

Application Ready Networks: Built-in ROI

NETWORKING GEAR EXISTS

to host applications, which exist to front data, which exists to make users more productive. If even a single application on the network shows poor performance, you'll hear about it.

Up until now, if you wanted to get the most from your complex network infrastructure, you had two choices: Go out, do the research, build your own by doing it yourself, deal with growing pains, and eventually build a solid foundation; or hire specialized consultants to do it for you. Both of these solutions have their problems, but when users are upset because "email is too slow" or "our app is not responding," IT managers have little choice but to go with one of these options.

At F5, we're aware of this dilemma and are working with some of our larger application business partners to reduce your invested time and/or the consultant dollars you spend implementing some of the more common applications. In a nutshell, that is what the F5 Application Ready Network (ARN) is all about—giving you what you need to increase the security, reliability, and efficiency of your application deployment.

F5 ARNs currently cover Microsoft, Oracle, and SAP technologies, with more in the works. Already you can speed up the performance of applications like

SharePoint with just a few quick changes to your network, knowing we've ironed out the kinks for you.

Lori and I have helped to set up "solution centers" at large publications and think that the F5 ARN takes the idea a step further. Solution centers tend to become "advertising centers"—with limited reliable information but lots of marketing fluff. F5 ARNs are built on the precept that you're already a customer, so we're just giving you the technical information you need to do your job. And the information is tested in the data center, in the same "real-life" conditions you deal with every day.

But the ARN doesn't end with handing you a fistful of PDFs. We offer prebuilt configurations, like our WebAccelerator policy that is customized for use with PeopleSoft v9. Perhaps most importantly, we offer support on DevCentral through dedicated ARN pages, forums that are monitored by F5ers like me and Lori, and a growing collection of documents that help you get the most out of your network infrastructure.

The value that you gain from the ARN program is increased productivity, less time spent searching for the best way to implement or enhance your applications, and a commitment from us to help you solve problems not covered in the ARN documents. Best of all, the configurations and policies that we include in the ARN were tested by F5 and the specific application partner. These are tried-and-true implementation samples that will guide you to the best solution for your most central applications. So far we haven't seen this offered anywhere else.

You also benefit from the power of your peers. Many DevCentral threads are answered by knowledgeable users just like you who only want to lend a hand.

Visit www.f5.com or <http://devcentral.f5.com> and find out what the ARN can do for you. The chances are good that you're running some of the applications we cover, and that we can help you improve your integration with those applications. And best of all, it won't cost you a thing.

Written by Lori MacVittie, technical marketing manager for application services, and Don MacVittie, strategic architect at F5. Read their blogs at <http://devcentral.f5.com>.

FINE-TUNING

The F5 Application Ready Network is the result of F5 engineers installing, configuring, and testing each application with F5's integrated product portfolio. Countless hours are spent exploring different scenarios and tuning our products, and then carefully documenting the procedures to ensure optimal and reproducible results.

For example, for Oracle Application Server 10g Portal, we used BIG-IP LTM, Symmetric WebAccelerator, and WANJet to improve the speed of downloading a document over the WAN from Oracle Application Server 10g Portal by more than 22 times, reaching near LAN speed over the WAN.

For Microsoft Office SharePoint 2007, the combination of BIG-IP LTM, WebAccelerator, and WANJet improved application performance over the WAN by more than five times.

For more information on these results, visit www.f5world.com.

DevCentral updates

F5 performance report pages are live!

With the ever-shrinking number of enterprise-class testing publications out there, the industry must do more to set and maintain valid benchmarks for products that claim to improve your business performance. You, the customer, have a need for unbiased information about the performance of products that you purchase. F5 has taken the bold step of offering not only the results of testing, but also the test methodology and the device configuration files for the tests.

DevCentral, in cooperation with the F5 Test Lab, is pleased to offer you this information free of charge as a DevCentral registered user. Simply visit <http://devcentral.f5.com> and choose “Performance” from the Docs menu to access all sorts of great performance-related content.

DevCentral—now home to the F5 Application Ready Network for Oracle

F5 has been working hard to help you get the most from your F5 and Oracle solution. We have set aside a space on DevCentral and gathered information to get you started, from notes on streamlining service-oriented architecture (SOA) web services to information that will help you with Oracle/F5 troubleshooting.

Drop by the DevCentral/Oracle web pages to get information direct from the F5 experts, links to Oracle Technology Network pages, and access to forums

and other content relevant to Oracle users. We are also thrilled to offer you the Oracle F5 SOA Blueprint—information about how Oracle and F5 products can be used together to implement and optimize your SOA.

The 101 series: getting started

As part of our efforts to provide you with digestible information for all different levels of expertise, we have started a series of “101” introductory articles. They take a deep dive into some of the more popular areas of iRules and iControl, and show you not only how to get started using these tools, but also how to get the most out of them.

Our experts will show you the important basics of their topics and then build upon that knowledge to teach you even more. It’s like an introductory college course, but without the homework assignments.

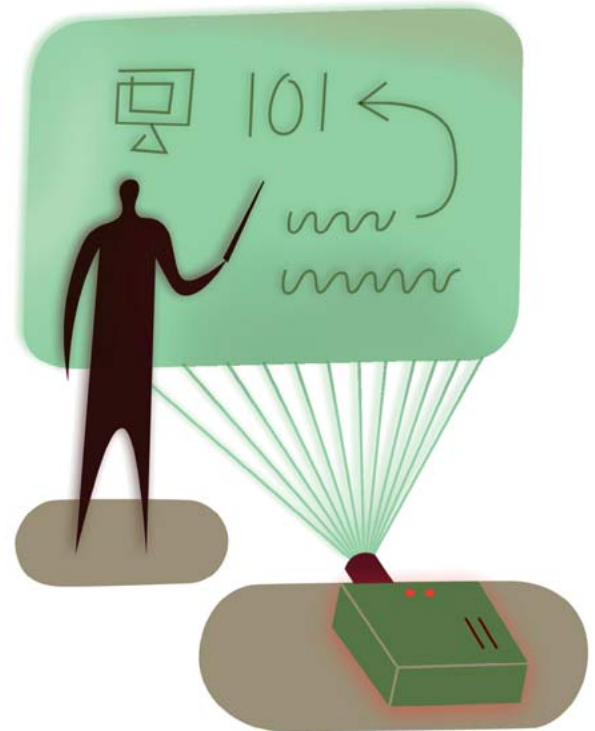
Need to get performance out of your iRules from the day they go live? Check out the *iRules Optimizations 101* series, exploring everything you need to tweak performance of your iRules.

Worried about security? Check out the *iRules Security 101* series, where

we discuss using iRules to implement security policies, and much more.

Wondering if iControl is right for you? Check out the *iControl 101* series, which takes you from data sheets and white papers to the basics of iControl development, through parsing iControl error messages, and beyond.

Be sure to watch for more “101 series” articles in the near future as we expand the depth of coverage and add some new topic areas.



DevCentral community expertise just keeps growing

Our favorite aspect of DevCentral is that community members share samples and tips on a daily basis. In this issue of *F5 World*, we would like to spotlight one user, hoolio, and his expertise in one of the forums:

Testing iRule commands

You can test basic commands in the RULE_INIT event. This event is triggered when the

rule is created or saved. You can check the `/var/log/ltn` file for the log output. This is an easy way to check string commands and other commands that are allowed in the RULE_INIT event.

Determining if a request was HTTP or HTTPS

If you want to determine whether to use HTTP or HTTPS in the location header

value for wildcard port virtual servers, you can check the requested port (using `TCP::local_port` in the client side context).

Checking the port number of a request

The `HTTP::host` command will return information after the protocol (`http://` or `https://`) and before the start of the URL (`/path/to/file`.

`txt`) in the address bar of the browser. If a client requested `https://test.example.com:80/index.jsp?pageno=1`, `HTTP::host` would return `test.example.com:80`.

Typically, most requests don't include the port number, so unless the requested URL contained the port, splitting the `HTTP::host` value on a colon will return an empty second field.