Get started with VI Java API

Steve JIN, 2008-05-22

Highlights of the key benefits using this API package:

- Enables OO programming with a well defined managed object model
- Reduces the need to use ManagedObjectReference
- Hides the complexity of the PropertyCollector
- Provides necessary utility classes to simplify VI SDK web interfaces
- Leverages current VI SDK web services interface while keeping it intact
- Very small footprint of the API itself

To get started with it, just go to the download the latest jar file and include it in the classpath of your existing VI SDK projects.
http://sourceforge.net/project/showfiles.php?group_id=228007

For new projects, please refer to Developer's Setup Guide: http://www.vmware.com/support/developer/vc-sdk/visdk25pubs/visdk25setupguide.pdf

Now, you can new a Java project and kick off your first HelloVM as the following.

```
package com.vmware.vim25.mo.samples;

import java.net.URL;
import com.vmware.vim25.*;
import com.vmware.vim25.mo.*;
import com.vmware.vim25.mo.util.*;

public class HelloVM
{
        public static void main(String[] args) throws Exception
        {
            CommandLineParser clp = new CommandLineParser(new OptionSpec[]{}, args);
            String urlStr = “https://esx-server/sdk”;
            String username = "username";
            String password = "password";

            ServiceInstance si = new ServiceInstance(new URL(urlStr), username, password,
true);
            Folder rootFolder = si.getRootFolder();
            VirtualMachine vm = (VirtualMachine) new
InventoryNavigator(rootFolder).searchManagedEntities("VirtualMachine")[0];

            VirtualMachineConfigInfo vminfo = vm.getConfig();
            VirtualMachineCapability vmc = vm.getCapability();

            System.out.println("Hello " + vm.getName());
            System.out.println("GuestOS: " + vminfo.getGuestFullName());
            System.out.println("Multiple snapshot supported: " +
        vmc.isMultipleSnapshotsSupported());

            si.getServerConnection().logout();
        }

}
```

To run this program, you will need to change the urlStr, username, and password to your own environment. The urlStr can be pointing to either an ESX server or a VC server.

When you are done with your first HelloVM, let's talk a little more about general ways to use this APIs:

1. Always starts with a ServiceInstance with URL/username/password, or URL/sessionID:

```
ServiceInstance si = new ServiceInstance(new URL(urlStr), username, password, true);
```

2. From the ServiceInstance object, you can:

*Get its properties like capability:

```
Capability cap = si.getCapability();
```

*Get root folder object of the inventory tree.

```
Folder rootFolder = si.getRootFolder();
```

Since property content (ServiceContent) only holds AboutInfo and ManagedObjectReferences to all different manager, we move the these info up by provide methods like getAboutInfo and various get() methods to get manager objects directly.

*Get all different “manager” object or SearchIndex object:

```
SearchIndex searchIndex = si.getSearchIndex();
EventManager em = si.getEventManager();
```

3. With the manager objects returned from ServiceInstance object, you can do many different things. For example: list all the events with a VM by using EventManager.

4. From the root Folder or any ManagedEntity in the inventory,

- You can navigate to the sub-nodes in the inventory tree:

```
rootFolder.getChildEntity();
```

- You can use InventoryNavigator class to search based on your criteria.
- All the items you see in the inventory tree are of type ManagedEntity.
- You can test the exact type of a ManagedEntity

  ```
  if(me instanceof VirtualMachine){ }
  ```

- You can call the methods defined on the exact subtypes of the ManagedEntity.

```
Task task = ((VirtualMachine)me).powerOffVM_Task();
```

To get help, you can find exact description of each managed object and its properties/methods in the VI SDK API reference. Just look for the managed object type and its method with the same name. Please skip the first _this parameter and the rest are the same except the difference of real type and ManagedObjectReference. The description should be applicable despite the difference.

You will find some samples inside the com.vmware.vim25.mo.samples package. As you will find out, the samples using this APIs are much shorter and more readable than the ones using VI SDK web service interfaces.

Please note that this is an open source project and not supported by VMware. But community members can help each other via bug report, forum, mailing list. Also please let us know your wish list via feature request.