

# VI Client Plug-ins Technical FAQs

---

## Technical FAQs for VI Client Plug-ins

### General

Q: What software is needed for developing VI Client Plug-in?

A: You will need VI infrastructure 3.5 and other software tools depending on the platform you use for development. For example, if you use a Java-based backend, you will need JDK, Tomcat (or other application server) and a related IDE.

Q: Does a VI Client Plug-in work without VC server?

A: No. VI Client Plug-ins require VC server.

Q: Can I develop or use a plug-in with VI 3.0 or earlier?

A: No. VI Client Plug-ins is a new feature (experimental) since VI 3.5 (VirtualCenter 2.5).

Q: Do I have to develop registration code to register my plug-in?

A: Registration utilities are available in Perl, Java, and PowerShell. You can take extend one of these examples or build your own from scratch.

### Development

Q: Do I need to restart the VC server to activate a VI Client Plug-in?

A: No. Once you register a VI Client Plug-in via the registration API, it's available right away. When a VI Client Plug-in is registered, any active VI Clients will need to be restarted.

Q: Is it possible to show an extension point based on some attribute of the selected object?

A: Yes. You can do this by using the customAttribute tag in the configuration file with the extension point. For example,

```
<customAttribute name=  
"Attr1">My VM</customAttribute>
```

Under this configuration, the extension point will only be available if the selected object has a custom attribute called Attr1 with the value "My VM".

Q: How can I get the name and IP address of the selected item?

A: You cannot get the information directly. However the VI client sends you enough information, including sessionId, moref, and serviceUrl, to retrieve them from the VC server. In fact, you can ask much more than name and IP address. See next question for more detail.

Q: How can I use the sessionId to login back to the VC server?

A: The sessionId is essentially an HTTP cookie. The following Java code shows how to use the session ID to share a login session with VI Client.

```
VimBindingStub vimStub =  
(VimBindingStub) vimService;  
vimStub._setProperty(org.apache.axis.transport.http.HTTPConstants.HEADER_COOKIE,  
"vmware_soap_session=\"B3240D15-34DF-4BB8-B902-A844FDF42E85\"");
```

For more information, please check the tech note: [Sharing sessions among multiple clients in VI SDK](#).

Q: Can I change the default size of the popup window?

A: No. The default popup window size is width - 714 pixels, Height - 601 pixels.

Q: Can I change the title of the popup window?

A: Yes. You can call `window.external.SetTitle(title)` using javascript to set the title.

Q: Can clicking a button in the popup window close the window?

A: No. This feature is planned for a future release.

Update: associate `window.external.CloseWindow()`; JavaScript with a button to close it.

Q: When I register my plug-in, some fields are empty. What are these fields used for?

A: These parameters are reserved for future use. Please provide empty values for these fields when registering your plug-in.

## Troubleshooting

Q: How can I troubleshoot my plug-in?

A: First, make sure your configuration file complies with the schema. You can validate your configuration file against the schema using a XML tool. Note that in some rare cases, the configuration looks fine in some XML editors, but fails due to invisible characters inserted by the editor into the configuration file. In such cases, we recommend using a plain text editor or a hex editor to examine every byte of the configuration file.

On the VI client machine, open a browser and verify the URLs in your configuration file are valid links.

Check the log file. See the next question for its location.

Q: Is there a log file available?

A: The VI Client log file has some debug information. The log files can be found in:

```
%USERPROFILE%\Local  
Settings\Application Data\VMware\vpv
```

There might be several log files with names like `viclient-?.log`, where ? is a digit. `viclient-index.xml` file in the same directory indicates which file is the current active one: (a little math needed to subtract the `NextIndex` by 1.)

```
<?xml version=  
"1.0"?>  
<LogSettings xmlns:xsi=  
"http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd=  
"http://www.w3.org/2001/XMLSchema" xmlns=  
"www.vmware.com">  
  <MaxLogFiles>10</MaxLogFiles>  
  <NextIndex>3</NextIndex>  
</LogSettings>
```

Q: Some URLs open fine in my browser but don't work when I put them in my XML configuration file.

A: Since URLs are included inside the configuration XML file, any special characters within the URL string must be escaped. To avoid this problem, use the CDATA tag around the URL if it has any special characters. The following is a small sample showing how to wrap the URL string.

```
<url><![CDATA[http://maps.google.com/maps?f=q&hl=en&geocode=&time=&date=&ttype=&q=3210+Porter,+Palo+Alto,+CA]
```

## Internationalization and Localization

Q: How can I localize my VI Client Plug-in?

A: There are several different things you will need to do:

In the configuration file, find the title tag and provide all the locales you intend to support. For example, if you plan to support English and Chinese, then you should provide respective locale entries.

```
<title locale="en">VMware Communities</title>
<title locale="cn">VMware##</title>
```

NOTE: there is no default locale, so you have to provide all the locale entries; otherwise the extension point won't show up.

If you are using a backend web application, it should parse the URL sent from VI client to get locale information and serve a page in that locale. The real URL that is going to be sent to the server could be:

```
http://dev:8000/vmAction.cgi?cmd=powerOn&moref=VirtualMachine:16&sessionId=9241E7B8-A37B-4264-A
```

Check the links on each localized page to make sure it links to corresponding localized pages as well.