

# Interacting Clients from the Solution in Servoy, Now, Doable!!

By Arup Ranjan Sahoo

[Mindfire Solutions](#)

## Abstract

Interacting with the Servoy Application Server or any connected clients, such as shutting down/restarting the Application Server, alerting/shutting down/checking alive or not for any client connected to the Application Server or to the Application Server running remotely, from the Servoy solutions is not doable. The Servoy solution can not able to get the handle of the Servoy Application Server to the Developers. This article describes how we can do all those stuffs from your Servoy solution by using the Server Utility Plug-in for Servoy, "**mfServerUtils**", developed by me 😊.

## Problem statements

Have you ever wanted to get the number of Clients connected a specific Servoy Application Server?

Have you ever wanted to alert a specific client from your Servoy Solution itself?

Have you ever wanted to interact with clients connected to the Application Server running remotely?

Have you ever wanted to check whether a Client is alive or not or want to forcefully terminate the Client session from your Servoy Solution itself?

## Previous Situation

Alerting or shutdown a client from another Client is not doable in Servoy.

We also can not terminate a Rich Client session, as because the Rich Client Session does not have expired Mechanism, only Web Client has due to the fact that it has a low level HTTP server Web Session.

## Current Situation

Hurrah! Now, you can do all those stuffs, alerting or shutting down a client (Rich Client/ Web Client), straight from your Servoy Solution, by the help of a Plug-in. The Plug-in uses Java RMI to talk with the Servoy Application Server Instance.

## Introducing Server Utility Plug-in for Servoy – "mfServerUtils 1.0"

"mfServerUtils", is a FREE Servoy Plug-in (Developed by me 😊) that gives the power to developer to control the Application Server and any connected Clients, running on the server. The Plug-in is smart enough to give you the capability to control the client's (Rich Client & Web Client) behavior such as alerting Client, shutting down Client etc. Not only about the Clients can you also control your Servoy Application Server such as alerting all Clients, shutting down the server, restarting the server etc. And all these powerful things right from your Servoy Solution it self. For the security reason, I have added one property called "com.mindfire.plugins.mfserverutils.accesscode" to the Servoy Properties. I recommend, this property should be set and checked while calling any methods of the Plug-in.

The Plug-in uses Java RMI to talk with the Servoy Application Server. A RMI service will be started when the Plug-in get loaded in to the Servoy environment. Any call to plug-in will be handled by the service.

The plug-in has been tested in the Servoy version 4.1.x under the Java version 1.5+ & 1.6+ in windows, Macintosh & Ubuntu Linux 8.10 platform with Servoy Rich/Web Client.

For more information about the Plug-in, [click here](#).

Some Silent Feature of the Plug-in...

- Getting all/specific Client connected to the Application Server
- Shutting down Application Server
- Restart Application Server
- Alerting all the clients connected to the application server
- Checking Client is alive or not
- Shutting down clients
- Alerting clients
- Interacting with the Application Server running remotely
- ...and much more

## Plug-in Installation

Installation of mfServerUtils is very easy. Please, follow the below steps to install the Plug-in.

### Steps:

- Download the Plug-in from [here](#) and unzip the file.
- Copy all the files (mfServerUtils.jar & mfServerUtils.jar.jnlp) to the <Servoy\_Root\_Directory> \application\_server\plugins folder & Restart Servoy.
- Go to the Servoy Admin Page by navigating to <http://serverIPAddress:port/servoy-admin>. Move on to the Server Plug-ins page, by clicking on the link "Server Plug-ins" at left menu.
- Now, you can able to see the "com.mindfire.plugins.mfserverutils.accesscode" property listed under the Header, "Mindfire Solutions: Server Utility Plug-in".
- Set a value for the access-code for the Plug-in. FOR SECURITY REASON, ACCESS-CODE MUST BE SET UP FROM THE SERVOY ADMIN PAGE & CALLED PRIOR TO EVERY CALL TO THE PLUG-IN.
- Restart Servoy Server.
- Now, you are ready with the Plug-in and its powerful features.

## Sample Servoy Solution: Demonstrating the Plug-in

I have also developed the sample Servoy solution to demonstrate each feature of the plug-in live in action. Import the sample solution to become familiar with the plug-in methods. Here are the steps to download and import the sample solution. The "mfServerUtils" Plug-in must be installed prior to importing sample solution.

### Steps:

- Install mfServerUtils Plug-in (Follow the above steps). Ignore if already done.
- Download the Sample Servoy Solution from [here](#).
- Import the sample Servoy solution.
- Checkout the solution to your local drive.
- Navigate to the List of Global Variables of the solution, Change the value of "gAccessCode" global variable to your access-code setup from your Servoy Admin Page (As described in the Plug-in installation).
- Now, you are ready with the Sample Solution, Run the Solution in Smart Client.

## How to USE the Plug-in

### Accessing Application Server Instance

You can get Instance of the Application Server by calling to the `getApplicationServer()` method of the Plug-in. Any calls to the plug-in must followed by checking the access-code. This will return you an object of class `JSApplicationServer`. Now, you can call any of the methods under the `JSApplicationServer` Node.

```
//Check for access-code
var appServer = null;
var access = plugins.mfServerUtils.checkAccessCode('<to_be_filled>');
if(access) {
    //Get Application Server Instance
    appServer = plugins.mfServerUtils.getApplicationServer();
}else {
    application.output('Oops!! access-code does not match.')}
}
```

You can get an instance of the Application Server running remotely by passing server address & the port number as arguments to the `getApplicationServer ()` method.

```
//Get Instance of an Application Server running remotely.
appServer = plugins.mfServerUtils.getApplicationServer("remote_ip","remote_port_number");
```

With the instance of the Application Server, you can shutdown the Application Server, alert all clients connected to the Application Server or restart Server.

```
//Alerting all clients connected to the Application Server
appServer.alertAllClients('Hey, this message is from the server.');
```

```
//Shutting down the Application Server
appServer.shutdownServer();
```

```
//Restarting the Application Server
appServer.restartServer();
```

## Accessing Client Instance

You can get the instance of a specific Client or instances of list of all connected Clients by calling to the getClient() or getClients() method respectively. Any calls to the plug-in must followed by checking the access-code. This will return you an object of class JSClient. Now, you can call any of the methods under the JSClient Node.

```
//Check for access-code
var access = plugins.mfServerUtils.checkAccessCode('<to_be_filled>');

if(access){
    //Getting a specific Client Instance
    var client = plugins.mfServerUtils.getClient(security.getClientID().toString());

    //Getting instance of All Clients connected to the Application Server.
    var clientList = plugins.mfServerUtils.getAllClients();

    for(var count=0; count<clientList.length; count++){
        var curClient = clientList[count];
        //Now, the curClient is single Client instance. Call any Client specific methods.
    }
} else {
    application.output('Oops!! access-code doesn't match.')}
}
```

You can get an instance of the Clients running in the remote Application Server by passing server address & the port number as arguments to the getClient() or getClients() method.

```
//Getting a specific Client Instance of the Application Server running remotely
var client = plugins.mfServerUtils.getClient(security.getClientID().toString(),
                                             "remote_ip","remote_port_number");
```

```
//Getting instance of All Clients connected to the Application Server running remotely
var clientList = plugins.mfServerUtils.getAllClients("remote_ip","remote_port_number");
```

With the instance of Client, you can alert client, check the Client is alive or not or can shutdown clients.

```
//Alerting the Client
client.alertClient('Hey, you are alerted!!');
```

```
//Checking the Client is alive or not
if(client.isClientAlive()){
    application.output('Client is Alive');
}else{
    application.output('Client is not Alive');
}
```

```
//Shutting down Client
client.shutdownClient();
```

## Conclusion

You can interact with any connected client of the same Application Server or other running remotely by using the "mfServerUtils" Plug-in. For Comments/Feedbacks/Enhancements related to the Plug-in, please feel free to write me at [arupranjans@mindfiresolutions.com](mailto:arupranjans@mindfiresolutions.com).

## Reference

For More Information about the Plug-in: [http://www.mindfiresolutions.com/mfserverutils\\_detail.htm](http://www.mindfiresolutions.com/mfserverutils_detail.htm)  
To know more about Servoy: <http://www.servoy.com>