FOLLOW THE LEADER

ACTIVETCL ON ROUTERS AND NETWORK APPLIANCES

ActiveState

FOLLOW THE LEADER

The network infrastructure sector is a competitive space. Router and network appliance manufacturers are faced with the challenge of providing rock-solid hardware while making sure their devices are flexible and configurable enough to meet a wide range of user needs. Cisco added Tcl (Tool Command Language) as the primary scripting interface to it's IOS router operating system, allowing their customer to automate and customize their network environment with an established, standardized dynamic programming language. As the industry leader in the networking space, their decision to use Tcl is important. Should you consider doing the same?

> *"Scripting languages are sometimes referred to as glue languages or system integration languages."*
>
> *Scripting: Higher Level Programming for the 21st Century*
>
> - JOHN OUSTERHOUT

CHOOSING OR IMPLEMENTING A SCRIPTING INTERFACE

System administrators need to be able to automate much of what they do, especially when working with large networks. Configuring or re-configuring dozens, or hundreds, of routers can be tedious or nearly impossible without some method of automation. For example, if a change in network configuration requires setting a new gateway for dozens of routers, logging in to each one to make the change could tie up a sysadmin for hours.

Interface designers for these devices can foresee many, but not all, possible configuration options and scenarios. Standard network configuration operations will, ideally, be easy to propagate across a network using some administrative tool provided by the manufacturer. However, not all networks are the same, and some users will need to get at device functionality directly, bypassing limitations in the configuration interface, or at least automating interactions with that interface.

Many devices provide some kind of command line access (via telnet or ssh). Linux or other POSIX-based systems can expose a system shell which offers some control. However, these shells provide, at best, only a limited subset of the features available in a full scripting language.

> *Ideally, the scripting interface should already be familiar to the majority of target users.*

Scripting languages, also called dynamic languages, can provide the glue necessary to link system functions with programmatic control. Though a system shell can access commands to perform certain functions, a scripting language will often have libraries for accessing system APIs directly while still retaining the ability to execute the system commands in a programmatic way.

CHOOSING THE RIGHT OPTION

There are a number of questions that need to be asked by network appliance makers when deciding how to expose the functionality of a device through a scripting interface:

FOLLOW THE LEADER

What is technically feasible?

Whatever solution is chosen has to be delivered and tested within the parameters of the product's release schedule. The solution has to be able to integrate with whatever software and hardware platform the product uses.

What would be easy to implement?

Obviously, creating a scripting interface is not easy, but even integrating an existing dynamic language is a technically challenging task. Even the first step of compiling the language interpreter for a particular platform may not be straightforward, and there's a great deal more complexity involved in creating or modifying libraries to mesh with network interfaces and APIs.

What would users find familiar or easy to learn?

Ideally, the scripting interface should already be familiar to the majority of target users. If it isn't, it should be easy to learn so that it doesn't discourage newcomers.

What makes good business sense?

The effort expended on extending a product to support a scripting interface cannot get in the way of the development and maintenance of the core functionality. Core programmers may not have the time or the expertise to deliver the best scripting solution possible in-house. If third-party software is chosen, the provider has to have proven expertise and be able to deliver a solution that offers significant cost advantages to in-house development.

What did Cisco do?

Cisco Systems chose Tcl. The Cisco IOS Tcl shell is now standard on most Cisco routers.

Why should you follow Cisco's lead?

Cisco is the market leader in networking technology. With over 60% of the market share in core routers, their decision to use Tcl means that thousands of network managers and system administrators all around the world are using this language to configure and automate the admin¬istration of routers on their networks. Companies such as f5 and Nortel use re-branded versions of Tcl (iRules and Bay Command Console respectively) in many of their devices.

> *"The language has only a few fundamental constructs and relatively little syntax, which makes it easy to learn. The Tcl syntax is meant to be simple."*
>
> - PRACTICAL PROGRAMMING IN TCL AND TK
> BRENT B. WELCH

CHOOSING TCL

Technical Benefits

Cross Platform

Tcl is available for a wide range of operating systems: Windows, Linux, OS X, and almost any UNIX variant. Additionally, it has been ported to a variety of embedded operating systems (Windows CE, QNX, VxWorks).

Well Suited for Network and System Management

The shell environment is familiar to most system administrators. If they can write a shell script, they can write a Tcl program.

A Large Corpus of Add-on Modules

Tcl has a number of networking libraries that can be used to directly access network protocols such as DNS, FTP, NTP, SMTP, HTTP, SSL and more.

FOLLOW THE LEADER

Easy to Learn

Among the scripting languages, Tcl is perhaps the simplest syntacti¬cally, having only 13 rules:

> *"The language has only a few fundamental constructs and relatively little syntax, which makes it easy to learn. The Tcl syntax is meant to be simple." Practical Programming in Tcl and Tk – Brent B. Welch*

Free online resources and a number of excellent books are readily available to get new users up-to-speed quickly. The core Tcl documentation itself is often enough to allow a novice to start writing useful applications.

High Quality and Constantly Improving

The Tcl language has been evolving and maturing since it's creation in the late 1980s[1]. As with other open source programming languages, Tcl's codebase is under constant scrutiny and benefits from having many "eyes on the code".

Compatibility

Unlike other languages which have changed drastically over time causing upgrade headaches for those responsible for deploying them, Tcl has maintained excellent backwards compatibility. With few exceptions, 10 to 15 year old Tcl code runs unmodified in the latest version.

Expect Scripting remote command-line applications is difficult. Most scripting languages cannot deal with this problem effectively, but using the Expect package in Tcl makes these interactions easy to "drive" remote command line applications, or automate existing or legacy tools that were never built to be automated.

Expect

programs written in Tcl are the easiest way to "drive" remote command line applications. Numerous Expect scripts for controlling routers and switches have been written and shared on the web by system administrators.[2]

Testing

Tcl and Expect are exceptionally well suited for use in automated testing. In addition to helping customers configure and deploy your devices, a Tcl scripting interface will help QA teams test them much more efficiently and thoroughly.

> *"... everybody who likes Unix needs to know TCL at least to the level that is necessary to use Expect, a really brilliant, breakthrough application based on TCL." A Slightly Skeptical View on Scripting Languages – Dr. Nikolai Bezroukov*

> *Thousands of active Tcl developers worldwide have contributed to a vibrant community. Many online resources are available for sharing scripts and advice.*
> *http://www.tcl.tk/community*

BUSINESS BENEFITS

Unencumbered Open Source

Tcl has a very liberal, BSD-style, open source license. It does not restrict your right to sell hardware that runs it, or any software you create that uses it. You are also not locked in to a proprietary solution with only one vendor.

Open source software offers numerous advantages to strictly proprietary software: liberal licensing, large developer communities, lack of a vendor lock-in, and lots of people examining and improving the code.

FOLLOW THE LEADER

On the other hand, Quality proprietary software offers businesses the reassurance inherent in a vendor/buyer relationship. The vendor has a monetary incentive to provide a working, affordable solution for their customer.

It is possible to combine the best of both of these models by using open source software from a vendor as if it were proprietary. Matt Assay quotes an IT executive and Bank of New York as saying:

“Open or closed source, if we touch the source code we are idiots. The whole reason we are buying product instead of developing it in-house is to pass the ownership for upgrades, QA/Testing, etc. to a company and paying a fee for that.”[3]

A Large Community Of Developers

Thousands of active Tcl developers worldwide have contributed to a vibrant community. Many online resources are available for sharing scripts and advice.

ActiveTcl Enterprise support put the expertise of core Tcl developers at your fingertips, and provides a guarantee that you will get the help you need, when you need it.

Tcl Developer Xchange: **http://www.tcl.tk/community/**

Many Cisco IOS users are already familiar with Tcl. Following in the footsteps of an industry leader allows you to leverage the existing expertise of system administrators.

ACTIVETCL

Tcl is free. You can download the sources, build it, and distribute it on a commercial device. However, integrating the source distribution can be costly on many levels. Enter ActiveTcl Enterprise.

ActiveTcl is the industry-standard distribution of Tcl from ActiveState. On average, 15,000 people download it every month. It is available for all major platforms and includes popular extensions, the Tcl Package Manager (TEApot), and complete documentation.

ActiveTcl Enterprise is a software, support and maintenance package for the ActiveTcl distribution.

ActiveTcl Enterprise provides:

Solid Binary Builds

Building a language interpreter from source is a hassle. Even if there are no hiccups, and the target platform is supported, it’s time that could be better spent on adding core functionality to your software or device or fixing bugs. Having a quality assured binary build saves time and eliminates a host of potential problems and delays. Each ActiveTcl release is thoroughly tested and put through its paces to ensure there are no regressions or other unpleasant surprises.

Enterprise-Level Support

The open source language communities are generally helpful in answering questions on mailing lists and forums, but businesses can’t wait for peer support that may, or may not, solve their problem.

ActiveTcl Enterprise support puts the expertise of core Tcl developers at your fingertips, and provides a guarantee that you will get the help you need, when you need it.

FOLLOW THE LEADER

Freedom From Licensing Worries

When choosing an open source language, you are faced with the task of trying to figure out exactly how the license applies to what you are selling and being prepared to defend your use of the software. You can either spend the time and effort to become an expert in open source licensing law, or you can rely on the experience of those who have dealt with these issues for years.

As the numerous lawsuits launched by SCO Group against various Linux distributors[4] illustrated, even unproven allegations can initiate a series of legal machinations that can tie up a company's lawyers for years. Companies at the front lines in these battles must have complete knowledge of the source code in question in order to defend their posi¬tion. You don't want to go there, and you don't have to.

The optional indemnification component of ActiveTcl Enterprise ensures that ActiveState bears the responsibility for dealing with potential copyright infringement claims and other licensing issues.

Don't Reinvent the Wheel

Technology decision makers don't like to think of themselves as followers. The expression "thinking outside of the box" has become an industry buzz phrase which, in some people's minds, suggests that doing anything non-revolutionary is boring, unimportant, or just not interesting enough to consider.

While it's not a good idea to choose a technology solely because a market leader has done so, if they choose a technology that is suitable, inexpensive, easy to implement, open, and above all useful, there's much to be gained by following their lead.

Modern network devices need to be flexible and easily configurable in order to be competitive in the marketplace. Tcl is a proven choice for providing this functionality, being widely used thanks to it's inclusion in Cisco IOS. Cisco Systems has paved the way by introducing Tcl to thousands of system and network administrators worldwide. ActiveState has made it easy to deploy Tcl on a wide range of hardware. Sometimes a technology choice lies conveniently "inside the box".

Talk To Us About Tcl

If you're building a network device or writing software for one, talk to ActiveState about how ActiveTcl could be used for your application. The Tcl experts at ActiveState can go through the specific requirements of your projects and determine if ActiveTcl Enterprise or ActiveTcl OEM licensing would be right for you. ActiveState's years of experience working with companies in the network appliance industries could save you time and money.

Contact ActiveState at 778.786.1101, or business-solutions@activestate.com for a complimentary consultation with ActiveState's language experts.

1. http://www.tcl.tk/about/history.html
2. http://www.samag.com/documents/s=1180/sam9903c/9903c.htm
3. http://news.cnet.com/8301-13505_3-10031348-16.html
4. http://www.groklaw.net/staticpages/index.php?page=20050315132709446

ActiveState Software Inc.
sales@activestate.com

Phone: **+1.778.786.1100**
Fax: **+1.778.786.1133**

Toll-free in North America:
1.866.631.4581

ABOUT ACTIVESTATE

ActiveState believes that enterprises gain a competitive advantage when they are able to quickly create, deploy and efficiently manage software solutions that immediately create business value, but they face many challenges that prevent them from doing so. The company is uniquely positioned to help address these challenges through our experience with enterprises, people and technology. ActiveState is proven for the enterprise: more than two million developers and 97 percent of Fortune 1000 companies use ActiveState's end-to-end solutions to develop, distribute, and manage their software applications written in Java, Perl, Python, Node.js, PHP, Tcl and other dynamic languages. Global customers like Cisco, CA, HP, Bank of America, Siemens and Lockheed Martin trust ActiveState to save time, save money, minimize risk, ensure compliance and reduce time to market.