JiJi

Active Directory Reports

User Manual

Table of Contents

1. Introduction

JiJi Active Directory Reports (ADR) enables IT organizations to extract vital data from Active Directory in seconds after installation. ADR provides the best solution to meet Active Directory Reporting requirements. ADR has an extensive reports list with over 150 out of box reports. Armed with this information, organizations can quickly make strategic and tactical security decisions that involve their Active Directory and Windows environment.

Active Directory Reporter retrieves and reports information efficiently from the active directory while hiding the complexities of the native Active Directory reporting tools. Active Directory Reporter can generate reports to help organizations gather information for regulatory audits including SOX/PCI/HIPAA audits.

2. Benefits of Active Directory Reports

JiJi Active Directory Reports allows an administrator to accurately retrieve required information about Active Directory Infrastructure and Objects quickly and displays it in a clear and logical format. Active Directory Reporter's interface accurately extracts data, saving time involved in troubleshooting, controlling and managing attributes of the active directory objects such as:

- Users
- Groups
- Computers
- Exchange Servers
- Organizational Units (OU)
- Group Policy Objects
- Printers
- Contacts

3. Features

- **Actions**

  Helps to do actions like delete, move, disable, enable and reset based on the reports in bulk.

- **Restore**

  Helps to restore the deleted users/computers in bulk.

- **Scheduler**

  Help to schedule the automatic generation of the reports. The generated reports are sent via E-mail.

- **Bulk Report Generation**

  Help to generate set of selected reports and provides option to save and send via E-mail.

- **Domain Controller Settings**

Users can specify the list of domain controllers not to be used. Also user can specify which domain controller to be used by default. By using this setting, the user can isolate the faulty, far away domain controller during the report generation.

- **Search**

  Help to locate a specific Active Directory Object quickly and accurately.

- **Print Reports**

  Help to print reports.

- **Export Reports**

  Reports can be exported to PDF, CSV and Excel formats.

- **Add/Remove Columns**

  Help to customize the report columns. It provide option to list all the schema attributes based on the report type.

- **PowerShell Scripting Support**

  Reports Generation can be automated using powershell script.

- **Scope**

  The report generation can be limited to Organizational Units (OU) in a domain, facilitating an OU based administration.

- **Sort**

  Users can sort the columns of their interest.

- **Custom reports**

  Users can generate the reports based on the user-defined custom LDAP query

- **Templates**

  It is a sub-feature added to **Bulk Report Generation** and **Scheduler**. It helps to store set of reports and its arguments in memory. And provides option to reuse the stored reports.

4. Report Categories

Active Directory Reports out of the box reports are divided into the following categories.

- Active Directory User Reports
- Active Directory Group Reports
- Active Directory Computer Reports
- Active Directory Exchange Reports
- Active Directory GPO Reports
- Active Directory OU Reports
- Active Directory Security Reports
- Active Directory NTFS Reports

- Active Directory Other Reports

5. Report Generation

This section lists the reports available in each of the categories, provide the filter used and PowerShell command for each of the report category.

5.1 Active Directory User Reports

5.1.1.General Reports

All Users

It provides the details of all the users in the selected scope.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370))"
```

PowerShell Command:

```
Export-ADReport -Type AllUsers -FilePath "C:\Reports\AllUsers.pdf"
```

Users With Empty Attributes

It provides the list of users whose specified attributes are empty. User can either check against all the specified attributes as empty or even one of the specified attribute.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=Person)(objectClass=user)(&(!attribute1=*)(!attribute2=*)))"
```

The above filter is used to check against all the specified attributes as empty.

```
"(&(objectCategory=Person)(objectClass=user)(|(!attribute1=*)(!attribute2=*)))"
```

The above filter is used to check even one of the specified attributes as empty.

PowerShell Command:

```
Export-ADReport -Type UsersWithoutManagers -FilePath
"C:\Reports\UsersWithoutManagers.pdf"
```

Managers

It provides details of all the managers in the selected scope.

How it works:

The report is generated by querying the Directory Service with the filter

```
“(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)
(manager=*))”
```

PowerShell Command:

```
Export-ADReport -Type Managers -FilePath "C:\Reports\Managers.csv"
```

Users without Managers

It provides the list of users who do not have any managers assigned to them.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)(!
manager=*))"
```

PowerShell Command:

```
Export-ADReport -Type UsersWithoutManagers -FilePath
"C:\Reports\UsersWithoutManagers.pdf"
```

Manager Based Users

It provides the list of users that directly report to the selected user (Manager). The users listed in report are those who have the manager property set to this selected user.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(& (objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)
(manager=managerDN))"
```

PowerShell Command:

```
Export-ADReport -Type ManagerBasedUser -FilePath "C:\Reports\ManagerBasedUser.pdf"
-arguments "CN=Administrator,CN=Users,DC=JiJiTechnologies,DC=com"
```

Users in more than One Group

It provides the details of users who belong to more than one group.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)
(memberOf=*))"
```

PowerShell Command:

```
Export-ADReport -Type UsersInMoreThanOneGroup -FilePath
"C:\Reports\UsersInMoreThanOneGroup.pdf"
```

Users with Domain Users as Primary Group

It provides the details of users who has domain users as primary group.

How it works:

The report is generated by querying the Directory Service with the filter

```
“"(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)
(primarygroupid=513))"”
```

PowerShell Command:

```
Export-ADReport -Type UsersWithDomainUsersAsPrimaryGroup -FilePath
"C:\Reports\UsersWithDomainUsersAsPrimaryGroup.csv"
```

Users without Domain Users as Primary Group

It provides the details of users who has primary group other than domain users.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)(!
primarygroupid=513))"
```

PowerShell Command:

```
Export-ADReport -Type UsersWithoutDomainUsersAsPrimaryGroup -FilePath
"C:\Reports\UsersWithoutDomainUsersAsPrimaryGroup.csv"
```

Recently Created Users

It provides the details of the user accounts created recently.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)
(createTimeStamp>=givenTime))"
```

PowerShell Command:

```
Export-ADReport -Type RecentlyCreatedUsers -FilePath
"C:\Reports\RecentlyCreatedUsers.pdf" -Arguments 7
```

Recently Modified Users

This report generates the lists of user accounts modified recently.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)
(modifyTimeStamp>=givenTime))"
```

PowerShell Command:

```
Export-ADReport -Type RecentlyModifiedUsers -FilePath
"C:\Reports\RecentlyModifiedUsers.pdf" -arguments 7
```

Dial-in Allow Access

This report generates the list of users who have access to dial-in.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)
(msNPAllowDialin=TRUE))"
```

PowerShell Command:

```
Export-ADReport -Type DialInAllowAccess -FilePath
"C:\Reports\DialInAllowAccess.pdf"
```

Dial-in Deny Access

This report generates the list of users who don't have access to dial-in.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)(|
(msNPAllowDialin=FALSE)(!msNPAllowDialin=*)))"
```

PowerShell Command:

```
Export-ADReport -Type DialInDenyAccess -FilePath "C:\Reports\DialInDenyAccess.pdf"
```

Users with Logon Script

This report generates the list of users who have logon scripts. Logon scripts are those which run automatically when the user logon.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)
(scriptPath=*))"
```

PowerShell Command:

```
Export-ADReport -Type UsersWithLogonScript -FilePath
"C:\Reports\UsersWithLogonScript.pdf"
```

Users without Logon Script

This report generates the list of users who don't have logon scripts. Logon scripts are those which run automatically when the user logon.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)(!
scriptPath=*))"
```

PowerShell Command:

```
Export-ADReport -Type UsersWithoutLogonScript -FilePath
"C:\Reports\UsersWithoutLogonScript.pdf"
```

Users with Profile

This report generates the list of users who have profile path.

How it works:

The report is generated by querying the Directory Service with the filter

```
“(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)
(profilePath=*))”
```

PowerShell Command:

```
“Export-ADReport -Type UsersWithProfile -FilePath
"C:\Reports\UsersWithProfile.csv"”
```

Users without Profile

This report generates the list of users who do not have profile path.

How it works:

The report is generated by querying the Directory Service with the filter

```
“(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)(!
profilePath=*))”
```

PowerShell Command:

```
Export-ADReport -Type UsersWithoutProfile -FilePath
"C:\Reports\UsersWithoutProfile.csv"
```

Users with Share

This report generates the list of users who have share.

How it works:

This report is generated by querying the Directory Service with the filter

```
“(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)
(homedirectory=*))”
```

PowerShell Command:

```
Export-ADReport -Type UsersWithShare -FilePath "C:\Reports\UsersWithShare.csv"
```

Users without Share

This report generates the list of users who do not have share.

How it works:

This report is generated by querying the Directory Service with the filter

```
“(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)(!
homedirectory=*))”
```

PowerShell Command:

```
Export-ADReport -Type UsersWithoutShare -FilePath
"C:\Reports\UsersWithoutShare.csv"
```

User with Local Share

This report generates the list of users who have local share

How it works:

This report is generated by querying the Directory Service with the filter

```
“(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)
(homedirectory=*)(!homedirectory=\\\\*))”
```

PowerShell Command:

```
Export-ADReport -Type UsersWithLocalShare -FilePath
"C:\Reports\UsersWithLocalShare.csv
```

Users with Network Share

This report generates the list of users who have network share.

How it works:

This report is generated by quering the Directory Service with the filter

```
“(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)
(homedirectory=\\\\*))”
```

PowerShell Command:

```
Export-ADReport -Type UsersWithNetworkShare -FilePath
"C:\Reports\UsersWithNetworkShare.csv"
```

All Deleted Users

This report generates the list of all deleted users in the domain.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectClass=user)(!objectClass=computer)(isDeleted=TRUE))"
```

PowerShell Command:

```
Export-ADReport -Type AllDeletedUsers -FilePath "C:\Reports\AllDeletedUsers.pdf"
```

Recently Deleted Users

This report generates the list of all user account deleted recently in the domain.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectClass=user)(!objectClass=computer)(isDeleted=TRUE)
(whenChanged>=givenTime))"
```

PowerShell Command:

```
Export-ADReport -Type RecentlyDeletedUsers -FilePath
"C:\Reports\RecentlyDeletedUsers.pdf" -Arguments 7
```

5.1.2.Account Status Report

Enabled Users

This report generates the list of all enbled user accounts.

How it works:

This report is generated by querying the Directory Service with the filter

```
“(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)(!
userAccountControl:1.2.840.113556.1.4.803:=2))”
```

PowerShell Command:

```
Export-ADReport -Type EnabledUsers -FilePath "C:\Reports\EnabledUsers.csv"
```

Enabled Locked Users

This report generates the list of enabled and locked user accounts.

How it works:

This report is generated by querying the Directory Service with the filter

```
“(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)(!
userAccountControl:1.2.840.113556.1.4.803:=2)(lockouttime>=1))”
```

PowerShell Command:

```
Export-ADReport -Type EnabledLockedUsers -FilePath
"C:\Reports\EnabledLockedUsers.csv"
```

Enabled Unlocked Users

This report generates the list of enabled and unlocked user accounts.

How it works:

This report is generated by querying the Directory Service with the filter

```
“(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)(!
userAccountControl:1.2.840.113556.1.4.803:=2)(!lockouttime>=1))”
```

PowerShell Command:

```
Export-ADReport -Type EnabledUnlockedUsers -FilePath
"C:\Reports\EnabledUnlockedUsers.csv"
```

Disabled Users

This report generates the list of all disabled user accounts.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)
(userAccountControl:1.2.840.113556.1.4.803:=2))"
```

PowerShell Command:

```
Export-ADReport -Type DisabledUsers -FilePath "C:\Reports\DisabledUsers.pdf"
```

Disabled Locked Users

This report generates the list of all disabled and locked user accounts.

How it works:

The report is generated by querying the Directory Service with the filter

```
“(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)
(userAccountControl:1.2.840.113556.1.4.803:=2)(lockouttime>=1))”
```

PowerShell Command:

```
Export-ADReport -Type DisabledLockedUsers -FilePath
"C:\Reports\DisabledLockedUsers.csv"
```

Disabled or Locked Users

This report generates the list of all disabled or locked user accounts.

How it works:

This report is generated by querying the Directory Service with the filter

```
“(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)(|
(userAccountControl:1.2.840.113556.1.4.803:=2)(lockouttime>=1)))”
```

PowerShell Command:

```
Export-ADReport -Type DisabledOrLockedUsers -FilePath
"C:\Reports\DisabledOrLockedUsers.csv"
```

Disabled Unlocked Users

This report generates the list of all disabled and unlocked user accounts.

How it works:

This report is generated by querying the Directory Service with the filter

```
“(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)
(userAccountControl:1.2.840.113556.1.4.803:=2)(!lockouttime>=1))”
```

PowerShell Command:

```
Export-ADReport -Type DisabledUnlockedUsers -FilePath
"C:\Reports\DisabledUnlockedUsers.csv"
```

Locked Out Users

This report generates the list of all user accounts that have been locked out.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)
(lockouttime>=1))"
```

PowerShell Command:

```
Export-ADReport -Type LockedOutUsers -FilePath "C:\Reports\LockedOutUsers.pdf"
```

Unlocked Users

This report generates the list of all user accounts that have been unlocked.

How it works:

The report is generated by querying the Directory Service with the filter

```
“(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)(!
lockouttime>=1))”
```

PowerShell Command:

```
Export-ADReport -Type UnlockedUsers -FilePath "C:\Reports\UnlockedUsers.csv"
```

Account Expired Users

This report generates the list of all user accounts that have expired.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)(!
accountExpires=0)(!accountExpires=9223372036854775807)
(accountExpires<=currentTime))"
```

PowerShell Command:

```
Export-ADReport -Type AccountExpiredUsers -FilePath
"C:\Reports\AccountExpiredUsers.pdf"
```

Recently Account Expired Users

This report generates the list of all user accounts that have expired in the given number of days.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)(!
accountExpires=0)(!accountExpires=9223372036854775807)(accountExpires<=currentTime)
(accountExpires>=givenTime))"
```

PowerShell Command:

```
Export-ADReport -Type RecentlyAccountExpiredUsers -FilePath
"C:\Reports\RecentlyAccountExpiredUsers.pdf" -Arguments 7
```

Soon-to-Expire User Accounts

This report generates the list of all user accounts that will expire within the given number of days.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)(!
accountExpires=0)(!accountExpires=9223372036854775807)(!
accountExpires<=currentTime)(accountExpires<=givenTime))"
```

PowerShell Command:

```
Export-ADReport -Type SoonToExpireUserAccount -FilePath
"C:\Reports\SoonToExpireUserAccount.pdf" -Arguments 7
```

Account Never Expires

This report generates the list of all user accounts which will never expire.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)(|
(accountExpires=0)(accountExpires=9223372036854775807)))"
```

PowerShell Command:

```
Export-ADReport -Type AccountNeverExpires -FilePath
"C:\Reports\AccountNeverExpires.pdf"
```

Account Expires Between

This report generates the list of user accounts that expires within the given period of days.

How it works:

This report is generated by querying the Directory Service with the filter

```
“(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)(!
accountExpires=0)(!accountExpires=9223372036854775807)
(accountExpires>={giventime1})(accountExpires<={giventime2}))”
```

PowerShell Command:

```
Export-ADReport -Type AccountExpiresBetween -FilePath
"C:\Reports\AccountExpiresBetween.csv" -Arguments "6/6/2008","7/7/2009"
```

Users with Account Set to Expire

This report generates the list of users whose account set to expire.

How it works:

This report is generated by querying the Directory Service with the filter

```
“(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)(!
accountExpires=0)(!accountExpires=9223372036854775807))”
```

PowerShell Command:

```
Export-ADReport -Type UsersWithAccountSetToExpire -FilePath
"C:\Reports\UsersWithAccountSetToExpire.csv"
```

5.1.3.Logon Reports

Inactive Users

This report generates the list of all users who have not logged on for the past 'n' days. The inactive users are determined based on their last logon time. All the domain controllers are scanned for the last logon time to ensure accuracy. If any of the DC's could not be contacted while report generation, the report generation will fail.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)(|(!
lastlogon=*)(lastlogon<=givenTime)))"
```

PowerShell Command:

```
Export-ADReport -Type InactiveUsers -FilePath "C:\Reports\InactiveUsers.pdf"
-Arguments 7
```

Recently Logged on Users

This report generates the list of all users who have logged during the past 'n' days. The recently logged on users are determined based on their last logon time. All the domain controllers are scanned for the last logon time to ensure accuracy. If any of the DC's could not be contacted while report generation, the report generation will fail.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)
(lastlogon>=givenTime))
```

PowerShell Command:

```
Export-ADReport -Type RecentlyLoggedOnUsers -FilePath
"C:\Reports\RecentlyLoggedOnUsers.pdf" -Arguments 7
```

Users Never Logged On

This report generates the list of all users who have not logged on to the domain. The Users never logged on are determined based on their last logon time. All the domain controllers are scanned for the last logon time to ensure accuracy. If any of the DC's could not be contacted while report generation, the report generation will fail.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)(|
(lastlogon=0)(!lastlogon=*)))"
```

PowerShell Command:

```
Export-ADReport -Type UsersNeverLoggedOn -FilePath
"C:\Reports\UserNeverLoggedOn.pdf"
```

Recently Bad Logged on Users

This report generates the list of all users who tried to logon with bad password.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)
(badPasswordTime>=givenTime))"
```

PowerShell Command:

```
Export-ADReport -Type RecentlyBadLoggedOnUsers -FilePath
"C:\Reports\RecentlyBadLoggedOnUsers.pdf" -Arguments 7
```

5.1.4.Password Reports

Users whose Password Never Expires

This report generates the list of all users whose password never expires.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)
(userAccountControl:1.2.840.113556.1.4.803:=65536))"
```

PowerShell Command:

```
Export-ADReport -Type UsersWhosePasswordNeverExpires -FilePath
"C:\Reports\UsersWhosePasswordNeverExpires.pdf"
```

Password Expired Users

This report generates the list of all users whose passwords are expired.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!(sAMAccountType=805306370))(!
userAccountControl:1.2.840.113556.1.4.803:=65536)(!pwdLastSet=0)(pwdLastSet<=time
based on maximum password age))"
```

PowerShell Command:

```
Export-ADReport -Type PasswordExpiredUsers -FilePath
```

```
"C:\Reports\PasswordExpiredUsers.pdf" -Arguments 42
```

Soon-to-Expire User Passwords

This report generates the list of all users whose passwords will expire in ‘n’ days

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!(sAMAccountType=805306370))(!
userAccountControl:1.2.840.113556.1.4.803:=65536)(!pwdLastSet<={0})
(pwdLastSet<=time based on maximum password age and the given time))"
```

PowerShell Command:

```
$arg = 42,7
```

```
Export-ADReport -Type SoonToExpireUserPassword -FilePath
"C:\Reports\SoonToExpireUserPassword.pdf" -Arguments $arg
```

Password Expires Between

This report generates the list of users whose password expires within the given period of days.

How it works:

This report generates by querying the Directory Service with the filter

```
“(&(objectCategory=person)(objectClass=user)(!(sAMAccountType=805306370))(!
userAccountControl:1.2.840.113556.1.4.803:=65536)(pwdLastSet>={giventime1})
(pwdLastSet<={giventime2}))"”
```

PowerShell Command:

```
Export-ADReport -Type PasswordExpiresBetween -FilePath
"C:\Reports\PasswordExpiresBetween.csv" -Arguments "6/6/2008","7/7/2009"
```

Password Changed Users

This report generates the list of all users whose passwords are modified during the given ‘n’ days

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)(!
pwdLastSet=0)(!pwdLastSet<=givenTime))"
```

PowerShell Command:

```
Export-ADReport -Type PasswordChangedUsers -FilePath
"C:\Reports\PasswordChangedUsers.pdf" -Arguments 7
```

Password Unchanged Users

This report generates the list of all users whose passwords are not modified during the given ‘n’ days

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)(!
pwdLastSet=0)(!pwdLastSet>=givenTime))"
```

PowerShell Command:

```
Export-ADReport -Type PasswordUnchangedUsers -FilePath
"C:\Reports\PasswordUnchangedUsers.pdf" -Arguments 7
```

Users with Password Set to Expire

This report generates the list of users whose password set to expire.

How it works:

This report is generated by querying the Directory Service with the filter

```
“(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)(!
userAccountControl:1.2.840.113556.1.4.803:=65536))”
```

PowerShell Command:

```
Export-ADReport -Type UsersWithPasswordSetToExpire -FilePath
"C:\Reports\UsersWithPasswordSetToExpire.csv"
```

Password Required Users

This report generates the list of user accounts which requires password.

How it works:

This report is generated by querying the Directory Service with the filter

```
“(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)(!
userAccountControl:1.2.840.113556.1.4.803:=32))”
```

PowerShell Command:

```
Export-ADReport -Type PasswordRequiredUsers -FilePath
"C:\Reports\PasswordRequiredUsers.csv"
```

Password Not Required Users

This report generates the list of users accounts which does not requires password.

How it works:

This report is generated by querying the Directory Service with the filter

```
“(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)
(userAccountControl:1.2.840.113556.1.4.803:=32))”
```

PowerShell Command:

```
Export-ADReport -Type PasswordNotRequiredUsers -FilePath
"C:\Reports\PasswordNotRequiredUsers.csv"
```

Password must change on next Logon

This report generates the list of users whose password must change on next logon.

How it works:

This report is generated by querying the Directory Service with the filter

```
“(&(objectCategory=person)(objectClass=user)(!sAMAccountType=805306370)
(pwdlastset=0))”
```

PowerShell Command:

```
Export-ADReport -Type PasswordMustChangeOnNextLogon -FilePath
"C:\Reports\PasswordMustChangeOnNextLogon.csv"
```

5.2 Active Directory Group Reports

5.2.1.General Reports

All Groups

This report generates the list of all groups within the given scope.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(objectCategory=group)"
```

PowerShell Command:

```
Export-ADReport -Type AllGroups -FilePath "C:\Reports\AllGroups.pdf"
```

Top Big Groups

This report generates the list of top ‘n’ large groups based on the members count.

How it works:

The report is generated by querying the Directory Service for all groups and list top ‘n’ groups based on the ‘member’ attribute.

PowerShell Command:

```
Export-ADReport -Type TopBigGroups -FilePath "C:\Reports\TopBigGroups.pdf"
-Arguments 5
```

Groups with Members

This report generates the list of groups that have members.

How it works:

This report is generated by querying the Directory Service with the filter

```
“(&(objectCategory=group)(member=*))”
```

PowerShell Command:

```
Export-ADReport -Type GroupsWithMembers -FilePath
"C:\Reports\GroupsWithMembers.csv"
```

Groups without Members

This report generates the list of groups without members.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=group)(!member=*))"
```

PowerShell Command:

```
Export-ADReport -Type GroupsWithoutMembers -FilePath
"C:\Reports\GroupsWithoutMembers.pdf"
```

Managed Groups

This report generates the list of all groups that have managers.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=group)(managedby=*))"
```

PowerShell Command:

```
Export-ADReport -Type ManagedGroups -FilePath "C:\Reports\ManagedGroups.pdf"
```

Unmanaged Groups

This report generates the list of all groups without managers.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=group)(!managedby=*))"
```

PowerShell Command:

```
Export-ADReport -Type UnmanagedGroups -FilePath "C:\Reports\UnmanagedGroups.pdf"
```

Group Members

This report generates the list of all users within the selected group.

How it works:

The report searches the selected group recursively and returns all nested group members.

PowerShell Command:

```
Export-ADReport -Type GroupMembers -FilePath "C:\Reports\GroupMembers.pdf"
-Arguments "LDAP://CN=Administrators,CN=Builtin,DC=JiJiTechnologies,DC=Com"
```

All Deleted Groups

This report generates the list of all deleted groups in the domain.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectClass=user)(!objectClass=computer)(isDeleted=TRUE))"
```

PowerShell Command:

```
Export-ADReport -Type AllDeletedGroups -FilePath "C:\Reports\AllDeletedGroups.pdf"
```

Recently Deleted Groups

This report generates the list of all groups which are deleted recently in the domain.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectClass=group)(isDeleted=TRUE)(whenChanged>=givenTime))"
```

PowerShell Command:

```
Export-ADReport -Type RecentlyDeletedGroups -FilePath
"C:\Reports\RecentlyDeletedGroups.pdf" -Arguments 7
```

5.2.2.Type and Scope Reports

Security Groups

This report generates the list of all security groups available within the selected scope.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=group)(groupType:1.2.840.113556.1.4.804:=2147483648))"
```

PowerShell Command:

```
Export-ADReport -Type SecurityGroups -FilePath "C:\Reports\SecurityGroups.pdf"
```

Distribution Groups

This report generates the list of all distribution groups available within the selected scope.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=group)(!groupType:1.2.840.113556.1.4.804:=2147483648))"
```

PowerShell Command:

```
Export-ADReport -Type DistributionGroups -FilePath
"C:\Reports\DistributionGroups.pdf"
```

Local Security Groups

This report generates the list of all local security groups available within the selected scope..

PowerShell Command:

```
Export-ADReport -Type LocalSecurityGroups -FilePath
"C:\Reports\LocalSecurityGroups.csv"
```

Local Distribution Groups

This report generates the list of all local distribution groups available within the selected scopes.

PowerShell Command:

```
Export-ADReport -Type LocalDistributionGroups -FilePath
"C:\Reports\LocalDistributionGroups.csv"
```

Global Security Groups

This report generates the list of all global security groups available within the selected scope.

PowerShell Command:

```
Export-ADReport -Type GlobalSecurityGroups -FilePath
"C:\Reports\GlobalSecurityGroups.csv"
```

Global Distribution Groups

This report generates the list of all global distribution groups available within the selected scope.

PowerShell Command:

```
Export-ADReport -Type GlobalDistributionGroups -FilePath
"C:\Reports\GlobalDistributionGroups.csv"
```

Universal Distribution Groups

This report generates the list of all universal distribution groups available within the selected scope.

PowerShell Command:

```
Export-ADReport -Type UniversalDistributionGroups -FilePath
"C:\Reports\UniversalDistributionGroups.csv"
```

5.3 Active Directory Computer Reports

5.3.1.General Reports

All Computers

This report generates the list of all computers within the selected scope.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectcategory=computer)(objectClass=computer))"
```

PowerShell Command:

```
Export-ADReport -Type AllComputers -FilePath "C:\Reports\AllComputers.pdf"
```

Workstations

This report generates the list of all workstations within the selected scope. Workstations are all computers except Servers and Domain Controllers.

How it works:

The report is generated by querying the Directory Service for all computers with "userAccountControl=ADS_UF_WORKSTATION_TRUST_ACCOUNT". The filter is

"(&(objectCategory=computer)(objectClass=computer)(userAccountControl: 1.2.840.113556.1.4.803:=4096))"

PowerShell Command:

```
Export-ADReport -Type Workstations -FilePath "C:\Reports\Workstations.pdf"
```

Domain Controllers

This report generates the list of all Domain Controllers within the selected scope.

How it works:

The report is generated by querying the Directory Service for all computers with

"userAccountControl=ADS_UF_SERVER_TRUST_ACCOUNT". The filter is

```
"(&(objectCategory=computer)(objectClass=computer)(userAccountControl:
1.2.840.113556.1.4.803:=8192))"
```

PowerShell Command:

```
Export-ADReport -Type DomainControllers -FilePath
"C:\Reports\DomainControllers.pdf"
```

OS Based

This report provides the details of the computers based on the given Operating System type.

How it works:

The report is generated by querying the Directory Service for all computers with the attributes "operatingSystem" and "operatingSystemServicePack".

PowerShell Command:

```
$arg = "WindowsXPwithallSP"

Export-ADReport -Type OSBased -FilePath "C:\Reports\OSBased.pdf" -Arguments $arg
```

Computers Trusted for Delegation

This report generates the list of all computers that are trusted for delegation.

How it works:

The report is generated by querying the Directory Service for all computers with "userAccountControl=ADS_UF_TRUSTED_FOR_DELEGATION". The filter is

```
"(&(objectCategory=computer)(objectClass=computer)(userAccountControl:
1.2.840.113556.1.4.803:=524288))"
```

PowerShell Command:

```
Export-ADReport -Type ComputersTrustedForDelegation -FilePath
"C:\Reports\ComputersTrustedForDelegation.pdf"
```

Recently Modified Computers

This report generates the list of all computers that are modified recently.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=computer)(objectClass=computer)(modifyTimeStamp>=givenTime))"
```

PowerShell Command:

```
Export-ADReport -Type RecentlyModifiedComputers -FilePath
"C:\Reports\RecentlyModifiedComputers.pdf" -Arguments 7
```

Managed Computers

This report generates the list of all computers that are managed by any of the user.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectcategory=computer)(objectClass=computer)(managedBy=*))"
```

PowerShell Command:

```
Export-ADReport -Type ManagedComputers -FilePath "C:\Reports\ManagedComputers.pdf"
```

Unmanaged Computers

This report generates the list of all computers that are not managed by any of the user.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectcategory=computer)(objectClass=computer)(!managedBy=*))"
```

PowerShell Command:

```
Export-ADReport -Type UnmanagedComputers -FilePath
"C:\Reports\UnmanagedComputers.pdf"
```

All Deleted Computers

This report generates the list of all deleted computers.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectClass=computer)(isDeleted=TRUE))"
```

PowerShell Command:

```
Export-ADReport -Type AllDeletedComputers -FilePath
"C:\Reports\AllDeletedComputers.pdf"
```

Recently Deleted Computers

This report generates the list of all computers which are deleted during the last ‘n’ days.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectClass=computer)(isDeleted=TRUE)(whenChanged>=givenTime))"
```

PowerShell Command:

```
Export-ADReport -Type RecentlyDeletedComputers -FilePath
"C:\Reports\RecentlyDeletedComputers.pdf" -Arguments 7
```

Computers with Domain Computers as Primary Group

It provides the details of computers which have domain computers as primary group.

How it works:

This report is generated by querying the Directory Service with the filter

```
“(&(objectcategory=computer)(objectClass=computer)(primarygroupid=515))”
```

PowerShell Command:

```
Export-ADReport -Type ComputersWithDomainComputersAsPrimaryGroup -FilePath
"C:\Reports\ComputersWithDomainComputersAsPrimaryGroup.csv"
```

Computers without Domain Computer as Primary Group

It provides the details of computers which have primary group other than domain computers.

How it works:

This report is generated by querying the Directory Service with the filter

```
“(&(objectcategory=computer)(objectClass=computer)(!primarygroupid=515))”
```

PowerShell Command:

```
Export-ADReport -Type ComputersWithoutDomainComputersAsPrimaryGroup -FilePath
"C:\Reports\ComputersWithoutDomainComputersAsPrimaryGroup.csv"
```

5.3.2.Account Status Reports

Inactive Computers

This report generates the details of the inactive computers for the given number of days. The inactive computers are determined based on their last logon time. All the domain controllers are scanned for the last logon time to ensure accuracy. If any of the DC's could not be contacted while generating report, the report generation will fail.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=computer)(objectClass=computer)(|(!lastlogon=*)
(lastlogon<=givenTime)))"
```

PowerShell Command:

```
Export-ADReport -Type InactiveComputers -FilePath
"C:\Reports\InactiveComputers.pdf" -Arguments 7
```

Disabled Computers

This report generates the list of all computers that are disabled.

How it works:

The report is generated by querying the Directory Service for all computers with "userAccountControl=ADS_UF_ACCOUNTDISABLE". The filter is

```
"(&(objectCategory=computer)(objectClass=computer)(userAccountControl:
1.2.840.113556.1.4.803:=2))"
```

PowerShell Command:

```
Export-ADReport -Type DisabledComputers -FilePath
"C:\Reports\DisabledComputers.pdf"
```

5.4 Active Directory Exchange Reports

5.4.1.General Reports

Mailbox enabled users

This report generates the list of all mailbox enabled users on the exchange server within the given scope.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(mailnickname=*)(|(homeMDB=*)
(msExchHomeServerName=*)))"
```

PowerShell Command:

```
Export-ADReport -Type MailboxEnabledUsers -FilePath
"C:\Reports\MailboxEnabledUsers.pdf"
```

Mail enabled users Report

This report generates the list of all mail enabled users on the exchange server within the given scope.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(mailnickname=*)(targetAddress=*))"
```

PowerShell Command:

```
Export-ADReport -Type MailEnabledUsers -FilePath "C:\Reports\MailEnabledUsers.pdf"
```

Mailbox enabled Groups

This report generates the list of all mailbox enabled groups on the exchange server.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=Group)(mailNickname=*))"
```

PowerShell Command:

```
Export-ADReport -Type MailEnabledGroups -FilePath
"C:\Reports\MailEnabledGroups.pdf"
```

Users with Email Proxy Enabled

This report generates the list of all users with the given email proxy address.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectcategory=person)(objectClass=user)(proxyAddresses=*givenProxyAddress*))"
```

PowerShell Command:

```
Export-ADReport -Type UsersWithEmailProxyEnabled -FilePath
"C:\Reports\UsersWithEmailProxyEnabled.pdf" -Arguments
"support@jijitechnologies.com"
```

Groups with Email Proxy Enabled

This report generates the list of all groups with the given email proxy address.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectcategory=group)(objectClass=group)(proxyAddresses=*givenProxyAddress*))"
```

PowerShell Command:

```
Export-ADReport -Type GroupsWithEmailProxyEnabled -FilePath
"C:\Reports\GroupsWithEmailProxyEnabled.pdf" -Arguments
"support@jijitechnologies.com"
```

Users Hidden from Exchange Address Lists

This report generates the list of all users with the mail address hidden from exchange address list.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(mailnickname=*)(|(homeMDB=*)
(msExchHomeServerName=*)(targetAddress=*))(msExchHideFromAddressLists=TRUE))"
```

PowerShell Command:

```
Export-ADReport -Type UsersHiddenFromExchageAddressLists -FilePath
"C:\Reports\UsersHiddenFromExchageAddressLists.pdf"
```

5.4.2.Distribution Lists

Distribution List Members

This report generates the list of all users and contacts who is a member of any distribution group.

How it works:

The report is generated by querying the Directory Service with the filter

```
(&(|(&(objectclass=user)(objectCategory=person))(&(objectclass=contact)
(objectCategory=person)))(|(memberOf=distributionGroups)))
```

PowerShell Command:

```
Export-ADReport -Type DistributionListMembers -FilePath
"C:\Reports\DistributionListMembers.pdf"
```

Non Distribution List Members

This report generates the list of all users and contacts who is not a member of any distribution group.

How it works:

The report is generated by querying the Directory Service with the filter

```
(&(|(&(objectclass=user)(objectCategory=person))(&(objectclass=contact)
(objectCategory=person)))(&(!memberOf=distributionGroups)))
```

PowerShell Command:

```
Export-ADReport -Type NonDistributionListMembers -FilePath
"C:\Reports\NonDistributionListMembers.pdf"
```

5.4.3.Mailbox Setting Reports

Default Deleted Item Retentions

This report generates the list of all users who have to use the default deleted item retention setting specified in the mailbox database.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(mailnickname=*)(|(!deletedItemflags=*)
(!deletedItemflags>=0)))"
```

PowerShell Command:

```
Export-ADReport -Type DefaultDeletedItemRetention -FilePath
"C:\Reports\DefaultDeletedItemRetention.pdf"
```

Deleted Item Retention Limits

This report generates the list of all users who have deleted item retention setting specified particularly.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(mailnickname=*)(deletedItemflags=*)(!
deletedItemflags<=0))"
```

PowerShell Command:

```
Export-ADReport -Type DeletedItemRetentionLimits -FilePath
"C:\Reports\DeletedItemRetentionLimits.pdf"
```

Default Storage Limit Report

This report generates the list of all users who have to use the default storage limit specified in the mailbox database.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(mDBUseDefaults=TRUE))"
```

PowerShell Command:

```
Export-ADReport -Type DefaultStorageLimit -FilePath
"C:\Reports\DefaultStorageLimit.pdf"
```

Mailbox Size Limits

This report generates the list of all users who have limited mailbox size specified particularly.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(mDBUseDefaults=FALSE))"
```

PowerShell Command:

```
Export-ADReport -Type MailboxSizeLimits -FilePath
"C:\Reports\MailboxSizeLimits.pdf"
```

5.4.4.Mail Flow Setting Reports

Default Sending Size

This report generates the list of all users who can send messages of the default sending size.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(mailnickname=*)(|(homeMDB=*)
(msExchHomeServerName=*))(!(submissionContLength=*)))"
```

PowerShell Command:

```
Export-ADReport -Type DefaultSendingSize -FilePath
"C:\Reports\DefaultSendingSize.pdf"
```

Restricted Sending Size

This report generates the list of all users who have restrictions on the size of the sending message.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(mailnickname=*)(|(homeMDB=*)
(msExchHomeServerName=*))(submissionContLength=*))"
```

PowerShell Command:

```
Export-ADReport -Type RestrictedSendingSize -FilePath
"C:\Reports\RestrictedSendingSize.pdf"
```

Default Recipient Size

This report generates the list of all users who can send messages to the default number of recipients.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(mailnickname=*)(|(homeMDB=*)
(msExchHomeServerName=*))(!(msExchRecipLimit=*)))"
```

PowerShell Command:

```
Export-ADReport -Type DefaultRecipientSize -FilePath
"C:\Reports\DefaultRecipientSize.pdf"
```

Restricted Recipient Size

This report generates the list of all users who have restriction on recipient’s number.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(mailnickname=*)(|(homeMDB=*)
(msExchHomeServerName=*))(msExchRecipLimit=*))"
```

PowerShell Command:

```
Export-ADReport -Type RestrictedRecipientSize -FilePath
"C:\Reports\RestrictedRecipientSize.pdf"
```

Default Receiving Size

This report generates the list of all users who can receive messages of default size.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(mailnickname=*)(|(homeMDB=*)
(msExchHomeServerName=*)(targetAddress=*))(!(delivContLength=*)))"
```

PowerShell Command:

```
Export-ADReport -Type DefaultReceivingSize -FilePath
"C:\Reports\DefaultReceivingSize.pdf"
```

Restricted Receiving Size

This report generates the list of all users who have restriction on the receiving message size.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(mailnickname=*)(|(homeMDB=*)
(msExchHomeServerName=*)(targetAddress=*))(delivContLength=*))"
```

PowerShell Command:

```
Export-ADReport -Type RestrictedReceivingSize -FilePath
"C:\Reports\RestrictedReceivingSize.pdf"
```

Accept Messages from Everyone

This report generates the list of all users who can receive message from all users.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(mailnickname=*)(|(homeMDB=*)
(msExchHomeServerName=*)(targetAddress=*))(!(authOrig=*))(!(unauthOrig=*)))"
```

PowerShell Command:

```
Export-ADReport -Type AcceptMessageFromEveryone -FilePath
"C:\Reports\AcceptMessageFromEveryone.pdf"
```

Accept Messages Restricted

This report generates the list of all users who have restriction in receiving messages from a set of users.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(mailnickname=*)(|(homeMDB=*)
(msExchHomeServerName=*)(targetAddress=*))(|(authOrig=*)(unauthOrig=*)))"
```

PowerShell Command:

```
Export-ADReport -Type AcceptMessageRestricted -FilePath
"C:\Reports\AcceptMessageRestricted.pdf"
```

Users Based on Forward To

This report generates the list of all users whose mails are forwarded to the given user.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectcategory=person)(objectClass=user)(altRecipient=givenUser))"
```

PowerShell Command:

```
Export-ADReport -Type UsersBasedOnForwardTo -FilePath
"C:\Reports\UsersBasedOnForwardTo.pdf" -Arguments
"CN=Administrator,CN=Users,DC=JiJiTechnologies,DC=Com"
```

5.4.5.Feature Based Reports

OMA Enabled

This report generates the list of all Outlook Mail Access enabled users.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(mailnickname=*)(|(homeMDB=*)
(msExchHomeServerName=*))(|(!(msExchOmaAdminWirelessEnable=*))(!
(msExchOmaAdminWirelessEnable:1.2.840.113556.1.4.803:=2))))"
```

PowerShell Command:

```
Export-ADReport -Type OMAEnabled -FilePath "C:\Reports\OMAEnabled.pdf"
```

OWA Disabled

This report generates the list of all Outlook web Access disabled users.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(mailnickname=*)(|(homeMDB=*)
(msExchHomeServerName=*))(|(protocolsettings=*http01*)(protocolsettings=*owa0*)))"
```

PowerShell Command:

```
Export-ADReport -Type OWADisabled -FilePath "C:\Reports\OWADisabled.pdf"
```

OWA Enabled

This report generates the list of all Outlook Web Access enabled users.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(mailnickname=*)(|(homeMDB=*)
(msExchHomeServerName=*))(!(|(protocolsettings=*http01*)
(protocolsettings=*owa0*))))"
```

PowerShell Command:

```
Export-ADReport -Type OWAEnabled -FilePath "C:\Reports\OWAEnabled.pdf"
```

POP3 Disabled

This report generates the list of all POP3 disabled users.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(mailnickname=*)(|(homeMDB=*)
(msExchHomeServerName=*))(protocolsettings=*POP301*))"
```

PowerShell Command:

```
Export-ADReport -Type POP3Disabled -FilePath "C:\Reports\POP3Disabled.pdf"
```

IMAP4 Disabled

This report generates the list of all IMAP4 disabled users.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=user)(mailnickname=*)(|(homeMDB=*)
(msExchHomeServerName=*))(protocolsettings=*IMAP401*))"
```

PowerShell Command:

```
Export-ADReport -Type IMAP4Disabled -FilePath "C:\Reports\IMAP4Disabled.pdf"
```

5.5 OU Reports

5.5.1.General Reports

All OUs

This report generates the list of all OUs within the selected scope.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(objectclass=organizationalUnit)"
```

PowerShell Command:

```
Export-ADReport -Type AllOUs -FilePath "C:\Reports\AllOUs.pdf"
```

Managed OUs

This report generates the list of OUs that have managers

How it works:

This report is generated by querying the Directory Service with the filter

```
"(&(objectclass=organizationalUnit)(managedby=*))"
```

PowerShell Command:

```
Export-ADReport -Type ManagedOUs -FilePath "C:\Reports\ManagedOUs.csv"
```

Unmanaged OUs

This report generates the list of OUs that do not have managers.

How it works:

This report is generated by querying the Directory Service with the filter

```
"(&(objectclass=organizationalUnit)(!managedby=*))"
```

PowerShell Commands:

```
Export-ADReport -Type UnmanagedOUs -FilePath "C:\Reports\UnmanagedOUs.csv"
```

GPO Inheritance Enabled OUs

This report generates the list of all OUs which are GPO inheritance enabled.

How it works

This report is generated by querying the Directory Service with the filter

```
"(&(objectclass=organizationalUnit)(!gpOptions=1))"
```

PowerShell Commands:

```
Export-ADReport -Type GPOInheritanceEnabledOUs -FilePath
"C:\Reports\GPOInheritanceEnabledOUs.csv"
```

Empty OUs

This report generates the list of all empty OUs within the selected scope.

How it works:

The report is generated by querying the Directory Service for all OUs that have child objects.

PowerShell Command:

```
Export-ADReport -Type EmptyOUs -FilePath "C:\Reports\EmptyOUs.pdf"
```

Users only OUs

This report generates the list of all OUs that contains only users.

How it works:

The report is generated by querying the Directory Service for all OUs that only have user objects.

PowerShell Command:

```
Export-ADReport -Type UsersOnlyOUs -FilePath "C:\Reports\UsersOnlyOUs.pdf"
```

Computers only OUs

This report generates the list of all OUs that contains only computers.

How it works:

The report is generated by querying the Directory Service for all OUs that only have computer objects.

PowerShell Command:

```
Export-ADReport -Type ComputersOnlyOUs -FilePath "C:\Reports\ComputersOnlyOUs.pdf"
```

Recently Created OUs

This report generates the list of all OUs that are created during the past given number of days.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectclass=organizationalUnit)(createTimeStamp>=givenTime))"
```

PowerShell Command:

```
Export-ADReport -Type RecentlyCreatedOUs -FilePath
"C:\Reports\RecentlyCreatedOUs.pdf" -Arguments 7
```

Recently Modified OUs

This report generates the list of all OUs that are modified during the past given number of days.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectclass=organizationalUnit)(modifyTimeStamp>=givenTime))"
```

PowerShell Command:

```
Export-ADReport -Type RecentlyModifiedOUs -FilePath
"C:\Reports\RecentlyModifiedOUs.pdf" -Arguments 7
```

GPO Linked OUs

This report generates the list of all OUs that have GPO link.

How it works:

The report is generated by querying the Directory Service for all OUs with the attribute ‘gpLink’ which has any GPO Link.

PowerShell Command:

```
Export-ADReport -Type GPOLinkedOUs -FilePath "C:\Reports\GPOLinkedOUs.pdf"
```

GPO Blocked inheritance OUs

This report generates the list of all OUs which are blocked from GPO inheritance.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectclass=organizationalUnit)(gpOptions=1))"
```

PowerShell Command:

```
Export-ADReport -Type GPOBlockedInheritanceOUs -FilePath
"C:\Reports\GPOBlockedInheritanceOUs.pdf"
```

All Deleted OUs

This report generates the list of all deleted OUs in the domain.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectclass=organizationalUnit)(isDeleted=TRUE))"
```

PowerShell Command:

```
Export-ADReport -Type AllDeletedOUs -FilePath "C:\Reports\AllDeletedOUs.pdf"
```

Recently Deleted OUs

This report generates the list of all OUs which are deleted recently.

How it works:

```
The report is generated by querying the Directory Service with the
filter"(&(objectClass=organizationalUnit)(isDeleted=TRUE)(whenChanged>=givenTime))"
```

PowerShell Command:

```
Export-ADReport -Type RecentlyDeletedOUs -FilePath
"C:\Reports\RecentlyDeletedOUs.pdf" -Arguments 7
```

5.6 GPO Reports

5.6.1.General Reports

All GPOs

This report generates the list of all group policy objects in the domain.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(objectClass=groupPolicyContainer)"
```

PowerShell Command:

```
Export-ADReport -Type AllGPOs -FilePath "C:\Reports\AllGPOs.pdf"
```

Recently Created GPOs

This report generates the list of all group policy objects which are created during the past ‘n’ days.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectclass=groupPolicyContainer)(createTimeStamp>=givenTime))"
```

PowerShell Command:

```
Export-ADReport -Type RecentlyCreatedGPOs -FilePath
"C:\Reports\RecentlyCreatedGPOs.pdf" -Arguments 5
```

Recently Modified GPOs

This report generates the list of all group policy objects which are modified during the past ‘n’ days.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectclass=groupPolicyContainer)(modifyTimeStamp>=givenTime))"
```

PowerShell Command:

```
Export-ADReport -Type RecentlyModifiedGPOs -FilePath
"C:\Reports\RecentlyModifiedGPOs.pdf" -Arguments 5
```

All Deleted GPOs

This report generates the list of all deleted group policy objects.

How it works:

The report is generated by querying the Directory Service with the filter

"(&(objectClass=computer)(isDeleted=TRUE))"

PowerShell Command:

Export-ADReport -Type AllDeletedGPOs -FilePath "C:\Reports\AllDeletedGPOs.pdf"

Recently Deleted GPOs

This report generates the list of all group policy objects which are deleted recently.

How it works:

The report is generated by querying the Directory Service with the filter

"(&(objectClass=groupPolicyContainer)(isDeleted=TRUE)(whenChanged>=givenTime)
)"

PowerShell Command:

Export-ADReport -Type RecentlyDeletedGPOs -FilePath "C:\Reports\RecentlyDeletedGPOs.pdf" -Arguments 7

5.6.2.Frequently Modified GPOs

Frequently Modified GPOs

This report generates the list of all group policy objects which are frequently modified.

How it works:

The report is generated by querying the Directory Service for the attribute ‘versionNumber’.

PowerShell Command:

```
Export-ADReport -Type FrequentlyModifiedGPOs -FilePath
"C:\Reports\FrequentlyModifiedGPOs.pdf" -Arguments 5
```

Frequently Modified Computer Settings GPOs

This report generates the list of all group policy objects whose computer settings are frequently modified.

How it works:

The report is generated by querying the Directory Service for the attribute ‘versionNumber’.

PowerShell Command:

```
Export-ADReport -Type FrequentlyModifiedComputerSettingsGPOs -FilePath "C:\Reports\
FrequentlyModifiedComputerSettingsGPOs.pdf" -Arguments 5
```

Frequently Modified User Settings GPOs

This report generates the list of all group policy objects whose user settings are frequently modified.

How it works:

The report is generated by querying the Directory Service for the attribute ‘versionNumber’.

PowerShell Command:

```
Export-ADReport -Type FrequentlyModifiedUserSettingsGPOs -FilePath
"C:\Reports\FrequentlyModifiedUserSettingsGPOs.pdf" -Arguments 5
```

5.6.3.Linked GPOs

Domain Linked GPOs

This report generates the list of all group policy objects which are linked to the domain.

How it works:

The report is generated by querying the Directory Service for all GPOs that are linked to the Domain.

PowerShell Command:

```
Export-ADReport -Type DomainLinkedGPOs -FilePath "C:\Reports\DomainLinkedGPOs.pdf"
```

OU Linked GPOs

This report generates the list of all group policy objects which are linked to OUs.

How it works:

The report is generated by querying the Directory Service for all GPOs that are linked to the OUs.

PowerShell Command:

```
Export-ADReport -Type OULinkedGPOs -FilePath "C:\Reports\OULinkedGPOs.pdf"
```

Site Linked GPOs

This report generates the list of all group policy objects which are linked to Sites.

How it works:

The report is generated by querying the Directory Service for all GPOs that are linked to the Sites.

PowerShell Command:

```
Export-ADReport -Type SiteLinkedGPOs -FilePath "C:\Reports\SiteLinkedGPOs.pdf"
```

5.6.4.Disabled and Unused GPO

Disabled GPOs

This report generates the list of all disabled GPOs.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectClass=groupPolicyContainer)(flags=3))"
```

PowerShell Command:

```
Export-ADReport -Type DisabledGPOs -FilePath "C:\Reports\DisabledGPOs.pdf"
```

All Settings Enabled GPOs

This report generates the list of all settings enabled GPOs.

How it works:

The report is generated by querying the Directory Service with the filter

```
“(&(objectClass=groupPolicyContainer)(flags=0))”
```

PowerShell Command:

```
Export-ADReport -Type AllSettingsEnabledGPOs -FilePath
"C:\Reports\AllSettingsEnabledGPOs.csv"
```

Computer Settings Enabled GPOs

This report generates the list of all computer settings enabled GPOs.

How it works:

The report is generated by querying the Directory Service with the filter

```
“(&(objectClass=groupPolicyContainer)(|(flags=0)(flags=1)))”
```

PowerShell Command:

```
Export-ADReport -Type ComputerSettingsEnabledGPOs -FilePath
"C:\Reports\ComputerSettingsEnabledGPOs.csv"
```

Computer Settings Disabled GPOs

This report generates the list of all GPOs with computer settings disabled.

How it works:

The report is generated by querying the Directory Service with the filter

"(&(objectClass=groupPolicyContainer)(|(flags=3)(flags=2)))"

PowerShell Command:

```
Export-ADReport -Type ComputerSettingsDisabledGPOs -FilePath
"C:\Reports\ComputerSettingsDisabledGPOs.pdf"
```

User Settings Enabled GPOs

This report generates the list of all GPOs with user settings enabled.

How it works:

This report is generated by querying the Directory Service with the filter

```
“(&(objectClass=groupPolicyContainer)(|(flags=0)(flags=2)))”
```

PowerShell Command:

```
Export-ADReport -Type UserSettingsEnabledGPOs -FilePath
"C:\Reports\UserSettingsEnabledGPOs.csv"
```

User Settings Disabled GPOs

This report generates the list of all GPOs with user settings disabled.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectClass=groupPolicyContainer)(|(flags=3)(flags=1)))"
```

PowerShell Command:

```
Export-ADReport -Type UserSettingsDisabledGPOs -FilePath
"C:\Reports\UserSettingsDisabledGPOs.pdf"
```

Unused GPOs

This report generates the list of all GPOs which are not used.

How it works:

The report is generated by querying the Directory Service for all GPOs that are not linked to any other objects in the domain. And the following filter is used

```
"(|(objectClass=domainDNS)(objectClass=organizationalUnit))"
```

PowerShell Command:

```
Export-ADReport -Type UnusedGPOs -FilePath "C:\Reports\UnusedGPOs.pdf"
```

5.7 Active Directory Security Reports

5.7.1.General Reports

Non-Inheritable Objects

This report generates the list of non-inheritable objects in the selected Directory Service container. Noninheritable objects are those that do not allow inheriting their permissions to its child objects.

PowerShell Command:

```
Export-ADReport -Type NonInheritableObjects -FilePath c:\NonInheritableObjects.pdf
-Arguments "LDAP://CN=Users,DC=JiJi,DC=local"
```

Users/Groups with Full Control

This report generates the list of the Active Directory objects where a specific user has full control over that object.

PowerShell Command:

```
$arg="LDAP://CN=Users,DC=JiJi,DC=local","LDAP://JiJi.local/CN=Administrator,CN=User
s,DC=JiJi,DC=local"
```

```
Export-ADReport -Type FullControlPermissionObjects -FilePath
d:\FullControlPermissionObjects.pdf -Arguments $arg
```

Users/Groups with Any Control

This report generates the list of the Active Directory objects where a specific user has permissions.

PowerShell Command:

```
$arg="LDAP://CN=Users,DC=JiJi,DC=local","LDAP://JiJi.local/CN=Administrator,CN=User
s,DC=JiJi,DC=local"
```

```
Export-ADReport -Type UserPermissionsOverObjects -FilePath
d:\UserPermissionsOverObjects.pdf -Arguments $arg
```

AD Object Permissions

This report generates the list of permission given by the selected AD object to the other objects in AD.

PowerShell Command:

```
$arg="LDAP://JiJi.local/CN=Administrator,CN=Users,DC=JiJi,DC=local"
```

```
Export-ADReport -Type ADObjectPermissions -FilePath d:\ADObjectPermissions.pdf
-Arguments $arg
```

5.8 Active Directory NTFS Reports

5.8.1.General Reports

Non-Inheritable Folders/Files

This report generates the list of all folders and files that are restricted to inherit the permissions from their parent objects.

PowerShell Command:

```
Export-ADReport -Type NonInheritableFoldersOrFiles -FilePath
d:\NonInheritableFoldersOrFiles.pdf -Arguments "D:\ActiveDirectory"
```

Users/Groups with Full Control

This report generates the list of all folders and files over which the specified user has full permission.

PowerShell Command:

```
$arg="D:\ActiveDirectory","LDAP://JiJi.local/CN=Administrator,CN=Users,DC=JiJi,DC=l
ocal"
```

```
Export-ADReport -Type FoldersOrFilesOverFullControl -FilePath
d:\FoldersOrFilesOverFullControl.pdf -Arguments $arg
```

Users/Groups with Any Control

This report generates the list of all folders and files over which the specified user has any permission.

PowerShell Command:

```
$arg="D:\ActiveDirectory","LDAP://JiJi.local/CN=Administrator,CN=Users,DC=JiJi,DC=l
ocal"
```

```
Export-ADReport -Type FoldersOrFilesOverAnyControl -FilePath
d:\FoldersOrFilesOverAnyControl.pdf -Arguments $arg
```

File/Folder Permissions

This report generates the list of permission given by the selected file or folder to the AD objects.

PowerShell Command:

```
Export-ADReport -Type FileOrFolderPermissions -FilePath
"C:\Reports\FileOrFolderPermissions.csv" -Arguments
"c:\Reports\AllContacts.csv","File"
```

5.9 Other Reports

5.9.1.Custom Report

User can provide their own LDAP query for the report generation.

PowerShell Command:

```
Export-ADReport -Type Custom -FilePath "C:\Custom.pdf"
```

5.9.2.Printer Report

All Printers

This report generates the list of all printers within the selected scope.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(objectclass=printQueue)"
```

PowerShell Command:

```
Export-ADReport -Type AllPrinters -FilePath "C:\Reports\AllPrinters.pdf"
```

5.9.3.Contact Report

All Contacts

This report generates the list of all Contacts within the selected scope.

How it works:

The report is generated by querying the Directory Service with the filter

```
"(&(objectCategory=person)(objectClass=contact))"
```

PowerShell Command:

```
Export-ADReport -Type AllContacts -FilePath "C:\Reports\AllContacts.pdf"
```

5.9.4.Policy Report

Password Policy

This report generates the details of the password polices, such as Maximum Password Age, Minimum Password Age, Maximum Password Length, Complexity, and so on, of the selected domain(s).

PowerShell Command:

```
Export-ADReport -Type PasswordPolicy -FilePath "C:\PasswordPolicy.pdf"
```

Account Lockout Policy

This report generates the details of the account lockout polices, such as Lockout Duration, Lockout Threshold, and so on, of the selected domain(s).

PowerShell Command:

```
Export-ADReport -Type AccountLockoutPolicy -FilePath "C:\AccountLockoutPolicy.pdf"
```

6. Scheduling the reports

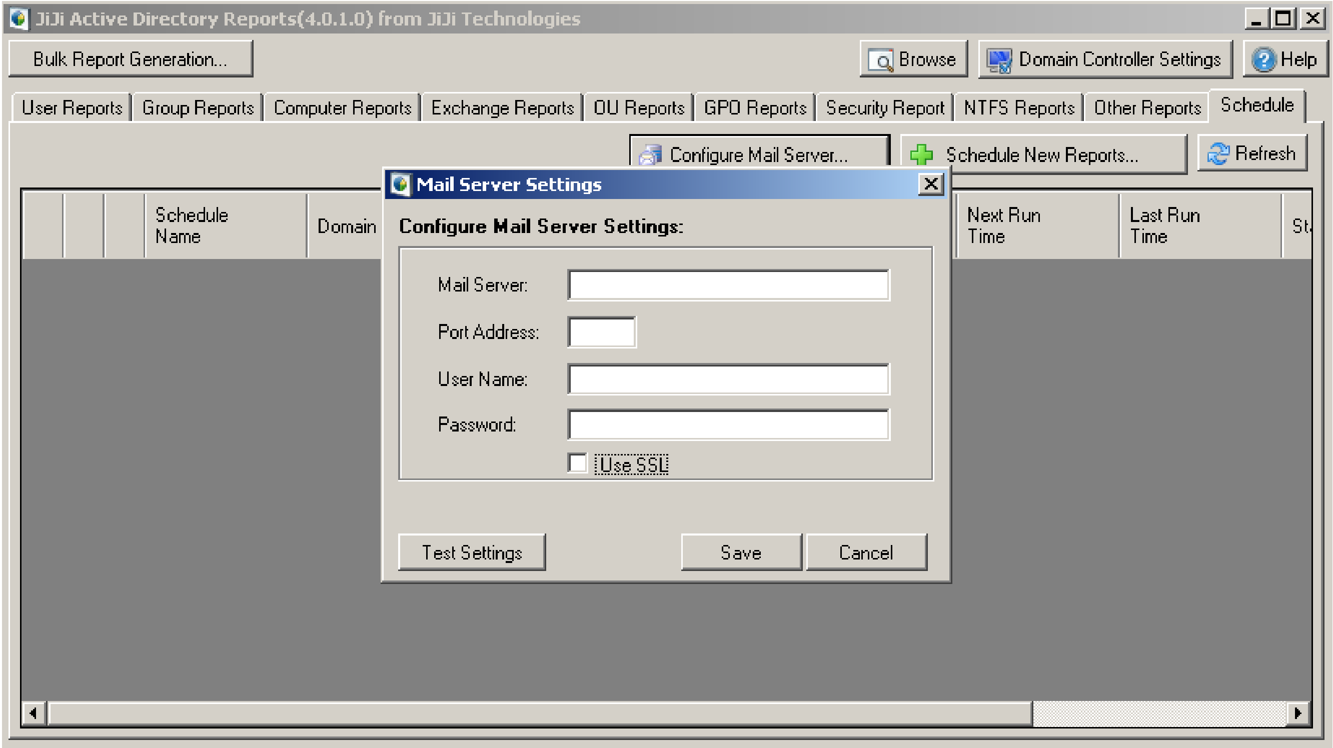
6.1 Before you start

Before scheduling the reports, we need to configure the mail server.

1.Goto **Schedule TAB**.

2.Click **Configure Mail Server button**.

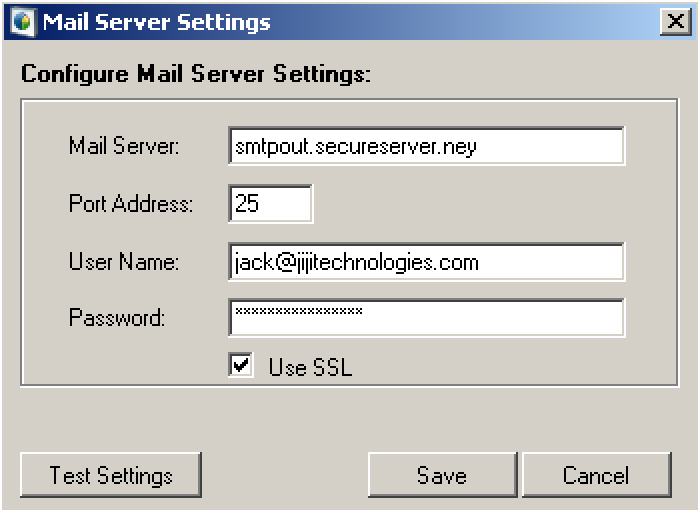
Now the following window is opened



Enter the following details

- SMTP mail server name.
- Port number.
- User name and password for authentication by the server.
- Finally, select Use SSL .

Click Test Setting button to check the connection. Once the testing is completed, status will be display.



Mail Server Settings

Configure Mail Server Settings:

Mail Server: smtpout.secureserver.ney

Port Address: 25

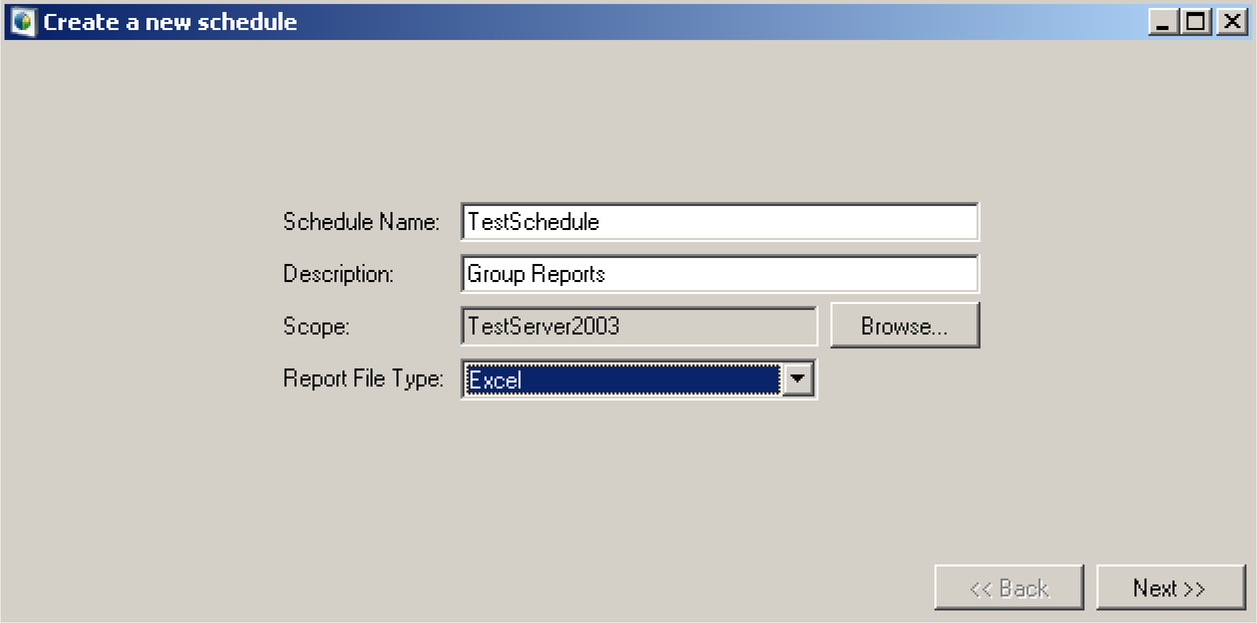
User Name: jack@jijitechnologies.com

Password: ****************

☑ Use SSL

6.2 Steps to schedule the reports

- Goto **Schedule TAB**.
- Click **Schedule New Reports.**



Create a new schedule

Schedule Name: TestSchedule

Description: Group Reports

Scope: TestServer2003

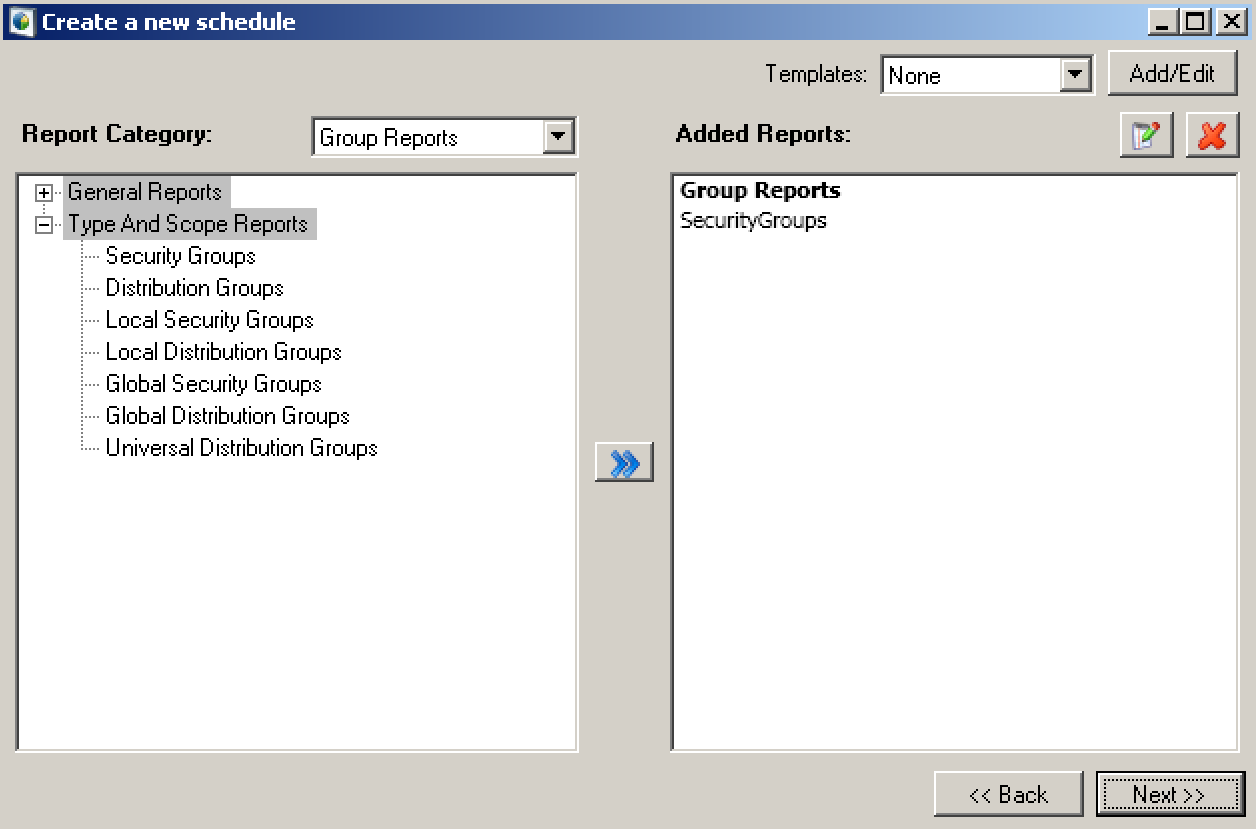
Report File Type: Excel

Now the following window is opened

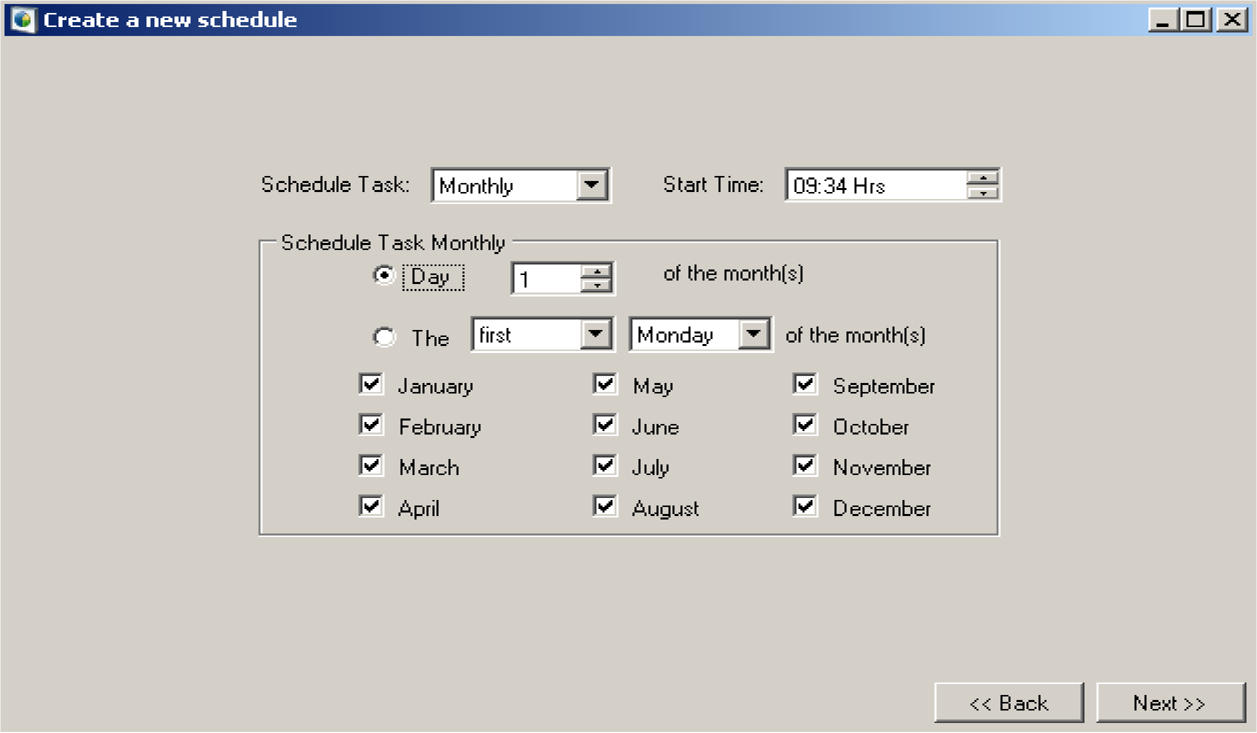
In this page

- Enter the Schedule Name,
- Enter the Description for the schedule,
- Select the Scope and
- Select the reports file type.

Then click Next button. Now the following page is displayed



- Move the required reports from left side panel to right side panel.
- Press **Next** button.



Create a new schedule

Schedule Task: Monthly

Start Time: 09:34 Hrs

Schedule Task Monthly

● Day 1 of the month(s)

○ The first Monday of the month(s)

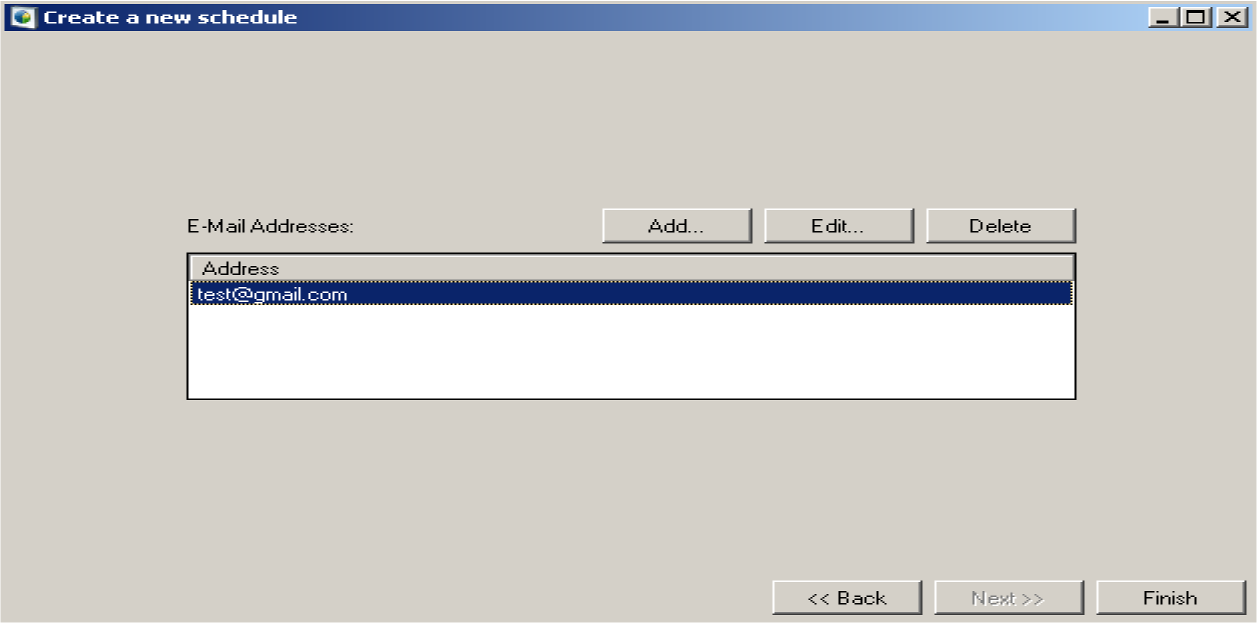
☑ January ☑ February ☑ March ☑ April ☑ May ☑ June ☑ July ☑ August ☑ September ☑ October ☑ November ☑ December

- In the above enter the time details for the schedule.

Note:

User can select **Schedule Task** for Daily, Weekly, Monthly and Once

- Press **Next** button.

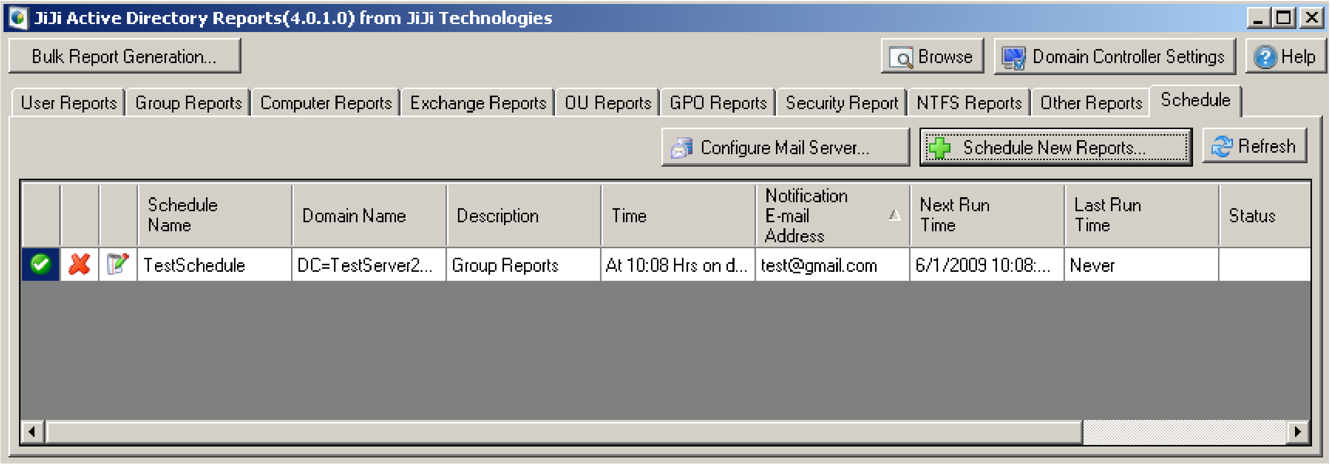


Create a new schedule

E-Mail Addresses:

Address
test@gmail.com

- Enter the recipient E-mail Addresses.
- click **Finish** button. Now the newly created schedule is added to the scheduler list as shown below.



			Schedule Name	Domain Name	Description	Time	Notification E-mail Address	Next Run Time	Last Run Time	Status
✓	✗	✎	TestSchedule	DC=TestServer2...	Group Reports	At 10:08 Hrs on d...	test@gmail.com	6/1/2009 10:08:...	Never	

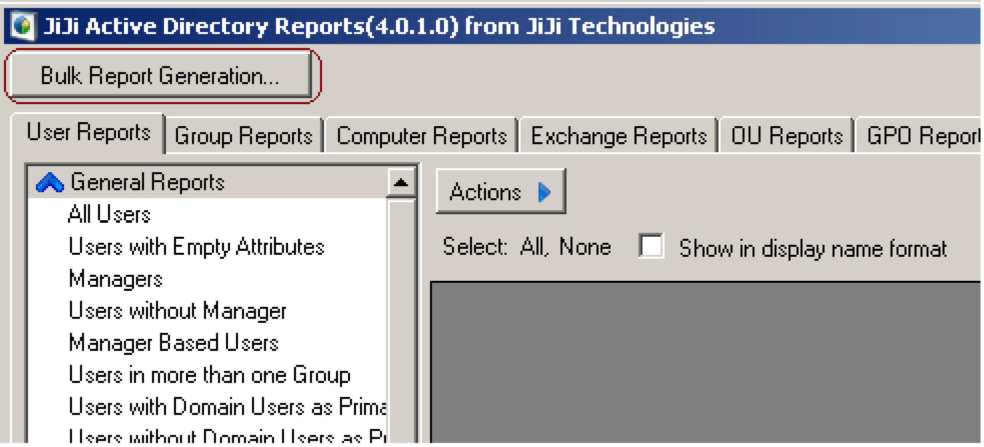
Click ✎ to **Edit the schedule**.

Click ✗to **Delete the schedule**.

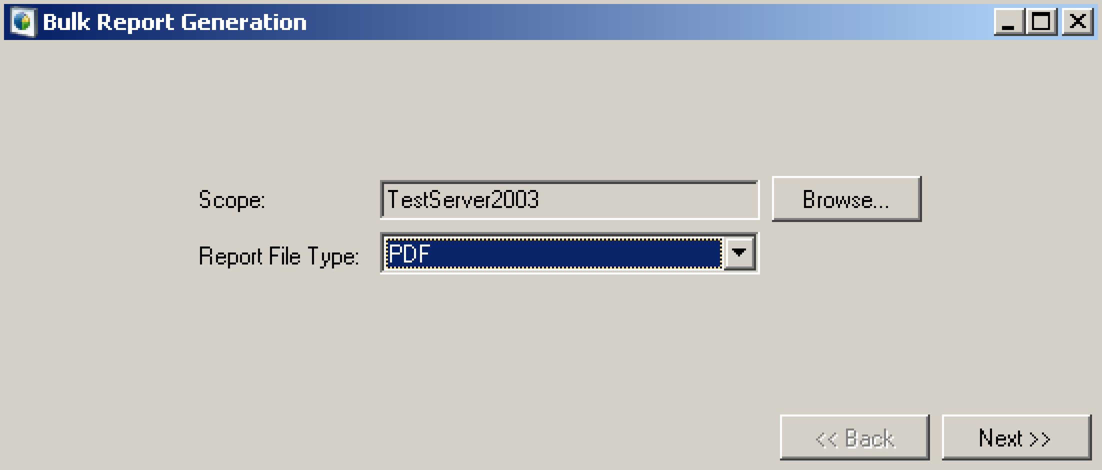
Click ✗✓ to **enable/disable the schedule**.

7. Bulk Report Generation

- Press **Bulk Report Generation** button on the main page.


JiJi Active Directory Reports(4.0.1.0) from JiJi Technologies
Bulk Report Generation...
User Reports
Group Reports
Computer Reports
Exchange Reports
OU Reports
GPO Report
General Reports
All Users
Users with Empty Attributes
Managers
Users without Manager
Manager Based Users
Users in more than one Group
Users with Domain Users as Prima
Users without Domain Users as Pr
Actions
Select: All, None
Show in display name format

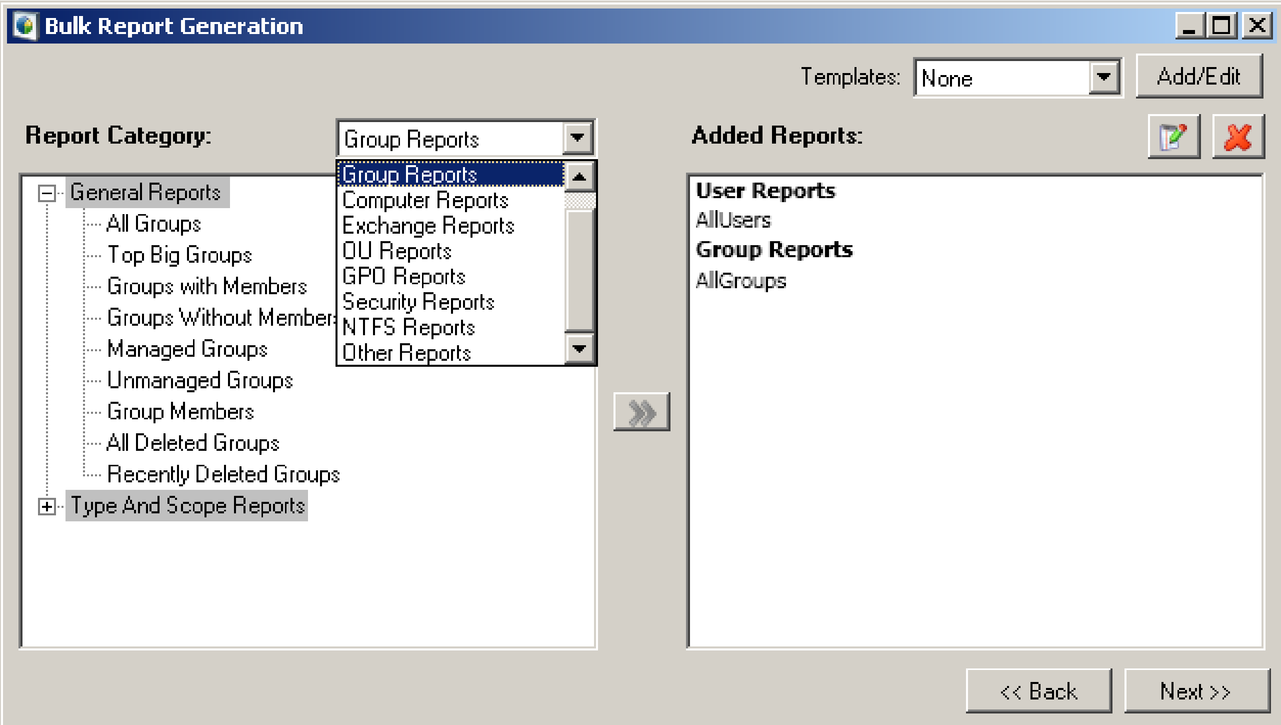
- Select scope to extract information and select report file type.



Scope: TestServer2003

Report File Type: PDF

- Press **Next** button.
- Next page shows the different type of reports. Move the required reports from left panel to right panel.



Templates: None

Report Category: Group Reports

Added Reports:

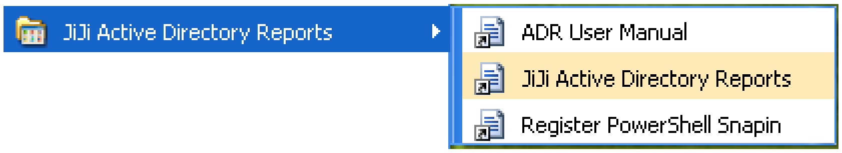
User Reports
AllUsers
Group Reports
AllGroups

- Finally, press **Next** button.
- Enter E-mail Addresses of the recipient or select a folder path to store.
- Press **Finish** button.

8. How to use with PowerShell

Before start using with PowerShell, please do the following steps

Click **Start-> All Programs -> JiJi Active Directory Reports -> Register PowerShell Snapin**


JiJi Active Directory Reports
ADR User Manual
JiJi Active Directory Reports
Register PowerShell Snapin

- Open Windows PowerShell prompt
- Enter Add-PSSnapin ActiveDirectoryReporter command

Now the ActiveDirectoryReporter snapin is registered with PowerShell.

To check whether the ActiveDirectoryReporter snapin is registered with PowerShell use the following command

> Get-PSSnapin -registered

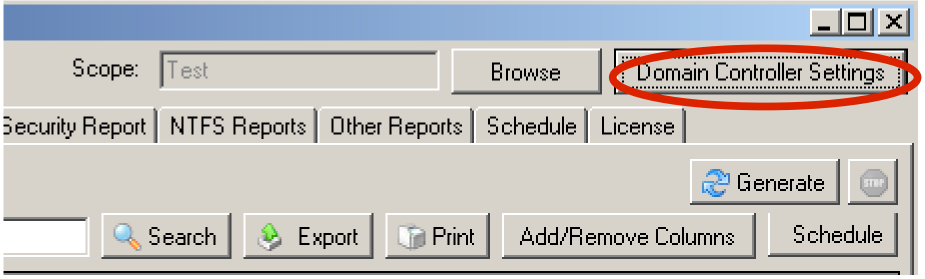
9. Domain Controller Settings

By using **Domain Controller Settings**, the users have the following options

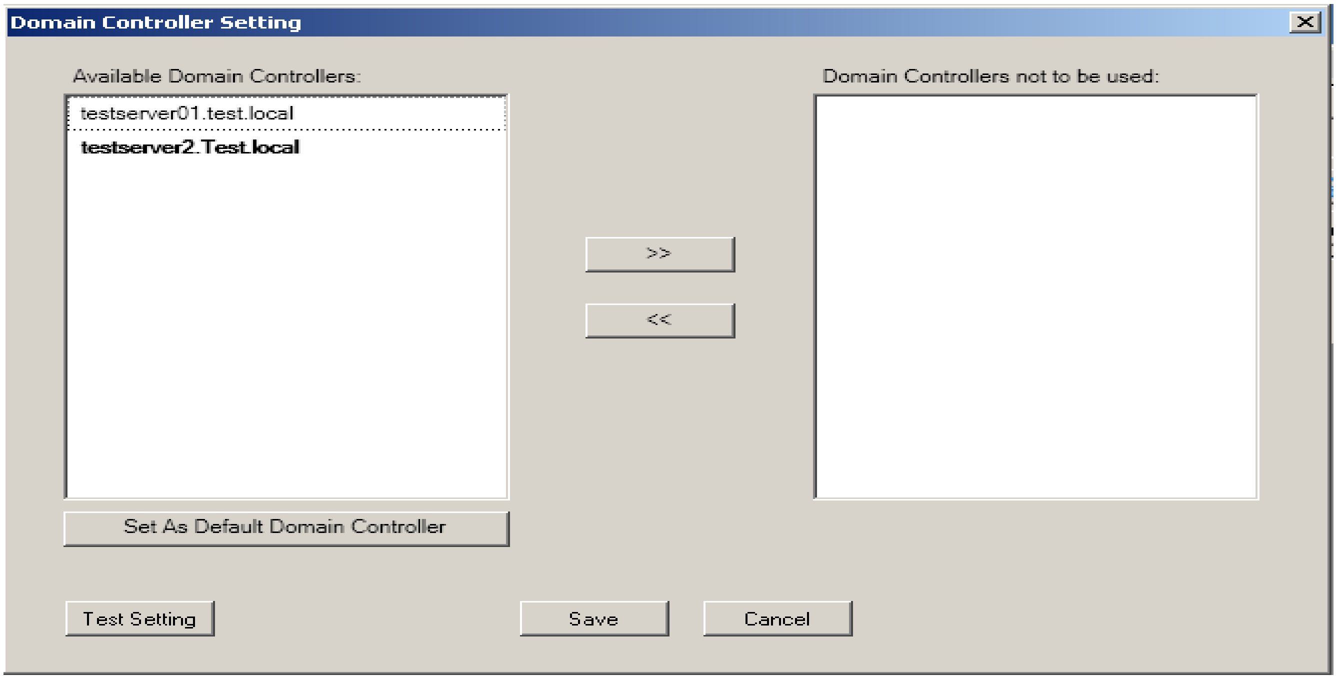
- Default Domain Controller
- List of domain domain controllers that are not to be used

  By using the above option, user can isolate the far away domain controllers or faulty domain controllers from the report generation.

To make the domain controller settings, click **Domain Controller Setting** button on the main page as shown below.



Then the following dialog will open


Domain Controller Setting
Available Domain Controllers:
Domain Controllers not to be used:
>>
<<
Set As Default Domain Controller
Test Setting
Save
Cancel

testserver01.test.local
testserver2.Test.local

The domain controller which is marked in **bold**, is the default domain controller. You can change the default controller by using **Set As Default Domain Controller**.

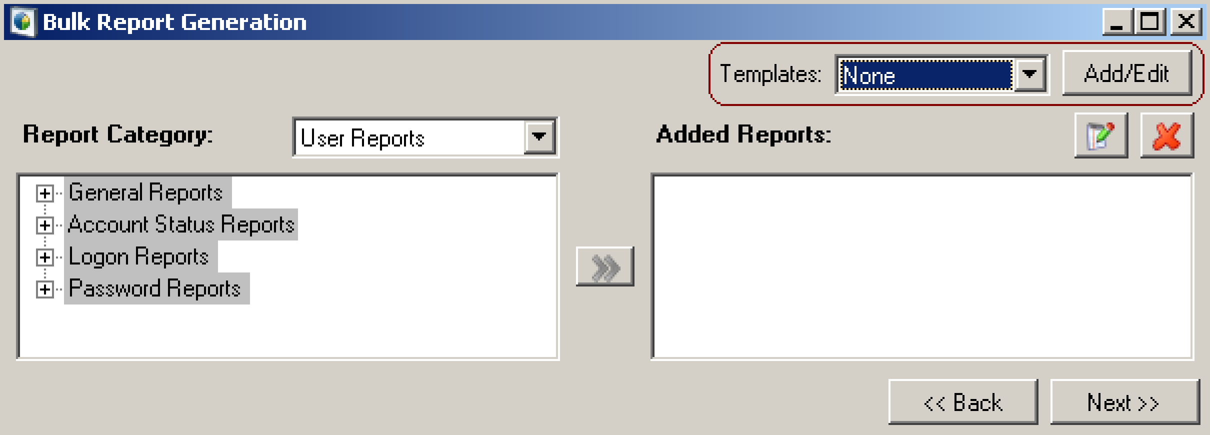
If you don't want to use a particular domain controller, you can move it to **Domain Controllers not to be used** list by using **>>** button.

You can test whether the default domain controller is accessible by using **Test Setting** button.

10. How to Use Templates?

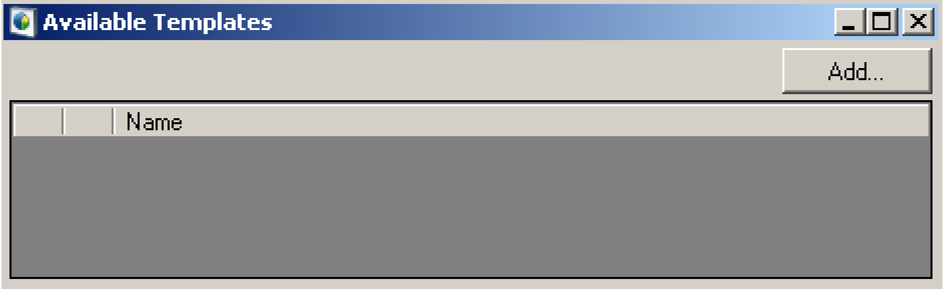
Template is a sub-feature added to Bulk report generation and scheduler. It helps to store the set of selected report in memory and provides option to reuse the stored report types.

- Press **Bulk Report Generation** button

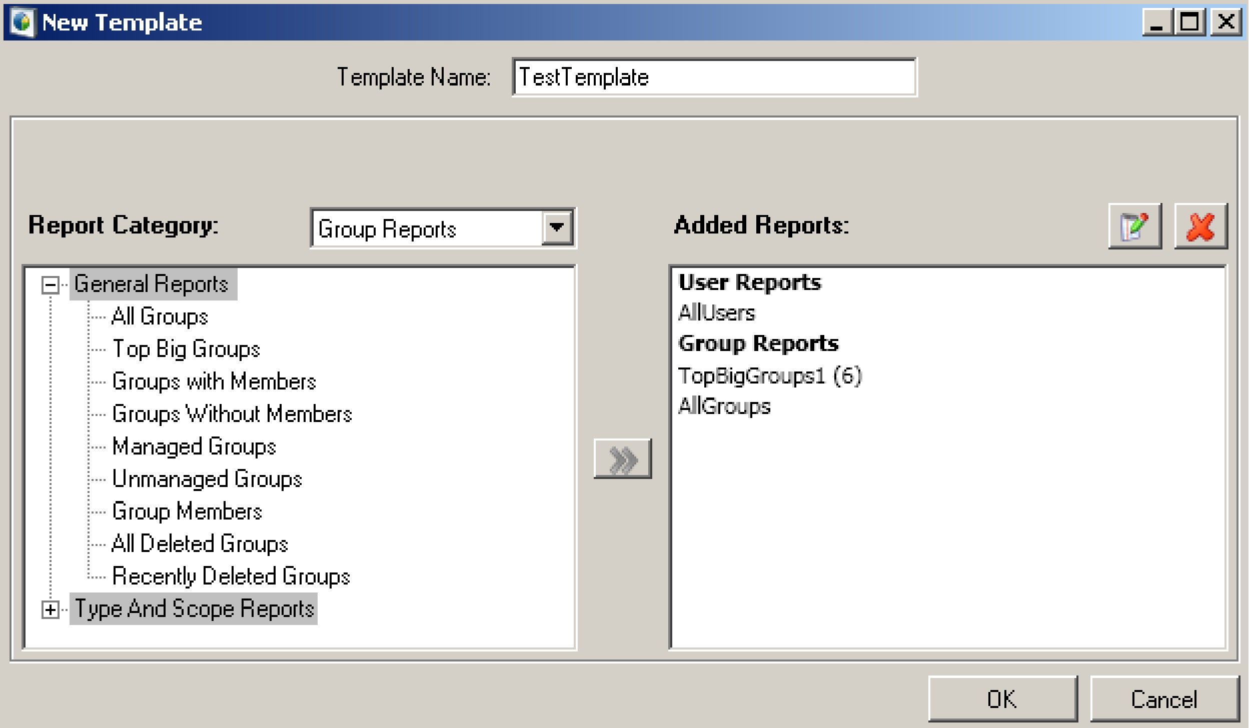


Templates: None
Report Category: User Reports

- Press **Add/Edit** button.



- Press **Add** button.
- Enter the template name.
- Move the Required report type from left side panel to right side panel.
- Finally, press **OK** button.



Template Name: TestTemplate

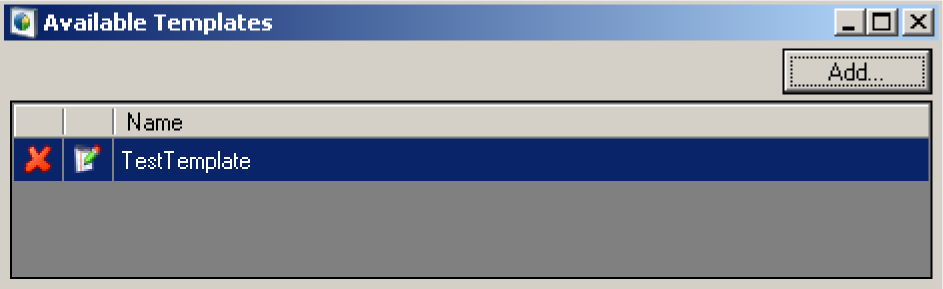
Report Category: Group Reports

- General Reports
  - All Groups
  - Top Big Groups
  - Groups with Members
  - Groups Without Members
  - Managed Groups
  - Unmanaged Groups
  - Group Members
  - All Deleted Groups
  - Recently Deleted Groups
- Type And Scope Reports

Added Reports:

User Reports
AllUsers
Group Reports
TopBigGroups1 (6)
AllGroups

- Now, you can view your templates as a list.

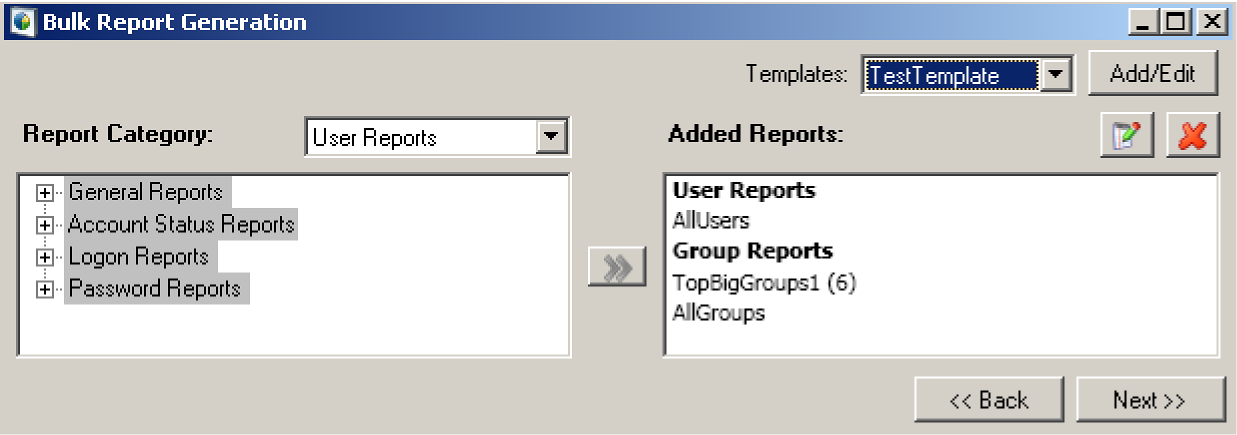


		Name
✗	✎	TestTemplate

Click ✗ to **Delete the schedule**.

Click 📝 to **Edit the schedule**.

- Now **Templates List Box** lists all the available templates. Select the required template



Templates: TestTemplate

Report Category: User Reports

- General Reports
- Account Status Reports
- Logon Reports
- Password Reports

Added Reports:

User Reports
AllUsers
Group Reports
TopBigGroups1 (6)
AllGroups