

Debugging Connections

Debugging Connections

ManifoldCF's various repository and authority connections often have to operate in complex enterprise environments, which have a myriad of different configuration options and pathways. It can therefore be extremely challenging to get a connection working. Many different debugging techniques may need to come into play to determine the difficulty. Worse, the kinds of techniques involved depend strongly on the underlying repository technology.

This document is meant to give ideas to those who get stuck with the job of figuring out why an ManifoldCF connection isn't working as planned. It cannot obviously be a comprehensive manual, but it can suggest avenues that might be profitable to pursue.

Debugging JCIFS connection types

JCIFS connections operate in the Windows world, and need to resolve several things in order to function properly. These are:

- Machine names
- User names

DFS, which basically is Window's system of cross-machine node referrals, requires that the connection both be able to resolve the name of the reference (which means that the crawling server's DNS setup has to be compatible with those for Windows). It also requires that the crawling server be able to authenticate reliably against all machines that host DFS content. The easiest way to guarantee that this will be the case is to join the crawling server to the local AD domain.

If you find that a JCIFS connection is not working as planned, you can try the following approaches to resolving the problem:

- Turn on connector-level debugging. To do that, add `<property name="org.apache.manifoldcf.connectors" value="DEBUG"/>` to the properties.xml file, and restart whatever services are appropriate.
- Obtain packet captures, using tools such as tcpdump and/or Wireshark. This can give a clue as to what is going wrong.

Debugging SharePoint connection types

The SharePoint connection type works through IIS. IIS can be configured in many ways that are not helpful to the SharePoint connector. For example, redirections that occur when a SharePoint connection accesses a SharePoint server are bad news, because redirections of POST requests (which is what SOAP uses) are not well supported in the commons-httpclient library. Other examples of configuration problems include tunnels, because IIS may not properly authenticate connections that are made in this manner. Finally, sometimes the order in which SharePoint is installed and Windows security policies are modified can inadvertently break some specific SharePoint web services that the connection type relies on.

Subtle problems of these kinds may require saintly patience to work through, because you do not usually get very good feedback from SharePoint web services as to the underlying reason for the failure. The tools found to be helpful include the following:

- Wire-level debugging. In your logging.properties file, add `"log4j.logger.org.apache.http=DEBUG"` and/or `"log4j.logger.org.apache.http.wire=DEBUG"`, and restart the appropriate service to make the logging take effect. For versions of ManifoldCF after 2.7, you will need to edit logging.xml instead, and add the following to your loggers section:

```
<Logger name="log4j.logger.org.apache.http" level="DEBUG" additivity="false"><Appender-ref ref="MyFile" level="DEBUG" /></Logger> <Logger name="log4j.logger.org.apache.http.wire" level="DEBUG" additivity="false"><Appender-ref ref="MyFile" level="DEBUG" /></Logger>
```

- Windows event logs. Depending on the problem, the security log might be appropriate; for other problems, the application log would be better.
- Packet captures, using tcpdump and/or Wireshark. This helps finding the causes for unexpected redirections, or other http-gated issues.

SharePoint Web Services

In addition to these basic kinds of connectivity problems, some SharePoint instances can have configuration modifications that effectively break the web services that the ManifoldCF connector relies on. Debugging such changes is also a challenge, because the feedback you get from SharePoint's errors is very limited. If you get an Axis exception from the connector, make a note of the exception. If the error number is 1000 or 1010, the exception is being reported by the ManifoldCF SharePoint plugin. If you have read the how-to-build-and-deploy page, you will know that installation of the plugin is required in order to crawl SharePoint 2007 and SharePoint 2010 systems. The plugin provides a new web service, which supplies required functionality that is either missing or broken in SharePoint's standard distribution. The plugin does little real work on its own; mostly it calls either the existing SharePoint Permissions.asmx web service, or direct Microsoft.SharePoint.dll functionality.

In cases where you see an Axis exception, server-side logs are your best bet for more detailed exposition of the problem. But there is absolutely no guarantee that these logs will have enough information to permit a correct diagnosis. In that case, sometimes the only way to chase down the problem is to start by crawling a plain-vanilla, out-of-the-box installation of SharePoint, and then modifying the SharePoint instance step-by-step until you can find what modification was the one that breaks the ability to crawl. That is hopefully a last resort.

If you aren't seeing actual Axis exceptions, it is worthwhile first just trying to see if you can reach the services. Use Internet Explorer and browse to the following URLs:

```
http://<your-server-name>/<your-site-path>/_vti_bin/Permissions.asmx
http://<your-server-name>/<your-site-path>/_vti_bin/Lists.asmx
http://<your-server-name>/<your-site-path>/_vti_bin/DspSts.asmx
http://<your-server-name>/<your-site-path>/_vti_bin/Webs.asmx
http://<your-server-name>/<your-site-path>/_vti_bin/UserGroup.asmx
http://<your-server-name>/<your-site-path>/_vti_bin/Versions.asmx
```

... and, for SharePoint 2007 and SharePoint 2010:

`http://<your-server-name>/<your-site-path>/_vti_bin/MCPermissions.asmx`

For each one of these, you should see a well-formatted help message that includes a listing of all the methods available for the web service. If you don't see that, find out why, because the SharePoint connector won't be able to see them either.

Version of SharePoint

If you get crazy Axis errors when the connector is trying to use the MCPermissions plugin, it is conceivable that you have a non-standard version of SharePoint. To make sure you have a compatible version of SharePoint, the obvious thing to do is to find the version of your sharepoint.dll. The dll should be in one of the standard locations where assembly dlls are deployed on your server. The assembly name is Microsoft.SharePoint.dll - nothing else, not MicrosoftOffice, or anything else. There are a number of tools for determining the .NET version of such DLLs; here's a link that might help: <http://stackoverflow.com/questions/227886/how-do-i-determine-the-dependencies-of-a-net-application>. The ManifoldCF-Sharepoint-2013 plugin is built against:

```
<Reference Include="Microsoft.SharePoint, Version=15.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c, processorArchitecture=MSIL" />
```

The ManifoldCF-SharePoint-2010 plugin is built against:

```
<Reference Include="Microsoft.SharePoint, Version=14.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c, processorArchitecture=MSIL" />
```

... which can be found in the webservice/MCPermissionsService.csproj file in the source package for the service. The ManifoldCF-SharePoint-2007 plugin is, obviously, built against a different version:

```
<Reference Include="Microsoft.SharePoint, Version=12.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c, processorArchitecture=MSIL" />
```

The version of your Microsoft.SharePoint.dll **must** match one of these, or the plugin will not work. If you have an OEM distribution of SharePoint, I can well imagine this could be a problem. If there is a mismatch, you have no choice but to modify the plugin dependencies yourself, and build your own version of the MCPermissions assembly.