

URL <small>HostName for connection</small>	<input type="text" value="ldaps://192.168.55.55:636"/>
USERNAME <small>LDAP user dn (e.g. CN=Savlynt,CN=Users,DC=ad,DC=savlynt,DC=com)</small>	<input type="text" value="cn=DavidK,cn=Users,dc=talanos,dc=local"/>
PASSWORD * <small>Password for connection</small>	<input type="password" value="*****"/>
LDAP_OR_AD <small>Type of Endpoint - LDAP or AD, default is AD</small>	<input type="text" value="AD"/>
SEARCHFILTER <small>Search Filter can be used to specify the starting point of the directory from where the data needs to be imported (eg DC=savlynt,DC=com will import all data within this OU)</small>	<input type="text" value="DC=talanos,dc=local"/>
OBJECTFILTER <small>Object filter is a search query expressed in the form of a logical expression. BaseObject Filter is used to filter the objects that will be returned (it can be either JSON or string. Eg 1 JSON : (incremental: (&(&ObjectCategory=person)(objectClass=user))) (&(&createDateDatt)=&(&lastCreateDate&))(&updateDateDatt)=&(&astUpdateDateStr))) Eq 2 : (&(&ObjectCategory=person)(objectClass=user)(sAMAccountName=*))((userAccountControl:1.2.840.113556.1.4.803:=2)) will import all active users)</small>	<input type="text" value="(&(objectCategory=person)(objectClass=user)"/>
IMPORTJSON <small>JSON to specify LDAP Import Parameters - e.g { 'envproperties' : { 'com.sun.jndi.ldap.connect.timeout': '90000', 'com.sun.jndi.ldap.read.timeout': '50000' } }</small>	<input type="text"/>
ACCOUNT_ATTRIBUTE <small>Controls the LDAP Attribute to Savlynt Account Mapping and ACCOUNTID parameter is a mandatory parameter (eg - [DISPLAYNAME:displayName#String,NAME:employeeID#String,NAME:sAMAccountName#String,COMMENTS:distinguishesName#String,STATUS:userAccountControl#number,ACCOUNTID:objectid#String,RECONCLATION_FIELD:ACCOUNTID])</small>	<input type="text" value="[status:userAccountControl#String, CUSTOMPROPERTY24:employeeID#String, CUSTOMPROPERTY1:CN#String, CUSTOMPROPERTY2:name#String, "/>
ENTITLEMENT_ATTRIBUTE <small>LDAP Attribute which stores the entitlements. For AD it memberOf</small>	<input type="text" value="memberOf"/>
PAGE_SIZE <small>Page size defines the number of objects to be returned from each LDAP operation for AD the default is 1000</small>	<input type="text" value="1000"/>
USER_ATTRIBUTE <small>Controls the LDAP Attribute to Savlynt User Mapping. For Incremental import to work, CREATEDATE and UPDATEDATE fields must be mapped. (eg - [DISPLAYNAME:displayName#String,COMPANYNAME:company#String,lastname:lastName#String,firstname:givenName#String,TITLE:title#String,PHONENUMBER:telephoneNumber#String,COUNTRY:country#String,ENDDATE:accountExpires#millisecond,location:physicalDeliveryOfficeName#String,employeetype:extensionAttribute2#String,STREET:streetAddress#String,DEPARTMENTNAME:department#String,username:sAMAccountName#String,owner:manager#String,CREATEDATE:whenCreated#date,employeeid:employeeID#String,email:mail#String,State:state#String,City:city#String,COMMENTS:distinguishesName#String,statuskey:userAccountControl#number])</small>	<input type="text"/>
DEFAULT_USER_ROLE <small>Default SAV Role to be assigned to all the new users that gets imported - only valid for User Import</small>	<input type="text"/>
ENDPOINTS_FILTER <small>This field provides details as to which endpoint in Savlynt should the AD accounts be associated to after the accounts have been imported</small>	<input type="text"/>

CREATEACCOUNTJSON

JSON to specify the Field Value which will be used to Create the New Account Objects Exposed -{task,user,manager,account,all the objects defined in CHECKFORUNIQUE} -
eg({samaccountname:\$account[sn:\$user.lastname].cn:\$cn},objectclass {top,person,organizationalPerson,user},userprincipalname:\$userPrincipalName@saviyntdc.com))

```
{
  "accountExpires": "0",
  "cn": "$cn",
  "co": "${user.country}",
  "department": "${user.departmentname}",
  "displayName": "${user.displayName}",
  "employeeID": "${user.employeeid}",
  "employeenumber": "1",
  "employeetype": "${user.employeeType}",
  "givenName": "${user.firstname}",
  "l": "${user.city}",
  "mail": "${user.email}",
  "name": "${user.displayName}",
  "objectClass": [
    "top",
    "person",
    "organizationalPerson",
    "user"
  ],
  "physicaldeliveryofficename": "${user.employeeid}",
  "manager": "${managerAccount?.accountID}",
  "pwdLastSet": "0",
  "sAMAccountName": "${task.accountName}",
  "sn": "${user.lastname}",
  "st": "${user.customproperty10}",
  "streetAddress": "${user.street}",
  "title": "${user.title}"
}
```

UPDATEACCOUNTJSON

JSON to specify the Field Value which will be used to Update existing Account Objects Exposed -{task,user,manager,account,all the objects defined in CHECKFORUNIQUE} -
eg({samaccountname:\$account[sn:\$user.lastname].cn:\$cn},objectclass {top,person,organizationalPerson,user},userprincipalname:\$userPrincipalName@saviyntdc.com))

```
{
  "sn": "${user.lastname}",
  "manager": "${managerAccount.accountID}",
  "title": "${user.title}",
  "department": "${user.departmentname}",
  "displayName": "${user.displayName}",
  "employeeID": "${user.employeeid}",
  "employeenumber": "1",
  "employeetype": "${user.employeeType}",
  "company": "${user.company}",
}
```

REUSEACCOUNTJSON

JSON to specify the different attributes that needs to be checked and the ou where the account should be moved after re-using the account for

eg({ATTRIBUTESTOCHECK: {userAccountControl:514,sn:\$user.lastname}.cn:\$user.firstname},REUSEACCOUNTOU:OU=CloudUsers,DC=saviyntazure,DC=com))

```
{
  "ATTRIBUTESTOCHECK": {
    "userAccountControl": "514",
    "sn": "${user.lastname}",
    "cn": "${user.firstname}"
  },
  "REUSEACCOUNTOU": "OU=Accounts,DC=talanos,DC=local"
}
```

UPDATEUSERJSON

JSON to specify the Field Value which will be used to Update the Existing User Objects Exposed -{task,user,manager} -
eg({l:\$user.email,sn:\$user.lastname,accountExpires:\$user.termDate !=null ? 10000*(user.termDate.getTime() + 11644473600000) : 0}))

ENABLEACCOUNTJSON

JSON to specify the different attributes to be checked and action to be performed for enabling a disabled account eg({
DISABLEACCOUNTCHECKRULE: { CN=\${user.firstname}
\$user.lastname}.cn=Users,dc=corpAD,dc=saviyntdc=com,
CN=\${user.firstname}
\$user.middleName}.cn=Users,dc=corpAD,dc=saviyntdc=com},ENABLEACCOUNT
OU : OU=CloudUsers,DC=saviyntazure,DC=com,ATTRIBUTESTOCHECK:
{userAccountControl:514,sn:\$user.lastname,cn:\$user.firstname} ,
REMOVEGROUPS : YES , USEDINFROMACCOUNT : YES, MOVEDN : NO,
AFTERENABLEACTIONS : { userAccountControl : 512 }))

RESETANDCHANGEPASSWORDJSON

JSON to Reset and Change Password. If no json is defined, then the default change password actions will be taken. Critical LDAP attributes can also be modified using CRITICALCONTROLDATA block (optional) in this json. Example:
{"RESET":{"initials":"X","telephoneNumber":"XXXXXXXXX"},"CHANGE":{"initials":"X","telephoneNumber":"XXXXXXXXX"},"CRITICALCONTROLDATA":{"ATTRIBUTES":{"RESET":{"pwdReset":"true"},"CHANGE{}}},"CONTROLINIPARAMS":{"id":"X.X.X.X.X.X.X.X.X.X","criticality":"true","value":null}}

BASE

Base Search filter eg(DC=saviyntazure,DC=com) to be used for provisioning searches

DC=talanos,DC=local

FILTER

Filters the objects that will be returned

ACCOUNTNAMERULE

Specify the Rules to generate the DN for the Account for provisioning separated by ### eg(CN={user.firstName}
S(user.lastName),CN=Users,DC=saviyntazure,DC=com###CN={user.lastName}
S(user.firstName),CN=Users,DC=saviyntazure,DC=com)

```
CN=S({users.systemUserName}),OU=Accounts,DC=talanos,DC=local###cn=S({account}),OU=Accounts,DC=talanos,DC=local
```

ADVSEARCH**CHECKFORUNIQUE**

JSON to specify the attributes what needs to be checked in LDAP for unique constraint with rules separated by ### eg {{userPrincipalName: S(user.firstName)###S(user.firstName).S(user.lastName))}

```
{  
  "cn": "S({user.systemUserName})###S({account})",  
  "samaccountname": "S({account})###S({user.systemUserName})",  
  "userPrincipalName": "S({account})@saviynt.com###S({user.systemUserName})@talanos.local"  
}
```

REUSEINACTIVEACCOUNT

While Creating Account if the application should try to re-use the inactive account - TRUE or FALSE, Default is TRUE

```
FALSE
```

REMOVEACCOUNTACTION

JSON to specify the action that needs to be taken when the account is removed eg
{[removeAction:SUSPEND,moveUserToOU:OU=CloudUsers,DC=saviyntazure,DC=com,deleteAllGroups:Yes,userAccountControl:514]} will set the status of the account to 514,Delete all Groups and move the user to the OU Specified, To delete replace SUSPEND with DELETE

```
{  
  "removeAction": "DELETE",  
  "deleteAllGroups": "No",  
  "userAccountControl": "514"  
}
```

SETRANDOMPASSWORD

Set the Random Password when creating the new account - TRUE or FALSE

```
....
```

PASSWORD_MIN_LENGTH

Specify the Min length for the random password