

the trusted technology learning source

[Home](#) > [Articles](#) > [Operating Systems, Server](#) > [Solaris](#)

Sun ONE Portal Server 3.0 Rewriter Configuration and Management Guide

By [Sun Microsystems](#)

Sep 13, 2002

[Contents](#) [Print](#) [Share This](#) [Discuss](#) [< Back](#) [Page 10 of 21](#) [Next >](#)

Like this article? We recommend



[Capacity Planning for Internet Services](#)

[Learn More](#) [Buy](#)

Rewriting JavaScript Content

Because JavaScript is a programming language, there can be any number of ways to represent the same functional result. The trick is that if the result contains a URL, the gateway will need to understand that fact. The reasoning is exactly the same as the reason for rewriting HTML: If there is some URL-handling performed using JavaScript content, then the request would still need to be sent back to the gateway component rather than attempting to directly contact the internal web application server.

For many Sun ONE Portal Server gateway administrators, rewriting JavaScript content will be the most difficult task deploying and maintaining a secure

Related Resources

[Store](#) [Articles](#) [Blogs](#)



[Network Distributed Computing: Fitscapes and Fallacies](#)

By [Max K. Goff](#)

Book \$35.99



[J2EE Platform Web Services](#)

By [Ray Lai](#)

Book \$43.99



[Secure Shell in the Enterprise](#)

By [Jason Reid](#)

Book \$39.94

[See All Related Store Items](#)

Portal Server installation. The randomness at which URLs can occur throughout scripted code contributes to the difficulty, and the fact that JavaScript content is intertwined with HTML SPEC 4.0 adds to the challenge.

Areas where JavaScript content may need to be rewritten include the obvious SCRIPT elements, event handlers within HTML tags, and imported JavaScript content. Specific things to look for are references to window and document object methods, events, and properties that may affect or refer to URLs. Otherwise, obvious areas may include variable assignments, function parameters, and JavaScript object arrays.

Having a good understanding of how to write JavaScript content will be very helpful when trying to mine out URLs contained in it. While the introduction to, and usage of JavaScript content, is beyond the scope of this guide, the JavaScript objects and methods that you will need to be concerned with, as they relate to the rewriter, will be covered in detail. The browser implementation of the document object model is different for different browsers, and each browser offers JavaScript content its own set of capabilities. Those that are for the most part common between the Internet Explorer and Netscape Navigator browsers are discussed in this guide.

Web Browser Document Object Properties

Document object properties are generally manipulated through a JavaScript assignment statement. In most cases, the full object path will be on the left side of the assignment operator, and a raw URL will be on the right side of the assignment operator.

With that known, you can now differentiate when a rule should be added to the Rewrite JavaScript Variables in URLs section of the gateway profile or the Rewrite JavaScript Variables Function section. Generally speaking, when the right side of a variable assignment is a raw URL and the left side is a document object property, then the full object path should be added to the Rewrite JavaScript Variables in URLs section of the gateway profile. While assignments of this type can be handled by either section, having the server rewrite the URL will save client compute resources required to render and/or use the page.

For example, consider the following JavaScript assignment:

```
document.location.href = "http://www.ipplanet.com"
```

The right side is clearly a raw URL, so the rule to be added to the Rewrite JavaScript Variables in URLs section is: `document.location.href`

Consider the following assignment:

```
document.location.href=newURL;
```

There are two options to handle the rewriting of this statement. The first is to see if `newURL` can be rewritten elsewhere so that `document.location.href` would not have to be added to the gateway profile for this particular instance. The second is to add `document.location.href` to the Rewrite JavaScript Variables Function section of the gateway profile. As mentioned earlier, adding this rule will result in the entire right side being wrapped in an `iplanet` function call.

```
document.location.href=iplanet(newURL);
```

The `iplanet` function will evaluate its parameter and return a rewritten URL using the browser JavaScript engine. The `iplanet` function body will also be inserted within the `SCRIPT` tags. As of SP3 Hot Patch 3, adding a rule to this section of the gateway profile to handle a JavaScript assignment called from an event handler will result in the `iplanet` function body being defined inside of the HTML tag itself. Thus, the following code cannot be rewritten in versions before SP4 Hot Patch 1.

```
<INPUT TYPE="BUTTON" NAME="button" VALUE="Click"
onClick="document.location.href=newURL;">
```

This particular scenario will only arise if the `newURL` variable has been defined within a script tag in the document `HEAD`, due to JavaScript variable scoping. However, that also means that `newURL` must have an assignment defined in the `HEAD` element that might be rewritten there.

The following are some browser object properties that can contain URLs:

- `document.location.href`—Used to change the URL for the current page.
- `document.location.path`—Used as part of a relative URL.
- `document.location.protocol`—Used to form a URL.
- `document.location.host`—Used to form a URL.
- `document.referrer`—Used for the URL of the document which referred to the current one.

- `document.URL`—Used for the URL of the current document.

Because of the JavaScript object scoping, each of the proceeding properties can also have `window` prepended to the full property path. The `window` object itself also has several event handlers that can contain additional JavaScript content. Two of the more well-used `window` object events are `onLoad` and `onUnload`. `window` events are often located in the `BODY` tag of the HTML document and execute when the page is first being rendered (as in the case of the `onLoad` event). Frames can also use `onLoad` and `onUnload` events.

When referring to specific frames or other objects in the DOM hierarchy, the object path may differ, which would require either an additional rule or a wildcard rule, if possible. Neither `window.location.href` nor `parent.location.href` would be matched by the `location.href` rule. However using SP3 Hot Patch 3, if the right side is a raw URL, an entry such as `*.location.href` can be added to the Rewrite JavaScript Variables in URLs section of the gateway profile that will handle both cases with a single rule.

Web Browser Document Object Methods

There is not any one specific way in which to rewrite browser document object method calls that contain URLs. The rule syntax and appropriate section to be considered in the gateway profile depends on the method's parameter(s) and its semantics. For example, the `window.open` method takes several parameters, but they must be in a specific order, and all of them start with a URL as the first parameter. If the URL is a raw URL, then the function name can be added to the Rewrite JavaScript Function Parameters section of the gateway profile.

The syntax for a rule in this section is `funcName:y`, where `funcName` is the function name that is followed by a colon separator and either by a `y` or a comma. A comma is used to signify multiple parameters, and a `y` is used to tell the gateway that a particular parameter requires rewriting. This may be easier to understand in practice. Consider a call to `window.open` in the following example:

```
<HTML>
<HEAD>
<SCRIPT>
function myWin() {
window.open('/channels/stocks_channel.html', 'S
'width=300,height=250,directories=no,location=
scrollbars=yes,status=no,toolbar=no,resizable=
}
</SCRIPT>
</HEAD>
<BODY onLoad="myWin()";
```

```
</BODY>
</HTML>
```

After this document loads, another with stock information will automatically open. The `onLoad` attribute could be listed in the Rewrite HTML Attributes Containing JavaScript section of the gateway profile, but it is not necessary in this case because the right side of the assignment operator is not a raw URL, but instead, a function call. Inside the body of the `myWin` function, however, there is a call to the browser document object method `window.open`, whose first parameter is a relative URL. So, the following rule would be added to the Rewrite JavaScript Function Parameters section of the gateway profile:

`window.open:y`

This particular rule has already been added to the gateway profile, out-of-box, but it is referred to, in this case, to explain when and how to add a rule to this particular section of the gateway profile. If the content is passed through the gateway that contains a call to `window.open`, where the first parameter is not a raw URL, then the **`window.open:y`** method would have to be moved from the default section of the gateway profile to the Rewrite JavaScript Function Parameters Function section.

Consider the example:

```
<HTML>
<HEAD>
<SCRIPT>
function myWin() {
myURL = '/channels/stocks_channel.html'; window.open(myURL, 'myWin',
'width=300,height=250,directories=no,location=no,menubar=no,scrollbars=yes,status=no,toolbar=no');
}
</SCRIPT>
</HEAD>
<BODY onLoad="myWin()">
</BODY>
</HTML>
```

There are two ways to rewrite the page so that it will function correctly. The first, which has been discussed already, is to add `myURL` to the Rewrite JavaScript Variables in URLs section of the gateway profile. The second is to move **`window.open:y`** from the Rewrite JavaScript Function Parameters section of the gateway profile to the Rewrite JavaScript Function Parameters Function section. Although the name may be a bit confusing, this

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.