QAD Inc.
100 Innovation Place
Santa Barbara, CA
93108 USA

Tel +1 805 566 6000
Fax +1 805 565 4202
http://www.qad.com

Migrating Zerto VPG's from one vCenter to a new vCenter

Contents

QAD Inc.
100 Innovation Place
Santa Barbara, CA
93108 USA

Tel +1 805 566 6000
Fax +1 805 565 4202
http://www.qad.com

Purpose

This document will explain the process of migrating Source VPG's from one vCenter to a new vCenter by manipulating the VPG XML export. Zerto has told us this is impossible and does not offer any tools to automate this task, which is beyond frustrating when there are hundreds of VM's and VPG's to migrate.

The current solution offered by Zerto is to delete all the VPG's while retaining the data at the DR site, create new VPG's manually and use the data for pre-seeding the VPG's. When you're talking about 50 VPG's with 100's of VM's and at least quadruple that amount of individual VMDK's to pre-seed, manual migration could easily take 100's of man-hours to complete.

After much back-and-forth with Zerto support, searching, and trial & error, we have successfully performed a migration of VPG's from one vCenter to a new one. All the information needed can be pulled from the vCenter databases with a little Perl Fu.

Assumptions & Prerequisites

- Live (production) vCenter is being replaced
  - The new vCenter and associated ZVM for the new vCenter should already be installed and active.
  - Communication between the old and new source vCenters, ZVM's and the DR site should already be verified working
  - Networks and distributed switches must be identical on the old and new vCenter servers.
- The DR vCenter and ZVM must remain the same
- vCenter/vSphere versions do not seem to matter
  - We went from v5.1.0 to v6.0.0 with no issues
- DRS temporarily disabled during this process to prevent VM's from moving around.
- ESXi hosts containing the protected VPG will be migrated to the new vCenter
  - All protected VM's residing on each ESX host will also be migrated
  - It is likely possible to perform this procedure with a completely different ESXi host by editing the XML accordingly, but we have not had the need to test this theory
- Sane/safe settings for Bandwidth Throttling have been configured in the Zerto GUI
- ALL protected VM's for a given VPG must exist on the same ESXi server being migrated
  - Simply re-locate any VM's in vCenter to fix any discrepancies
  - DRS must be disabled to prevent separation of VM's in the VPG's
- RDP access with admin privileges to the ZVM for both the existing vCenter and new vCenter
  - moRefFinder.pl, PowerCLI, VMware vSphere PowerCLI and VMware vSphere Perl SDK installed
    - Download moRefFinder.pl from here: http://www.virtuallyghetto.com/2011/11/vsphere-moref-managed-object-reference.html
- Admin permissions for the Zerto Diagnostics utility on both ZVM's
- Notepad++ (32-bit!) with "XML Tools" plugin installed
  - As of this writing, the 64-bit version does not have plugin support
- This has only been tested on Zerto 5.x

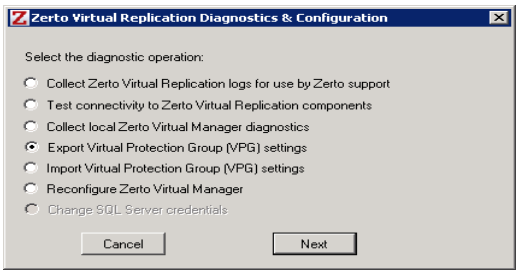
QAD Inc.
100 Innovation Place
Santa Barbara, CA
93108 USA

Tel +1 805 566 6000
Fax +1 805 565 4202
http://www.qad.com

Data Collection - Old vCenter

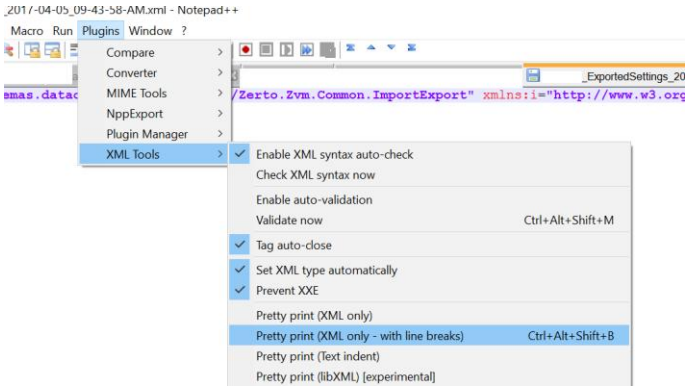
Export VPG XML from the Source ZVM

First, login to the ZVM for the old source vCenter and then launch the Zerto Diagnostics Utility. Select the export option, follow the prompts and save the file.



Re-format XML file.

Save a copy of this XML file. Open the XML export copy in Notepad++ and then use the XML Tools plugin "Pretty print" with line breaks option. This will take some time if you have a large environment, so just be patient:



Once complete, it should look like this:

QAD Inc.
100 Innovation Place
Santa Barbara, CA
93108 USA

Tel +1 805 566 6000
Fax +1 805 565 4202
http://www.qad.com

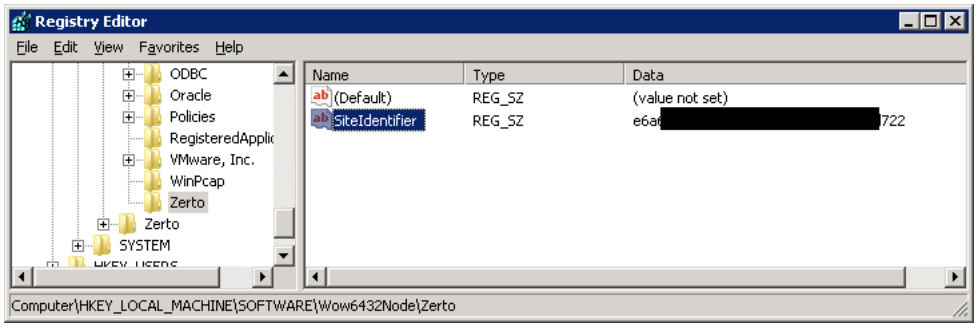
The XML export from Zerto Diagnostics contains several unique identifiers (moRef's) for several vCenter components that must be altered before the XML file can be used for importing in the new vCenter implementation.

1. ## Zerto Site ID (SiteGuid) & Site Name (LocalSiteName)

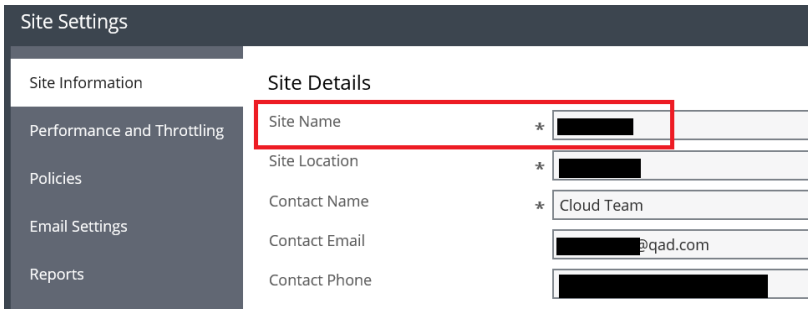
   The current Site ID and Site Name occur once at the end of the exported XML file. They are tagged as `SiteGuid` and `LocalSiteName` respectively:

```
45754     <Version xmlns:a="http://schemas.datacontract.org/2004/07/Zerto.Zvm.Common">
45755         <a:Branch>athena_u2p1</a:Branch>
45756         <a:Build>2</a:Build>
45757         <a:Major>5</a:Major>
45758         <a:Minor>0</a:Minor>
45759         <a:Update>21</a:Update>
45760         <a:HotFixes/>
45761     </Version>
45762     <LocalSiteId xmlns:a="http://schemas.datacontract.org/2004/07/Zerto.Zvm.Common.SiteControl">
45763         <a:SiteGuid>e6a[redacted]4722</a:SiteGuid>
45764     </LocalSiteId>
45765     <LocalSiteName>[redacted]</LocalSiteName>
45766 </ExportedSettings>
```

   `SiteGuid` can be found in the registry of the ZVM for each vCenter.
   Find it under `HKLM\SOFTWARE\Wow6432Node\Zerto\SiteIdentifier`



   `LocalSiteName` is the descriptive name configured under "Site Name" in the Zerto GUI. Find it under "Site Settings" -> "Site Information".



   Replace the current `SiteGuid` and `LocalSiteName` in the XML export with the values from the new ZVM and new Site Name, as configured in the Zerto GUI. `LocalSiteName` must match the "Site Name" exactly.

2. ## Obtain `InternalVmName` for every protected VM

   `InternalVmName` is the internal, unique moRef ID for a given protected VM. (One of the reasons Zerto

QAD Inc.
100 Innovation Place
Santa Barbara, CA
93108 USA

Tel +1 805 566 6000
Fax +1 805 565 4202
http://www.qad.com

asserted that this type of migration was impossible is that the vCenter assigns new moRef ID's when an ESX server and associated VM's are imported into the vCenter.)

This is where moRefFinder.pl comes into play. After installing the vSphere Perl SDK on the ZVM, open a command prompt and gather this information from the OLD vCenter before the migration.

```
C:\scripts>moRefFinder.pl --server OLDVCENTER --username USERNAME --type vm --name SOMEVMNAME
Password: xxxxxx

vCenterInstanceUUID: 9b47xxxx-xxxx-xxxx-xxxx-xxxxxxxx50b5
EntityName: SOMEVMNAME
MoRefID: vm-608912
```

The VM name for moRefFinder.pl is CaSE SEnSiTIVE!
As you can see from the output, the `InternalVmName` for `SOMEVMNAME` in this test is `vm-608912`. Save this information for later. It will be used for a search and replace in the XML export once the new moRef ID from the new vCenter is obtained. It's not too difficult to whip up a modified version of moRefFinder.pl that will query multiple servers in an array. (Look for a sanitized version of the one Aaron created in the Zerto forums.)

3. ServerGuid

`ServerGuid` **is the GUID for VM's host vCenter. It is found throughout the XML file, but we are ONLY interested in the one listed in the context of the** `InternalVmName` **section:**

```
<b:VMIdentifier
xmlns:c="http://schemas.datacontract.org/2004/07/Zerto.Zvm.Common.VirtualizationManager.VCenter">
   <c:InternalVmName>vm-608912</c:InternalVmName>
   <c:ServerIdentifier>
      <c:ServerGuid>9b47xxxx-xxxx-xxxx-xxxx-xxxxxxxx50b5</c:ServerGuid>
   </c:ServerIdentifier>
</b:VMIdentifier>
```

The `moRefFinder.pl` script output above happens to include the `ServerGuid` used in the XML file, which is called `vCenterInstanceUUID` in the output of the perl script. Save this for search and replace later.

```
vCenterInstanceUUID: 9b47xxxx-xxxx-xxxx-xxxx-xxxxxxxx50b5
```

4. OwnersGuid

This information is located just under the Name section for every VPG in the XML file. It is the same GUID for all VPG's:

```
<b:Name>Lab Test</b:Name>
<b:OwnersIdentifier xmlns:c="http://schemas.datacontract.org/2004/07/Zerto.Zvm.Common.SiteControl">
   <c:OwnersGuid>c2e9xxxx-xxxx-xxxx-xxxx-xxxxxxxx26d6</c:OwnersGuid>
</b:OwnersIdentifier>
```

QAD Inc.
100 Innovation Place
Santa Barbara, CA
93108 USA

Tel +1 805 566 6000
Fax +1 805 565 4202
http://www.qad.com

Make note of this GUID.

Thus far we've been unable to locate what this parameter actually references in the vCenter database. Using `moRefFinder.pl` to query every known component was a futile effort. It's quite possible the data was overlooked in the massive sea of information. Perhaps the Zerto devs can explain this parameter.

The easiest way to obtain this information is to create a NEW VPG on the NEW vCenter, then use Zerto Diagnostics on the new ZVM to export the VPG data. Use the smallest, previously unused VM possible, and make sure to select thin provisioning to decrease initial sync time. Search this exported XML file for `OwnersGuid` and you will have the replacement value needed. Delete the temporary VPG when finished.

5. ### ServiceProfileIdentifier

We ran across issues with this parameter in our initial testing. The XML would not import due to `ServiceProfileIdentifier` being defined in the XML file, but not in the new ZVM:

```
<b:ServiceProfileIdentifier
xmlns:c="http://schemas.datacontract.org/2004/07/Zerto.Zvm.Common.ServiceProfiles">
   <c:InternalId>d02exxxx-xxxx-xxxx-xxxx-xxxxxxxx083a</c:InternalId>
</b:ServiceProfileIdentifier>
```

This is still a head scratcher, as the "Service Profile" refers the SLA settings in the VPG itself, so it should already be defined in the VPG XML. Instead, it appears to have an internal GUID stored in the vCenter DB.

We were in the middle of a change window and didn't have time to chase this down, so we opted to just revert it to default settings and update the VPG later. The previously mentioned VPG export from the new vCenter provided a suitable replacement value. Replace the entire `ServiceProfileIdentifier` node listed above with what is found in the new VPG export XML file. In our case, it was simply "nil"ed out with no `InternalId`:

```
<b:ServiceProfileIdentifier i:nil="true"
xmlns:c="http://schemas.datacontract.org/2004/07/Zerto.Zvm.Common.ServiceProfiles"/>
```

Migration

Before migration of the ESX hosts and protected VM's to the new vCenter, the following will need to be performed on the old vCenter and old ZVM:

1. ### Export VPG data from the Zerto web interface on the DR site

This is performed on the VPG tab by clicking on the EXPORT link. This information will only be used in case the XML import fails for some reason and it is required to manually re-create VPG's using VMDK pre-seed locations.

This MUST be run on the DR site due to a bug in Zerto. (As of Zerto v5.0, the target data store information is missing from the CSV exported from the source site.)

QAD Inc.
100 Innovation Place
Santa Barbara, CA
93108 USA

Tel +1 805 566 6000
Fax +1 805 565 4202
http://www.qad.com

2. **Delete all VPG's containing VM's being migrated**
   Make sure to retain the disks at the DR site, as they will be used for pre-seeding during the XML import.
3. **Delete the ZVRA associated with any ESX hosts being migrated**
   This is done in the Zerto interface. Make note of the settings for each ZVRA so the information can be used when re-creating them in the new site.
4. **Migrate ESX hosts and protected VM's**
   Assuming everything is setup correctly, this should be as simple as disconnecting the ESXi host from the old vCenter and then re-connecting it in the new vCenter.

   Don't forget about the prerequisite that all VPG's associated with an ESXi server being migrated must have ALL Zerto protected VM's in those VPG's live on the ESXi host being moved.
5. **Re-create ZVRA's for each ESX host on the new site**

Data Collection - New vCenter

Once the ESX and protected VM's have been migrated to the new vCenter, gather the following information from the new site:

- `InternalVmName` for every protected VM
  This value will have changed due to the new vCenter assigning a new, unique identifier for each VM imported along with the ESXi server. Follow the same `moRefFinder.pl` procedure above, using the new vCenter name when connecting.
- `ServerGuid`
  You'll get this when performing the previous step
- `OwnersGuid`
  You should already have this from step 4 in data collection from the old vCenter.

Search and Replace

Armed with the data collection from old and new sites, you can now edit the XML export and then search and replace:

- `SiteGuid` from step 1
  MAKE SURE you only replace the `SiteGuid` that exists at the BOTTOM of the XML file. There is another one at the top of the file that should not be changed.
- `InternalVmName` from step 2.
  It SHOULD be safe to perform a global search and replace of this value (no XML tags). Make sure the value of `InternalVmName` doesn't exist anywhere outside of `<C:InternalVmName>` XML tags.

  Rinse and repeat for every VM in the XML file.
- `ServerGuid` from step 3
  This is found throughout the XML file, but we are ONLY interested in the one listed in the context of the `InternalVmName` section.

  It's safe to global search and replace by the full GUID, without the XML tags.

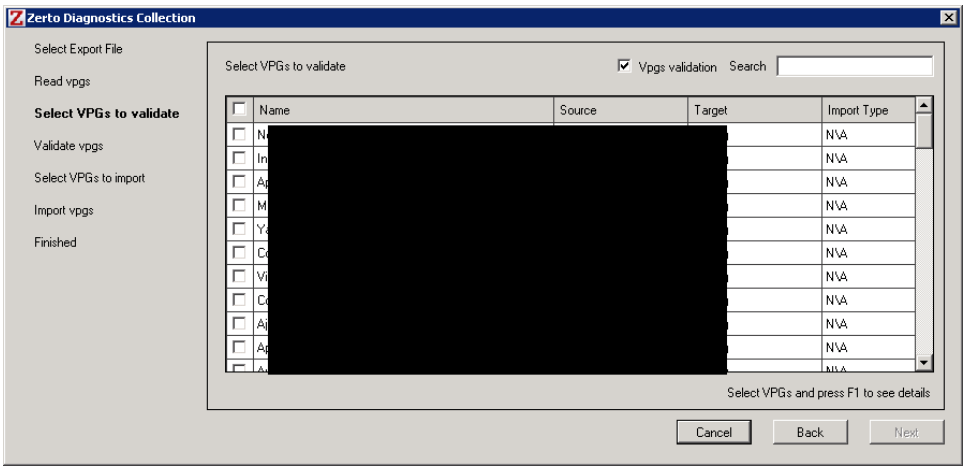
QAD Inc.
100 Innovation Place
Santa Barbara, CA
93108 USA

Tel +1 805 566 6000
Fax +1 805 565 4202
http://www.qad.com

- `OwnersGuid` from step 4
  It's safe to global search and replace by the full GUID, without the XML tags.
- `ServiceProfileIdentifier` from step 5
  You'll need to manually search and replace, as the entire node needs to be altered.

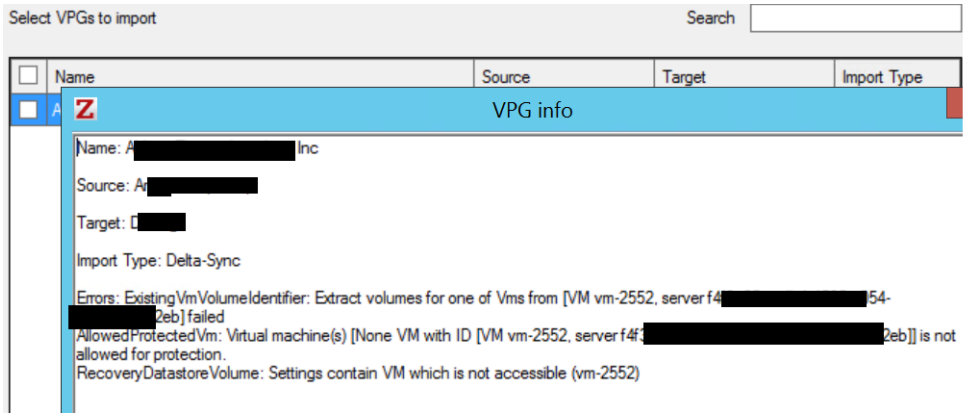
Import XML

Once the search and replace is complete, save the XML file on the new ZVM. Use Zerto Diagnostics on the ZVM for the new vCenter to import the VPG's that were deleted for the migration. It's best to import only 3 or 4 at a time to prevent overloading Zerto and vCenter:



The validation process can take some time if you have a lot of VPG's, so be patient. If there are any issues with the XML file you've modified, Zerto should let you know and prevent the import.

If there are no errors and the VPG selection is grayed out, just click on the VPG and press F1 to see the error message:



In that case, the moRefId for the VM was not updated in the XML file with the new ID from the new vCenter, so Zerto is unable to locate one or more VM's contained in the VPG.

Don't forget to re-enable DRS when the import and delta syncs have completed.

QAD Inc.
100 Innovation Place
Santa Barbara, CA
93108 USA

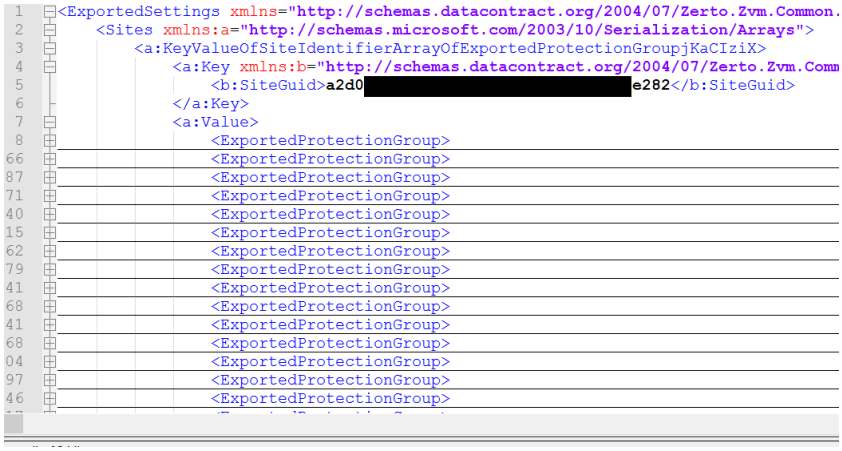
Tel +1 805 566 6000
Fax +1 805 565 4202
http://www.qad.com

Large Environment/Selective VPG Importing

If your environment is huge like ours, you probably won't be importing everything all at once. We performed a migration of two or three ESXi servers at a time to minimize risk and avoid downtime.

Since Zerto Diagnostics only has the ability to export ALL VPG's, this can be a problem if you only plan on moving a few at a time. Thankfully, Notepad++ has the excellent "XML Tools" plugin that makes this much easier to deal with.

Once you've opened the XML in Notepad++ and used XML Tools to Pretty Print, you can use the "Fold All" (ALT + 0) command from the View menu to collapse all nodes. Then just click on the first five "+" signs on the left. This will expand the nodes to where you should then see multiple `ExportedProtectionGroup` nodes under the `<a:Value>` node:

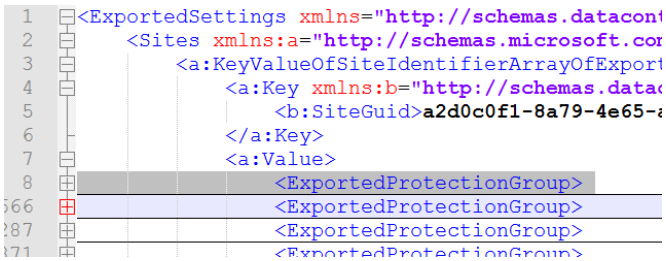


Each of the `ExportedProtectionGroup` nodes contains all the information for a single VPG. If you click on the next three "+" signs, you should now see the VPG name for that node:

Once you've located the VPG you DO NOT need to migrate, collapse the node, click at the beginning of the line, hold down SHIFT, press the down arrow once, and then press DELETE or BACKSPACE to remove the node from the XML:

QAD Inc.
100 Innovation Place
Santa Barbara, CA
93108 USA

Tel +1 805 566 6000
Fax +1 805 565 4202
http://www.qad.com



Rinse and repeat with any other VPG's not being migrated at this time.

moRef Finder Script

Here is a sanitized version of the moRef Finder script used for querying the moRef ID for each VM. It is based on William Lam's moRefFinder.pl script, but was altered to suit our needs.

```
#! perl
# Perl script to grab moRefId's from vCenter for multiple VM's at once.
# (See "entityMapping" below for other entities that can be queried.)
#
# ** Based on the excellent moRefFinder.pl script by William Lam; altered to suit our needs.
# ** https://github.com/lamw/vghetto-scripts/blob/master/perl/moRefFinder.pl
#
# Author: Aaron Ciarlotta, QAD - 2017/02
# Requires VMware vSpherePowerCLI and VMware vSphere Perl SDK
# Copy and run on the ZVM for your environment
#
# Just update the @servers variable below with the VM server names to query, save, then run.
# You will be prompted for the rest of the information required.
#
#! perl
use strict;
use warnings;
use VMware::VILib;
use VMware::VIRuntime;
use Term::ReadKey;
use Data::Dumper;
use Term::ANSIColor;
no warnings 'uninitialized';

# Server names are CaSE SenSiTiVE!
my @servers=('servername01', 'SomeOtherServer','myserver122');


# Do not edit anything below this line:
$| = 1;

if(@ARGV > 0) {
        print color('bold red');
        print "\n\nERROR: Do not specify any parms, you will be queried!\n\n";
        print color('reset');
        exit;
}

my $type="vm";
print "Enter vCenter Host: (ex: vcenter01.yourdomain.com) ";
chomp(my $vcenter = <STDIN>);
print "Enter vCenter username (ex: YourLogin\@yourdomain): ";
chomp(my $username = <STDIN>);
print "Enter password for $username on $vcenter: ";
ReadMode('noecho');
chomp(my $password = ReadLine(0));
```

QAD Inc.
100 Innovation Place
Santa Barbara, CA
93108 USA

Tel +1 805 566 6000
Fax +1 805 565 4202
http://www.qad.com

```
print "\n\nAttempting to connect to vCenter '$vcenter' as '$username'...\n";

if(!Util::connect("https://$vcenter/sdk/webService",$username,$password)) {
        print color('bold red');
        print "\n\nERROR: Unable to connect to vCenter\n\n";
        print color('reset');
        exit;
}

my %entityMapping = (
        'vm' => 'VirtualMachine',
        'host' => 'HostSystem',
        'cluster' => 'ComputeResource',
        'datacenter' => 'Datacenter',
        'rp' => 'ResourcePool',
        'network' => 'Network',
        'dvs' => 'DistributedVirtualSwitch',
        'folder' => 'Folder',
        'vapp' => 'ResourcePool',
        'datastore' => 'Datastore'
);

print "\nConnected! Starting queries...\n";

foreach my $server (@servers) {
        local $| = 1;
        print "\n$server\n";
        &getMoRef($type,$server);
}


Util::disconnect();

sub getMoRef {
        my ($type,$name) = @_;

        if(!$entityMapping{$type}) {
                print "Error: Invalid Entity Type: $type\n";
                Util::disconnect();
                exit 1;
    }

        my $entity = Vim::find_entity_view(view_type => $entityMapping{$type}, filter => {"name" => $name},
properties => ['name']);
        my $ServerName = $entity->{'name'};
        if(defined($ServerName) and length($ServerName)) {
                if(Vim::get_service_content()->about->apiType eq "VirtualCenter") {
                        print "vCenterInstanceUUID (ServerGuid): " . Vim::get_service_content()->about-
>instanceUuid . "\n";
                }
                print "EntityName: " . $entity->{'name'} . "\nMoRefID: " . $entity->{'mo_ref'}->value . "\n\n";
        } else {
                print "Server '$name' was not found in vCenter '$vcenter'\n\tServer names are CaSE SenSITiVe!\n";
        }
}


sub listVMs {
        my ($host_view) = @_;

        my $vms = Vim::get_views(mo_ref_array => $host_view->vm, properties => ['name']);
        foreach(@$vms) {
                my $vm_mo_ref_id = $_->{'mo_ref'}->value;

                print "Virtual Machine: ".$_->{'name'}."\n";
                print "VMID: " . $vm_mo_ref_id . "\n";
                print "\n";
        }
}
```