6. – API and Plug-In

- 6.1 – Reporting API (query server status, client side JavaScript API)
- 6.2 – RDP library (client side JavaScript API)
- 6.2.1 – RDP parameters
- 6.2.2 – Passing parameters via URL (Connection String)
- 6.2.3 – Passing parameter via object or cookie
- 6.2.4 – Usage of RDP class
- 6.2.5 – Extend RDP: Virtual Channel and Dynamic Virtual Channel
- 6.2.6 – Extend Gateway: Gateway Channel
- 6.3 – Plug-ins (server side Java API)
- 6.4 – HTTP Request API (server side)

6.1 – Reporting API (query server status, client side JavaScript API)

Include "gateway.js" in your web page:

```
var gw = new Gateway("192.168.9.118");

gw.login("password");
gw.report(); //output to console
gw.report(callback); //output to callback method
gw.checkLicense(callback2); //check license status

function callback(obj) {
  if (obj.error) {
    console.log("Error on getting report:" + obj.error);
    return;
  }
  var c, cs = obj.connections;
  var len = cs.length;
  console.log("\nTotal connections:" + len);
  for ( var i = 0; i < len; i++) {
    c = cs[i];
    console.log("connection " + i + " ------");
    console.log("clientIp: " + c.clientIp);
    console.log("clientAgent: " + c.clientAgent);
    console.log("server: " + c.server);
    console.log("startTime: " + c.startTime);
  }
}
```

We also have a Java command line tool available for the same purpose. Please download it from our website.

We also have WebSocket client available which can be used to communicate with our gateway or other WebSocket servers.

6.2 – RDP library (client side JavaScript API)

Further links to the coresponding subpages:

- 6.2.1 – RDP parameters
- 6.2.2 – Passing parameter via URL
- 6.2.3 – Passing parameter via object or cookie
- 6.2.4 – Usage of RDP class
- 6.2.5 – Extend RDP: Virtual Channel and Dynamic Virtual Channel
- 6.2.6 – Extend Gateway: Gateway Channel

6.2.1 – RDP parameters

Parameter	Value
gateway	BindString, address and listening port of the gateway. For example: 192.168.0.8:443
server	String, address of the RDP host.
port	Integer, RDP listening port. Optional, default is 3389
user	String, user name (Windows User).
pwd	String, password for user name.
domain	String, domain name
keyboard	Integer, keyboard layout, default is 0x409 (US)
useConsole	Boolean, connecting to console session/Admin mode. Default is false.
legacyMode	Boolean, connecting to xrdp or VirtualBox RDP
width	Integer, screen width of RDP session. Default is 800
height	Integer, screen height of RDP session. Default is 600
server_bpp	Integer, color depth of RDP session. Default is 16
playSound	Integer, Default is 1: Do not play sound; 0: bring sound to local; 2: leave sound on remote computer.
startProgram	String, "shell": start a program on connection; "app": start a RemoteApp
command	Command for “Start a program on connection (startProgram=shell)”
directory	Directory for running command (startProgram=shell)
exe	Program or file for RemoteApp (startProgram=app)
args	Arguments for RemoteApp (startProgram=app)
background	Boolean, default is false, disable background.
smoothfont	Boolean, default is false, disable font smoothing.
contents	Boolean, default is false, disable full windows drag.
animation	Boolean, default is false, disable menu animations.
composition	Boolean, default is false, disable desktop composition.
styles	Boolean, default is false, disable theming.
mapClipboard	Boolean, default is false, disable clipboard rediretioin.

Parameter	Value
mapPrinter	Boolean, default is false, disable printer redirection.
mapDisk	Boolean, default is false, disable disk redirection.
touchpad	Boolean, default is false, touchpad mode (relative mouse movement, touch interface only).
waWidth	Work area width for RemoteApp, Default value is the screen width. You may want to change it if you display app in a iframe.
waHeight	Work area height for RemoteApp. Default value is the screen height. You may want to change it if you display app in a iframe.
printer	Printer name. You can specify multiple printer names by using “;” as seperator, e.g. “Printer1;Printer2”. The first one will always be the default printer.
useSSL	Use WSS (WebSocket secure connection), only used by Rdp2 class.
timezone	The client time zone name. Please check the values you can use in rdp.html
loadBalanceInfo	Load balance information
vmid	Hyper-V VM GUID, For example: B3D5444C-2611-405A-9CA0-7AA8DA94DF0B, it’s for Hyper-V console connection.
minWidth	Minimum width, some applications can only work on a minimum resolution
minHeight	Minimum height, some applications can only work on a minimum resolution
connectType	Connecting type: **CONNECTION_TYPE_MODEM 1** Modem (56 Kbps) **CONNECTION_TYPE_BROADBAND_LOW 2** Low-speed broadband (256 Kbps - 2 Mbps) **CONNECTION_TYPE_SATELLITE 3** Satellite (2 Mbps - 16 Mbps with high latency) **CONNECTION_TYPE_BROADBAND_HIGH 4** High-speed broadband (2 Mbps - 10 Mbps) **CONNECTION_TYPE_WAN 5** WAN (10 Mbps or higher with high latency) **CONNECTION_TYPE_LAN 6** LAN (10 Mbps or higher) Default value is 5.

Parameter	Value
shareClipboardId	Connections will same shareClipboardId will use shared clipboard on gateway. User can copy anything among these connections
shareClipboard	on: enabled shared clipboard, it’s enabled by default if mapClipboard is on.
symlink	Id of the created symlink.
sessionRecord	0: no session recording, 1: recording grphic only (no sound), 3 means recording graphic and sound. This can only be configured in pluign
__record_name	Recording file name. This can only be configured in pluign before version 4.8.8.
recording	on: record session on client side
audioRecord	on: enable remote audio recording
pingInterval	Seconds for ping
soundPref	0: low quality sound; 1: high quality sound
smartcard	on: enable smart card redirection
passwordIsPin	Use password as the pin of smart card
name	Show the name on the cursor on joined users
mac	Computer MAC address for Wake on LAN. Need to enable it in gateway.conf (timeoutWoL)
macIP	Network broadcast ip address for WoL.
decompressingRDP61	Enable RDP6.1 compression.
portCheckTimes	Gateway will check if the port is open before connecting
portCheckInterval	Interval of checking open port in ms.
copyToLocal	on: user can copy data to local from remote. It can be used to disable bi-directional copy/paste
copyToRemote	on: user can copy data to remote from local. It can be used to disable bi-directional copy/paste
copyTextOnly	on: user can only copy plain text.
textPrinter	on: redirect a text only printer (for receipts, label printers)
mapCamera	Boolean, default is false, enable webcam redirection. Enable it in RDP server with Group Policy: Computer Configuration > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Device and Resource Redirection > Do not allow supported Plug and Play device redirection: Disabled

Parameter	Value
scanner	Boolean, default is false, enable scanner redirection.

6.2.2 – Passing parameters via URL (Connection String)

URL components

> Important
>
> Please use "on" and "off" for Boolean value if you are using URL to pass parameters.

Basically, a RDP connection string consists of the following main components:

`[protocol]://[domain]/rdpdirect.html?[parameters]`

The parameters are the same as the parameters that can be set in servers.json for each target system. Please note the correct URL encoding (see below)! Link paramters with the ampersand (`&`). A parameters list can be found here.

Some example parameters can be:

```
&keyboard=1031
&fullBrowser=Full%20browser
&fullScreen=Full%20screen
&server_bpp=16
&timezone=W.%20Europe%20Standard%20Time
&playSound=0
&soundPref=0
&mapClipboard=on
&mapPrinter=on
&mapDisk=on
&startProgram=noapp
&smoothfont=on
```

The correct URL encoding:

If characters are to be transmitted as values that have special meaning in the context of a URL, they must be encoded.

Example: `name=SparkView Server` becomes `name=SparkView%20Server`.

For correct encoding, an online tool can also be applied for help: https://meyerweb.com/eric/tools/dencoder/.

Example connection string:

A minimal connection string can look like this:

```
https://mygateway.com/rdpdirect.html?server=192.168.12.55&port=3389&gateway=mygateway.com
```

Connecting to desktop

```
var parameters = "server=192.168.0.2&user=admin&pwd=" + encodeURIComponent("&=@#");
//use encodeURIComponent to escape special characters in value
var width = 800, height = 600, server_bpp = 16;
varr = new svGlobal.Rdp("http://" + gateway + "/RDP?" + parameters, width, height,
server_bpp);
r.addSurface(new svGlobal.LocalInterface());
r.run();
```

Connecting to RemoteApp in current window

You only need to add some extra parameters:

```
parameters += encodeURIComponent("startProgram=app&exe=||EXCEL");
```

Connecting to RemoteApp in a new window

```
varparameters = "server=192.168.0.2&user=admin&pwd=" + encodeURIComponent("&=@#");
//use encodeURIComponent to escape special characters in value
parameters += encodeURIComponent("startProgram=app&exe=||EXCEL");
var width = 800, height = 600, server_bpp = 16;
var r = svManager.getInstance(); //try use the existed session.
if (r == null){
  r = new svGlobal.Rdp(protocol + gw + "/RDP?" + s, width, height, server_bpp);
}
var rail = window.open("rail.html");
function onSurfaceReady(surface){
```

```
  r.addSurface(surface);
  r.startApp(frmConn["exe"].value, frmConn["args"].value, "");
};
rail.svOnSurfaceReady = onSurfaceReady;
r.run();
```

6.2.3 – Passing parameter via object or cookie

Rdp2 class can be used to replace Rdp class. You can pass a object to Rdp2:

```
varobj = {gateway: "192.168.0.2", server:"192.168.0.8", user: "user"};
var r = new svGlobal.Rdp2(obj);
```

Rdp2 class will create an object from cookies if obj is undefined. If value of the document.cookie is "gateway=192.168.0.2;server=192.168.0.8;user=user", Rdp2 will create a object automatically for connection.

6.2.4 – Usage of RDP class

Properties:

Name	Type	Description
appTimeout	Int	Close the RemoteApp if no Windows found after this period, default is 800 ms.
displayMsg	Boolean	If display error or warning message, default is true
reconnectOnResize	Boolean	If reconnect when resize the browser window, default is true
reconnectTimes	Int	Automatically reconnecting time,default 0.
sessionTimeout	Int	Close the session disconnection if no RemoteApp running after this period. Default value is 3000 (3 seconds). You may want to make it longer, so user can use the current session for new RemoteApp.
windowState	Int	State of RemoteApp main window, 3: always maximized, 0: controlled by user
openLink	Boolean	Display a web link button when user copy a web link in remote computer, default is true
sessionInfo	Object	Current session information. Including following properties: sessionId, userName, domain etc.
remoteAppLogin	Boolean	If display login details when using RemoteApp, default is true
setTitle	Boolean	If allow client to change browser title, default is true, the browser title will be set as the RDP host name.

Methods:

Name	Description
close()	Close current RDP session
exeAppCmd(cmd)	Execute command on current RemoteApp, cmd (Int) values: 0xF020: Minimize the window 0xF030: Maximize the window 0xF060: Close the window 0xF100: Display the windo’s system menu 0xF120: Restore the window
mouseDown(x, y, button)	Send mouse down to RDP host. X, Y: mouse position, button: which button, same as event.button.
mouseUp(x, y, button)	Send mouse up to RDP host.
mouseMove(x, y)	Send mouse move to RDP host.
pauseSession(numericId, pause)	Pause a session output, pause: true or false. Pause all joined sessions if numericId is 0
running()	Check if connected to a RDP session
setAudioBuffer(seconds)	Audio buffer size, default is 2.0 seconds
writeKeyCode(down, keyCode)	Send browser keyCode to RDP host. “down”: Boolean
writeText(txt, interval, doneCallback)	Send Unicode to RDP host. RDP server could ignore some characters if the client is sending them too fast. Please use the optional interval and call back arguments.
writeRawInput(input)	Send raw keyboard, mouse events to RDP host. Please use
onactivity event to get the raw input events.	
writeKeyComb(keys, interval, doneCallback)	Write any key combinations like “Ctrl+Alt+Del”. if interval > 0, every key stroke (down or up) will be sent with delay.
notify(message, sessionIDArray)	Send message to other session. sessionIDArray is optional. If sessionIDArray is not given, this will send message to all other joined and main session. If sessionIDArray is provided, it will send message to sessions with the given ID. For example: [id1, id2, id3]. The id can be the 9 digit number id or the GUID.
saveSession(info)	This will leave the session open on the gateway, so you can connect to it later from same or different device. For example: saveSession({save: true, timeout:5, id: xxxx}) will leave the session open for 5 minutes on gateway with id xxxx. You can connect to this cached session with the given id. Please check 3.14 for more information about cached session. You can also use plugin to save session. This feature will be disabled if savedSessionTimeout in gateway.conf is 0.

Name	Description
startPing(interval, noResponTime)	Start to ping the gateway when the session is idle. Interval: seconds. noResponTime: seconds, time limit for no response. onnoresponse event will be fired if this value reached, default is 0 will never fire onnoresponse event.
setJoinMode(mode)	0: Every one can control; 1: Only one can control; 2: Only one can control (others can move mouse)
setJoinCloseMode(mode)	0: close all joined sessions after initiated session is closed; 1: leave all joined sessions open even initiated session is closed.

writeClipboard(text, type)	dp.writeClipboard('Plain text');
	rdp.writeClipboard('<h>Header</h>', 'text/html');
	var reader = new FileReader();
	reader.onloadend = function (e) {
	if (e.target.readyState == FileReader.DONE) {
	var imgB64 = hi5.Base64.enc(new Uint8Array(e.target.result));
	rdp.writeClipboard(imgB64, 'image/png');
	}
	};
	reader.readAsArrayBuffer(blob);

sendKeyboardSynchronize(scroll, numlock, capital)	Turn off/on lock key state on remote computer.

Events:

Name	Description
onclose(expected)	Fired when RDP session is closed. Expected is true if the session is closed by user (log off) or administrators. Please be aware, the Rdp object will be released and cannot be reused after this event.
onerror(error)	Fired on an error. Use error.name, error.message to get error details.
onloggedin()	Fired when user logged in (It's not supported by xRDP and VirtualBox RDP)
beforeupload(file, path)	Fired before uploading a file: file: File object path: file absolute path or name if path cannot be obtained. Return true if it's processed and prevent default behavior happening.

Name	Description
onfileuploaded(fileName)	Fired after a file was uploaded
onuploaded()	Fired after all files were uploaded
onsessionjoin(appInfo)	Fired when a new user joined the session. appInfo has following properties: numericId: the 9 digit id __ip: joined user's hone name or ip address name: if provided Return true will override the default behavior.
onsessionexit(sessionInfo)	Fired when a joined user existed the session. Return true will override the default behavior.
ongivecontrol()	Fired when current user can control the session. Return true will override the default behavior.
ontakebackcontrol()	Fired when current user cannot control the session anymore
onrequirecontrol()	Fired when another require current user to control the session.
onremoteappstart(e)	Fired after a RemoteApp was started, you can get the RemoteApp id from e.id which is like: {1AC14E77-02E7-4E5D-B744-2EB1AE5198B7}\mspaint.exe
onactivity(input)	Fired after a client activity was sent to the server (Mouse, keyboard events). Input (string) is the mouse and keyboard event. You can use writeRawInput to send them to the RDP host.
onurlredirection(URL)	Fired when an URL was copied on remote computer
ontitlechange(title)	Fired when client before change the browser title. You can return a new title in this event.
onsessionstart(sessionInfo)	sessionInfo is same as the sessionInfo property
onservercopy(strValue)	Fired when copying from remote to local. strValue is in "mimeType;value" format, for example: "text/plain;textCopied".
onrequestcredential(info)	Fired when wrong credential was given and credSSP is enabled on gateway). Return true will override the default behavior. Info object has following properties: domain: server's FQDN domain name nbDomain: server's NetBIOS domain name nbComputer: server's NetBIOS computer name

Name	Description
onprintingready(printJob)	Return true will override the default behavior. printJob has following properties: link: the PDF file link printerName: target printer name printerDriver: target printer driver name
onresolutionchange(width, height)	Fired when resolution of RDP session changes.
onnoresponse()	Need to be used with startPing(interval, missTimes)
onLockKeyState({"capsLock": capsLock, "numLock": numLock, "scrollLock": scrollLock)	

6.2.5 – Extend RDP: Virtual Channel and Dynamic Virtual Channel

You can create multiple virtual channels and dynamic virtual channels on client side using JavaScript (you can only create one VC before 4.0):

```
var r = new svGlobal.Rdp(protocol + gw + "/RDP?"+ s, w, h, server_bpp);
var vc = new r.VirtualChannel(); //Use r.DynamicChannel to create a dynamic virtual channel
vc.name = "CUST";
vc.process = function(buffer){
  console.log(buffer.getByte());
  console.log(buffer.getLittleEndian16());
};
vc.onopen = function(){
  var data = new Array(7);
  var rb = new RdpBuffer(data, 0, 7);
  rb.setByte(1);
  rb.setLittleEndian16(345);
  rb.setLittleEndian32(567);
  rb.markEnd();
  vc.send(rb);
};
r.addChannel(vc);
```

Virtual Channel is used to communicate with RDP host. You also need to write a plugin for the RDP host. Please check followings for more information:

- http://msdn.microsoft.com/en-us/library/aa383546(v=vs.85).aspx
- http://www.codeproject.com/Articles/16374/How-to-Write-a-Terminal-Services-Add-in-in-Pure-C

6.2.6 – Extend Gateway: Gateway Channel

You can create multiple gateway channels to create a communication layer between client browser and the gateway:

```
var gvc = new r.GatewayChannel();
gvc.name = "gwc";
gvc.process = function(buffer){
  console.log(buffer.getByte());
  console.log(buffer.getLittleEndian16());
};
gvc.onopen = function(){
  var data = new Array(7);
  var rb = new RdpBuffer(data, 0, 7);
  rb.setByte(3);
  rb.setLittleEndian16(45);
  rb.setLittleEndian32(678);
  rb.markEnd();
  gvc.send(rb);
};
r.addGatewayChannel(gvc);
```

On gateway side, you class must extend com.toremote.gateway.plugin.AbstractGatewayChannel and register it with the same name using HandlerManager.registerChannel(). Please check the plugin example for more information.

6.3 – Plug-ins (server side Java API)

Your plug-in must implement com.toremote.gateway.plugin.ManagerInterface. With the plugin, you can do authentication integration, session querying and reporting, RemoteApp management and RDP virtual channel extensions, player integration, new websocket protocol handler etc. Please download our plugin example project for more information:

http://www.remotespark.com/Plugin.zip

6.4 – HTTP Request API (server side)

You can use HTTP request to create server, symlink dynamically if you don't want to write a plug-in.

Create servers on gateway:

http://gatewayAddress/SERVER?id=serverId&displayName=Name&server=hostName&gatewayPwd=pas swordInGateway.conf&...

gatewayPwd is hexadecimal MD5 hash of the password which is configured in gateway.conf. Please check 5.2.1 for other parameters you can use.

Create symlink on gateway:

http://gatewayAddress/SYMLINK?symlink=symlinkId&server=existingServerId&validTime=20m&gatewayPwd=passwordInGateway.conf&...

You can also use "validFrom", "validTo" parameters. Please check http://www.remotespark.com/view/doc/com/toremote/gateway/connection/SymLink.html for more information.

To delete a server or symlink, add "&action=delete" to the URL; To update a server or symlnk, add "&action=update" to the URL.

HTTP request will return status code 200 if operation succeeded, 500 if operation failed.

For more details, please check the Integration Guide.