Documentation - Secret Server Integrations - 1.0.0

Table of Contents

Secret Server Integrations

Integrations are supported to the extent of the third-party product procedures documented for this integration. Please contact the third-party for any customized setup of the integrated product.

Secret Server integration documents are moving to Thycotic's new documentation platform. Here you find verified third-party product integrations. This area will be growing while existing contents is moved over and more integrations are verified.

Access each integrated product folder to learn more about the integration details.

Support for ThycoticCentrify Integrations

Prerequisites: As a prerequisite of support for any ThycoticCentrify integrations the customer must have sufficient access to the ThycoticCentrify product and all parts of the third-party integration and must be able to provide ThycoticCentrify with requested information and meetings to examine in order to progress reported issues.

ThycoticCentrify In Product Integrations

Integrations that are directly built into paid ThycoticCentrify products will be supported by the ThycoticCentrify support team and defects will be handled by the ThycoticCentrify product developers who maintain the ThycoticCentrify product where the issue occurs.

ThycoticCentrify In Product Customization

Many ThycoticCentrify products can be customized in order to achieve an integration between the ThycoticCentrify product and third-party systems. If ThycoticCentrify documents an integration as a supported integration, the integration will be configured as specified in our documentation and is verified at the time of their creation by ThycoticCentrify to ensure that they work as designed.

> **Note:** Assistance with design, configuration, or troubleshooting of customization designed to interact with third-party systems is not within the scope of what the ThycoticCentrify Support organization can provide at this time. ThycoticCentrify does not guarantee that every configuration of third-party systems will work with in product customizations. Assistance with design, configuration or troubleshooting for customization of ThycoticCentrify products can be worked on as part of a paid engagement with the Professional Services team.

ThycoticCentrify Created Unpaid Integrations

Unpaid integrations created by ThycoticCentrify are code or applications that are not sold by Thycotic for monetary compensation. They are provided for the use of ThycoticCentrify customers and in some cases are available to the public.

An example of this type of integration would be the RabbitMQ Helper, migration tools created by ThycoticCentrify, and code provided on the Thycotic Github.These integrations were verified at the time of their creation by ThycoticCentrify to ensure that they work as designed.

> **Note:** Assistance with configuration or troubleshooting of these tools with third-party systems is not within the scope of what the ThycoticCentrify Support organization can provide at this time. ThycoticCentrify does not guarantee that every configuration of third-party systems will work with 3rd-Party Integrations. Assistance with the use of these tools, configuration, or troubleshooting for customization of ThycoticCentrify products can be worked on as part of a paid engagement with the Professional Services team.

Third Party Integrations to ThycoticCentrify

This category of integration encompasses any code or script which integrates with ThycoticCentrify usually by **API** that is written by a third-party vendor. Thycotic does not guarantee that third-party code is written correctly or that it respects ThycoticCentrify product limitations.

For instance, the third-party code may fail to respect Token expiry or issue calls too quickly without waiting for responses and time-outs. Third-party integrations are supported by verifying that the ThycoticCentrify application is functioning correctly. ThycoticCentrify does not support, code or maintain third-party code or scripts.

For commercially sold third-party products which have vendor support, ThycoticCentrify may elect to attend calls. The third-party product must be able to provide a knowledgeable resource and share specifics about how they integrate with the ThycoticCentrify application. **The goal of such calls would be to advise the third-party vendor about what they need to change to better integrate with ThycoticCentrify products**.

Professional Services Integrations

Code or scripts written for or provided to customers as part of Professional Services are not included in the definitions above. Please refer to the terms of the warranty on your Professional Services engagement.

Automation Anywhere Integration with Secret Server

Automation Anywhere's platform employs software robots, or bots, that make business processes self-running. The application uses a combination of traditional RPA and cognitive elements such as unstructured data processing and natural language understanding to learn and observe human behavior.

The integration between Secret Server and Automation Anywhere ensures that:

- Passwords are securely vaulted in Secret Server
- All access by bots is captures in Secret Server Audit Trail
- Role-based access controls limit which passwords each bot can use

Integration Requirements

System Requirements

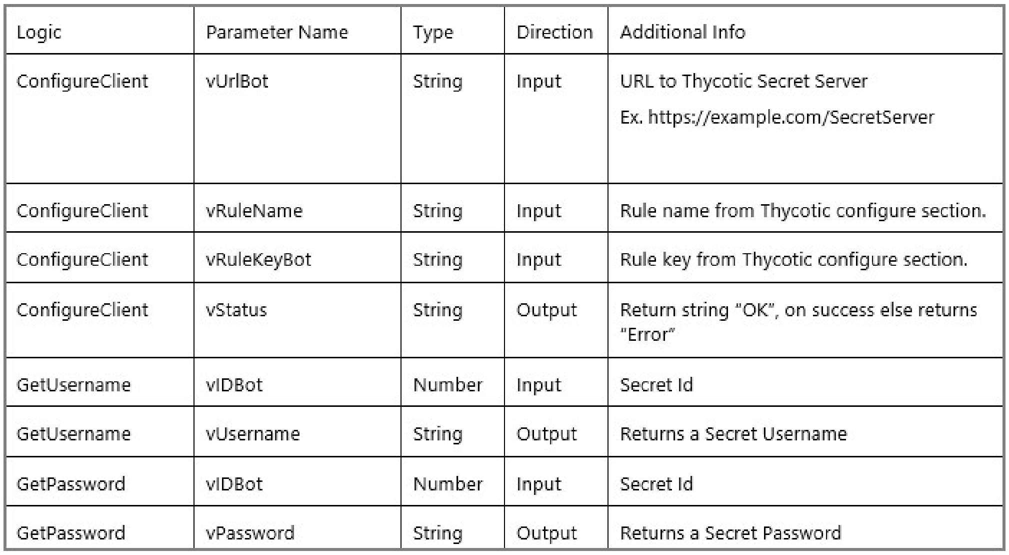
Product	Details	
Automation Anywhere		
	Version	Enterprise v.11.3 or later.
	Hardware	For further information about hardware requirements please contact Automation Anywhere.
Secret Server		
	Version	Thycotic Secret Server 10.8 or later.
	Hardware	For further information about system requirements for Secret Server please visit: https://docs.thycotic.com/ss/10.8.0/secret-server-setup/system-requirements/index.md.

Implementation Considerations

The provided .mbot file contains 3 logic assets that can be called form a Task Solution in Automation Anywhere.

- ConfigureClient - connects to the secret server, obtains an authorization token and fills in the secret structure.
- GetUsername - returns username field from structure for the secret Id.
- GetPassword – returns password field from structure for the secret Id.

Below is a list of available Inputs and Outputs:



Logic	Parameter Name	Type	Direction	Additional Info
ConfigureClient	vUrlBot	String	Input	URL to Thycotic Secret Server Ex. https://example.com/SecretServer
ConfigureClient	vRuleName	String	Input	Rule name from Thycotic configure section.
ConfigureClient	vRuleKeyBot	String	Input	Rule key from Thycotic configure section.
ConfigureClient	vStatus	String	Output	Return string "OK", on success else returns "Error"
GetUsername	vIDBot	Number	Input	Secret Id
GetUsername	vUsername	String	Output	Returns a Secret Username
GetPassword	vIDBot	Number	Input	Secret Id
GetPassword	vPassword	String	Output	Returns a Secret Password

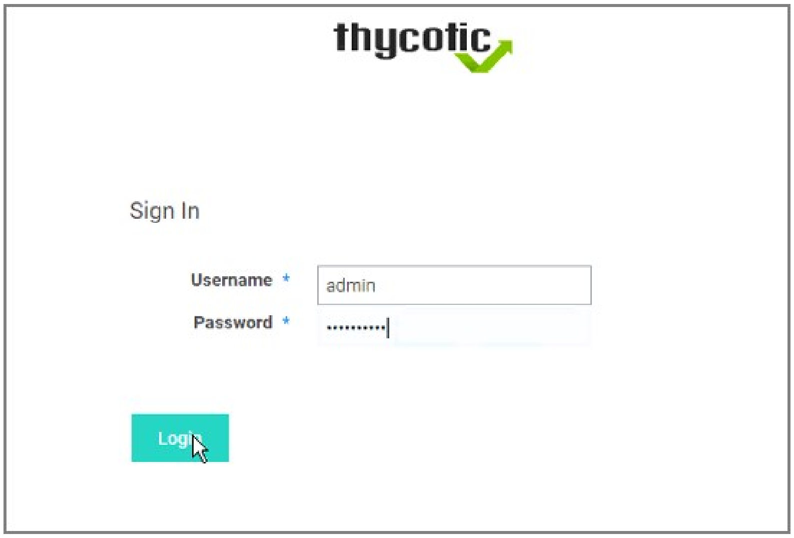
Configuration

- Configuration Steps for Secret Server
- Setup a new MetaBot in Automation Anywhere
- Retrieving a Secret UserID and Password

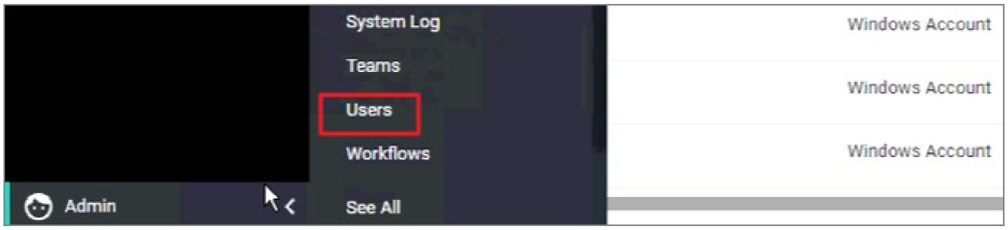
Configuration Steps for Secret Server

Create an Application Account

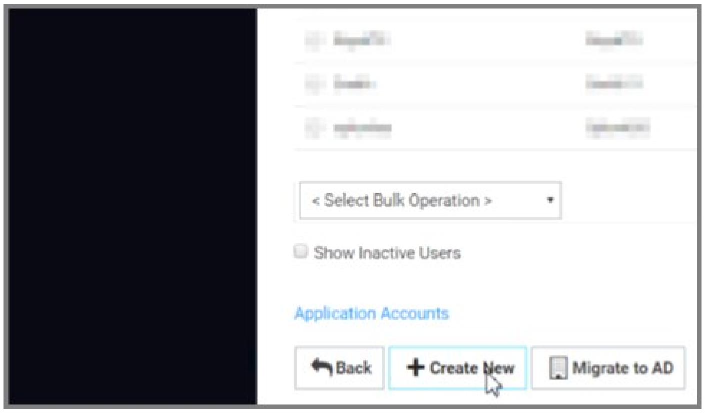
1. Navigate to Secret Server and login.



2. Click **Admin I Users**.



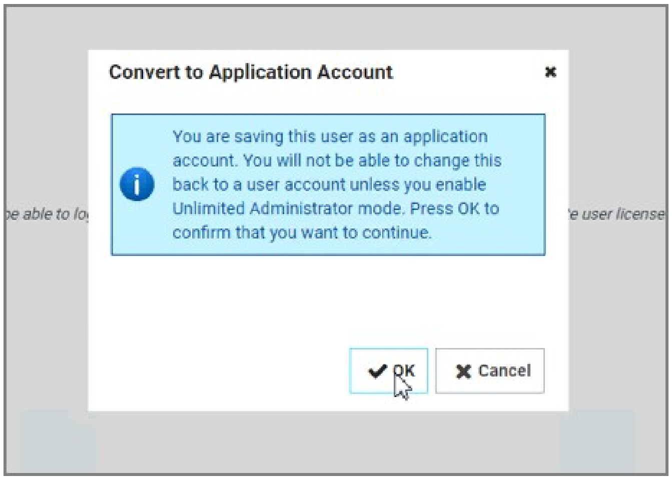
3. Click the **Create New** button at the bottom of the screen.



4. Enter in a **User Name, Email Address and Password**.
5. Click the **Advanced** option.
6. Check the box for **Application Account**.

7. Click **Save**.

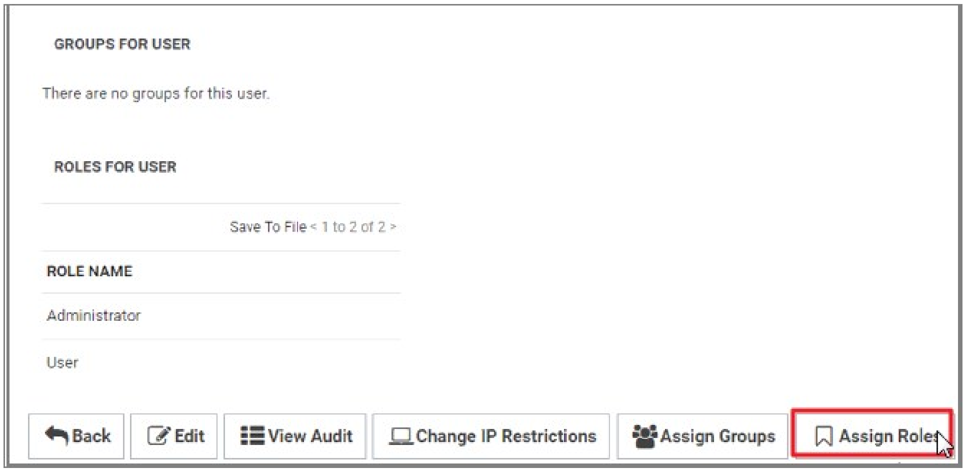
8. Click **Ok** on the next screen to convert the user to an Application Account.



Convert to Application Account

You are saving this user as an application account. You will not be able to change this back to a user account unless you enable Unlimited Administrator mode. Press OK to confirm that you want to continue.

9. Click on **Assign Roles**.



GROUPS FOR USER

There are no groups for this user.

ROLES FOR USER

Save To File < 1 to 2 of 2 >

ROLE NAME
Administrator
User

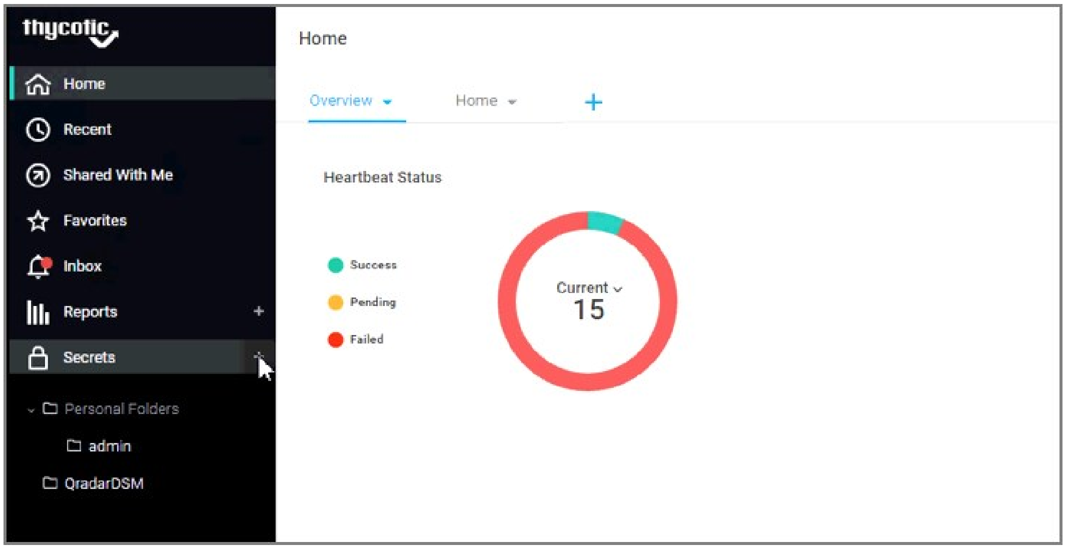
10. Assign the **User** role.

11. Click **Save Changes**.

Create a Secret for Automation Anywhere

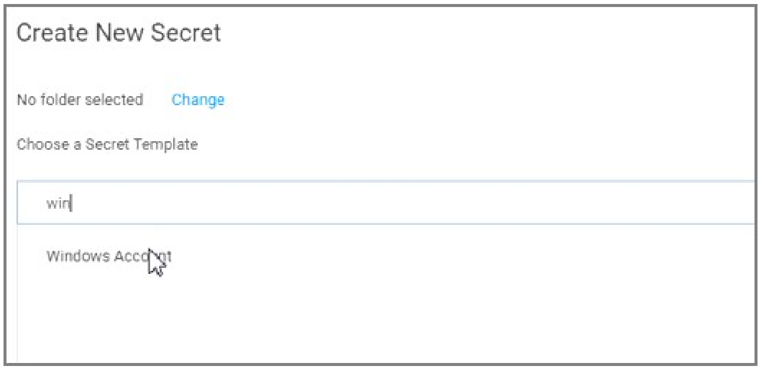
1. Navigate back to **Home** in Secret Server.

2. Click on **(+)** icon next to **Secrets**.



3. Search for the **Windows Account** template.

4. Click on **Windows Account**.



5. For the New Windows Account template, you will need to complete the following fields:

   - **Name**: Name of the Secret
   - **Machine**: The server that Automation Anywhere is installed onto
   - **Username and Password**: The account that has access to the Automation Anywhere Server

6. Click **Create Secret**.

7. Click on the **Sharing** Tab.

8. Click on **Edit**.

9. Under the Search area for **Add Groups/Users**, enter in your **Automation Anywhere application account**.

10. Click **Save**.

Setup a new MetaBot in Automation Anywhere

1. Go to the Automation Anywhere home directory where robots are located. By default, this directory is C:\Users\<current user>\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks\Bot Store.
2. Download and unzip installation package.
3. The unpacked package will contain the following directory:

```
< Automation Anywhere Directory >
  My Task
    Bot Store
      SecretServer-Thycotic
        Error Folder
        Input Folder
        My MetaBots
          File: Thycotic-SDK.mbot
        My Tasks
          File: Thycotic-SDK.atmx
  My Metabots
```

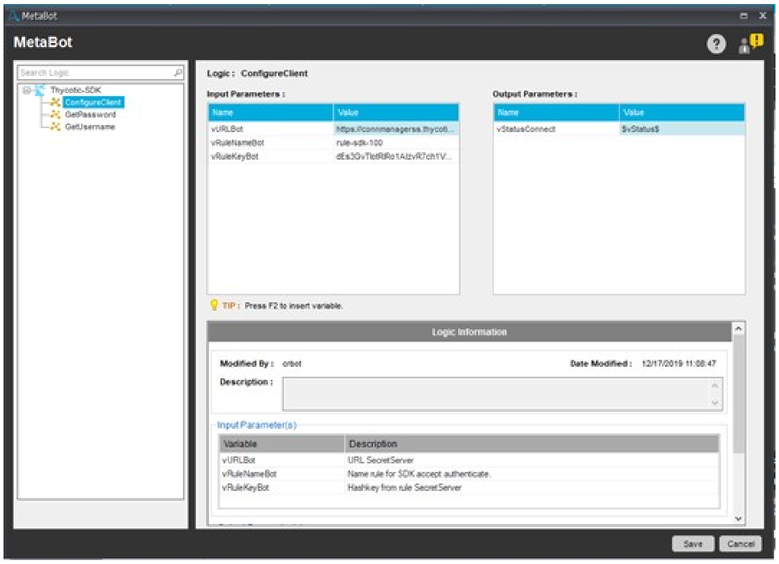
1. Copy MetaBot file (<Automation Anywhere>\My Task\Bot Store\SecretServer-Thycotic\My Metabots\Thycotic-SDK.mbot)

into:

<Automation Anywhere>\My Metabots director.

Retrieving a Secret UserID and Password

1. Configure the connection information for your Thycotic Server in the MetaBot Designer.



Logic : ConfigureClient

Input Parameters :

Name	Value
vURLBot	https://connmanagersa.thycoti...
vRuleNameBot	rule-sdk-100
vRuleKeyBot	dEs3QvTlotRlRo1AlzvR7ch1V...

Output Parameters :

Name	Value
vStatusConnect	$vStatus$

TIP : Press F2 to insert variable.

Logic Information

Modified By : orbot

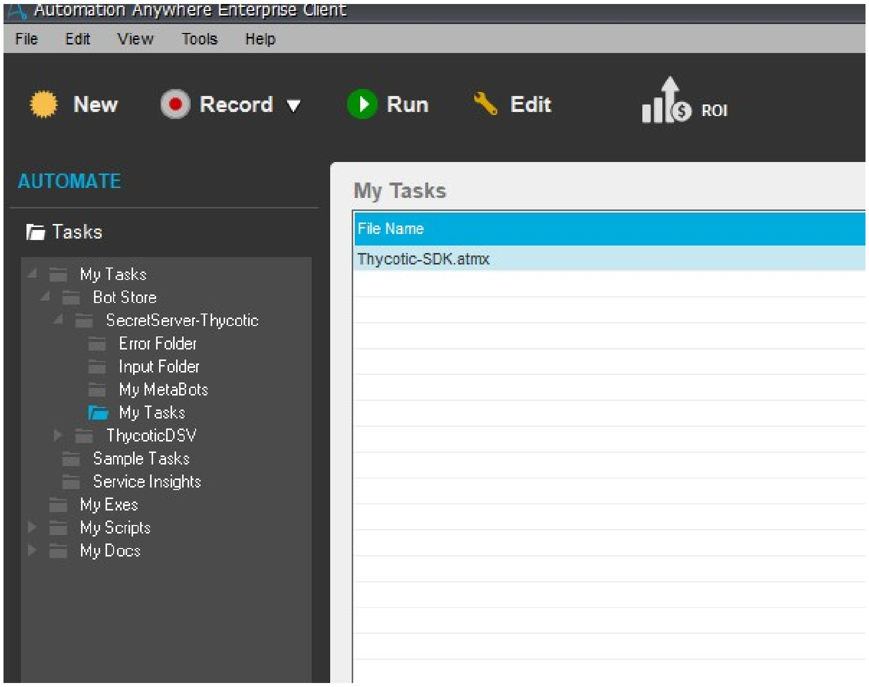
Date Modified : 12/17/2019 11:08:47

Description :

Input Parameter(s)

Variable	Description
vURLBot	URL SecretServer
vRuleNameBot	Name rule for SDK accept authenticate.
vRuleKeyBot	Hashkey from rule SecretServer

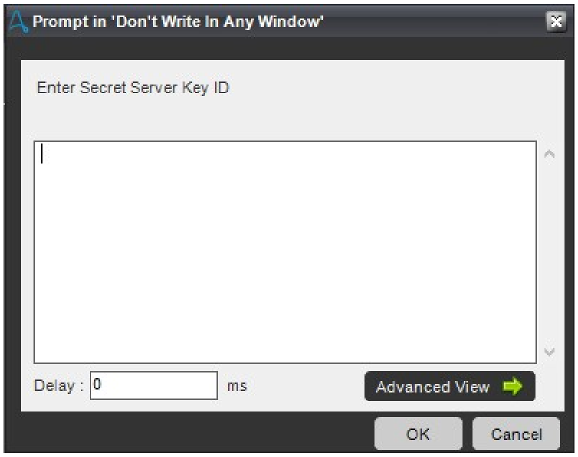
2. Run the provided Automation Anywhere Task File (Thycotic-SDK.atmx) in the Enterprise Client.



My Tasks

File Name
Thycotic-SDK.atmx

3. In the window prompt, enter in a Secret Server Key ID.

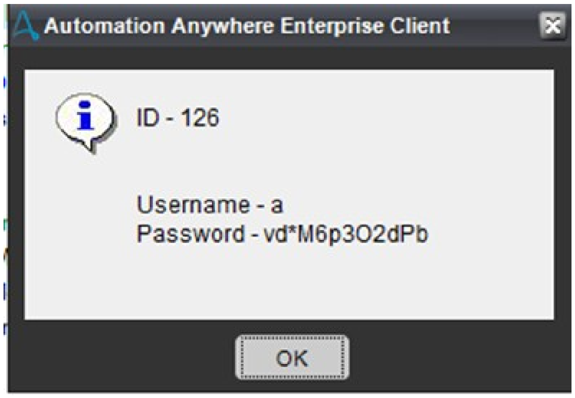


Prompt in 'Don't Write In Any Window'

Enter Secret Server Key ID

Delay : 0 ms

4. Secret Server will return the username and Password for the Secret ID.



Automation Anywhere Enterprise Client

ID - 126

Username - a
Password - vd*M6p3O2dPb

Blue Prism Integration with Secret Server

Blue Prism makes enterprise robotic process automation (RPA) software that provides a digital workforce designed to automate complex, end-to-end operational activities. The product is built on the Microsoft .NET Framework. It automates any application and supports any platform (mainframe, Windows, WPF, Java, web, etc.); and can be consumed in a variety of ways (terminal emulator, thick client, thin client, web browser, Citrix and web services).

The integration between Secret Server and Blue Prism ensures that:

- Passwords are securely vaulted in Secret Server
- Blue Prism bots can request passwords whenever they are needed
- All access by bots is captures in Secret Server Audit Trail
- Role-based access controls limit which passwords each bot can use

System Requirements

Pre-requisites

Product		Details
Blue Prism RPA		
	Version	Blue Prism RPA Version 6.7 or later
	Software	
		BPA Object - Utility - Collection Manipulation
		BPA Object - Utility - Environment
		BPA Object - Utility – General
		BPA Object - Utility – HTTP
		BPA Object - Utility – JSON
		BPA Object - Utility – Strings
		BPA Object - Webservices – REST
		Note: all above visual business objects can be found in Program Files\Blue Prism Limited\Blue Prism Automate\VBO
		BPA Login Agent - required only if bot will need to log into a remote computer (see Login Agent). An example of this implementation can be found here: Program Files\Blue Prism Limited\Blue Prism Login Agent
	Hardware	
		For further information about hardware requirements please contact Blue Prism.
Secret Server		
	Version	
		Thycotic Secret Server 10.8 or later.
	Hardware	
		For further information about system requirements for Secret Server please visit: https://docs.thycotic.com/ss/10.8.0/secret-server-setup/system-requirements/index.md.

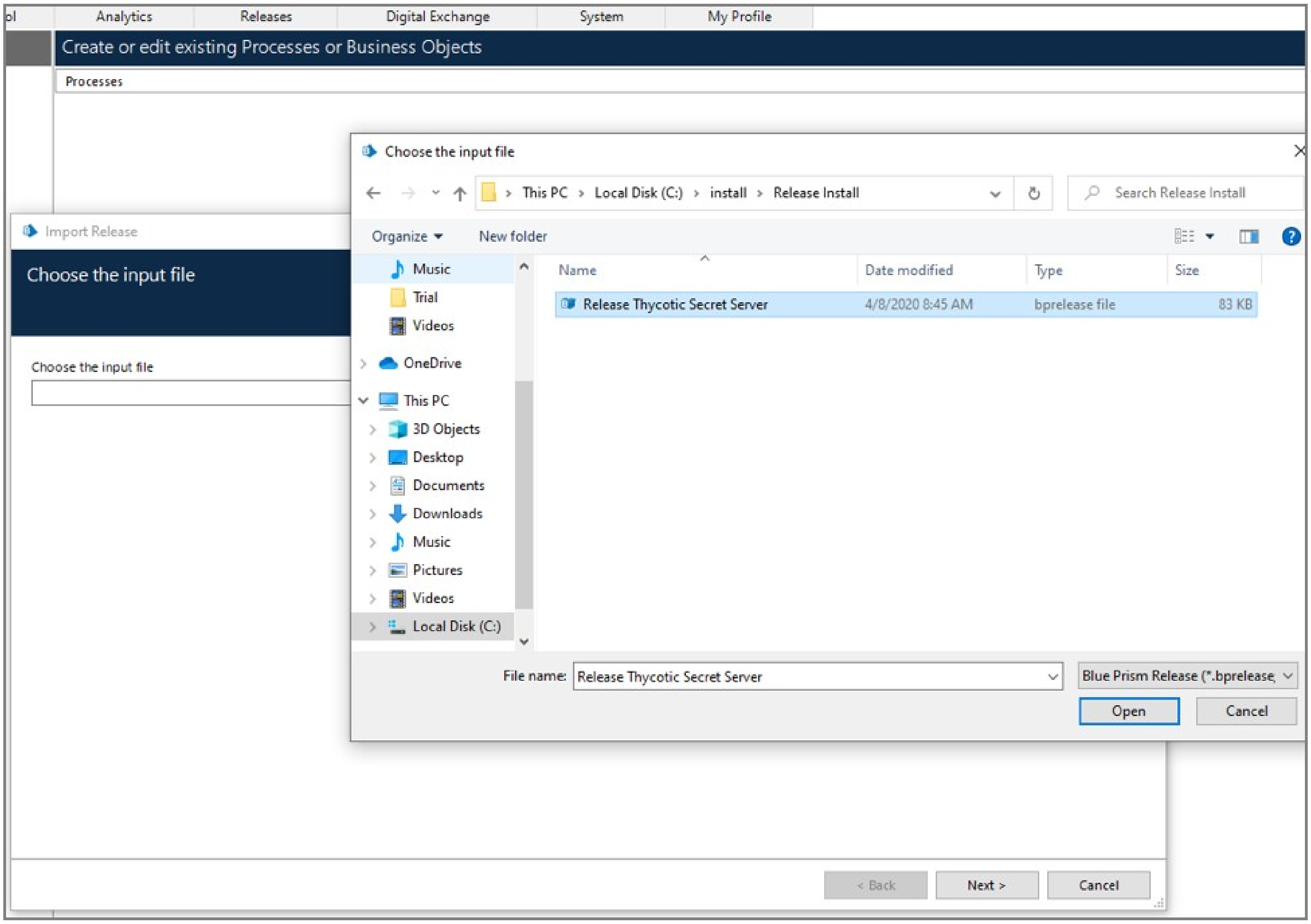
Configuration

- Import the Thycotic Secret Server Integration
- Configure Credentials for Secret Server
- Implementation Considerations

Import the Thycotic Secret Server Integration

1. In order to connect Secret Server with BluePrism, the package “Release Thycotic Server.bprelease” must be present on the BluePrism host. Please click here to download the package for Blue Prism.

2. In BluePrism, select **Import** and choose the Thycotic integration as the input file.

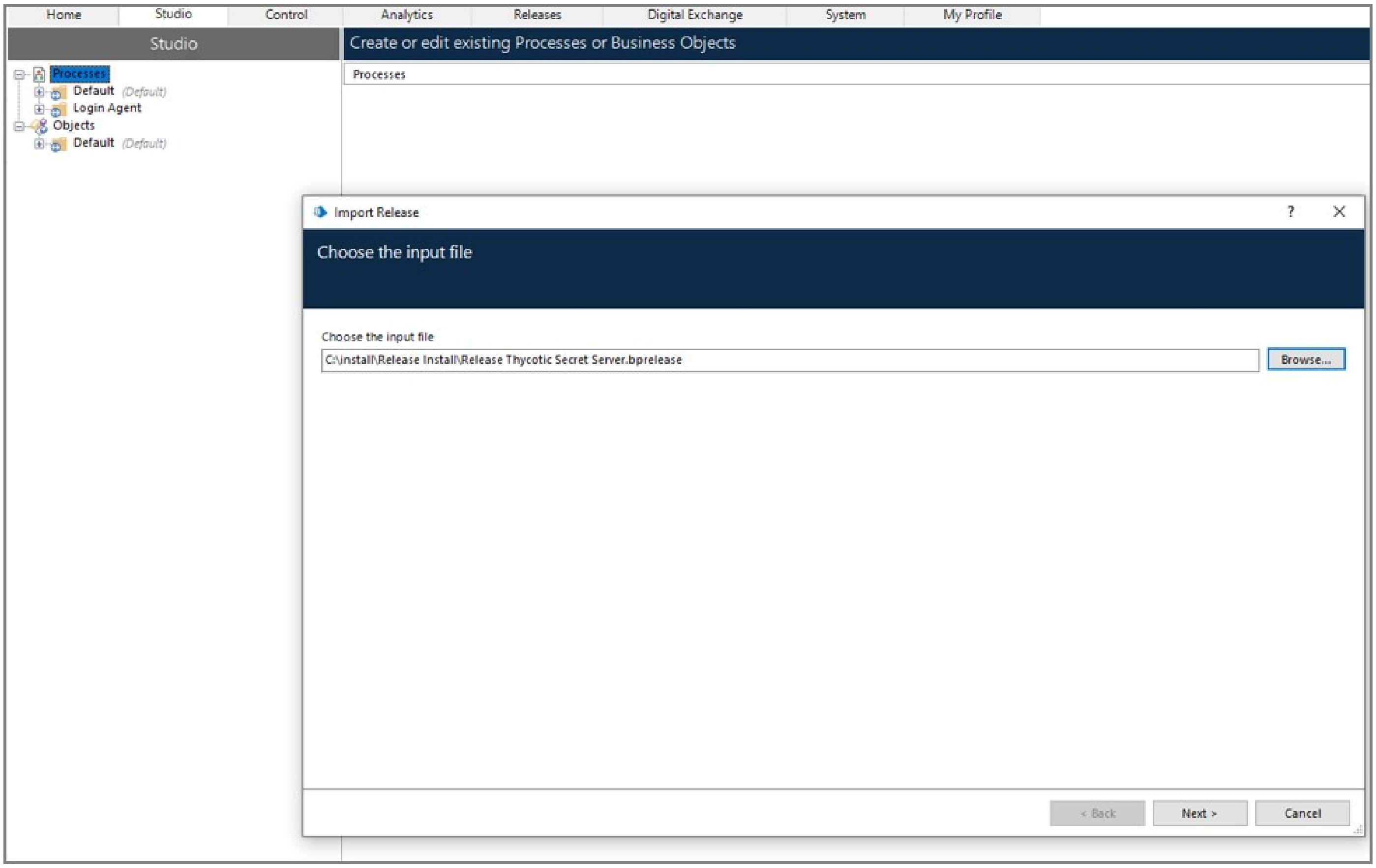


Name	Date modified	Type	Size
Release Thycotic Secret Server	4/8/2020 8:45 AM	bprelease file	83 KB

File name: Release Thycotic Secret Server

Blue Prism Release (*.bprelease

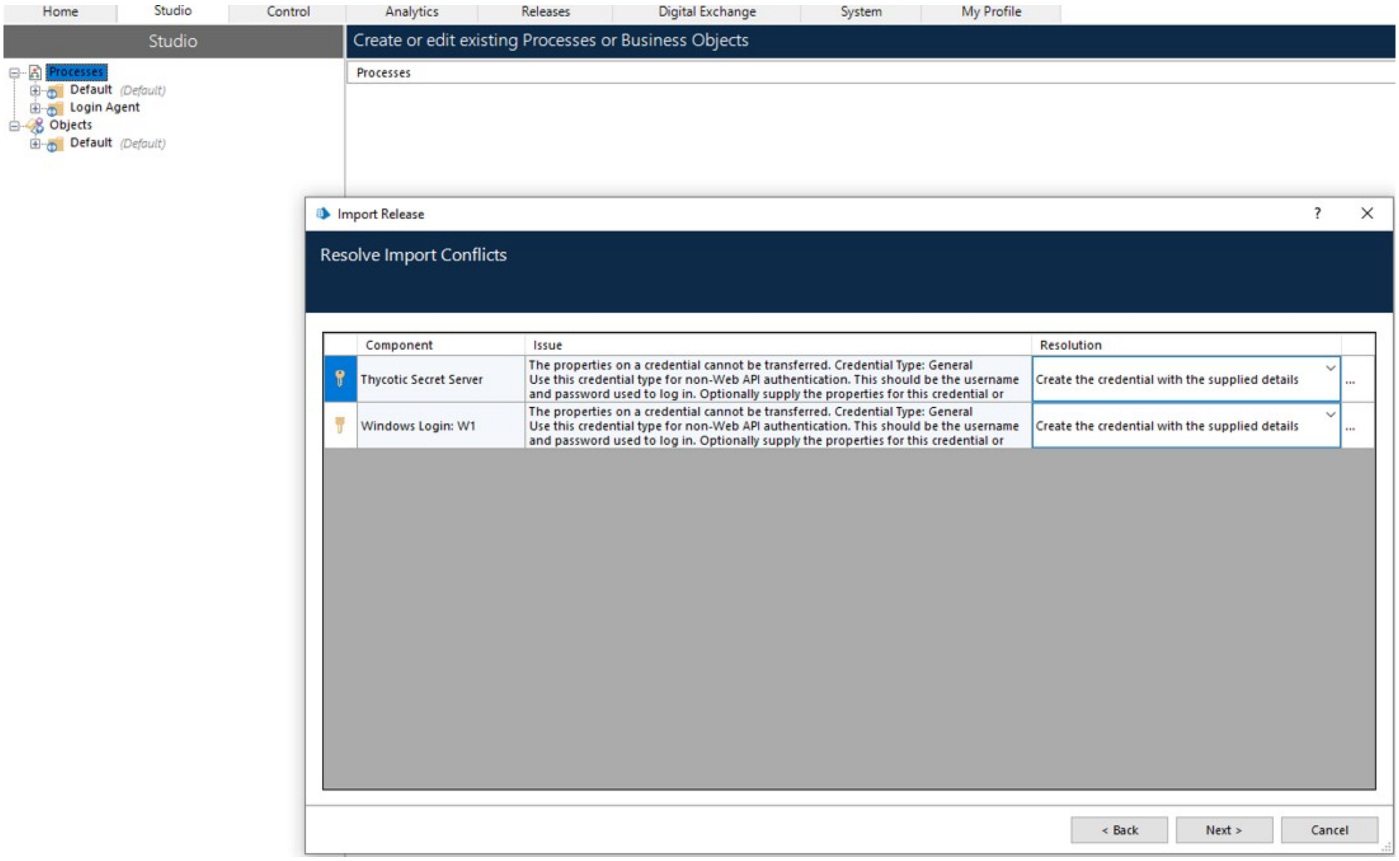
3. Confirm the input file is **Release Thycotic Secret Server.bprelease** and click **Next**.



Choose the input file

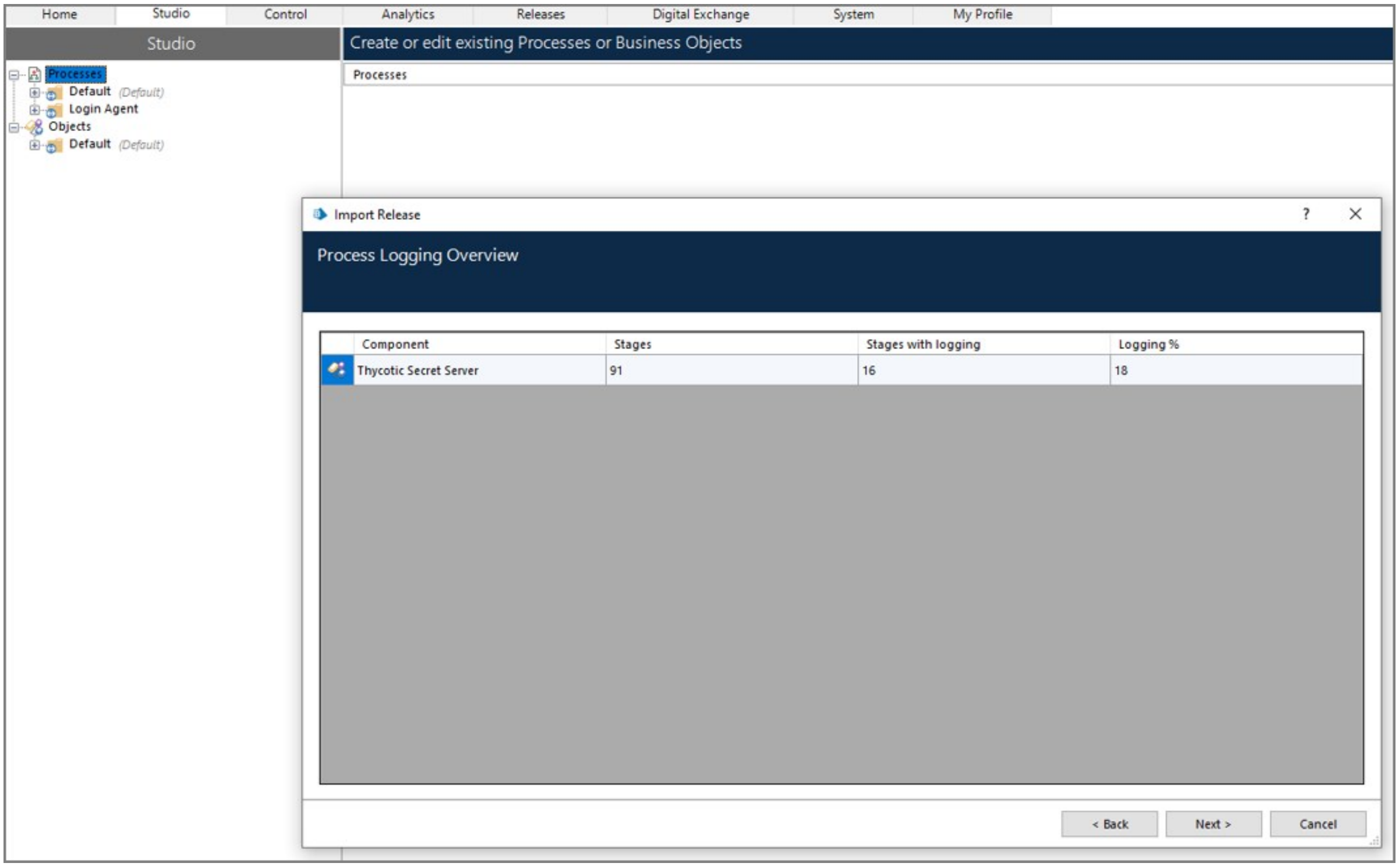
C:\install\Release Install\Release Thycotic Secret Server.bprelease

4. On the Resolve Import Conflicts page, confirm that the Resolution is **Create the credential with the supplied details** and click **Next**.



Component	Issue	Resolution
Thycotic Secret Server	The properties on a credential cannot be transferred. Credential Type: General Use this credential type for non-Web API authentication. This should be the username and password used to log in. Optionally supply the properties for this credential or	Create the credential with the supplied details
Windows Login: W1	The properties on a credential cannot be transferred. Credential Type: General Use this credential type for non-Web API authentication. This should be the username and password used to log in. Optionally supply the properties for this credential or	Create the credential with the supplied details

5. On the Process Logging Overview page, click **Next**.



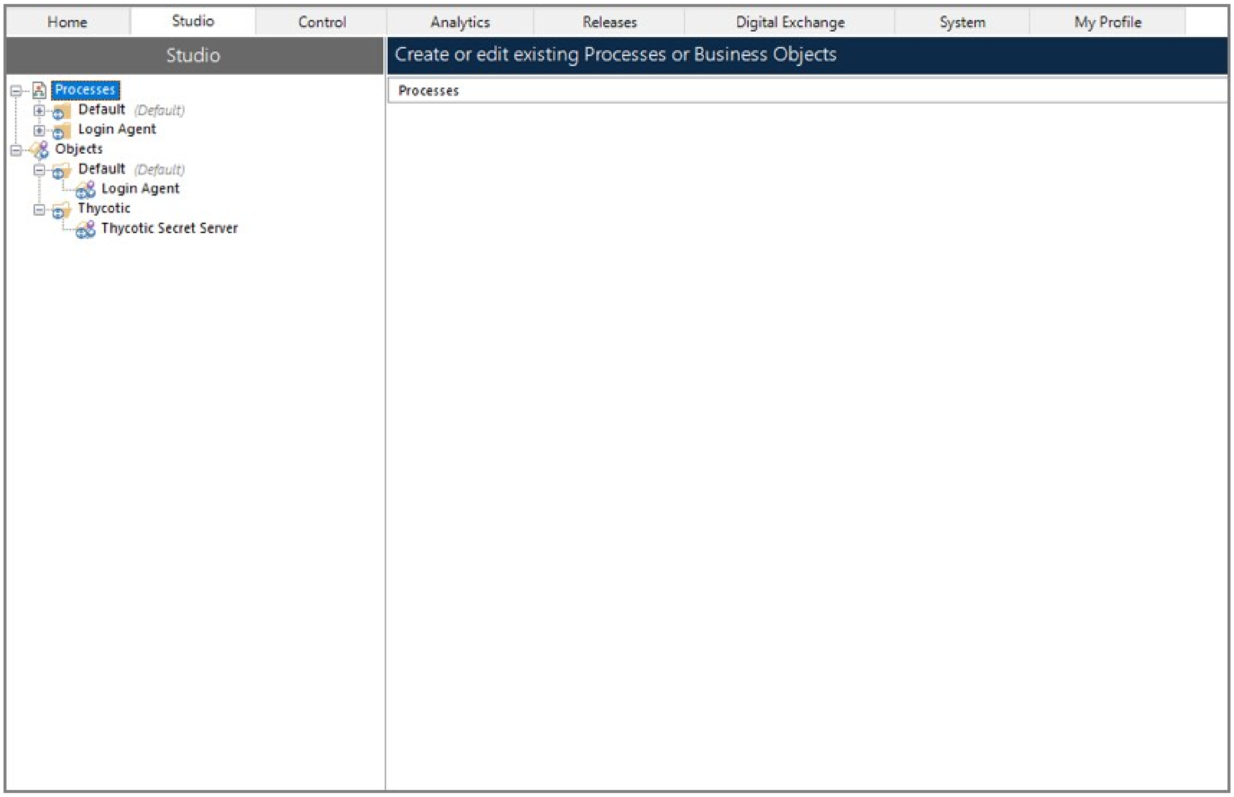
Create or edit existing Processes or Business Objects

Processes

Process Logging Overview

Component	Stages	Stages with logging	Logging %
Thycotic Secret Server	91	16	18

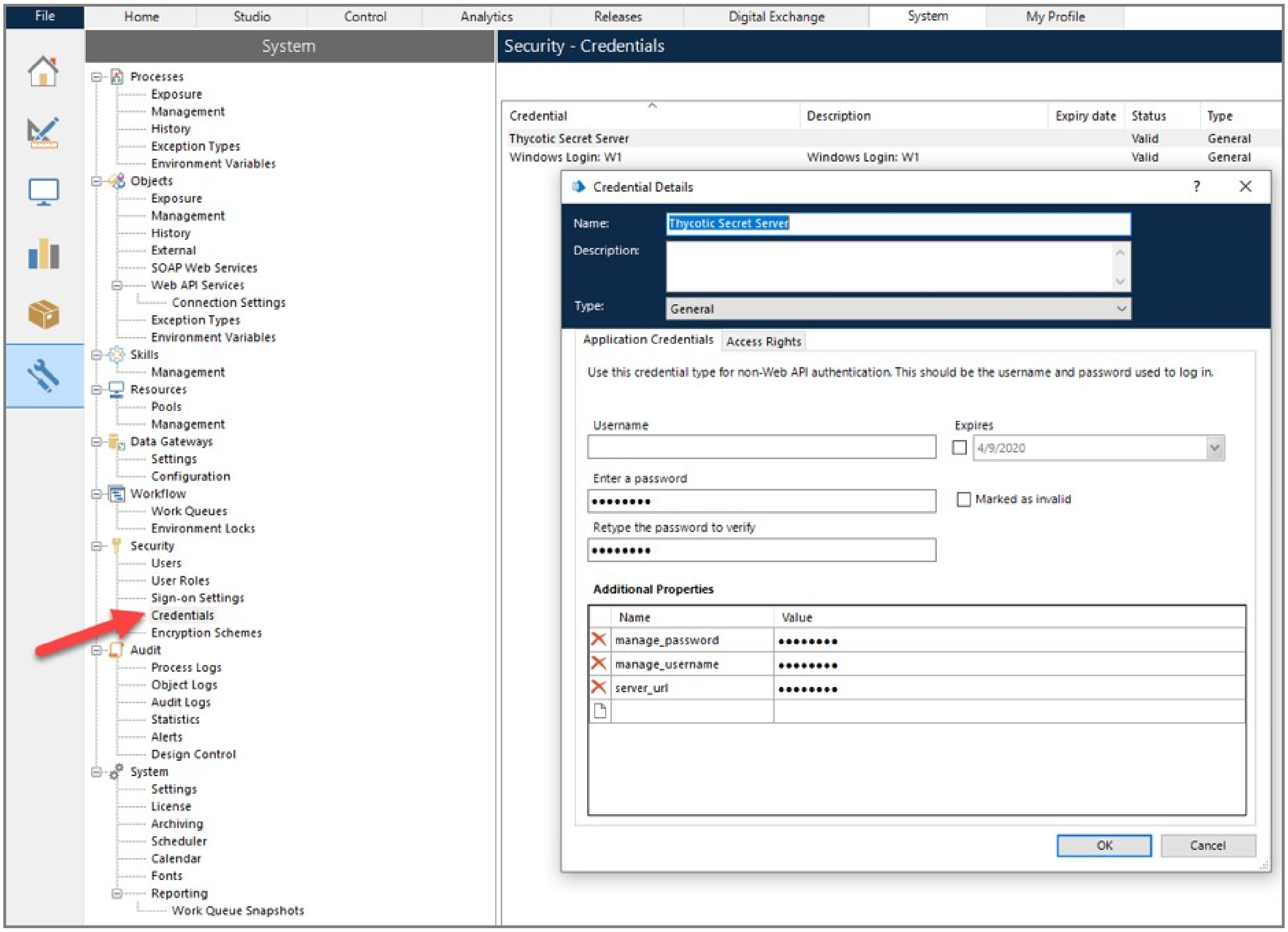
6. The package import should now be complete.



Home
Studio
Control
Analytics
Releases
Digital Exchange
System
My Profile
Studio
Create or edit existing Processes or Business Objects
Processes
Default (Default)
Login Agent
Objects
Default (Default)
Login Agent
Thycotic
Thycotic Secret Server
Processes

Configure Credentials for Secret Server

After a successful BP package import, configure the account settings for Secret Server.



Credential	Description	Expiry date	Status	Type
Thycotic Secret Server			Valid	General
Windows Login: W1	Windows Login: W1		Valid	General

Name: Thycotic Secret Server

Description:

Type: General

Use this credential type for non-Web API authentication. This should be the username and password used to log in.

Username

Expires □ 4/9/2020

Enter a password ••••••••

□ Marked as invalid

Retype the password to verify ••••••••

Additional Properties

Name	Value
manage_password	••••••••
manage_username	••••••••
server_url	••••••••

1. Create a credential entry. In the example above we used the name **Thycotic Secret Server**.

2. Enter the required settings:

   - **server_url**: base URL for access Thycotic Secret Server, for example https://node1.thycotic.com/SecretServer
   - **manage_username**: username for access Secret Server
   - **manage_password**: password for access Secret Server

3. In the **Access Rights** tab, select the required rights.

Implementation Considerations

Blue Prism Visual Business Object Studio allows you to call Thycotic Secret Server REST APIs. You can find additional information regarding the available APIs here.

Information about the BluePrism Login Agent can be found here here.

There are two primary use-cases for using BluePrism and Secret Server.

1. Requesting credentials for the purpose of logging into a machine with a BluePrism robot.
2. Requesting credentials as part of a workflow that requires an additional password.

For both cases, you will need the following:

- A Secret Server access token for use with API requests
- A SecretID to query the correct Secret in Secret Server
- The username associated with the Secret
- The password associated with the Secret

The **SecretID** of a Secret can be found by navigating to that Secret in the Secret Server UI, and looking at the URL. The Secret Server API also offers the ability to look up a SecretID by the Secret's name, or by the Secret Name and Secret Template.

An optional Secret Server feature called Checkout enables users to reserve exclusive access to a given Secret for a period of time, or until it is checked back in. API methods for checking Secrets out and in can be used in this situation.

Implementation Examples

This section covers the following implementation examples:

- GetToken
- GetIdByName
- GetIdByNameAndTemplate
- GetUsernameById
- GetPasswordById
- Check-Out
- Check-in
- Executing an automated process on a Blue Prism Runtime Resource

GetToken

GetToken allows you to retrieve an access token for use with other API requests APIOAuth2

REST API URL /oauth2/token

Please see the settings below for the input parameters:

- **CredentialName**: Credential name where the store accesses the parameters.
- **Property username**: Accesses parameters store in the credential vault for Blue Prism
- **Property password**: Accesses parameters store in the credential vault for Blue Prism
- **Property server_url**: Accesses parameters store in the credential vault for Blue Prism

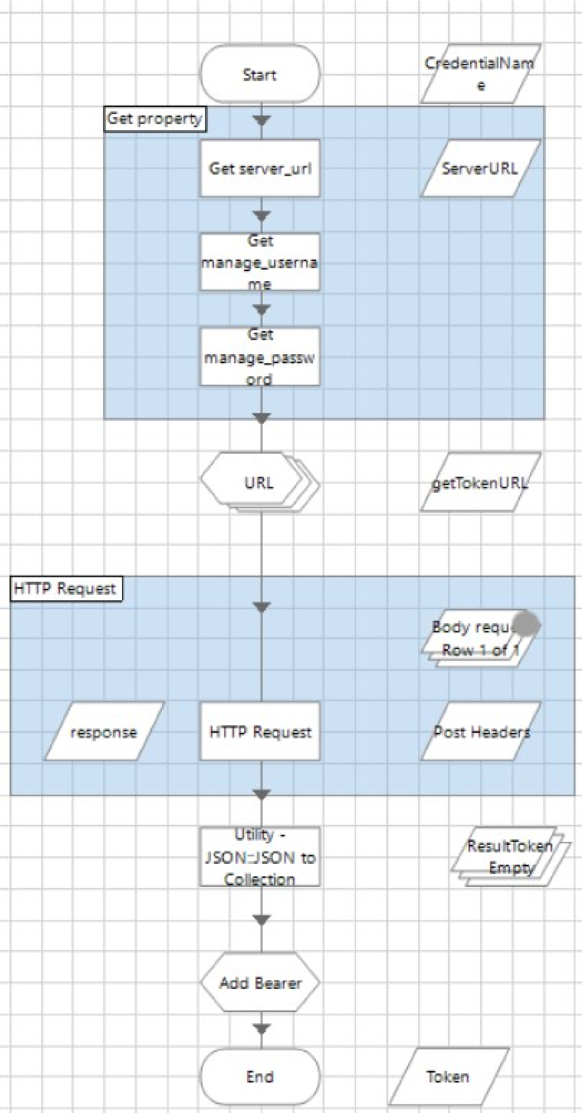
Please see the settings below for the output parameters:

- String Token authenticate Bearer token

Example

```
bearer
gLv3yXL4uf2g004_JfNwqcKciSB30JJ9aXFiAJGzdqpeQrYtIiZ2vbiIkag0bVMQ6Ic-v2r4ugpcKfW2wCpLWia3Gh0uwYDjoFeFpQ-
DZFzz56TNptwud8WWM6S2YVBzYVCek_5QevLpHar897vl-iSHUlM4nYd4YcyGwIdhtiy_pEb-
UaAXfjXN8aJcxNGuvHb0aZ5DjPIJH6ynKlImoRGKd2A897z3fdesAxcBpcUjDfBcRx-
Q7eSOcykKxFUfsJrcpgWJHdhz5xzxvB2kERsLVxBVcn_FE3JMRyprA4DiSFYEsf1KIKEBql57MpC8zyj2tPg1sS0Zp-
mH6ZERIIn9JqjIwixfMxZeSUxQ4677ffZVLgjWrGcjnnuqAXDmqARt_3uTljMVMoXXqut9BToJXG8AY7wjW0z29HH6D70vZ3s70bPozAEu8k6oK8f0EsSFoUhjEwI7T3oGdRNNW
NhncGp4VHqtxUYqw-EnaUY2PPq2hub10hQ1UODDyoE0j8CrVgZXDChlKKjljNtdEAavw-mawqbjo7t7KUmFVO0GGjnYun_9rR6WG2ZlM5z200
```

Scheme


Start
CredentialName
Get property
Get server_url
ServerURL
Get manage_username
Get manage_password
URL
getTokenURL
HTTP Request
Body reque
Row 1 of 1
response
HTTP Request
Post Headers
Utility - JSON::JSON to Collection
ResultToken
Empty
Add Bearer
End
Token

GetIdByName

This method searches for a Secret ID using the Secret name field (string value)

REST API URL /api/v1/secrets?filter.searchField=name&filter.searchText=<search name>

Please see the settings below for the input parameters:

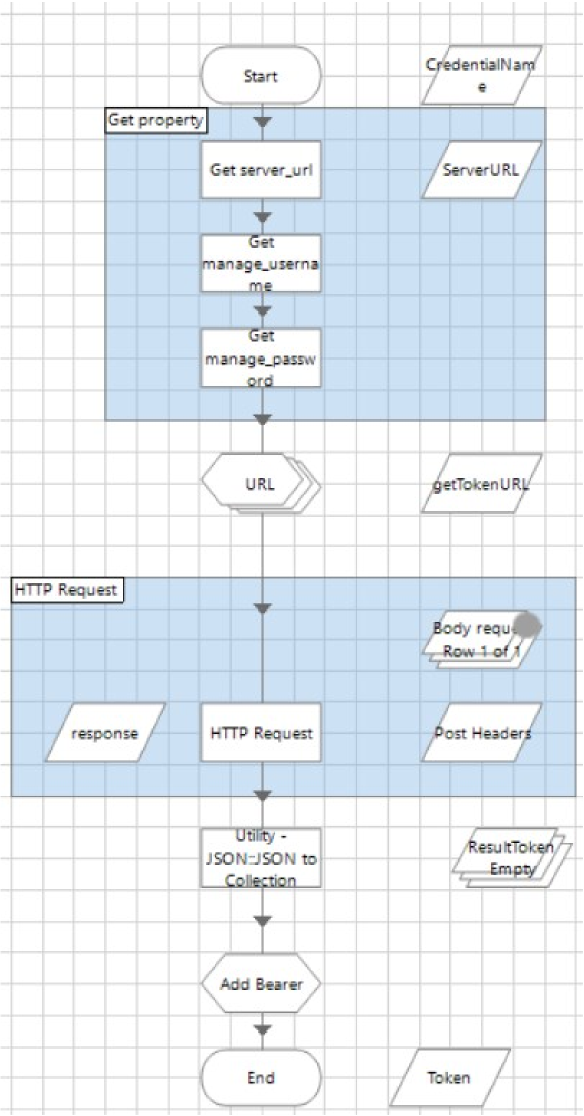
<secret name> search value, secret name string

- **CredentialName**: Credential name where the store accesses the parameters.
- **Property username**: Accesses parameters store in the credential vault for Blue Prism
- **Property password**: Accesses parameters store in the credential vault for Blue Prism
- **Property server_url**: Accesses parameters store in the credential vault for Blue Prism

Please see the settings below for the output parameters:

- **Collection Secret Model**: where key collection ID secret.

Scheme


Start
CredentialNam
e
Get property
Get server_url
ServerURL
Get
manage_userna
me
Get
manage_passw
ord
URL
getTokenURL
HTTP Request
Body requ
Row 1 of 1
response
HTTP Request
Post Headers
Utility -
JSON::JSON to
Collection
ResultToken
Empty
Add Bearer
End
Token

GetIdByNameAndTemplate

This method searches for a Secret ID using the Secret Name and Template ID.

REST API URL /api/v1/secrets?filter.searchField=name&filter.secretTemplateId=<templateID>&filter.searchText=<search name>

Please see the settings below for the input parameters:

<secret name> search value, secret name string

<TemplateID> restricted template for search, number TemplateID

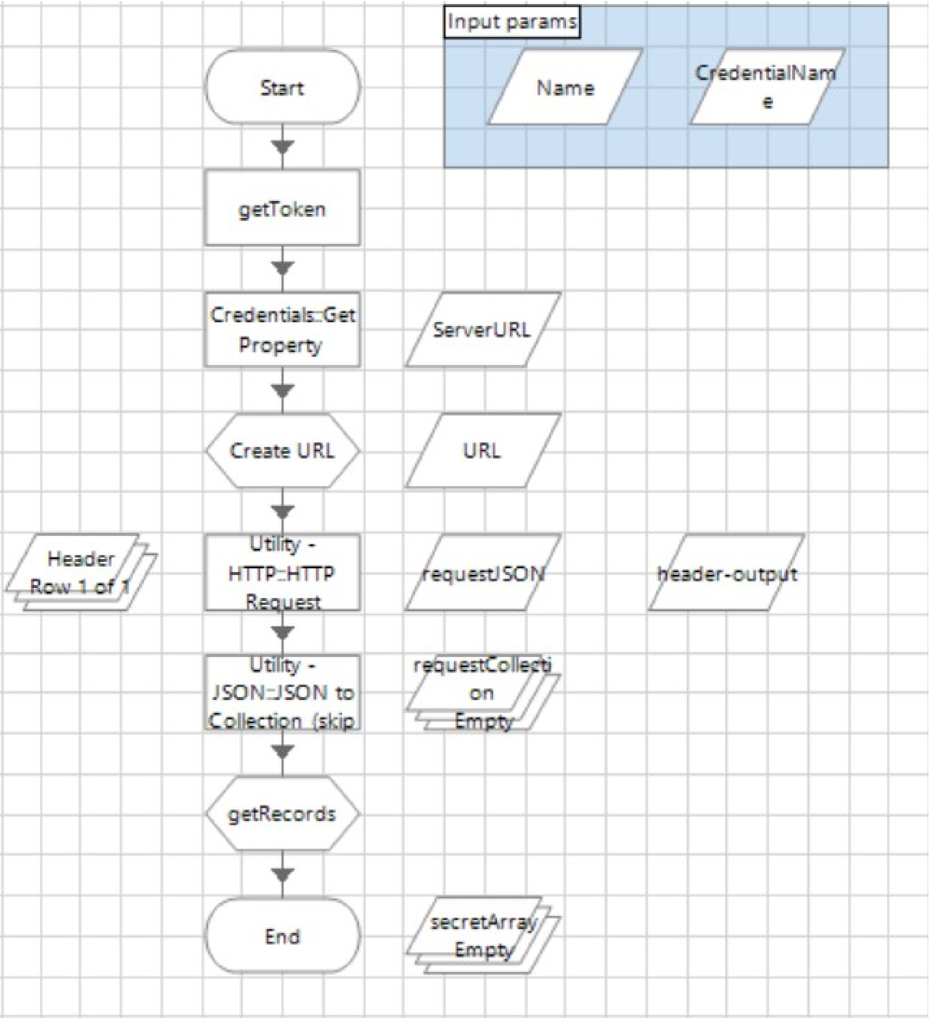
- **CredentialName**: Credential name where the store accesses the parameters.
- **Property username**: Accesses parameters store in the credential vault for Blue Prism
- **Property password**: Accesses parameters store in the credential vault for Blue Prism
- **Property server_url**: Accesses parameters store in the credential vault for Blue Prism

Please see the settings below for the output parameters:

Please see the settings below for the output parameters:

- **Collection Secret Model**: Where key collection ID secret.

Scheme


Input params
Start
Name
CredentialName
getToken
Credentials::Get Property
ServerURL
Create URL
URL
Header Row 1 of 1
Utility - HTTP::HTTP Request
requestJSON
header-output
Utility - JSON::JSON to Collection (skip
requestCollection Empty
getRecords
End
secretArray Empty

GetUsernameById

This method retrieves the current password from a secret using an ID.

REST API URL "/api/v1/secrets/<secretid>"

Please see the settings below for the input parameters:

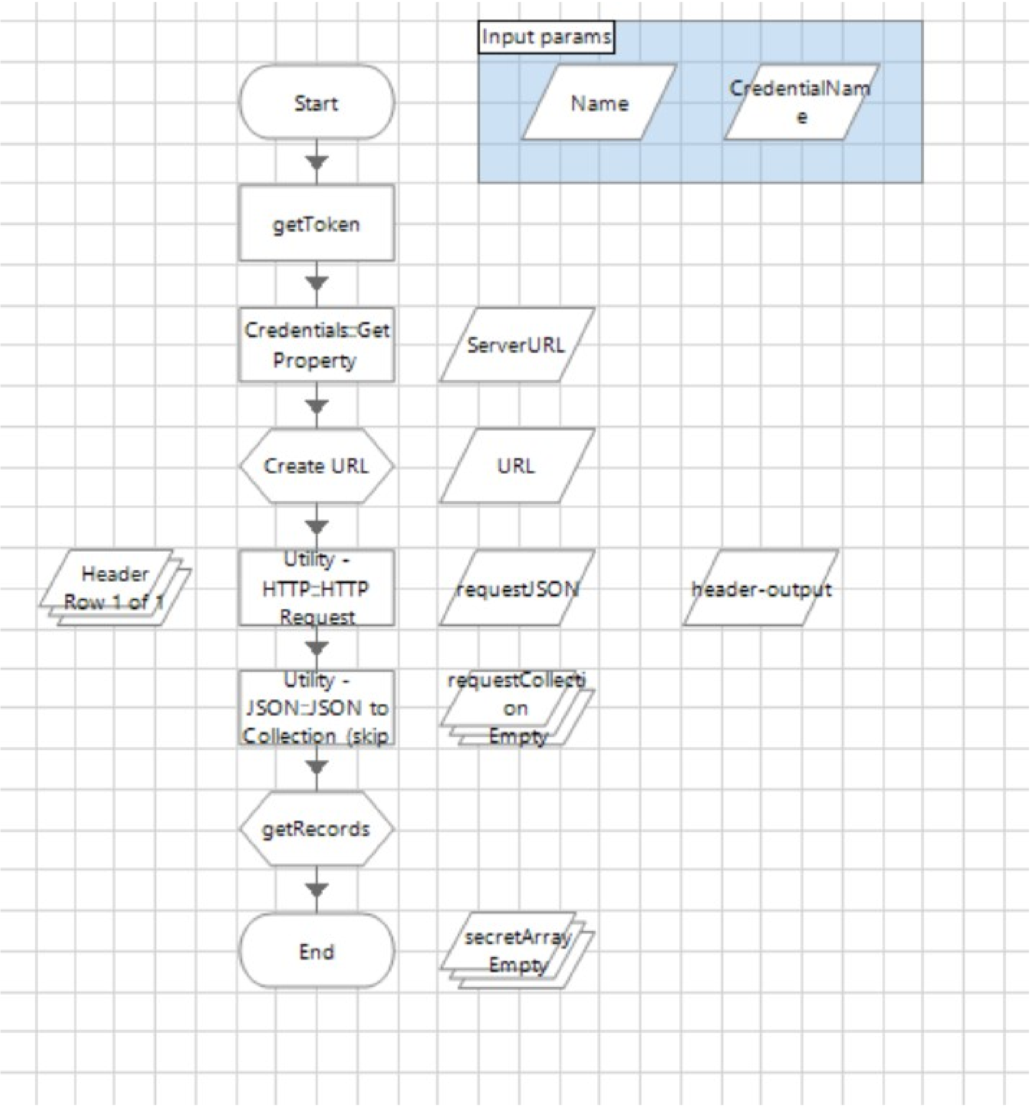
<secret ID> secret ID value number

- **CredentialName**: Credential name where the store accesses the parameters.
- **Property username**: Accesses parameters store in the credential vault for Blue Prism
- **Property password**: Accesses parameters store in the credential vault for Blue Prism
- **Property server_url**: Accesses parameters store in the credential vault for Blue Prism

Please see the settings below for the output parameters:

- Username string field from the secret
- *Exception*: if secret is in checkOut status, throws an exception and returns an error code

Scheme


Input params
Start
Name
CredentialNam
e
getToken
Credentials::Get
Property
ServerURL
Create URL
URL
Header
Row 1 of 1
Utility -
HTTP::HTTP
Request
requestJSON
header-output
Utility -
JSON::JSON to
Collection (skip
requestCollecti
on
Empty
getRecords
End
secretArray
Empty

GetPasswordById

This method retrieves the current password from a secret ID.

REST API URL "/api/v1/secrets/<secretid>"

Please see the settings below for the input parameters:

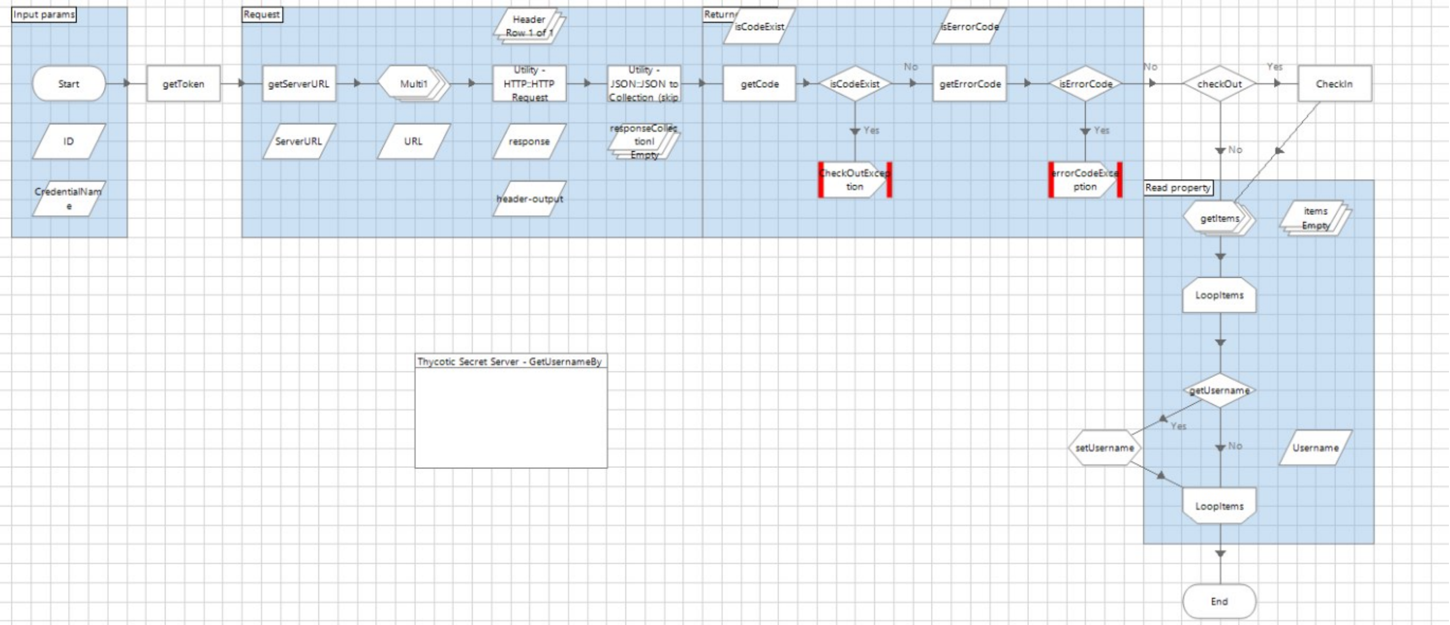
<secret ID> secret ID value number

- **CredentialName**: Credential name where the store accesses the parameters.
- **Property username**: Accesses parameters store in the credential vault for Blue Prism
- **Property password**: Accesses parameters store in the credential vault for Blue Prism
- **Property server_url**: Accesses parameters store in the credential vault for Blue Prism

Please see the settings below for the output parameters:

- Password invisible string password
- *Exception*: if secret is in checkOut status, throws an exception and returns an error code

Scheme



Check-Out

This method checks out a secret

REST API URL /api/v1/secrets/<ID>/check-out

Input params

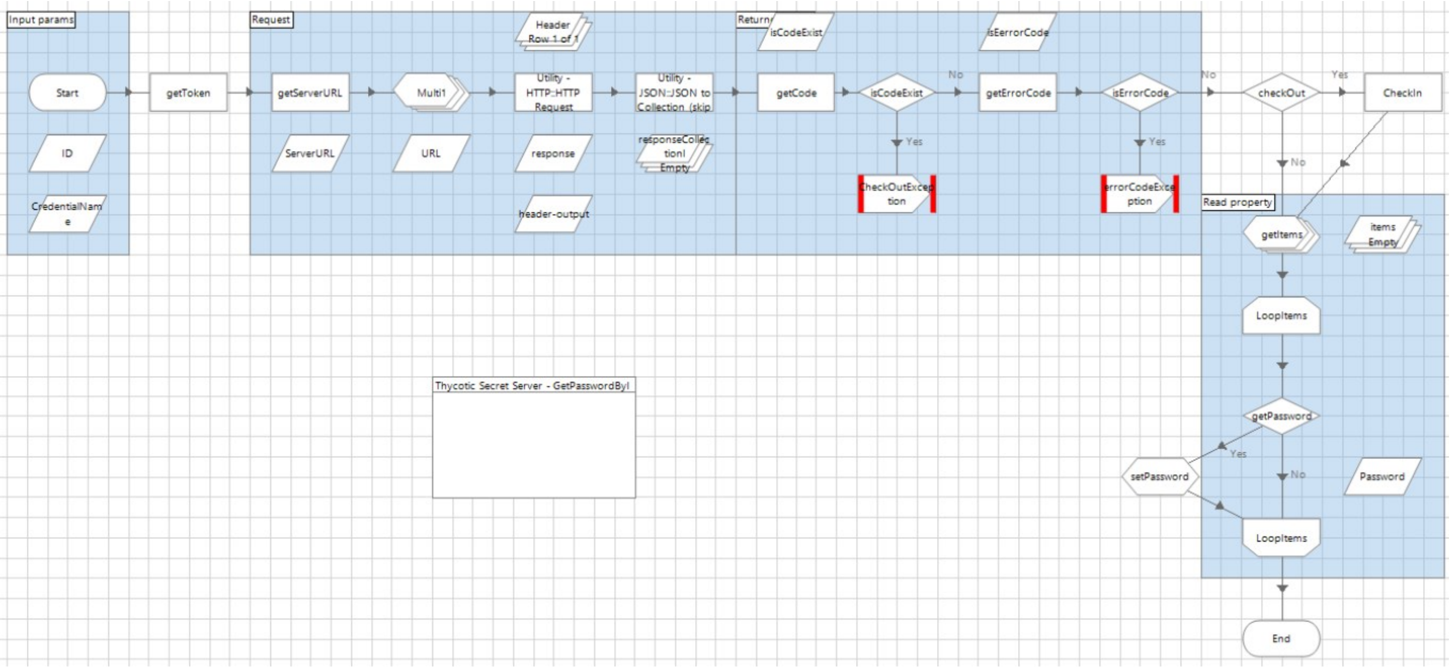
<ID> ID secret for operation, ID number

- **CredentialName**: Credential name where the store accesses the parameters.
- **Property username**: Accesses parameters store in the credential vault for Blue Prism
- **Property password**: Accesses parameters store in the credential vault for Blue Prism
- **Property server_url**: Accesses parameters store in the credential vault for Blue Prism

Please see the settings below for the output parameters:

- *Exception*: if request returns error code, throws an exception

Scheme



Check-In

This method checks out a secret

REST API URL /api/v1/secrets/<ID>/check-in

Input params

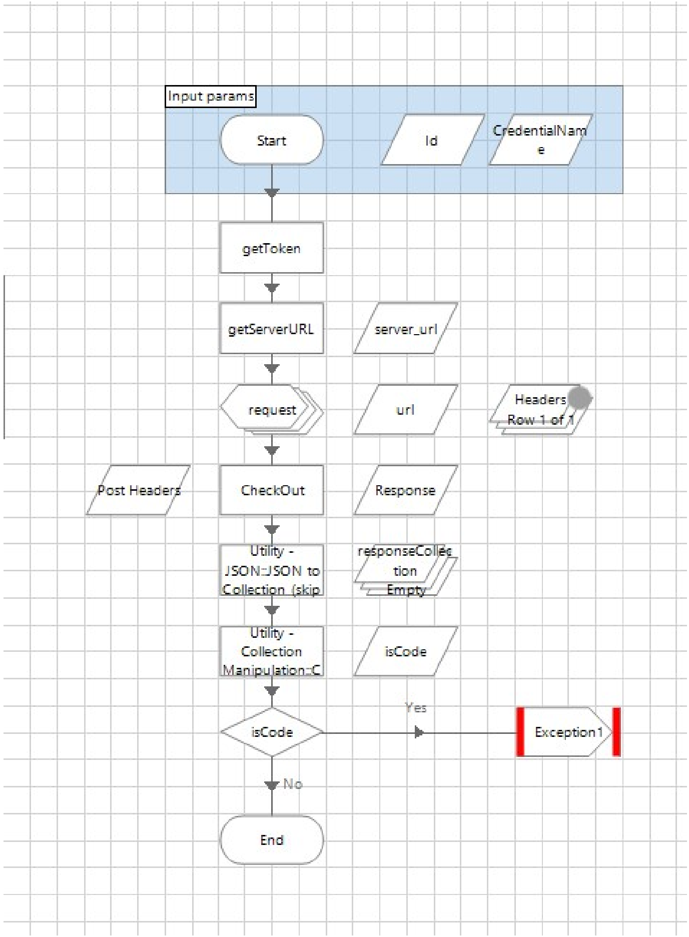
<ID> ID secret for operation, ID number

- **CredentialName**: Credential name where the store accesses the parameters.
- **Property username**: Accesses parameters store in the credential vault for Blue Prism
- **Property password**: Accesses parameters store in the credential vault for Blue Prism
- **Property server_url**: Accesses parameters store in the credential vault for Blue Prism

Please see the settings below for the output parameters:

- *Exception*: if request returns error code, throws an exception

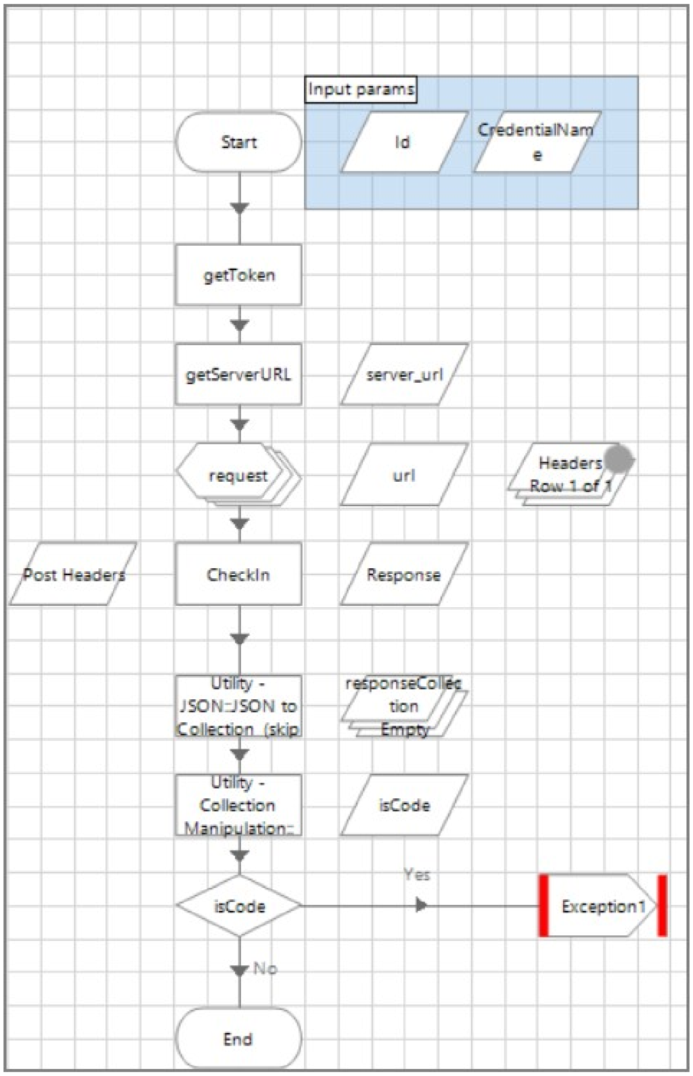
Scheme



Executing an automated process on a Blue Prism Runtime Resource

The VBO LoginAgent allows a process to operate under the context of the logged in user and provides access to all local applications and network resources that it may need.

Scheme



Rotating the BluePrism API Account

This document is designed to provide a guide for rotating the password to the privileged account (Secret) that is used by BluePrism to access privileged accounts (Secrets) stored in Thycotic Secret Server on behalf of Digital Workers in the BluePrism platform.

The credential that is stored by BluePrism contains three “credential properties” which store the relevant information required for getting access to Secret Server. The properties are “manage_username”,”manage_password” and “server_url”. These details contain the username and password for an “application API” account that the integration uses to get access to Secret Server.

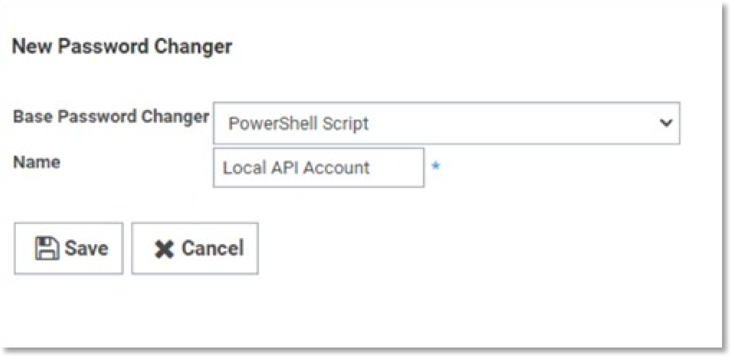
In order to rotate this credential, first Secret Server needs to rotate the password for the local API account that’s being used. Next, it then reaches out to BluePrism to modify the “manage_password” credential property stored against this credential utilizing the AutomateC application provide by BluePrism. This operation is performed remotely on the BluePrism server through the use of Powershell. This guide assumes that the BluePrism integration is already set up and enabled against Secret Server. If documentation around this process is required, an alternative document should be sought.

How to Rotate the BluePrism API Account

Step 1

The first step is to enable local API account rotation for an application account that is stored within Secret Server.

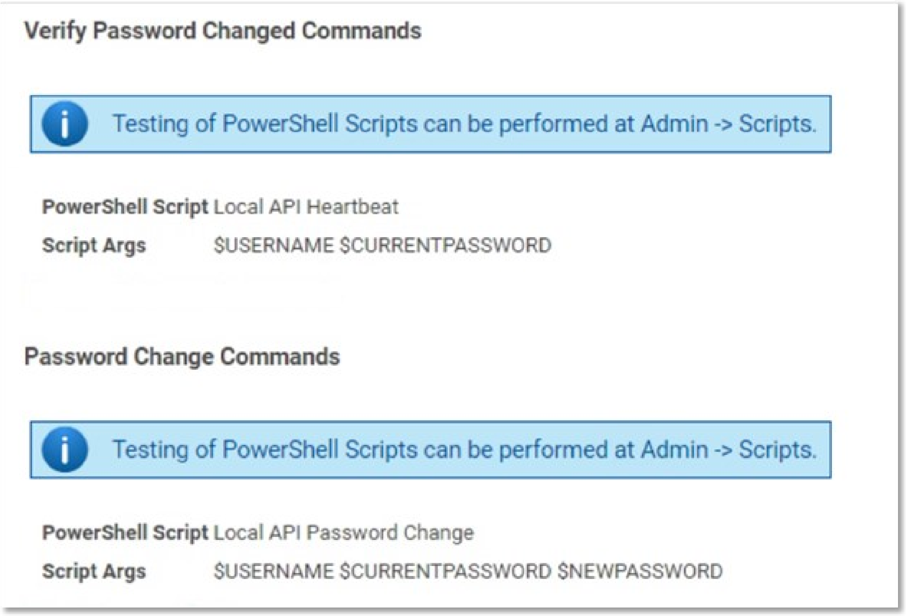
1. Create a custom remote password changer.



New Password Changer

Base Password Changer: PowerShell Script

Name: Local API Account *



Verify Password Changed Commands

Testing of PowerShell Scripts can be performed at Admin -> Scripts.

PowerShell Script: Local API Heartbeat

Script Args: $USERNAME $CURRENTPASSWORD

Password Change Commands

Testing of PowerShell Scripts can be performed at Admin -> Scripts.

PowerShell Script: Local API Password Change

Script Args: $USERNAME $CURRENTPASSWORD $NEWPASSWORD

Heartbeat Script

```
$params = $args
$username = $params[0]
$password = $params[1]
$resturl = "https://<Secret Server Address>/api/v1"
$tokenurl = "https://<Secret Server Address>/oauth2/token"

try
{
 $creds = @{
    username = $username
    password = $password
    grant_type = "password"
 }

 $token = ""
 $token = $response.access_token;
 if($response = Invoke-RestMethod $tokenurl -Method Post -Body $creds)
 {
 Write-Host $response
 try
 {
 $token = $response.access_token;
 $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
 $headers.Add("Authorization", "Bearer $token")
 $tokenreturn = Invoke-RestMethod $resturl/oauth-expiration -Method Post -Headers $headers
 }
 catch
 {
     throw $("Error Details: " + $_)
 }
     return $true
 }
 catch
 {
     throw $("Error Details: " + $_)
 }
 else
 {
     throw $("Error Details: " + $_)
 }

}
catch [System.Net.WebException]
{
 Write-Host "----- Exception -----"
 Write-Host  $_.Exception
 Write-Host  $_.Exception.Response.StatusCode
 Write-Host  $_.Exception.Response.StatusDescription
 $result = $_.Exception.Response.GetResponseStream()
 $reader = New-Object System.IO.StreamReader($result)
 $reader.BaseStream.Position = 0
 $reader.DiscardBufferedData()
 $responseBody = $reader.ReadToEnd() | ConvertFrom-Json
 Write-Host  $responseBody.errorCode " - " $responseBody.message
 foreach($modelState in $responseBody.modelState)
 {
     $modelState
 }
}
```

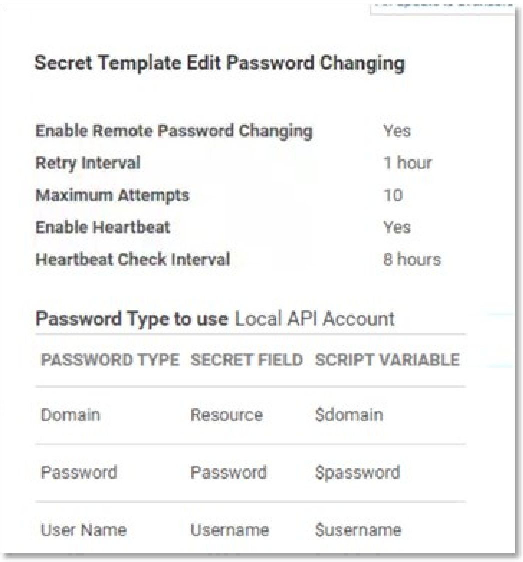
Password Changing Script

```
$params = $args
$username = $params[0]
$password = $params[1]
$newpassword = $params[2]
$resturl = "https://<Secret Server Address>/api/v1"
$tokenurl = "https://<Secret Server Address>/oauth2/token"

try
{
 $creds = @{
     username = $username
     password = $password
     grant_type = "password"
 }

 $userpassargs = @{
     currentPassword = $password
     newPassword = $newpassword
 }

 $token = ""
 $token = $response.access_token;
 if($response = Invoke-RestMethod $tokenurl -Method Post -Body $creds)
 {
 try
 {
 $token = $response.access_token;
 Write-Host $token
 $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
 $headers.Add("Authorization", "Bearer $token")
 $changepw = Invoke-RestMethod $resturl/users/change-password -Method Post -Headers $headers -Body $userpassargs
 $tokenexpire = Invoke-RestMethod $resturl/oauth-expiration -Method Post -Headers $headers
 }
 catch
 {
    throw $("Error Details: " + $_)
 }
    return $true
 }
 catch
 {
    throw $("Error Details: " + $_)
 }
 else
 {
    throw $("Error Details: " + $_)
 }
}
catch [System.Net.WebException]
{
 Write-Host "----- Exception -----"
 Write-Host  $_.Exception
 Write-Host  $_.Exception.Response.StatusCode
 Write-Host  $_.Exception.Response.StatusDescription
 $result = $_.Exception.Response.GetResponseStream()
 $reader = New-Object System.IO.StreamReader($result)
 $reader.BaseStream.Position = 0
 $reader.DiscardBufferedData()
 $responseBody = $reader.ReadToEnd() | ConvertFrom-Json
 Write-Host  $responseBody.errorCode " - " $responseBody.message
 foreach($modelState in $responseBody.modelState)
 {
    $modelState
 }
}
```

Step 2

1. Create a Secret Template for the Local API account.

   Note: This template is the same as the "Password" default template, but with the Local API Password Changer identified in Step 1 added to it as its password changing mechanism.



Secret Template Edit Password Changing

Enable Remote Password Changing	Yes
Retry Interval	1 hour
Maximum Attempts	10
Enable Heartbeat	Yes
Heartbeat Check Interval	8 hours

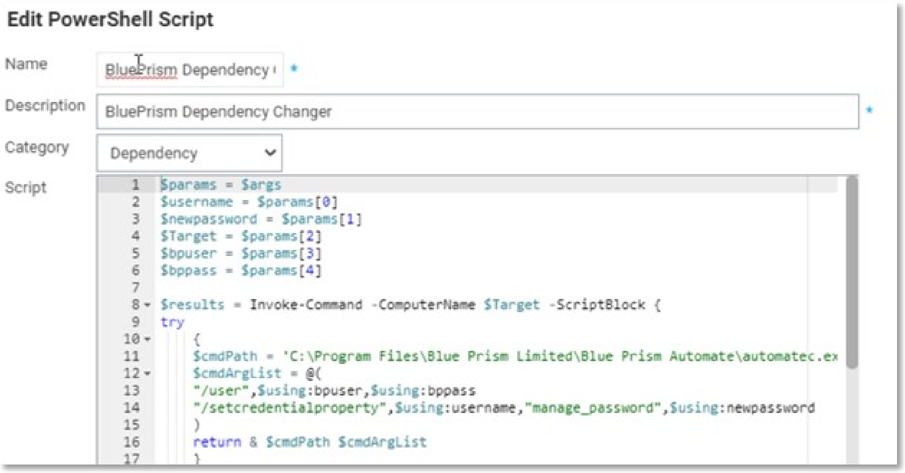
Password Type to use Local API Account

PASSWORD TYPE	SECRET FIELD	SCRIPT VARIABLE
Domain	Resource	$domain
Password	Password	$password
User Name	Username	$username

2. The template will need a **resource**,**target** or **Host** field to identify the location of the BluePrism server.

Step 3

1. Create a **dependency changer for BluePrism**. This remotely invokes AutomateC from BluePrism in order to set the credential property to the new password for the Secret.



Edit PowerShell Script

Name: BluePrism Dependency

Description: BluePrism Dependency Changer

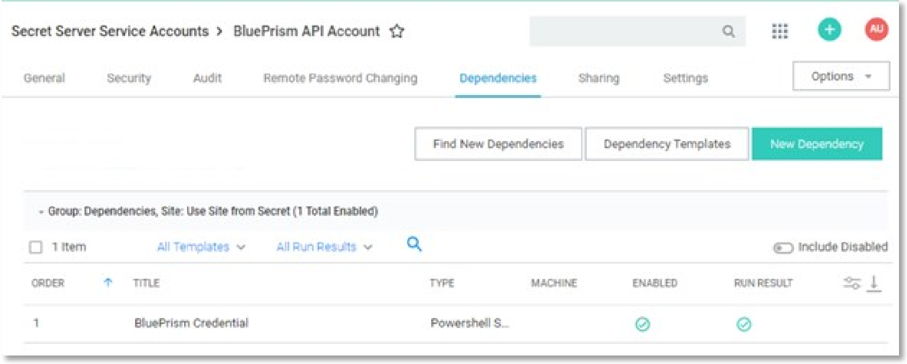
Category: Dependency

Script

```
$params = $args
$username = $params[0]
$newpassword = $params[1]
$Target = $params[2]
$bpuser = $params[3]
$bppass = $params[4]

$results = Invoke-Command -ComputerName $Target -ScriptBlock {
try
    {
    $cmdPath = 'C:\Program Files\Blue Prism Limited\Blue Prism Automate\automatec.ex
    $cmdArgList = @(
    "/user",$using:bpuser,$using:bppass
    "/setcredentialproperty",$using:username,"manage_password",$using:newpassword
    )
    return & $cmdPath $cmdArgList
    }
```

Dependency Script

```
$params = $args
$username = $params[0]
$newpassword = $params[1]
$Target = $params[2]
$bpuser = $params[3]
$bppass = $params[4]

$results = Invoke-Command -ComputerName $Target -ScriptBlock {
try
 {
 $cmdPath = 'C:\Program Files\Blue Prism Limited\Blue Prism Automate\automatec.exe'
 $cmdArgList = @(
 "/user",$using:bpuser,$using:bppass
 "/setcredentialproperty",$using:username,"manage_password",$using:newpassword
 )
     return & $cmdPath $cmdArgList
 }
 catch
 {
     Write-Debug $("Error Details: " + $_)
     return
 }
 else
 {
    Write-Debug $("Error Details: " + $_)
    return
 }
 }
 if($results -eq "Successfully set credential property")
 {
 Write-Debug "Succesful dependency update"
 return
 }
 else
 {
 throw $results
 }
```

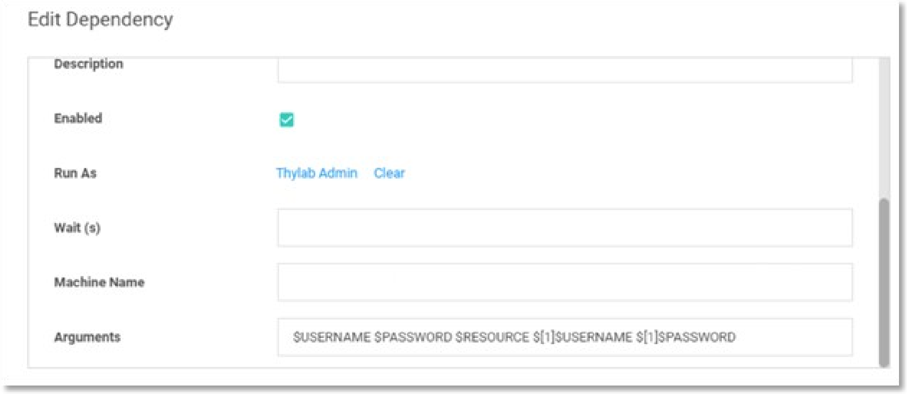


Group: Dependencies, Site: Use Site from Secret (1 Total Enabled)

□ 1 Item

ORDER	TITLE	TYPE	MACHINE	ENABLED	RUN RESULT
1	BluePrism Credential	Powershell S...		✓	✓

BluePrism Credential dependency settings are shown below.



Edit Dependency

Description	
Enabled	☑
Run As	Thylab Admin Clear
Wait (s)	
Machine Name	
Arguments	$USERNAME $PASSWORD $RESOURCE $[1]$USERNAME $[1]$PASSWORD

Note: A dependency will need to be created for each BluePrism server.

ConnectWise

Below are the following integrations that are available with ConnectWise:

- ConnectWise Control
- Connectwise Manage
- Connectwise Automate

ConnectWise Control Integrations with Secret Server

ConnectWise Control is a self-hosted remote desktop software application. ConnectWise Control has a proprietary protocol but does allow users to implement custom extensions. The Secret Server integrations is implemented as an extension which is accessible via the helper window. The on site ConnectWise Control server is self-hosted, and runs as a .NET Framework application. On Microsoft Windows it runs as a set of services. On Linux and OS X it runs as a daemon. ConnectWise Control launched a cloud platform in 2015, providing another way to use the ConnectWise Control software with no local server installation required.

The integration between Secret Server and ConnectWise Control ensures that:

- Passwords are securely vaulted in Secret Server.
- Users can connect to a Secret Server instance.
- Users can search and select folders.
- Users can search and select secrets located within a folder.
- Users can pass the userid/password (and domain) to the Connected Machine.
- All access by users is captures in Secret Server Audit Trail.

Integration Requirements

System Requirements

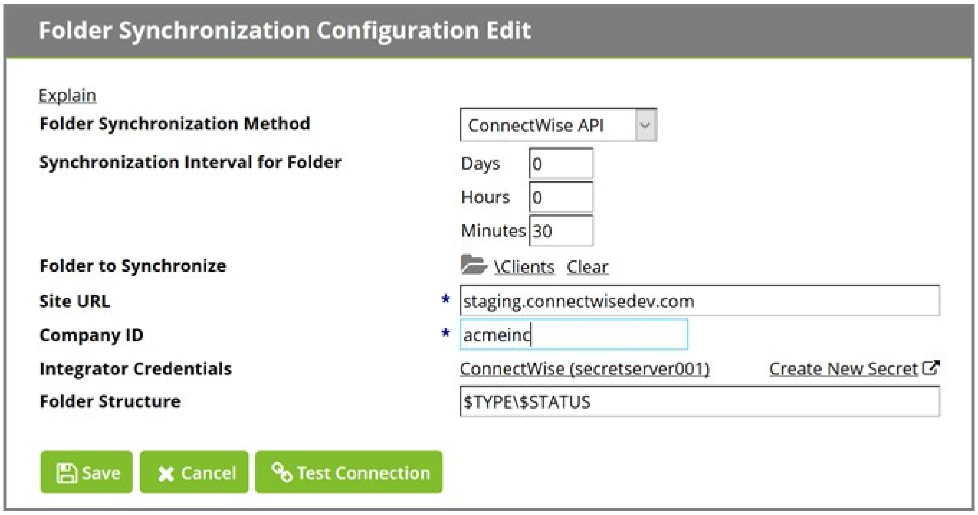
Product	Details	
ConnectWise Control		
	Version	ConnectWise Control Version 20.6.28494.7444
	Hardware	For further information about hardware requirements please contact ConnectWise.
Secret Server		
	Version	Thycotic Secret Server 10.9 or later
	Hardware	For further information about system requirements for Secret Server please visit: https://docs.thycotic.com/ss/10.8.0/secret-server-setup/system-requirements/index.md.

Pre-requisites and Limitations

- SAML connectivity is not currently supported.
- Thycotic's Helper Extension requires an on-prem installation of ConnectWise.
- Installation requires access to ControlWise' Developer Extension.

Setting up Folder Synchronization

The following image shows the ConnectWise synchronization view in Secret Server.



Folder Synchronization Configuration Edit

Explain

Folder Synchronization Method	ConnectWise API
Synchronization Interval for Folder	Days 0 Hours 0 Minutes 30
Folder to Synchronize	\Clients Clear
Site URL *	staging.connectwisedev.com
Company ID *	acmeinc
Integrator Credentials	ConnectWise (secretserver001) Create New Secret
Folder Structure	$TYPE\$STATUS

To set up this feature, click **Folder Synchronization** from the **Administration** menu.

> **Note**: For the ability to edit these settings, an administrator must be assigned a role within Secret Server containing the Administer ConnectWise Integration permission.

Enabling Folder Synchronization will require specifying the synchronization interval in days, hours, and minutes. Folder to Synchronize references the parent folder where Secret Server will create the ConnectWise companies as child folders.

Creating the ConnectWise Integrator Login

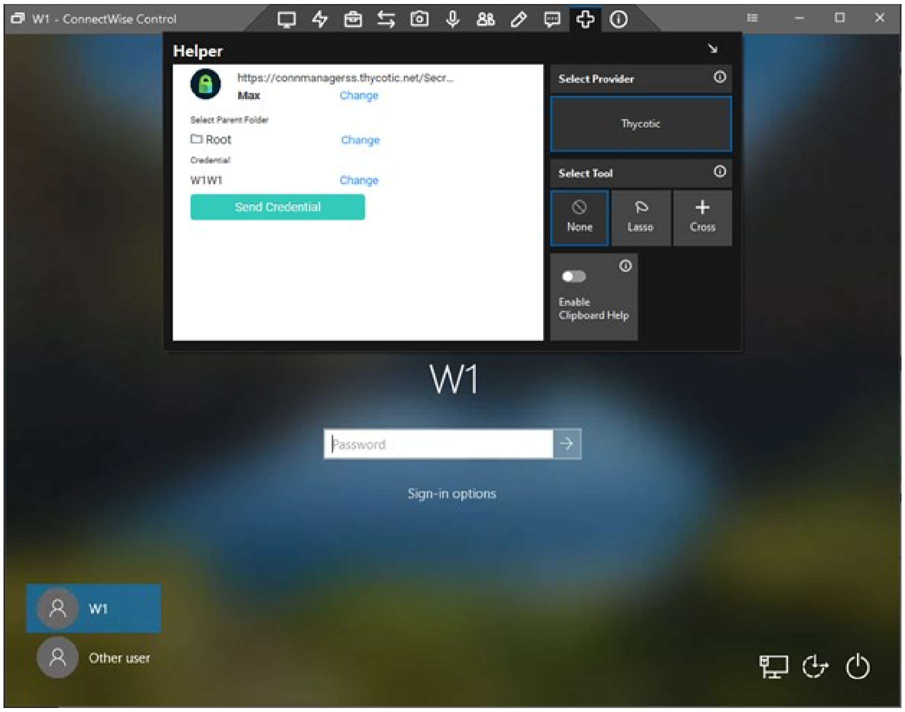
If an integrator login doesn't exist in ConnectWise, create one by:

- Entering in the hostname of your ConnectWise instance
- Enter your company id
- Select the Secret of the ConnectWise integrator login

Configuration

Installing Thycotic Helper Extension (beta) for ConnectWise Control

Thycotic has developed an extension for ConnectWise Control. The extension is located inside the Helper Window of the the ScreenConnect client.



W1 - ConnectWise Control

Helper

https://connmanagerss.thycotic.net/Secr...
Max

Select Parent Folder
Root

Credential
W1W1

Select Provider: Thycotic

Select Tool: None

The extension provides the following features:

- Users can connect to a Secret Server instance.
- Users can search and select folders.
- Users can search and select secrets located within a folder.
- Users can pass the userid/password (and domain) to the Connected Machine.

Please see the following sections in order to configure Connectwise Control.

- Installing Thycotics Helper Extension
- Install the Developer Extension
- Create an Extension
- Edit an Extension
- Using Thycotics Helper Extension

Installing Thycotic's Helper Extension

The installation instructions below are largely derived from ConnectWise University' Extension Development Guide.

What's the Process?

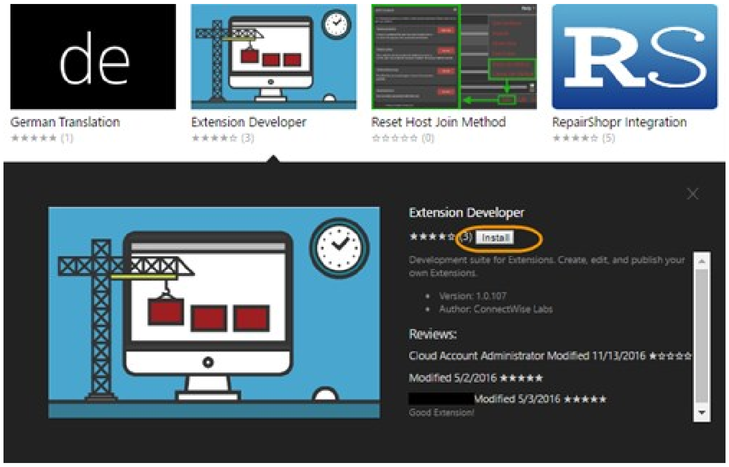
To create an extension with ConnectWise Control, you'll have to follow this process:

1. **Get a developer instance**. A developer instance has a special license that allows you to develop and test your extensions.
2. **Create your extension** and test it in your developer instance. Download the Extension Developer extension to get started.

Install the Developer Extension

Before you can create an extension, you will need to download the Extension Developer extension. This extension contains the tools necessary to create, edit, and publish extensions.

1. Navigate to the **Administration l Extensions page**.
2. Click the **Browse Online Extensions links**.
3. Click **Browse Online Extensions to open the Extension Marketplace**.
4. Install the **Extension Developer**.
5. To find the Extension Developer extension, Click the **thumbnail** and then click the **Install button**.



Development suite for Extensions. Create, edit, and publish your own Extensions.

- Version: 1.0.107
- Author: ConnectWise Labs

Reviews:

Cloud Account Administrator Modified 11/13/2016 ★☆☆☆☆

Modified 5/2/2016 ★★★★★

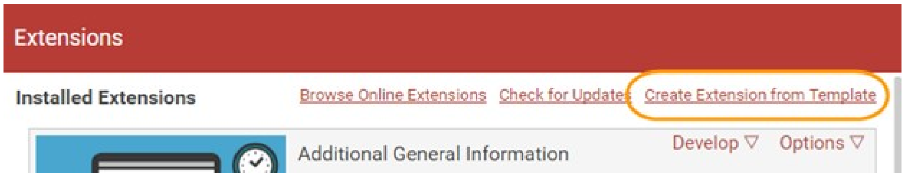
[redacted] Modified 5/3/2016 ★★★★★

Good Extension!

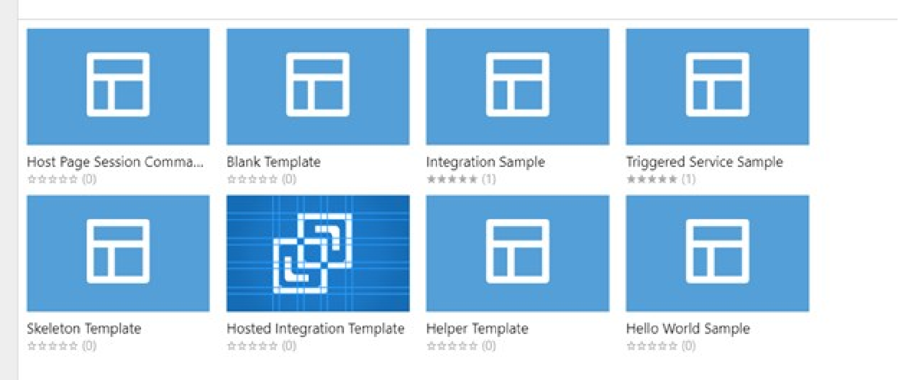
Note: The Extension Developer will add a new Develop tab to your existing extensions.

Create an Extension

1. Open the **Extensions page**.
2. Click **Create Extension from Template**.



3. Click the **Helper Template**.
4. Click **Clone | Install**.
5. Your cloned extension will be automatically installed and listed on your Extensions page.

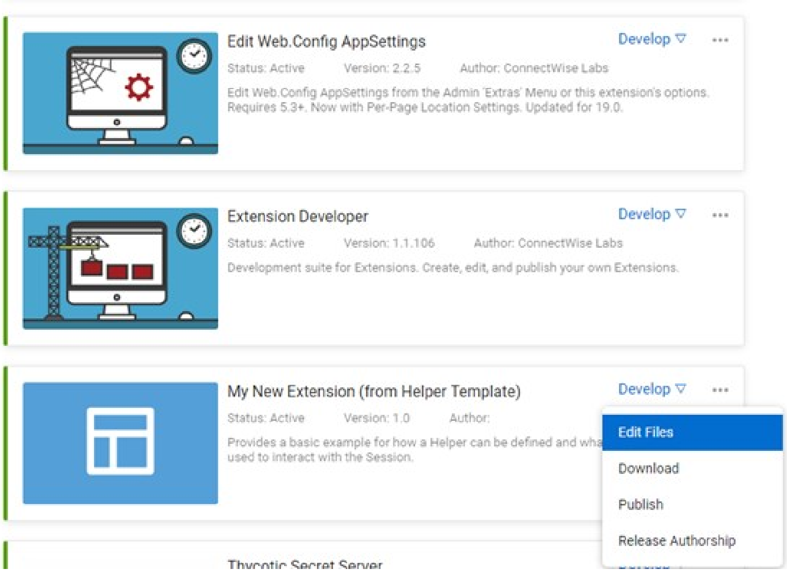


Edit an Extension

From within ConnectWise Control

The ConnectWise Control extension editor is an embedded version of the Ace code editor. View the Ace website for the complete details on the features included in the editor .

1. Navigate to the **Extensions page**.
2. Click **Develop**.
3. Click **Edit Files**.



Edit Web.Config AppSettings

Status: Active Version: 2.2.5 Author: ConnectWise Labs

Edit Web.Config AppSettings from the Admin 'Extras' Menu or this extension's options. Requires 5.3+. Now with Per-Page Location Settings. Updated for 19.0.

Extension Developer

Status: Active Version: 1.1.106 Author: ConnectWise Labs

Development suite for Extensions. Create, edit, and publish your own Extensions.

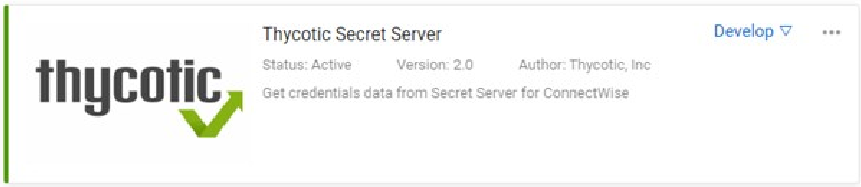
My New Extension (from Helper Template)

Status: Active Version: 1.0 Author:

Provides a basic example for how a Helper can be defined and wha used to interact with the Session.

Thycotic Secret Server

4. Import the files provided from Thycotic using the **Upload** option.
5. The extension should look like the image below.



Thycotic Secret Server

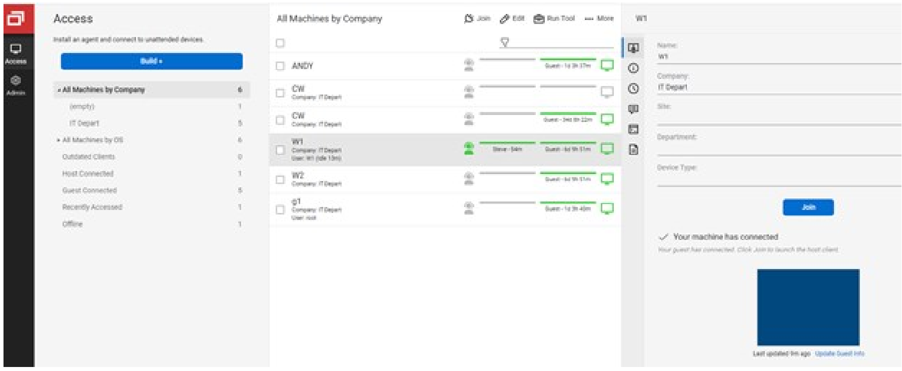
Status: Active Version: 2.0 Author: Thycotic, Inc

Get credentials data from Secret Server for ConnectWise

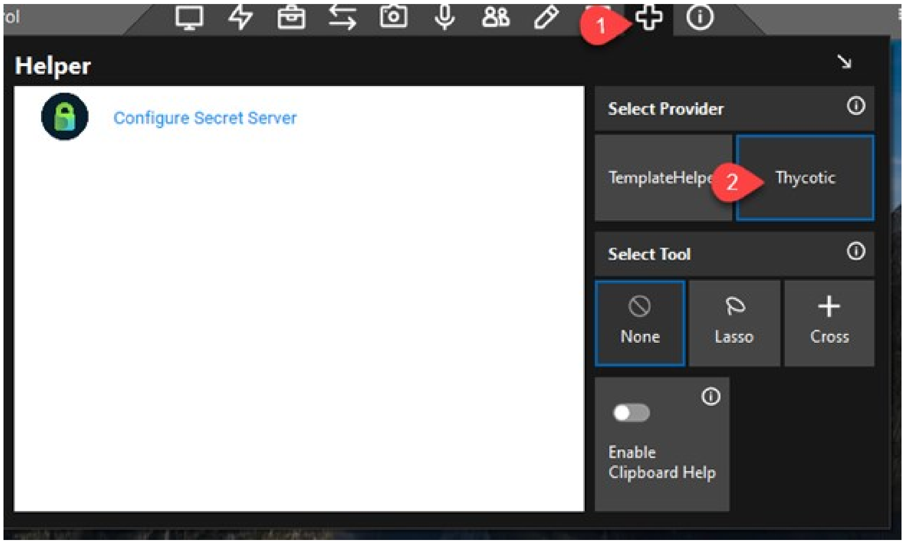
Using Thycotic's Helper Extension

After installation, Thycotic's extension will be available from the Helper window. The instruction below provide a typical use case for logging into a machine, but can be extended to any other resource available from the ScreenControl Client.

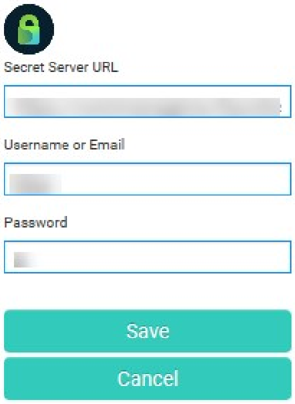
1. Connect to the machine from the **ControlWise Control window**.



2. Once connected, click on the **Helper icon** and **Thycotic Provider** (if not already selected).



3. Click on **Configure Secret Server** (if not already connected) and enter the connection parameters.



Secret Server URL

[redacted]

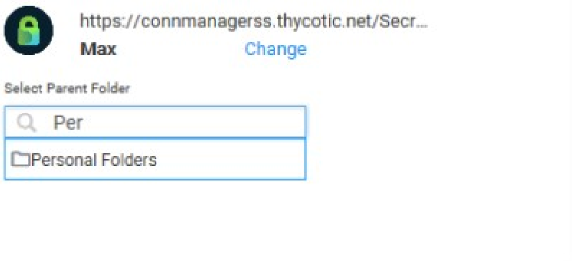
Username or Email

[redacted]

Password

[redacted]

4. In the **Select Parent Folder** field, type in the full or partial folder name and click on **Enter** to Search. You can select a folder by clicking on it.



https://connmanagerss.thycotic.net/Secr...

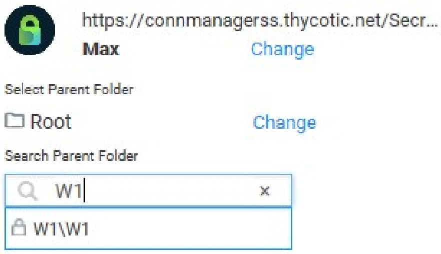
Max

Select Parent Folder

Per

Personal Folders

5. In the **Search Parent Folder** field enter a full or partial secret name and click **Enter** to Search. You can select a secret by clicking on it.



https://connmanagerss.thycotic.net/Secr...

Max

Select Parent Folder

Root

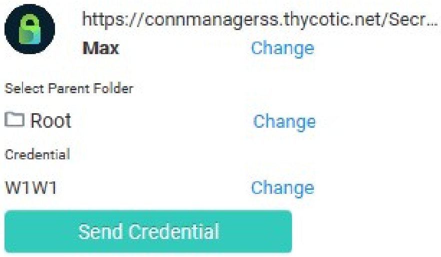
Search Parent Folder

W1

W1\W1

Note: The search for Secrets is limited to the selected folder unless the Root folder is selected. In that case the search will look for a Secret in the entire system.

6. Click the **Send Credential** button to push the **Domain/User ID** and **Password** to the selected connected Computer.



https://connmanagerss.thycotic.net/Secr...

Max

Select Parent Folder

Root

Credential

W1W1

Connectwise Manage

> **Important**: This integration is scheduled to go through a new verification cycle. The documentation provided for this integration is based on older versions of Secret Server and ConnectWise Manage.

Managing Customer Privileged Accounts

Integrating the Secret Server folder structure with ConnectWise company information allows companies to quickly organize their privileged accounts based on the customers they are managing in ConnectWise. This integration helps eliminate the need to manually update the same data in two different systems.

The Secret Server Approach to Privileged Account Management

Many environments with strict Information Security policies require methods to control and monitor access to privileged accounts. Companies often apply security policies such as physical access restrictions to hardware, network firewalls, appropriate-use guidelines, and user account restrictions. In the case of privileged accounts, access is more difficult to track and verify. Implementing Secret Server enables organizations to strictly control and track access to both customer and internal privileged accounts.

Secret Server ConnectWise Integration Explained

Secret Server has the ability to read the list of companies via the ConnectWise API and populate Folders based on Companies and Status.

Setting up Folder Synchronization

> **Note**: Integration to be verified with new product versions.

Figure 1 shows the ConnectWise synchronization view in Secret Server. To set up this feature, click **Folder Synchronization** from the **Administration** menu.

> **Note**: For the ability to edit these settings, an administrator must be assigned a role within Secret Server containing the Administer ConnectWise Integration permission.

Enabling Folder Synchronization will require specifying the synchronization interval in days, hours, and minutes. Folder to Synchronize references the parent folder where Secret Server will create the ConnectWise companies as child folders.

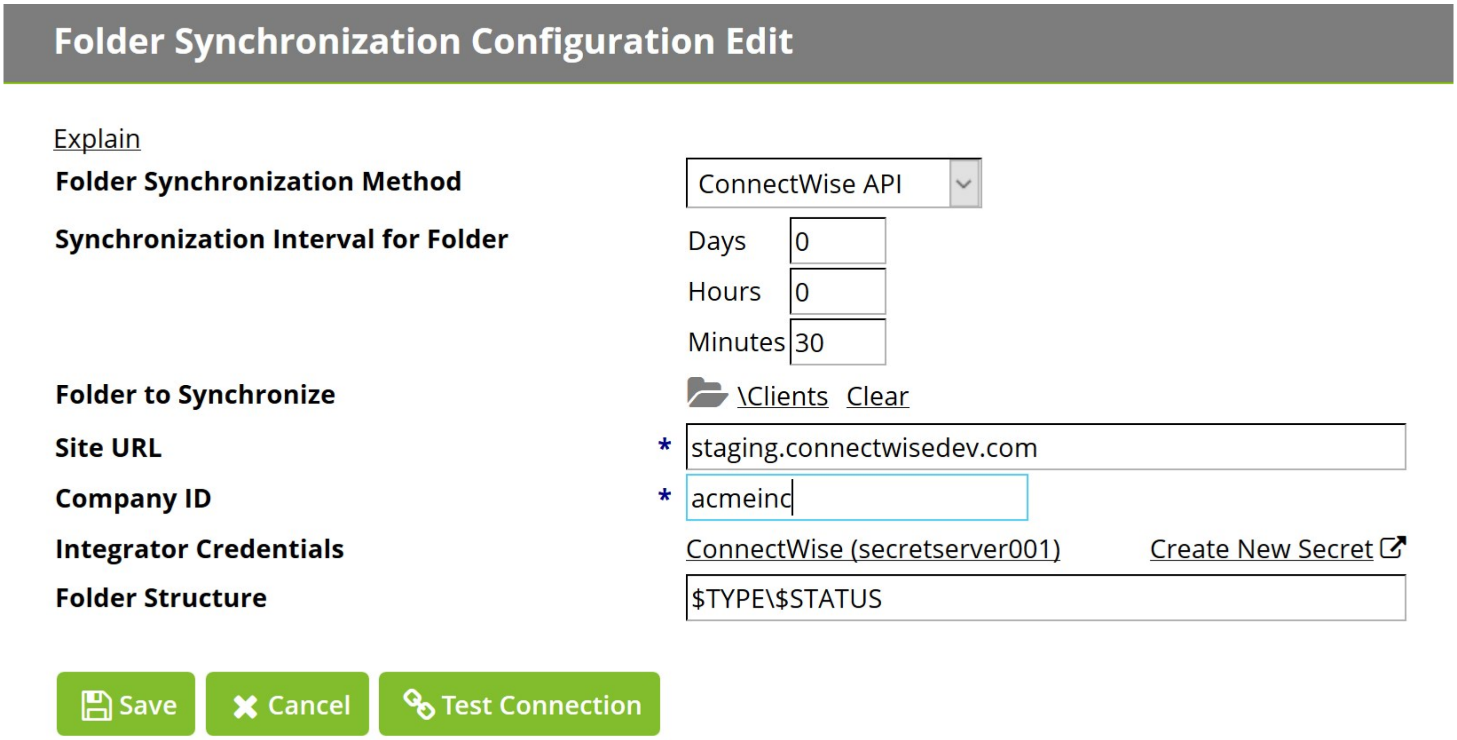
If an integrator login doesn't exist in ConnectWise, create one by following the steps in Creating the ConnectWise Integrator Login.

Enter the hostname of your ConnectWise instance, your company id, and select the Secret of the ConnectWise integrator login.

Figure 2 shows the result of synchronization with ConnectWise. A folder has been created in Secret Server for each company and placed in the folder tree based on the company type, status, and name in ConnectWise.

Managing Synchronized Folders

During synchronization, the folders are created in the Secret Server database. Therefore, if at any point the ConnectWise synchronization is turned off, the synchronized folders will remain in Secret Server. To delete a synchronized folder, turn off folder synchronization and delete the folder in the same way as a standard Secret Server folder.



Folder Synchronization Configuration Edit

Explain

Folder Synchronization Method: ConnectWise API

Synchronization Interval for Folder: Days 0, Hours 0, Minutes 30

Folder to Synchronize: \Clients Clear

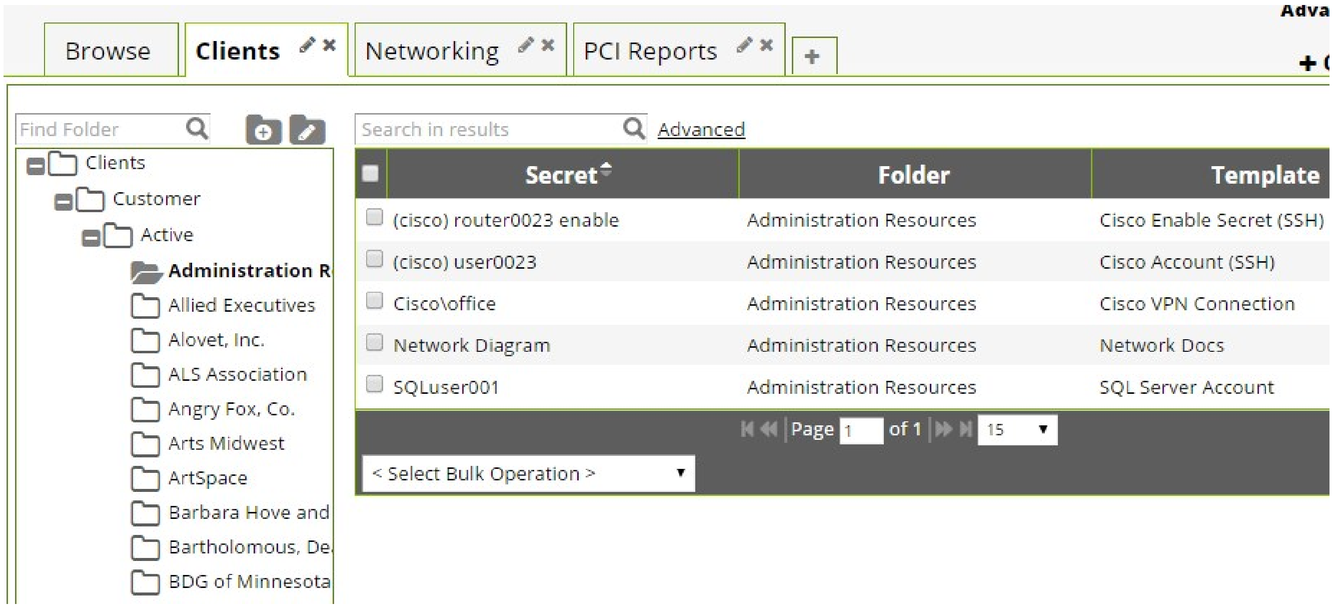
Site URL *: staging.connectwisedev.com

Company ID *: acmeinc

Integrator Credentials: ConnectWise (secretserver001) Create New Secret

Folder Structure: $TYPE\$STATUS

Figure 1 ConnectWise Synchronization



Clients
Customer
Active
Administration R
Allied Executives
Alovet, Inc.
ALS Association
Angry Fox, Co.
Arts Midwest
ArtSpace
Barbara Hove and
Bartholomous, De
BDG of Minnesota

□	Secret	Folder	Template
□	(cisco) router0023 enable	Administration Resources	Cisco Enable Secret (SSH)
□	(cisco) user0023	Administration Resources	Cisco Account (SSH)
□	Cisco\office	Administration Resources	Cisco VPN Connection
□	Network Diagram	Administration Resources	Network Docs
□	SQLuser001	Administration Resources	SQL Server Account

Page 1 of 1 15

< Select Bulk Operation >

Figure 2 Synchronized Folders

Creating Company Links in ConnectWise

ConnectWise supports creation of links to Secret Server folders using the company name as a dynamic parameter. This can be used to build a link from within ConnectWise that will instantly open Secret Server to view the stored Secrets for the selected company.

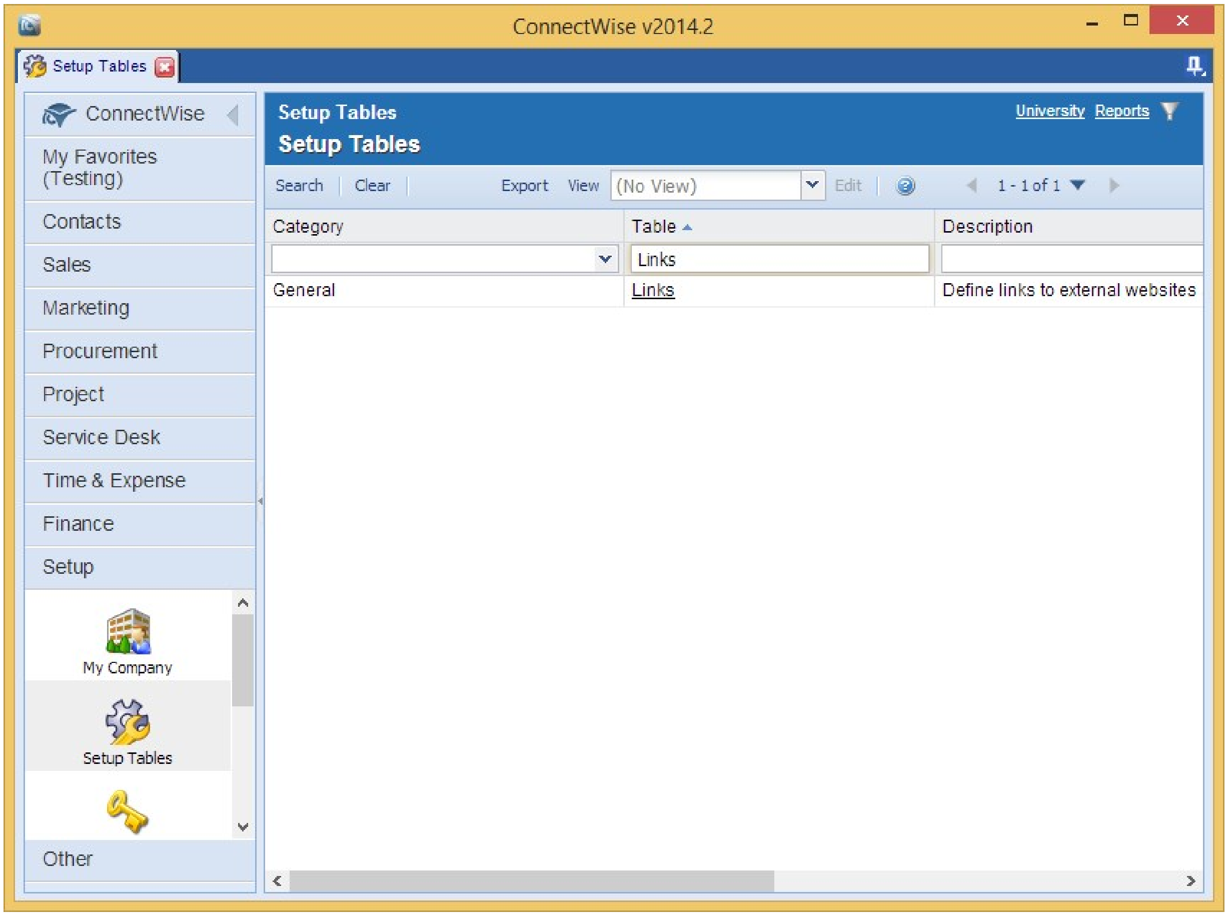
To create a new link in ConnectWise, click the **Links** table from the **Setup Tables** menu (*Figure 3*).

Next, create a link to the Secret Server dashboard and specify the company name as the folder parameter in the query string. Your URL should look something like:

https://yoursecretserverurl/dashboard.aspx?folder=Clients\Customer\Active\[companyname]

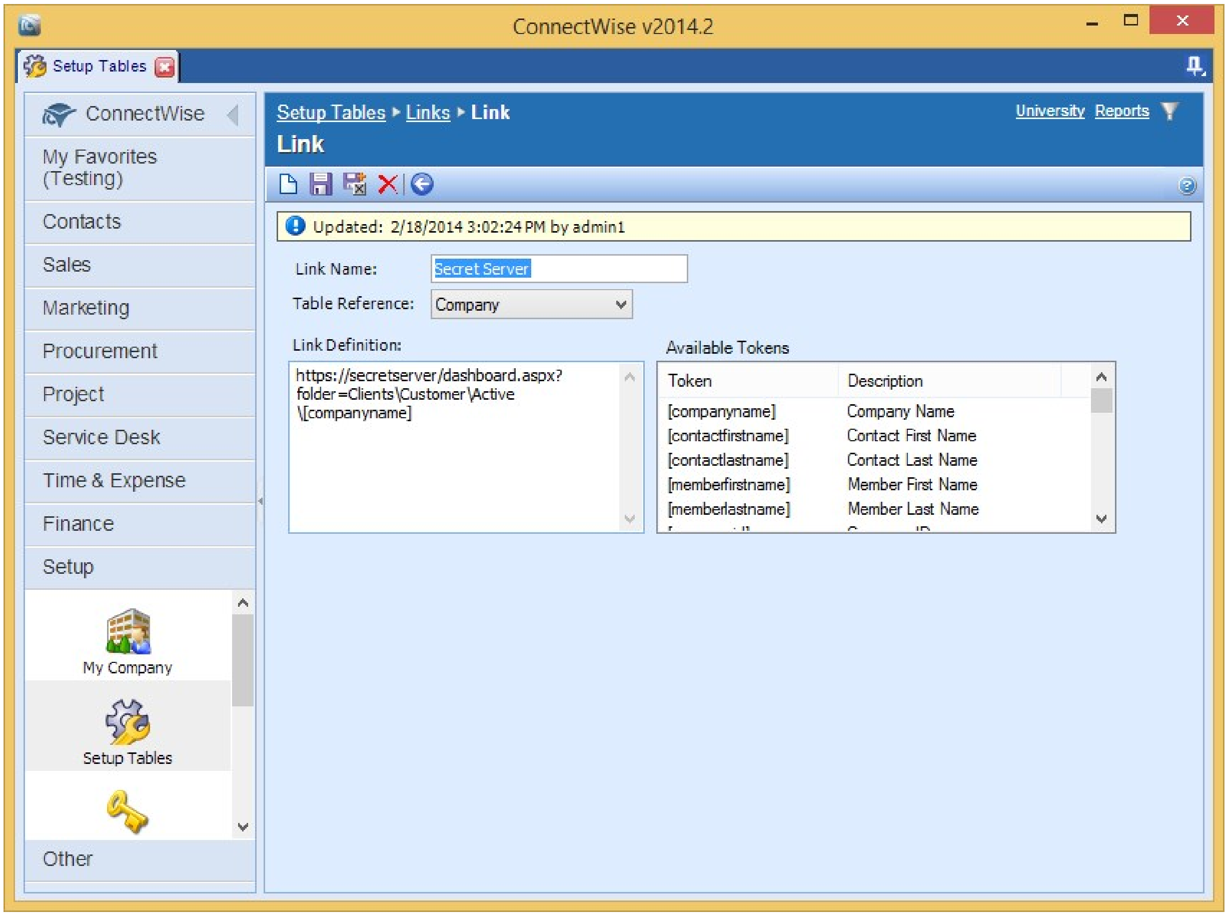
Note Because ConnectWise does not use a token for the status or type of company, those parts of the folder path will need to be hardcoded, and multiple links created.

The new link will now be available on the company information screen (*Figure 5*). Select it from the drop-down menu to automatically navigate to a search for that company's information on the Secret Server dashboard (*Figure 6*).



Category	Table	Description
	Links	
General	Links	Define links to external websites

Figure 3 Links Table



Updated: 2/18/2014 3:02:24 PM by admin1

Link Name: Secret Server

Table Reference: Company

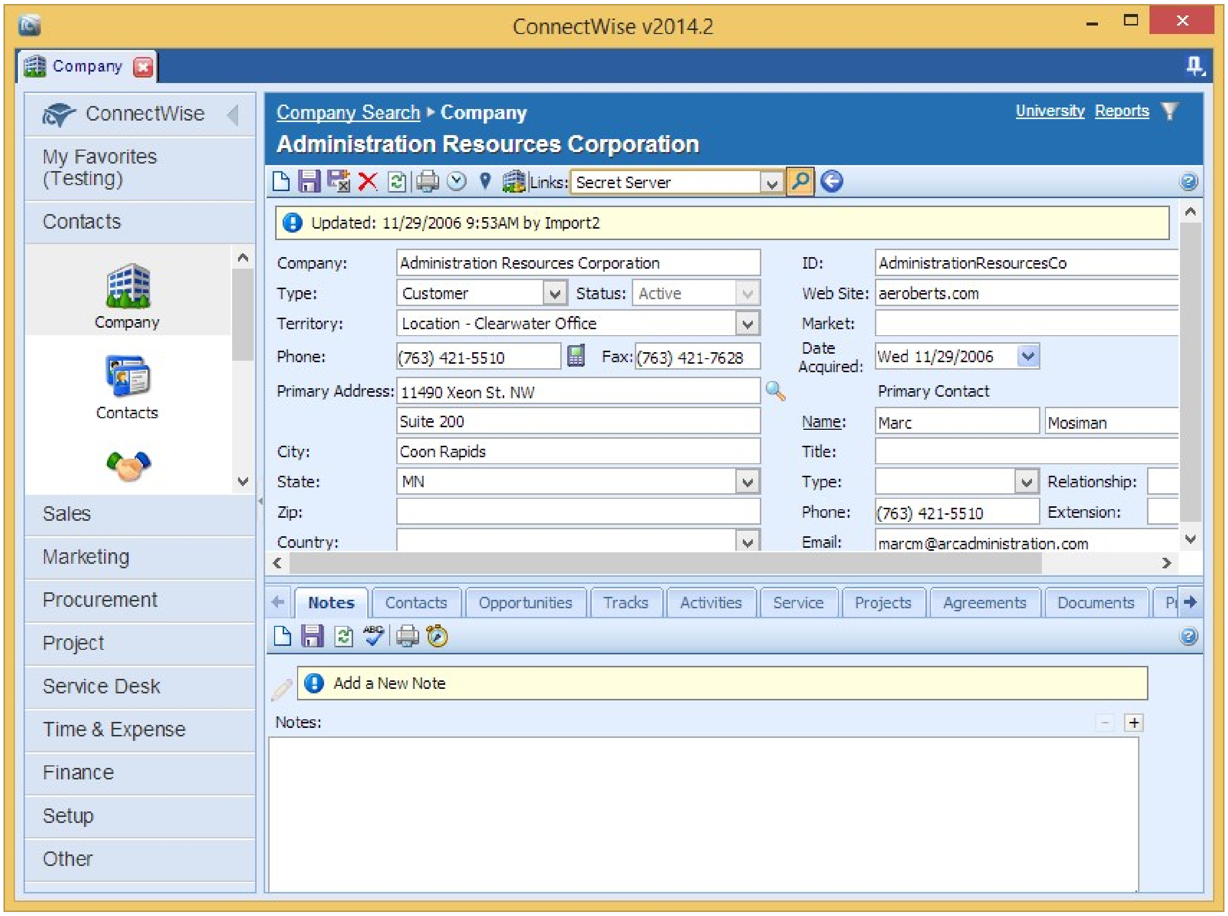
Link Definition:

```
https://secretserver/dashboard.aspx?
folder=Clients\Customer\Active
\[companyname]
```

Available Tokens

Token	Description
[companyname]	Company Name
[contactfirstname]	Contact First Name
[contactlastname]	Contact Last Name
[memberfirstname]	Member First Name
[memberlastname]	Member Last Name

Figure 4 Link Configuration



Administration Resources Corporation

Updated: 11/29/2006 9:53AM by Import2

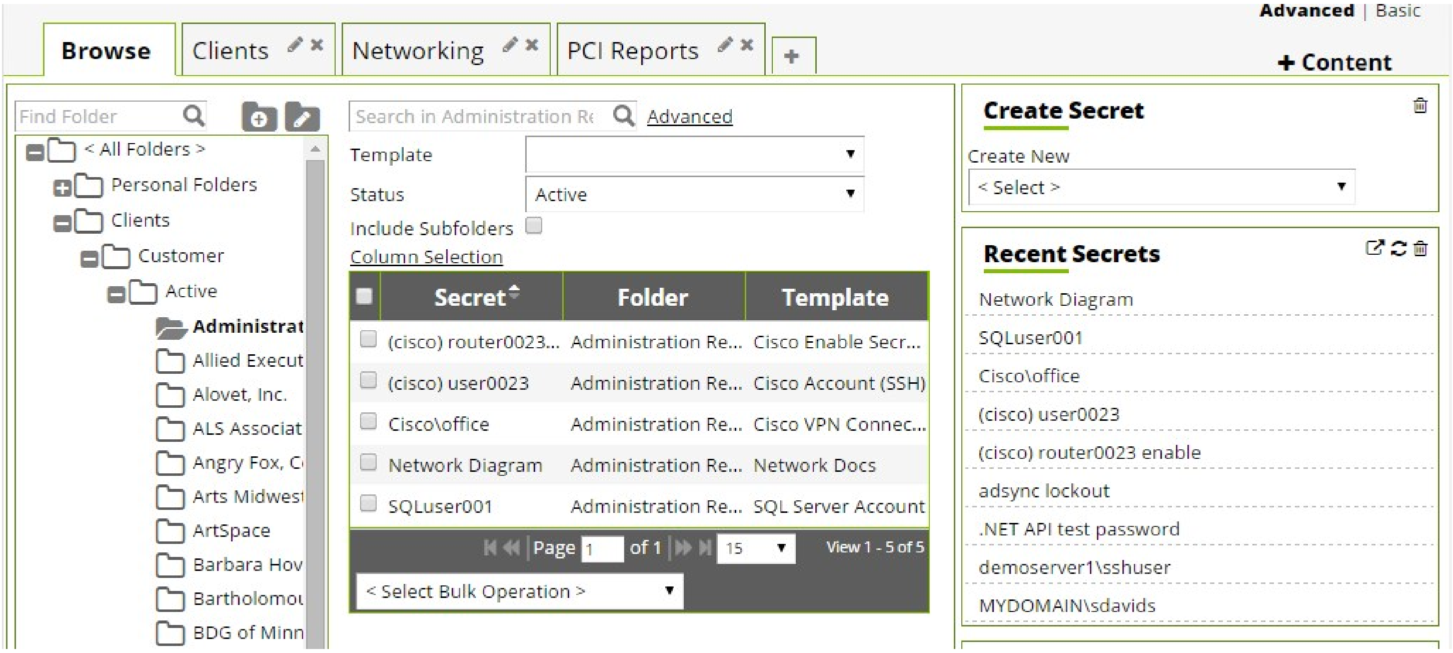
Company:	Administration Resources Corporation	ID:	AdministrationResourcesCo
Type:	Customer	Web Site:	aeroberts.com
Status:	Active	Market:	
Territory:	Location - Clearwater Office	Date Acquired:	Wed 11/29/2006
Phone:	(763) 421-5510	Primary Contact	
Fax:	(763) 421-7628	Name:	Marc Mosiman
Primary Address:	11490 Xeon St. NW	Title:	
	Suite 200	Type:	
City:	Coon Rapids	Relationship:	
State:	MN	Phone:	(763) 421-5510
Zip:		Extension:	
Country:		Email:	marcm@arcadministration.com

Links: Secret Server

Add a New Note

Notes:

Figure 5 Company Link to Secret Server



Template:

Status: Active

Include Subfolders ☐

☐	Secret	Folder	Template
☐	(cisco) router0023...	Administration Re...	Cisco Enable Secr...
☐	(cisco) user0023	Administration Re...	Cisco Account (SSH)
☐	Cisco\office	Administration Re...	Cisco VPN Connec...
☐	Network Diagram	Administration Re...	Network Docs
☐	SQLuser001	Administration Re...	SQL Server Account

Create Secret

Create New: < Select >

Recent Secrets

- Network Diagram
- SQLuser001
- Cisco\office
- (cisco) user0023
- (cisco) router0023 enable
- adsync lockout
- .NET API test password
- demoserver1\sshuser
- MYDOMAIN\sdavids

Figure 6 Client Secrets from ConnectWise Link

Connectwise Automate

> **Important**: This integration is scheduled to go through a new verification cycle. The documentation provided for this integration is based on older versions of Secret Server and ConnectWise Manage.

The Secret Server Approach

Many environments with strict information security policies require methods to control and monitor access to privileged accounts. Companies often apply security policies, such as physical access restrictions to hardware, network firewalls, appropriate-use guidelines, and user account restrictions. In the case of privileged accounts, access is more difficult to track and verify. Implementing Secret Server enables organizations to strictly control and track access to both customer and internal privileged accounts.

Secret Server Automate (LabTech) Plugin Explained

The Secret Server plugin for Automate (LabTech) gives the Automate (LabTech) user a view into any stored passwords for a client or particular computer. When a user goes to the Secret Server tab in Automate (LabTech), they will automatically be taken to a list of corresponding passwords for that client or computer in the Secret Server.

Managing Customer Privileged Accounts

Note: Integration to be verified with new product versions.

Managing Folders

Secret Server has a folder structure that needs to correspond to the client names in Automate (LabTech). When a client is selected, the plugin automatically searches Secret Server's folders for a folder name matching the client name and selects it, showing all Secrets stored in the folder for that client.

Figures 1 & 2 show a folder structure in Secret Server that corresponds to the Automate (LabTech) client structure.

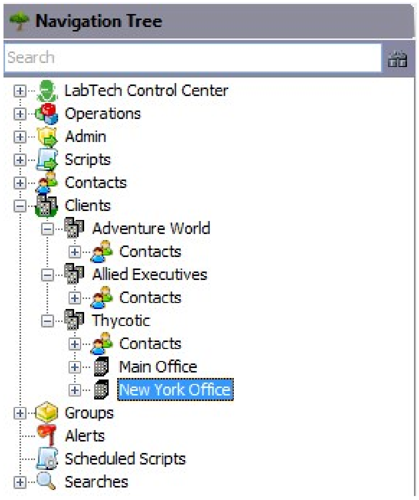


Figure 1 Automate (LabTech) Clients

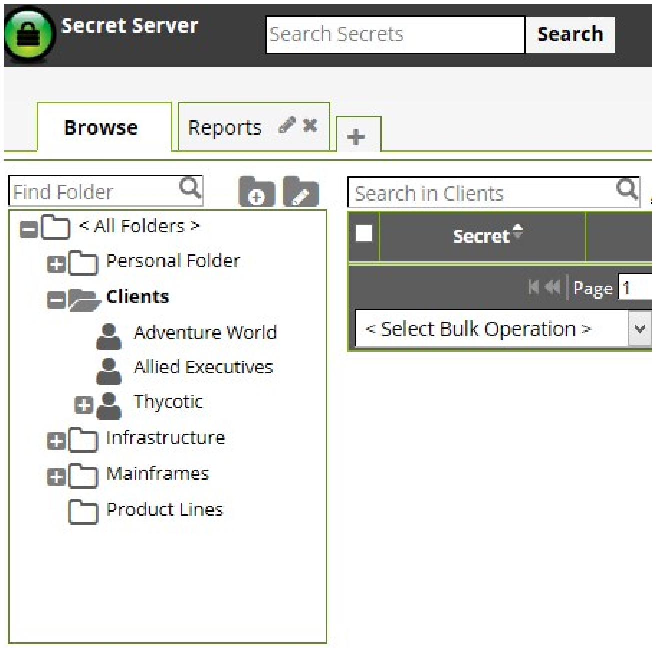
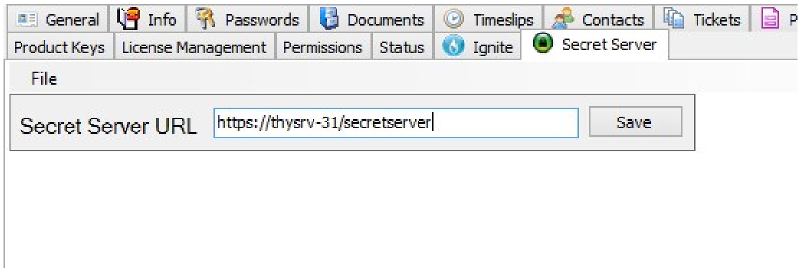


Figure 2 Secret Server Folders

Using the Plugin

When the plugin is installed, you will need to specify your Secret Server URL by navigating to any client and clicking **File > Configure URL** as a super admin. Once it is set on one client all clients will read the same configuration information.



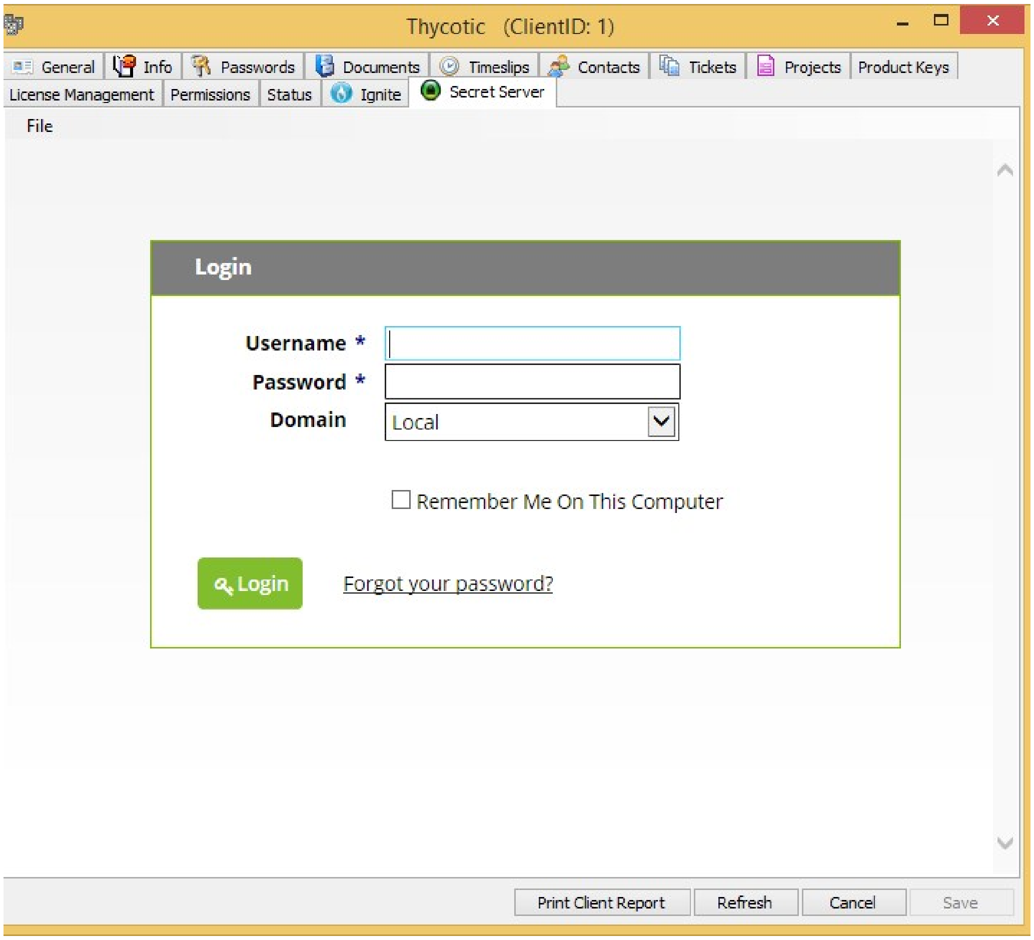
Secret Server URL https://thysrv-31/secretserver

Figure 3 Setting the Secret Server URL

Your URL should look something like: *https://servername/secretserver* or *https://yoursecretserverurl*

> **Note**: The URL should not include any page names, such as index.html or login.aspx. The plugin will navigate to the correct page once saved. If the URL is incorrect, click **File > Configure URL** again to reset it.

Once the URL is saved, the Secret Server login page should appear and you can enter in your Secret Server credentials. Your Secret Server login may be different than your Automate (LabTech) login.

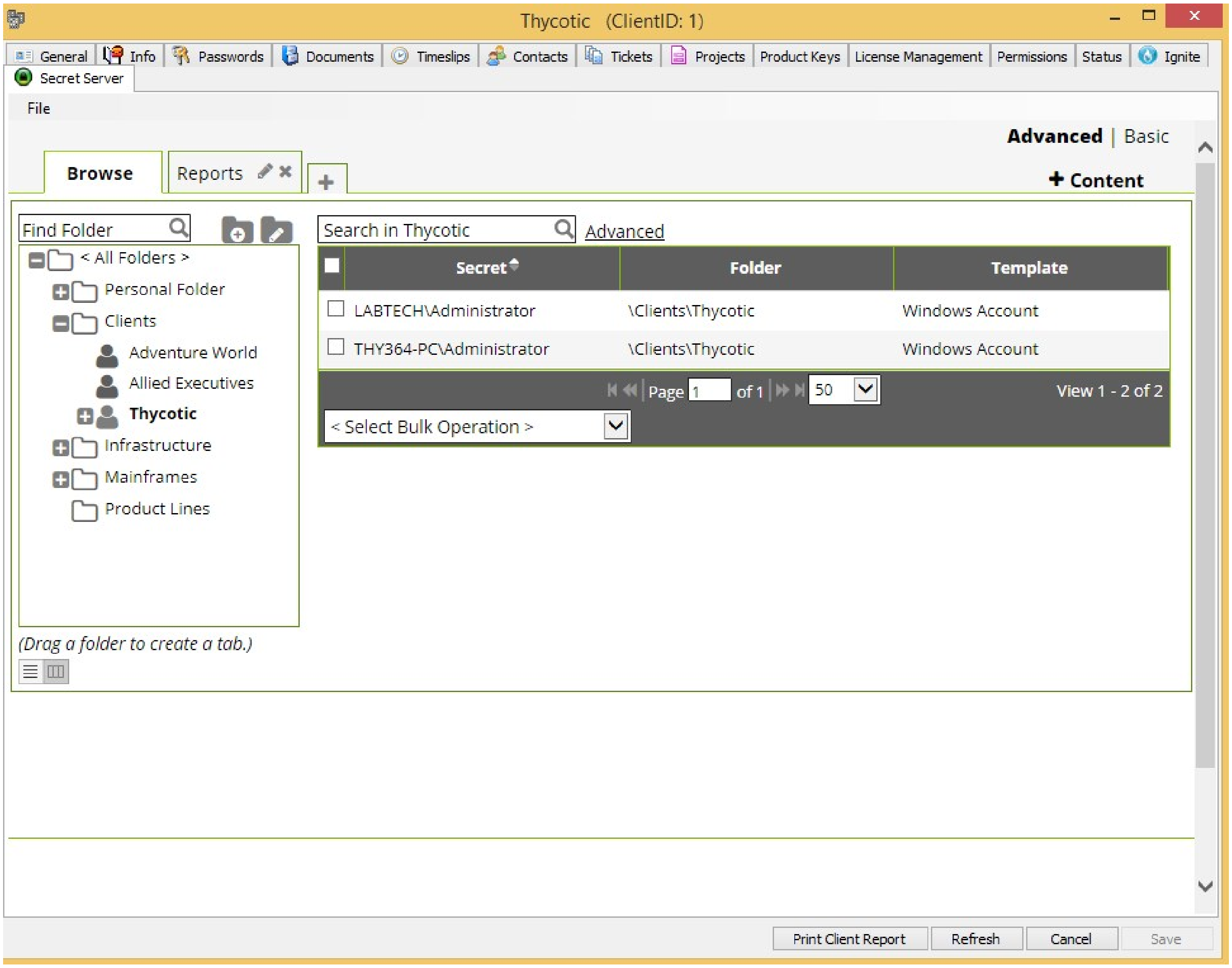


Domain: Local

☐ Remember Me On This Computer

Figure 4 Logging into Secret Server

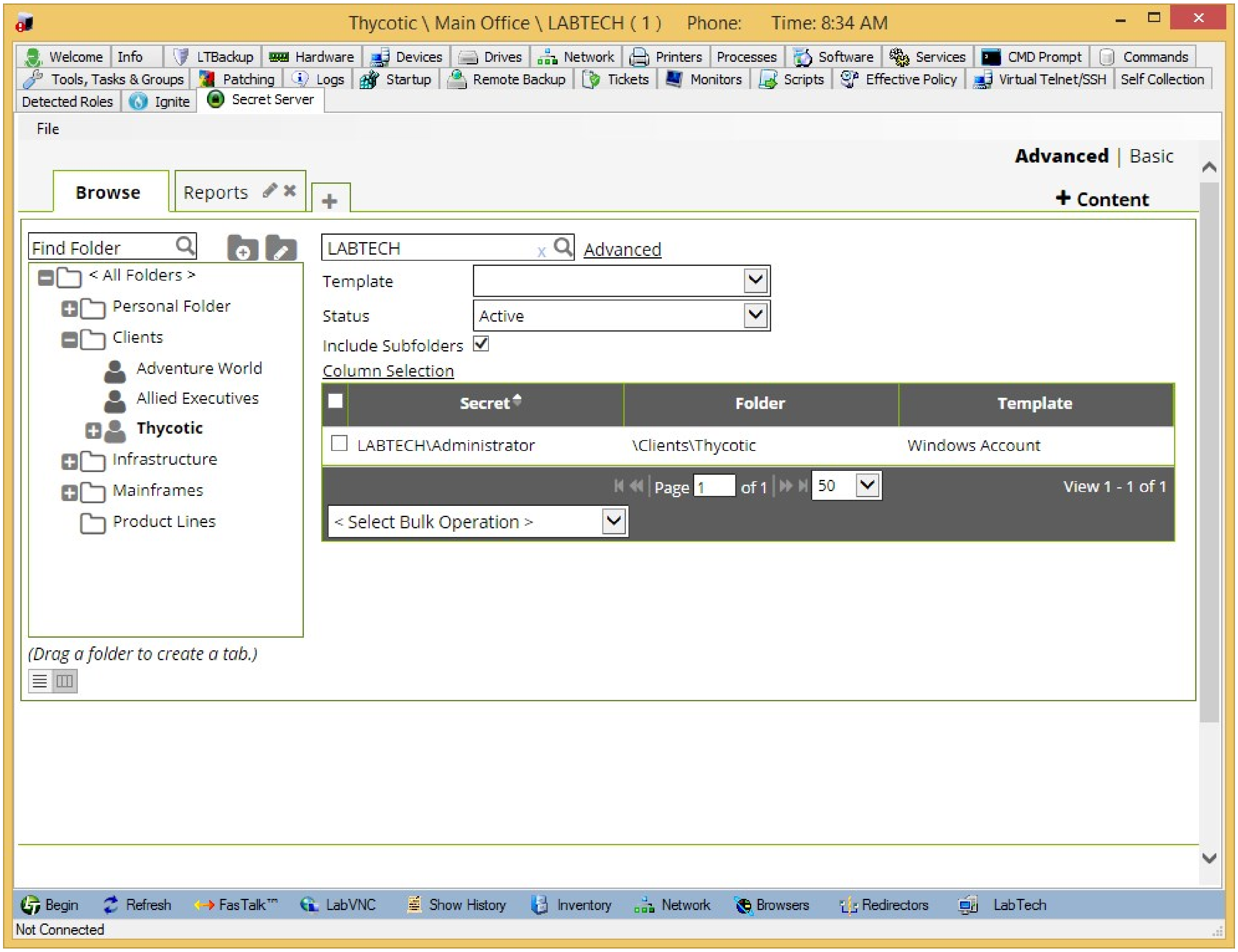
Once you're logged in, the client's folder will be automatically selected and you can view any passwords stored for that client.



□	Secret	Folder	Template
□	LABTECH\Administrator	\Clients\Thycotic	Windows Account
□	THY364-PC\Administrator	\Clients\Thycotic	Windows Account

Figure 5 Company Link to Secret Server

If you open the Secret Server plugin from a computer, that computer name will be set as a search term inside the folder to automatically filter to any credentials stored specifically for that machine.



LABTECH

Template:

Status: Active

Include Subfolders ☑

□	Secret	Folder	Template
□	LABTECH\Administrator	\Clients\Thycotic	Windows Account

50

Figure 6 Filtering to a Computer

Synchronizing Folders

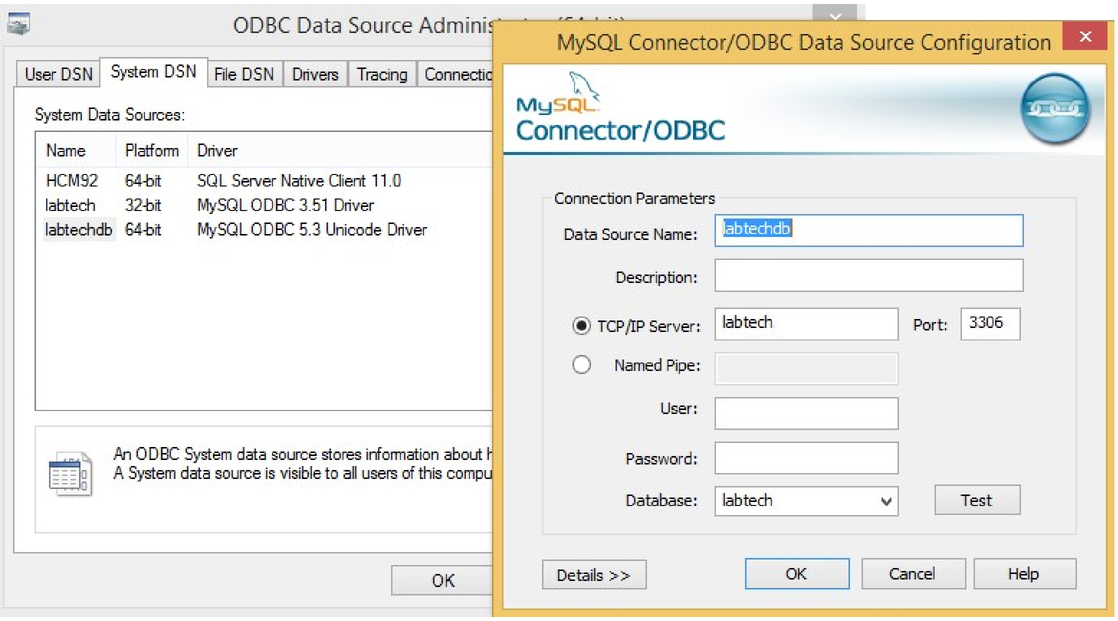
Secret Server can automatically create a Folder hierarchy from a custom database view. Note that this is optional and you can test the plugin by simply manually creating a Folder in Secret Server to correspond to a Automate (LabTech) client.

To automatically create Folders that correspond to Automate (LabTech) Clients you will need to do the following:

1. Create an ODBC system DSN on the Secret Server that points to the Automate (LabTech) database using the MySQL ODBC Driver.
2. Create a linked server in SQL Server.
3. Create a custom view in the SQL Server Database.
4. Turn on Folder Synchronization in Secret Server.

Step 1 - Create an ODBC DSN

1. On the Secret Server machine go to **Administrative Tools I ODBC Data Source (64 bit)**.



System Data Sources:

Name	Platform	Driver
HCM92	64-bit	SQL Server Native Client 11.0
labtech	32-bit	MySQL ODBC 3.51 Driver
labtechdb	64-bit	MySQL ODBC 5.3 Unicode Driver

An ODBC System data source stores information about h
A System data source is visible to all users of this compu

Connection Parameters

Data Source Name: labtechdb

Description:

● TCP/IP Server: labtech   Port: 3306

○ Named Pipe:

User:

Password:

Database: labtech

Figure 7 Configuring an ODBC MySQL System DSN

2. Create a new MySQL System DSN and enter in the information for connecting to the Automate (LabTech) database. If the MySQL ODBC driver is not installed, you will need to download it from Oracle and install it.

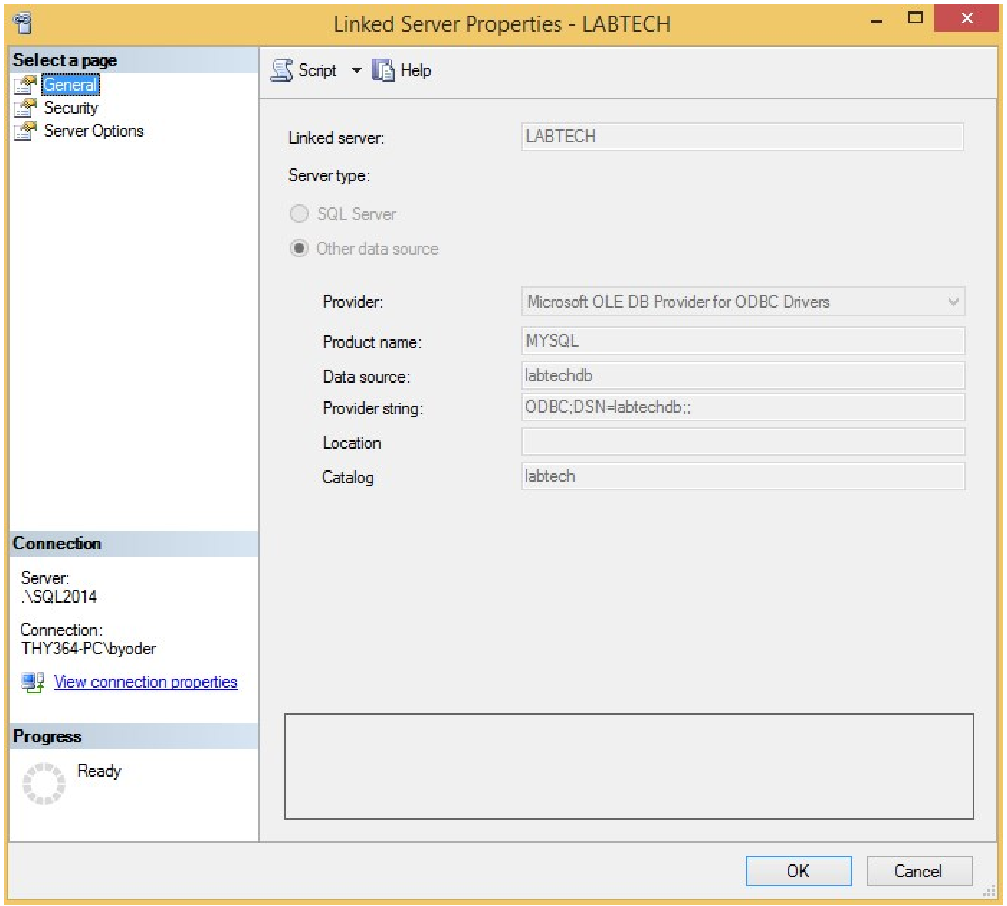
Step 2 - Create a Linked Server in SQL

1. Connect to the SQL Server instance where Secret Server is installed.

2. Under **Server Objects > Linked Servers** right click and add a new linked server.

3. Configure the linked server to reference the System DSN that was set up. Your settings should look like:

   - **Provider:** Microsoft OLE DB Provider for ODBC Drivers
   - **Product name:** MYSQL
   - **Data source:** The ODBC System DSN Name, in this case **Automate (LabTech)db**
   - **Provider string:** ODBC provider with the DSN specified. E.g.
   - **ODBC;DSN=Automate (LabTech)db;;**
   - **Location:** Optional
   - **Catalog:** The Automate (LabTech) database name



Linked server: LABTECH

Server type:

○ SQL Server

● Other data source

Provider: Microsoft OLE DB Provider for ODBC Drivers

Product name: MYSQL

Data source: labtechdb

Provider string: ODBC;DSN=labtechdb;;

Location:

Catalog: labtech

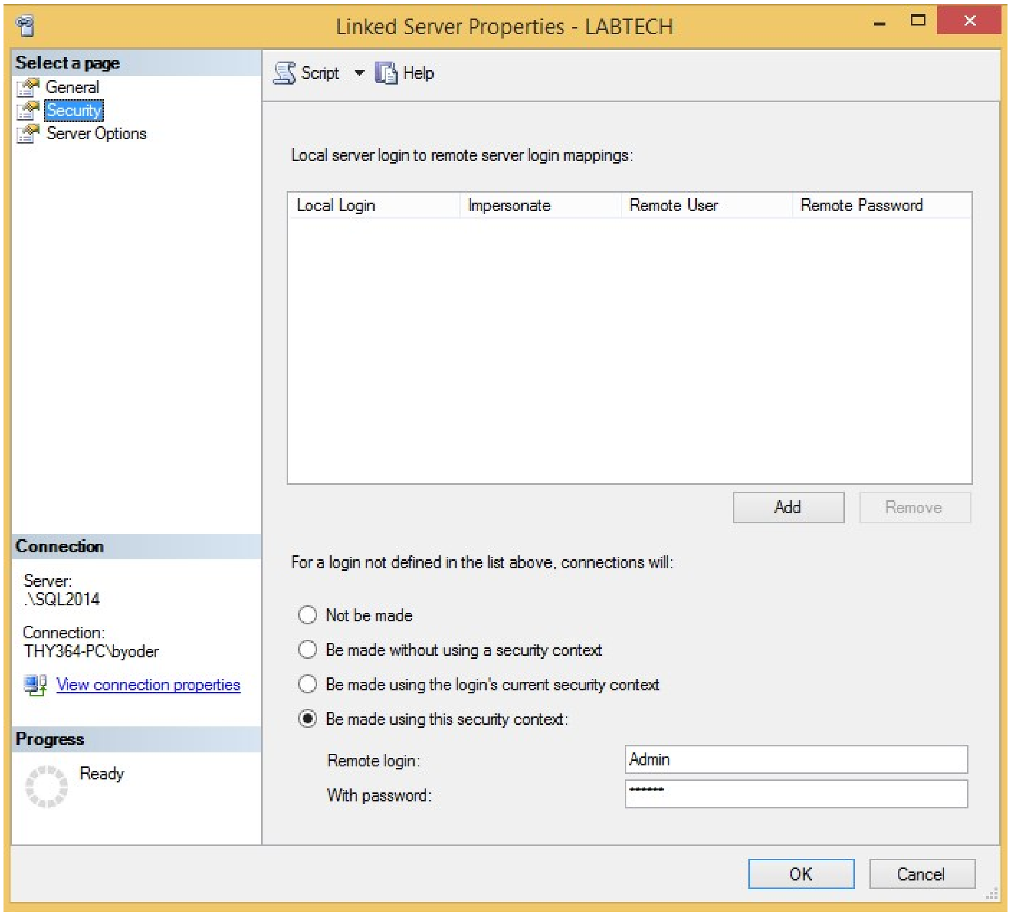
Server:
.\SQL2014

Connection:
THY364-PC\byoder

Ready

Figure 8 Configuring a MS SQL Linked Server

4. Specify a username / password for the linked server to connect to the Automate (LabTech) MySQL database.



Local server login to remote server login mappings:

Local Login	Impersonate	Remote User	Remote Password

For a login not defined in the list above, connections will:

○ Not be made
○ Be made without using a security context
○ Be made using the login's current security context
● Be made using this security context:

Remote login: Admin
With password: *******

Server:
.\SQL2014

Connection:
THY364-PC\byoder

Ready

Figure 9 Setting a login for the Automate (LabTech) database

5. Save the linked server configuration.

Step 3 - Create a Custom View

In the Secret Server MS SQL Database run the following query to create a custom view. Once the view is created, you can test it by running the query SELECT * FROM vLabTechFolderView in the Secret Server database.

```
CREATE VIEW [dbo].[vLabTechFolderView]

AS

SELECT FolderId, FolderName, FolderTree

FROM OPENQUERY(LabTech),

'SELECT CONCAT(CAST(c.ClientID AS CHAR), IFNULL(CAST((l.LocationId + 20000) AS

CHAR),"")) AS FolderId,

c.Name AS FolderName,

""AS FolderTree

FROM clients c

left join locations l

ON l.ClientID = c.ClientID
```

WHERE

l.LocationId IS NULL

UNION

SELECT l.LocationID + 20000 AS FolderId,

l.Name AS FolderName,

c.Name AS FolderTree

FROM locations l

JOIN clients c

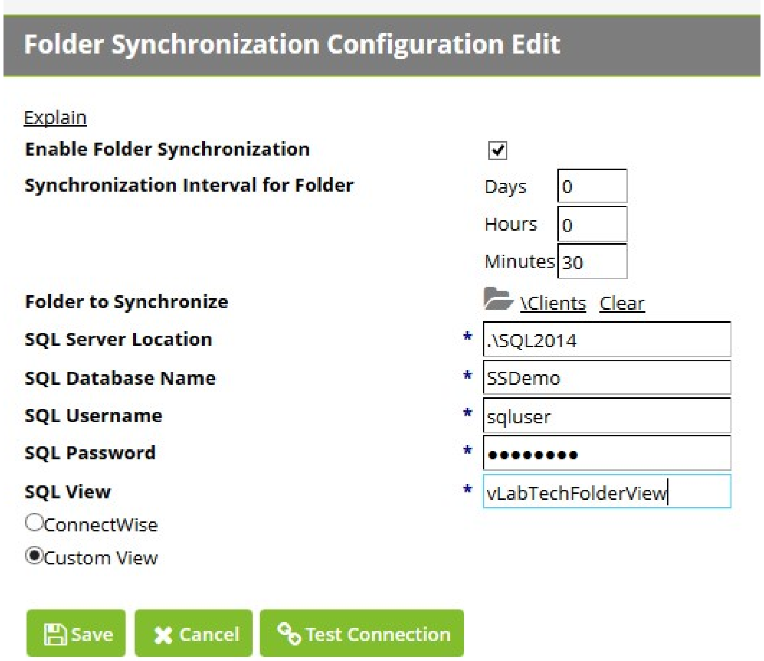
ON c.ClientID = l.ClientID

')

Step 4 - Enable Folder Synchronization in Secret Server

More explanation of how the Folder Synchronization feature works can be found under Managing Synchronized Folders.

1. Log into the Secret Server web site.
2. Click on **Admin > More** and click **Folder Synchronization.**
3. Edit to enable Folder Synchronization and fill out the connection information so that Secret Server can read from the newly created view. Note that all the Clients in Automate (LabTech) will be created as subfolders under the selected **Folder to Synchronize.**



Folder Synchronization Configuration Edit

Explain

Enable Folder Synchronization	☑	
Synchronization Interval for Folder	Days	0
	Hours	0
	Minutes	30
Folder to Synchronize	\Clients Clear	
SQL Server Location *	.\SQL2014	
SQL Database Name *	SSDemo	
SQL Username *	sqluser	
SQL Password *	••••••••	
SQL View *	vLabTechFolderView	

○ConnectWise

●Custom View

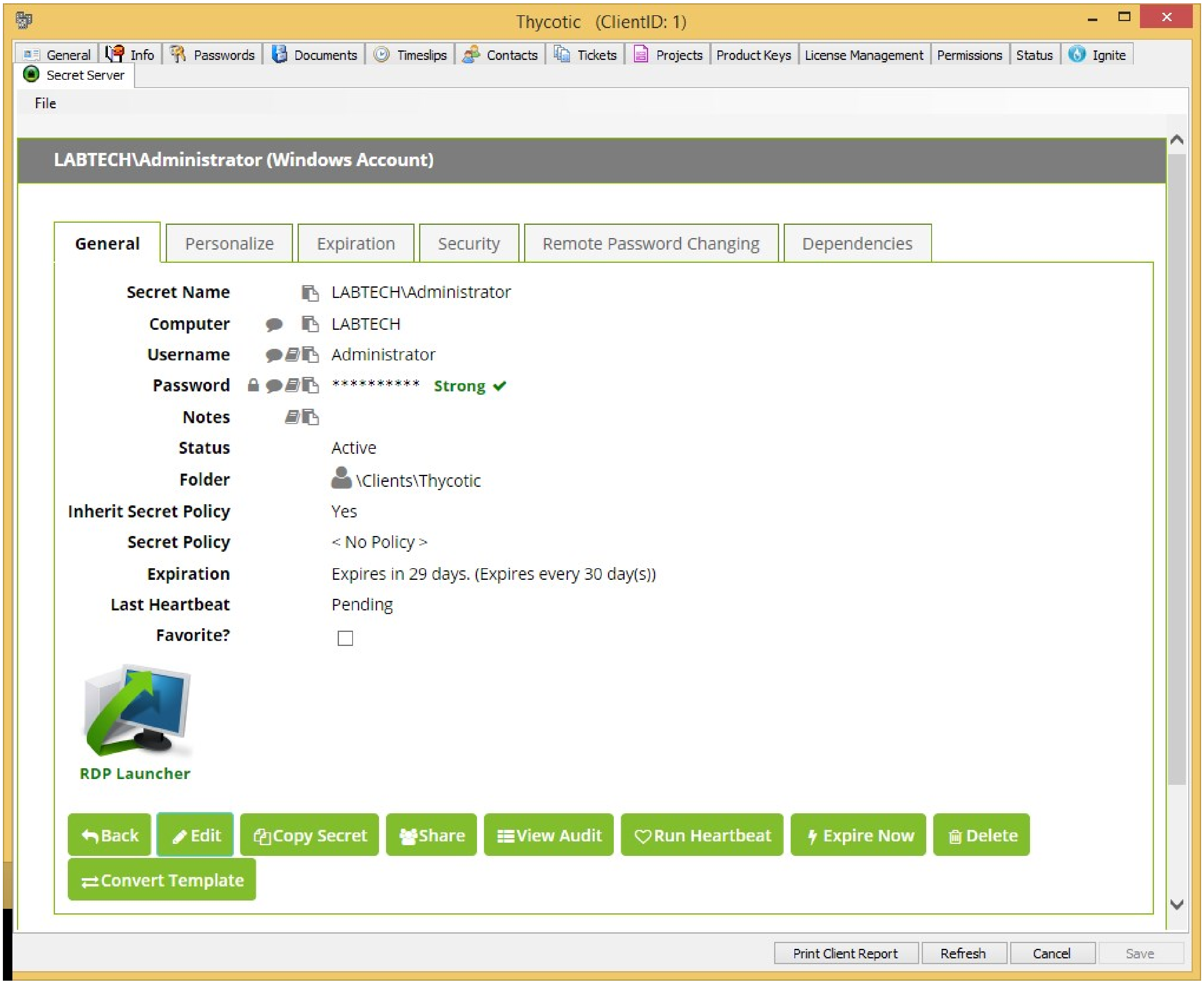
Figure 10 Enabling Folder Synchronization in Secret Server

4. Save and Secret Server will synchronize the Automate (LabTech) Clients as Folders.

Other Considerations

Launchers

Secret Server has the ability to run Session Launchers, which open Remote Desktop, PuTTY, and other applications and make a connection from the user's workstation to the target server.



LABTECH\Administrator (Windows Account)

Secret Name	LABTECH\Administrator
Computer	LABTECH
Username	Administrator
Password	********** Strong ✓
Notes	
Status	Active
Folder	\Clients\Thycotic
Inherit Secret Policy	Yes
Secret Policy	< No Policy >
Expiration	Expires in 29 days. (Expires every 30 day(s))
Last Heartbeat	Pending
Favorite?	□

Figure 11 Secret with an RDP Launcher

These can be used within the Automate (LabTech) plugin, but require a configuration change in Secret Server. By default the launchers are run using Microsoft ClickOnce, which automatically downloads the launcher file to the user's workstation. This is not supported using the plugin, so a supported protocol handler must be installed.

For information on configuring and using the protocol handler, see the instructions under the Secret Launchers and Protocol Handlers section.

Secret Server Availability

Secret Server typically is a web application typically installed on premises. In order for the plugin to function properly, the location running the Control Center must be able to connect to the Secret Server instance. For example, if a user was offsite and Secret Server was not publicly available, the user would have to connect via VPN in order for the Secret Server plugin to properly show.

Remote Desktop Manager

The integration between Thycotic Secret Server and Devolutions is created and maintained by Devolutions. This document provides guidance and best practice for implementing the integration. It is based on the following publicly available documentation from the vendor and testing performed by Thycotic. Integrations are supported to the extent of the third-party product procedures documented for this integration. Please contact the third-party for any customized setup of the integrated product.

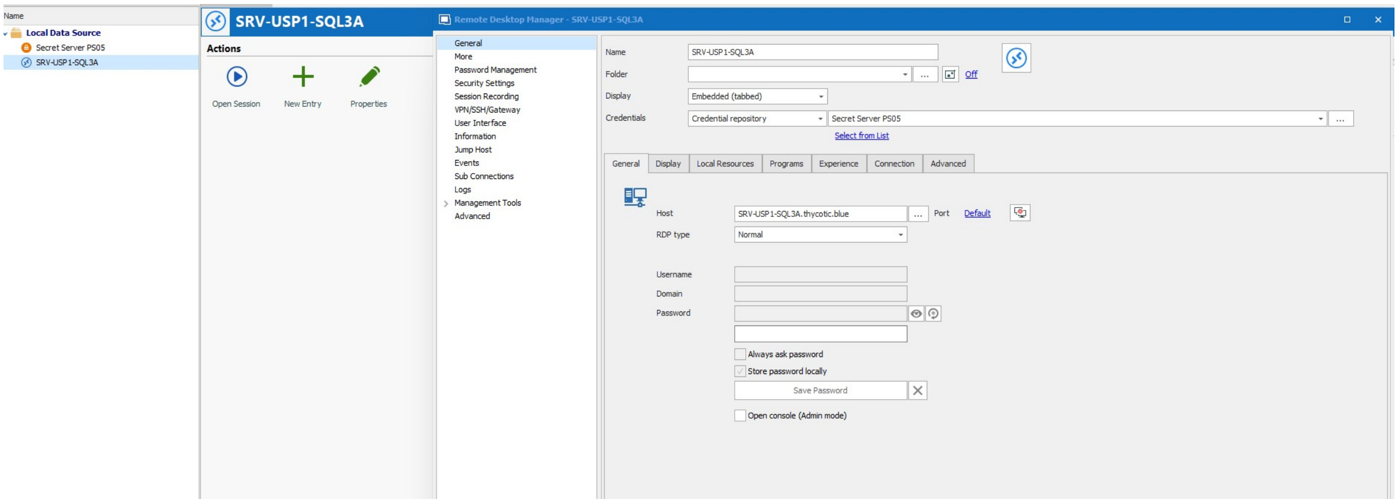
This document is to serve as an updated tutorial for natively integrating Devolutions Remote Desktop Manager with Secret Server. This was tested on the latest release version 14.0.3.0 64-bit.

Steps to Link to a Specific Server

To link a specific Secret to a specific server, create a new entry.

The following example shows a basic Remote Desktop Connection session for a server called SRV-USP1-SQL3A.

1. Right-click on the **entry**.
2. Click **Properties**.



Name: SRV-USP1-SQL3A

Folder: Off

Display: Embedded (tabbed)

Credentials: Credential repository | Secret Server PS05

Host: SRV-USP1-SQL3A.thycotic.blue | Port: Default

RDP type: Normal

Username:

Domain:

Password:

☐ Always ask password

☑ Store password locally

☐ Open console (Admin mode)

3. On the Credentials section of the entry, change it to__Credential Repository__ and choose the Secret Server credential repository.
4. Click **OK**.
5. Right click on the server and click **Open Session**. You will be prompted to choose a Secret from Secret Server.
6. Click **OK** and you will be able to log in to your server with the chosen credentials.

For accessibility/ease of use, it is recommended that secret names match the session name within Remote Desktop Manager, but it is not required.

Any type of session can be configured to use this as a credential repository.

For additional security, we recommend locking the application. This can be configured in Remote Desktop Manager security settings as shown below. Additionally, you may provide 2FA to Remote Desktop Manager itself.

HSM

Secret Server integrates with hardware security modules, or HSMs, to provide additional security of Secret Server's encryption key. When Secret Server is configured to use an HSM, the encryption key is protected by the HSM. HSMs offer several security features that traditional servers cannot. Depending on the model and design of the HSM, most HSMs are designed to be physically tamper-proof. HSMs may also be independent hardware that are on a network, which allows physically placing the HSM is a more secure location that might otherwise be too inconvenient for a server. To provide broad support for HSMs, Secret Server supports any HSM that can be configured with Microsoft's Cryptography Next Generation provider, or CNG. CNG is a layer provided by Windows Server 2008 and later that HSM manufacturers can interface with. If your HSM properly supports CNG and supports the right algorithms, Secret Server will be able to utilize your HSM. CNG provider installation and configuration varies from HSM to HSM, however, documentation is available from each HSM vendor on how to correctly install CNG providers.

HSM Requirements

Each HSM must provide support for certain algorithms through CNG.

- **RSA** 4096 – support for RSA with 4096-bit keys is required. The HSM must also support RSA for encryption and decryption, in addition to signing.
- **PKCS#1 v1.5 Padding** – The HSM must support PKCS#1 v1.5 padding for RSA encryption.

Additionally, closely follow the requirements and recommendations of the HSM vendor for things such as minimum latency, redundancy, and operating environment.

Due to limitations of the account, the NETWORK SERVICE account is not supported as an account for the IIS Application Pool. It is recommended to configure Secret Server's Application Pool as a service account.

In the advanced settings for the application pool, set "Load User Profile" to true.

> **Note:** Some HSM CNG providers interfere with each other. It is recommended that no more than one HSM CNG provider is configured on a Windows installation at a time.

Further Considerations

- If your HSM properly supports CNG and supports the right algorithms
- Each HSM must provide support for certain algorithms through CNG
- Since Secret Server is using Microsoft's CNG API to communicate with HSMs the following two references should be included:
    - The NCRYPT_IMPL_TYPE_PROPERTY (L"Impl Type") value must be NCRYPT_IMPL_HARDWARE_FLAG (0x00000001, The provider is hardware based).

  https://docs.microsoft.com/en-us/windows/win32/seccng/key-storage-property-identifiers

    - The CNG Algorithm Identifiers must support BCRYPT_RSA_ALGORITHM ("RSA") - The RSA public key algorithm. Standard: PKCS #1 v1.5 and v2.0.

  https://docs.microsoft.com/en-us/windows/win32/seccng/cng-algorithm-identifiers

Configuration

- Silent HSM Operation
- Configuring HSM Integration
- Securing HSM Integration
- HSM Redundancy
- Testing HSM CNG Configuration

Silent HSM Operation

Because Secret Server is a web application with no one physically present at the server at most times, Secret Server interacts with the HSM in “silent” mode. This will prevent the HSM from attempting to interact with any users logged onto the server.

Some HSM features require interaction. If the HSM is configured in such a way that requires interaction, Secret Server will be unable to communicate with the HSM and fail during the configuration steps. An example of such a configuration is Operator Card Sets in Thales network HSMs. If the Thales CNG provider is configured to use an Operator Card Set, or OCS, for key protection instead of Module protection, someone must be physically present at the HSM and the Server to insert their operator card when the key is needed. If the OCS quorum is more than a single card, Secret Server cannot interact with the HSM because it requires inserting and removing the OCS cards.

It is recommended that Thales’ CNG provider is configured to use Module protection instead of an OCS. It is possible to use an OCS with Secret Server if the quorum is exactly one card and the card is left in the HSM at all times.

It is recommended to consult your HSM vendor and their documentation to ensure that the HSM and their CNG provider are able to operate in silent mode and are configured to do so.

Configuring HSM Integration

To configure the HSM integration, go to the **ADMIN** menu and click **Configuration**, then select the **HSM** tab. This will start the HSM wizard, which will guide the process of selecting the HSM's CNG provider.

The list of available CNG Providers is done by querying for the list of registered CNG providers. Each provider must correctly report that it is a "Hardware" provider, and that it is not a Smart Card reader. If an error occurs while querying the CNG provider for its properties, it will not appear in the list, however the error is reported to Secret Server's system log. If the desired CNG provider does not appear in the list of CNG providers, ensure that the CNG provider is correctly registered and that IIS has been restarted after the CNG provider is registered. Also check that an error is not occurring while querying the HSM by examining the system log.

Once the CNG providers are selected, Secret Server will simulate encryption and decryption operations and verify the results to check that it is functioning properly. The final step will be to verify the selected providers, and then enable HSM integration. Detailed steps are provided throughout the HSM configuration wizard.

Securing HSM Integration

The wizard to enable and disable HSM integration is protected by the “Administer HSM” role permissions in Secret Server. These permissions should be carefully assigned – if at all. Additionally, an Event Subscription can be created that sends alerts when this role permission is assigned or unassigned from a role.

Configuring the HSM also has its own Event Subscriptions for when the HSM integration is enabled or disabled.

Additionally, an application setting can be added to Secret Server to prevent changes to HSM configuration. Disabling and enabling this requires direct access to the file system where Secret Server is installed.

To enable this, edit the web-appSettings.config file within Secret Server to contain a key called **LockHsmConfiguration** with a value of True as follows:

```
<?xml version="1.0" encoding="utf-8" ?>
<appSettings>
 <add key="LockHsmConfiguration" value="True" />
</appSettings>
```

This will prevent access to the HSM configuration pages regardless of role permissions. The only way to gain access is to remove this setting, thus proving you at least have access to the server where Secret Server is installed.

HSM Redundancy

This varies from HSM to HSM, and the vendor’s documentation on how to back up the HSM should be referenced. Backups are typically either made to common file location, or another HSM, or onto a smart card with the HSM’s built-in smart card reader.

As long as the CNG provider is installed on the server and a key exists on the HSM with the same identifier, Secret Server will attempt to use that key.

Testing HSM CNG Configuration

Secret Server does its own testing and verification of the HSM and its CNG provider before the HSM integration can be enabled. To further diagnose any issues with the HSM, the **certutil** command line utility that is part of Windows can test the HSM with the **–csptest** option specified. An example output may contain something like this:

- **Provider Name:** SafeNet Key Storage Provider
- **Name:** SafeNet Key Storage Provider

```
Asymmetric Encryption Algorithms:
 RSA
 BCRYPT_ASYMMETRIC_ENCRYPTION_INTERFACE -- 3
 NCRYPT_ASYMMETRIC_ENCRYPTION_OPERATION -- 4
 NCRYPT_SIGNATURE_OPERATION -- 10 (16)
 NCryptCreatePersistedKey(SafeNet Key Storage Provider, RSA)
 Name: cngtest-6166f8fe-8caf-4e30-8e5c-a-24575
```

Pass

Examine the output of the test by looking for your CNG Provider Name for your HSM and verifying the result. It is recommended that this test be run using the same account as the Application Pool Secret Server is using. If the testing tool reports errors, consult your HSM's vendor or documentation for resolution.

IBM

Below are the following integrations that are available with IBM:

- IBM IGI
- QRadar
- IBM Verify Gateway for RADIUS Server
- WebSphere

IBM IGI

The integration between Thycotic Secret Server and IBM IGI is created and maintained by IBM. This document provides guidance and best practice for implementing the integration. It is based on the following publicly available documentation from IBM and testing performed by Thycotic. Integrations are supported to the extent of the third-party product procedures documented for this integration. Please contact the third-party for any customized setup of the integrated product.

Why should IBM IGI be Integrated with Thycotic Secret Server? IBM Secret Server now integrates with IBM Identity Governance & Intelligence (IGI) to provide an enhanced privileged identity and access governance solution. IBM Secret Server securely stores privileged credentials and other sensitive information, known as secrets, and provides visibility into who has access to those secrets. IGI ensures that users' access levels are compliant with regulations and prevent SoD violations.

By integrating these two products, organizations can centrally manage and control all identities, including privileged identities and entitlements. Compliance use cases such as managing entitlements, certification, new user provisioning, and application access permissions are now unified for both privileged and non-privileged users. This integration unifies processes for privileged and non-privileged users. It ensures that privileged users are granted appropriate access permissions based on similar privileged users' attributes (e.g. job role, department), and in accordance with the organization's access policy. It reduces the attack surface and enhances regulatory compliance by limiting access privileges and deactivating orphan privileged accounts. It streamlines governance and compliance processes by generating reports and auditing all identities and access permissions directly from IGI.

For further information please visit IBM's website for Security Identity Governance & Intelligence (IGI).

Getting Started with IBM IGI

Before configuring IBM IGI to integrate with Secret Server you will need to make sure you have met all of the system requirements.

After installing IBM IGI please complete the following steps below before configuring the connection between Secret Server and IBM IGI.

1. Configure the OpenID Connect Provider.
2. Installing Security Directory Integrator 7.2.
3. Installing Tivoli Dispatcher.

Pre-requisites

System Requirements

Product	Details	
IBM IGI		
	Version	Version 5.2
	Hardware	For further information about system requirements for IBM IGI please visit: https://www.ibm.com/support/knowledgecenter/en/SSGHJR_5.2.0/com.ibm.igi.doc/installing/cpt/c_hardware_reqs.html.
Secret Server		
	Version	Thycotic Secret Server version 10.5 or later.
	Hardware	For further information about system requirements for Secret Server please visit: https://docs.thycotic.com/ss/10.8.0/secret-server-setup/system-requirements/index.md.

Note: For the purpose of this guide, the IBM IGI internal database was used. To use a different data and directory server, refer to the configuration documentation from IBM.

Disable the OpenID Connector

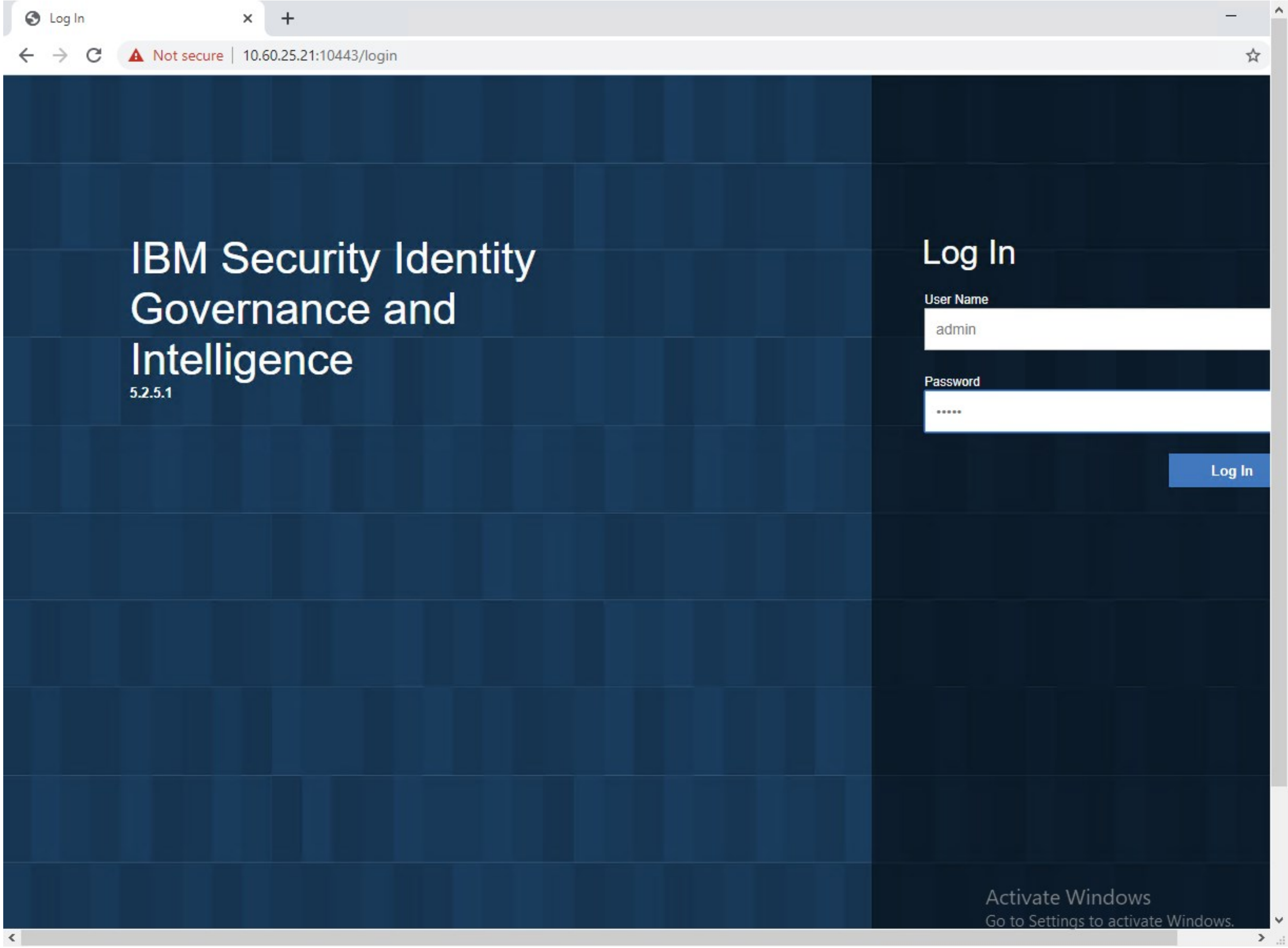
The next step after installing IBM IGI is to configure the OpenID Connect Provider.

To configure OpenID Connect Provider:

1. Open a browser and navigate to the IBM IGI login page. This should be an IP address and port number (Example: 10.60.25.21:9443). The default port number is **9443**.

    Note: If you receive an error screen due to not having a certificate configured, please proceed by clicking **Advanced | proceed (unsafe)**.

2. The **IBM IGI Login** page appears.



Log In
Not secure | 10.60.25.21:10443/login
IBM Security Identity Governance and Intelligence
5.2.5.1
Log In
Log In
Activate Windows
Go to Settings to activate Windows.

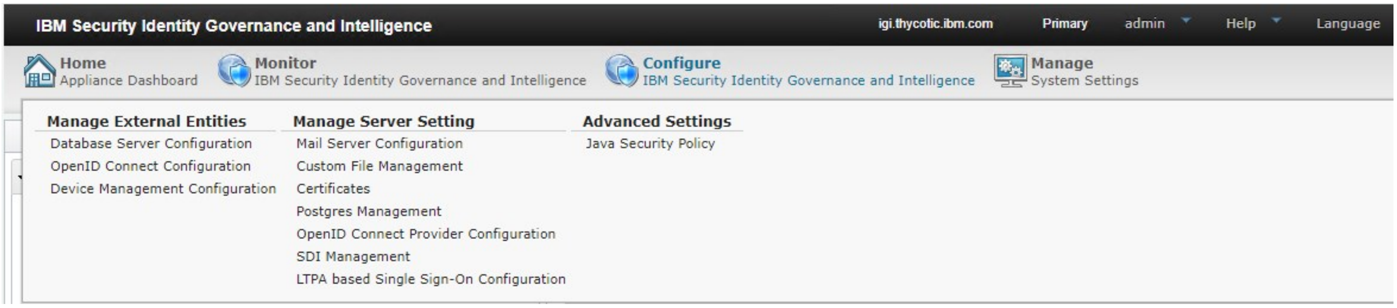
User Name: admin

Password: •••••

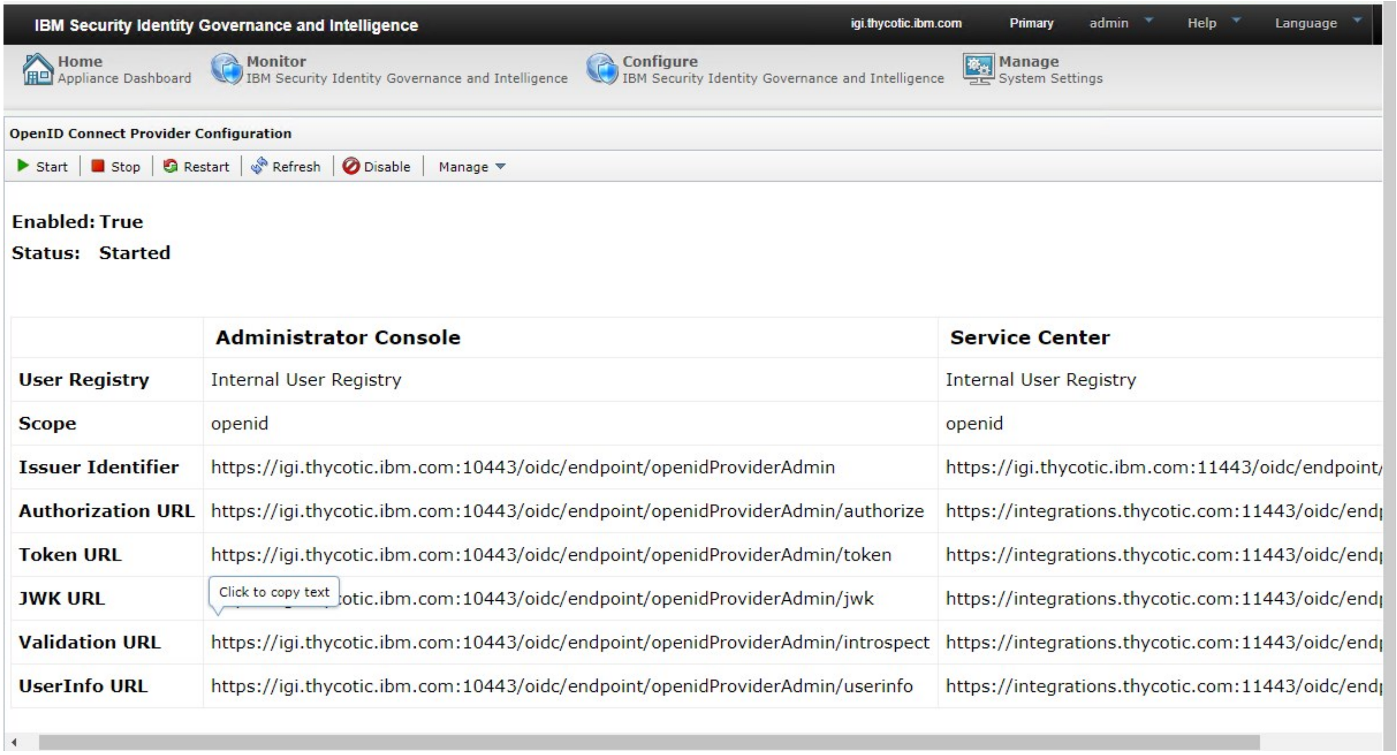
3. Fill in the required information, such as the user name, password, and click **Log In**. The IBM IGI user interface appears.

    Note: The default value for user name and password is admin.

4. Click **Configure** | **Manage Server Setting** | **OpenID Connect Provider Configuration**. The **Connect Provider Configuration** page appears.


IBM Security Identity Governance and Intelligence
igi.thycotic.ibm.com
Primary
admin
Help
Language
Home
Appliance Dashboard
Monitor
IBM Security Identity Governance and Intelligence
Configure
IBM Security Identity Governance and Intelligence
Manage
System Settings
Manage External Entities
Database Server Configuration
OpenID Connect Configuration
Device Management Configuration
Manage Server Setting
Mail Server Configuration
Custom File Management
Certificates
Postgres Management
OpenID Connect Provider Configuration
SDI Management
LTPA based Single Sign-On Configuration
Advanced Settings
Java Security Policy

5. Click **Disable**.



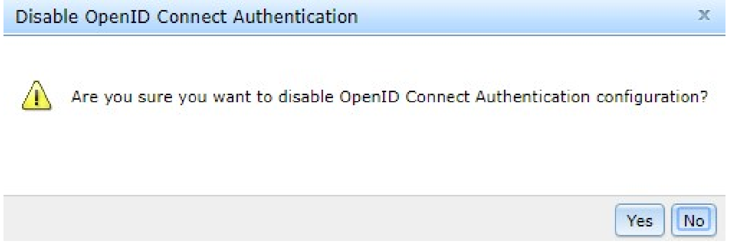
OpenID Connect Provider Configuration

Enabled: True

Status: Started

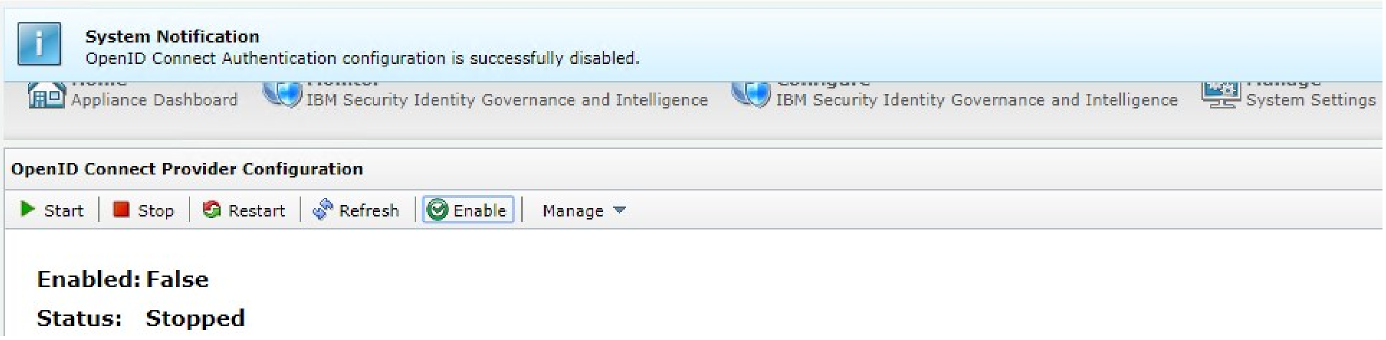
	Administrator Console	Service Center
User Registry	Internal User Registry	Internal User Registry
Scope	openid	openid
Issuer Identifier	https://igi.thycotic.ibm.com:10443/oidc/endpoint/openidProviderAdmin	https://igi.thycotic.ibm.com:11443/oidc/endpoint/
Authorization URL	https://igi.thycotic.ibm.com:10443/oidc/endpoint/openidProviderAdmin/authorize	https://integrations.thycotic.com:11443/oidc/end
Token URL	https://igi.thycotic.ibm.com:10443/oidc/endpoint/openidProviderAdmin/token	https://integrations.thycotic.com:11443/oidc/end
JWK URL	otic.ibm.com:10443/oidc/endpoint/openidProviderAdmin/jwk	https://integrations.thycotic.com:11443/oidc/end
Validation URL	https://igi.thycotic.ibm.com:10443/oidc/endpoint/openidProviderAdmin/introspect	https://integrations.thycotic.com:11443/oidc/end
UserInfo URL	https://igi.thycotic.ibm.com:10443/oidc/endpoint/openidProviderAdmin/userinfo	https://integrations.thycotic.com:11443/oidc/end

6. A message, '**Are you sure you want to disable OpenID Connect Authentication configuration?**' appears.



Are you sure you want to disable OpenID Connect Authentication configuration?

7. Click **Yes**. The status of the configuration appears.

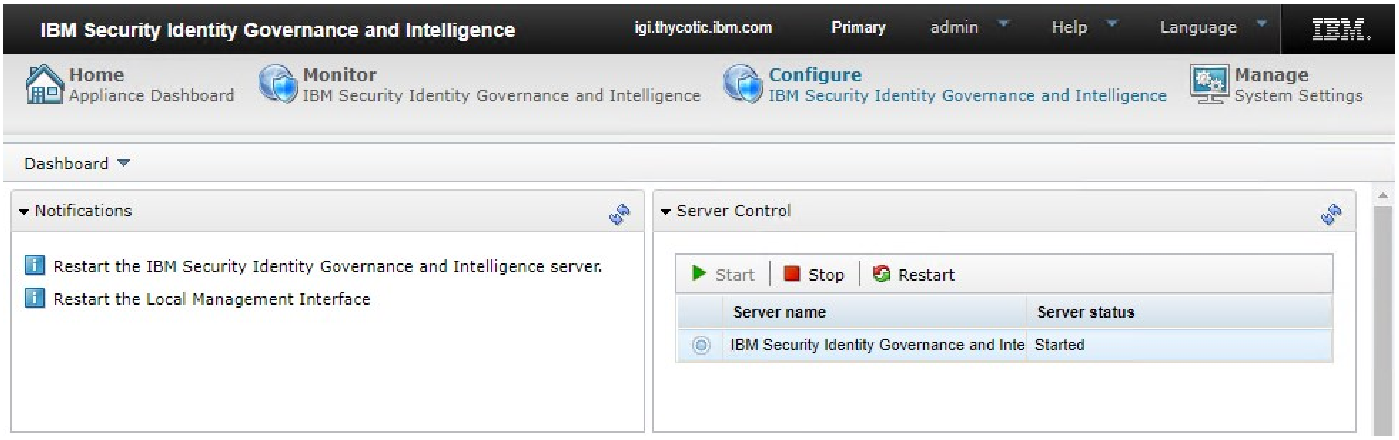

Home
Appliance Dashboard
Monitor
IBM Security Identity Governance and Intelligence
Configure
IBM Security Identity Governance and Intelligence
Manage
System Settings
Start
Stop
Restart
Refresh
Enable
Manage

System Notification
OpenID Connect Authentication configuration is successfully disabled.

OpenID Connect Provider Configuration

Enabled: False
Status: Stopped

Note: In the IBM IGI UI, the notifications are listed in the **Notifications** section.

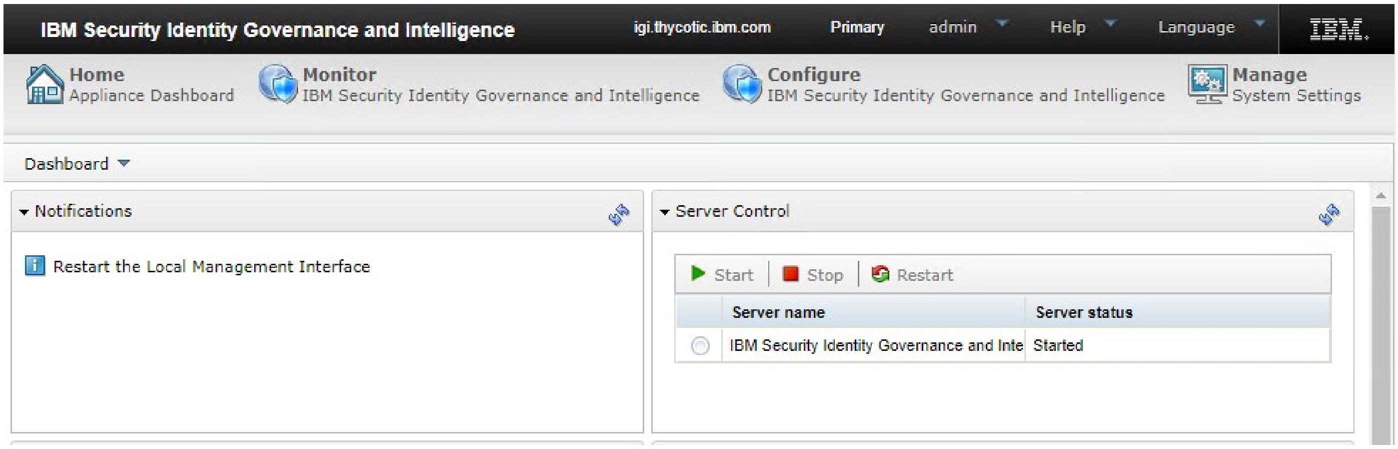


Notifications
- Restart the IBM Security Identity Governance and Intelligence server.
- Restart the Local Management Interface

Server Control

	Server name	Server status
○	IBM Security Identity Governance and Inte	Started

8. In the **Server Control** section, select the server and click **Restart**.

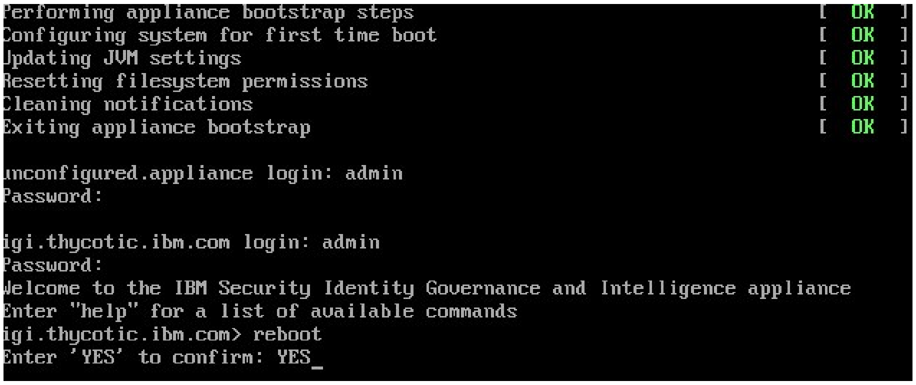


Notifications
- Restart the Local Management Interface

Server Control

	Server name	Server status
○	IBM Security Identity Governance and Inte	Started

9. Go to the **IBM IGI Virtual Appliance**.

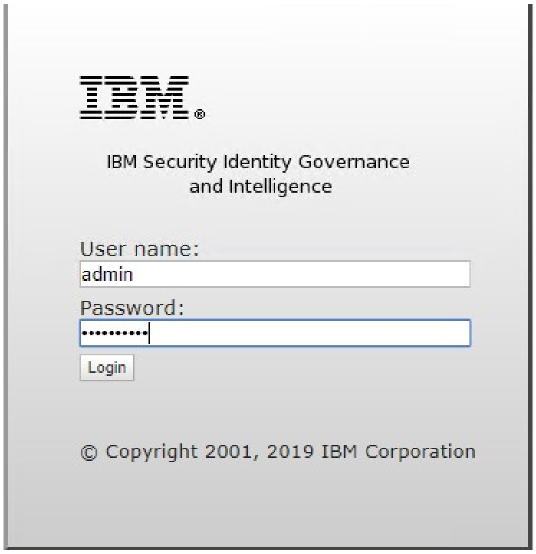


```
Performing appliance bootstrap steps                              [  OK  ]
Configuring system for first time boot                            [  OK  ]
Updating JVM settings                                             [  OK  ]
Resetting filesystem permissions                                  [  OK  ]
Cleaning notifications                                            [  OK  ]
Exiting appliance bootstrap                                       [  OK  ]

unconfigured.appliance login: admin
Password:

igi.thycotic.ibm.com login: admin
Password:
Welcome to the IBM Security Identity Governance and Intelligence appliance
Enter "help" for a list of available commands
igi.thycotic.ibm.com> reboot
Enter 'YES' to confirm: YES_
```

10. In the Virtual Appliance, type reboot and press **Enter**.

11. To confirm, type YES and press **Enter**.

12. The **IBM IGI login** dialog box appears.



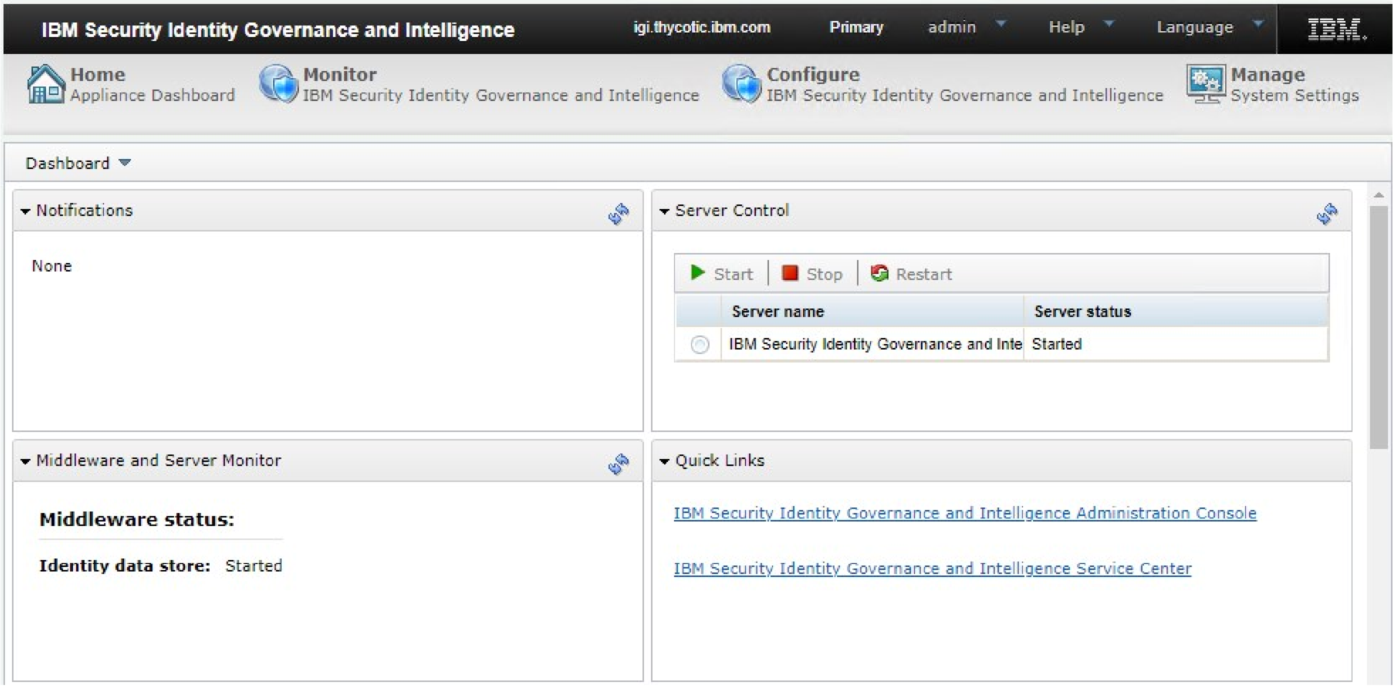
IBM Security Identity Governance
and Intelligence

User name: admin

Password: ••••••••••

© Copyright 2001, 2019 IBM Corporation

13. Fill in the required information, such as the user name, password, and click **Log In**. The IBM IGI user interface appears.


IBM Security Identity Governance and Intelligence
igi.thycotic.ibm.com
Primary
admin
Help
Language
IBM.
Home
Appliance Dashboard
Monitor
IBM Security Identity Governance and Intelligence
Configure
IBM Security Identity Governance and Intelligence
Manage
System Settings
Dashboard
Notifications
Server Control
Start
Stop
Restart
Middleware and Server Monitor
Quick Links

None

	Server name	Server status
○	IBM Security Identity Governance and Inte	Started

Middleware status:

Identity data store: Started

IBM Security Identity Governance and Intelligence Administration Console

IBM Security Identity Governance and Intelligence Service Center