

Session management with Cookies

Just thought I'd share this bit 'o knowledge with the community. When building the VI Toolkits for .NET and Java I created methods for connecting to VI with an existing session cookie (like Perl and PowerShell). In .NET that looks like this:

```
// The next line is self-explanatory.
m_vim_svc_content =
m_vim_svc.RetrieveServiceContent( vim_svc_ref );

if ( GetOptionValue( "sessionfile" ) != null )
{
Trace.TraceInformation( "Connecting with session file" );

BinaryFormatter bf = new BinaryFormatter();
Stream s = File.Open(
GetOptionValue( "sessionfile" ), FileMode.Open );
Cookie c = bf.Deserialize( s ) as Cookie;
s.Close();
m_vim_svc.CookieContainer.Add( c );
}
```

In the above example the session content is retrieved and then the Cookie object is deserialized from a text file where it was stored (you can store just the cookie value and create a new Cookie object as well, but storing the object by serializing it to a file means that the VI Toolkit for Windows (PowerShell) can use it as well).

In Java however, the same logic does not work. You have to do this:

```
m_vim_svc = vim_svc_loc.getVimPort( new java.net.URL( getVIUrl() ) );
VimBindingStub vbs = ( VimBindingStub ) m_vim_svc;
vbs.setMaintainSession( true );
if ( getOptionValue( "sessionfile" ) != null )
{
/*
* Load the session cookie. It is important that when using Axis,
* the session cookie is loaded BEFORE the service content is
* retrieved, otherwise the session will not be authenticated.
*/
String session_file = getOptionValue( "sessionfile" );
BufferedReader br =
new BufferedReader( new FileReader( session_file ) );
String session_cookie = br.readLine();
br.close();
```

Session management with Cookies

```
vbs._setProperty(  
org.apache.axis.transport.http.HTTPConstants.HEADER_COOKIE,  
session_cookie );  
m_vim_svc_content = m_vim_svc.retrieveServiceContent( vim_svc_ref );  
}  
else  
{  
m_vim_svc_content = m_vim_svc.retrieveServiceContent( vim_svc_ref );  
m_vim_svc.login( m_vim_svc_content.getSessionManager(),  
getOptionValue( "username" ), getOptionValue( "password" ),  
null );  
}
```

In Java you must set the cookie BEFORE you retrieve the service content. However, you still must retrieve the service content before you login via username and password.

I am putting this out there just in case one or two of you trip over the same issue.

Many thanks to Steve Jin @ VMware for pointing me in the right direction on this.

Let's say there large number of concurrent client requests to ESX hostd coming from a mgmt application.

The app is running inside a VM and connects to the same ESX server that hosts the VM.

Each request makes similar VI SDK java api calls.

Is there a limit on the max number of concurrent sessions that are allowed? We see messages:

```
2008-08-26 14:45:36.833 'SoapAdapter' 22461360 error SOAP session count limit reached
```

Another question, is if we allow one client thread to establish a session cookie and other threads to reuse the same session cookie, would this give a substantial performance gain on the ESX server? (session pooling)

Is connection pooling possible?

The documentation states that the session limit is set to 100.

This document was generated from the following thread: [!! Session management with Cookies !!](#)