API - Creating an Agent, a Disk Safe, and a Policy

The following example demonstrates how to create an Agent, a Disk Safe, and a Policy.

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

Sequence of Automated Actions

The following steps can be accomplished by using this script:

1. Create an Agent. The Agent is the server or PC you want to back up. Read more in CDP Agents.
2. Create a Disk Safe. This is the storage location of the backed-up data. Read more in Disk Safes.
3. Create a Policy. This is the task which configures the backup. Read more in Policies.

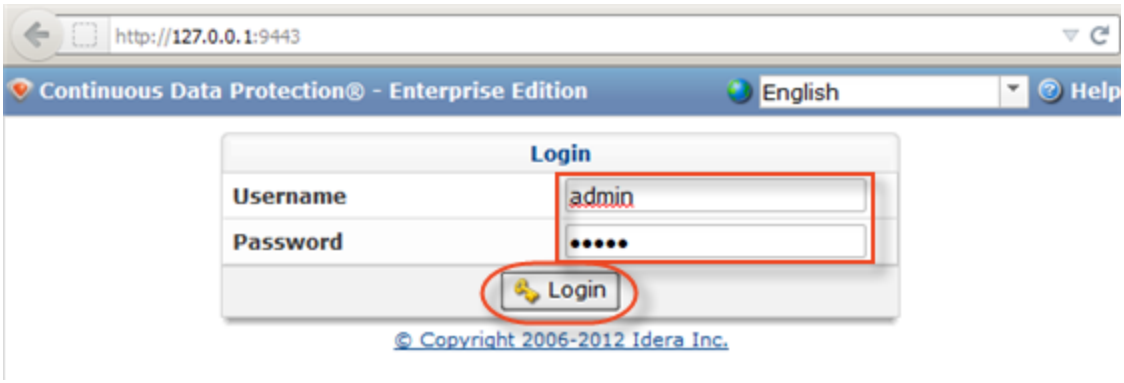
How to Fulfill Appropriate Actions in CDP User Interface

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

First Step | Creating an Agent | Creating a Disk Safe | Creating a Policy

First Step

```
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====########
```

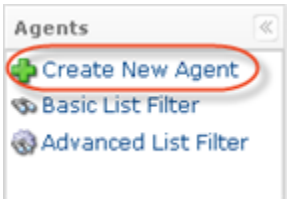


Creating an Agent

```
########====Create Agent Start====########

 ##Add all the basic properties to the agent object

 # Description of agent
 $agentObj->description = "agentDescription";
 # Host name of the agent ( It can be an IP or name )
 $agentObj->hostname = "127.0.0.1";
 # port number on whcih the agent will connect to the CDP server
 $agentObj->portNumber = 1167;
 # Set the agent OS type (ex: LINUX or WINDOWS) see Java Doc for more details
 $agentObj->osType = "WINDOWS";
 #set database addon to true if you have a database addon license and you want to backup a database
 $agentObj->databaseAddOnEnabled = true;

 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)
    );

 $createdAgent=$agentClient->createAgentWithObject(array('agent'=>$agentObj));
 echo "Successfully executed created Agent\n";

 }
 catch (SoapFault $exception)
 {
  echo "Failed to create agent \n";
  echo $exception;
  exit(1);
 }

########====Create Agent End====########
```

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

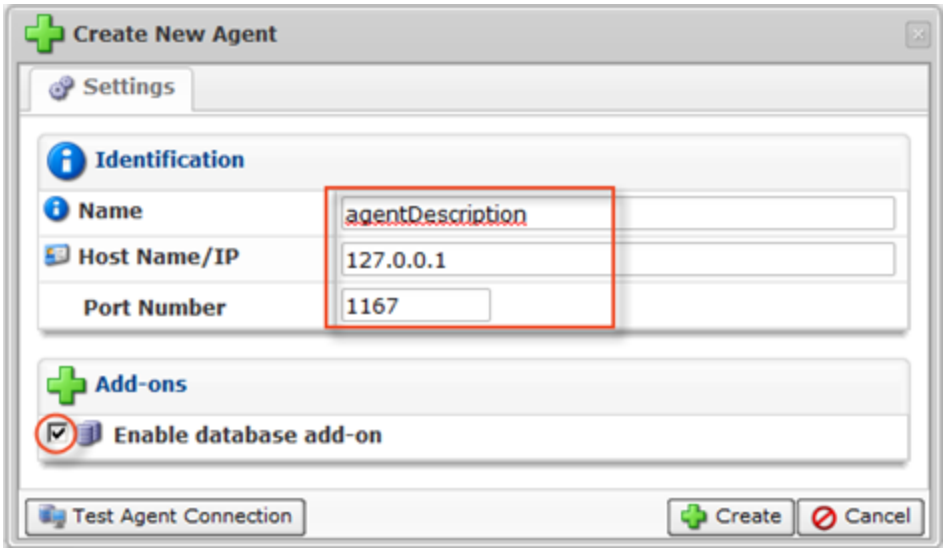
2. Then click "Create New Agent" in the "Agents" menu located in the top left area of the interface.



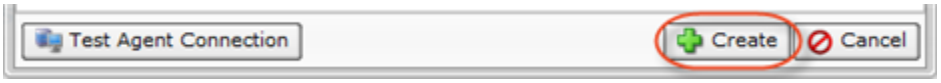
3. The "Create New Agent" window will appear. You will need to define the following options:

- Name - Enter a name for the Agent. It will be displayed in the "Agents" List. In this case, it is "agentDescription."
- Host Name - Enter the Host Name or IP address of the Agent. In this case, it is 127.0.0.1.
- Port Number - Define a Port to connect to the Agent, if different from the default value (1167).

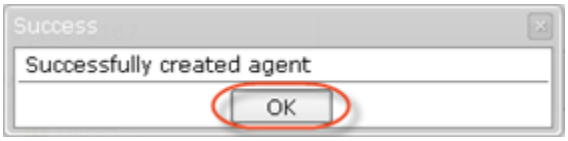
You should also enable a database add-on in the corresponding check-box.



4. Click "Create."



5. Click "OK" in the "Success" window.



See also Adding the Agent to the CDP Server.

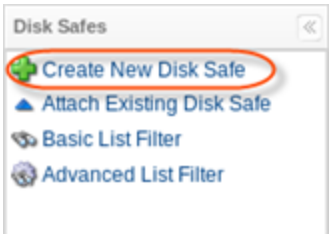
Creating a Disk Safe

```
########====Create DiskSafe Start====########

##Add all the basic properties to the diskSafe object

# description of the diskSafe
$disksafe->description ="description";
# use the agent we have just created,
$disksafe->agentID = $createdAgent->return->id;
# Path where the diskSafe will be created. This path has to be an absolute path
$disksafe->path = "C:\longAPIScript";
# type of compression required for the diskSafe. There are different types of compression levels. See "CompressionType" in Java Docs for more deta
$disksafe->compressionType = "QUICKLZ";
# This property defines how the devices will added to the CDP server, See "DeviceBackupType" in Java Doc for more details
$disksafe->deviceBackupType = "AUTO_ADD_DEVICES";
# set if partition tables should be replicated. This should be set to true if you plan to BMR
$disksafe->backupPartitionTable = true;
# set if unmounted devices should be replicated. This option is only recommend for windows with unmounted system partitions. This should be set to
$disksafe->backupUnmountedDevices = true;

# set diskSafe attributes. See DiskSafeAttributes in Java Doc for more info
$disksafe->diskSafeAttributeMap = array(
 array( "key" => "FILE_EXCLUDES_ENABLED", "value" => "true"),
 array( "key" => "ARCHIVING_ENABLED", "value" => "true"),
 array( "key" => "CONTROLPANELS_ENABLED", "value" => "true"),
 array( "key" => "REPLICATION_FREQUENCY_LIMIT", "value" => "NO_LIMIT"),
```

```
 # RECOVERY_POINT_LIMIT set the value to "-1" to disable recovery point limit
 array( "key" => "RECOVERY_POINT_LIMIT", "value" => "-1"),
 # ARCHIVE_POINT_LIMIT set the value to "-1" to disable archive point limit
 array( "key" => "ARCHIVE_POINT_LIMIT", "value" => "-1"),
 array( "key" => "QUOTA_TYPE", "value" => "NONE"),
 array( "key" => "SOFT_QUOTA_VALUE", "value" => "-1"),
 array( "key" => "HARD_QUOTA_VALUE", "value" => "-1")
);

#Sample encryption parameters if they are needed. This is not a required filed
/*
$disksafe->diskSafeEncryptionType = "XTS_AES_256";
$disksafe->diskSafeEncryptionPassphrase = "This is the disk safe passphrase";
*/

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)
 );
 $createdDiskSafe = $diskSafeClient->createDiskSafeWithObject(array('disksafe'=>$disksafe));
 echo "Successfully executed created DiskSafe\n";

}
catch (SoapFault $exception)
{
 echo "Failed to create diskSafe \n";
 echo $exception;
 exit(1);
}

########====Create DiskSafe End====########
```

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

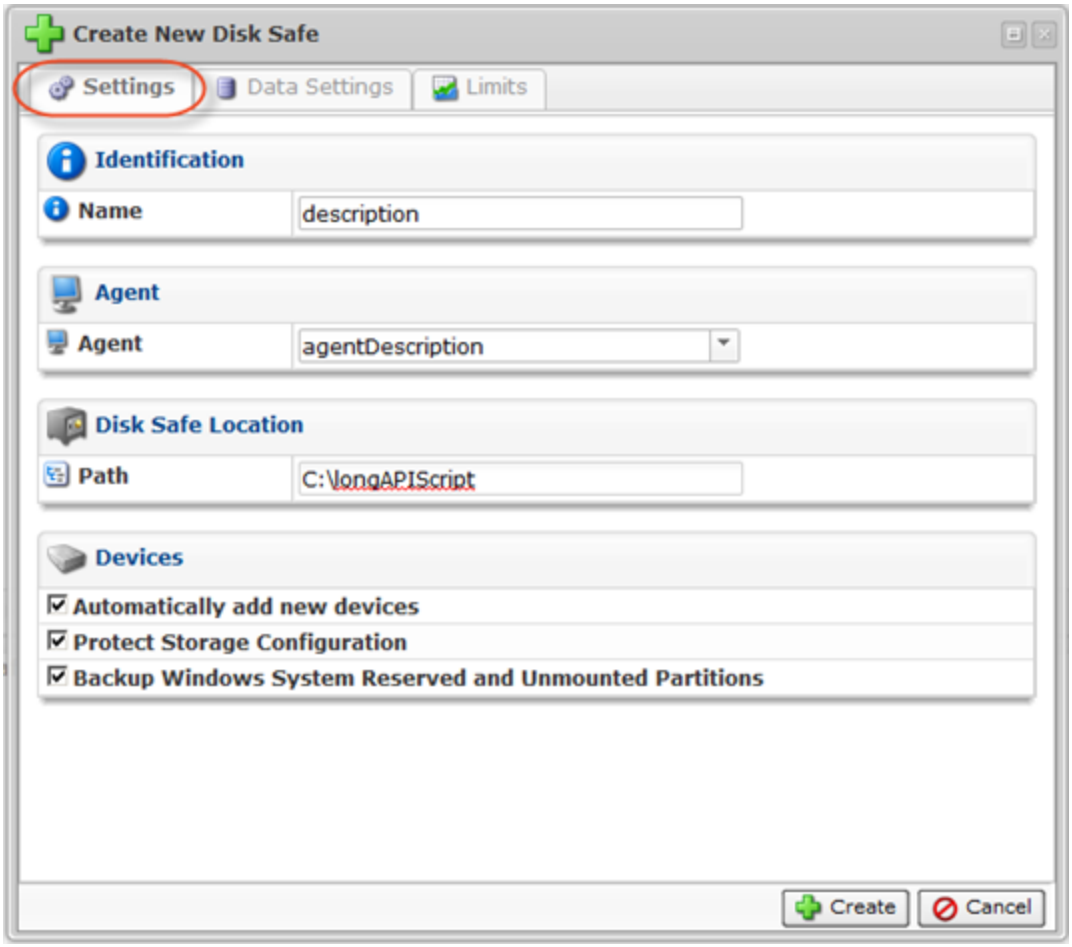
2. In the "Disk Safes" menu, click on "Create New Disk Safe."



3. The "Create New Disk Safe" window will open. You will need to define the following options:

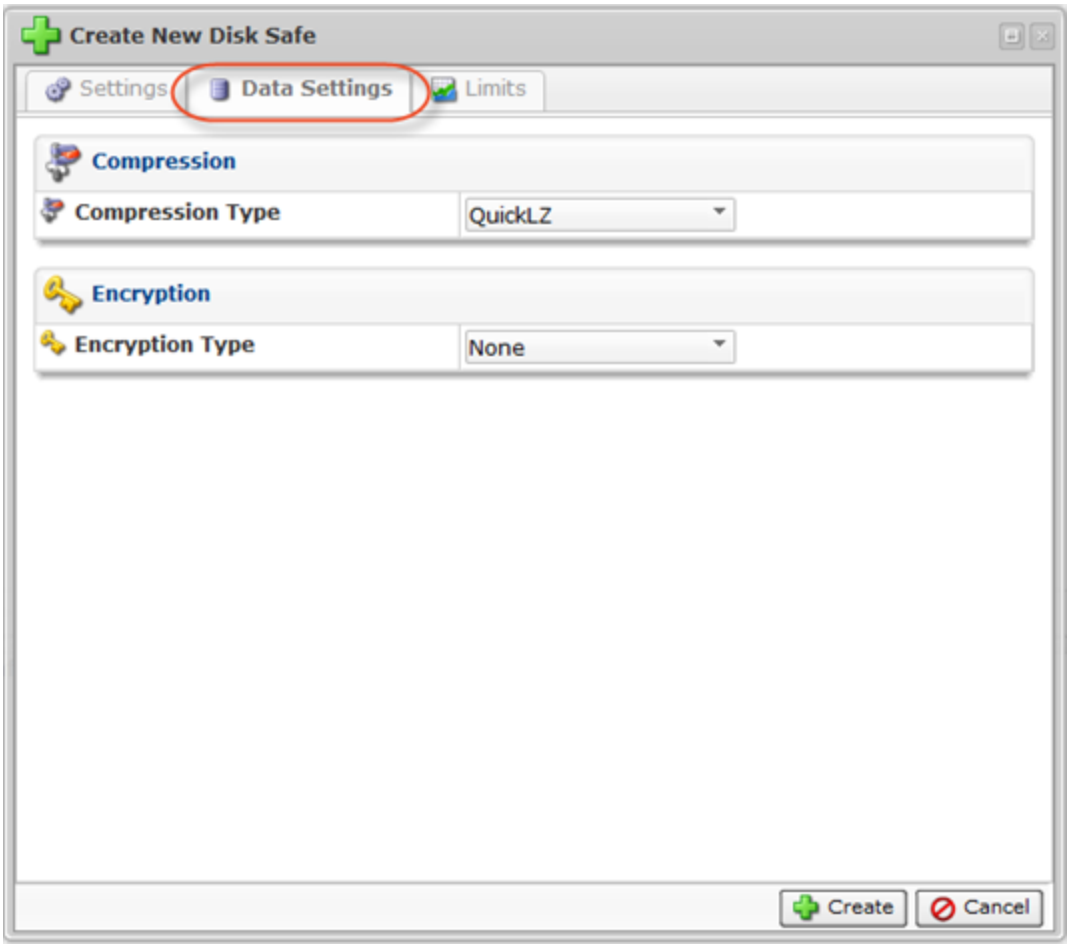
Settings Tab

- Name - Define the name for the Disk Safe. It will be used to identify the Disk Safe in the list. In this case, it is "description."
- Agent - Here, you should select an Agent from the drop-down menu. In this case, it is "agentDescription."
- Path - Path where the Disk Safe is physically located. In this case, it is C:\longAPIScript.
  You can also enable the following options in the corresponding check-boxes:
- Automatically add new devices - This option will automatically add all available Devices to the Disk Safe.
- Protect Storage Configuration (Enterprise Edition, Advanced Edition) - This option allows you to back up partition tables.
- Backup Windows System Reserved and Unmounted Partitions (Advanced Edition) - This option allows you to back up a hidden Windows system partition.

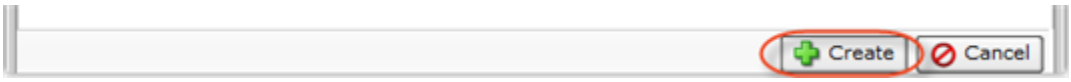


Data Settings Tab

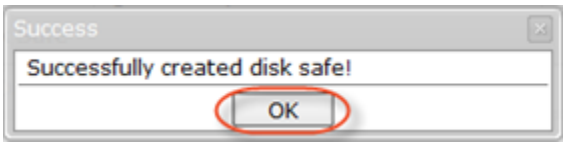
- Compression Type - The compression level set for the Disk Safe. In this case, it is "QuickLZ."



4. Click "Create."



5. Click "OK" in the "Success" window.



See Creating Disk Safes.

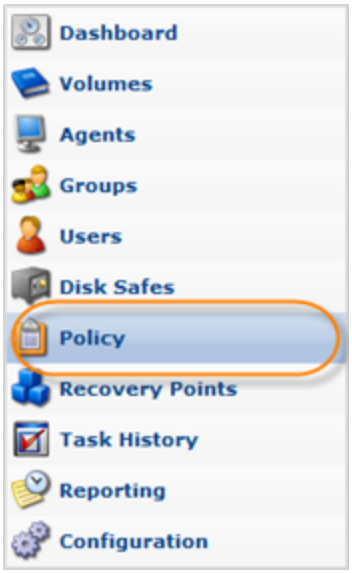
Creating a Policy

```
########====Create Policy Start====########

##Add all the basic properties to the policy object


$policyObj->enabled = true;
$policyObj->name = "nameOfPolicy";
$policyObj->description = "descriptionOfPolicy";
```

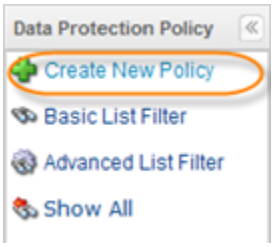
```
$policyObj->diskSafeID = $createdDiskSafe->return->id;

$policyObj->replicationScheduleFrequencyType = "HOURLY";
$replScheFreqVals->startingMinute= 5;
$policyObj->replicationScheduleFrequencyValues = $replScheFreqVals;

$policyObj->mergeScheduleFrequencyType = "DAILY";
$mergScheFreqVals->startingMinute=1;
$mergScheFreqVals->hoursOfDay=array("1");
$policyObj->mergeScheduleFrequencyValues = $mergScheFreqVals;
$policyObj->recoveryPointLimit = 10;
$policyObj->forceFullBlockScan = false;
$policyObj->multiVolumeSnapshot = false;

#sample databaseInstance

$DatabaseInstance->enabled=true;
$DatabaseInstance->dataBaseType="MYSQL";
$DatabaseInstance->name="Apitest";
$DatabaseInstance->description="newApiTest";
$DatabaseInstance->useAlternateHostname=false;
$DatabaseInstance->username="root";
$DatabaseInstance->password="password";
$DatabaseInstance->useAlternateDataDirectory=false;
$DatabaseInstance->useAlternateInstallDirectory=false;
$DatabaseInstance->backupAllDatabases=true;
$DatabaseInstance->advancedOptions=array(array( key => "DO_TABLE_STATS_OPTION", value => false));
$DatabaseInstance->hostName="localhost";
$DatabaseInstance->portNumber=3306;
$DatabaseInstance->virtuozzoContainerID=1254;

$DatabaseInstance1->enabled=true;
$DatabaseInstance1->dataBaseType="MYSQL";
$DatabaseInstance1->name="Apitest1";
$DatabaseInstance1->description="newApiTest1";
$DatabaseInstance1->useAlternateHostname=false;
$DatabaseInstance1->username="root";
$DatabaseInstance1->password="password";
$DatabaseInstance1->useAlternateDataDirectory=false;
$DatabaseInstance1->useAlternateInstallDirectory=false;
$DatabaseInstance1->backupAllDatabases=true;
$DatabaseInstance1->advancedOptions=array(array( key => "DO_TABLE_STATS_OPTION", value => false));
$DatabaseInstance1->hostName="localhost";
$DatabaseInstance1->portNumber=3306;
$DatabaseInstance1->virtuozzoContainerID=1254;


$policyObj->databaseInstanceList=array($DatabaseInstance, $DatabaseInstance1);


#sample controlPanelInstance
$ControlPanelInstance->enabled=true;
$ControlPanelInstance->controlPanelType="CPANEL";
$ControlPanelInstance->name="Apitest";
$ControlPanelInstance->description="newApiTest";
$ControlPanelInstance->virtuozzoContainerID=1254;
$ControlPanelInstance->advancedOptions=array();

$ControlPanelInstance1->enabled=true;
$ControlPanelInstance1->controlPanelType="PLESK";
$ControlPanelInstance1->name="Apitest1";
$ControlPanelInstance1->description="newApiTest1";
$ControlPanelInstance1->virtuozzoContainerID=1255;
$ControlPanelInstance1->advancedOptions=array();

$ControlPanelInstance2->enabled=true;
$ControlPanelInstance2->controlPanelType="VIRTUOZZO";
$ControlPanelInstance2->name="Apitest2";
$ControlPanelInstance2->description="newApiTest2";
$ControlPanelInstance2->advancedOptions=array();


$policyObj->controlPanelInstanceList=array($ControlPanelInstance, $ControlPanelInstance1, $ControlPanelInstance2);
```

```
#Sample ArchiveSchedule

$ArchiveScheduleInstance1->archiveScheduleType = "DAILY";
$ArchiveScheduleInstance1->retentionCount = 50;
$DailyFrequencyValues->startingMinute=1;
$DailyFrequencyValues->hoursOfDay=array("5");
$ArchiveScheduleInstance1->archiveScheduleFrequencyValues = $DailyFrequencyValues;

$ArchiveScheduleInstance2->archiveScheduleType = "WEEKLY";
$ArchiveScheduleInstance2->retentionCount = 50;
$WeeklyFrequencyValues->daysOfWeek=TUESDAY;
$WeeklyFrequencyValues->startingHour=array(1, 5);
$WeeklyFrequencyValues->startingMinute = 10;
$ArchiveScheduleInstance2->archiveScheduleFrequencyValues = $WeeklyFrequencyValues;

$ArchiveScheduleInstance3->archiveScheduleType = "MONTHLY";
$ArchiveScheduleInstance3->retentionCount = 50;
$MonthlyFrequencyValues->daysOfMonth = 3;
$MonthlyFrequencyValues->startingHour = 3;
$MonthlyFrequencyValues->startingMinute = 20;
$ArchiveScheduleInstance3->archiveScheduleFrequencyValues = $MonthlyFrequencyValues;

$policyObj->archiveScheduleInstanceList=array($ArchiveScheduleInstance1, $ArchiveScheduleInstance2, $ArchiveScheduleInstance3);

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
  )
 );

 $createdPolicy = $policyClient->createPolicy(array('policy'=>$policyObj));
 echo "Successfully executed created Policy\n";

}
catch (SoapFault $exception)
{
 echo "Failed to create policy \n";
 echo $exception;
 exit(1);
}
########====Create Policy End====########
```

1. Click on "Policy" in the Main Menu to open the "Policies" screen.

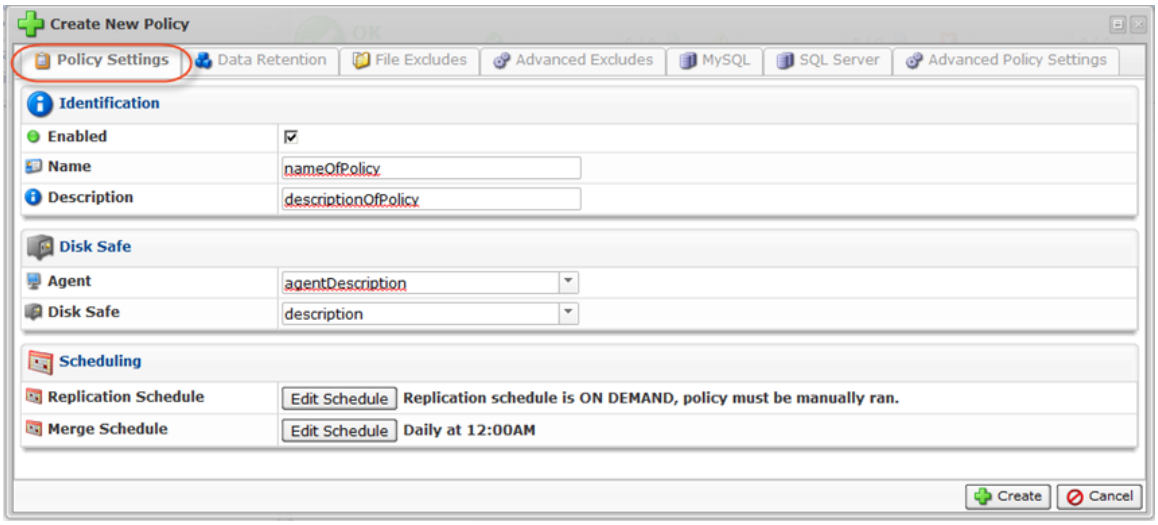


2. In the Policy menu, click on "Create New Policy."

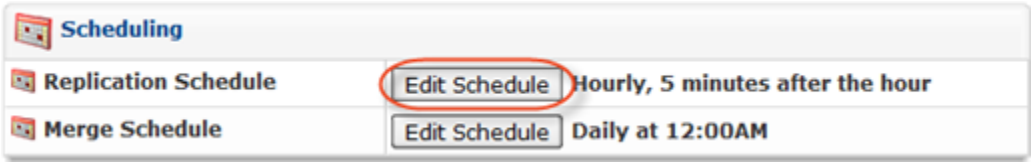


3. The "Create New Policy" window will open. Define the following options in the "Policy Settings" tab:

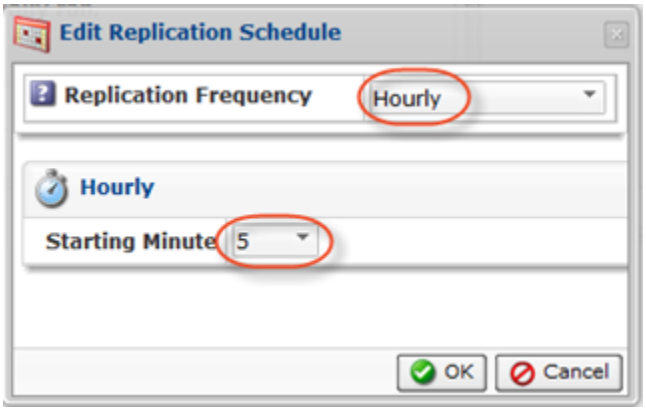
- Enabled - Select this check-box to enable the Policy. The Enabled Policy will run according to the schedule.
- Name - Enter a name you can use to identify this Policy among others in the Policies list. In this case, it is "nameOfPolicy."
- Description - Describe the details of your Policy. In this case, it is "descriptionOfPolicy."
- Disk Safe - From the drop-down menu, select a Disk Safe in which to save the replicated data. In this case, it is "description."



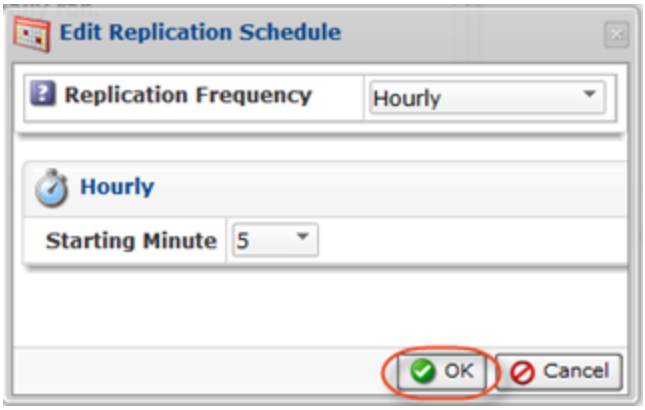
4. Click "Edit Schedule" to edit the replication schedule.



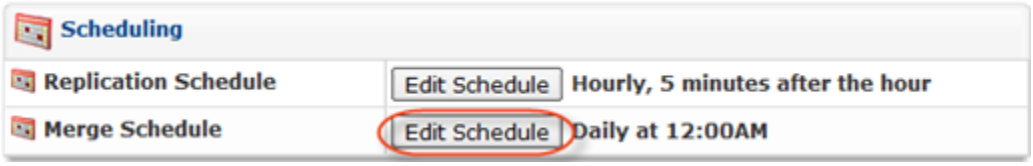
Choose "Hourly" as the Replication Frequency and "5" as the Starting Minute.



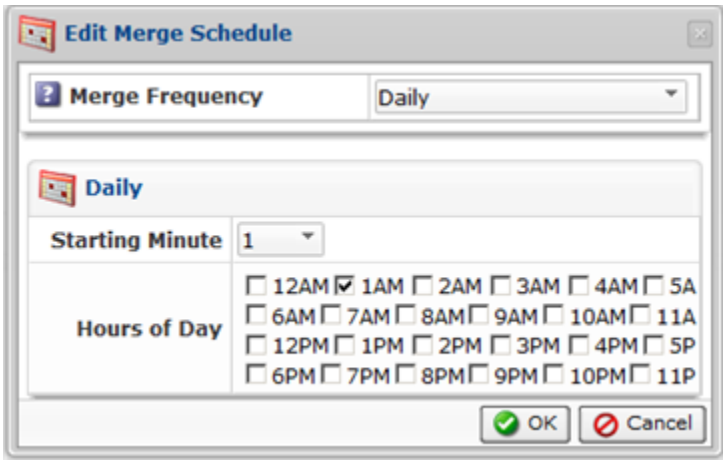
Click "OK."



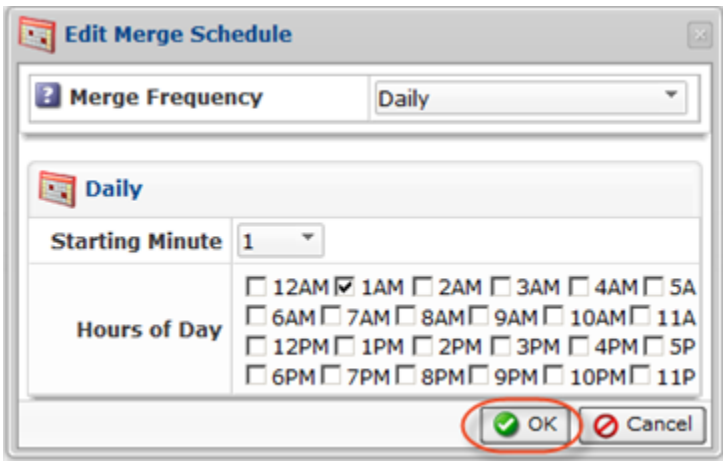
5. Click "Edit Schedule" to edit the merge schedule.



Define "Daily" as the Merge Frequency and "1" as the Starting Minute.



Click "OK."



6. Click "Create."

7. Click "OK" in the "Success" window.

See Creating Policies.