Saviynt – Exchange Integration & Design Document

February 2023

Table of Contents

[1 Document Control 3](#_Toc126848280)

[2 Introduction 4](#_Toc126848281)

[2.1 Scope 4](#_Toc126848282)

[3 Pre-requisite 5](#_Toc126848283)

[3.1 Windows Connector Server Setup Pre-requisite 5](#_Toc126848284)

[3.2 Deploy SaviyntApp on Windows Server 5](#_Toc126848285)

[4 Architecture 7](#_Toc126848286)

[4.1 Application Connector 7](#_Toc126848287)

[5 Application Connector Design 8](#_Toc126848288)

[5.1 Configuration 8](#_Toc126848289)

[5.1.1 Connection 8](#_Toc126848290)

[5.1.2 Security System 8](#_Toc126848291)

[5.1.3 Endpoint 9](#_Toc126848292)

[5.2 Create Operation 10](#_Toc126848293)

[5.2.1 Connection Configuration 10](#_Toc126848294)

[5.2.2 PowerShell 11](#_Toc126848295)

[5.3 Disable 11](#_Toc126848296)

[5.3.1 Connection Configuration 12](#_Toc126848297)

[5.3.2 PowerShell 12](#_Toc126848298)

[5.4 Rehire 12](#_Toc126848299)

[5.4.1 Connection Configuration 13](#_Toc126848300)

[5.4.2 PowerShell 14](#_Toc126848301)

[5.5 Delete 14](#_Toc126848302)

[5.6 Reconciliation 14](#_Toc126848303)

[5.6.1 Connection Configuration 16](#_Toc126848304)

[5.6.2 Scheduled Task 17](#_Toc126848305)

[6 Rules 18](#_Toc126848306)

[6.1 Technical Rules 18](#_Toc126848307)

# Document Control

|  |  |  |  |
| --- | --- | --- | --- |
| **Version control** | | | |
| **Version** | **Name** | **Role** | **Date** |
| V1 | Gaurav Khandelwal | Author | 01/31/2023 |
| V2 | Vineet Gupta | Author | 02/03/2023 |

# Introduction

This design document covers integration design and approach of application 2016 with Saviynt. It will cover the pre-requisite to setup windows server on which PowerShell script will be executed and various account lifecycle management operations.

## Scope

Scope of this design document

* Pre-requisite
* Architecture
* Connectors
* Account Operations
* Create
* Disable
* Rehire
* Reconciliation

# Pre-requisite

## Windows Connector Server Setup Pre-requisite

1. Windows Connector Server OS – 2012/16 R2.
2. IIS with .NET 4.5 Framework inbuilt support
3. Network Connectivity between Application Server & Windows Connector Server, so that Remote PowerShell commands can be executed from Windows Server to create a mailbox.
4. Network Connectivity between Saviynt Cloud Application & Windows Server - For Windows Connector Test environment, you can enable both Port 80/443 on and Self-Signed certificate are used for SSL access. For Windows Connector Production environment, 3rd Party Valid certificates are required.
5. Service Account credentials with permissions to execute Windows PowerShell commands on the application Server. This service account should also be granted Admin rights on Windows Connector Server.

## Deploy SaviyntApp on Windows Server

Setup the server using all the pre-requisites covered in the preceding section.

1. Open Windows PowerShell in “Run as administrator” mode and execute the following command.

Set-ExecutionPolicyRemoteSigned

1. Login to Windows server with Active Directory Service Account
2. Use the following link to download the Saviynt App (or “Saviynt PowerShell Agent”) to Window Server system.

|  |  |
| --- | --- |
| Saviynt PowerShell Agent | |
| URL | <https://SAVIYNTAWSURL/Saviynt-Powershell/Saviynt-PowershellAgent.zip> |

Note: SAVIYNTAWSURL should be provided by the Saviynt support team

1. Open IIS Manager and perform the following steps:

**Note**: If this setup is not carried on the machine used for ADSI agent, then install the IIS server before proceeding further. Refer Win-PS connector documentation on Freshdesk for detailed step: <https://saviynt.freshdesk.com/support/solutions/articles/43000514996-win-ps-connector-guide#Win-PSConnectorGuide-InstallationofWindowsInternetInformationServicesServer>

* 1. Create a new application pool with following configurations:
     1. Name: SaviyntAppPool
     2. .NET CLR Versions: .Net CLR version v4.0
     3. Managed pipeline mode: Integrated
     4. Start application pool immediately – Checked
  2. Select the application pool created i.e. **SaviyntAppPool** and go to “**Advance Settings**”. Within “Advance Settings”, under “**Process Model**” settings, set the following values:
     1. **Load User Profile**: “True
     2. **Identity**: “Select Custom and Enter User Name & password of user who has Administrator access on Windows Connector Server”.
  3. Create a new application under Default Web Site with following configurations:
     1. **Alias**: Saviynt
     2. **Application pool**: Select “SaviyntAppPool” Application pool
     3. **Physical Path**: Path to location of unzipped content of the agent i.e. C:\SaviyntWinCon
     4. Restart the IIS Server after performing the preceding steps.

# Architecture

The high-level architecture includes Saviynt Rest API connector for application integration. This connector will support provisioning as wells as reconciliation.

## Application Connector

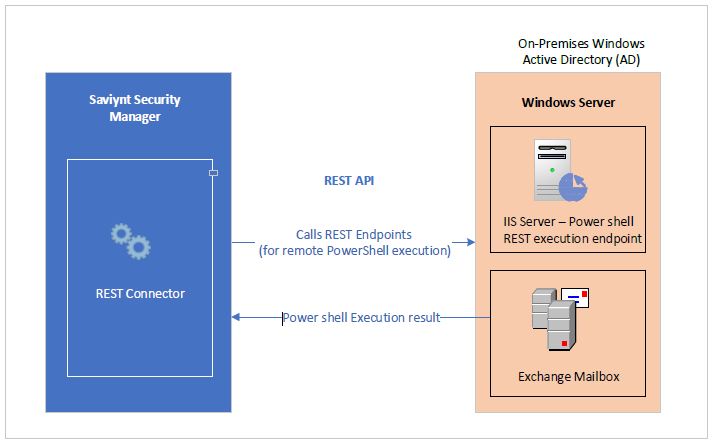


Figure 1 Application Integration Architecture

# Application Connector Design

Saviynt REST connector will be configured to perform provisioning and reconciliation operations. The windows connector server setup for application will enable Saviynt to invoke REST endpoint to execute PowerShell scripts.

Saviynt REST connector link: [REST Connector Guide : Customer Portal (freshdesk.com)](https://saviynt.freshdesk.com/support/solutions/articles/43000521736-rest-connector-guide)

## Configuration

### Connection

Create a connection to reconcile Application with name AD Production.

**Type of the connector**: REST

General connection parameters:

|  |  |
| --- | --- |
| Attribute Name | Attribute Value |
| Connection Name | AD Production |
| Connection Type | REST |
| ConnectionJSON | Specify the json to authenticate the .NET API |

Similarly, create and configure connection for the application with following name.

**Connection Name:** AD Production Endpoint

### Security System

Create a security system with name AD Production for integrating application with Saviynt.

The security system will bind the connection, approval workflow (if any) together.

General configuration parameter for security systems:

|  |  |
| --- | --- |
| Attribute Name | Attribute Value |
| System Name | AD Production |
| Display Name | AD Production |
| Connections | AD Production |
| Provisioning Connection | AD Production |
| Access Add Workflow | NA |
| Access Remove Workflow | NA |
| Number of tries for provisioning | 1 |
| Default System | No |
| Automated Provisioning | Enable |
| Manage Entity | No |
| Persistent Data | No |
| Use open connector | No |
| Recon Application | Yes |
| Instant Provisioning | Enable |

Similarly, create and configure security system for Application with following name.

**Security System Name:** AD Production Path

### Endpoint

Create Endpoint with name ‘AD Production Endpoint’ and attached the security system to it. Endpoint represents the target application under the Security System.

General configuration parameter for security systems:

|  |  |
| --- | --- |
| Attribute Name | Attribute Value |
| Endpoint Name | AD Production |
| Display Name | AD Production |
| Security System | AD Production |
| Add Email Template | Provide any email templates for specific provisioning operation if any |
| Description | Application Provisioning and Reconciliation |
| User Account Correlation rule | Lanid of user profile to SamAccountName |

Along with the above attributes, provide additional information like owners etc.

Similarly, create and configure endpoint for application with following name.

**Endpoint Name:** AD Production endpoint

## Create Operation

When a new user is created in Saviynt from the HR Services trusted reconciliation, a configured technical rule will be evaluated to trigger Active Directory account provisioning. Application provisioning will be triggered after the user is successfully provisioned to AD.



**Description:**

The following actions take place when a new Application Account is created in Saviynt

* + 1. Once the AD account is provisioned, Saviynt will trigger the application account provisioning.
    2. For all the users having AD account, Application account will be provisioned. The exception is for users with job code X. These users will be excluded from application provisioning.
    3. For and the company users, respective application endpoint will be provisioned. The “Legacy Path” of HR Services will be used to identify the users origin.
    4. Saviynt will invoke the PowerShell script for mailbox creation.
    5. Application pre-defined policies will generate and populate attributes for the application.

### Connection Configuration

|  |  |
| --- | --- |
| Attribute Name | Attribute Value |
| Connection Name | Core AD Prod Endpoint  Application AD Prod Endpoint |
| ConnectionJSON | Provide the Json to connect to the windows connector server setup in the [Section 2](#_Pre-requisite) |
| CreateAccountJSON | Json to invoke the rest API to execute the PowerShell script with input parameters as   * + 1. SamAccountName     2. Domain     3. Environment switch |

### PowerShell

The KPMG team will provide the PowerShell script for the following:

* + - 1. Get input variables passed by Saviynt as stated int the above section
      2. Import required modules if any
      3. Initialize logger
      4. Create a PSSession
      5. Commands to create the application on-prem mailbox.

## Disable

When a user is disabled in Saviynt by termination process in HR System, Saviynt will immediately disable the application Mailbox. The steps involved in disabling the application Mailbox are depicted in the workflow diagram.



**Description:**

The following actions take place to disable the application Account from Saviynt

* + 1. User is terminated in HR systems
    2. Saviynt will set users status to inactive and update the “Disabled Date” on user’s Profile
    3. Update rules will be triggered to disable the application account. Application account will be disabled irrespective of the legal hold.
    4. Saviynt will disable the mailbox by invoking the PowerShell script which will set attribute hiddenFromAddressListEnabled to true
    5. After X days from the application account disablement, no action will be taken by Saviynt.

### Connection Configuration

|  |  |
| --- | --- |
| Attribute Name | Attribute Value |
| Connection Name | Prod Endpoint |
| DisableAccountJSON | Json to invoke the rest API to execute the PowerShell script with input parameters as   * + 1. SamAccountName     2. Domain     3. Disable switch     4. Environment switch |

### PowerShell

The KPMG team will provide the PowerShell script for the following:

* + - 1. Get input variables passed by Saviynt as stated int the above section
      2. Import required modules if any
      3. Initialize logger
      4. Create a PSSession
      5. Commands to set hiddenFromAddressListEnabled attribute of the application on-prem mailbox to true.

## Rehire

When a user is enabled in Saviynt by re-hire process in HR service, Saviynt will immediately enable the application Mailbox if rehired within X days. Saviynt will provision application mailbox if rehired after X days. The steps involved in enabling or provisioning application Mailbox on rehire are depicted in the workflow diagram.



**Description:**

The following actions take place on rehire for application Account from Saviynt

* + 1. User is rehired in HR Service
    2. Saviynt will set users status to active.
    3. If user is rehired within x day
    4. Update rules will be triggered to enable the application account.
    5. Saviynt will enable the mailbox by invoking the PowerShell script which will set attribute hiddenFromAddressListEnabled to false
    6. If user is rehired after X days, birthright rules will trigger the application account provisioning after AD account is provisioned.

Saviynt will create the mailbox by invoking the PowerShell script. Refer [Section Create Operation](#_Create_Operation)

* + 1. Application pre-defined policies will generate and populate attributes for the application.

### Connection Configuration

|  |  |
| --- | --- |
| Attribute Name | Attribute Value |
| Connection Name | Prod Endpoint |
| EnableAccountJSON | Json to invoke the rest API to execute the powershell script with input parameters as   * + 1. SamAccountName     2. Domain     3. Enable switch     4. Environment switch |

### PowerShell

The KPMG team will provide the PowerShell script for the following:

* + - 1. Get input variables passed by Saviynt as stated int the above section
      2. Import required modules if any
      3. Initialize logger
      4. Create a PSSession
      5. Commands to set hiddenFromAddressListEnabled attribute of application on-prem mailbox to false.

## Delete

Saviynt will not take any action for deletion. The deletion of mailbox will be based on the policies defined in the application for on-prem mailbox.

Hence legal hold will not be accountable for application de-provisioning operations.

## Reconciliation

Account reconciliation will be configured for application mailbox. This process will make Saviynt aware of the existing application account.

The reconciliation flow if depicted in the below diagram.



**Description:**

The following actions take place when a new application Account is created in Saviynt

* + 1. The same REST connection will be used to reconcile account into Saviynt.
    2. Saviynt will execute the PowerShell command configured in the ImportAccountEntitlementJSON on the windows .Net server. It will fetch all the on-prem mailboxes.
    3. The data returned from this command will mapped provided in to the json.
    4. The status of the account will be based on value of attribute HiddenFromAddressListsEnabled.

The minimum attributes to be reconciled are as follows:

|  |  |  |
| --- | --- | --- |
| Application Attribute | Saviynt Account Attribute | Saviynt Label |
| SamAccountName | name | Name |
| WhenMailboxCreated | created\_on | Created On |
| DisplayName | displayname | Display Name |
| SamAccountName | accountID | Account ID |
| RecipientType | accounttype | Account Type |
| Name | description | Description |
| DistinguishedName | comments | Comments |
| UserPrincipalName | customproperty1 | User Principal Name |
| AddressListMembership | customproperty2 | Address List Membership |
| RecipientTypeDetails | customproperty3 | Recipient Type Details |
| WindowsEmailAddress | customproperty4 | Windows Email Address |
| GrantSendOnBehalfTo | Customproperty5 | Grant Send On Behalf To |
| Alias | customproperty6 | Alias |
| Guid | customproperty7 | Guid |
| Application Guid | customproperty8 | Application Guid |
| WhenMailboxCreated | customproperty9 | When Mailbox Created |
| EmailAddressPolicyEnabled | customproperty10 | Email Address Policy Enabled |
| PrimarySmtpAddress | customproperty11 | Primary Smtp Address |
| HiddenFromAddressListsEnabled | customproperty12 | Hidden From Address Lists Enabled |

### Connection Configuration

|  |  |
| --- | --- |
| Attribute Name | Attribute Value |
| Connection Name | Prod Endpoint |
| ImportAccountEntJSON | Powershell script that will be passed in the httpParams of the rest call  Sample Format:   "httpParams": {  "Script": "\\$pass=convertto-securestring '${connection.password}' -asplaintext -force; \\$mycred=new-object -typename System.Management.Automation.PSCredential -argumentlist '${connection.username}',\\$pass; \\$Session=New-PSSession -ConfigurationName Microsoft.application -ConnectionUri http://ENVIRONMENTURL/powershell/ -Authentication Kerberos -Credential \\$mycred; \\$ses=Import-PSSession \\$Session -DisableNameChecking -AllowClobber; Get-Mailbox -ResultSize unlimited |Select ApplicationGuid,SamAccountName,UserPrincipalName,WhenMailboxCreated,AddressListMembership,Alias,DisplayName,PrimarySmtpAddress,RecipientType,RecipientTypeDetails,WindowsEmailAddress,GrantSendOnBehalfTo,Name,DistinguishedName,Guid,EmailAddressPolicyEnabled,HiddenFromAddressListsEnabled; Remove-PSSession \* -ErrorAction SilentlyContinue"  }  The mapping of the attributes will be provided in the following attribute of the rest call  "colsToPropsMap": {  "name": "SamAccountName~#~char",  "created\_on": "WhenMailboxCreated~#~datetime",  "displayname": "DisplayName~#~char",  "accountID": "UserPrincipalName~#~char",  "accounttype": "RecipientType~#~char",  "description": "Name~#~char",  "comments": "DistinguishedName~#~char",  "customproperty1": "UserPrincipalName~#~char",  ……  } |
| Status\_Threashold\_Config | Used to specify the expression to set the status of account.  Status of the account will be set as active of inactive based on HiddenFromAddressListsEnabled. |

### Scheduled Task

Configure job to execute target reconciliation for both connection Core AD Prod Endpoint and Application AD Prod Endpoint

# Rules

## Technical Rules

Technical rules can be configured to provide Birthright access to the application based on specified criteria.

|  |  |
| --- | --- |
| Rule | Description |
| Company Birthright Provisioning for the Application | This rule will provision the application on-prem account to the users. The rule will be triggered on AD provisioning |
| Application Birthright application Provisioning | This rule will provision the application mailbox to all application colleagues satisfying any particular job role conditions but only after application AD Account is created |