API - Deleting Multi-Level Resellers, Power Users, and their Associations

The following example demonstrates how to delete multi-level reseller(s) and Power User(s) and their associations.

A PHP file called `Delete_Multi_Level_Reseller_And_Power_User_And_Their_Associations.php` can be found in `<installdir>/apisamples`. Read more in Accessing Example API Functions .

The PHP script searches and deletes objects in the system and prints the following result messages on the screen:

- "Successfully retrieved all the users" / "No user found with the specified ID" / "Failed to get the specified user"
- "Successfully retrieved all the agents" / "No Agents are owned by the specified user" / "Failed to get the specified agent"
- "Successfully retrieved all the volumes" / "No Volumes are exclusively assigned to the specified user" / "Failed to get all volumes"
- "Successfully retrieved all the diskSafes" / "No Disksafes are associated with the agents owned by the user" / "Failed to get all Disk Safes"
- "Successfully retrieved all the policies" / "No Policies are associated with the disksafe assigned to the agents owned by the user" / "Failed to get all the policies"
- "Failed to delete all policie(s)" / "All policie(s) deleted successfully"
- "Failed to delete all Disk Safe(s)" / "All Disk Safe(s) deleted successfully"
- "Failed to delete all Sub-User(s)" / "All Sub-User(s) deleted successfully"
- "Failed to delete all Agent(s)" / "All Agent(s) deleted successfully"
- "Failed to delete all Volume(s)" / "All Volume(s) deleted successfully"
- "Failed to delete Power user" / "Power user deleted successfully"

Sequence of Automated Actions

The following steps can be accomplished by using this script:

1. Find a User with the specified ID. If a User with this ID does not exist, then exit or save the User ID.
2. Find Sub-Users for the specified Power User, if there are any. Then save the found Sub-User IDs.
3. Find Agents which have the specified User ID as their owner, and then save the found Agent IDs.

4. Find Disk Safes which have the specified Agent ID, and then save the Disk Safe IDs.
5. Find Policies associated with the found Disk Safes. Then save the found Policies' IDs.
6. Delete found Policies.
7. Delete found Disk Safes.
8. Delete found Agents.
9. Delete found Volumes.
10. Delete found User.
11. Delete found Sub-Users.

How to Fulfill Appropriate Actions in CDP User Interface

Below, you can find the steps to take in the program user interface in order to perform the same actions as the script. Also see the accompanying screen-shots illustrating the scripts for each step.

Defining server configuration variables | Retrieving a User | Retrieving Agents | Retrieving Volumes | Retrieving Disk Safes | Retrieving Policies | Deleting the Policies, Disk Safes, and User | Deleting Agent | Deleting Volumes

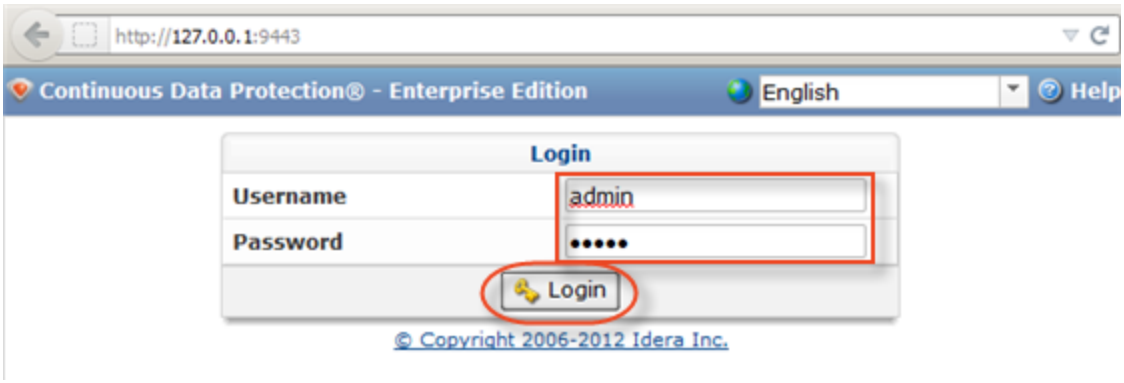
Defining server configuration variables

```
date_default_timezone_set('America/Chicago');


########====CDP Server Configuration Start====########
#set CDP server host name
$HOST="127.0.0.1";
#set CDP server to access API
$PORT="9443";
#set CDP user
$USER="admin";
#set CDP user password
$PASS="admin";
########====CDP Server Configuration End====########



########====SetUser ID to be deleted Start====########
$ID = "dcad5f7c-27d0-46d0-86b2-24741b6bf25f";
########====SetUser ID to be deleted End====########
```



Retrieving a User

```
########====Get User Start====########

 try{
   $userClient = new soapclient("https://$HOST:$PORT/User?wsdl",
       array('login'=>"$USER",
       'password'=>"$PASS",
       'trace'=>1,
    'cache_wsdl' => WSDL_CACHE_NONE,
    'features' => SOAP_SINGLE_ELEMENT_ARRAYS
        )
 );
 # get all the Users
 $allUsers=$userClient->getUsers();
 echo "Successfully retrived all the users \n";
 $selectedSubUserIDs = array();
 foreach($allUsers->return as $tmp) {
  // check to see if the specified user id matches the any of the retrived user ids
  if (isset($tmp->id) && $tmp->id == $ID){
   // if it matches store the id
   $selectedUserID =$tmp->id;
  }
  else if (isset($tmp->adminIDs)){
   $tmpAdminIDArray = array($tmp->adminIDs);
   if (in_array($ID , $tmpAdminIDArray)){
    array_push($selectedSubUserIDs, $tmp->id);
   }
  }
 }

 // if nothing matches then exit the program
 if (!isset($selectedUserID)){
  echo " No user found with the specified ID $ID \n" ;
  exit(1);
 }
}
catch (SoapFault $exception)
{
 echo "Failed to get the specified user \n";
 echo $exception;
 exit(1);
}

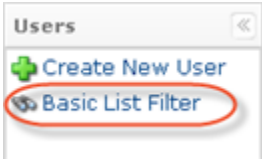
########====Get User End====########
```

To find a User by username, follow the instructions below.

1. Click on "Users" in the Main Menu to access the Users page.

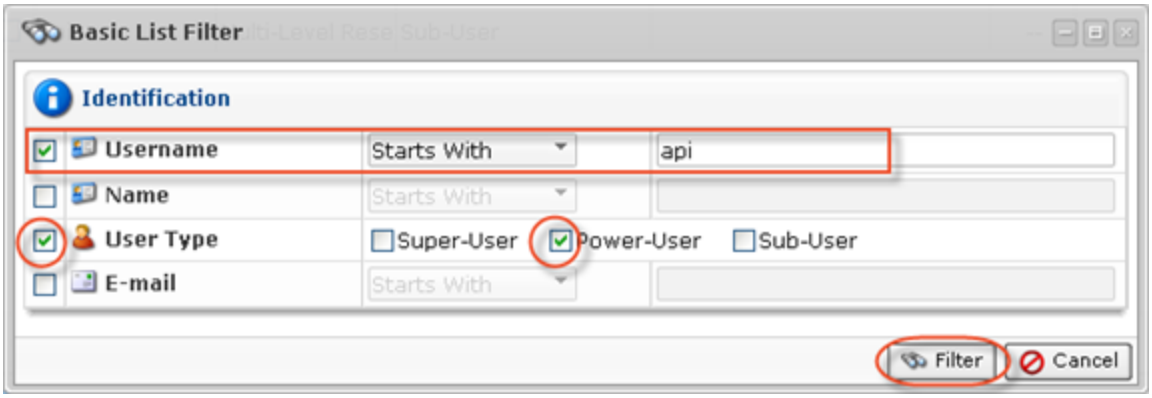


2. Click on "Basic List Filter" located in the Users sub-menu.



3. Enter a username, select the "Power-User" check-box, and click "Filter."

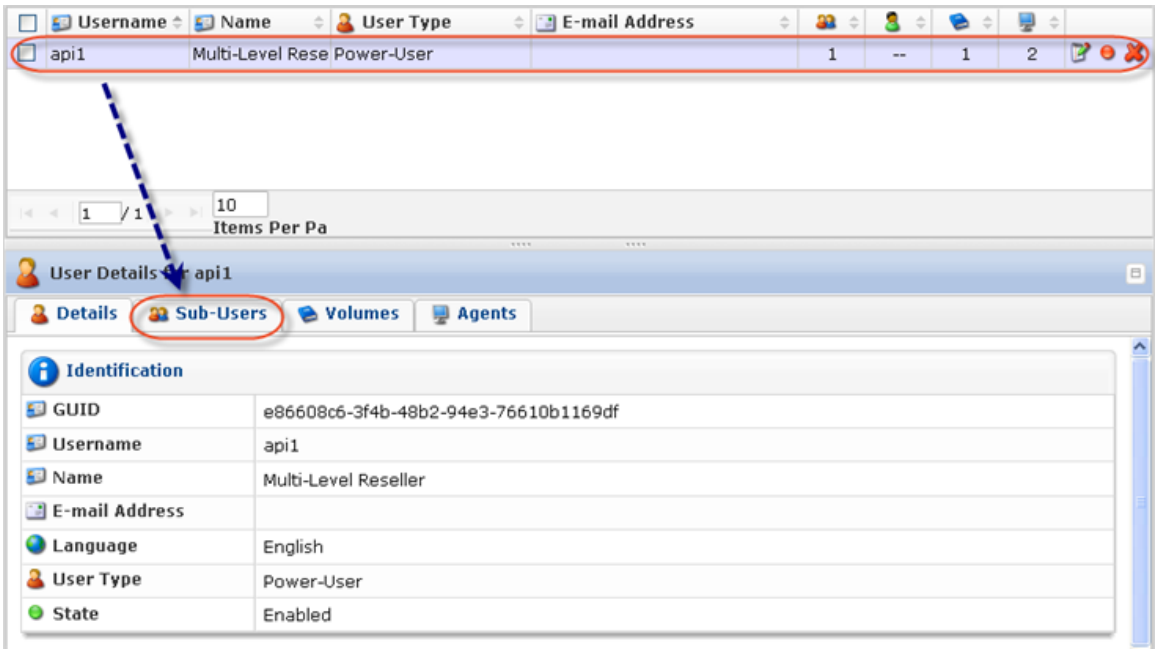
> **Note**
> While the script searches by User ID, we search by username in the CDP interface.



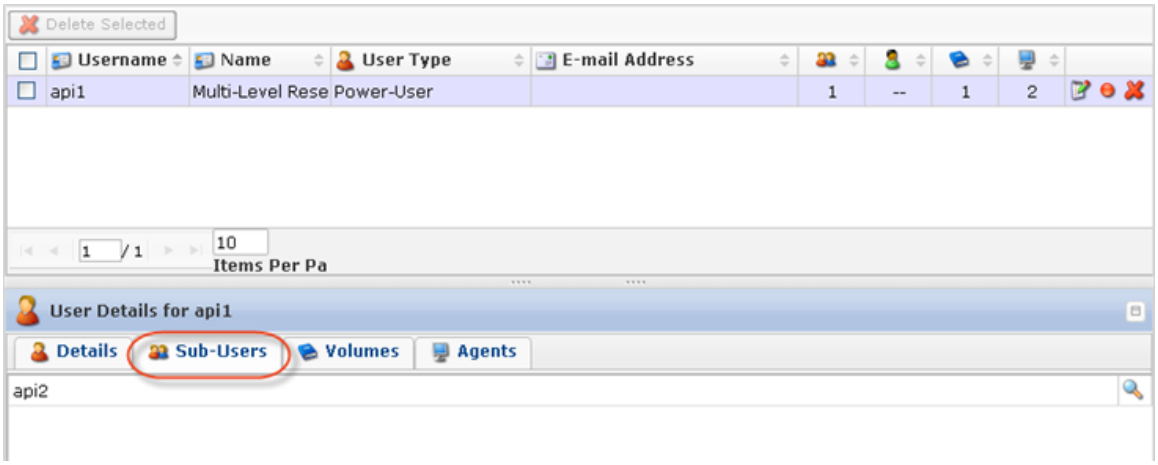
4. The found Users are displayed in the list.



5. Click on the User in the list, and then select the "Sub-Users" tab in the "Details" pane located in the bottom area of the interface.



Once the "Sub-Users" tab is selected, all Sub-Users will be listed there.



Retrieving Agents

```
########====Get Agent Start====########

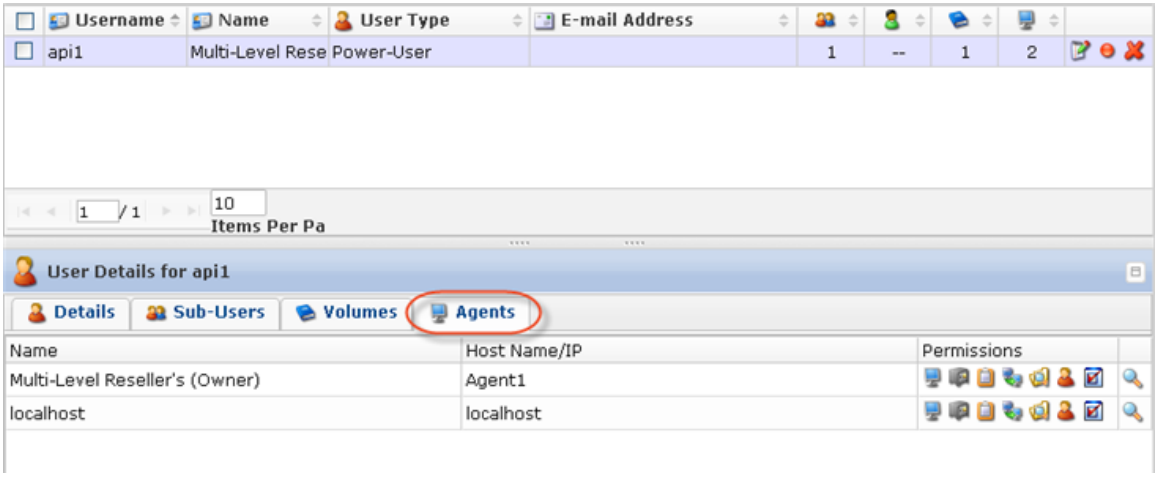
 try{
   $agentClient = new soapclient("https://$HOST:$PORT/Agent?wsdl",
       array('login'=>"$USER",
       'password'=>"$PASS",
       'trace'=>1,
    'cache_wsdl' => WSDL_CACHE_NONE,
    'features' => SOAP_SINGLE_ELEMENT_ARRAYS
        )
 );
 # get all the agents
 $allAgents=$agentClient->getAgents();
 echo "Successfully retrived all the agents \n";
 $selectedAgentIDs = array();
 foreach($allAgents->return as $tmp) {
  // check to see if the specified agent has the specified user as owner
  if (isset($tmp->ownerId) && $tmp->ownerId == $ID){
```

```
  // if it matches store the id
  array_push($selectedAgentIDs, $tmp->id);
 }
 }

 if (!isset($selectedAgentIDs) || count($selectedAgentIDs) == 0){
  echo " No Agents are owned by the specified user \n" ;
 }
}
catch (SoapFault $exception)
{
 echo "Failed to get the specified agent \n";
 echo $exception;
 exit(1);
}

########====Get Agent End====########
```

Select the "Agents" tab in the "Details" pane in the bottom area of the interface to list all associated Agents.



Retrieving Volumes

```
########====Get Volumes Start====########

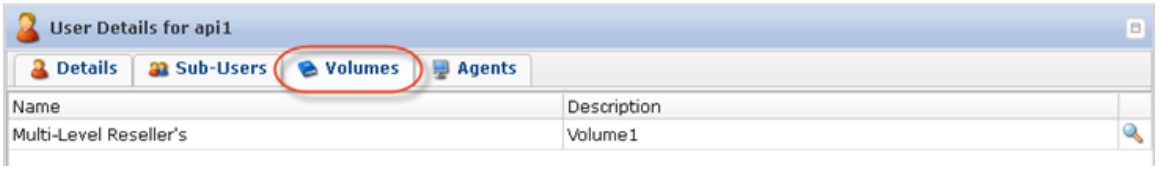
try {
 $volumeClient = new soapclient("https://$HOST:$PORT/Volume?wsdl",
  array('login'=>"$USER",
  'password'=>"$PASS",
  'cache_wsdl' => WSDL_CACHE_NONE,
  'features' => SOAP_SINGLE_ELEMENT_ARRAYS,
  'trace'=>1
   )
 );

 $allVolumes=$volumeClient->getVolumes();
 echo "Successfully retrived all the volumes \n";
 $selectedVolumeIDs = array();
 foreach($allVolumes->return as $tmp) {
  // check if the user is assigned the volume and the user is the only user in the assigned list
  if (isset($tmp->userIDs) && count($tmp->userIDs) == 1 && in_array($selectedUserID, $tmp->userIDs)) {
   // if the above condition is true put the volume in the volume id in the selectedVolume list
   array_push($selectedVolumeIDs, $tmp->id);
  }
 }
 if (!isset($selectedVolumeIDs) || count($selectedVolumeIDs) == 0){
```

```
  echo " No Volumes are exclusively assigned to the specified user \n" ;
 }
}
catch (SoapFault $exception)
{
 echo "Failed to get all volumes \n";
 echo $exception;
 exit(1);
}

########====Get Volumes Start====########
```

In the bottom "Details" pane, select the "Volumes" tab to list the Volumes assigned to the selected User.



Retrieving Disk Safes

```
########====Get DiskSafes Start====########

 try{

 $diskSafeClient = new soapclient("https://$HOST:$PORT/DiskSafe?wsdl",
  array('login'=>"$USER",
  'password'=>"$PASS",
  'trace'=>1,
  'cache_wsdl' => WSDL_CACHE_NONE,
  'features' => SOAP_SINGLE_ELEMENT_ARRAYS)
  );

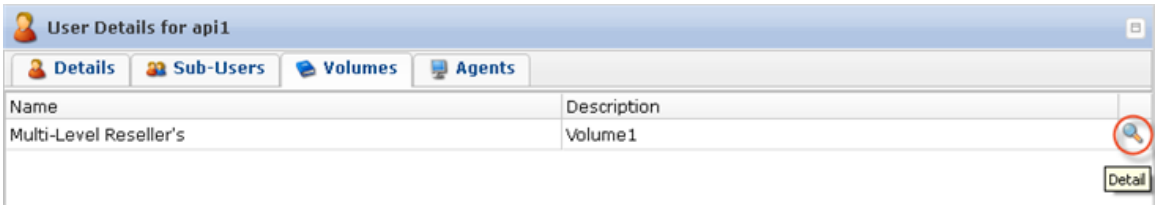


 // retrive all the ids
 $allDiskSafes = $diskSafeClient->getDiskSafes();
 echo "Successfully retrived all the diskSafes \n";
 $selectedDiskSafeIDs = array();
 foreach($allDiskSafes->return as $tmp) {
  // check if the disksafe has a valid agent set and it's id == the agent to be deleted
  if (isset($tmp->agentID) && in_array($tmp->agentID, $selectedAgentIDs)){
   // if the condition is true store then store the ids
   array_push($selectedDiskSafeIDs, $tmp->id);
  }
 }
 if (!isset($selectedDiskSafeIDs) || count($selectedDiskSafeIDs) == 0){
  echo " No Disksafes are associated with the agents owned by the user \n" ;
 }

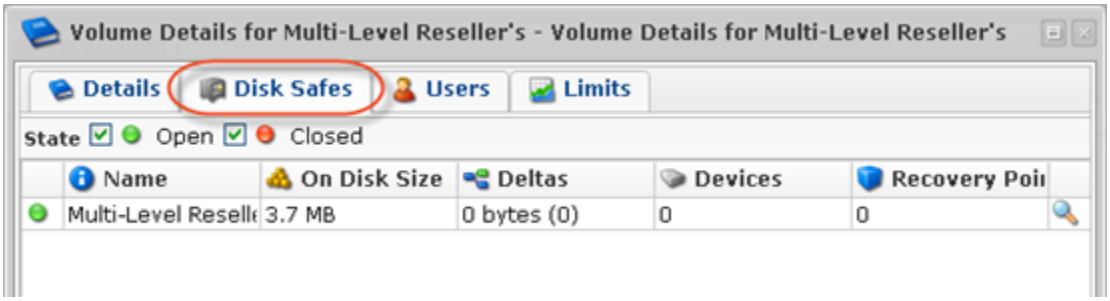
}
catch (SoapFault $exception)
{
 echo "Failed to get all diskSafes \n";
 echo $exception;
 exit(1);
}

########====Get DiskSafes End====########
```

Click on the "Detail" icon in front of the Volume to drill down to the Disk Safes.



In the displayed window, select the "Disk Safes" tab to list the Disk Safes assigned to the selected Volume.



Retrieving Policies

```
########====Get policies Start====########

 if (count($selectedDiskSafeIDs) > 0){
  try{
   $policyClient = new soapclient("https://$HOST:$PORT/Policy2?wsdl",
    array(
     'login'=>"$USER",
     'password'=>"$PASS",
     'cache_wsdl' => WSDL_CACHE_NONE,
     'features' => SOAP_SINGLE_ELEMENT_ARRAYS,
     'trace'=>1
    )
   );
   // get all the policies
   $allPolicies = $policyClient->getPolicies();

   echo "Successfully retrived all the policies \n";
   $selectedPolicyIDs = array();
   foreach($allPolicies->return as $tmp) {
    // check to see if disksafe id of the policy belongs in the list selectedDiskSafeIDs
    if (in_array($tmp->diskSafeID, $selectedDiskSafeIDs)){
     // if the condition is true save the policy id
     array_push($selectedPolicyIDs, $tmp->id);
    }
   }
   if (!isset($selectedPolicyIDs) || count($selectedPolicyIDs) == 0){
   echo " No Policies are associated with the disksafe assigned to the agents owned by the user \n" ;
  }

  }
  catch (SoapFault $exception)
  {
   echo "Failed to get all the policies \n";
   echo $exception;
   exit(1);
  }
 }
```

```
########====Get Policies End====########
```

Deleting the Policies, Disk Safes, and User

```
########====Delete Policies Start====########
// check to see if there are any policies to be deleted
 if (isset($selectedPolicyIDs) && count($selectedPolicyIDs) > 0){
  try {
   // iterate over the list of policy IDs and delete them
   foreach($selectedPolicyIDs as $tmp) {
    $policyClient->deletePolicyByID(array('policyID'=>$tmp));
   }
  }
  catch (SoapFault $exception)
  {
   echo "Failed to delete all policie(s) \n";
   echo $exception;
   exit(1);
  }
  echo "All policie(s) deleted successfully \n";
 }

########====Delete Policies End====########


########====Delete DiskSafes Start====########
// check to see if there are any disksafes to be deleted
 if (isset($selectedDiskSafeIDs) && count($selectedDiskSafeIDs) > 0){
  try {
   // iterate over the list of disksafe IDs and delete them
   foreach($selectedDiskSafeIDs as $tmp) {
    $diskSafeClient->deleteDiskSafeByID(array('diskSafeID'=>$tmp));
   }
  }
  catch (SoapFault $exception)
  {
   echo "Failed to delete all DiskSafe(s) \n";
   echo $exception;
   exit(1);
  }
  echo "All DiskSafe(s) deleted successfully \n";
 }

########====Delete DiskSafe End====########

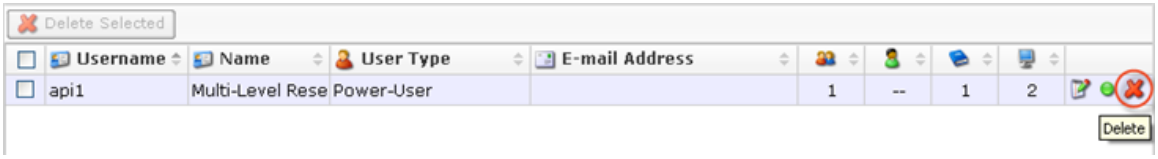
########====Delete User Start====########
 try {
  # finally delete the specified user
  $userClient->deleteUserByID(array('userID'=>$selectedUserID));
 }
 catch (SoapFault $exception)
 {
  echo "Failed to delete Power user\n";
  echo $exception;
  exit(1);
 }
 echo "Power user deleted Successfully \n";

 // delete all sub users if power users had any
 if (isset($selectedSubUserIDs) && count($selectedSubUserIDs) > 0){
  try {
   // iterate over the list of sub user IDs and delete them
   foreach($selectedSubUserIDs as $tmp) {
    $userClient->deleteUserByID(array('userID'=>$tmp));
   }
  }
  catch (SoapFault $exception)
  {
   echo "Failed to delete all SubUser(s) \n";
```

```
  echo $exception;
  exit(1);
 }
 echo "All SubUser(s) deleted successfully \n";
}

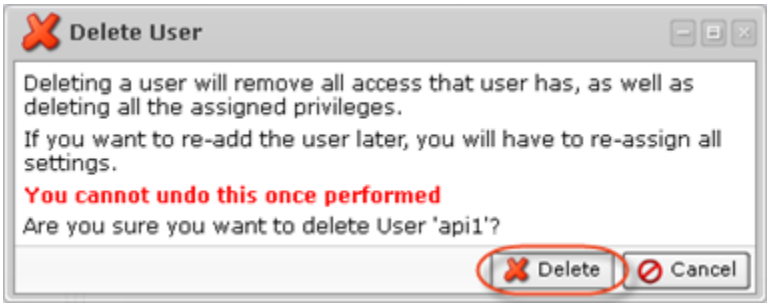
########====Delete User End====########
```

1. Click on the "Delete" (red X) icon under "Actions" for the corresponding User in the list.

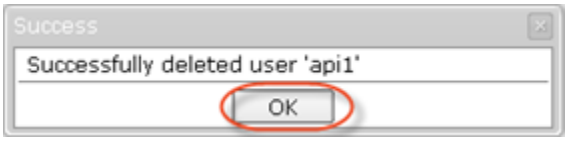


Alternatively, select the check-box(es) in front of the Users and click on the "Delete Selected" button.

2. Familiarize yourself with the information displayed on the pop-up. Click "Delete."



3. Click "OK" in the success pop-up.

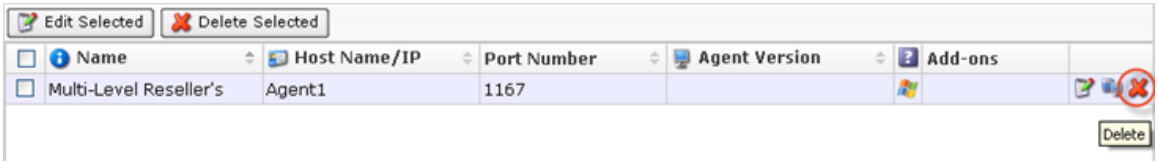


Deleting Agent

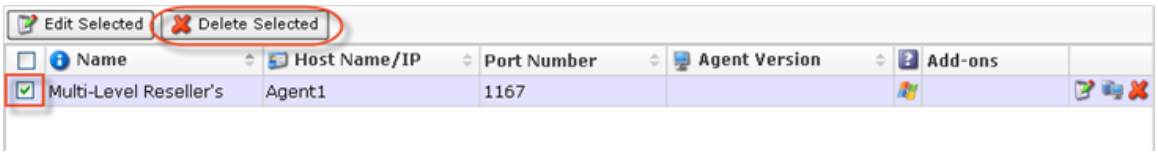
```
########====Delete Agent Start====########
 if (isset($selectedAgentIDs) && count($selectedAgentIDs) > 0){
  try {
   // iterate over the list of agent IDs and delete them
   foreach($selectedAgentIDs as $tmp) {
    $agentClient->deleteAgentByID(array('id'=>$tmp));
   }
  }
  catch (SoapFault $exception)
  {
   echo "Failed to delete all Agent(s) \n";
   echo $exception;
   exit(1);
  }
  echo "All Agent(s) deleted successfully \n";
```

```
}
########====Delete Agent End====########
```

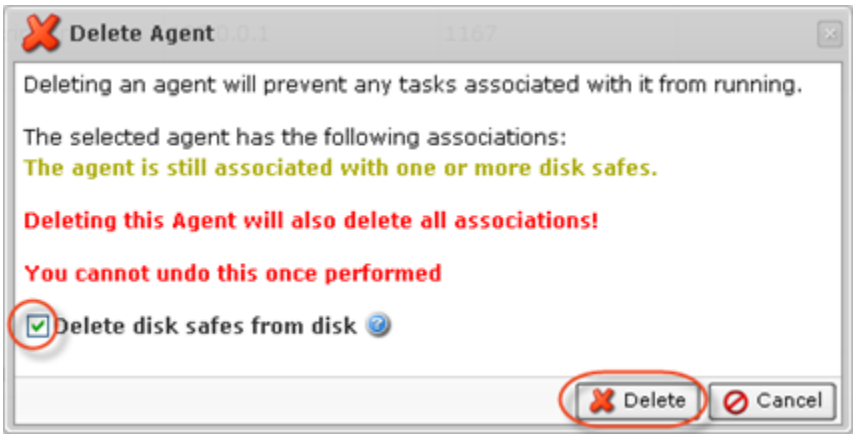
1. Click on the "Delete" (red X) icon under "Actions" for the corresponding Agent in the list.



Alternatively, select the check-box(es) in front of the Agents and click on the "Delete Selected" button.



2. Familiarize yourself with the information displayed on the pop-up. Check the "Delete disk safes from disk" option. Click "Delete."



3. The Agent will disappear from the system.

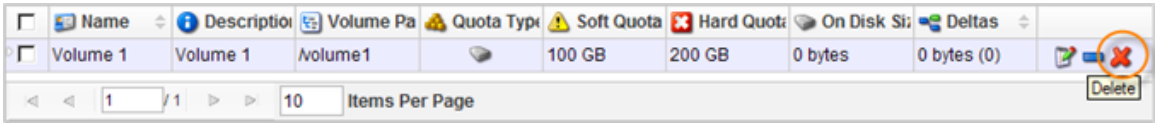
Deleting Volumes

```
########====Delete Volumes Start====########
 if (isset($selectedVolumeIDs) && count($selectedVolumeIDs) > 0){
  try {
   // iterate over the list of volume IDs and delete them
   foreach($selectedVolumeIDs as $tmp) {
    $volumeClient->deleteVolumeByID(array('id'=>$tmp));
   }
  }
  catch (SoapFault $exception)
  {
   echo "Failed to delete all Volume(s) \n";
   echo $exception;
   exit(1);
```

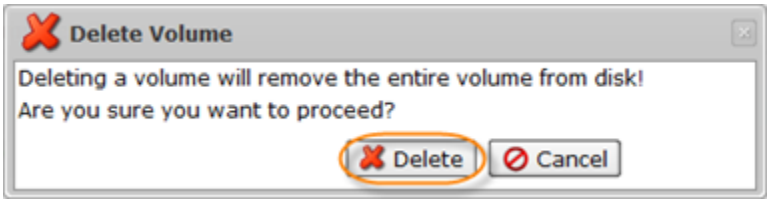
```
 }
 echo "All Volume(s) deleted successfully \n";
}

########====Delete Volumes End====#########
```

1. Click on the "Delete" (red X) icon under "Actions" for the corresponding Volume in the list.



2. Confirm your request to delete the Volume.



In a few moments, the volume record disappears from the "Volumes" list.