On Demand Backup

1. Versions

Version	Date	User	Comment

Current version:0.0.0

2. Inputs

Name	Type	Description
wfRequestorName	string	
wfTransactionID	string	
wfRestHost	string	
wfBackupType	string	
wfClientName	string	
wfApplicationName	string	
wfInstance	string	
wfBackupSet	string	
wfSubClientName	string	
wfBackupLevel	string	
wfRunIncrementalBackup	string	
wfIncrementalLevel	string	
wfBackupFailedVmsOnly	string	

3. Attributes

Name	Type	Description
attBackupLevel	string	
attToken	SecureString	
attClientId	number	

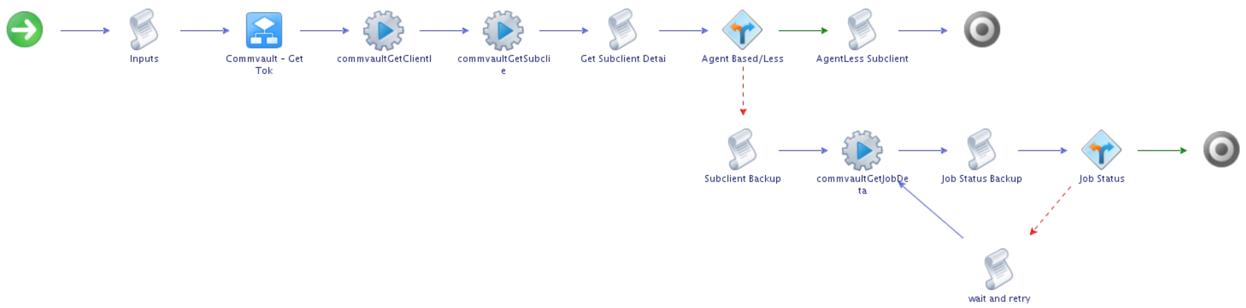
Name	Type	Description
attSubclientId	number	
attSubclientProp	string	
attBackupSetId	number	
attInstanceId	number	
attApplicationId	string	
attIncrementalLevel	string	
attJobId	number	
attJobDetails	string	
attCurrentStatus	string	
attReason	string	
attCounter	number	
attApplicationName	string	

4. Parameter presentation

Parameter	Description	Parameter Info
wfRequestorName	wfRequestorName	
wfTransactionID	wfTransactionID	
wfRestHost	wfRestHost	
wfBackupType	wfBackupType	
wfClientName	wfClientName	
wfApplicationName	wfApplicationName	
wfInstance	wfInstance	
wfBackupSet	wfBackupSet	
wfSubClientName	wfSubClientName	
wfBackupLevel	wfBackupLevel	

Parameter	Description	Parameter Info
wfRunIncrementalBackup	wfRunIncrementalBackup	
wfIncrementalLevel	wfIncrementalLevel	
wfBackupFailedVmsOnly	wfBackupFailedVmsOnly	

5. Workflow schema



6. Workflow items

Name	Type	Next item	Description
Inputs	WorkflowItemTask	Commvault - Get Token	
commvaultGetClientIdFromName	WorkflowItemTask	commvaultGetSubclientDetails	Query Rubrik for all current SLA Domains and try to match one against the name provided. If a match is found return the ID of the matching SLA Domain
Commvault - Get Token	WorkflowItemLink	commvaultGetClientIdFromName	
	WorkflowItemEnd		
commvaultGetSubclientDetails	WorkflowItemTask	Get Subclient Details	Query Rubrik for all current SLA Domains and try to match one against the name provided. If a match is found return the ID of the matching SLA Domain

Name	Type	Next item	Description
Get Subclient Details	WorkflowItemTask	Agent Based/Less	
Agent Based/Less	WorkflowItemCustomCondition	AgentLess Subclient Backup	
Subclient Backup	WorkflowItemTask	commvaultGetJobDetailsFromJobId	
commvaultGetJobDetailsFromJobId	WorkflowItemTask	Job Status Backup	Query Rubrik for all current SLA Domains and try to match one against the name provided. If a match is found return the ID of the matching SLA Domain
	WorkflowItemEnd		
Job Status Backup	WorkflowItemTask	Job Status	
Job Status	WorkflowItemCustomCondition		
wait and retry	WorkflowItemTask	commvaultGetJobDetailsFromJobId	
AgentLess Subclient Backup	WorkflowItemTask		

7. Source code for the used actions

7.1. Inputs

```
System.log("******************************");
System.log("Order ID : "+wfTransactionID);
System.log("Requestor Name : "+wfRequestorName);
System.log("******************************");

//Check the Backup Type
```

```
if(!wfBackupType.trim()){throw "Backup Level cannot be empty."}

//Check the client
if(!wfClientName.trim()){throw "Client Name cannot be empty."}

//Check the Application Name
if(!wfApplicationName.trim()){throw "Application Name cannot be empty."}
if(wfApplicationName.trim().toLowerCase().indexOf("file system")!=-1){
  attApplicationName = "File System"
}
else{
  attApplicationName = wfApplicationName
}

//Check the subclient
if(!wfSubClientName.trim()){throw "SubClient Name cannot be empty."}

//Create the actual Backup Level Name
if(wfBackupLevel)
{
  if(wfBackupLevel=="SYNTHETIC FULL"){attBackupLevel="SYNTHETIC_FULL"}
  else{attBackupLevel=wfBackupLevel}
}
else
{
  throw "Backup Level cannot be empty."
}

//Check the incremental Level
if(wfIncrementalLevel.toLowerCase()=="before sync"){
  attIncrementalLevel = "BEFORE_SYNC"
}
else if(wfIncrementalLevel.toLowerCase()=="after sync"){
  attIncrementalLevel = "AFTER_SYNC"
}
else{
  attIncrementalLevel = null;
}
```

7.2. commvaultGetClientIdFromName

```
//Auto generated script, cannot be modified !
actionResult =
System.getModule("com.commvault.library.rest").commvaultGetClientIdFromName(commvaultUrl,c
ommvaultToken,clientName) ;
```

7.3. commvaultGetSubclientDetails

```
//Auto generated script, cannot be modified !
actionResult =
System.getModule("com.commvault.library.rest").commvaultGetSubclientDetails(commvaultUrl,c
ommvaultToken,clientName,applicationName,backupSet,instanceName,subclientName) ;
```

7.4. Get Subclient Details

```
System.log("******** Get Subclient Details Inputs ********");
System.log("Subclient Properties : "+attSubclientProp);
System.log("");

try
{
  var subclientPropJson = JSON.parse(attSubclientProp);
  attSubclientId = subclientPropJson.subclientId;
  attApplicationId = subclientPropJson.applicationId;
  attBackupSetId = subclientPropJson.backupsetId;
  attInstanceId = subclientPropJson.instanceId;

  System.log("Client Name : "+subclientPropJson.clientName);
  System.log("Client ID : "+subclientPropJson.clientId);
  System.log("Subclient Name : "+subclientPropJson.subclientName);
  System.log("Subclient ID : "+attSubclientId);
  System.log("Application Name : "+subclientPropJson.appName);
  System.log("Application ID : "+attApplicationId);
  System.log("Backupset Name : "+subclientPropJson.backupsetName);
  System.log("Backupset ID : "+attBackupSetId);
  System.log("Instance Name : "+subclientPropJson.instanceName);
  System.log("Instance ID : "+attInstanceId);

}
catch(e)
{
  var error = "Error occured in 'Get Subclient Details' Task. Error Details : "+e;
  throw error;
}
```

7.5. Agent Based/Less

```
if(wfBackupType.toLowerCase()=="agentless")
{
  System.log("Proceeding with taking backup on subclient from vCenter : "+wfBackupType);
  return true;
}
else if(wfBackupType.toLowerCase()=="agentbased")
{
  System.log("Proceeding with taking backup on agent based subclient : "+wfBackupType);
  return false;
}
else
{
  throw "The type should be eithe Agentbased or Agentless";
}
```

7.6. Subclient Backup

```
throw "error"
System.log("******** Subclient Backup Inputs ********");
System.log("Rest Host : "+wfRestHost);
System.log("Client Name : "+wfClientName);
System.log("Application Name : "+attApplicationName);
System.log("Subclient Name : "+wfSubClientName);
System.log("Subclient ID : "+attSubclientId);
System.log("Backup Level : "+attBackupLevel);
System.log("Run Incremental Backup : "+wfRunIncrementalBackup);
System.log("Incremental Level : "+attIncrementalLevel);
System.log("");

try
{
  var url = null;
  if(attApplicationName.toLowerCase()=="file system"){
     if(attBackupLevel.toLowerCase=="synthetic_full"){
        if(wfRunIncrementalBackup.toLowerCase()=="yes"){
           url =
"Subclient/"+attSubclientId+"/action/backup?backupLevel="+attBackupLevel+"&runIncrementalB
ackup=TRUE&incrementalLevel="+attIncrementalLevel;
        }
        else{
           url =
"Subclient/"+attSubclientId+"/action/backup?backupLevel="+attBackupLevel+"&runIncrementalB
ackup=FALSE";
        }
     }
     else{
        url = "Subclient/"+attSubclientId+"/action/backup?backupLevel="+attBackupLevel;
     }
  }
  else if(attApplicationName.toLowerCase()=="sql server"){
     url = "Subclient/"+attSubclientId+"/action/backup?backupLevel="+attBackupLevel;
  }
  else if(attApplicationName.toLowerCase()=="virtual server"){
     url = "Subclient/"+attSubclientId+"/action/backup?backupLevel="+attBackupLevel;
  }
  else{
     throw "Application should be either File System, SQL Server or Virtual Server."
  }

  //Trigger the Subclient Backup
  if(url){
     var method = "POST";
     var content = undefined;
     //var response = 
System.getModule("com.commvault.library.rest").commvaultGetFromAPI(wfRestHost,attToken,url
,method,content) ;
     var json = JSON.parse(response.contentAsString);
     if(json.taskId)
     {
        System.log("Job ID : "+json.jobIds);
        attJobId = json.jobIds[0];
     }
     else
     {
        throw response.contentAsString;
     }
  }
  else{
     throw "URL is empty."
  }
}
catch(e)
{
  var error = "Error occured in 'Subclient Backup' Task. Error Details : "+e;
  throw error;
}
```

7.7. commvaultGetJobDetailsFromJobId

```
//Auto generated script, cannot be modified !
actionResult =
System.getModule("com.commvault.library.rest").commvaultGetJobDetailsFromJobId(commvaultUr
l,commvaultToken,jobId) ;
```

7.8. Job Status Backup

```
System.log("******** Job Status Backup Inputs ********");
System.log("Job Details : "+attJobDetails);
System.log("");

try
{
  var jobProp = JSON.parse(attJobDetails);
  System.log("Backup Job Status Details: ")
  var attCurrentStatus = jobProp.jobs[0].jobSummary.status;
  var progress = jobProp.jobs[0].jobSummary.percentComplete;
  var attReason = jobProp.jobs[0].jobSummary.pendingReason;

  System.log("****************************************");
  System.log("Job status : "+attCurrentStatus);
  System.log("Progress of job : "+progress+"%");
  System.log("****************************************");
}
catch(e)
{
  var error = "Error occured in 'Job Status Backup' Task. Error Details : "+e;
  throw error;
}
```

7.9. Job Status

```
if(attCurrentStatus == "Completed" || attCounter == 20)
{
  System.log("Job is either completed or the time exceeded 10 minutes..");
  return true;
}
else if(attCurrentStatus == "Failed")
{
  var errReason =  "Job has been failed with error : "+attReason;
  throw errReason;
}
else if(attCurrentStatus == "Pending")
{
  var errReason =  "Job is Pending with reason : "+attReason+". Kindly check with 
respective commvaut admin for the Job ID : "+attJobId;
  throw errReason;
}
else
{
   return false;
}
```

7.10. wait and retry

```
var delay = 30;
attCounter = attCounter+1
System.log("counter : "+attCounter)
System.log("Retrying after : " + delay + " seconds.");
System.sleep(delay*1000);
```

7.11. AgentLess Subclient Backup

```
System.log("******** AgentLess Subclient Backup Inputs ********");
System.log("Rest Host : "+wfRestHost);
System.log("Client Name : "+wfClientName);
System.log("Client ID : "+attClientId);
System.log("Application Name : "+attApplicationName);
System.log("Application ID : "+attApplicationId);
System.log("Instance Name : "+wfInstance);
System.log("Instance ID : "+attInstanceId);
System.log("BackupSet Name : "+wfBackupSet);
System.log("BackupSet ID : "+attBackupSetId);
System.log("Subclient Name : "+wfSubClientName);
System.log("Subclient ID : "+attSubclientId);
System.log("Backup Level : "+attBackupLevel);
System.log("Backup Failed VMs Only : "+wfBackupFailedVmsOnly);
System.log("");

//try
//{
  var content = null;
  var content = JSON.stringify(
{
  "taskInfo": {
     "associations": {
        "_type_": "SUBCLIENT_ENTITY",
        "appName": attApplicationName,
        "applicationId": attApplicationId,
        "backupsetName": wfBackupSet,
        "backupsetId": attBackupSetId,
        "clientName": wfClientName,
        "clientId": attClientId,
        "instanceName": wfInstance,
        "instanceId": attInstanceId,
        "subclientName": wfSubClientName,
        "subclientId": attSubclientId
     },
     "subTasks": {
        "options": {
           "backupOpts": {
              "mediaOpt":{
              "allowOtherSchedulesToUseMediaSet":true,
              "markMediaFullOnSuccess":false,
              "reserveResourcesBeforeScan":false,
              "retentionJobType":"STORAGE_POLICY_DEFAULT",
              "startNewMedia":true
              },
              "vsaBackupOptions": {
                 "backupFailedVMsOnly": true
              },
              "backupLevel": attBackupLevel,
              "doNotTruncateLog": false,
              "isSpHasInLineCopy": false,
              "runIncrementalBackup": false,
              "runSILOBackup": false,
              "sybaseSkipFullafterLogBkp": false,
              "truncateLogsOnSource": false
           },
           "commonOpts": {
              "jobRetryOpts": {
                 "runningTime": {
                    "enableTotalRunningTime": false,
                    "totalRunningTime": 3600
                 },
                 "enableNumberOfRetries": false,
                 "killRunningJobWhenTotalRunningTimeExpires": false
              },
              "startUpOpts": {
                 "priority": 100,
                 "startInSuspendedState": false,
                 "useDefaultPriority": true
              },
              "jobDescription": wfTransactionID
           }
        },
        "subTask": {
           "operationType": "BACKUP",
           "subTaskType": "BACKUP"
        }
     },
     "task": {
        "taskFlags": {
           "disabled": false
        },
        "initiatedFrom": "COMMANDLINE",
        "policyType": "DATA_PROTECTION",
        "sequenceNumber": 10,
        "taskId": 0,
        "taskType": "IMMEDIATE"
     }
  }
}
);
  System.log("JSON Content : "+content);
  throw "Error";
  //Trigger the Subclient Backup
  if(content){
     var url = "CreateTask";
     var method = "POST";
     //var response =
System.getModule("com.commvault.library.rest").commvaultGetFromAPI(wfRestHost,attToken,url
,method,content);
     var json = JSON.parse(response.contentAsString);
     if(json.taskId)
     {
        System.log("Job ID : "+json.jobIds);
        //attJobId = json.jobIds[0];
     }
     else
     {
        throw response.contentAsString;
     }
  }
  else{
     throw "URL is empty."
  }
/*}
catch(e)
{
  var error = "Error occured in 'Subclient Backup' Task. Error Details : "+e;
  throw error;
}*/
```