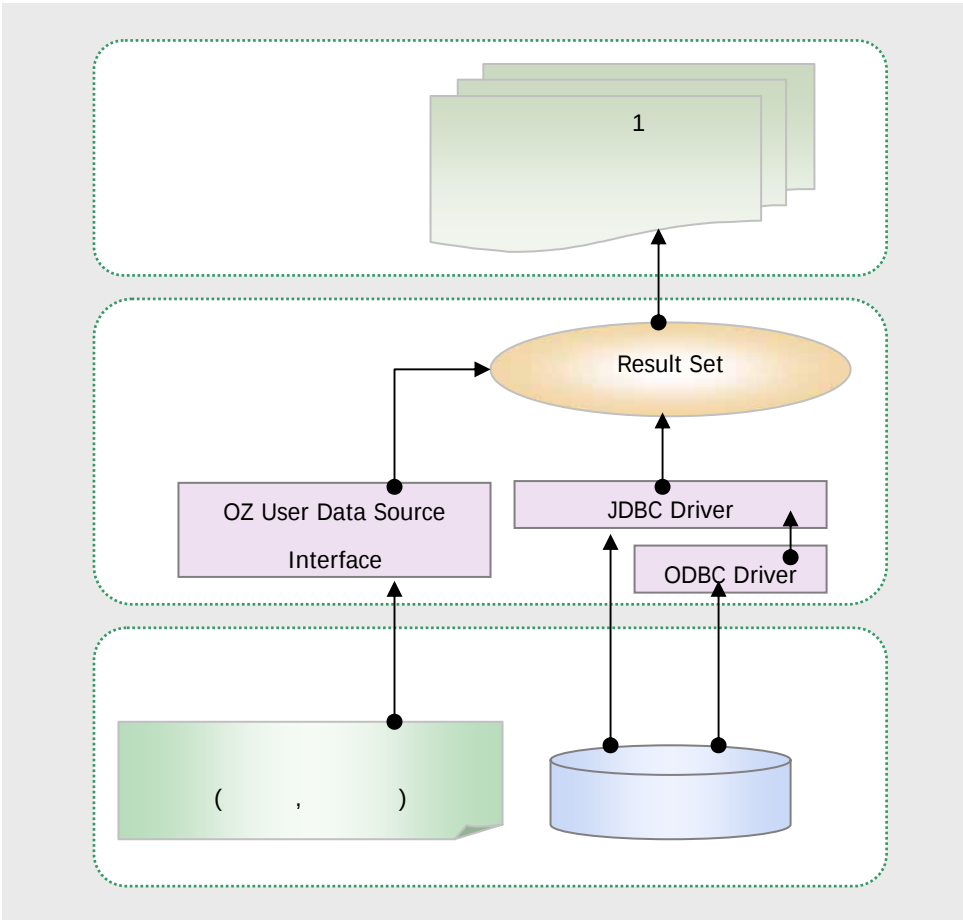


## . User Data Store

 UDS UDS UDS

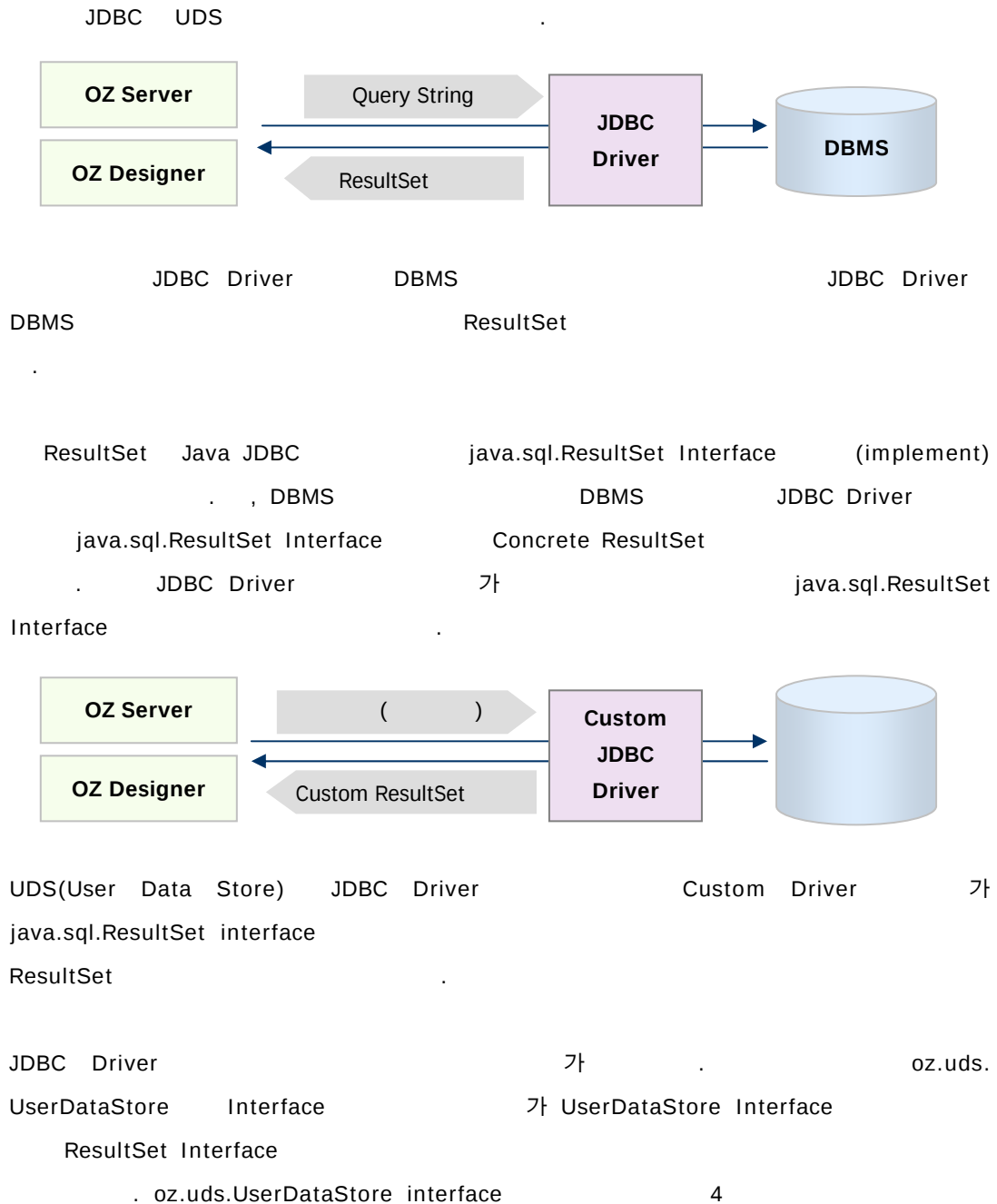
UDS

UDS(User Data Store) 가 Non-DB Java Interface  
Servlet, ASP, JSP, Applet , CSV, XML EJB,



UDS  
가 ResultSet  
SQL  
. UDS

## UDS



```
package oz.uds;

import java.sql.*;

public interface OZUserDataStore {
 public void init() throws OZUserDataStoreException;
 public ResultSet getResultSet(String argument) throws
 OZUserDataStoreException;
 public void freeResultSet(ResultSet rst);
 public void close();
}
```

| <b>void init()</b>                             | UDS가                                |
|------------------------------------------------|-------------------------------------|
| <b>ResultSet getResultSet(String argument)</b> | Argument( )<br>ResultSet            |
| <b>void freeResultSet(ResultSet rst)</b>       | ResultSet<br>getResultSet ResultSet |
| <b>void close()</b>                            | UDS                                 |

: init(), close() 가

## UDS

UDS 가 getResultSet

### UDS Source

#### ■ UDS Main

Argument

```
import oz.uds.*;
import java.sql.ResultSet;

public class Extdata implements OZUserDataStore{

 public void init() throws OZUserDataStoreException {
 // UDS가 DB
 }

 public ResultSet getResultSet(String argument) throws OZUserDataStoreException
 {
 // Argument Resultset
 // Argument OZ
 }

 public void freeResultSet(ResultSet rst){
 // getResultSet ResultSet
 // ResultSet Close
 }

 public void close() {
 // UserDataSet
 // DB Disconnection
 }
}
```

## ■ ResultSet

ResultSet  
가 .  
ResultSet . ( .)  
JDBC `java.sql.ResultSet` Interface 30 가  
UDS  
가 .

```
import java.sql.*;
import java.math.BigDecimal;
import java.util.Calendar;
import java.util.Enumuration;

public class ResultSet_User implements ResultSet{

 public ResultSet_User() { }
 public boolean next() throws SQLException{
 //
 }

 public String getString(String columnName) throws SQLException{
 //
 }

 // getString
 // Method 가
 // getBoolean(String columnName)
 // getByte(String columnName)
 // getShort(String columnName)
 // getInt(String columnName)
 // getLong(String columnName)
 // getFloat(String columnName)
 // getDouble(String columnName)
 // getBigDecimal(String columnName)
 // getDate(String columnName)
 // getTime(String columnName)
 // getTimestamp(String columnName)
 // getBlob(String columnName)
 // getClob(String columnName)
 // getBinaryStream(String columnName)

 public ResultSetMetaData getMetaData() throws SQLException{
```

```
// ResultSetMetaData
// ResultSetMetaData Method
}
}
```

#### ■ ResultSetMetaData

ResultSetMetaData  
가

```
import java.sql.SQLException;
import java.sql.Types;
import java.util.*;

public class ResultSetMetaData_User implements ResultSetMetaData{

 public ResultSetMetaData_User(String[] meta) { }

 public int getColumnCount() throws SQLException{
 //
 }

 public String getColumnName(int column) throws SQLException{
 // (column)
 }

 public int getColumnType(int column) throws SQLException{
 // (column)
 }
}
```

## UDS

UDS

```
<UdsFile.java>

import oz.uds.*;
import java.sql.ResultSet;
import java.io.FileReader;
import java.io.IOException;
import java.util.StringTokenizer;
import java.util.Vector;
```

```

public class UdsFile implements OZUserDataStore{
 private StringBuffer Sbuffer;
 private Vector mainV; //subV
 private Vector subV;
 private String[][] data;
 private String[] meta;

 private static final String rootFileName = "";
 private static final String mainDiv = "\n";
 private static final String subDiv = "\",";

 public void init() throws OZUserDataStoreException {
 mainV = new Vector();
 }

 public ResultSet getResultSet(String command) throws
 OZUserDataStoreException {
 FileReader in = null;
 Sbuffer = new StringBuffer();
 try{
 in = new FileReader(rootFileName + command);
 char[] buffer = new char[256];
 int n;
 while((n = in.read(buffer)) > -1){
 Sbuffer.append(new String(buffer, 0, n));
 }
 } catch(IOException e){
 e.printStackTrace();
 throw new OZUserDataStoreException(e.getMessage());
 } finally {
 try{
 if(in != null)in.close();
 } catch(IOException e){
 } finally{
 in = null;
 }
 }
 String s = Sbuffer.toString();
 //////////////////////////////////////
 StringTokenizer st1 = new StringTokenizer(s,mainDiv);
 StringTokenizer st2;
 String tmpString;
 //////////////////////////////////////meta
 if(st1.hasMoreTokens()){
 tmpString = st1.nextToken().trim();
 st2 = new StringTokenizer(tmpString,subDiv);
 subV = new Vector();

```

```
 while (st2.hasMoreTokens()){
 subV.addElement(st2.nextToken().trim());
 }
 meta = new String[subV.size()];
 subV.copyInto(meta);
 }
 //////////////////////////////////////
 while (st1.hasMoreTokens()) {
 tmpString = st1.nextToken().trim();
 st2 = new StringTokenizer(tmpString, subDiv);
 subV = new Vector();
 while (st2.hasMoreTokens()){
 subV.addElement(st2.nextToken().trim());
 }
 mainV.addElement(subV);
 }
 data = new String[mainV.size()][];
 for(int i = 0; i < mainV.size(); i++){
 subV = (Vector)mainV.elementAt(i);
 data[i] = new String[subV.size()];
 subV.copyInto(data[i]);
 }
 Sbuffer = null;
 mainV.clear();
 //////////////////////////////////////
 if(meta == null || data == null){
 throw new OZUserDataStoreException("Data is Invalide");
 }
 return new ResultSet_File(meta, data);
}
public void freeResultSet(ResultSet rst){
 Sbuffer = null;
 data = null;
 meta = null;
 mainV.clear();
}
public void close() {
 Sbuffer = null;
 data = null;
 meta = null;
 mainV = null;
}
}

<ResultSet_File.java>

import java.sql.*;
import java.math.BigDecimal;
```

```
import java.util.Calendar;
import java.util.Enumeration;

public class ResultSet_File implements ResultSet{

 private String[][] data;
 private String[] meta;
 private int index;
 private int size;

 public ResultSet_File(String[] meta, String[][] data) {
 this.meta = meta;
 this.data = data;
 size = data.length;
 index = -1;
 }
 public boolean next() throws SQLException{
 index++;
 if(index == size) return false;
 return true;
 }
 public String getString(String columnName) throws SQLException{
 int i = findIndex(columnName);
 if(i > -1){
 try{
 return data[index][i];
 }catch(Exception e){
 throw new SQLException("Internal Error");
 }
 }else{
 return null;
 }
 }
 private final int findIndex(String columnName){
 for(int i = 0; i < meta.length; i++){
 if(columnName.equals(meta[i])){
 return i;
 }
 }
 return -1;
 }
 public String getString(int columnIndex) throws SQLException{
 return null;
 }
 public ResultSetMetaData getMetaData() throws SQLException{
 return new ResultSetMetaData_File(meta);
 }
}
//
```

```
// 가 .
public boolean getBoolean(String columnName) throws SQLException{
 return false;
}
public byte getByte(String columnName) throws SQLException{
 return -1;
}
public short getShort(String columnName) throws SQLException{
 return -1;
}
public int getInt(String columnName) throws SQLException{
 return -1;
}
public long getLong(String columnName) throws SQLException{
 return -1;
}
public float getFloat(String columnName) throws SQLException{
 return -1;
}
public double getDouble(String columnName) throws SQLException{
 return -1;
}
public BigDecimal getBigDecimal(String columnName) throws SQLException{
 return null;
}
public java.sql.Date getDate(String columnName) throws SQLException{
 return null;
}
public java.sql.Time getTime(String columnName) throws SQLException{
 return null;
}
public java.sql.Timestamp getTimestamp(String columnName) throws
SQLException{
 return null;
}
public Blob getBlob(String columnName) throws SQLException{
 return null;
}
public Clob getClob(String columnName) throws SQLException{
 return null;
}
public java.io.InputStream getBinaryStream(String columnName)
throws SQLException{
 return null;
}
}
////////////////////////////////////
public boolean getBoolean(int columnIndex) throws
SQLException{return false;}
```

```

public byte getByte(int columnIndex) throws SQLException{return -1;}
public short getShort(int columnIndex) throws SQLException{return -1;}
public int getInt(int columnIndex) throws SQLException{return -1;}
public long getLong(int columnIndex) throws SQLException{return -1;}
public float getFloat(int columnIndex) throws SQLException{return -1;}
public double getDouble(int columnIndex) throws SQLException{return -1;}
public BigDecimal getBigDecimal(int columnIndex) throws
 SQLException{return null;}
public java.sql.Date getDate(int columnIndex) throws
 SQLException{return null;}
public java.sql.Time getTime(int columnIndex) throws
 SQLException{return null;}
public java.sql.Timestamp getTimestamp(int columnIndex) throws
 SQLException{return null;}
public Blob getBlob(int columnIndex) throws SQLException{return null;}
public Clob getClob(int columnIndex) throws SQLException{return null;}
public java.io.InputStream getBinaryStream(int columnIndex)
 throws SQLException{return null;}
////////////////////////////////////
public void close() throws SQLException{}
public boolean wasNull() throws SQLException{return false;}
public byte[] getBytes(int columnIndex) throws SQLException{return null;}
public java.io.InputStream getAsciiStream(int columnIndex) throws
 SQLException{ return null;}
public java.io.InputStream getUnicodeStream(int columnIndex) throws
 SQLException{return null;}
public byte[] getBytes(String columnName) throws SQLException{return null;}
public java.io.InputStream getAsciiStream(String columnName) throws
 SQLException{return null;}
public java.io.InputStream getUnicodeStream(String columnName) throws
 SQLException{return null;}
public SQLWarning getWarnings() throws SQLException{return null;}
public void clearWarnings() throws SQLException{}
public String getCursorName() throws SQLException{return null;}
public Object getObject(int columnIndex) throws SQLException{return null;}
public Object getObject(String columnName) throws SQLException{return null;}
public int findColumn(String columnName) throws SQLException{return -1;}
public java.io.Reader getCharacterStream(int columnIndex) throws
 SQLException{return null;}
public java.io.Reader getCharacterStream(String columnName) throws
 SQLException{return null;}
public BigDecimal getBigDecimal(int columnIndex, int scale) throws
 SQLException{return null;}
public BigDecimal getBigDecimal(String columnName, int scale) throws
 SQLException{return null;}
public boolean isBeforeFirst() throws SQLException{return false;}
public boolean isAfterLast() throws SQLException{return false;}
public boolean isFirst() throws SQLException{return false;}

```

```
public boolean isLast() throws SQLException{return false;}
public void beforeFirst() throws SQLException{}
public void afterLast() throws SQLException{}
public boolean first() throws SQLException{return false;}
public boolean last() throws SQLException{return false;}
public int getRow() throws SQLException{return -1;}
public boolean absolute(int row) throws SQLException{return false;}
public boolean relative(int rows) throws SQLException{return false;}
public boolean previous() throws SQLException{return false;}
public void setFetchDirection(int direction) throws SQLException{}
public int getFetchDirection() throws SQLException{return -1;}
public void setFetchSize(int rows) throws SQLException{}
public int getFetchSize() throws SQLException{return -1;}
public int getType() throws SQLException{return -1;}
public int getConcurrency() throws SQLException{return -1;}
public boolean rowUpdated() throws SQLException{return false;}
public boolean rowInserted() throws SQLException{return false;}
public boolean rowDeleted() throws SQLException{return false;}
public Statement getStatement() throws SQLException{return null;}
public Object getObject(int i, java.util.Map map) throws
 SQLException{return null;}
public Ref getRef(int i) throws SQLException{return null;}
public Array getArray(int i) throws SQLException{return null;}
public Object getObject(String colName, java.util.Map map) throws
 SQLException{return null;}
public Ref getRef(String colName) throws SQLException{return null;}
public Array getArray(String colName) throws SQLException{return null;}
public java.sql.Date getDate(int columnIndex, Calendar cal) throws
 SQLException{return null;}
public java.sql.Date getDate(String columnName, Calendar cal) throws
 SQLException{return null;}
public java.sql.Time getTime(int columnIndex, Calendar cal) throws
 SQLException{return null;}
public java.sql.Time getTime(String columnName, Calendar cal) throws
 SQLException{return null;}
public java.sql.Timestamp getTimestamp(int columnIndex, Calendar cal)
 throws SQLException{return null;}
public java.sql.Timestamp getTimestamp(String columnName, Calendar cal)
 throws SQLException{return null;}
////////////////////////////////////
public void updateNull(int columnIndex) throws SQLException{}
public void updateBoolean(int columnIndex, boolean x) throws SQLException{}
public void updateByte(int columnIndex, byte x) throws SQLException{}
public void updateShort(int columnIndex, short x) throws SQLException{}
public void updateInt(int columnIndex, int x) throws SQLException{}
public void updateLong(int columnIndex, long x) throws SQLException{}
public void updateFloat(int columnIndex, float x) throws SQLException{}
public void updateDouble(int columnIndex, double x) throws SQLException{}
```

```
public void updateBigDecimal(int columnIndex, BigDecimal x) throws
 SQLException{}
public void updateString(int columnIndex, String x) throws SQLException{}
public void updateBytes(int columnIndex, byte x[]) throws SQLException{}
public void updateDate(int columnIndex, java.sql.Date x) throws
 SQLException{}
public void updateTime(int columnIndex, java.sql.Time x) throws
 SQLException{}
public void updateTimestamp(int columnIndex, java.sql.Timestamp x)
 throws SQLException{}
public void updateAsciiStream(int columnIndex, java.io.InputStream x,
 int length) throws SQLException{}
public void updateBinaryStream(int columnIndex, java.io.InputStream x,
 int length) throws SQLException{}
public void updateCharacterStream(int columnIndex, java.io.Reader x,
 int length) throws SQLException{}
public void updateObject(int columnIndex, Object x, int scale)
 throws SQLException{}
public void updateObject(int columnIndex, Object x) throws SQLException{}
public void updateNull(String columnName) throws SQLException{}
public void updateBoolean(String columnName, boolean x) throws
 SQLException{}
public void updateByte(String columnName, byte x) throws SQLException{}
public void updateShort(String columnName, short x) throws SQLException{}
public void updateInt(String columnName, int x) throws SQLException{}
public void updateLong(String columnName, long x) throws SQLException{}
public void updateFloat(String columnName, float x) throws SQLException{}
public void updateDouble(String columnName, double x) throws SQLException{}
public void updateBigDecimal(String columnName, BigDecimal x) throws
 SQLException{}
public void updateString(String columnName, String x) throws SQLException{}
public void updateBytes(String columnName, byte x[]) throws SQLException{}
public void updateDate(String columnName, java.sql.Date x) throws
 SQLException{}
public void updateTime(String columnName, java.sql.Time x) throws
 SQLException{}
public void updateTimestamp(String columnName, java.sql.Timestamp x)
 throws SQLException{}
public void updateAsciiStream(String columnName, java.io.InputStream x,
 int length) throws SQLException{}
public void updateBinaryStream(String columnName, java.io.InputStream x,
 int length) throws SQLException{}
public void updateCharacterStream(String columnName, java.io.Reader reader,
 int length) throws SQLException{}
public void updateObject(String columnName, Object x, int scale)
 throws SQLException{}
public void updateObject(String columnName, Object x) throws SQLException{}
public void insertRow() throws SQLException{}
```

```
public void updateRow() throws SQLException{}
public void deleteRow() throws SQLException{}
public void refreshRow() throws SQLException{}
public void cancelRowUpdates() throws SQLException{}
public void moveToInsertRow() throws SQLException{}
public void moveToCurrentRow() throws SQLException{}
////////////////////////////////////
}

<ResultSetMetaData_File.java>

import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.sql.Types;
import java.util.*;

public class ResultSetMetaData_File implements ResultSetMetaData{
 String[] meta;
 public ResultSetMetaData_File(String[] meta) {
 this.meta = meta;
 }
 public int getColumnCount() throws SQLException{
 return meta.length;
 }
 public String getColumnName(int column) throws SQLException{
 if(0 < column && column < getColumnCount()+1){
 return meta[column-1];
 }else{
 throw new SQLException();
 }
 }
 public int getColumnType(int column) throws SQLException{
 return Types.VARCHAR;
 }
 public int isNullable(int column) throws SQLException{return -1;}
 public int getColumnDisplaySize(int column) throws SQLException{return -1;}
 public int getPrecision(int column) throws SQLException{return -1;}
 public int getScale(int column) throws SQLException{return -1;}
 public boolean isAutoIncrement(int column) throws SQLException{
 return false;}
 public boolean isCaseSensitive(int column) throws SQLException{
 return false;}
 public boolean isSearchable(int column) throws SQLException{return false;}
 public boolean isCurrency(int column) throws SQLException{return false;}
 public boolean isSigned(int column) throws SQLException{return false;}
 public boolean isReadOnly(int column) throws SQLException{return false;}
 public boolean isWritable(int column) throws SQLException{return false;}
 public boolean isDefinitelyWritable(int column) throws SQLException{
```

```
 return false;}
 public String getColumnLabel(int column) throws SQLException{return null;}
 public String getSchemaName(int column) throws SQLException{return null;}
 public String getTableName(int column) throws SQLException{return null;}
 public String getCatalogName(int column) throws SQLException{return null;}
 public String getColumnTypeName(int column) throws SQLException{
 return null;}
 public String getColumnClassName(int column) throws SQLException{
 return null;}
}
```

## . User Security Logic

 USL USL USL

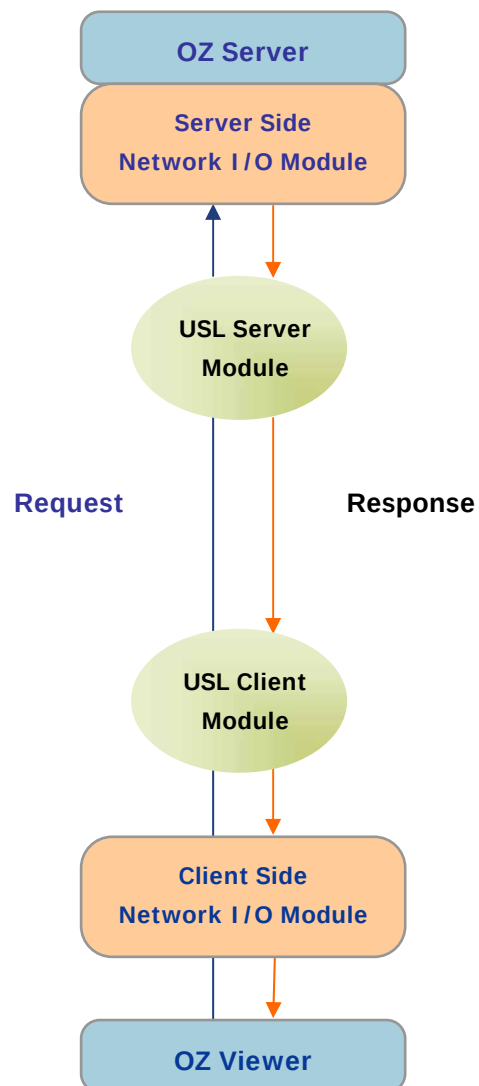
## USL

USL(User Security Logic)

```

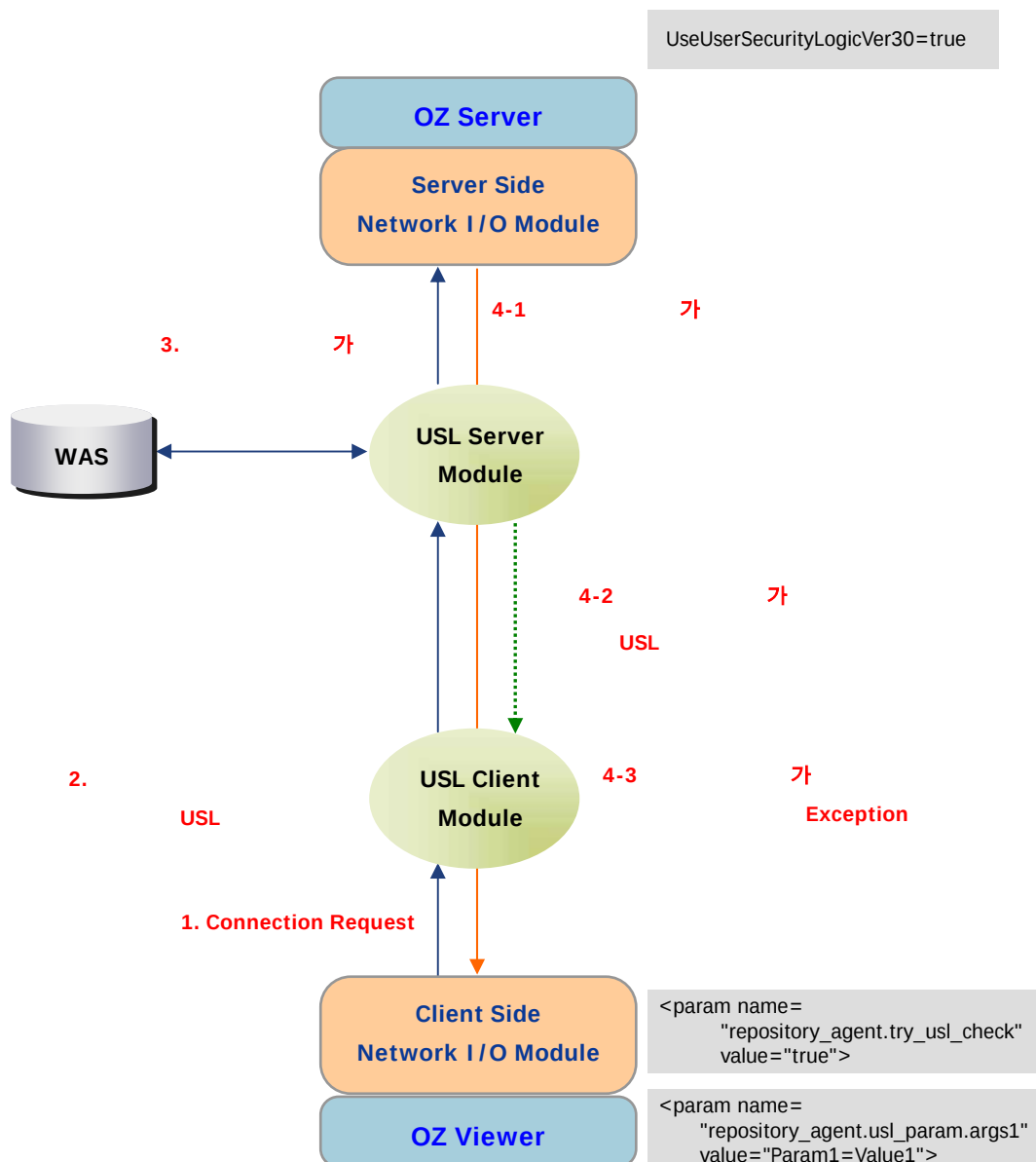
. USL USL Server USL Client
I/O
.
```

## USL Flow



| USL     | Web    | WAS                     |
|---------|--------|-------------------------|
| Session | Cookie | (가), Network I/O Stream |
| •       | , PKI  | .                       |

**Session ( Session USL )**



## USL

USL



- USL : OZ\_HOME/lib/ozsfw40.jar
- USL : OZ\_HOME/conf/uslmngr.properties



- ozsfw40.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/ozservlet/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>
```

## USL

### USL

USL                      USL                      Request Stream  
Response Stream                      .

#### ■ USL

oz.cp.OZUSLServer                      createSecureOutputStream(),  
createSecureInputStream()                      .  
Request Stream  
createSecureInputStream()                      가  
Response Stream                      createSecureOutputStream()

#### ■ InputStream

(                      )  
InputStream                      .  
java.io.InputStream                      read()  
USL                      DataInputStream                      read()가  
DataInputStream                      .  
가  
read가                      .

#### ■ OutputStream

(                      )                      OutputStream  
.  
java.io.OutputStream                      write(), flush()  
USL                      DataOutputStream                      write  
DataOutputStream  
.  
write()                      가 flush

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.datafactory.util.OZAttributeList;
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("oz40.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){
 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"));
 }

 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();
 String value = (String)http_session.getAttribute(key);
 params_to_client.put(key, value);
 cat.debug("OZUSLServerSession:http_session: "+key+", "+value);
 }
 }
 params_to_client.write(new OZDataOutputStream(raw_out));
 }catch(Exception e){
 cat.error("OZUSLServerSession: write params to client:" , e);
 throw new OzcmException(e.getMessage());
 }
 return null;
}
}
```

Servlet

WAS /oz

(Servlet)

WEB-

```

 INF 가 WEB-INF\sessions
 .
 , http://localhost:8100/setcookie.jsp
 /http://localhost:8100/test/getcookie.jsp WAS
 /oz , http://localhost:8100/oz/getcookie.jsp
 . 가 OZ
 .
 ■
 WAS
 .
 Process /oz
 .
 Cookie
 .

```

```

■ USL -
OZUSLClientSession OZUSLClient
createSecureInputStream InputStream
.

```

```

<<OZUSLClientSession.cpp>>

#include <stdafx.h>
#include <oz/usl/OZUSLClient.h>
#include <oz/usl/OZUSLClientSession.h>
#include <common/io/PC1Encode128OutputStream.h>
#include <common/io/PC1Decode128InputStream.h>
#include <common/io/OZFuncInputStream.h>
#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 bInit)
{
 __super::createSecureInputStream(raw_in, params_from_server_to_fillup, bInit);
 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(Write)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 *rawAvailable)(void* env_raw, int *available_len),
 int (__cdecl *rawReadClose)(void* env_raw, BOOL do_not_close_raw_in),
 int (__cdecl *rawWriteByte)(void* env_raw, byte value),
 int (__cdecl *rawWriteBArray)(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, rawReadBArray, rawAvailable,
 rawReadClose, rawGetLastErrorMessage), TRUE);
 uslw->raw_out = new CJDataOutputStream(new OZFuncOutputStream(env_raw,
 rawWriteByte, rawWriteBArray, rawFlush,
 rawWriteClose, rawGetLastErrorMessage), TRUE);
}

```

```

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 - (getUSLServerParam) USL 가 . 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)

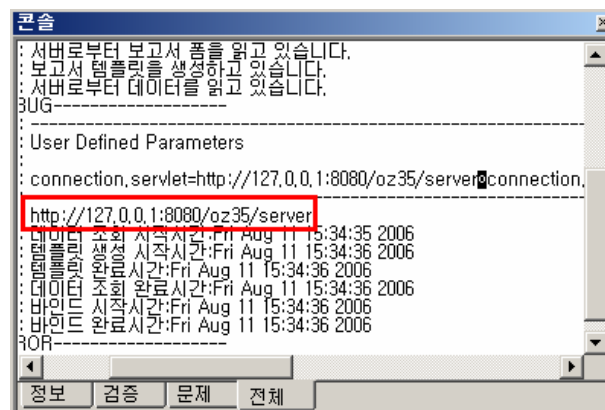
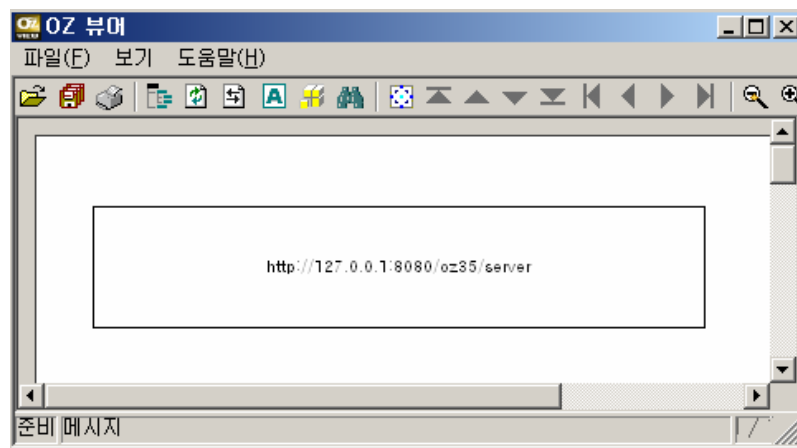
```
- [] [] () 'getUSLServerParam.ozr' . 'D:\Repository' .
 - html .

```
<HTML>
<BODY>
<OBJECT id = "ozviewer" CLASSID="CLSID:0DEF32F8-170F-46f8-B1FF-
4BF7443F5F25" width="100%" height="100%">

<param name="connection.servlet" value="http://127.0.0.1:8080/oz40/server">
<param name="connection.reportname" value="/getUSLServerParam.ozr">
<param name="viewer.isframe" value="true">
<param name="information.debug" value="true">

</OBJECT>
</body>
</HTML>
```

- caption USL



USL 2 - PKI

USL

PKI

■ USL

PKI

Output Stream

write(), flush()
flush()

PKI

```
package ozusl;

import java.io.*;
import oz.cp.OZUSLServer;
import javax.servlet.http.HttpSession;

import java.security.*;
import java.security.spec.*;

import oz.cp.message.OzcmException;
import pki.lib.*; // PKI import

public class PKIUSLServer extends OZUSLServer
{
    protected String share_key_tag = null;
    protected String share_key_value = null;

    protected Secure _secu = null;
    protected EnvelopedMessage _se = null;

    public PKIUSLServer() {
    }

    class PKIUSLServerOutputStream extends OutputStream
    {
        ByteArrayOutputStream bout;
        DataOutputStream out_org = null;
        int b_length = 0;
    }
}
```

```

public PKIUSLServerOutputStream(HttpSession _http_session,
                                String _report_name,
                                DataOutputStream _out_org) throws OzcmException
{
    ////////////////////////////////////////////////// Session ///////////////////////////////////
    share_key_value = (String)_http_session.getValue(share_key_tag);
    if (share_key_value==null)
        throw new OzcmException("session was not found");
    //////////////////////////////////////////////////
    bout = new ByteArrayOutputStream();
    out_org = _out_org;
}

public void write(int b) throws IOException {
    bout.write(b);
    b_length++;
}

public void close() throws IOException {
}

public void flush() throws IOException {
    try{
        //////////////////////////////////////////////////
        _secu = Secure.getInstance();
        _se = new EnvelopedMessage( _secu);
        String dummy=_se.encryptInit(secu.WEB, secu.KM, share_key_value);

        bout.flush();
        byte[] b_org = bout.toByteArray();

        byte[] encrypt_base64 = _se.encryptUpdate(b_org).getBytes();
        //////////////////////////////////////////////////
        out_org.writeInt(encrypt_base64.length);
        out_org.write(encrypt_base64, 0, encrypt_base64.length);
        out_org.flush();
    } catch(Exception e){
        e.printStackTrace();
        throw new IOException("PKI: error... " + e.toString());
    }
}

public InputStream createSecureInputStream(DataInputStream org_in) throws
OzcmException {
    try{

```



```

/////////////////////////////////////////////////////////////////

CUSLClient::CUSLClient()
{
    m_str_ip = _T("");
    m_str_url = _T("");
    m_str_pub_key_path = _T("");
    m_parameters = NULL;
}

CUSLClient::~CUSLClient()
{
}

void CUSLClient::SetServerIP(CString str_ip)
{
    m_str_ip = str_ip;
}

void CUSLClient::SetServerURL(CString str_url)
{
    m_str_url = str_url;
}

void CUSLClient::SetParameter(Parameter * parameters)
{
    m_parameters = parameters;
}

CJDataInputStream * CUSLClient::GetSecuredInputStream(CJDataInputStream *pIn)
{
    return new CJUSLDataInputStream(pIn, TRUE, m_str_pub_key_path);
}

CJDataOutputStream * CUSLClient::GetSecuredOutputStream(
CJDataOutputStream *pOut)
{
    CJUSLDataOutputStream out(pOut, FALSE, m_parameters);
    out.WriteSessionKeyToServer();
    m_str_pub_key_path = out.GetPublicKeyPath();
    return pOut;
}

<<JUSLDataInputStream.cpp>>
// JUSLDataInputStream.cpp: implementation of the CJUSLDataInputStream class.
//
/////////////////////////////////////////////////////////////////

#include "stdafx.h"

```

```
#include "JUSLDataInputStream.h"
#include <ozuslclient/axmofe.h>
#include <ozuslclient/base64.h>

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif _DEBUG

/////////////////////////////////////////////////////////////////
// Construction/Destruction
/////////////////////////////////////////////////////////////////

CJUSLDataInputStream::CJUSLDataInputStream(CJDataInputStream *pIn,
BOOL _isShouldDelete,
CString str_pub_key) : CJDataInputStream(pIn, _isShouldDelete)
{
    try{
        Jint jsize = pIn->readInt();
        //byte* b_org = new byte[size];

        CJArray<Jbyte>* arrayBuffer;
        char * buf = new char[2048];
        int readSize = -1;
        CJByteArrayOutputStream bout;
        try{
            while(true) {
                readSize = pIn->read(buf,0,2048);
                if (readSize ==0) break;
                bout.write(buf,0,readSize);
            }
            arrayBuffer = bout.toByteArrayForSign();
            in->close();
            delete buf;
        }catch(CZException *eof){
            throw eof;
        }
        IAxMOFE_NCA pki;
        if(!pki.CreateDispatch(_T("AxMOFE.AxMOFE_NCA.1"), NULL))
        {
        }
        char * tmp = arrayBuffer->getBuffer();
        int ssize = arrayBuffer->length();

        CString decrypt_base64 =
            pki.SymmDecryptDataEx(tmp, pki.ReadDataFile(str_pub_key), _T("SEED"));
```

```
byte* tmpByte2 = new byte[decrypt_base64.GetLength()];
int dsie = decrypt_base64.GetLength();

int size = 0;

CTP_Base64Decode((unsigned char*)
    decrypt_base64.GetBuffer(decrypt_base64.GetLength()), tmpByte2, &size);

m_pIn =
    new CJDataInputStream(new CJByteArrayInputStream((char*)tmpByte2,0,size,
        TRUE), TRUE);
in = m_pIn;

delete arrayBuffer;
}catch(CZException * ex){
    throw ex;
}
}

CJUSLDataInputStream::~CJUSLDataInputStream()
{
/*
    if(m_pIn)
    {
        m_pIn->close();
        delete m_pIn;
    }
*/
}

void CJUSLDataInputStream::GetBytes (const CString str, byte* b, int len)
{
    for (int i = 0 ; i < len ; i++) {
        b[i] =str.GetAt (i);
    }
}

Jint CJUSLDataInputStream::available()
{
    return m_pIn->available();
}

Jbyte CJUSLDataInputStream::readByte()//throw (CJIOException *)
{
    return m_pIn->readByte();
}
```

```
Jlong CJUSLDataInputStream::readLong()//throw (CJIOException *)
{
    return m_pIn->readLong();
}

Jdouble CJUSLDataInputStream::readDouble()//throw (CJIOException *)
{
    return m_pIn->readDouble();
}

Jfloat CJUSLDataInputStream::readFloat()//throw (CJIOException *)
{
    return m_pIn->readFloat();
}

Jboolean CJUSLDataInputStream::readBoolean()//throw (CJIOException *)
{
    return m_pIn->readBoolean();
}

Jshort CJUSLDataInputStream::readShort()//throw (CJIOException *)
{
    return m_pIn->readShort();
}

Jchar CJUSLDataInputStream::readChar()//throw (CJIOException *)
{
    return m_pIn->readChar();
}

Jint CJUSLDataInputStream::readInt()//throw (CJIOException *)
{
    return m_pIn->readInt();
}

CString CJUSLDataInputStream::readUTF()//throw (CJIOException *)
{
    return m_pIn->readUTF();
}

CString CJUSLDataInputStream::readString()//throw (CJIOException *)
{
    return m_pIn->readString();
}

Jint CJUSLDataInputStream::read(CJArray<Jbyte> &b)//throw (CJIOException *)
{
    return m_pIn->read(b);
}
```

```

}

Jint CJUSLDataInputStream::read(CJArray<Jbyte> &b, Jint off, Jint len)
//throw (CJIOException *)
{
    return m_pIn->read(b, off, len);
}

int CJUSLDataInputStream::getRemainSize()
{
    return m_pIn->getRemainSize();
}

Jbyte * CJUSLDataInputStream::getBufferPointer()
{
    return m_pIn->getBufferPointer();
}

<<JUSLDataOutputStream.cpp>>
// JUSLDataOutputStream.cpp: implementation of the CJUSLDataOutputStream class.
////////////////////////////////////

#include "stdafx.h"
#include "JUSLDataOutputStream.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif _DEBUG

////////////////////////////////////
// Construction/Destruction
////////////////////////////////////
CJUSLDataOutputStream::CJUSLDataOutputStream(CJDataOutputStream *pOut,
BOOL _isShouldDelete,
Parameter *parameter) : CJDataOutputStream(pOut, _isShouldDelete)
{
    m_pOut = pOut;
    m_parameter = parameter;
    m_str_pub_key_path = _T("");
}

CJUSLDataOutputStream::~CJUSLDataOutputStream()
{
}

void CJUSLDataOutputStream::WriteSessionKeyToServer()

```

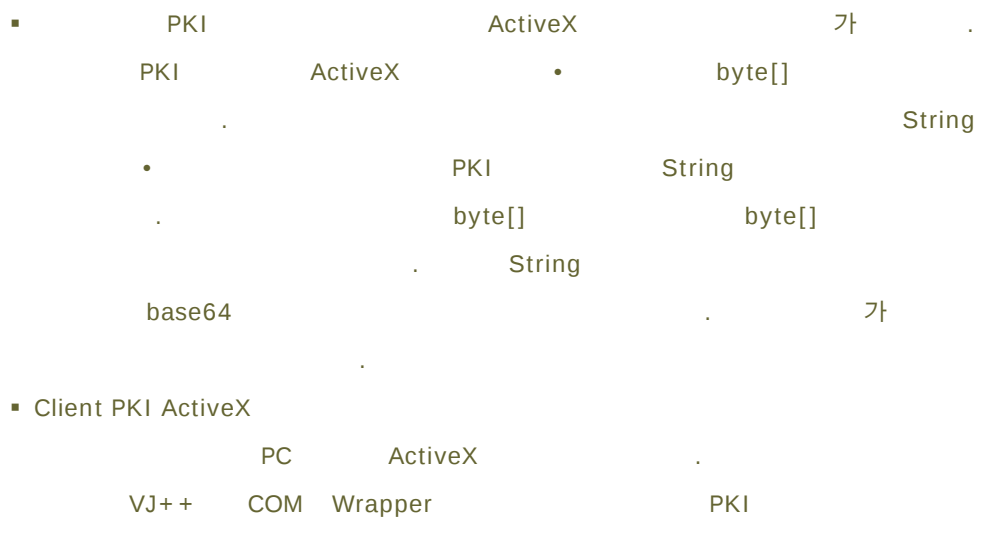
```

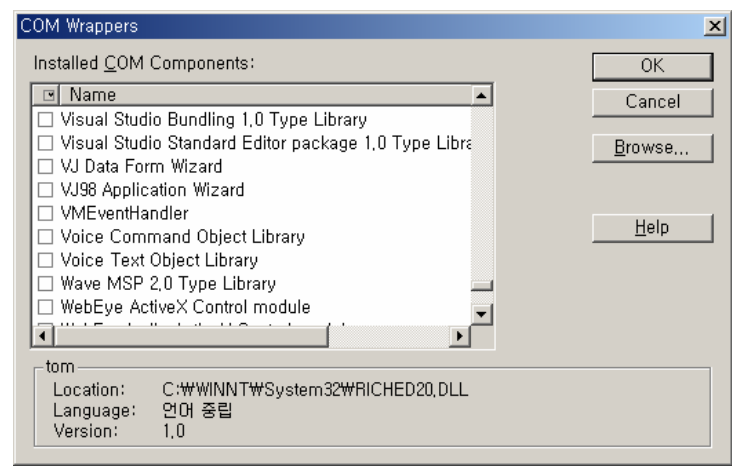
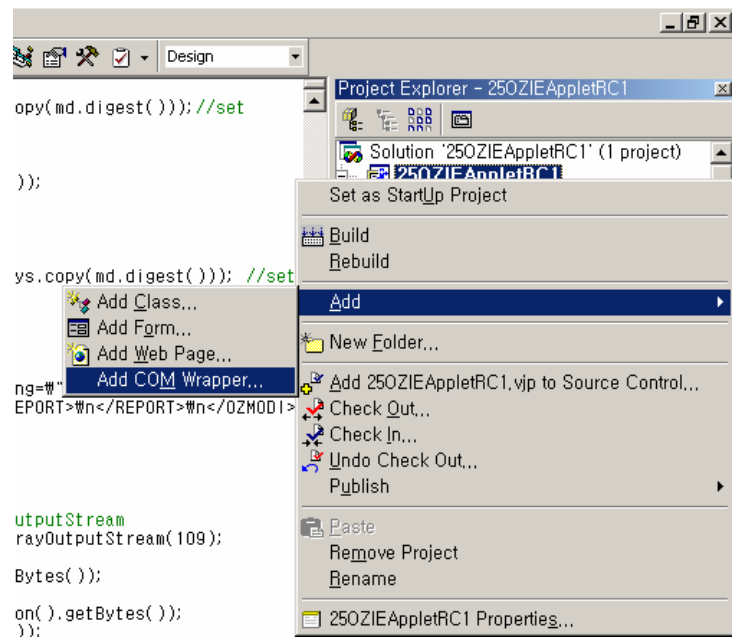
{
    try{
        BOOL b = TRUE;
        for (int i=0; i<m_parameter->GetSize(); i++) {
            // Read session key from viewer's parameter to write this information to
            //
            if(m_parameter->ElementAt(i).name.CompareNoCase(_T("session_name")) == 0){
                m_pOut->writeUTF(m_parameter->ElementAt(i).value);
                b=FALSE;
            }
            //Read ksign's key path to decode the encoded data from
            if (m_parameter->ElementAt(i).name.CompareNoCase(_T("ksign_path")) == 0)
            {
                m_str_pub_key_path = m_parameter->ElementAt(i).value;
            }
        }
    }catch(CZException * e){
        throw e;
    }
}

CString CJUSLDataOutputStream::GetPublicKeyPath()
{
    return m_str_pub_key_path;
}
}

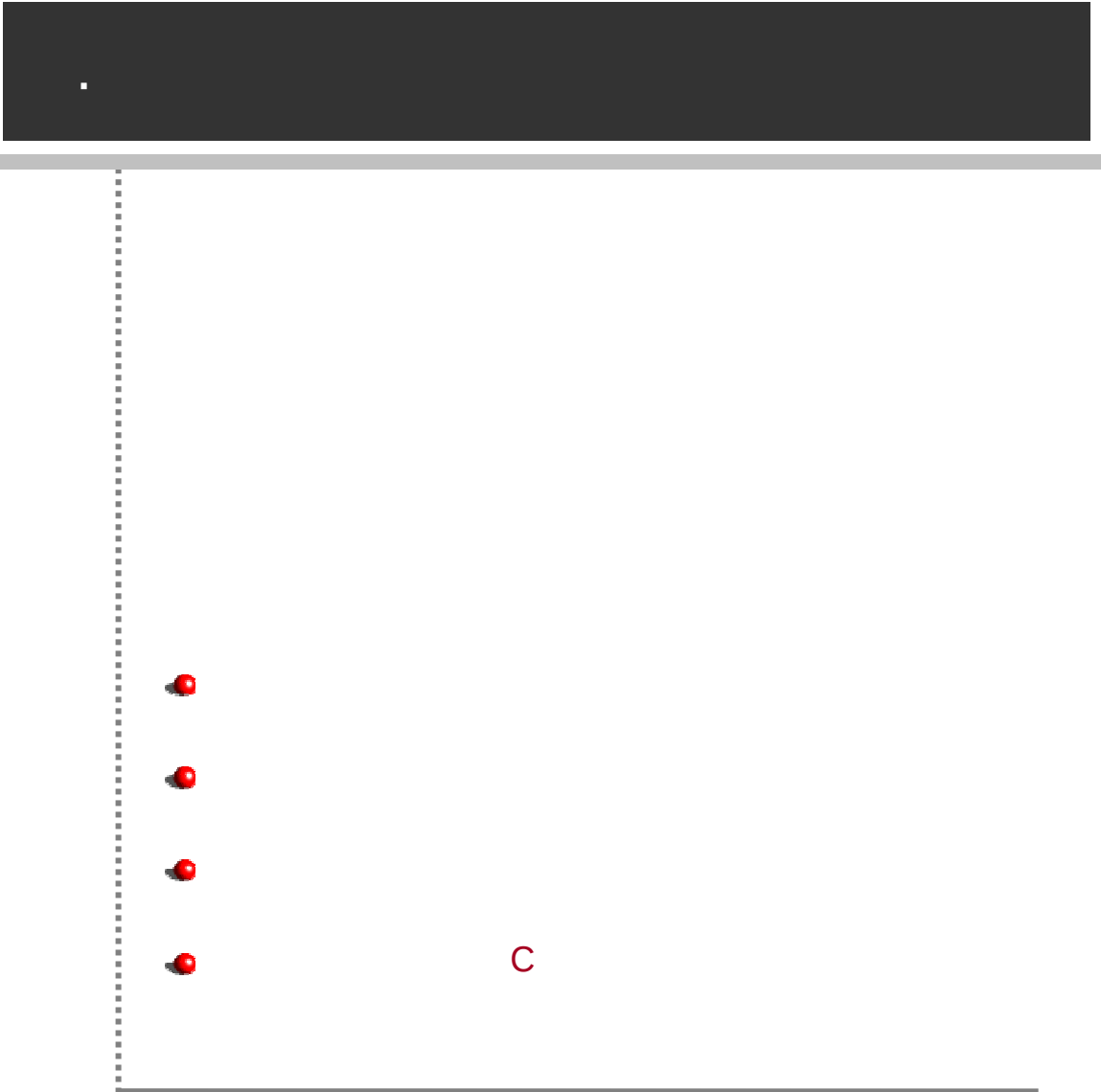
```

PKI

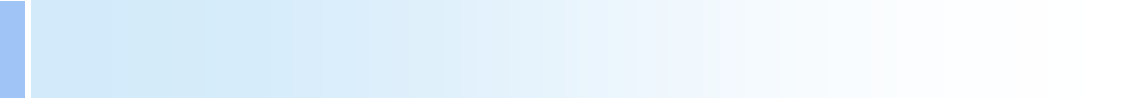




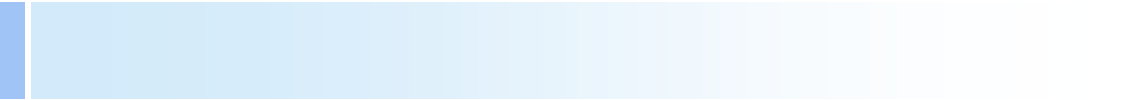
PKI Import USL



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
    .

```

■ interface oz.ucomp.OZUserComp

- `oz.ucomp.OZUserComp` : `oz.ucomp.OZUserCompMeta`의 메타데이터를 관리하는 인터페이스.
- Method
 - `public void setMeta(OZUserCompMeta meta)`
 `meta` : `OZUserCompMeta` 객체
 - `public void paint(oz.client.IGraphics g, oz.client.OZPoint basePoint, oz.client.shape.ILabel label, String data)`
 `g` : `IGraphics` 객체
 `basePoint` : `OZPoint` 객체
 `label` : `ILabel` 객체
 `data` : Data
 `caption` : Caption
 - `public void paintEX(oz.client.IGraphics g, oz.client.OZPoint basePoint, oz.client.shape.ILabel label, String data, int width, int height)`
 `g` : `IGraphics` 객체
 `basePoint` : `OZPoint` 객체
 `label` : `ILabel` 객체
 `data` : Data
 `caption` : Caption
 `width` : `width`
 `height` : `height`
 `paintEX` : `paintEX` 메소드
 - `public oz.client.OZDimension getAutosize(oz.client.shape.ILabel label, String data)`
 `label` : `ILabel` 객체
 `data` : Data
 `getAutosize` : `getAutosize` 메소드

■ interface oz.ucomp.OZUserCompMeta

- `oz.ucomp.OZUserCompMeta` : `oz.ucomp.OZUserComp`의 메타데이터를 관리하는 인터페이스.
- Method
 - `public void xmlBind(java.util.Hashtable t)`
 `t` : `Hashtable` 객체
 - `public void setAttr(String s, String v)`
 `s` : `String` 객체
 `v` : `String` 객체
 `setAttr` : `setAttr` 메소드
 - `public void writeMe(oz.framework.cp.io.OZDataOutputStream out)`
 `out` : `OZDataOutputStream` 객체
 `writeMe` : `writeMe` 메소드

- public void readMe(oz.framework.cp.io.OZDataInputStream in)

in (Serialize)

가 ,

가 .

■ interface oz.ucomp.OZUserCompMetaBeanInfo

- : .

- Method

- public PropertyDescriptor[] getPropertyys()

PropertyDescriptor

Class

■ oz.client.OZPoint, oz.client.OZDimension

java.awt.Point java.awt.Dimension float

■ oz.framework.cp.io.OZDataOutputStream, oz.framework.cp.io.OZDataInputStream

java.io.DataOutputStream java.io.OZDataInputStream

■ oz.client.shape.ILabel

■ oz.client.IGraphics

java.awt.Graphics Wrapping

PDF417 2D

■ **PDF417.java**

```
package ucomp;
import oz.ucomp.*;

public class PDF417 implements OZUserComp{
    protected PDF417Meta meta = null;
    public void setMeta(OZUserCompMeta meta){
        // PDF417Meta setting
        this.meta = (PDF417Meta)meta;
    }
    public void paint(oz.client.IGraphics g, oz.client.OZPoint basePoint,
        oz.client.shape.ILabel label, String data)throws OZUserCompException{
        // fill black rectangle
        g.setOZColor(oz.client.OZColor.black);
        g.fillRect(basePoint.x, basePoint.y, label.getWidth(), label.getHeight());
    }
    public oz.client.OZDimension getAutosize(oz.client.shape.ILabel label,
        String data) throws OZUserCompException{
        // no effect
        return new oz.client.OZDimension(label.getWidth(), label.getHeight());
    }
}
```

■ **PDF417Meta.java**

```
package ucomp;

import oz.ucomp.*;

public class PDF417Meta extends OZUserCompMeta{
    //propertyNames is text for designer property name
    public final static String[] propertyNames =
        new String[]{"", "X", "Y", "d11", "d11 URL",
            "d11 "};
    public PDF417Meta() {
    }
    public void setAttr(String s, String v)throws OZUserCompException{
        //attribute setting function
        // s is attribute name
        // v is attribute value string
        if(s.equalsIgnoreCase("PDF417_ROWNUM")){
            setRowNum( Integer.parseInt(v) );
        }else if(s.equalsIgnoreCase("PDF417_COLNUM")){
            setColNum( Integer.parseInt(v) );
        }else if(s.equalsIgnoreCase("PDF417_ECC")){
            setECC( Integer.parseInt(v) );
        }else if(s.equalsIgnoreCase("PDF417_XSCALE")){
            setXScale( Integer.parseInt(v) );
        }
    }
}
```

```
}else if(s.equalsIgnoreCase("PDF417_YSCALE")){
    setYScale( Integer.parseInt(v) );
}else if(s.equalsIgnoreCase("PDF417_DLLNAME")){
    setDllName( v );
}else if(s.equalsIgnoreCase("PDF417_DLLURL")){
    setDllURL( v );
}else if(s.equalsIgnoreCase("PDF417_DLLSIZE")){
    setDllSize( Integer.parseInt(v) );
}
}

public void xmlBind(java.util.Hashtable t)throws OZUserCompException{
    // attribute save function
    // Hashtable key is attribute name
    // Hashtable value is attribute value string
    t.put("PDF417_ROWNUM", String.valueOf(getRowNum()));
    t.put("PDF417_COLNUM", String.valueOf(getColNum()));
    t.put("PDF417_ECC", String.valueOf(getECC()));
    t.put("PDF417_XSCALE", String.valueOf(getXScale()));
    t.put("PDF417_YSCALE", String.valueOf(getYScale()));
    t.put("PDF417_DLLNAME", getDllName());
    t.put("PDF417_DLLURL", getDllURL());
    t.put("PDF417_DLLSIZE", String.valueOf(getDllSize()));
}

public void writeMe(oz.framework.cp.io.OZDataOutputStream out) throws
Exception{
    //serialize write function
    out.writeInt(rowNum);
    out.writeInt(colNum);
    out.writeInt(ecc);
    out.writeInt(xScale);
    out.writeInt(yScale);
    out.writeInt(dllSize);
    out.writeUTF2(dllName);
    out.writeUTF2(dllURL);
}

public void readMe(oz.framework.cp.io.OZDataInputStream in)throws Exception{
    //serialize read function
    rowNum = in.readInt();
    colNum = in.readInt();
    ecc = in.readInt();
    xScale = in.readInt();
    yScale = in.readInt();
    dllSize = in.readInt();
    dllName = in.readUTF2();
    dllURL = in.readUTF2();
}

int rowNum = 0;
int colNum = 8;
```

```
int ecc = 0;
int xScale = 1;
int yScale = 1;
String dllName = "";
String dllURL = "";
int dllSize = 67000;
public int getRowNum(){
    return rowNum;
}
public int getColNum(){
    return colNum;
}
public int getECC(){
    return ecc;
}
public int getXScale(){
    return xScale;
}
public int getYScale(){
    return yScale;
}
public String getDllName(){
    return dllName;
}
public String getDllURL(){
    return dllURL;
}
public int getDllSize(){
    return dllSize;
}
//firePropertyChangeListener is fuction for attribute change event
public void setRowNum(int i){
    if(i < 0)return;
    int oldValue = getRowNum();
    rowNum = i;
    firePropertyChangeListener(propertyNames[0], new Integer(oldValue),
                                new Integer(i));
}
public void setColNum(int i){
    if(i < 0)return;
    int oldValue = getColNum();
    colNum = i;
    firePropertyChangeListener(propertyNames[1], new Integer(oldValue),
                                new Integer(i));
}
public void setECC(int i){
    if(i < 0)return;
    int oldValue = getECC();
```

```
        ecc = i;
        firePropertyChangeListener(propertyNames[2], new Integer(oldValue),
                                   new Integer(i));
    }
    public void setXScale(int i){
        if(i < 1)return;
        int oldValue = getXScale();
        xScale = i;
        firePropertyChangeListener(propertyNames[3], new Integer(oldValue),
                                   new Integer(i));
    }
    public void setYScale(int i){
        if(i < 1)return;
        int oldValue = getYScale();
        yScale = i;
        firePropertyChangeListener(propertyNames[4], new Integer(oldValue),
                                   new Integer(i));
    }
    public void setDllName(String i){
        String oldValue = getDllName();
        dllName = i;
        firePropertyChangeListener(propertyNames[5], oldValue, i);
    }
    public void setDllURL(String i){
        String oldValue = getDllURL();
        dllURL = i;
        firePropertyChangeListener(propertyNames[6], oldValue, i);
    }
    public void setDllSize(int i){
        if(i < 1)return;
        int oldValue = getDllSize();
        dllSize = i;
        firePropertyChangeListener(propertyNames[7], new Integer(oldValue),
                                   new Integer(i));
    }
}
```

■ PDF417MetaBeanInfo.java

```
package ucomp;

import java.beans.*;
import oz.ucomp.OZUserCompMetaBeanInfo;

public class PDF417MetaBeanInfo extends OZUserCompMetaBeanInfo{

    private static Class beanClass = PDF417Meta.class;
    private static PropertyDescriptor[] rv = null;
```

```

private static String error = null;
public PropertyDescriptor[] getPropertyDescriptors() {
    if(rv == null) throw new Error(beanClass.getName() + " : " + error);
    return rv;
}
static{
    try
    {
////////////////////////////////////
//PDF417Meta.propertyNames[0] : Attribute Text
//"getRowNum": Attribute get fuction in PDF417Meta
//"setRowNum": Attribute set fuction in PDF417Meta

//setShortDescription have to start "cb"
////////////////////////////////////

        PropertyDescriptor rowNum = new PropertyDescriptor
        (PDF417Meta.propertyNames[0], beanClass, "getRowNum", "setRowNum");
        rowNum.setShortDescription("cba");
        rowNum.setConstrained(true);

        PropertyDescriptor colNum = new PropertyDescriptor
        (PDF417Meta.propertyNames[1], beanClass, "getColNum", "setColNum");
        colNum.setShortDescription("cbb");
        colNum.setConstrained(true);

        PropertyDescriptor ecc = new PropertyDescriptor
        (PDF417Meta.propertyNames[2], beanClass, "getECC", "setECC");
        ecc.setShortDescription("cbc");
        ecc.setConstrained(true);

        PropertyDescriptor xscale = new PropertyDescriptor
        (PDF417Meta.propertyNames[3], beanClass, "getXScale", "setXScale");
        xscale.setShortDescription("cbd");
        xscale.setConstrained(true);

        PropertyDescriptor yscale = new PropertyDescriptor
        (PDF417Meta.propertyNames[4], beanClass, "getYScale", "setYScale");
        yscale.setShortDescription("cbe");
        yscale.setConstrained(true);

        PropertyDescriptor dllName = new PropertyDescriptor
        (PDF417Meta.propertyNames[5], beanClass, "getDllName", "setDllName");
        dllName.setShortDescription("cbf");
        dllName.setConstrained(true);

        PropertyDescriptor dllUrl = new PropertyDescriptor
        (PDF417Meta.propertyNames[6], beanClass, "getDllURL", "setDllURL");

```

```
dllUrl.setShortDescription("cbg");
dllUrl.setConstrained(true);

PropertyDescriptor dllSize = new PropertyDescriptor
(PDF417Meta.propertyNames[7], beanClass, "getDllSize", "setDllSize");
dllSize.setShortDescription("cbh");
dllSize.setConstrained(true);

rv = new PropertyDescriptor[]{
    rowNum, colNum, ecc, xscale, yscale, dllName, dllUrl, dllSize};
} catch(IntrospectionException e){
    rv = null;
    error = e.toString();
}
}
```

C



- GetNewInstance

Prototype	DWORD GetNewInstance()
Definition	.
Argument	
Return	, 0(NULL)

- GetCopyInstance

Prototype	DWORD GetCopyInstance(DWORD src)
Definition	.
Argument	<i>src</i>
Return	, 0(NULL)

- DeleteInstance

Prototype	void DeleteInstance(DWORD src)
Definition	.
Argument	<i>src</i>
Return	



Serialize

- getAttrListLength

Prototype	int getAttrListLength(DWORD src)
Definition	가 .
Argument	<i>src</i>

Return, -1

- getAttrList

Prototype	BOOL getAttrList(DWORD src, char** attrs, const int length)		
Definition	. attrs		
Argument	src		
	attrs	Pointer	
	length	getAttrListLength	
	Return	true, false	

- getAttrLength

Prototype	int getAttrLength(DWORD src, const char* name)		
Definition	.		
Argument	src		
	name		
Return	, () -1		

- getAttr

Prototype	BOOL getAttr(DWORD src, const char* name, char* value, const int value_length)		
Definition	name value .		
Argument	src		
	name		
	value		
	value_length	getAttrLength	
Return	true, () false		

- setAttr

Prototype	BOOL setAttr(DWORD src, const char* name, const char* value)		
Definition	.		
Argument	src		

	<i>name</i>
	<i>value</i>
Return	, () -1

- writeMe

Prototype	char* writeMe(DWORD src, int * length)
Definition	serialize .
	<i>src</i>
Argument	<i>length</i> serialize binary data int pointer
Return	serialize binary data pointer , NULL

- readMe

Prototype	void readMe(DWORD src, const char* pData, const int length)
Definition	serialize binary data .
	<i>src</i>
Argument	<i>pData</i> binary data <i>length</i>
Return	



- paint

Prototype	void paint(DWORD src, HDC hDC, LPCTSTR data, const float x, const float y, const float w, const float h, const float scale)
Definition	.
Argument	<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>

	<i>scale</i>	
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	
	<i>src</i>	
	<i>hDC</i>	(WIN32 API)
	<i>data</i>	Caption
Argument	<i>x</i>	x
	<i>y</i>	y
	<i>w</i>	
	<i>h</i>	
	<i>scale</i>	
	<i>x_offset</i>	가 x offset
	<i>y_offset</i>	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	<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>	

	<i>scale</i>	
	<i>x_offset</i>	x Offset
	<i>y_offset</i>	x Offset
Return		

- getAutosize

Prototype	void getAutosize(DWORD src, HDC hDC, LPCTSTR data, float * w, float * h)	
Definition	가 ,	
	<i>src</i>	
	<i>hDC</i>	(WIN32 API)
Argument	<i>data</i>	Caption
	<i>w</i>	pointer
	<i>h</i>	pointer
Return	w, h	가 ,

C++ dll PDF 417 .

■ PDF417.h

```
// hanja.h : main header file for the HANJA DLL
//

#ifdef 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(CPdf417AppaApp)
   //}}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>

////////////////////
//PDF417 Barcode api include
#include "pdfapi.h"
////////////////////
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
```

```

#endif
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////
//PDF417 Barcode api implement
static PDFObject g_MacroObjIn, g_MacroObjOut;

PDFSZET Encode(LPCTSTR dataStr, unsigned short rowNum, unsigned short colNum,
unsigned short ecc, UINT xScale, UINT yScale)
{
    PDFSZET nInputLen = 0;
    char szAspect[ 15 ];
    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 );
    }
    sprintf( szAspect, "%d:%d", xScale+yScale-1, xScale );
    PDFSetRowHeight( szAspect );

    MPDFDisable();
    nInputLen = (PDFSZET)lstrlen( dataStr );
    MakeMemoryObject( &g_MacroObjIn, PDFINPUT, (LPSTR)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(LPCTSTR dataStr, BYTE * rt, unsigned short rowNum,
unsigned short colNum, unsigned short ecc, UINT xScale, UINT yScale)
{
    PDFSZET nInputLen = Encode(dataStr, rowNum, colNum, ecc, xScale, yScale);
    BOOL fStatus;
    PDFSZET 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;
}
LPPICTURE EncodePrint(LPCTSTR dataStr, unsigned short rowNum,
unsigned short colNum, unsigned short ecc, UINT xScale, UINT yScale)
{
PDFSIZET nInputLen = Encode(dataStr, rowNum, colNum, ecc, xScale, yScale);
    BOOL fStatus;
    PDFSIZET 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;
}
////////////////////////////////////

//
//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;

/////////////////////////////////////////////////////////////////
// The one and only OZPdf417Comp object
class OZPdf417Comp{
public:
/////////////////////////////////////////////////////////////////
//constructor
OZPdf417Comp();
```

```

////////////////////////////////////
////////////////////////////////////
//copy constructor
OZPdf417Comp(OZPdf417Comp & cp);
////////////////////////////////////
////////////////////////////////////
//destructor
virtual ~OZPdf417Comp();
////////////////////////////////////

////////////////////////////////////
//function for Attributes
int getAttrListLength();
BOOL getAttrList(TCHAR** attrs, const int length);
int getAttrLength(CString name);
BOOL getAttr(CString name, TCHAR* value, const int value_length);
BOOL setAttr(CString name, CString value);
char* writeMe(int * length);
void readMe(const char* pData, const int length);
////////////////////////////////////
////////////////////////////////////
//function for paint
void paint(HDC hDC, CString data, const float x, const float y, const float w,
const float h, const float scale);
////////////////////////////////////
////////////////////////////////////
//function for print
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);
////////////////////////////////////
////////////////////////////////////
//function for autosize
void getAutosize(HDC hDC, CString data, float * w, float * h);
////////////////////////////////////
private:
////////////////////////////////////
//Attributes
int rowNum;
int colNum;
int ecc;
int xScale;
int yScale;
////////////////////////////////////
};

////////////////////////////////////
// c api for oz viewer

```

```
#ifdef __cplusplus
extern "C"
{
#endif

__declspec( dllexport ) DWORD __cdecl GetNewInstance()
{
    return (DWORD)(void*)new OZPdf417Comp();
}
__declspec( dllexport ) DWORD __cdecl GetCopyInstance(DWORD src)
{
    return (DWORD)(void*)new OZPdf417Comp(*((OZPdf417Comp *) (void *)src));
}
__declspec( dllexport ) void __cdecl DeleteInstance(DWORD src)
{
    delete ((OZPdf417Comp *) (void *)src);
}
/////////////////////////////////////////////////////////////////
__declspec( dllexport ) int __cdecl getAttrListLength(DWORD src)
{
    return ((OZPdf417Comp *) (void *)src)->getAttrListLength();
}
__declspec( dllexport ) BOOL __cdecl getAttrList(DWORD src, TCHAR** attrs,
const int length){
    return ((OZPdf417Comp *) (void *)src)->getAttrList(attrs, length);
}
__declspec( dllexport ) int __cdecl getAttrLength(DWORD src, const TCHAR* name)
{
    return ((OZPdf417Comp *) (void *)src)->getAttrLength(name);
}
__declspec( dllexport ) BOOL __cdecl getAttr(DWORD src, const TCHAR* name,
TCHAR* value, const int value_length){
    return ((OZPdf417Comp *) (void *)src)->getAttr(name, value, value_length);
}
__declspec( dllexport ) BOOL __cdecl setAttr(DWORD src, const TCHAR* name,
const TCHAR* value)
{
    return ((OZPdf417Comp *) (void *)src)->setAttr(name, value);
}
__declspec( dllexport ) char* __cdecl writeMe(DWORD src, int * length)
{
    return ((OZPdf417Comp *) (void *)src)->writeMe(length);
}
__declspec( dllexport ) void __cdecl readMe(DWORD src, const char* pData,
const int length)
{
    ((OZPdf417Comp *) (void *)src)->readMe(pData, length);
}
}
```

```

__declspec( dllexport ) void __cdecl paint(DWORD src, HDC hDC, LPCTSTR 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( dllexport ) void __cdecl 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)
{
    ((OZPdf417Comp *)(void *)src)->print(hDC, data, x, y, w, h, scale, x_offset,
y_offset);
}
__declspec( dllexport ) void __cdecl getAutosize(DWORD src, HDC hDC,
LPCTSTR data, float * w, float * h)
{
    ((OZPdf417Comp *)(void *)src)->getAutosize(hDC, data, w, h);
}
#ifdef __cplusplus
}
#endif
/////////////////////////////////////////////////////////////////
// int to string convert function
CString _toString(int i){
    CString rst;
    rst.Format(_T("%d"), i);
    return rst;
}

/////////////////////////////////////////////////////////////////
// OZPdf417Comp implement

/////////////////////////////////////////////////////////////////
//constructor
OZPdf417Comp::OZPdf417Comp(){
    rowNum = 0;
    colNum = 8;
    ecc = 0;
    xScale = 1;
    yScale = 1;
}
/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////
//copy constructor
OZPdf417Comp::OZPdf417Comp(OZPdf417Comp & comp){
    rowNum = comp.rowNum;
    colNum = comp.colNum;
    ecc = comp.ecc;
    xScale = comp.xScale;

```

```
        yScale = comp.yScale;
    }
    ///////////////////////////////////
    ///////////////////////////////////
    //destructor
    OZPdf417Comp::~OZPdf417Comp(){
    }
    ///////////////////////////////////

    int OZPdf417Comp::getAttrListLength(){
    // Attribute count return
        return 5;
    }

    BOOL OZPdf417Comp::getAttrList(TCHAR** attrs, const int length){
    // Attribute names set to attrs
    // length is result of getAttrListLength() function call
        if(length < 0)return FALSE;
        int nIndex = 0;
        if(length == nIndex)return TRUE;
        attrs[nIndex++] = _T("PDF417_ROWNUM");
        if(length == nIndex)return TRUE;
        attrs[nIndex++] = _T("PDF417_COLNUM");
        if(length == nIndex)return TRUE;
        attrs[nIndex++] = _T("PDF417_ECC");
        if(length == nIndex)return TRUE;
        attrs[nIndex++] = _T("PDF417_XSCALE");
        if(length == nIndex)return TRUE;
        attrs[nIndex++] = _T("PDF417_YSCALE");
        return TRUE;
    }

    int OZPdf417Comp::getAttrLength(CString name){
        // name is Attribute name
        // return TCHAR length of value by name
        CString value;
        if(name == "PDF417_ROWNUM"){
            value = _toString(rowNum);
        }else if(name == "PDF417_COLNUM"){
            value = _toString(colNum);
        }else if(name == "PDF417_ECC"){
            value = _toString(ecc);
        }else if(name == "PDF417_XSCALE"){
            value = _toString(xScale);
        }else if(name == "PDF417_YSCALE"){
            value = _toString(yScale);
        }else{
            ///////////////////////////////////
            return -1;
        }
    }
```

```

    }
    return value.GetLength()+1;
}

BOOL OZPdf417Comp::getAttr(CString name, TCHAR* valueBuffer,
const int value_length){
// name is Attribute name
// Attribute value set to valueBuffer
// value_length is result of getAttrLength() function call
    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 can't find attribute
        return FALSE;
    }
    if(value.GetLength() >= value_length-1){
        // return false if length is differ
        return FALSE;
    }
    _tcscpy(valueBuffer, value);
    return TRUE;
}

BOOL OZPdf417Comp::setAttr(CString name, CString value){
// name is Attribute name
// value is Attribute 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 if can't find attribute
        return FALSE;
    }
}

```

```
        return TRUE;
    }
    char* OZPdf417Comp::writeMe(int * length){
        // serialize functin
        // binary length set to length
        // return my binary
        *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){
        // serialize functin
        // pData is result of writeMe() function call
        // length is result of writeMe() function call
        // length is binary length
        if(length != 20){
            throw new CException(0);
        }
        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, CString data, const float x, const float y,
const float w, const float h, const float scale){
// hDC is device context handle
// data is component caption
// x and y are component pixel positon
// w and h are component pixel size
// scale is extension ratio. (1.0f 1.5f 2.0f etc...)
//((int)(x*scale) and (int)(y*scale) are real position in HDC

    LPPICTURE gpPicture =
        EncodePrint(data, rowNum,colNum,ecc,xScale,yScale);//
    if(gpPicture == NULL) return;
    long hmWidth;
    long hmHeight;
    gpPicture->get_Width(&hmWidth);
    gpPicture->get_Height(&hmHeight);
    // convert himetric 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, 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){
// hDC is device context handle
// data is component caption
// x and y are component pixel positon
// w and h are component pixel size
// scale is extension ratio.

```

```
// x_offset and y_offset are print offset
//(int)(x*scale+x_offset) and (int)(y*scale+y_offset) are real position in HDC

LPPICTURE gpPicture =
    EncodePrint(data, rowNum,colNum,ecc,xScale,yScale);//
if(gpPicture == NULL) return;
long hmWidth;
long hmHeight;
gpPicture->get_Width(&hmWidth);
gpPicture->get_Height(&hmHeight);
// convert himetric 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 hesult = 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){
// hDC is device context handle
// data is component caption
// *w and *h are component pixel size
// preferred size may set to w and h (*w = 100; *h = 150;)

LPPICTURE gpPicture =
    EncodePrint(data, rowNum,colNum,ecc,xScale,yScale);//
if(gpPicture == NULL) return;
long hmWidth;
long hmHeight;
gpPicture->get_Width(&hmWidth);
gpPicture->get_Height(&hmHeight);
// convert himetric 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

Scheduler API	makePDF	export	ASP
SchedulerCom	makePDF	export	ASP

.

SchedulerCom

■ Com

- .Init

Prototype	.Init()
------------------	---------

Definition	COM
-------------------	-----

Argument	
-----------------	--

- .Clean

Prototype	.Clean()
------------------	----------

Definition	COM
-------------------	-----

Argument	
-----------------	--

■

- .SetServerType

Prototype	.SetServerType String ServerType
------------------	----------------------------------

Definition	가 "TCP" TCP
-------------------	-------------

Argument	ServerType "Servlet" 가 "Servlet" ()
-----------------	--------------------------------------

- .SetServerIP

Prototype	.SetServerIP String IP
------------------	------------------------

Definition	가 IP
-------------------	------

Argument	IP ex) "127.0.0.1"
-----------------	--------------------

- .SetServerPort

Prototype	.SetServerPort String Port		
Definition	가	Port	.
Argument	<i>Port</i>	Port	ex) "8003"

- .SetServerURL

Prototype	.SetServerURL String URL		
Definition	가	URL	.
Argument	<i>URL</i>	URL	ex) "http://localhost:7001/oz/server"



- .SetSchedulerIP

Prototype	.SetSchedulerIP String IP		
Definition		IP	.
Argument	<i>IP</i>	IP	ex) "127.0.0.1"

- .SetSchedulerPort

Prototype	.SetSchedulerPort String Port		
Definition		Port	.
Argument	<i>Port</i>	Port	ex) "9521"



- .SetUser

Prototype	.SetUser String UserID		
Definition			.
Argument	<i>UserID</i>		ex) "admin"

- .SetPassword

Prototype	.SetPassword String Password		
Definition			.
Argument	<i>Password</i>		ex) "admin"

■ Key

- .setProperty

Prototype	.setProperty String Key, String Value	
Definition	, .	
Argument	Key	Key
	Value	

: ".setProperty"

Key Value

Key	Value	
"launch_type"	"Immediately"	Scheduler Task "Immediately"
"report_name"	" "	(*.ozr)
"category_name"	"/<Category>"	
"export.confirmsave"	"false"	"false"
"parameter_count"	" "	ODI : "0" 가
"parameter_name_<index>"	" "	"[FORM]." ODI "ODI" .
"parameter_value_<index>"	" "	

: ".setProperty"

Key "launch_type",

"report_name", "category_name", "export.confirmsave",

"parameter_count"

- .SetExportProperty

Prototype .SetExportProperty String Key, String Value

Definition

Argument

<i>Key</i>	Key
<i>Value</i>	

: ".SetExportProperty" Key Value

Key	Value	
"connection.server"	" IP"	가 IP
"connection.port"	" Port"	가 Port
"connection.servlet"	" URL"	가 URL
"connection.reportName"	" "	
"connection.fetchtype"	"BATCH"	
"connection.pcount"	" "	
"connection.args<index> = "	" "	
"odi.odinames"	"ODI ,ODI ,..."	ODI ODI ODI (" ,")
"odi.ODI .pcount",	" "	ODI
"odi.ODI .args<index>"	" = "	

"export.format"	" "	"/" "ozd/html/jpg/xls/doc/svg/txt /ppt/tif/csv"
"<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

Prototype	.MakePDF(String ExportType)		
Definition	pdf	:	"ViewType"
		"None"	
		"SHOW"	PC PDF Reader
			pdf
Argument	ExportType	"ATTACH"	가 pdf
		"NONE"	pdf

- .Export

Prototype	.Export()		
	pdf, ozd, html, jpg, xls, doc, svg, txt, ppt, tif, csv		
Definition	:	"ViewType"	
		"None"	
Argument			

- .IsExportSucceeded

Prototype	.IsExportSucceeded()
------------------	----------------------

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

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.odiparam1"  
' .setProperty "parameter_value_3", "PARAM 3"  
' .setProperty "parameter_name_4", "parameter_test.odiparam2"  
' .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 .")  
End If  
>
```

```

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", "odiparam1=odi1"
        .setExportProperty "odi.parameter_test.args2", "odiparam2=odi2"
        .setExportProperty "odi.parameter_test.pcount", "2"
        .setExportProperty "odi.odinames", "parameter_test"

        .setExportProperty "export.format",
            "ozd/html/jpg/xls/doc/svg/txt/ppt/tif/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.tif"
        .setExportProperty "csv.filename", "test.csv"

        .setExportProperty "viewer.childcount", "1"
        .setExportProperty "child1.connection.server", "127.0.0.1"
        .setExportProperty "child1.connection.port", "8003"
    End With
End If

```

```
.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", "odiparam1=odi1"  
.setExportProperty "child1.odi.parameter_test.args2", "odiparam2=odi2"  
.setExportProperty "child1.odi.parameter_test.pcount", "2"  
.setExportProperty "child1.odi.odinames", "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.tiff.filename", "child_test.tif"  
.setExportProperty "child1.csv.filename", "child_test.csv"  
  
.Export()  
  
Response.Write(.IsExportSucceeded())  
  
.Clean()  
  
End With  
  
Set Com = Nothing  
  
End If  
%>
```


Appendix 2. Servlet API

Servlet API

Servlet API(ozsdmapi.jar)

Servlet API
가
가
URL
POST
ODI
DataAction
DataAction POST URL , DataAction
"ok" "success"
Servlet API ODI DB
(UDS) 가

POST

POST POST

■

POST	
_OZ_ODIFetchType_	Fetch DM_BATCH_FETCH DM_CONCURRENT_FETCH FetchUnit DM_PER_DATASET Fetch DM_CONCURRENT_FETCH
_OZ_ODIITEM_	ODI
_OZ_ODICATEGORY_	ODI

_OZ_DATASET_	FetchUnit DM_PER_DATASET .
MyParam	ODI .


DataAction
POST

MyParam	odi .
_OZ_DAC_CNT	DataAction <INDEX> .
<INDEX>.DATASET	DataAction
<INDEX>.TYPE	DataAction CUD Insert, Delete, RowUpdate
<INDEX>.EXT	DataAction Extra
<INDEX>.SRC_CNT	DataAction Source <INDEX2> .
<INDEX>.SF_<INDEX2>	DataAction <INDEX2> SourceName
<INDEX>.SV_<INDEX2>	DataAction <INDEX2> SourceValue
<INDEX>.TRG_CNT	DataAction Target <INDEX2> .
<INDEX>.DF_<INDEX2>	DataAction <INDEX2> TargetName
<INDEX>.DV_<INDEX2>	DataAction <INDEX2> TargetValue

: 3 ODI 가 , DataAction 3 Commit
POST

```
paramA=somevalue
paramB=somevalue
paramC=somevalue
_OZ_DAC_CNT=3
0.DATASET=dataset1
0.TYPE=Insert
0.EXT=
0.SRC_CNT=2
0.SF_0=FieldName1
0.SV_0=a
0.SF_1=FieldName2
0.SV_1=b
1.DATASET=dataset1
1.TYPE=Delete
1.EXT=
1.TRG_CNT=1
1.DF_0=FieldName1
```

```
1.DV_0=a
2.DATASET=dataset1
2.TYPE=RowUpdate
2.EXT=
2.SRC_CNT=2
2.SF_0=FieldName1
2.SV_0=newvalue
2.SF_1=FieldName2
2.SV_1=oldvalue
2.TRG_CNT=2
2.DF_0=FieldName1
2.DV_0=newvalue
2.DF_1=FieldName2
2.DV_1=oldvalue
```

Servlet API

■ DataModuleFactory

- getDataModule

Prototype	public static DataModule getDataModule(String FetchType) throws OZSDMException		
Definition	Fetch	가	.
Argument	<i>FetchType</i>	Fetch DM_CONCURRENT_FETCH	DM_BATCH_FETCH

■ DataModule

- init

Prototype	public void init(OutputStream out) throws IOException		
Definition		Stream	.
Argument	<i>out</i>	Response	OutputStream

- startBinding

Prototype	public void startBinding() throws IOException, SQLException		
------------------	---	--	--

Definition

- endBinding

Prototype `public void endBinding() throws IOException, SQLException`

Definition

- startSet

Prototype `public void startSet(String DatasetName) throws
IOException`

Definition

Argument *DatasetName*

- endSet

Prototype `public void endSet(String DatasetName) throws IOException`

Definition

Argument *DatasetName*

- makeSDM_SET

Prototype `public void makeSDM_SET(String DatasetName, ResultSet rs,
OutputStream out) throws OZSDMException, IOException`

Definition

DM_PER_DATASET	SDM
<i>DatasetName</i>	

Argument

<i>rs</i>	ResultSet
<i>out</i>	Response OutputStream

- addParameter

Prototype `public void addParameter(String ParamName, int FieldType,
Object Value)`

Definition

Argument *ParamName*
FieldType

	<i>Value</i>
--	--------------

- addSetInfo

Prototype	public void addSetInfo(String DatasetName, String MasterDatasetName, String[] FieldNames, int[] FieldTypes) throws IllegalArgumentException
Definition	.
	<i>DatasetName</i>
Argument	<i>MasterDatasetName</i> , "" <i>FieldNames</i> <i>FieldTypes</i>

- addrow

Prototype	public void addRow(String DatasetName, HashMap hMap) throws IOException, SQLException
Definition	ResultSet Row 가 . <i>DatasetName</i>
Argument	<i>hMap</i> HashMap 가 Row) ... rs = stmt.executeQuery(query); while(rs.next()) { HashMap hMap = new HashMap(); hMap.put(_FN[0], rs.getString(_FN[0])); hMap.put(_FN[1], rs.getString(_FN[1])); hMap.put(_FN[2], rs.getString(_FN[2])); hMap.put(_FN[3], rs.getString(_FN[3])); hMap.put(_FN[4], rs.getString(_FN[4])); hMap.put(_FN[5], rs.getString(_FN[5])); module.addRow("SET_1", hMap); } ...

- addrow

Prototype	public void addRow(String DatasetName, List list) throws IOException, SQLException		
Definition	ResultSet	Row	가 .
Argument	<i>DatasetName</i>		
	<i>list</i>	ArrayList	가 Row

- addrow

Prototype	public void addRow(String DatasetName, String[] arr) throws IOException, SQLException		
Definition	ResultSet	Row	가 .
Argument	<i>DatasetName</i>		
	<i>arr</i>	String[]	가 Row

- sendBindErrorMessage

Prototype	public void sendBindErrorMessage(String msg) throws IOException		
Definition	가 ,		
Argument	<i>msg</i>		

- sendErrorMessage

Prototype	public void sendErrorMessage(String msg, OutputStream out) throws IOException		
Definition	가 ,		
Argument	<i>msg</i>		
	<i>out</i>	Response	OutputStream

Servlet API 1 : Application - , DataAction

Servlet API Table ,
, ,
Servlet API ozsdmapi.jar ,
DataAction
UDS , , ODI
가 .

Step 1 ozsdmapi.jar

Servlet API ozsdmapi.jar
Tomcat 5.0 webapps\ROOT\WEB-INF\lib
ozsdmapi.jar .

Step 2

java
DataModuleSampleServlet.class
package sample;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import oz.framework.api.DataModule;
import oz.uds.rs.ListMapResultSet;
import oz.sdm.DataModuleFactory;
import oz.uds.rs.ListMapResultSet;
/**
 * <p>Title: OZ SDM API</p>
 * <p>Description: </p>
 * <p>Copyright: Copyright (c) 2005</p>
 * <p>Company: </p>
 * @author Forcs
 * @version 1.0
 */
public class DataModuleSampleServlet extends HttpServlet
{
 private static final String _KEY_ODI_FETCH_TYPE = "_OZ_ODIFetchType_";
 private static final String _KEY_ODIITEM = "_OZ_ODIITEM_";
 private static final String _KEY_ODICATEGORY = "_OZ_ODICATEGORY_";
 private static final String _KEY_OZ_DATASET = "_OZ_DATASET_";
 private Connection m_conn = null;
 public void init(ServletConfig config)
 throws ServletException
 {

```

        super.init(config);

    }

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

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

    private void process(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException
    {
        // init connection 가 .
        try {
            String _URL = "jdbc:odbc:ozdemokr30";
            Properties prop = new Properties();
            prop.put("user", "");
            prop.put("password", "");

            Driver driver = (Driver)
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver").newInstance();
            m_conn = driver.connect(_URL, prop);
            String aaa = null;
        }
        catch(Exception e) {
            DataModule.sendMessage("connection
error...", response.getOutputStream());
            throw new ServletException(e.getMessage());
        }

        try {
            System.out.println("-----");
            Enumeration enum = request.getParameterNames();
            while (enum.hasMoreElements()) {
                String temp = (String) enum.nextElement();
                System.out.println("name=" + temp + " value=" +
getEncode(request.getParameter(temp)));
            }

            // -----
            -----
            // fetch parameters
            // "DM_BATCH_FETCH|DM_CONCURRENT_FETCH"
            String odiFetchType = getEncode(request.getParameter(
_KEY_ODI_FETCH_TYPE));
            String item = getEncode(request.getParameter(_KEY_ODIITEM));
            String category = getEncode(request.getParameter(_KEY_ODICATEGORY));
            String dataset = getEncode(request.getParameter(_KEY_OZ_DATASET));
            // -----
            -----

            if (odiFetchType == null) odiFetchType = "DM_CONCURRENT_FETCH";

            if ( (dataset == null) || (dataset.length() == 0)) {
                // ODI FetchUnit DM_PER_DATAMODULE
                if (item.equals("JASMIN_SAMPLE3.odi")) {
                    // Master-Detail Sample
                    Master_Detail_Style(request, response, odiFetchType);
                }
                else if (item.equals("JASMIN_SAMPLE2.odi") ||
(item.equals("JASMIN_SAMPLE4.odi"))) {
                    // SET Sample
                    System.out.println("default...");
                }
            }
        }
    }

```

```

        default_Style(request, response, odiFetchType);
    }
    else {
        DataModule.sendMessage("unsupported odi error...",
                                response.getOutputStream());
        throw new ServletException("Unknown Item :" + item);
    }
}
else {
    // ODI FetchUnit    DM_PER_DATASET
    //                  odiFetchType    DM_CONCURRENT_FETCH

    //
    if (odiFetchType.equalsIgnoreCase("DM_CONCURRENT_FETCH")) {
        this.SET_Style(request, response, "DM_CONCURRENT_FETCH");
    }
    else {
        DataModule.sendMessage("unsupported error...",
                                response.getOutputStream());
        throw new ServletException("unsupported");
    }
}

if (m_conn != null) {
    try {
        m_conn.close();
    }
    catch (Exception e) {
    }
}

catch(Exception ex){
    //          sendMessage
    // throw          client가
    DataModule.sendMessage(ex.getMessage(),
                            response.getOutputStream());
    throw new ServletException(ex.getMessage());
}

}

private void Master_Detail_Style(HttpServletRequest request,
HttpServletResponse response, String fetchType)
    throws ServletException, IOException
{
    Statement stmt1 = null;
    Statement stmt4 = null;
    ResultSet rs1 = null;
    ResultSet rs4 = null;

    DataModule module = null;

    // jasmin/JASMIN_SAMPLE1.odi
    // SET_1
    String[] _FN1 = {"FirstName" };
    int[] _FT1 = {java.sql.Types.VARCHAR };
    // SET_4
    String[] _FN4 = {"ContactId", "FirstName", "LastName", "Phone",
"Fax", "Email", "OrgUnitId", "UserName" };
    int[] _FT4 = {4, 12, 12, 12, 12, 12, 5, 12 };
    try {
        module = DataModuleFactory.getDataModule(fetchType);

        module.init(response.getOutputStream());

        // set SET Info.
        module.addSetInfo("SET_1", "", _FN1, _FT1);
        module.addSetInfo("SET_4", "SET_1", _FN4, _FT4);
    }
    catch(Exception ex) {
        throw new ServletException(ex.getMessage());
    }
    try {
        module.startBinding();
    }
}

```

```

String query1 = "select distinct FirstName from contact";

module.startSet("SET_1");
stmt1 = m_conn.createStatement();
rs1 = stmt1.executeQuery(query1);
while(rs1.next()) {
    String f1 = rs1.getString(_FN1[0]);
    HashMap map1 = new HashMap();
    map1.put(_FN1[0], f1);

    module.addRow("SET_1", map1);

    String query4 = "select * from contact where FirstName='" +
f1 + "' order by contactid";

    module.startSet("SET_4");
    stmt4 = m_conn.createStatement();
    rs4 = stmt4.executeQuery(query4);
    while(rs4.next()) {
        HashMap map4 = new HashMap();
        map4.put(_FN4[0], new Integer(rs4.getInt(_FN4[0])));
        map4.put(_FN4[1], rs4.getString(_FN4[1]));
        map4.put(_FN4[2], rs4.getString(_FN4[2]));
        map4.put(_FN4[3], rs4.getString(_FN4[3]));
        map4.put(_FN4[4], rs4.getString(_FN4[4]));
        map4.put(_FN4[5], rs4.getString(_FN4[5]));
        map4.put(_FN4[6], new Integer(rs4.getInt(_FN4[6])));
        map4.put(_FN4[7], rs4.getString(_FN4[7]));

        module.addRow("SET_4", map4);
    }
    rs4.close(); rs4 = null;
    stmt4.close(); stmt4 = null;

    module.endSet("SET_4");
}

rs1.close(); rs1 = null;
stmt1.close(); stmt1 = null;

module.endSet("SET_1");

module.endBinding();
}
catch(Exception e) {
    e.printStackTrace();
    // sendBindErrorMessage
    if(module != null) {
        module.sendBindErrorMessage(e.toString());
    }
}
finally {
    if(rs1 != null) {
        try { rs1.close(); } catch(Exception e) {}
    }
    if(stmt1 != null) {
        try { stmt1.close(); } catch(Exception e) {}
    }
    if(rs4 != null) {
        try { rs4.close(); } catch(Exception e) {}
    }
    if(stmt4 != null) {
        try { stmt4.close(); } catch(Exception e) {}
    }
}
}

private void default_Style(HttpServletRequest request,
HttpServletResponse response, String fetchType)
throws ServletException, IOException
{
    System.out.println("select time==" + System.currentTimeMillis());
    Statement stmt = null;
    ResultSet rs = null;

    DataModule module = null;

```

```

// jasmin/JASMIN_SAMPLE2.odi
// SET_1
String[] _FN = {"CarID", "Maker", "EMaker", "CarName", "ECarName",
               "CarImageFile"};
int[] _FT = {
    java.sql.Types.VARCHAR, java.sql.Types.VARCHAR,
    java.sql.Types.VARCHAR, java.sql.Types.VARCHAR,
    java.sql.Types.VARCHAR, java.sql.Types.VARCHAR
};

try {
    module = DataModuleFactory.getDataModule(fetchType);
    module.init(response.getOutputStream());
//    module.init(output);

    // set Parameter Info.
    module.addParameter("PARAM1", java.sql.Types.VARCHAR,
                       getEncode(request.getParameter("PARAM1")));
    module.addParameter("PARAM2", java.sql.Types.VARCHAR,
                       getEncode(request.getParameter("PARAM2")));

    // set SET Info.
    module.addSetInfo("SET_1", "", _FN, _FT);
} catch (Exception ex) {
    throw new ServletException(ex.getMessage());
}

try {
    module.startBinding();

    String query = "select * from car";

    module.startSet("SET_1");
    stmt = m_conn.createStatement();
    rs = stmt.executeQuery(query);
    while(rs.next()) {
        HashMap map = new HashMap();
        String a = rs.getString(_FN[0]);
        String b = rs.getString(_FN[1]);
        String c = rs.getString(_FN[2]);
        String d = rs.getString(_FN[3]);
        String e = rs.getString(_FN[4]);
        String f = rs.getString(_FN[5]);

        map.put(_FN[0], a);
        map.put(_FN[1], b);
        map.put(_FN[2], c);
        map.put(_FN[3], d);
        map.put(_FN[4], e);
        map.put(_FN[5], f);

        module.addRow("SET_1", map);
        System.out.print("select FN0="+_FN[0]+" name="+a);
        System.out.print("    FN1="+_FN[1]+" name="+b);
        System.out.print("    FN2="+_FN[2]+" name="+c);
        System.out.print("    FN3="+_FN[3]+" name="+d);
        System.out.print("    FN4="+_FN[4]+" name="+e);
        System.out.print("    FN5="+_FN[5]+" name="+f);
        System.out.println("");
    }
    module.endSet("SET_1");
    module.endBinding();

    rs.close(); rs = null;
    stmt.close(); stmt = null;
}
catch (Exception e) {
    e.printStackTrace();
    // sendBindErrorMessage
    module.sendBindErrorMessage(e.toString());
}
finally {
    OutputStream out = response.getOutputStream();

    if(rs != null) {
        try { rs.close(); } catch (Exception e) {}
    }
}

```

```
        }
        if(stmt != null) {
            try { stmt.close(); } catch(Exception e) {}
        }
    }

    private void SET_Style(HttpServletRequest request, HttpServletResponse
response, String fetchType)
        throws ServletException, IOException
    {
        Statement stmt = null;
        ResultSet rs = null;

        DataModule module = null;
        try {
            // jasmin/JASMIN_SAMPLE2.odi
            // SET_1
            String[] _FN = {"CarID", "Maker", "EMaker", "CarName",
"ECarName", "CarImageFile" };
            int[] _FT = {java.sql.Types.VARCHAR, java.sql.Types.VARCHAR,
                        java.sql.Types.VARCHAR, java.sql.Types.VARCHAR,
                        java.sql.Types.VARCHAR, java.sql.Types.VARCHAR};

            module = DataModuleFactory.getDataModule(fetchType);

            // set Parameter Info.
            module.addParameter("PARAM1", java.sql.Types.VARCHAR,
getEncode(request.getParameter("PARAM1")));
            module.addParameter("PARAM2", java.sql.Types.VARCHAR,
getEncode(request.getParameter("PARAM2")));

            String query = "select * from car";

            stmt = m_conn.createStatement();
            rs = stmt.executeQuery(query);

            module.makeSDM_SET("SET_1", rs, response.getOutputStream());

            rs.close(); rs = null;
            stmt.close(); stmt = null;

        }
        catch(Exception e) {
            e.printStackTrace();
            // set sendBindErrorMessage
            // makeSDM_SET 가
            throw new ServletException(e.getMessage());
        }
        finally {
            if(rs != null) {
                try { rs.close(); } catch(Exception e) {}
            }
            if(stmt != null) {
                try { stmt.close(); } catch(Exception e) {}
            }
        }
    }

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

    private String getEncode(String value) {
        try {
            return new String(value.getBytes("8859_1"), "KSC5601");
        } catch(Exception ex){
            return value;
        }
    }
}
```

Step 3 DataAction

DataAction

java

DataActionSampleServlet.class

```

package sample;

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

import java.util.*;

/**
 * <p>Title: OZ SDM API</p>
 * <p>Description: </p>
 * <p>Copyright: Copyright (c) 2005</p>
 * <p>Company: </p>
 * @author Forcs
 * @version 1.0
 */

public class DataActionSampleServlet extends HttpServlet
{
    private static final String _KEY_ODI_FETCH_TYPE = "_OZ_ODIFetchType_";
    private static final String _KEY_ODIITEM = "_OZ_ODIITEM_";
    private static final String _KEY_ODICATEGORY = "_OZ_ODICATEGORY_";
    private static final String _KEY_OZ_DATASET = "_OZ_DATASET_";

    private static final String _KEY_OZ_DAC_CNT = "_OZ_DAC_CNT";

    private static String TABLENAME = "car";

    private String dac_insert_query = "";
    private String dac_update_query = "";
    private String dac_delete_query = "";

    private PreparedStatement p_insert_stmt = null;
    private PreparedStatement p_update_stmt = null;
    private PreparedStatement p_delete_stmt = null;

    private Connection m_conn = null;

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

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

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

    private void process(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException
    {
        try {
            String _URL = "jdbc:odbc:ozdemokr30";
            Properties prop = new Properties();
            prop.put("user", "");
            prop.put("password", "");

```

```

        Driver driver = (Driver)
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver").newInstance();
        m_conn = driver.connect(_URL, prop);
    }
    catch(Exception ex) {
        response.getOutputStream().write(ex.getMessage().getBytes());
        throw new ServletException(ex.getMessage());
    }

    try {
        dac_insert_query = "";
        dac_update_query = "";
        dac_delete_query = "";

        Enumeration enum = request.getParameterNames();
        while (enum.hasMoreElements()) {
            String temp = (String) enum.nextElement();
            System.out.println("name=" + temp + " value=" +
                               getEncode(request.getParameter(temp)));
        }

        // -----
        // fetch parameters
        // "DM_BATCH_FETCH|DM_CONCURRENT_FETCH"
        String odiFetchType = getEncode(request.getParameter(
            _KEY_ODI_FETCH_TYPE));
        String item = getEncode(request.getParameter(_KEY_ODIITEM));
        String category =
getEncode(request.getParameter(_KEY_ODICATEGORY));
        String dataset =
getEncode(request.getParameter(_KEY_OZ_DATASET));
        // -----
        System.out.println("-----");
        if (odiFetchType == null) odiFetchType = "DM_CONCURRENT_FETCH";

        m_conn.setAutoCommit(false);

        // Action
        int dac_cnt =
Integer.parseInt(request.getParameter(_KEY_OZ_DAC_CNT));

        for (int i = 0; i < dac_cnt; i++) {
            String type = request.getParameter(i + ".TYPE");
            if (type.equalsIgnoreCase("insert")) {
                insert(request, response, i);
            }
            else if (type.equalsIgnoreCase("rowupdate")) {
                update(request, response, i);
            }
            else if (type.equalsIgnoreCase("delete")) {
                delete(request, response, i);
            }
            System.out.println(type+"
time==" + System.currentTimeMillis());
        }
        m_conn.commit();
        response.getOutputStream().write(new String("OK").getBytes());
    } catch (Exception ex) {
        ex.printStackTrace();
        try { m_conn.rollback(); } catch (Exception e) {}
        response.getOutputStream().write(ex.getMessage().getBytes());
        throw new ServletException(ex.getMessage());
    }

    finally {
        if (p_insert_stmt != null) {
            try { p_insert_stmt.close(); } catch (Exception e) {}
        }
        if (p_update_stmt != null) {
            try { p_update_stmt.close(); } catch (Exception e) {}
        }
        if (p_delete_stmt != null) {
            try { p_delete_stmt.close(); } catch (Exception e) {}
        }

        if (m_conn != null) {
            try { m_conn.close(); } catch (Exception e) {}
        }
    }

```

```

    }
}

private void insert(HttpServletRequest request, HttpServletResponse
response, int cnt)
    throws Exception
{
    int insert_source_fieldCnt =
Integer.parseInt(request.getParameter(cnt + ".SRC_CNT"));

    if (dac_insert_query.equalsIgnoreCase("")) {
        // prepared
        dac_insert_query = "INSERT INTO "+TABLENAME+" (";
        for (int i = 0; i < insert_source_fieldCnt; i++) {
            String s_fieldName = getEncode(request.getParameter(cnt +
".SF_" + i));
            if (i == insert_source_fieldCnt - 1) {
                dac_insert_query += s_fieldName;
            }
            else {
                dac_insert_query += s_fieldName + ",";
            }
        }
        dac_insert_query += ") VALUES ( ";
        for (int i = 0; i < insert_source_fieldCnt; i++) {
            if (i == insert_source_fieldCnt - 1) {
                dac_insert_query += "?";
            }
            else {
                dac_insert_query += "? ,";
            }
        }
        dac_insert_query += ")";
        System.out.println("dac_insert_query="+dac_insert_query);
        p_insert_stmt = m_conn.prepareStatement(dac_insert_query);

        for (int i = 0; i < insert_source_fieldCnt; i++) {
            String value = getEncode(request.getParameter(cnt + ".SV_" +
i));
            p_insert_stmt.setString(i + 1, value);
        }

        p_insert_stmt.execute();
    }

    private void update(HttpServletRequest request, HttpServletResponse
response, int cnt)
        throws Exception
    {
        int update_source_fieldCnt =
Integer.parseInt(request.getParameter(cnt + ".SRC_CNT"));
        int update_target_fieldCnt = 0;

        try {
            update_target_fieldCnt =
Integer.parseInt(request.getParameter(cnt + ".TRG_CNT"));
        } catch (Exception ex) {}

        if (dac_update_query.equalsIgnoreCase("")) {
            // prepared
            dac_update_query = "UPDATE "+TABLENAME+" set ";
            for (int i = 0; i < update_source_fieldCnt; i++) {
                String s_fieldName = getEncode(request.getParameter(cnt +
".SF_" + i));
                if (i == update_source_fieldCnt - 1) {
                    dac_update_query += s_fieldName + " = ? ";
                }
                else {

```

```
        dac_update_query += s_fieldName + " = ?, ";
    }
    }
    dac_update_query += " WHERE ";
    for (int i = 0; i < update_target_fieldCnt; i++) {
        String t_fieldName = getEncode(request.getParameter(cnt +
".DF_" + i));
        if (i == update_target_fieldCnt - 1) {
            dac_update_query += t_fieldName + " = ? ";
        }
        else {
            dac_update_query += t_fieldName + " = ? AND ";
        }
    }

    System.out.println("dac_update_query="+dac_update_query);
    p_update_stmt = m_conn.prepareStatement(dac_update_query);
}

int i = 0;
for (i = 0; i < update_source_fieldCnt; i++) {
    String value = getEncode(request.getParameter(cnt + ".SV_" +
i));
    p_update_stmt.setString(i + 1, value);
}

for (int j = 0; j < update_source_fieldCnt; j++) {
    String value = getEncode(request.getParameter(cnt + ".DV_" +
j));
    p_update_stmt.setString(i + 1, value);
    i++;
}
p_update_stmt.execute();
}

private void delete(HttpServletRequest request, HttpServletResponse
response, int cnt)
    throws Exception
{
    int delete_target_fieldCnt =
Integer.parseInt(request.getParameter(cnt + ".TRG_CNT"));

    if (dac_delete_query.equalsIgnoreCase("")) {
        // prepared
        dac_delete_query = "DELETE FROM "+TABLENAME+" WHERE ";
        for (int i = 0; i < delete_target_fieldCnt; i++) {
            String t_fieldName = getEncode(request.getParameter(cnt +
".DF_" + i));
            if (i == delete_target_fieldCnt - 1) {
                dac_delete_query += t_fieldName + " = ? ";
            }
            else {
                dac_delete_query += t_fieldName + " = ? AND ";
            }
        }

        System.out.println("dac_delete_query="+dac_delete_query);
        p_delete_stmt = m_conn.prepareStatement(dac_delete_query);
    }

    int i = 0;
    for (i = 0; i < delete_target_fieldCnt; i++) {
        String value = getEncode(request.getParameter(cnt + ".DV_" +
i));
        p_delete_stmt.setString(i + 1, value);
    }

    p_delete_stmt.execute();
}
```

```

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

    private String getEncode(String value) {
        try {
            return new String(value.getBytes("8859_1"), "KSC5601");
        } catch (Exception ex) {
            return value;
        }
    }
}

```

Step 4

DataModuleSampleServlet.class, DataActionSampleServlet.class

➤ DataModuleSampleServlet.class, DataActionSampleServlet.class
Tomcat 5.0
webapps\ROOT\WEB-INF\classes sample

➤ DataModuleSampleServlet.class, DataActionSampleServlet.class
Tomcat 5.0 webapps\ROOT\WEB-INF
web.xml

```

...

<!-- JSPC servlet mappings start -->
...
<servlet>
    <servlet-name>sample.DataModuleSampleServlet</servlet-name>
    <servlet-class>sample.DataModuleSampleServlet</servlet-class>
</servlet>

<servlet>
    <servlet-name>sample.DataActionSampleServlet</servlet-name>
    <servlet-class>sample.DataActionSampleServlet</servlet-class>
</servlet>

...
<servlet-mapping>
    <servlet-name>sample.DataModuleSampleServlet</servlet-name>
    <url-pattern>/sample.DataModuleSampleServlet</url-pattern>
</servlet-mapping>

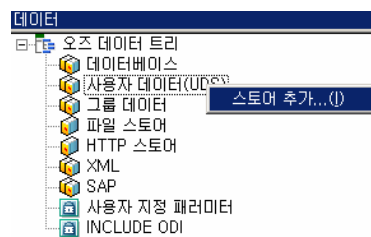
```

```
<servlet-mapping>
    <servlet-name>sample.DataActionSampleServlet</servlet-name>
    <url-pattern>/sample.DataActionSampleServlet</url-pattern>
</servlet-mapping>

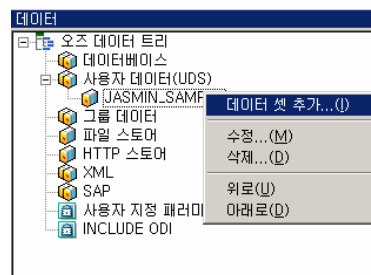
...
<!-- JSPC servlet mappings end -->
```

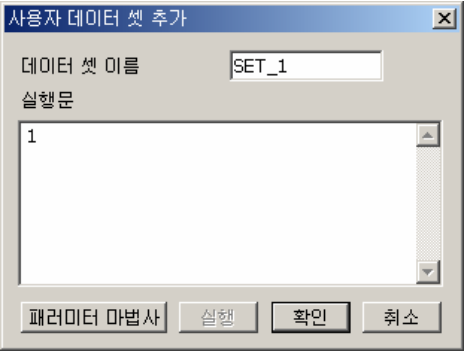
Step 5 ODI

Servlet API (UDS) ,
가 .
> (UDS) 가

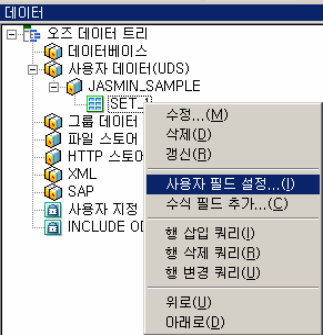


> 가 가 .

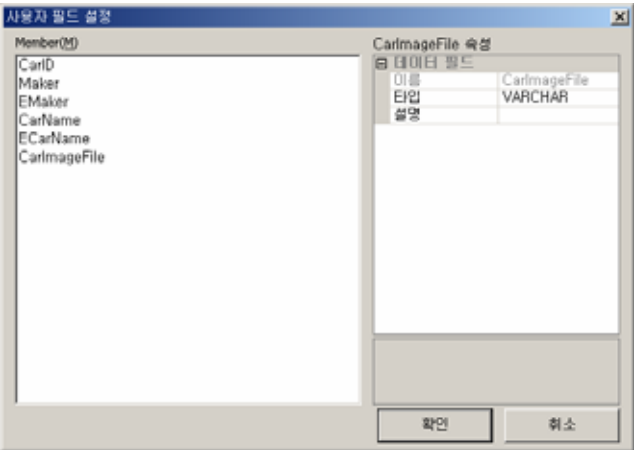




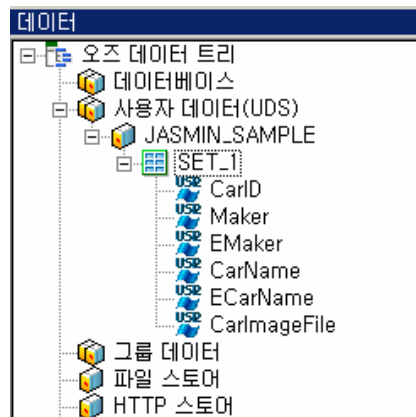
➤ 가 가 . 가
[]



➤ 가 []



가 가 .



➤ SET_1 [], [],
[] , ,

```
INSERT INTO car(##ARG_SF1#, ##ARG_SF2#, ##ARG_SF3#, ##ARG_SF4#,
##ARG_SF5#, ##ARG_SF6#)
VALUES('##ARG_SV1#', '##ARG_SV2#', '##ARG_SV3#', '##ARG_SV4#',
'##ARG_SV5#', '##ARG_SV6#')
```

```
DELETE FROM car WHERE [##ARG_DF1#] = '##ARG_DV1#';
```

```
UPDATE car SET
[##ARG_SF1#] = '##ARG_SV1#', [##ARG_SF2#] = '##ARG_SV2#',
[##ARG_SF3#] = '##ARG_SV3#', [##ARG_SF4#] = '##ARG_SV4#',
[##ARG_SF5#] = '##ARG_SV5#', [##ARG_SF6#] = '##ARG_SV6#'
WHERE [##ARG_DF1#] = '##ARG_DV1#'
```

➤ ODI "JASMIN_SAMPLE4.odi"

Step 6 OZF

: Servlet API 가
URL DataModule
RegisterUserDataModule JavaScript
. JavaScript
, Document.GlobalFunction OZF
OZF 가 .

OZF

가

.



OZF

"MyFrameworkURLUDS.ozf"

.

```

MyFrameworkURLUDS.prototype.GetFrameworkURL =
    MyFrameworkURLUDS_GetFrameworkURL;
MyFrameworkURLUDS.prototype.GetCUDFrameworkURL =
    MyFrameworkURLUDS_GetCUDFrameworkURL;
MyFrameworkURLUDS.prototype.GetFrameworkPostParam =
    MyFrameworkURLUDS_GetFrameworkPostParam;
MyFrameworkURLUDS.prototype.GetCUDFrameworkPostParam =
    MyFrameworkURLUDS_GetCUDFrameworkPostParam;

function      MyFrameworkURLUDS(_url,      _cud_url,      _url_param,
_cud_url_param){
    this.url = _url;
    this.cud_url = _cud_url;
    this.url_param = _url_param;
    if(this.url_param == null){
        this.url_param = "";
    }
    this.cud_url_param = _cud_url_param;
    if(this.cud_url_param == null){
        this.cud_url_param = "";
    }
}

function MyFrameworkURLUDS_GetFrameworkURL(dataset_name){
    return this.url;
}

function MyFrameworkURLUDS_GetCUDFrameworkURL(dataset_name){
    return this.cud_url;
}

function MyFrameworkURLUDS_GetFrameworkPostParam(dataset_name){
    return "default&"+this.url_param;
}

function MyFrameworkURLUDS_GetCUDFrameworkPostParam(dataset_name){
    return "default&"+this.cud_url_param;
}

```



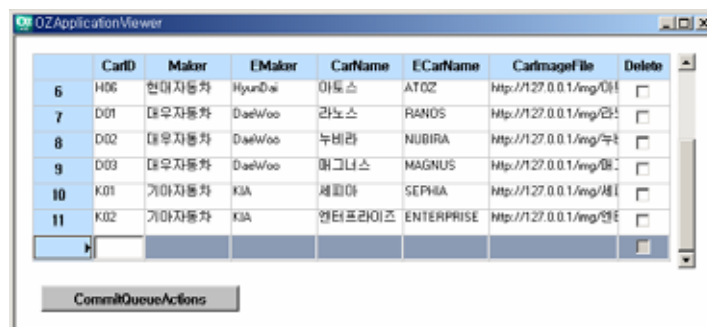
```
var datamanager = _GetDataManager();
var datamodule = datamanager.GetDataModule("JASMIN_SAMPLE4");
datamodule.RegisterUserDataModule(uds);
```

➤ Board Button 가 Button OnClick

```
var result = Table1.CommitQueuedActions();
if(result == "") {
    Table1.GetDataModule().RefreshAllDataSet();
} else {
    _MessageBox(result);
}

//var myODIObject =
    _GetDataManager().GetDataModule("odiName").GetUserDataModule();
//Select 가
// myODIObject.url_param = "key1=value1&key2=value2";
//DataAction 가
// myODIObject.cud_url_param = "key1=value1&key2=value2";
//_GetDataManager().GetDataModule("odiName").RefreshAllDataSet();
```

Step 7



가

Delete

	CarID	Maker	EMaker	CarName	ECarName	CarImageFile	Delete
6	H06	현대자동차	Hyundai	아토스	ATOZ	http://127.0.0.1/img/01	☐
7	D01	대우자동차	DaeWoo	라노스	RANOS	http://127.0.0.1/img/02	☐
8	D02	대우자동차	DaeWoo	누비라	NUBIRA	http://127.0.0.1/img/03	☐
9	D03	대우자동차	DaeWoo	매그너스	MAGNUS	http://127.0.0.1/img/04	☐
10	K01	기아자동차	KIA	세피아 I	SEPHIA	http://127.0.0.1/img/05	☐
11	K02	기아자동차	KIA	엔터프라이즈	ENTERPRISE	http://127.0.0.1/img/06	☒
	K03	기아자동차	KIA	세피아 II	SEPHIA II	http://127.0.0.1/img/07	☐

CommitQueueActions

DataAction

[CommitQueueActions]

	CarID	Maker	EMaker	CarName	ECarName	CarImageFile	Delete
6	H06	현대자동차	Hyundai	아토스	ATOZ	http://127.0.0.1/img/01	☐
7	D01	대우자동차	DaeWoo	라노스	RANOS	http://127.0.0.1/img/02	☐
8	D02	대우자동차	DaeWoo	누비라	NUBIRA	http://127.0.0.1/img/03	☐
9	D03	대우자동차	DaeWoo	매그너스	MAGNUS	http://127.0.0.1/img/04	☐
10	K01	기아자동차	KIA	세피아 I	SEPHIA	http://127.0.0.1/img/05	☐
11	K02	기아자동차	KIA	세피아 II	SEPHIA II	http://127.0.0.1/img/06	☐
							☐

CommitQueueActions

Servlet API 2 : Report -

Servlet API

. Servlet API

frameworkurl

(HTTP URL)

odi.odi .frameworkurl, odi.frameworkurl, connection.frameworkurl

: Servlet API

Step 1 ozsdmap.jar

Servlet API

ozsdmap.jar

Tomcat 5.0 webapps\ROOT\WEB-INF\lib

ozsdmap.jar

Step 2

DataModuleSampleServlet.class

- DataModuleSampleServlet.class

Tomcat 5.0 webapps\ROOT\WEB-INF\classes\sample

- DataModuleSampleServlet.class

Tomcat 5.0 webapps\ROOT\WEB-INF web.xml

```

...

<!-- JSPC servlet mappings start -->
...
<servlet>
    <servlet-name>sample.DataModuleSampleServlet</servlet-name>
    <servlet-class>sample.DataModuleSampleServlet</servlet-class>
</servlet>
...
<servlet-mapping>
    <servlet-name>sample.DataModuleSampleServlet</servlet-name>
    <url-pattern>/sample.DataModuleSampleServlet</url-pattern>
</servlet-mapping>
...
<!-- JSPC servlet mappings end -->

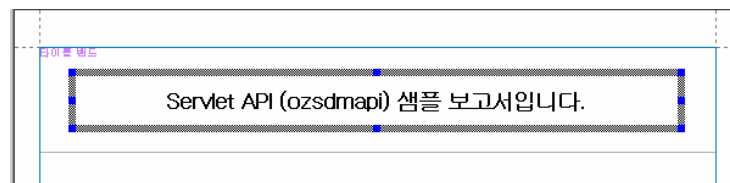
```

Step 3

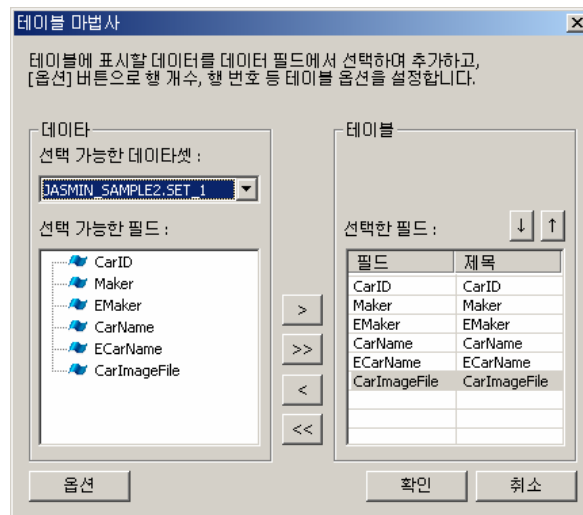
- JASMIN_SAMPLE2.odi

가 .

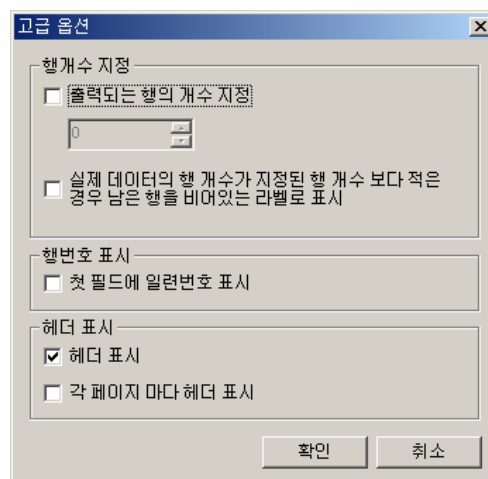
- 가 , 가



- "ODI" 가 "SET_1" .
- "JASMIN_SAMPLE2" , " " "SET_1" .
- & 가 .
- 가 .



- []
- " " []



- 가 []
- 가 , , .

데이터 뷰어

Servlet API (ozsdmapi) 샘플 보고서입니다.

데이터 뷰어

CarID	Maker	EMaker	CarName	ECarName	CarImageFile
<SET_1:CarID>	<SET_1:Maker>	<SET_1:EMaker>	<SET_1:CarName>	<SET_1:ECarName>	<SET_1:CarImageFile>



"Sample_ServletAPI.ozr"

Step 4



Sample_ServletAPI.htm

```
<HTML>
<BODY>
  <OBJECT width = "0" height = "0" ID="ZTransferX"
CLASSID="CLSID:C7C7225A-9476-47AC-B0B0-FF3B79D55E67"
codebase="127.0.0.1:8088/ozrviewer/ZTransferX.cab#version=2,1,0,2">
  <PARAM NAME="download.Server" VALUE="http://127.0.0.1/ozrviewer">
  <PARAM NAME="download.Port" VALUE="8088">
  <PARAM NAME="download.Instruction" VALUE="ozrviewer.idf">
  <PARAM NAME="install.Base" VALUE="<PROGRAMS>/Forcs">
  <PARAM NAME="install.Namespace" VALUE="Sample_ServletAPI">
  </OBJECT>
  <OBJECT id = "ozrviewer" CLASSID="CLSID:0DEF32F8-170F-46f8-B1FF-
4BF7443F5F25" width="100%" height="100%">
    <param name="connection.servlet"
value="http://127.0.0.1:8088/OZServlet40/server">
    <param name="connection.reportname" value="Sample_ServletAPI.ozr">
    <param name="odi.odinames" value="JASMIN_SAMPLE2">
    <param name="odi.JASMIN_SAMPLE2.frameworkurl"
value="http://127.0.0.1:8088/sample.DataModuleSampleServlet">
    <param name="odi.JASMIN_SAMPLE2.fetchtype" value="BATCH">
    <param name="viewer.isframe" value="false">
    <param name="viewer.namespace" value="Sample_ServletAPI\ozviewer">
  </OBJECT>
</body>
</HTML>
```

: Servlet API

odi.odi .frameworkurl odi.frameworkurl
 connection.frameworkurl
 URL

