



# SparkView

集成指南

**1.0**

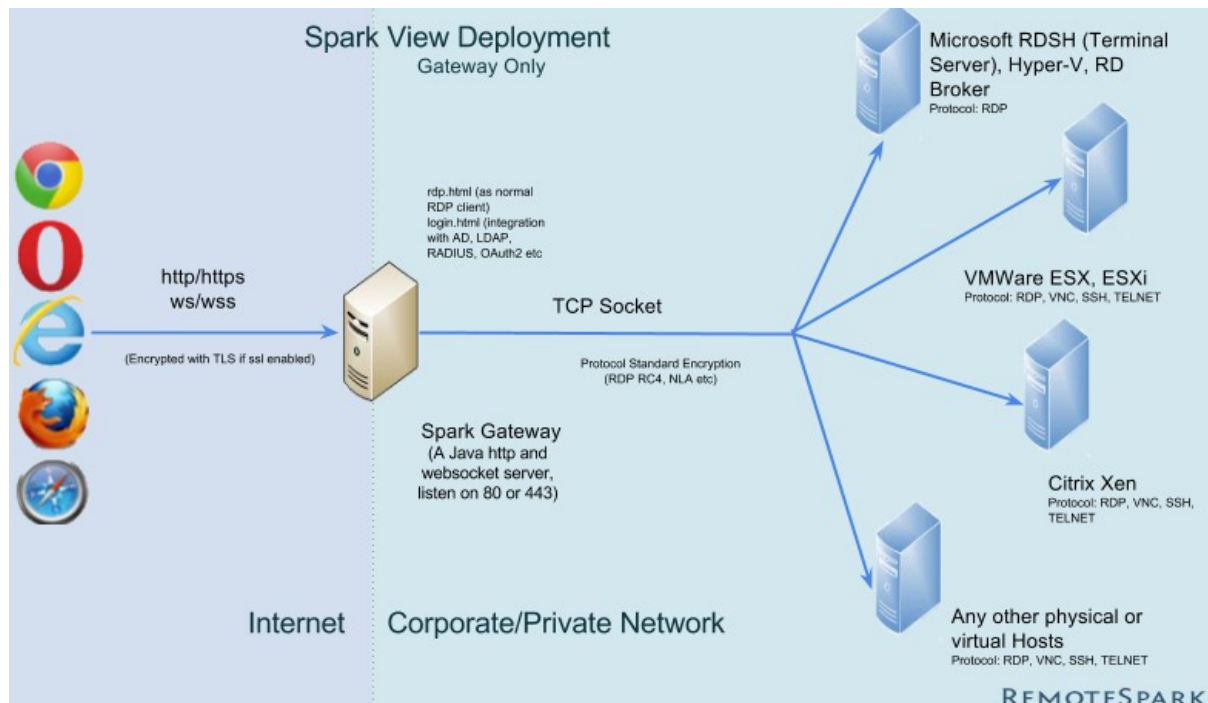
**8 月 20, 2016**

## 目录

内容.....	Error! Bookmark not defined.
1. 部署 .....	3
1.1. 仅安装 SparkGateway .....	3
1.2. SparkGateway 和反向代理（reverse proxy），负载均衡器（load balancer）或者 VPN .....	4
2. 无需编程的集成方法.....	4
2.1. 微软远程桌面集成 Web Access Portal (Microsoft RD Web Access portal).....	4
2.2. 活动目录（Active Directory）和 LDAP 集成.....	5
2.3. Radius 集成 .....	6
2.4. OAuth2 集成 .....	7
2.5. 使用 HTTP Basic Authentication 完成单点登录：Single Sign-On (SSO) .....	8
3. 客户端（浏览器）集成.....	8
3.1. 仅使用 8 行代码从你的网页上创建一个远程连接.....	8
3.2. 使用对象或 cookie 来传递参数.....	10
3.3. 资源重定向参数 .....	12
3.4. 在连接时启动一个程序(as Shell).....	21
3.5. 运行 RemoteApp.....	21
3.6. 虚拟通道（Virtual Channel）及动态虚拟通道（Dynamic Virtual Channel）扩展开发.....	23
3.7. Gateway 扩展通道（Gateway Channel） .....	23
3.8. 客户端配置文件 .....	24
4. 服务器端（SparkGateway）集成.....	24
4.1. HTTP API.....	24
4.2. 插件 Plug-in.....	27
4.2.1. 在 Eclipse 内运行 gateway 和你的 plug-in.....	28
4.2.2. 握手插件 Handshake plug-in.....	31
4.2.3. 部署你的插件.....	33

## 1. 部署

### 1.1. 仅安装 SparkGateway



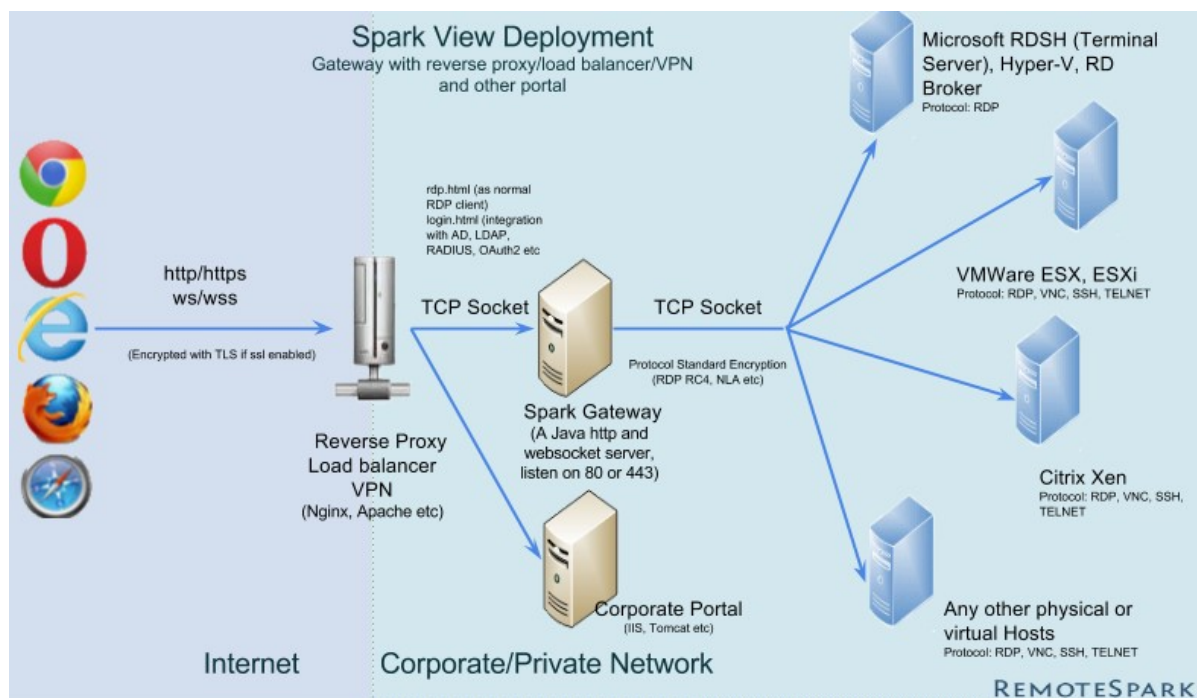
优点:

- 简单快速，安装 Java 和 Spark View 仅需 15 分钟。
- 可以如同普通 RDP/VNC/SSH/Telnet 客户端一样使用 Gateway, 并建立服务器黑名单/白名单
- 内置活动目录 ( Active Directory ) , LDAP, RADIUS, OAuth2, MS RD Web Feed 等集成
- 无需另外安装 Web 服务器

缺点:

- 如果需要动态网页 (dynamic web page) 的话，你需要为 SparkGateway 编写一个 Plug-in.

## 1.2. SparkGateway 和反向代理（reverse proxy），负载均衡器（load balancer）或者 VPN



优点:

- 高可用性（High availability）
- 无需暴露 SparkGateway 端口到外网
- 无缝集成到现有环境，从用户角度来看，最终产品界面和使用流程基本不变。
- 已测试通过的产品：Nginx, Apache 反向代理; F5, Cisco, Juniper, Dell, Check Point, Array Networks 等 VPN 产品
- 可以沿用已有服务器端编程技术：JSP, ASP.NET, PHP 等等

## 2. 无需编程的集成方法

### 2.1. 微软远程桌面集成 Web Access Portal (Microsoft RD Web Access portal)

如果你添加了 Microsoft RD Web Access portal 角色（Role），你可以使用

<https://MyRDPortal/RdWeb> 访问你的 IIS 服务器并进行远程连接。那么你可以在 gateway.conf 做如下设置:

webfeed = <https://MyRDPortal/RDWeb/feed/webfeed.aspx>

directoryIndex = login.html

这样你的用户，就可以访问 `http(s)://SparkGatewayAddress/` 来进行基于 HTML5 的远程连接

#### 检查清单:

- 使用你的浏览器来验证 `webfeed` 地址是否正确工作，如果正确的话，服务器会返回一个 `cookie` 或 `xml` 文件
- 清楚浏览器缓存如果浏览器没有返回你需要的页面
- 确保 IIS 正在使用 NTLM authentication: [https://technet.microsoft.com/en-us/library/cc754628\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc754628(v=ws.10).aspx)
- 确保 `RDWeb/Pages` 和 `RDWeb/Feed` on IIS 正在使用 “Windows authentication”，而不是 “Forms Authentication”。

## 2.2. 活动目录 (Active Directory) 和 LDAP 集成

如果你的所有用户是域用户，那么你可以创建如下 JSON 文本文件来设定你的用户来源。注意文件的编码是不带字节顺序标记位的 UTF-8 (UTF-8 without Byte Order Mark):

```
{
  "source": {
    "type": "AD",
    "properties": {
      "server": "ADServerAddress"
    }
  }
}
```

保存并命名为 `users.json` 或其他名称，在 `gateway.conf` 中指定该文件的位置:

```
user = C:\\workspace\\data\\users.json
```

如果你在连接一个 LDAP 服务，请设定 `type` 为 “LDAP”

你也可以在 `users.json` 中为不同的用户制定不同的 AD/LDAP 的服务器

```
{
  "users": [
    {
      "name": "user1",
```

```

    "password": "user1",

    "servers": [

        "RdpServer1",

        "TEST",

        "Excel 2010"

    ],

    "isDomainUser": true,

    "transferCredential": true,

        domainServer: "serverAddr"

    }

}

```

你可以使用另外一个文件来设定客户准备远程连接的服务器。将该文件保存并命名为 `servers.json` 或其他名称，在 `gateway.conf` 中制定该文件位置：

```
server = C:\\workspace\\data\\servers.json
```

## 2.3. Radius 集成

```

{

    "source":{

        "type": "RADIUS",

        "properties": {

            "server": "192.168.12.128",

            "port": "1812",

            "accountingPort": "1813",

            "sharedSecret": "test123"

            "timeout": "60000",

            "retryCount": "3"

        }

    }

}

```

```
}
```

```
}
```

- 确保在 RADIUS 服务器上设置 Spark Gateway 的 IP 地址 为被允许的客户端
- 如果你的 RADIUS 服务正在使用多重验证（multi factor authentication），比如微软的 Azure MFA，确保 timeout 值最小为 60000 毫秒。

## 2.4. OAuth2 集成

你需要生成一个 OAuth2 服务提供者的 JSON 配置文件，示例如下：

```
{
  "providers": [{
    "name": "Google",
    "client_id": "650561938988-
t2r66k1ms3hpoi3k1e2g7l2adlarau8s.apps.googleusercontent.com",
    "client_secret": "-D-nhxWn2E97tZWwLg5lQ6Ak",
    "request_uri": "https://accounts.google.com/o/oauth2/auth",
    "redirect_uri": "http://localhost/oauth2callback",
    "access_token_uri": "https://accounts.google.com/o/oauth2/token",
    "scope": "openid email"
  },
  {
    "name": "Live",
    "client_id": "0000000040133A31",
    "client_secret": "p9WwBr2Pyrq6mtaeZCwTSwqbIF39Br3Z",
    "request_uri": "https://login.live.com/oauth20_authorize.srf",
    "redirect_uri": "http://www.remotespark2.com/oauth2callback",
    "access_token_uri": "https://login.live.com/oauth20_token.srf",
    "scope": "wl.emails",
```

```

        "profile_uri": "https://apis.live.net/v5.0/me"
    }
}
]
}

```

然后，在 gateway.conf 中制定该文件位置：

oauth2 = [\\user\\local\\bin\\SparkViewGateway\\oauth2.json](#)

更多信息请查看 login.html 的源文件。

## 2.5. 使用 HTTP Basic Authentication 完成单点登录：Single Sign-On (SSO)

你可以在 gateway.conf 中启用 HTTP Basic Authentication。那么以后的任何 RDP 连接就会使用来自 HTTP Basic Authentication 的用户名和密码。该方法可以用来和 VPN 以及其他 web 服务器进行单点登陆集成。

## 3. 客户端（浏览器）集成

通过使用 Spark View 的 JavaScript 库，你可以从任何网页上创建一个远程连接，无论这个网页是来自 web 服务器还是一个本地网页文件。

### 3.1. 仅使用 8 行代码从你的网页上创建一个远程连接

保存下列内容到 **tutorial1.page.js**：

```

window.onload = function() {

    var gateway = '192.168.12.111',//改变次地址为你的 SparkGateway 服务器地址

    server = '192.168.12.117',// 改变次地址为你的 RDP 服务器地址

    url = 'ws://' + gateway + '/RDP?server=' + server +
    '&user=myUserName&pwd=myPassword';//改变用户名和密码

    var r = new svGlobal.Rdp(url);

    r.addSurface(new svGlobal.LocalInterface());

    r.run();

};

```

保存下列内容到 **tutorial1.html**：

```
<!doctype html>
```



```
<html>

<head>

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Spark View (RDP)</title>

<meta name="viewport" content="width = device-width, initial-scale = 1.0, user-scalable = yes,
minimum-scale = 0.1, maximum-scale = 8" />

<meta name="apple-mobile-web-app-capable" content="yes" />


<link rel="stylesheet" href="../hi5.css" />

<link rel="stylesheet" href="../rdp.css" />


<script type="text/javascript" src="../appcfg.js"></script>
<script type="text/javascript" src="../resource.js"></script>
<script type="text/javascript" src="../hi5_min.js"></script>
<script type="text/javascript" src="../surface_min.js"></script>
<script type="text/javascript" src="../rdp_min.js"></script>
<script type="text/javascript" src="tutorial1.page.js"></script>

</head>


<body>

    <div>

        <canvas id="remotectrl"></canvas>

    </div>

</body>

</html>
```

你可以直接双击该文件用浏览器打开它，或者把该文件放到你的 web 服务器上访问。

#### 检查清单:

- 确保使用<!doctype html> 来启用 HTML5.
- 使用<meta http-equiv="X-UA-Compatible" content="IE=edge">建议 Internet Explorer 使用最新的 edge 引擎。
- 如果你的用户在使用触摸设备，确保使用 Viewport meta 标记。
- “apple-mobile-web-app-capable” 标记允许用户为该网页创建快捷方式
- 为了创建一个 RDP 远程连接，你需要使用下列 JavaScript 文件： appcfg.js (客户端属性配置), resource.js (语言资源), hi5\_min.js (通用工具), surface\_min.js (用户界面), rdp\_min.js (RDP 协议处理)，另外还需要包括 hi5.css 和 rdp.css，以及一些用到图像文件(\*.png)。VNC 需要 vnc\_min.js 和 vncutil\_min.js。SSH 需要 ssh\_min.js。Telnet 需要 telnet\_min.js。
- 可以通过修改 hi5.css 和 rdp.css，或者增加自己的样式表(css)来修改默认样式。
- 一个 id 为 “remotectrl” 的 canvas 用来显示远程桌面。如果你使用了其他的 id，你需要在创建 LocalInterface 时指定该 canvas: new svGlobal.LocalInterface('MyCanvasId')
- 在你创建一个远程连接之前，确保所有用到的 JavaScript 文件已经加载完毕。比较稳妥的办法是在 window.onload 事件里面重建新的远程连接。
- SparkView 客户端会自动维护 canvas 的大小，不要在你的代码里直接修改其大小，更不要使用 css 来设置及大小。

#### 排错:

- 检查你的浏览器的 JavaScript 控制台（console）日志（F12 或从菜单里打开开发者工具）
- 检查你的 SparkGateway 日志（安装目录/logs/\*.\*)
- 检查 Internet Explorer 是否使用了兼容模式（legacy mode），确保其使用 Edge 模式。

### 3.2. 使用对象或 cookie 来传递参数

你也可以使用对象或者 cookie 以及 Rdp2 来设置远程连接需要的参数:

#### 对象参数和 Rdp2:

```
window.onload = function() {  
  
    var parameters = {gateway: '192.168.12.111',  
  
                      server: '192.168.12.117',
```

```

        user: 'vmuser',

        pwd: 'password'};

var r = new svGlobal.Rdp2(parameters);

r.addSurface(new svGlobal.LocalInterface());

r.run();

};

```

### **Cookie 参数和 Rdp2:**

```

window.onload = function() {

    document.cookie = 'gateway=192.168.12.111';

    document.cookie = 'server=192.168.12.117';

    document.cookie = 'user=vmuser';

    document.cookie = 'pwd=password';

    var r = new svGlobal.Rdp2();

    r.addSurface(new svGlobal.LocalInterface());

    r.run();

};

```

Cookie 参数只能用于 web 服务器上的网页，不能用于本地 web 文件

### **HTTP 头（Header）参数:**

如下参数也可以设置在 HTTP 头（Header）里面：

gw\_server, gw\_port, gw\_symlink, gw\_user, gw\_pwd

通常你可以在 VPN 或者反向代理里面设置 HTTP 头参数。

### **最佳实践:**

不要在你的网页里面直接加入 JavaScript 代码，最好把所有的 JavaScript 保存在一个独立的 js 文件里面。因为：

- Content-Security-Policy HTTP 头（header）可能会禁止混用网页和脚本代码。

- Chrome Web Store App 不允许此类混用
- 你将不能在浏览器开发工具里面格式化(Pretty print)你的代码。如果你的代码是压缩过的，格式化代码可以让代码调试更方便。
- 在调试时，你将不能在浏览器开发工具里面直接修改你的代码。
- 最好把 canvas 元素放在一个 DIV 里面。
- 你可以在一个网页里面使用多个 canvas 元素（使用 div 或 iframe）来创建多个远程连接。

### 3.3. 资源重定向参数

SparkView 默认禁止了所有的资源重定向，你需要通过参数来启用它们。

资源重定向可以在客户端（浏览器端）设置，也可以在服务器端（SparkGateway）通过 servers.json 文件来设置，服务器端的设置有更高优先级。

资源重定向	对应参数及示例
剪贴板	mapClipboard=on
磁盘	mapDisk=on  还需要在 gateway.conf 中设置 tmpdir。详见管理员手册
打印机	mapPrinter=on  printer=你想要的打印机名称  你也可以在 gateway.conf 里面统一设置 printerDriver （打印机的驱动名称）和 printer （名称）  你需要在 SparkGateway 所在的机器(注意不需要在 RDP 服务器上安装)上安装一个 PostScript 或 PCL 到 PDF 的转换程序，并且在 gateway.conf 里配置改转换程序的位置和所需参数：  converter = D:\\Programs\\gs\\gs9.16\\bin\\gswin32c.exe  arguments = -dBATCHE -dNOPAUSE -dUseCIEColor -dPDFSETTINGS=/printer -sDEVICE=pdfwrite -q -sOutputFile=%1 %2

	详见管理员手册
声音播放	<b>playAudio=0</b>  <b>soundPref=0:</b> 声音质量低, 占用很少带宽  <b>soundPref=1:</b> 声音质量高, 占用较多带宽
声音录制 (麦克风)	<b>audioRecord=on</b>
时区	<b>timezone= encodeURIComponent ('(GMT-07:00) Mountain Standard Time')</b>

更多参数介绍请参见管理员手册。

启用剪贴板的例子:

```
var r = new svGlobal.Rdp('ws://gatewayAddr/RDP?server=192.168.12.117&mapClipboard=on');
```

### 常用参数

参数	值
<b>port</b>	整形, 可选项, 欲连接的远程服务器端口, RDP 默认为 3389, VNC 默认为 5900, SSH 默认为 22, Telnet 默认为 23
<b>user</b>	字符串, 用户名
<b>pwd</b>	字符串, 密码
<b>domain</b>	字符串, 域名
<b>keyboard</b>	整形, 键盘布局代码, 默认为 0x409 (美国)
<b>width</b>	整形, 远程会话屏幕宽度
<b>height</b>	整形, 远程会话屏幕长度
<b>vmid</b>	Hyper-V 虚拟机 GUID, 比如: B3D5444C-2611-405A-9CA0-7AA8DA94DF0B, 用来连接到 Hyper-V 控制台 (console)
<b>minWidth, minHeight</b>	最小宽度和长度。如果屏幕分辨率过小, 某些程序可能不会正常工作, 所以可是设置此参数避免类似情况

## 打印机驱动问题:

SparkGateway 会自动在 RDP 服务器上添加一个重定向的打印机，并且使用“MS Publisher Imagesetter”作为默认的打印机驱动，因为所有的 Window 版本都包含此驱动。但是该驱动的某些版本在打印大文件时存在输出数据过大的问题(有可能产生上 GB 的数据)，所以最好在 gateway.conf 设置成其他的驱动，比如：

printerDriver = HP Color LaserJet 8500 PS

确保你的 RDP 服务器上存在该驱动，你可以通过添加一个本地打印机的办法来检查你可以用的驱动：

← Add Printer

Find a printer by other options

☐ My printer is a little older. Help me find it.

☐ Select a shared printer by name

Browse...

Example: \\computername\printername or  
http://computername/printers/printername/.printer


☐ Add a printer using a TCP/IP address or hostname

☐ Add a Bluetooth, wireless or network discoverable printer

☒ Add a local printer or network printer with manual settings

Next Cancel

✕

←  Add Printer

Choose a printer port

A printer port is a type of connection that allows your computer to exchange information with a printer.

☒ Use an existing port:

LPT1: (Printer Port) ▾

☐ Create a new port:

Type of port: Local Port ▾

Next

Cancel

Remote Spark Corp. Page 15 of 35



←  Add Printer

## Install the printer driver



Choose your printer from the list. Click Windows Update to see more models.

To install the driver from an installation CD, click Have Disk.

Manufacturer	Printers
HP	HP Color LaserJet 9500 PCL6 Class Driver
InfoPrint	HP Color LaserJet 9500 PS Class Driver
infotec	HP Color LaserJet A3/11x17 Hardware-Copy PCL6 Class Driver
KONICA MINOLTA	HP Color LaserJet A3/11x17 PCL6 Class Driver



This driver is digitally signed.

[Tell me why driver signing is important](#)

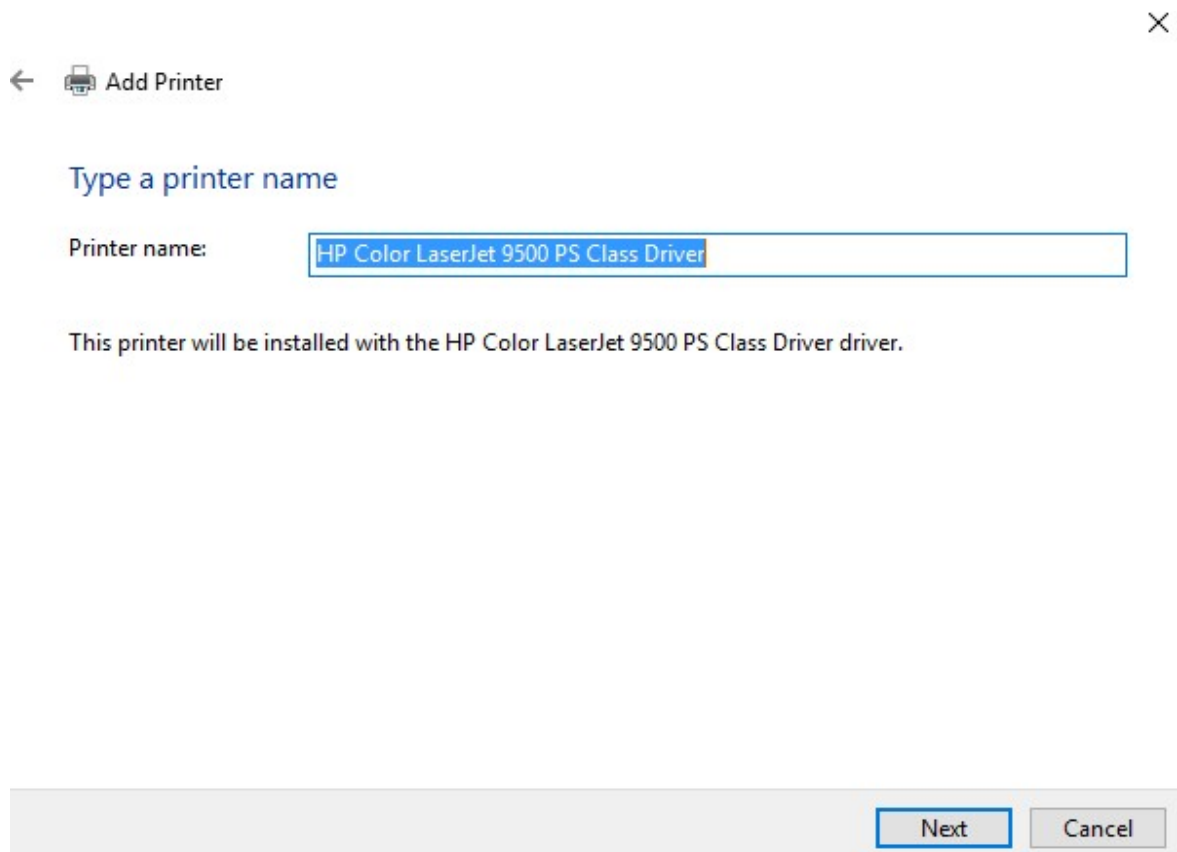
Windows Update

Have Disk...

Next

Cancel





复制此打印机名称（驱动名）并点击“取消”。你不需要真正安装此打印机。然后在 `gateway.conf` 设置打印机的驱动。

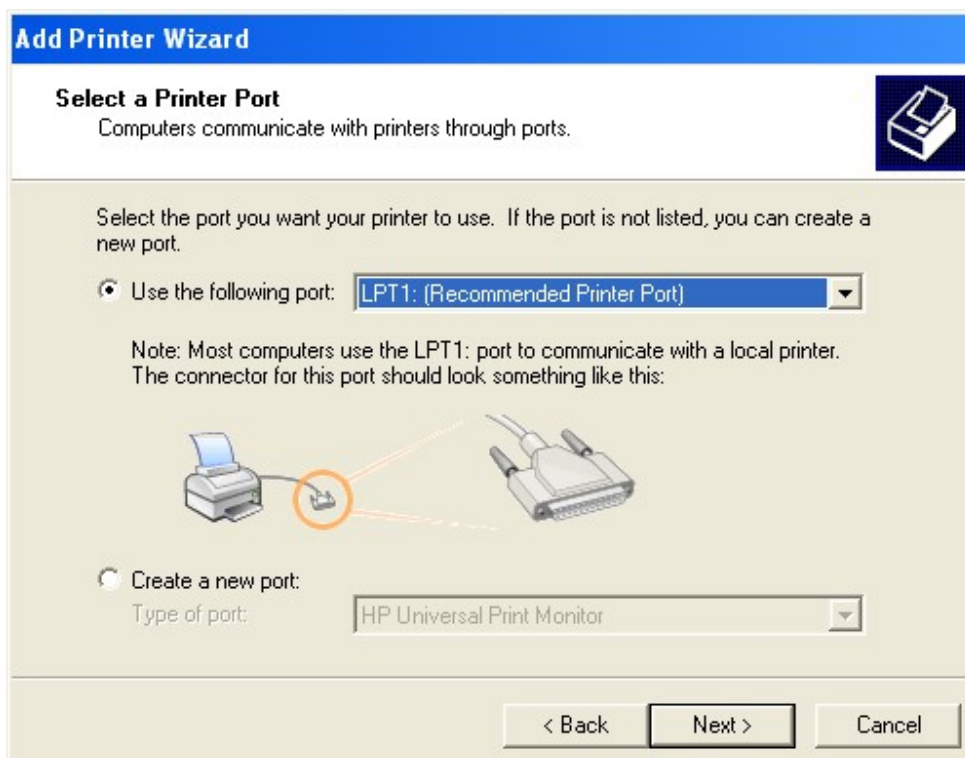
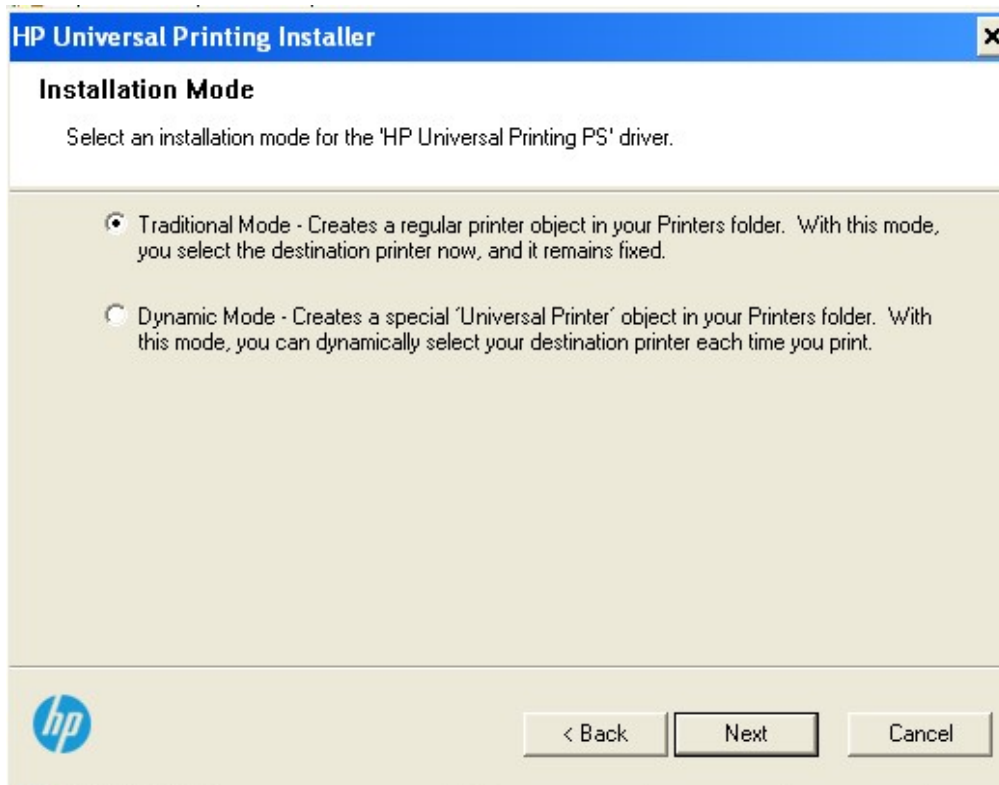
确保你选中的打印机是 **PostScript** 或 **PCL** 打印机(名称内包含 **PS** 或 **PCL**)。如果是 **PostScript** 打印机，你需要在 `gateway.conf` 中设置 **PostScript** 到 **PDF** 的转换程序。如果是 **PCL** 打印机，你则需要设置 **PCL** 到 **PDF** 的转换程序。

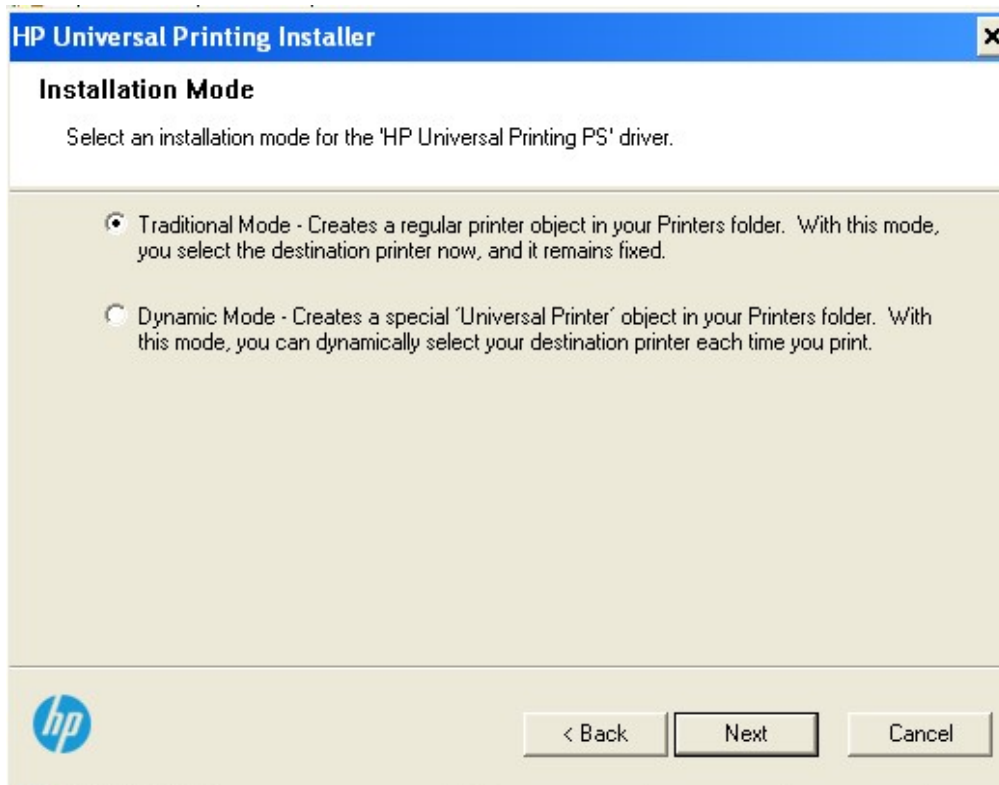
你也可以设置一个 **PDF** 打印机的驱动，这样就无需在 `gateway.conf` 中设置转换器，但目前只有“**Microsoft Print to PDF**”驱动可以在 **RDP** 环境中工作。

你也可以安装使用 **HP Universal PostScript** 打印机驱动：

下载：

[http://h20331.www2.hp.com/hpsub/cache/343033-0-0-225-121.html?jumpid=ex\\_r2845\\_go/upd](http://h20331.www2.hp.com/hpsub/cache/343033-0-0-225-121.html?jumpid=ex_r2845_go/upd)






Add Printer Wizard

### Local or Network Printer

The wizard needs to know which type of printer to set up.




Select the option that describes the printer you want to use:

☒ Local printer attached to this computer

☐ Automatically detect and install my Plug and Play printer

☐ A network printer, or a printer attached to another computer



To set up a network printer that is not attached to a print server, use the "Local printer" option.

< Back


Next >

Cancel

Add Printer Wizard

### Select a Printer Port


Computers communicate with printers through ports.



Select the port you want your printer to use. If the port is not listed, you can create a new port.

☒ Use the following port: LPT1: (Recommended Printer Port)

Note: Most computers use the LPT1: port to communicate with a local printer. The connector for this port should look something like this:



☐ Create a new port:

Type of port: HP Universal Print Monitor

< Back

Next >

Cancel

在 gateway.conf 设置如下打印机驱动:

```
printerDriver = HP Universal Printing PS
```

### 3.4. 在连接时启动一个程序(as Shell)

你需要设置如下参数：

```
startProgram=shell
```

```
command=encodeURIComponent('C:\\apps\\notepad.exe')
```

```
directory= encodeURIComponent('C:\\apps\\')
```

注意该模式下，你只能在一个会话内启动一个程序。如果你需要启动多个程序的话，可以考虑从改程序内启动其他程序或者使用 RemoteApp。

#### 检查清单：

你可能需要在 RDP 服务器上允许客户运行任意程序，详见管理员手册

### 3.5. 运行 RemoteApp

如果你想在在一个会话里面运行一个或多个 RemoteApp 的话，你可以使用下列参数：

```
startProgram=app
```

```
exe=||WINWORD
```

```
directory= encodeURIComponent('C:\\apps\\')
```

在新窗口内运行一个 RemoteApp :

```
function startRemoteApp(remoteApp, args, dir, url){  
  
    var r = svManager.getInstance() || new svGlobal.Rdp(url); //reuse existing session if available  
  
    function onSurfaceReady(surface){  
        r.addSurface(surface);  
  
        if (r.running()){  
            r.startApp(remoteApp, args, dir);  
        }else{  
            r.run();  
        }  
    }  
  
};
```

```

window.svOnSurfaceReady = onSurfaceReady;

var rail = window.open('rail.html');

rail.svOnSurfaceReady = onSurfaceReady;
}

```

```

window.onload = function() {

var gateway = 'w-think',//change this to your Spark gateway address

    server = '192.168.12.132',//change this to your RDP server address

    user = 'vmuser',

    password = 'password',

    remoteApp = '| |notepad',

    args = "",

    dir = "",

    url = 'ws://' + gateway + '/RDP?server=' + server + '&startProgram=app' + '&exe=' +
encodeURIComponent(remoteApp) + '&user=' + user + '&pwd=' + password;

    startRemoteApp(remoteApp, args, dir, url);

};

```

你也可以在一个窗口内启动多个 RemoteApp，这种情况下，建议在 appcfg.js 中设置如下：

```
noMinimize: false,
```

#### 检查清单：

- 确保使用 RemoteApp 的别名而不是路径名，并且需要在名称前有“||”，比如：  
||WINWORD
- 确保你先在服务器上发布你需要的 RemoteApp。

### 3.6. 虚拟通道（Virtual Channel）及动态虚拟通道（Dynamic Virtual Channel）扩展开发

SparkView JavaScript 客户端支持 RDP 虚拟通道和动态虚拟通道的扩展开发。你可以在客户端利用 JavaScript 创建多个你需要的 VC 和 DVC:

```
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);
```

VC 和 DVC 是用来和 RDP 服务器通信的。所以对于自定义的扩展 VC 和 DVC，你同时需要在 RDP 服务器上编写并安装相应的 VC 和 DVC 扩展插件。更多信息请查看微软相关资料：

[http://msdn.microsoft.com/en-us/library/aa383546\(v=vs.85\).aspx](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>

### 3.7. Gateway 扩展通道（Gateway Channel）

Gateway Channel 是客户端用来和 SparkGateway 进行通信的（不涉及 RDP 服务器）。利用 Gateway Channel，你可以从 Gateway 上获取需要的信息或发送客户端信息给 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);
```

在 Gateway 一端，你同样需要用 java 编写一个扩展插件。你的扩展类需要继承 `com.toremote.gateway.plugin.AbstractGatewayChannel` 类，并且使用 `HandlerManager.registerChannel()` 来注册这个新的扩展插件。请参考我们的插件示例：

<http://remotespark.com/Plugin.zip>

### 3.8. 客户端配置文件

你可以使用 `appcfg.js` 来集中设置客户端属性，比如：

`displayMsg: false`,//禁止显示所有的通知信息

`reconnectOnResize: false`,//禁止自动调整会话屏幕分辨率（如果浏览器窗口大小变动的話）。

`windowState: 0`,//禁止自动最大化 RemoteApp 窗口来适应浏览窗口大小

`hideLogin: true`,//启动 RemoteApp 时，不显示登陆过程

`noMinimize: false`,//允许最小化 RemoteApp 窗口。

更多设置，请参靠 `appcfg.js` 文件。

## 4. 服务器端（SparkGateway）集成

### 4.1. HTTP API

你可以使用 HTTP 请求 API 来动态在 Gateway 上创建 RDP/VNC/SSH/Telnet 服务器设置，symlink 等。这样你无须为 Gateway 编写插件也可以实现某些功能集成。

#### 创建服务器：

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

`gatewayPwd` 是密码的 16 进制 MD5 哈希码。该密码需要在 `gateway.conf` 中设置。你可以使用此链接得到你想要的 MD5 哈希码：<http://www.miraclesalad.com/webtools/md5.php>

#### 获取服务器列表：

<http://w-think/SERVER?action=list&gatewayPwd=21232f297a57a5a743894a0e4a801fc3&id=ATPlus>

如果指定了服务器 id，该请求会返回 JSON 格式的该服务器的设置：

```
{"id":"ATPlus","displayName":"ATPlus","server":"cloud.thinrdp.net","shadowing":false,"protocols":["rdp"],"icon":"kbd.png","rdp":{"username":"demo","password":"demo"}}
```

如果没有指定服务器 id，该请求会返回所有服务器列表：



```
{
  "display" : true,
  "type" : "NORMALLIST",
  "cols" : [{"name" : "id"},
             {"name" : "displayName"},
             {"name" : "server"},
             {"name" : "remoteProgram"},
             {"name" : "command"}],
  "rows" :
    [
      ["ATPlus", "ATPlus", "192.168.12.117", "", ""],
      ["WordPad", "WordPad", "192.168.0.118", "", "wordpad.exe"]
    ]
}
```

### 创建一个 symlink:

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

symlink 类似于一个指向服务器的快捷方式，但是你可以设定该快捷方式的有效时间及密码等。更多参数信息请参看

<http://www.remotespark.com/view/doc/com/toremote/gateway/connection/SymLink.html>

删除一个服务器或 symlink，请在 URL 中添加 "&action=delete" 参数；更新的话，请在 URL 中添加 "&action=update" 参数。

### 获取 symlink 列表:

<http://w-think/SYMLINK?action=list&gatewayPwd=21232f297a57a5a743894a0e4a801fc3&symlink=212a155e-e951-40db-95ea-177183174fa7>

如果指定了 symlink id 的话，它会返回该 symlink 的设置:

```
{"id":"212a155e-e951-40db-95ea-177183174fa7","resourceId":"169.254.146.243","validFrom":"Aug 26, 2015 12:00:00 AM","parameters":""}
```

如果没有制定 symlink id 的话，它会返回所有 symlink 列表:

```
{"cols":[{"name":"id"},
```

```

        {"name":"resourceId"},
        {"name":"password"},
        {"name":"validFrom"},
        {"name":"validTime"},
        {"name":"validTo"},
        {"name":"parameters"},
        {"name":"comment"}],
    "rows":[["212a155e-e951-40db-95ea-177183174fa7", "169.254.146.243", "", 1440568800000, "", 0, "",
""],
        ["c5c6bc9d-f8a7-42ca-af9c-bd28c86adab4", "169.254.146.243", "", 1452544860000, "",
0, "", ""]]
}

```

获取连接会话列表:

<http://w-think/SESSION?action=list&gatewayPwd=21232f297a57a5a743894a0e4a801fc3&id=a29575a9-08c2-4162-9bfb-4876820953db>

如果指定了会话 id (id)的话, 它会返回该会话的相关信息:

```

{"id":"a29575a9-08c2-4162-9bfb-4876820953db", "server":"192.168.12.118", "clientIp":"169.254.84.132", "clientAgent":"Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.111 Safari/537.36", "thumbnail":"", "startTime":"Tue Jan 26 15:38:55 MST 2016", "numericId":621779525, "user":"", "domain":"", "rdpId":1, "startTime2":1453847935826}

```

如果 id 没有指定的话, 它会返回所有会话列表:

```

{"cols":[{"name":"id"},
        {"name":"server"},
        {"name":"clientIp"},
        {"name":"clientAgent"},
        {"name":"startTime"},
        {"name":"numericId"},

```

```

        {"name":"user"},

        {"name":"domain"},

        {"name":"join"},

        {"name":"protocol"},

        {"name":"symlink"},

        {"name":"thumbnail"}],

    "size":1,

    "rows":[["a29575a9-08c2-4162-9bfb-4876820953db","192.168.12.118","169.254.84.132","Mozilla\5.0
(Windows NT 10.0; WOW64) AppleWebKit\537.36 (KHTML, like Gecko) Chrome\47.0.2526.111
Safari\537.36","Tue Jan 26 15:38:55 MST 2016",621779525,"","","RDP","",""]]

}

```

如果 HTTP 请求成功，会返回状态码 200；如果失败的话，会返回状态码 500。

**更新授权文件（license file）：**

http://w-  
think/CONTROL?licenseFile=base64encodedLicenseFile&gatewayPwd=21232f297a57a5a743894a0e4a8  
01fc3

确保你使用 `encodeURIComponent` 对 Base64 值进行编码。比如：  
`encodeURIComponent(base64encodedLicenseFile)`

## 4.2. 插件 Plug-in

Gateway 是一个多线程程序，所以在为 Gateway 编写插件必须注意以下事项：

- 你的插件是线程安全的。
- 如果某个操作需要较长时间，考虑在一个新的线程内运行，不然有可能导致网络超时错误。
- 使用 `Collections.synchronizedList()`, `Collections.synchronizedMap`, `ConcurrentHashMap` 等线程安全的类。
- 尽量避免使用匿名类，如果不当使用的话，可能会导致内存泄漏。
- 尽量使你的插件简单，最好不保存中间状态。
- 过度的日志可能会成为性能瓶颈。

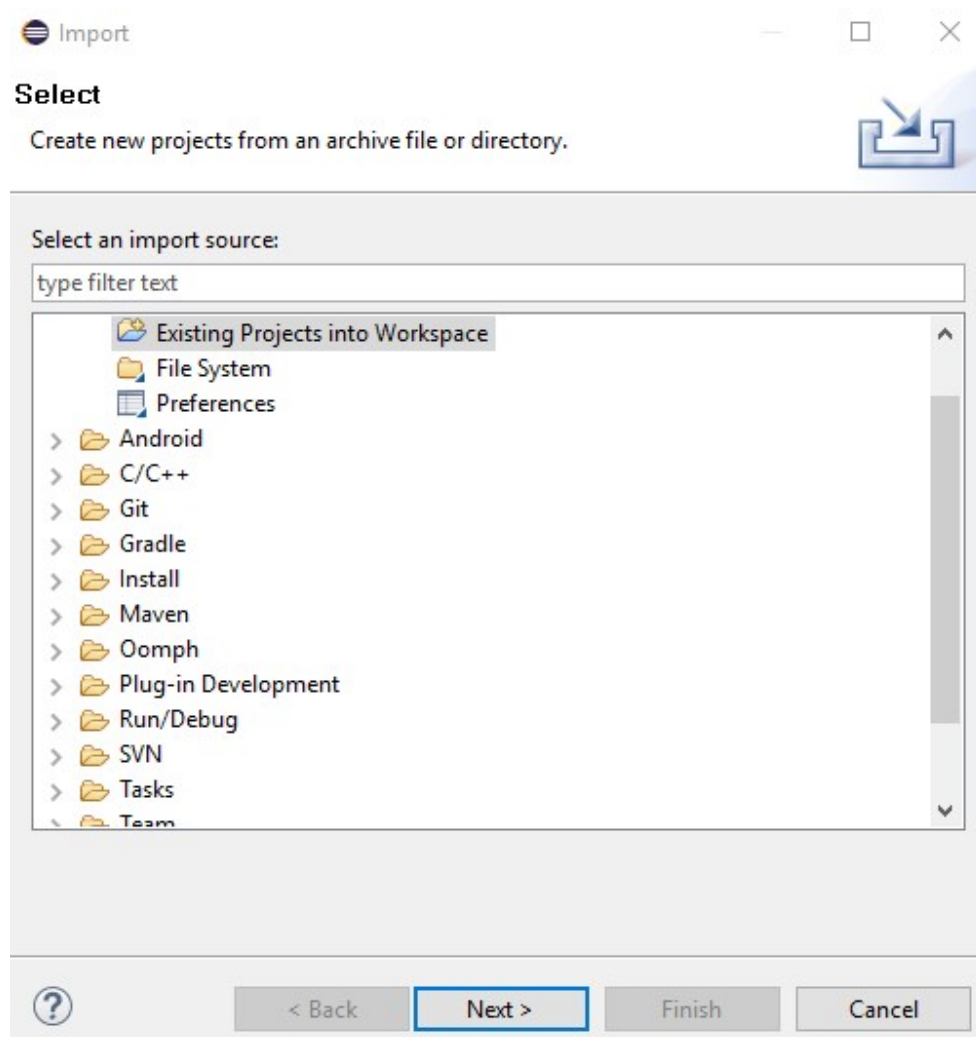
#### 4.2.1. 在 Eclipse 内运行 gateway 和你的 plug-in

下载 plug-in 示例

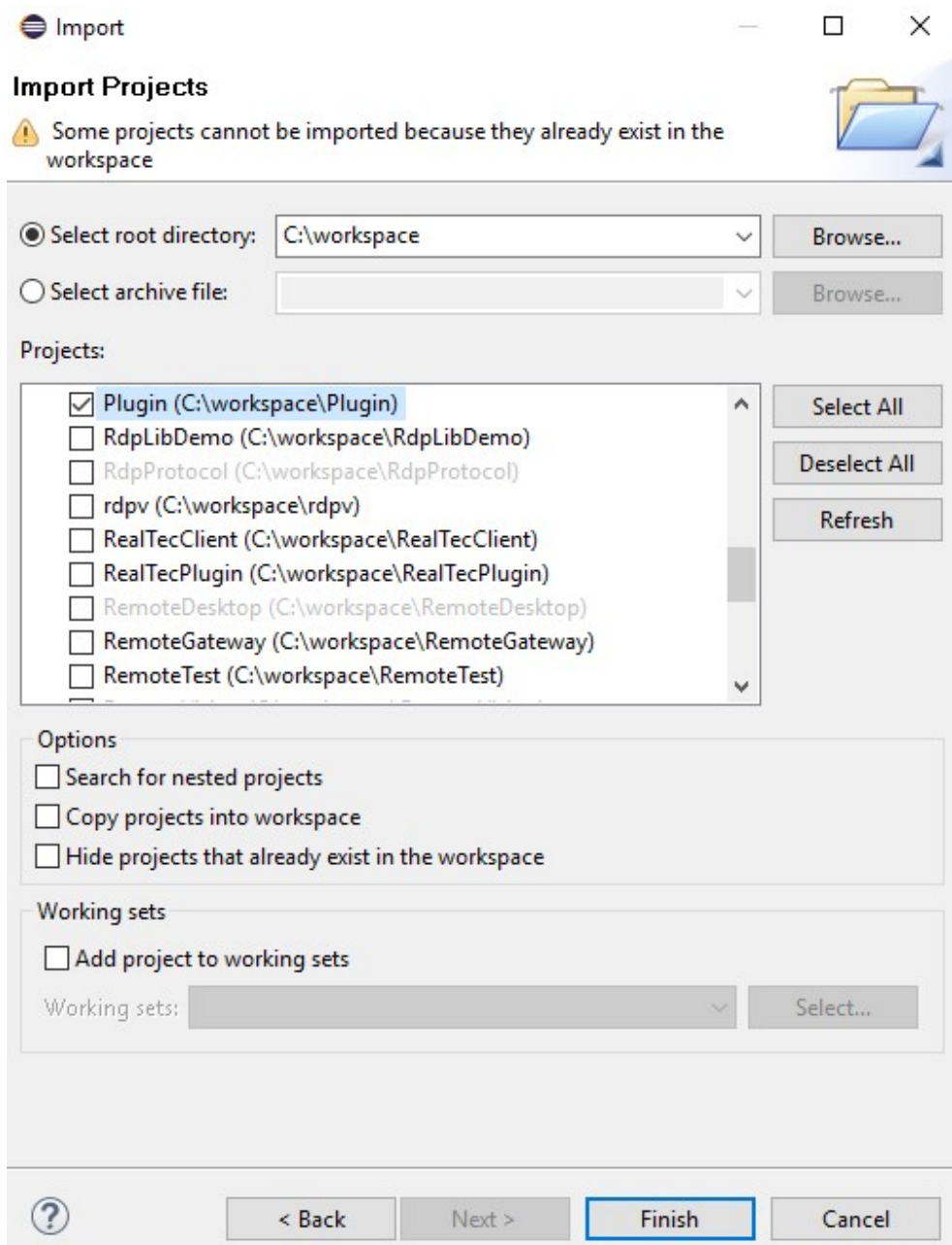
<http://remotespark.com/Plugin.zip>

解压文件到你的 Eclipse 工作空间目录

File -> Import:



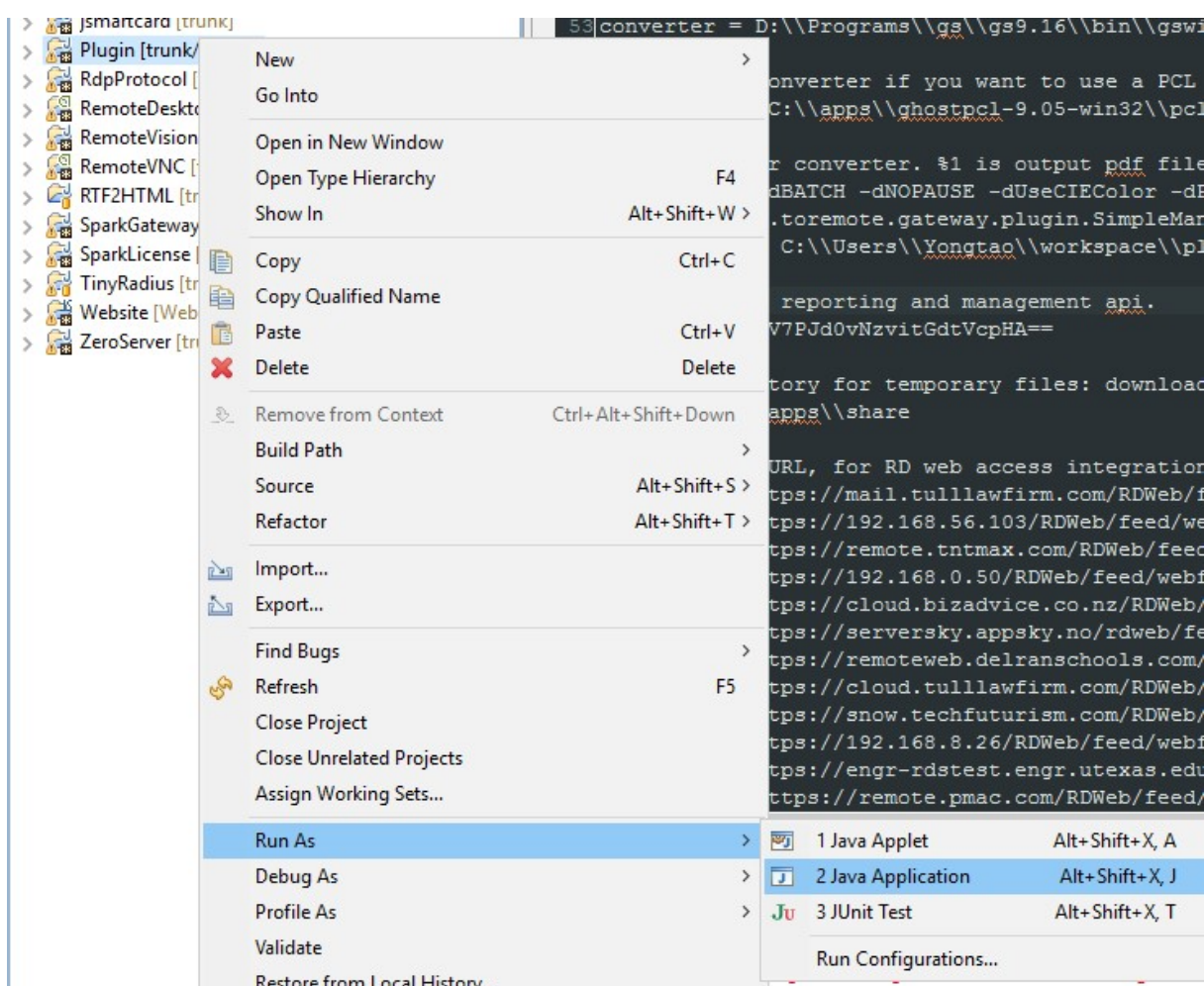
点击“Next”



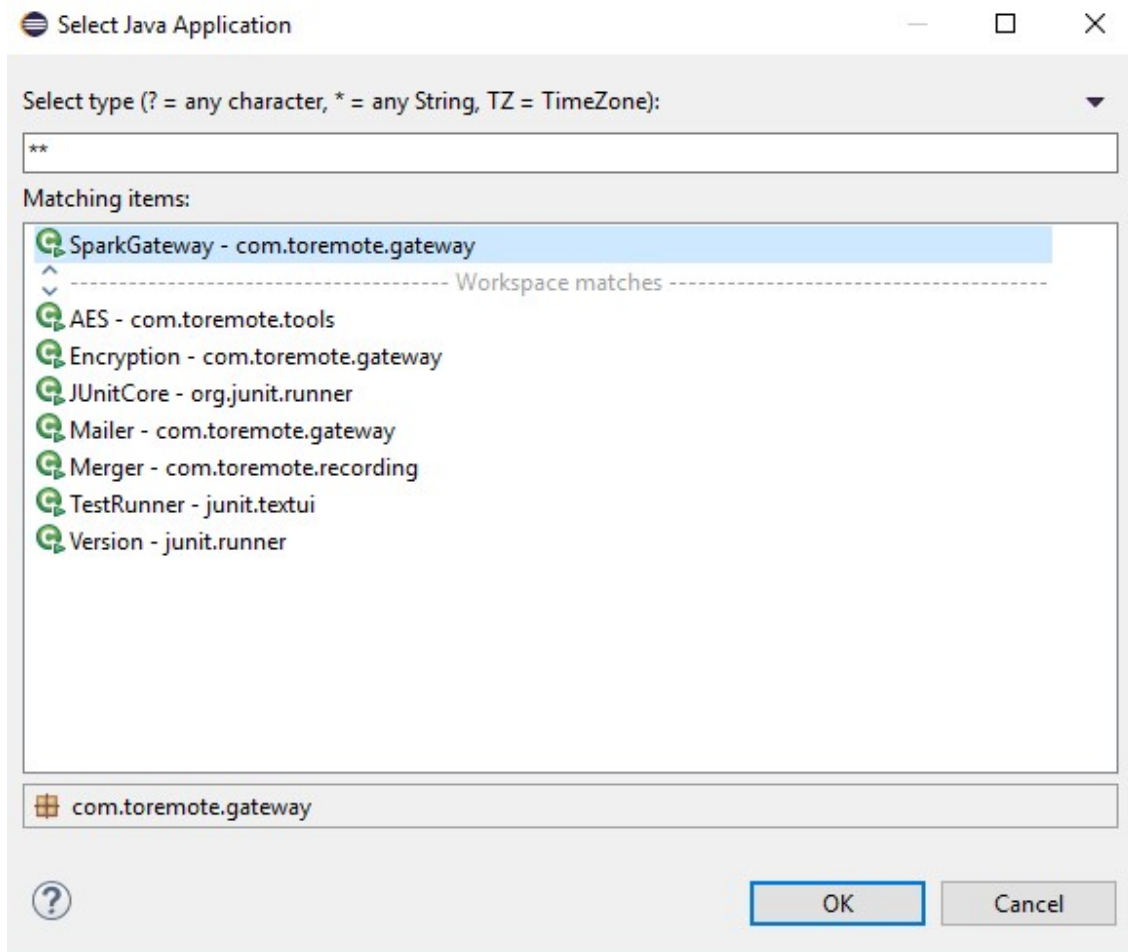
点击“Browse”到你的工作空间目录

选择 Plugin 项目。

点击 “Finish”



右键点击 Plugin 项目，“Run As” -> “Java Application”



确保选中“SparkGateway – com.toremote.gateway”，点击“OK”

现在 gateway 和你的插件一起运行在 Eclipse 内，你可以使用 “Debug As” -> Java Application 来调试你的插件代码。

#### 检查清单：

- 示例可能使用的是一个老版本的 SparkGateway.jar (Plugin\libs\SparkGateway.jar)。你可能需要用新的 SparkGateway.jar（可以从安装目录里找到）替换这个旧版本。

#### 4.2.2. 握手插件 Handshake plug-in

每次一个新的远程会话建立时，握手插件都会被调用，所以该插件可以用来验证，修改甚至拒绝此次远程会话。

当你在网页里面创建一个远程会话时，你可以在 URL 内添加任意参数，比如用户的 session id (用户登陆后获得的一个 session id，和 RDP 远程会话无关)，你可以将它加入到远程会话的 URL 中：

```
var rdp = new svGlobal.Rdp('ws://myGateway/RDP?server=myServer&token=mySessionId&...');
```

然后在 Gateway 握手插件中，你可以得到这个参数并且验证这是否是一个已登陆/授权的用户：

```
class SimpleHandshakePlugin implements HandshakeInterface{

    @Override

    public Map<String, String> onHandshake(Map<String, String> parameters) throws
ClientException{

        String token = parameters.get("token");

        if (isInvalidSession(token)){

            throw new ClientException("Invalid sessioin");

        }

        return parameters;

    }

}
```

你也可以对这个 token 进行加密（使用 ASP.net, JSP， PHP 等），然会在握手插件内进行解密：

```
public Map<String, String> onHandshake(Map<String, String> parameters) throws
ClientException{

    String token = decryptToken(token);//throw ClientException if not valid

    Map<String, String> paramsFromToken = parseToken(token);

    parameters.put(RdpParameter.server, paramsFromToken.get("server"));

    parameters.put(RdpParameter.user, paramsFromToken.get("user"));

    parameters.put(RdpParameter.pwd, paramsFromToken.get("pwd"));

    String userIp = parameters.get(RdpParameter.ARG_CLIENT_IP);

    //enable recording

    parameters.put(RdpParameter.sessionRecord, "1");//enable session recording

    //specify the recording file name (optional)
```



```

        parameters.put(RdpParameter.RECRODING_FILE_NAME, "myFileName");

        return parameters;
    }

```

#### 最佳实践:

- 确保代码线程安全
- 确保操作能在 3 到 5 秒内尽快完成，否则考虑使用另外一个线程。
- 你可以使用 HTTP API 来代替插件。

#### 4.2.3. 部署你的插件

合并你的插件和 SparkGateway.jar，将他们导出到一个 jar 文件：

右键点击插件项目， "Export", 在 Java 项目下面选择 "Runnable JAR file"

在 Launch configuration 内选择"SparkGateway - Plugin"

选择导出目的的

点击 "Finish"

确保在"Library handling"中选中"Extract required libraries into generated JAR".

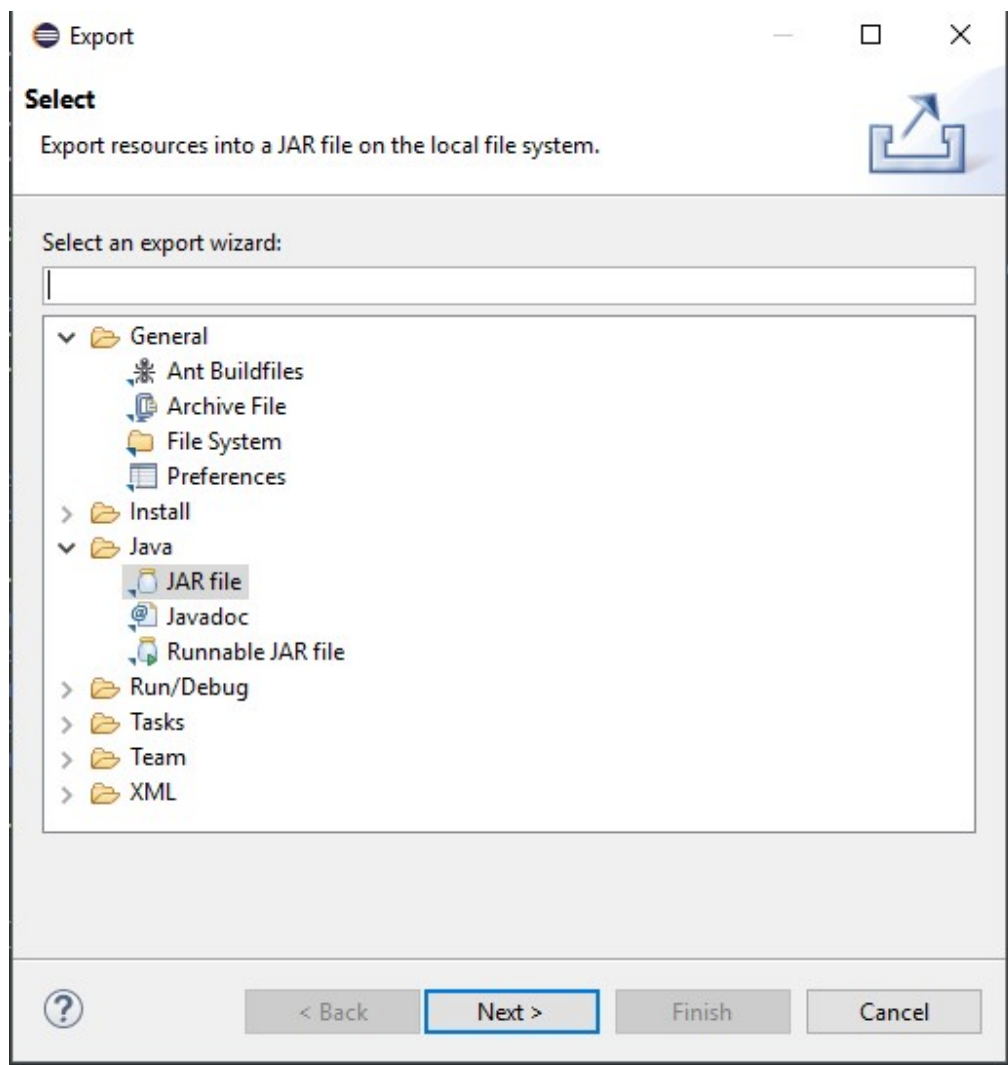
在 gateway.conf 中设置你的插件类：

```
plugin = com.mycompany.plugin.SimpleManager
```

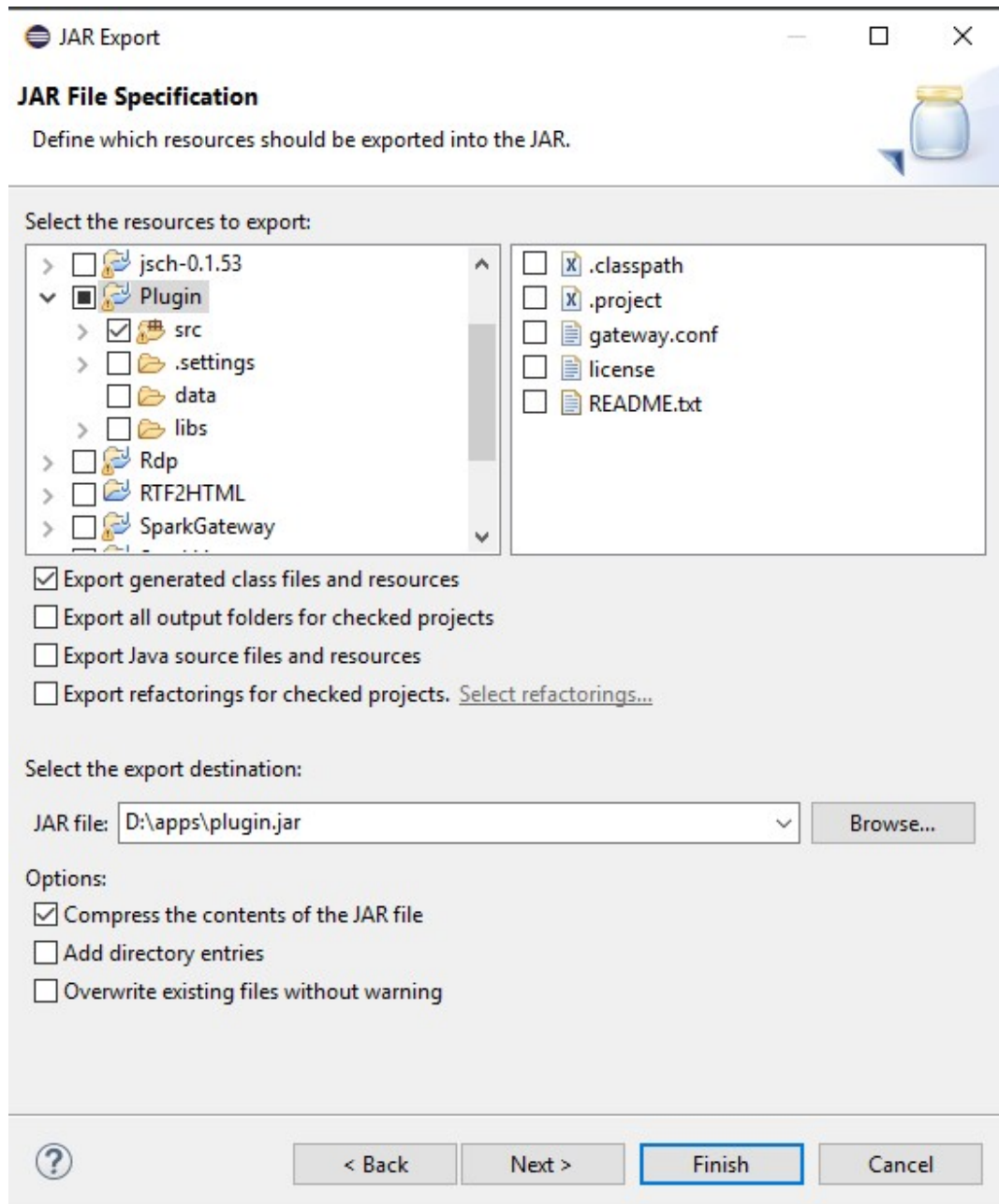
最好导出你的插件到一个独立的 jar 文件（不与 SparkGateway.jar 合并成一个文件），这样将来升级的时候，你不需要每次都要重新编译并导出你的插件，如果是这种情况你需要在 gateway.conf 中制定你的插件的位置：

```
pluginFile=theLocationOfYourPlugin.jar
```

下面是如何仅仅导出你的插件代码部分：



确保选中“src”目录:



点击“Finish”并确保在 `gateway.conf` 中设置 `pluginFile`: