



CHAPTER 9

Using Variables in Paths

This chapter describes how to use variables in service invocation paths. It covers these topics:

- [Understanding Path Variables, page 9-105](#)
- [Configuring Variables in Pathnames, page 9-106](#)

Understanding Path Variables

The local service invocation path is the one used by clients to call services at the ACE XML Gateway. You can use regular expressions in the path to provide flexibility in the request URL and to capture dynamic portions of the request URL, such as arguments. The dynamic portions of the URL can be propagated from the incoming request path to the URL the Gateway uses to invoke the backend service.

With regular expression path matching enabled, a request is processed as follows:

1. The virtual service that receives an incoming request matches the request against one or more regular expressions configured in the **Exposed Local Path** field of the **Consumer Interface** page.
2. The ACE XML Gateway stores each regular expression match in a parameterized variable in a manner similar to the way the bash shell parameterizes arguments to a script; that is, it stores the first matching value in the `\1` variable, the second in the `\2` variable, and so on.
3. The target service descriptor substitutes the matched values for `\n` expressions that appear in the **Path On Server** field of the virtual service or service descriptor **Service Interface** settings.

For example, it replaces the string `\1` in the **Path On Server** field with the value that matched the first regular expression in the **Exposed Local Path** field. Similarly, any value that matches the second regular expression in the handler path is used to substitute the `\2` string in the **Path On Server** field, and so on.

4. The ACE XML Gateway uses the resulting path to pass the request to the backend service.

For example, a service definition could employ backslash replacement by using the following **Exposed Local Path** on an HTTP GET service handler:

```
/catalog/(.*)/public/(.*)
```

The second and fourth steps in the URL of the request path use a regular expression match.

On the service descriptor, the **Path on Server** field could be:

```
/catalog?pid=\2&class=\1
```

The values of the fourth and second URL steps of the request path, respectively, would thereby form the value of the URL arguments of the request issued to the backend service.

For example, a request to the Gateway with
`http://xmlgateway.example.com:80/catalog/telescope/public/232`

Would match the service definition, and would cause the Gateway to generate a backend service request to the following resource:

`http://service.example.com:80/catalog?pid=232&class=telescope`

The event log describes the match replacement with an item similar to the following:

Backslash replacement `'/catalog?pid=\2&class=\1'` resulted in `'/catalog?pid=232&class=telescope'`

Notice that the values can be re-ordered, as in the example. Also, a matched regular expression gets substituted for every occurrence of the parameterized variable in the **Path On Server** field.

About Ambiguous Path Matches

The ACE XML Manager prevents you from configuring the same path for more than one consumer interface, since this would create ambiguity in request matching. However, with path variables, it's possible to have an incoming request match multiple paths.

For example, consider two handlers, one with an exposed local path of `/menu/d(.*)i` and another with `/menu/de(.*)`. The ACE XML Manager allows you to compile and deploy such a policy. At runtime, however, an incoming request for `/menu/deli` would match both handlers.

In general, you should take care to avoid the possibility of introducing path-matching ambiguity when working with regular expressions. If it does occur at runtime, no error will result—the ACE XML Gateway will assign the request to one of the handlers. In general, the ACE XML Gateway cannot guaranty a predictable resolution of the ambiguity—one of the paths is chosen. However, if an ambiguity exists between two paths one of which uses a regular expression while the other does not, the match to the path with a regular expression wins.

Configuring Variables in Pathnames

To propagate path variables from the incoming to the outgoing request, you need to configure both the handler and service descriptor that handles the request, as follows:

-
- Step 1** Log into the web console as an Administrator user or as a Privileged user with the Routing Or Access Control role.
 - Step 2** Set the active subpolicy to the one that provides the service descriptor to update.
 The name of the active subpolicy appears next to the **subpolicy** label at the top of the console. Note that objects in the Shared subpolicy are accessible in all subpolicies.
 For more information, see [“Working with Subpolicies” section on page 30-300](#).
 - Step 3** Click the **Virtual Services** browser item in the navigation menu.
 - Step 4** In the **Virtual Services** browser, click the name of the virtual service object or handler for which you want to configure pathname variables.
 - Step 5** Click the **Edit** link in the **Consumer Interface** heading.
 - Step 6** In the **Exposed Local Path** field, add a regular expression that matches the value you want to accept. The expression should appear in parenthesis. For example:

```
/order/(.*)/submit
```

- Step 7** Select the option labeled **Allow regular expression matching in the local path** to have the ACE XML Gateway evaluate regular expressions in the **Exposed Local Path** field against incoming request paths.
- Step 8** Click **Save Changes**, and then click **Virtual Services** browser from the console navigation menu.
- Step 9** Click the name of the service descriptor to which the handler is attached by a route.
- Step 10** Click the **Edit** link in the **Service Interface** settings.
- Step 11** In the **Service Interface** settings page, enter the back reference identifiers in the Path on Server field. For example:
- ```
/order/\1/submit
```
- Step 12** Select the option labeled **Replace back-references (\1, \2, etc.) using regular expression matches from the handler path** to enable the back-reference replacement feature.
- Step 13** Click **Save Changes** to commit your changes to the working policy.
- 

After the policy is deployed, the ACE XML Gateway matches incoming requests based on regular expression variables.

