

How To: Capture XMLA interaction using Fiddler



How-To #:	12060713
Last Revised:	2012-06-08
Product(s):	All versions of SimbaO2X. Screenshots below are from SimbaO2X v4.4.17.30 and Fiddler v2.3.9.5.
Client Application(s):	All
Client Platform(s):	Windows
XMLA Source:	Any XMLA data source
XMLA Platform(s):	Any platform

Overview

Fiddler is an HTTP(S) debugging proxy which logs HTTP traffic between your computer and the Internet. It is able to capture, and even edit, the XMLA data stream between SimbaO2X and your XMLA server. This article uses Microsoft Analysis Services (MSAS) as a reference, however, you can configure Fiddler to capture SimbaO2X interaction with any XMLA data source.

How to Setup Fiddler

You can download Fiddler from: <http://www.fiddler2.com/fiddler2/>.

Install Fiddler on the machine with the XMLA server on it.

How to Configure Fiddler

The first step should be to configure Fiddler to operate as a reverse proxy.

Start by going to Tools → Fiddler Options → Connections and check the “Allow remote computers to connect” box. Close and reopen Fiddler.

Next, you have to edit the FiddlerScript rules. Click Rules → Customize Rules and add the following line of code inside the OnBeforeRequest handler:

```
if (oSession.host.toLowerCase() == "webserver:8888") oSession.host =  
"webserver:your_XMLA_port_number";
```

Thus, if your XMLA server is 192.168.222.65:80, then the corresponding line would be:

```
if (oSession.host.toLowerCase() == "192.168.222.65:8888") oSession.host =  
"192.168.222.65:80";
```

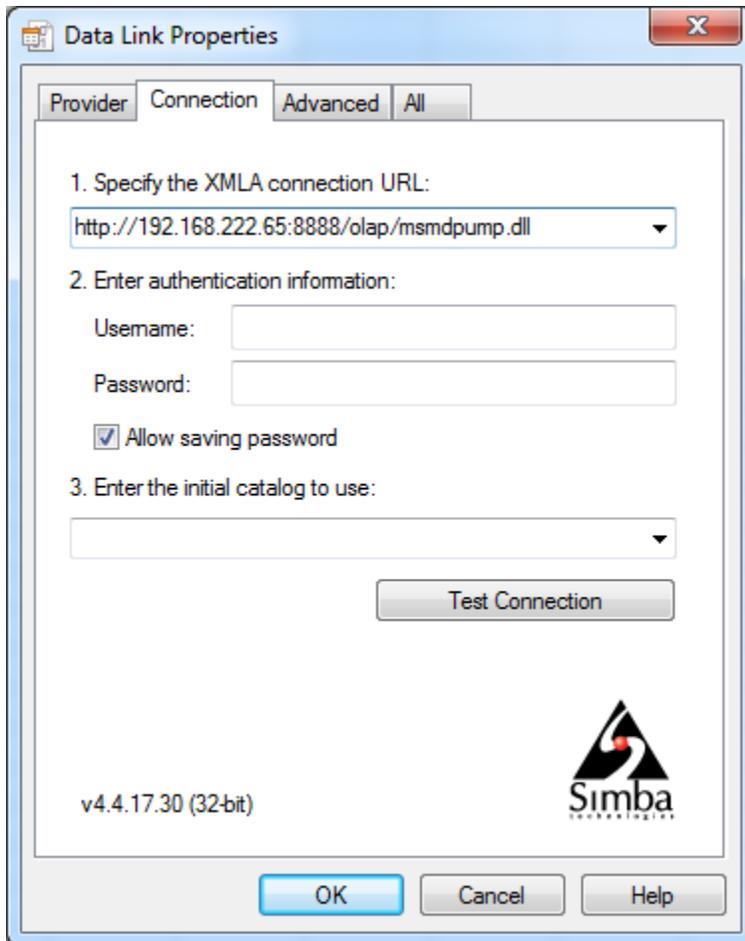
Note: The default port for Fiddler is 8888 – this can be changed as needed in Tools → Fiddler Options → Connections.

How to Use Fiddler with SimbaO2X

Once you have successfully configured Fiddler, connect to your data source from your desired application; for the purposes of this guide, we will use Microsoft Excel. Connect to your XMLA data source but with the port number that Fiddler is listening on (default 8888).

For example, if connecting to an MSAS server at <http://192.168.222.65:80/olap/msmdpump.dll>, then the URL you would enter into Excel would be <http://192.168.222.65:8888/olap.msmdpump.dll>.

The screenshot below shows the connection URL being used with the SimbaO2X Provider in Excel Data Link Properties pages.



You can proceed with SimbaO2X as usual. The interaction between SimbaO2X and the XMLA server will be captured in Fiddler.

How to set breakpoints in Fiddler

Go to Rules → Automatic Breakpoints to set generic breakpoints before requests, after responses or turn them off altogether. However, this may not be sufficient for debugging purposes because it is not possible to turn on both options at the same time with the GUI. To use Fiddler more effectively, it is important to use the GUI in conjunction with the in-program QuickExec command line to open up a plethora of new options.

The command line is the black box underneath the Web Sessions box in the bottom-left hand corner. Alt + Q will automatically shift focus to the QuickExec command line. Typing 'help' in the command line will also bring up the Fiddler QuickExec Reference.

To set more specific breakpoints, use the following commands:

- bpafter – break any response where the RequestURI contains the specified string; i.e. bpafter /favicon.ico
- bps – break any response where the status code matches; i.e. bps 404
- bpv or bpm – create a request breakpoint for the specified HTTP method; i.e. bpv POST
- bpu – create a request breakpoint for URIs containing the specified string; i.e. bpu /myservice.asmx

Calling any of the above commands with no parameter will disable the breakpoint.

How to edit a request/response

Once a breakpoint is hit, you can view the contents of the request (top pane) or response (bottom pane) through the Inspectors tab. The XML sub-tab will probably be of use in analyzing the XMLA since it hierarchizes everything. However, you cannot make changes in the XML sub-tab. To do so, you can navigate to the Raw sub-tab, make desired changes, then click the green Run to Completion button.

The screenshot shows the Fiddler interface with a request and its response. The top pane displays the raw request, and the bottom pane displays the XML response.

Request:

```
POST http://192.168.222.65/olap/msmdpump.dll HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; MS Web Services Client Protocol 2.0.50727.5448)
Content-Type: text/xml
Accept-Encoding: gzip
SOAPAction: "urn:schemas-microsoft-com:xml-analysis:Execute"
Host: 192.168.222.65
Content-Length: 1140
Expect: 100-continue

<?xml version="1.0" encoding="utf-8"?>
<Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.microsoft.com/analysis/2003/MSXMLA" >
  <Header>
    <Session SessionId="43E01F58-C11C-4961-B2CE-A4189950587D" mustUnderstand="1" xmlns="urn:schemas-microsoft-com:xml-analysis" />
  </Header>
  <Body>
    <Execute xmlns="urn:schemas-microsoft-com:xml-analysis">

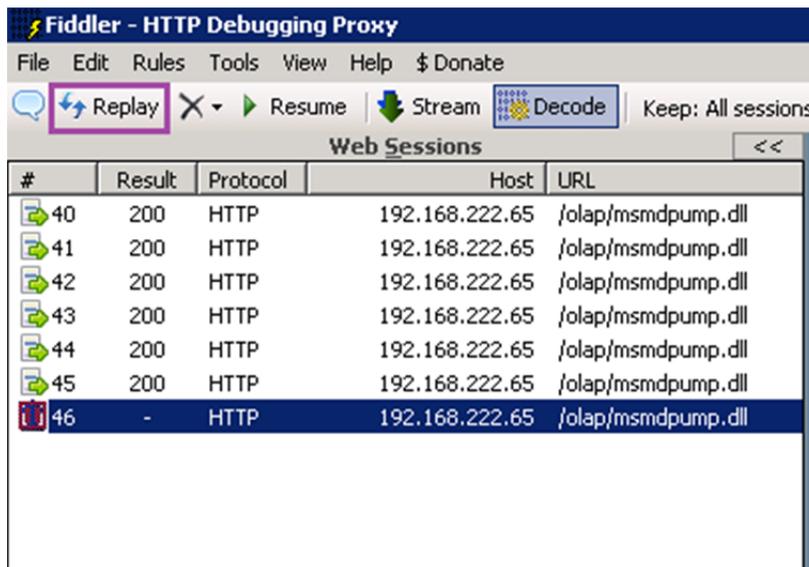
```

Response:

```
soap:Envelope [ xmlns:soap=http://schemas.xmlsoap.org/soap/envelope/ ]
  soap:Body
    ExecuteResponse [ xmlns=urn:schemas-microsoft-com:xml-analysis ]
      return
        root [ xmlns=urn:schemas-microsoft-com:xml-analysis:mddataset xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance xmlns:xsd=http://www.w3.org/2001/XMLSchema elementFormDefault=qualified xmlns=urn:schemas-microsoft-com:xml-analysis ]
          xs:schema [ targetNamespace=urn:schemas-microsoft-com:xml-analysis:mddataset elementFormDefault=qualified xmlns=urn:schemas-microsoft-com:xml-analysis ]
            xs:complexType [ name=MemberType ]
              xs:sequence
                <xs:any namespace="##targetNamespace" minOccurs="0" maxOccurs="unbounded" processContents="skip" xmlns:xs="http://www.w3.org/2001/XMLSchema" />
                <xs:attribute name="Hierarchy" type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" />
            xs:complexType [ name=PropType ]
              xs:sequence
                <xs:element name="Default" minOccurs="0" xmlns:xs="http://www.w3.org/2001/XMLSchema" />
                <xs:attribute name="name" type="xs:string" use="required" xmlns:xs="http://www.w3.org/2001/XMLSchema" />
                <xs:attribute name="type" type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema" />

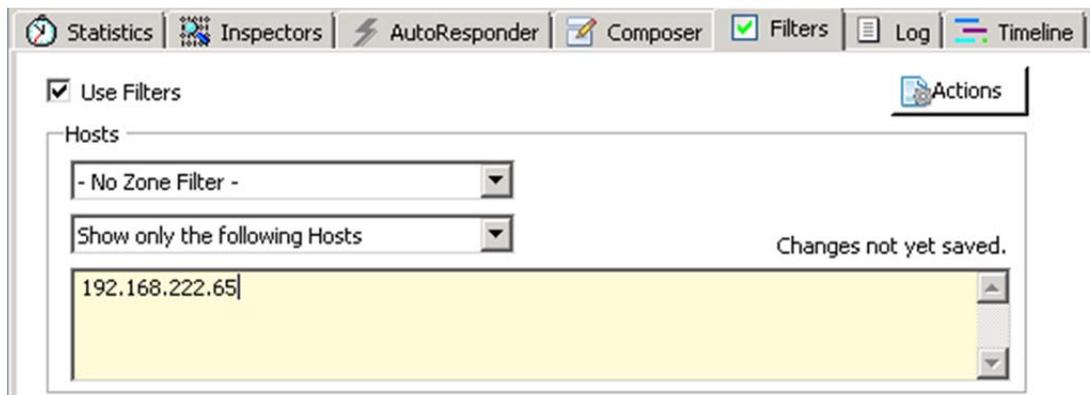
```

You may also find it necessary to resend a request; to do so, click on a request, then click the Replay button at the top.



Filters

If you find that there is a lot of traffic shown that you do not need, you can set up some filters to show only the information you want to see. Click on the Filters tab to set up filters. The Hosts section is where this feature will probably be of most use. Below is an example of a simple filter:



For additional information, refer to <http://www.fiddler2.com/fiddler2/help/Filters.asp>.

Tips and shortcuts

- Click the “Decode” button at the toolbar up top to automatically decode requests/responses
- It may be difficult to read XML in the Raw sub-tab; if that is the case, make a new XML file in Visual Studio and copy and paste the XML from Fiddler to there. Visual Studio will hierarchize your XML much like the XML sub-tab does. After making necessary changes, copy and paste back to Fiddler.