iControl 101 - #04 - Language Options

Joe Pruitt, 2007-11-10

So you know how iControl works and you've been through the taxonomy of the interfaces so you are fairly clear on how to navigate the API. The next logical step is to start writing some code. This article will discuss several languages and development tools and hopefully get you going in the right direction to start writing your very first iControl app.

First steps first:

The first thing you are going to have to decide on is which language/toolkit you will be using to write your iControl application. Sure, I guess you could hand roll your own SOAP, but why go to the effort when others have done all the hard work for you. This article will assume that you would rather spend your time creating your application instead of debugging the intricacies of XML Schema.

There are many SOAP toolkits out there in almost any language imaginable. In this article, I'll cover the three most popular platforms: .NET (C# & VB), Java, and Perl. While each language has options for all types of applications (windows native, web, command line, etc), I'm going to give recommendations for what I deem as the "most ideal" for each type. Of course, everyone has their own preferences, so I won't take offense if you disagree with me...

.NET Based Languages

Toolkit: Microsoft Visual Studio
Typical User: Application architect
Great for: Windows Applications, Web based ASP Apps, and Command line apps.

Whether you are a C# or VB developer, the setup is the same. You'll need to get your hands on a copy of Visual Studio. Microsoft has made "Express" versions available for free if you don't want to pony up for the full blown suites. No matter which version of Visual Studio you are using, you should be in great shape to start building an application. If you are building a windows native applications, the steps are as follows:

1. Download the iControl .NET assembly from DevCentral Labs and put it in a local folder (let's say c:\lib\iControl.dll)
2. Start Microsoft Visual Studio
3. Select File.New.Project and pick the type of project you would like to build (windows, command line, ASP.NET Web Application, ...).
4. Right click on the "References" Tree item in the Solution Explorer, and select "Add Reference".
5. In the "Add Reference" dialog, select the "Browse" tab and type in the location of the iControl assembly - ie. c:\lib\iControl.dll.
6. Click OK.
7. Create an instance of the iControl Interfaces class, configure it with your connection information, and start making iControl calls.

In 3 lines of code, you can be making remote API calls!

```
iControl.Interfaces interfaces = new iControl.Interfaces();
if ( interfaces.initialize(bigip, port, user, pass) ) {
  String version = interfaces.SystemSystemInfo.get_version();
}
```

At this point you have access to all 2500+ methods in the iControl SDK via the iControl assembly.

Java

Toolkit: Apache Axis2
Typical User: Application Architect
Great for: Web Based JSP apps.

Although there are several toolkits out there for the Java platform, Axis2 is probably the most actively developed Java toolkit on the market. For this fact, we recommend new users take a close look at Axis2 as their choice of toolkits for Java. Since Java is just a language without a corporate bound IDE around it, there are many options for a developer environment. Eclipse is likely the most popular out there so I will use that as a basis for this article. Other development platforms will be configured in a similar manner.

1. Download the iControl Java library from DevCentral Labs and download the Java binary distribution into a local folder (let's say c:\lib\iControl.jar).
2. Download the Apache Axis2 libraries from the Apache Axis2 website.
3. Start Eclipse
4. Select File.New.Project and select the project type you would like (java, jsp, etc), give it a name and click Finish.
5. Right click on your project and select "Properties"
6. In the "Properties" dialog, select the "Java Build Path" list item and then select the "Libraries" tab within that view.
7. Click the "Add External JARs..." button and browse to the iControl java library - ie. c:\lib\iControl.jar, and click OK.
8. Repeat #6 for the Axis.jar file downloaded in step #2.
9. Now you are ready to write your first application. Create an instance of the iControl Interfaces class, configure it with your connection information, and start making iControl calls.

In 4 lines of code you have access to all 2500+ methods in the iControl SDK!

```
iControl.Interfaces interfaces = new iControl.Interfaces()
if ( interfaces.initialize(bigip, port, user, pass) ) {
  iControl.SystemInfoPortType t = interfaces.getSystemSystemInfo();
  String version = t.get_version();
}
```

Perl

Toolkit: SOAP::Lite
Typical User: Network Architect
Great for: command line/script automation.

Perl is not for the faint of heart. Since there are no strong client bindings, you are going to have to package up your own parameters in the method calls. The benefit here though is that there is no client side code required outside of the Crypt::SSLeay module for SSL based SOAP messages. Here are the basic steps to get your started:

1. Install perl. On Unix, it's most likely already there. For windows, I'd recommend ActivePerl from ActiveState. The basic distribution is free so the only thing you have to lose is the disk space required to install it - but, it's a LOT smaller than Visual Studio or Eclipse!
2. Install the required perl modules (SOAP::Lite & Crypt::SSLeay). They are available from CPAN or PPM in Unix/windows respectively.
3. Now you are set, start writing some code. Here's the basic code to make an iControl call:

```
use SOAP::Lite
sub SOAP::Transport::HTTP::Client::get_basic_credentials {
  return "$sUID" => "$sPWD";
}
```

```
$SystemInfo = SOAP::Lite
  -> uri('urn:iControl:System/SystemInfo')
  -> proxy("https://$sHost:$sPort/iControl/iControlPortal.cgi");
$soapResponse = $SystemInfo->get_version();
$version = $soapResponse->result;
```

Not too shabby considering there are no client bindings required!

Conclusion:

Each language has it's own place and suits different needs. In some cases you may need to use multiple languages to build a solution that works for you. Whether you've already made your choice about which language/toolkit to use for your iControl application or you are still deciding which is best for you, this article will point you in the right direction to get the tools in place and configured to start being productive!

Get the Flash Player to see this player.

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com