ServicePass Installation Guide

SurePassID Authentication Server 2025

© 2013-2025 SurePassID, Corp. All rights reserved. SurePassID names and logos are trademarks of SurePassID, Corp. Other trademarks mentioned belong to their respective owners. Patents pending.

SurePassID, Corp.

360 Central Avenue

First Central Tower

Suite 800

St. Petersburg, FL 33701

USA

+1 (888) 200-8144

www.surepassid.com

Table of Contents

About SurePassID ServicePass 3
Other SurePassID Guides 4

What is SurePassID ServicePass? 4
Prerequisites 5
System Security 7
User Security 7
Database 7
Internet Information Server 7
Post Configuration Steps 8

Installing and Configuring ServicePass 8

Customizing the System 14
Web.config 15
Notification Messages 17
Notification Message Customization 17
Default Language 18

About SurePassID ServicePass

This guide explains how to install and configure SurePassID ServicePass Self-Service Portal (SSP). This guide's purpose is to provide a reference for system administrators.

This guide provides information on the following topics:

- **What is SurePassID ServicePass?**
  - A brief introduction to the ServicePass
- **Installing and Configuring ServicePass**
  - Detailed explanations for installing ServicePass

Other SurePassID Guides

ServicePass has the following companion guides that provide additional detail on specific topics for SurePassID:

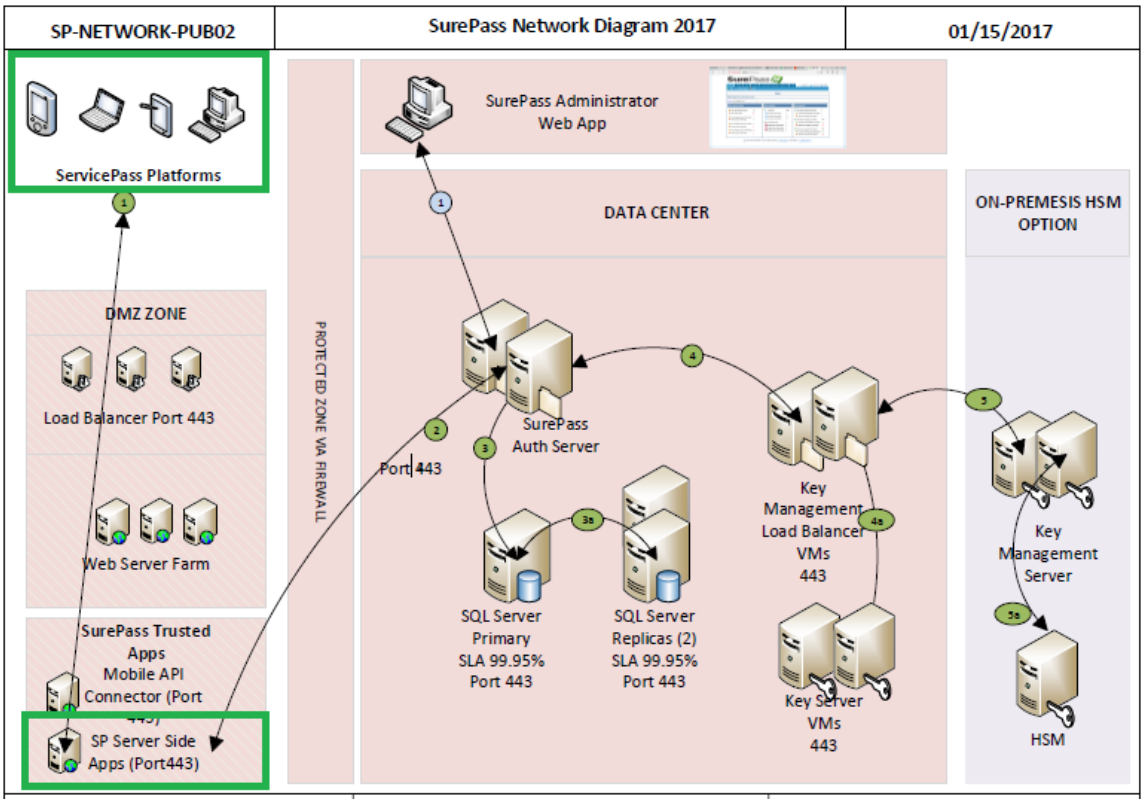
- Server API Guide
- FIDO U2F Mobile API Guide
- System Administration Guide
- Local Agent Guide
  - High performance Radius Server
  - Windows Event Log Integration
  - Active Directory Synchronization
- Desktop OTP Authenticator Guide
- Google Authenticator Guide
- Windows Login Authentication Guide

What is SurePassID ServicePass?

SurePassID ServicePass is a web-based self-service portal for managing One-Time Passwords (OTP), Push Authentication, and FIDO2 security tokens. It is available as an out-of-the-box installable image with easy branding options, or as an open-source solution for full customization.

ServicePass lets users manage strong authentication tokens independently, reducing help desk costs. It leverages SurePassID API's (REST) for full integration with existing enterprise service desk systems and supports new self-service applications.

Below is a high-level architectural view.



This document focuses on the areas in the green boxes.

Prerequisites

SurePassID Server can be installed on the following Windows versions:

- Windows Server 2016 – All versions
- Windows Server 2019 – All versions

- Windows Server 2022– All versions
- Windows Server 2025– All versions
- Windows 10
- Windows 11

Supported Tokens

ServicePass supports all the security tokens that SurePassID Server supports:

Soft Tokens:

- SurePassID Mobile Authenticator (OATH, PUSH)
- SurePassID Desktop Authenticator
- SurePassID Push OTP Technologies (SMS, Email, SMS challenge response, FIDO2)
- Mobile OTP
- SurePassID Google Authenticator

Hard Tokens:

- SurePassID TapID
- SurePassID Display Card
- SurePassID TapID Treo
- SurePassID OTP Key FOB
- OATH certified device
- FIDO2 certified device

Self-Service Functions

ServicePass offers all the functionality users need to manage their soft and hardware tokens. The ServicePass provides the following capabilities:

- Token Activation & Registration
- Token Synchronization
- Lost Token Disablement and Re-Issuance
- Automated Notifications
- Password Reset
- SurePassID API Advanced features
- Integration into 3rd Party applications

System Security

SurePassID ServicePass does not include SSL certificates. Configure ServicePass (IIS web app) for SSL with your organization's certificates for production or self-signed certificates for testing.

ServicePass should connect to the SurePassID Authentication Server using transport-level security (HTTPS) on a designated port (see Customizing the System). Only allow security system access on that port from the ServicePass server IP.

The configuration file can restrict permitted REST API requests; only enable those required by your mobile app.

For higher security, ServicePass can be daisy-chained across multiple servers. Always follow security best practices when deploying mobile applications.

User Security

ServicePass is delivered as a secure system. Users accessing ServicePass are required to provide their Windows Active Directory, LDAP or SurePassID username and password. By default, ServicePass also requires two factor authentication for access to the self-service portal. The reason for this is simple:

Adding/modifying a security token requires absolute proofing of the individual. Username and password alone are insufficient.

In situations where the user's two factor authentication token has not been issued (or not functioning properly) ServicePass can send a passcode to the user via email, SMS or SMS challenge response to securely authenticate the user.

For certain applications such as intranet portals, this can be turned off.

Database

The system does not require any database access.

Request Logging

The system does support request logging. Logging captures the payload and IP of the requesting application. By default, the logs will be persisted in the local file in the local file system. Alternatively, they can be sent to the Windows Event Log for persistence, troubleshooting or further analysis.

Internet Information Server

The Windows server must have the IIS feature enabled.

Post Configuration Steps

It is HIGHLY recommended that you proceed with the following steps after installation:

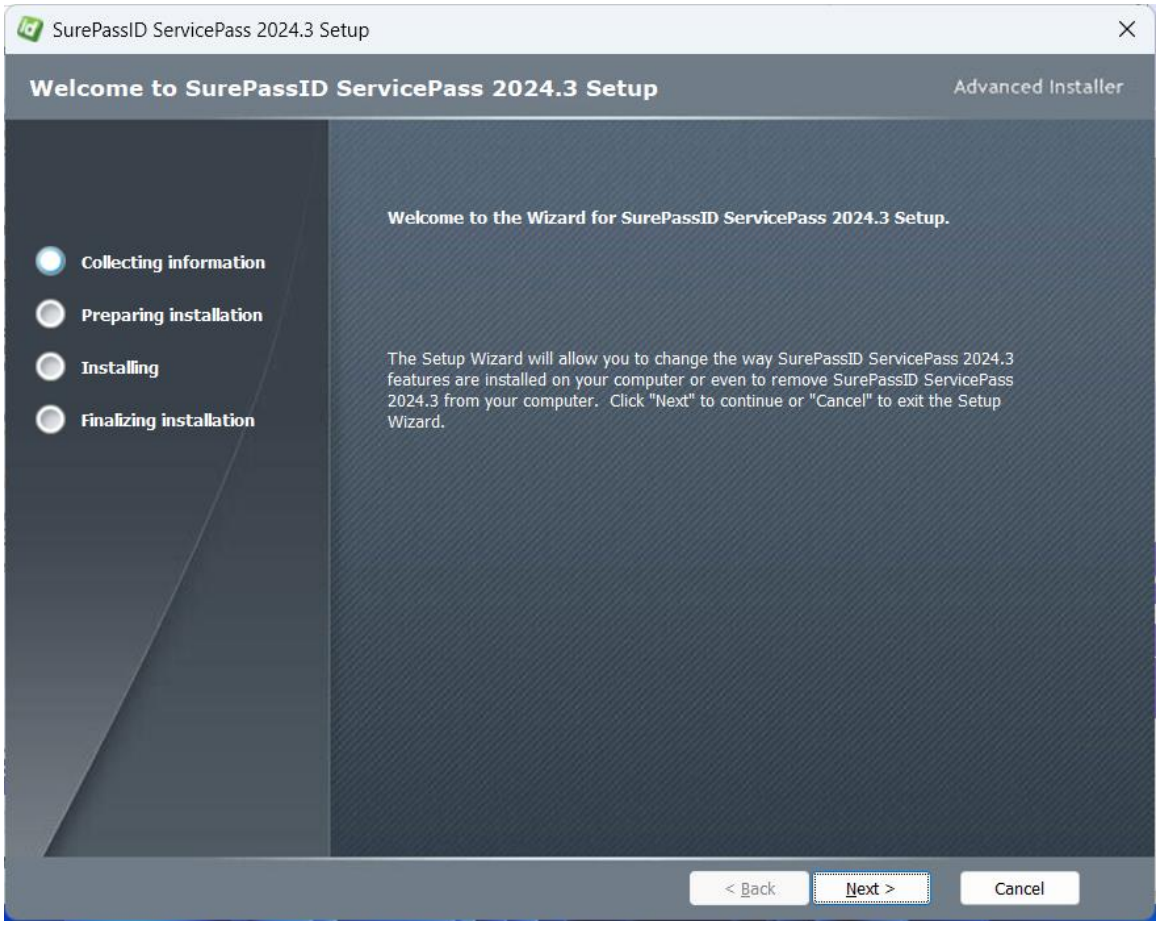
- Set up TLS for the ServicePass IIS virtual directory.
- Rename the ServicePass IIS virtual directory to something that conforms to your standards.
- Update DNS (internal or external depending on the use) to allow for access to the ServicePass via A record or CNAME.
- Customize the web.config file as per the Customizing the System section
- Protect the **web.config** file in the root folder of the SurePassID configuration by encrypting it using Aspnet_regiis utility. Detail procedures on how to do this can be found here:

  https://msdn.microsoft.com/en-us/library/zhhddkxy(v=vs.140).aspx

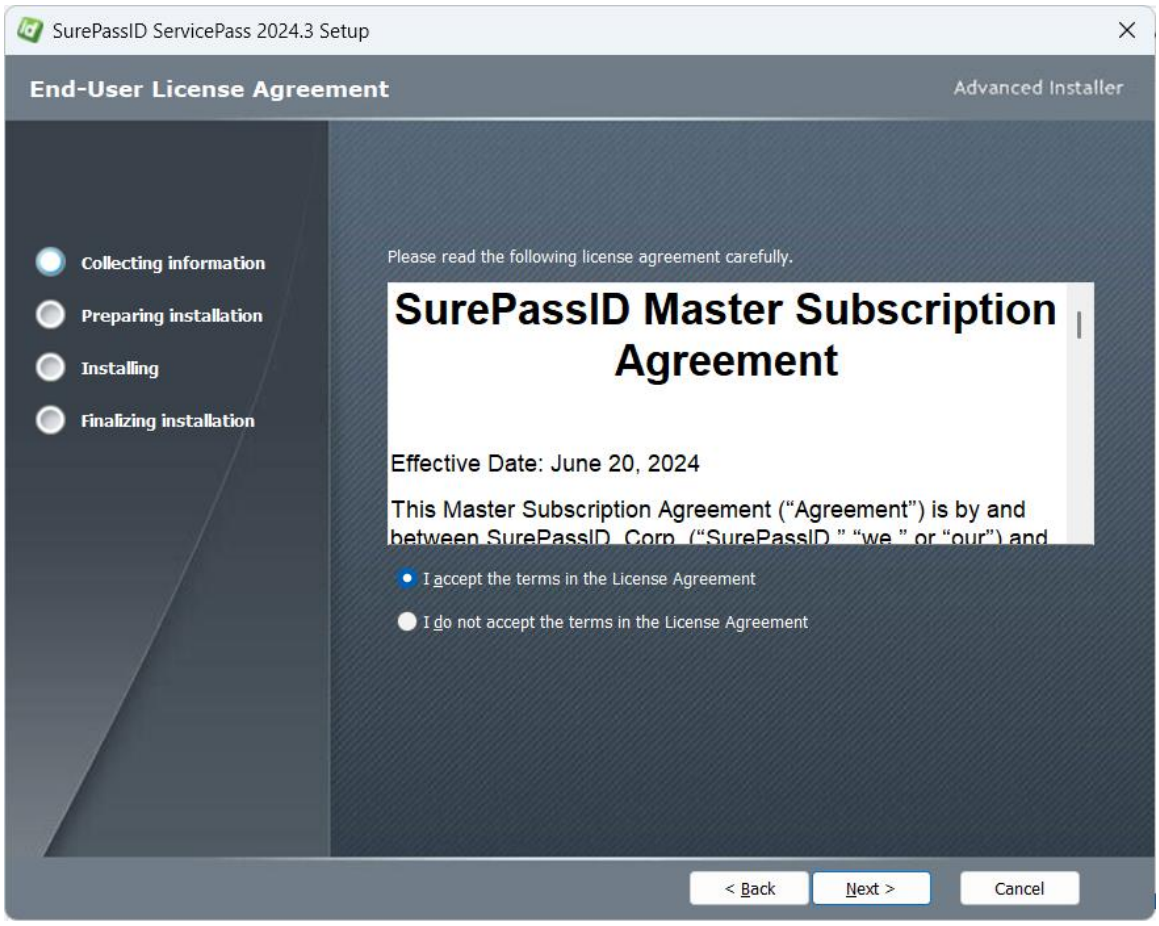
Installing and Configuring ServicePass

SurePassID ServicePass is distributed as a Windows msi installer file in a zip file (**SPPASS.ZIP**).

After downloading and unzipping **SPPASS.ZIP**, locate the installer file, copy the file to the appropriate Windows server (if not already there), and run the installer.

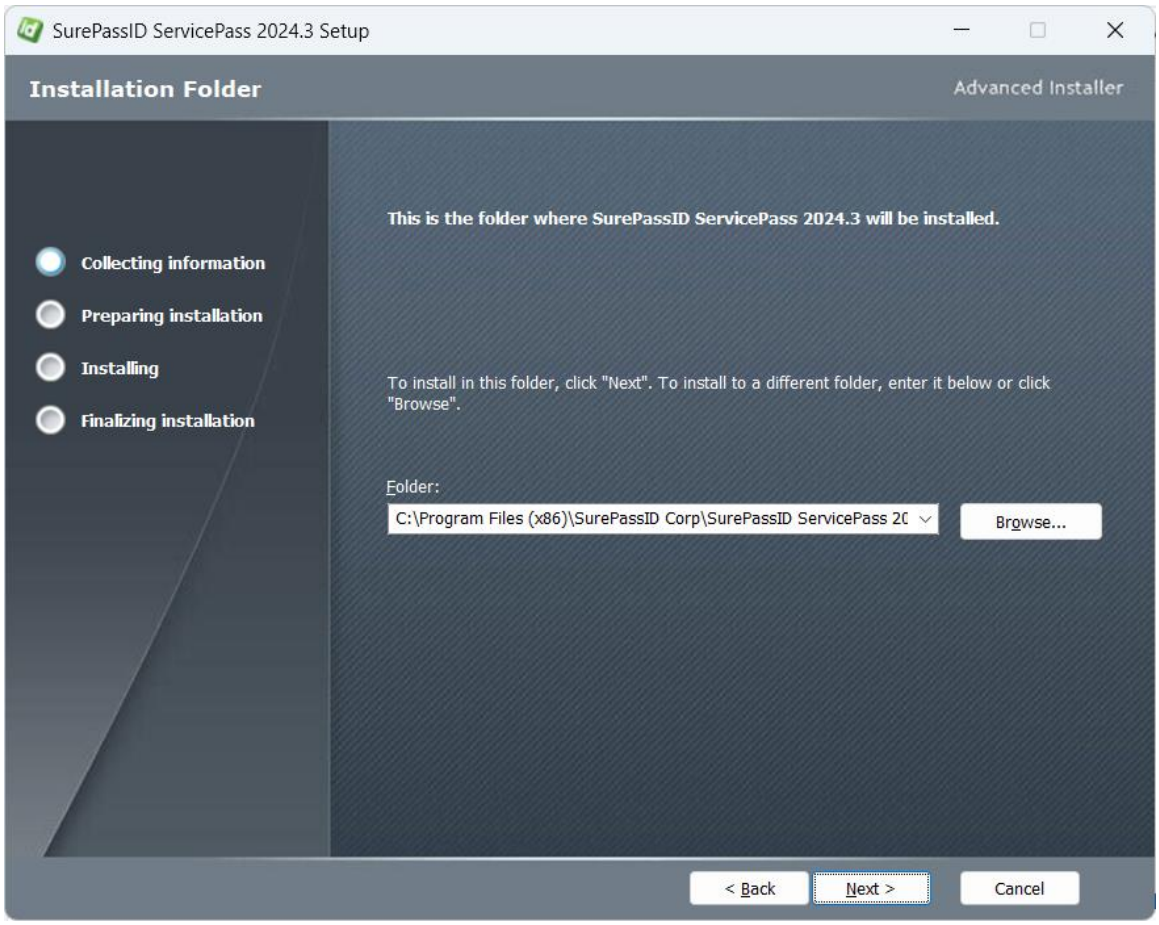


Click **Next**.

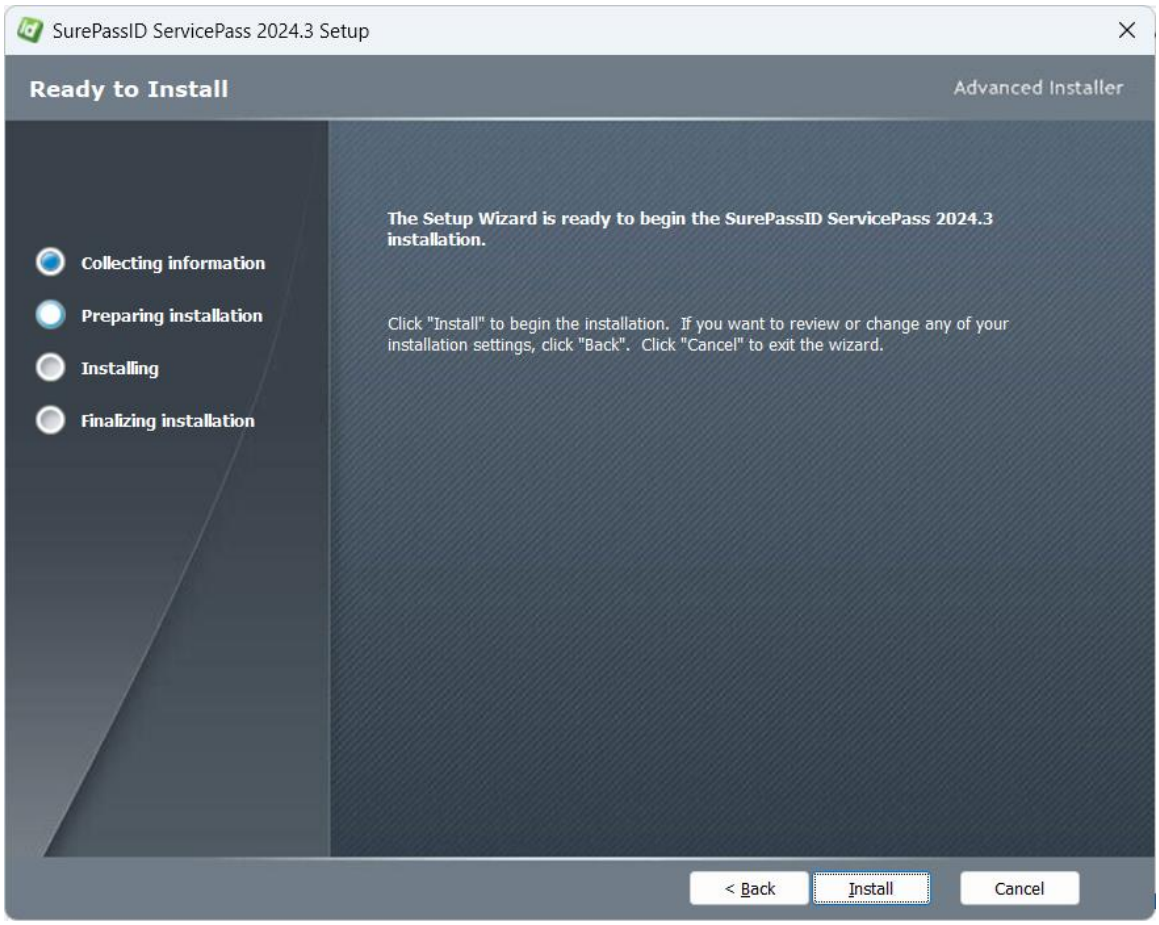


Review EULA

Read the End User License Agreement and click the **I accept the terms in the License Agreement** if the license Agreement is acceptable **and then** press the **Next** button to proceed with the install.

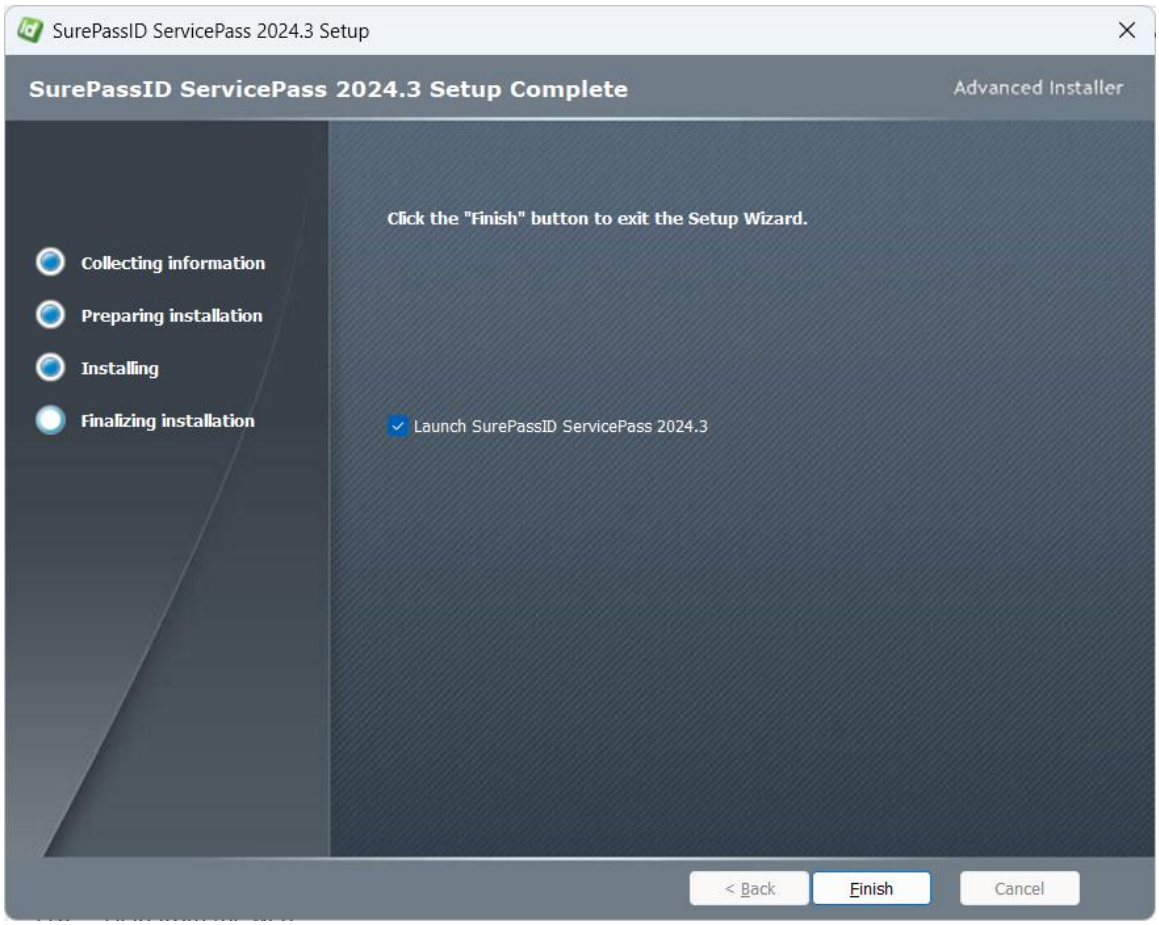

SurePassID ServicePass 2024.3 Setup
Installation Folder
Advanced Installer
Collecting information
Preparing installation
Installing
Finalizing installation
This is the folder where SurePassID ServicePass 2024.3 will be installed.
To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".
Folder:
C:\Program Files (x86)\SurePassID Corp\SurePassID ServicePass 2(
Browse...
< Back
Next >
Cancel

Click **Next**.

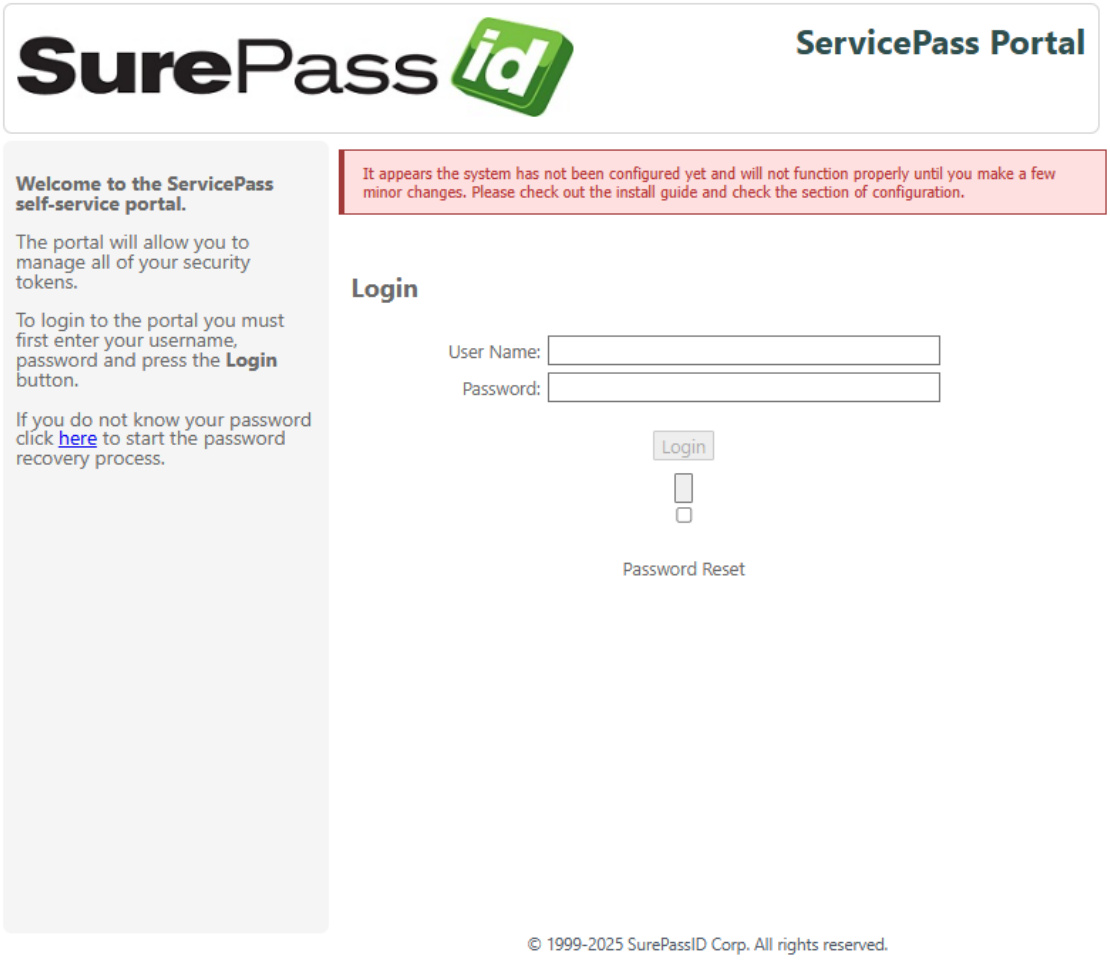


Click **Install**.

You will see files being installed and IIS being configured. When completed, you will see:



Click **Finish**. ServicePass will be started and you will see the following screen:



This screen indicates that the system is installed and requires customization before use. After customizing, this message will disappear. The next section explains how to configure your SurePassID Authentication Server instance and set default user functionality.

Customizing the System

After installation, the system will work but needs configuration to meet your company's needs before it can handle requests effectively.

Customizations are made in the **web.config** file located in the root folder of the SurePassID installation. Local customizations are made by each tenant using the SurePassID Admin portal.

Web.config

The web.config file is an XML file and is part of the .Net Framework. The file contains global customization settings. Some of the settings are SurePassID specific (<configuration>**<appsettings>)** and you should change them to suit your needs. Other settings affect the way that ASP .Net operates and you should not change these settings unless you have experience in this area. Some settings you can change and others you should not. If you make a change to web.config that violates the rules of xml syntax, the system will not run and you will receive an error. The table below describes the most notable SurePassID specific settings:

<configuration>**<appsettings> keys**

Parameter	Description
Server.RESTEndPoint	The SurePassID server endpoint that will process all ServicePass requests. • Sandbox – SurePassID cloud sandbox • Prod – SurePassID production cloud system • Use the SurePassID on-premises url
Server.CompanyAccount	Your account in the SurePassID Authentication server. Only used if Server.AllowSubDomains = FALSE.
Server.CompanyAccountKey	Your account key in the SurePassID Authentication server. Only used if Server.AllowSubDomains = FALSE.
Server.Company_<url>_Account	Maps a URL to a SurePassID tenant account. Only used if Server.AllowSubDomains = TRUE. Substitute <url> with the URL that points to ServicePass.
Server.Company_<url>_Key	Maps a URL to a SurePassID tenant account key. Only used if Server.AllowSubDomains = TRUE. Substitute <url> with the URL that points to ServicePass.
Server.AllowSubDomains	TRUE – ServicePass will map a URL to a specific SurePassID MFA server tenant. One of more Server.Company_<url>_Account and Server.Company_<url>_Key pairs must be defined. FALSE – Ignore the URL and always you Server.CompanyAccount and Server.CompanyAccountKey to identity the SurePassID MFA server tenant
Server.Trace	Logs all activity that passes from the ServicePass to the SurePassID Authentication server. The log output is stored in

	the **Trace** subfolder of the installation folder.
Server.2FARequired	0=2fa mandatory. User can use sms, email, or push notification if they do not have a token assigned to them yet. 1=no 2fa, single factor only 2=2fa required if the user has a token assigned to them. If not, single factor is okay. Good for intranet.
Server.AppId	FIDO U2F AppId. This will be the url of the ServicePassID DNS name or the FIDO U2F Facet URL.
Server.PushRelyingPartyAppAuthURL	When the user requests a push question sent to their mobile to login, this is what will be displayed in the message as the requesting system.
Server. EdgeHeaderKey	The custom header name that will be sent to SurePassID MFA server. The value can be tested at any edge reverse proxy/load balancer such to confirm trusted the traffic is from a trusted endpoint. Leave blank to not send a custom header.
Server. EdgeHeaderValue	The custom header value that will be sent to SurePassID MFA server. The value can be tested at any edge reverse proxy/load balancer such to confirm trusted the traffic is from a trusted endpoint.
Server.DefaultFromEmailAddress	Optional: Email FROM address when sending email notifications, overriding the values in the SurePassID Server.
Server.DefaultFromEmailName	Optional: Email FROM name when sending email notifications, overriding the values in the SurePassID Server.
Server.DefaultEmailFormat	Format of email notifications sent to the user (forgot password and password change, etc.). • html – send in html format • text – send in plain text format
Server.DefaultNewU2FToken	The default FIDO U2F token type when users register a new FIDO token to their account. Values are: • FIDOU2F – Any FIDO U2F token • Treo- SurePassID TapID Treo token
Server.DefaultNewSoftToken	The OATH token type when user adds a new soft token to their account. Values are: • DesktopAuthenticator - SurePassID desktop token • GoogleAuthenticator – Google Authenticator token • SurePassIDAuthenticatorMobile – SurePassID authenticator soft token app.
Server.ActivateSPMobileURL	The URL that is a server endpoint for over the air activations. The form is https://<serverendpoint>/oath-ota-provision/ where <serverendpoint> is a SurePassID Auth Server endpoint or SurePassID Mobile API Connector endpoint
Allow.SoftTokenCreation	TRUE – Allow user to add soft tokens to their account FALSE – Users can only activate soft tokens that are already assigned to their account.
Allow.TokenDisable	TRUE – Allow user to disable their tokens FALSE – Users cannot disable their tokens
Allow.TokenEnable	TRUE – Allow user to enable their tokens FALSE – Users cannot enable their tokens
Allow.TokenDelete	TRUE – Allow user to delete their tokens FALSE – Users cannot delete their tokens

Server.DefaultNewSoftTokenOtpType	The soft token One Time Passcode type. Values are: • OATH Event • OATH Time • CardSecurityCodeEvent • CardSecurityCodeTime
Server.DefaultNewSoftTokenOtpWindow	The default windows size for the One Time Passcode. For default value is 30.
Server.DefaultNewSoftTokenOtpDriftUnit	The default windows size for time based the Time based One Time Passcodes. .The default value is 3.
Allow.OfflineToken	TRUE – Allow user to add offline soft tokens to their account FALSE – User cannot add offline soft tokens to their account.
Allow.U2FTokenCreation	TRUE – Allow user to add Fido U2F tokens to their account FALSE – User cannot add Fido U2F tokens to their account.
Server.DirectoryEndpointType	The directory that will be used for first factor authentication. AD = ActiveDirecoty SP= SurePassID

Notification Messages

When users require password recovery, ServicePass will send them a recovery notification email with instructions on how to reset their password. The password recovery process is slightly different based on which directory type SurePassID Authentication Server is configure for. Below are the behaviors for the supported directory options:

- SurePassID Directory, Active Directory:
  - The user is sent a recovery email.
  - The email contains a recovery link
  - The user clicks the recovery link
  - The user is presented with a web form to allow a change to his/her password.
  - The user is sent an email to alert him/her the account password has been changed.
- LDAP, Azure AD:
  - The user is sent a recovery email.
  - The email body provides the steps the user must follow to reset the password, such as, call help desk, etc.

Notification Message Customization

ServicePass is delivered with predefined emails for password recovery and password change notification. The predefined emails are meant to be sent as html email, but you can change this by setting the **Server.DefaultEmailFormat** option as defined in the Customize the System section.

You can optionally set the FROM email address (e.g. support@yourco.com) and email name (e.g., Support) in the notification by changing the **Server.DefaultFromEmailAddress** and **Server.DefaultFromEmailName** configuration values, respectively. The email address that you use must be a valid email account for your SMTP server. The SMTP server that SurePassID Authentication Server uses is defined in the SurePassID Authentication Server **Customize Email Settings** section.

The pre-defined emails are located in the ServicePass install sub-folder named account_setup. These files are:

- forgot_password_subject.txt – Forgot password email subject
- forgot_password_body.txt – Forgot password email body
- pw_change_subject.txt – Password changed email subject
- pw_change_body.txt - Password changed email body

Default Language

The system ships with a default language file that is based on US English culture (en-US). The system is Unicode based so it can support every possible language including double byte and right to left character sets.

To make these pages culturally friendly, you will need to update the language file.

The system will automatically change language to the culture of the user (which is usually set by the underlying operating system) if the appropriate culture (language file) exists for their culture. This has two important uses:

1. Provide a language-centric experience to your users across cultural boundaries.
2. Change any constant field/message in the system to match the user's language.

To change English message text, edit the settings.resx file in the App_GlobalResources folder.

For other languages, copy this file and rename it to settings.xx-yy.resx (where xx is the language and yy is the dialect; e.g., en-us for US English, en-gb for British English, fr-fr for France, fr-ca for Canada).

The system will use the language set as default in the user's browser.