Single Sign-on 5.0

Oct 01, 2015

About

Sep 07, 2011

What's New

Single Sign-on 5.0 integrates the Single Sign-on Plug-in into Citrix Receiver, simplifies the user experience, enables the Single Sign-on Plug-in to be deployed using Merchandising Server, and includes Simplified Chinese as a supported Single Sign-on Plug-in language.

- **Users access the Single Sign-on Plug-in through the Citrix Receiver icon.** Instead of seeing one or more Single Sign-on Plug-in icons in the Windows notification area, users see only the Citrix Receiver icon. The Citrix Receiver icon appears only once in the Windows notification area no matter how many Single Sign-on sessions the user has active. Users manage logon information, pause and resume Single Sign-on, determine whether Single Sign-on is paused, and submit passwords manually using menu options accessed through the Citrix Receiver icon.
  Note: If older versions of the plug-in are installed, additional icons might appear in the Windows notification area. See Installing the Single Sign-on Plug-in for more information.
- **The Single Sign-on Plug-in is required on user devices for full functionality.** Unless the Single Sign-on Plug-in is installed on the user device, users might not be able to manage logon information, pause and resume Single Sign-on, determine whether Single Sign-on is paused, or submit passwords manually. See Single Sign-on Plug-in Software Deployment Scenarios for more information.
- **Users exit the Single Sign-on Plug-in by exiting Citrix Receiver**. Users exit Single Sign-on by choosing the Exit option from the Citrix Receiver icon menu. This closes the Citrix Receiver user interface and all plug-ins accessed through it.
- **Users manage logon information using the Manage Passwords window.** The Logon Manager has been renamed the Manage Passwords window and redesigned to simplify the user experience:
  - Users access the Manage Passwords window from a menu option accessed through the Citrix Receiver icon. One Manage Passwords window appears, containing logon information for applications from all the user's sessions.
  - You configure the Manage Passwords window to display columns for one or more of these attributes of stored credentials: name, description, group, time and date last used, time and date last modified. Users can sort on each of these attributes.
  - The Manage Passwords window has no drop-down menus. The functionality formerly accessed using the options on these menus in the Logon Manager is accessed differently or has been removed:

Menu	Option	What happens to this functionality in Single Sign-on 5.0?
File	New Logon or New Logon > Add One Logon	Users store credentials manually using the Submit option available through the Citrix Receiver icon menu.
	New Logon > Add Multiple Logons	Users create multiple sets of credentials for the same application by creating the first set of credentials, copying it, and editing the copy.

Menu	Option	What happens to this functionality in Single Sign-on 5.0?
	Copy	Replaced by the Copy button in the Manage Passwords window.
	Delete	Replaced by the Remove button in the Manage Passwords window.
	Properties	Replaced by the Edit button in the Manage Passwords window.
	Exit	Users exit the Manage Passwords window using the Windows close button.
View	Icon, List, and Detail	This functionality has been removed to simplify the user experience.
	Arrange Icons By	This functionality is not available, but users can sort the columns in the Manage Passwords window by clicking on the column heading.
	Refresh	Replaced by the Refresh link in the Manage Passwords window.
	Reveal Passwords	Users view one password at a time using the Reveal Password button in the Manage Passwords window. Users cannot reveal more than one password at a time.
Tools	Account Association	Users cannot enable Account Association using the Single Sign-on Plug-in. To give user the ability to enable account association, give them access to the AccAssoc.exe utility as a published application.
	Security Question Registration	Users cannot reregister answers to their security questions using the Single Sign-on Plug-in, unless you prompt them to reregister. To give user the ability to reregister answers to their security questions without being prompted, give them access to the QBAEnroll.exe utility as a published application.
	Options > Confirm Exit	Confirmation on exit is controlled through the Citrix Receiver. The Single Sign-on Plug-in does not ask for exit confirmation.
Help	Logon Manager Help	Replaced by the Help link in the Manage Passwords window.
	About	Replaced by the About link in the Manage Passwords window.

- The Manage Passwords window has no context menu. The functionality formerly accessed using this menu in the Logon Manager is accessed differently:

Option	What happens to this functionality in Single Sign-on 5.0?
Copy	Replaced by the Copy button in the Manage Passwords window.
Delete	Replaced by the Remove button in the Manage Passwords window.
Properties	Replaced by the Edit button in the Manage Passwords window.

- **Users cannot be prompted to store credentials the first time they use Single Sign-on.** The initial credential setup option has been eliminated.
- **The Single Sign-on Plug-in can be deployed and managed using Merchandising Server.** If Citrix Receiver Updater is installed on user devices, you can deploy and manage the Single-Sign-on Plug-in with Merchandising Server.
- **The Single Sign-on Plug-in can be deployed in Simplified Chinese.**

Known Issues

See Known Issues for XenApp 6.5 for Windows Server 2008 R2 for known issues in Single Sign-on 5.0.

Get Started

Jun 07, 2013

The main components of Single Sign-on are:

- The central store
- The Single Sign-on component of the Citrix AppCenter
- The Single Sign-on Plug-in
- The Single Sign-on Service (optional)

The Central Store

The central store is a centralized repository used by Single Sign-on to store and manage user and administrative data. User data includes user credentials, security question answers, and other user-focused data. Administrative data includes password policies, application definitions, security questions, and other wider-ranging data. When a user signs on, Single Sign-on compares that user's credentials to those stored in the central store. As the user opens password-protected applications or Web pages, the appropriate credentials are drawn from the central store.

The Single Sign-on Component of the Citrix AppCenter

The Single Sign-on component of the Citrix AppCenter is the command center of Single Sign-on. Here, you configure how Single Sign-on works, which features are deployed, which security measures are used, and other important password-related settings.

The component has four main items, or nodes, in the left pane. By selecting a node, tasks specific to that node appear. These nodes are:

- User Configurations allow you to tailor particular settings for your users based on their geographic locations or business roles.
- Application Definitions provide the required information for the Single Sign-on Plug-in to supply user credentials to applications and to detect error conditions if they occur. Use the application definition templates supplied with Single Sign-on to speed this process or create your own customized definitions for applications that cannot use these templates.
- Password Policies control password length and the type and variety of characters used in both user-defined and automatically-generated passwords. Password policies also allow you to identify characters to exclude from use in passwords and whether or not previous passwords can be reused. Creating password policies consistent with your company's security policies ensures that Single Sign-on can manage password security appropriately.
- Identity Verification enables you to create security questions that provide an added layer of security to the Single Sign-on Plug-in. Security questions protect against user impersonation, unauthorized password changes, and unauthorized account unlocking. Users who enroll and answer your security questions can then verify their identity by providing the same answers when challenged. Once verified, users can perform self-service tasks to their account, such as resetting their primary password or unlocking their user account. Security questions can also be used for key recovery.

The Single Sign-on Plug-in

The Single Sign-on Plug-in submits the appropriate credentials to the applications running on the user's client device, enforces password policies, provides self-service functionality, and enables users to manage their credentials with the Manage Passwords window (formerly known as Logon Manager). In addition, the plug-in provides users with a wide array of features as determined by the administrative settings you make in the user configurations.

The Single Sign-on Service

The Single Sign-on Service runs on a Web server that provides the foundation for optional features included in this release. Install the Single Sign-on Service if you plan to implement at least one of the following modules:

- Self-Service, which allows users to reset their Windows passwords and unlock their Windows accounts
- Data Integrity, which protects data from being compromised while in transit from the central store to the Single Sign-on Plug-in
- Key Management, which provides users with the capability to recover their secondary credentials when their primary password changes, either with automatic key recovery or after answering security questions with question-based authentication
- Provisioning, which allows you to use the Single Sign-on component of the Citrix AppCenter to add, remove, or update Single Sign-on user data and credential information
- Credential Synchronization, which synchronizes user credentials among domains using a Web service

If you are not implementing the modules mentioned above, do not install the Single Sign-on Service.

Evaluate

May 11, 2015

If you are using XenApp 6.5 for Windows Server 2008 R2 to publish applications and want to use Single Sign-on 5.0 to provide password security and single sign-on access to them, this topic helps you deploy Single Sign-on quickly. The Single Sign-on deployment described here can be used to evaluate Single Sign-on or as a pilot deployment that can be expanded to include more users and applications.

Note: To simplify the deployment process, the deployment described here excludes some components, features, and options that are available when using Single Sign-on 5.0 with XenApp 6.5.
The deployment described here includes these components of Single Sign-on:

- **Central store.** The central store is a centralized repository used by Single Sign-on to store and manage user and administrative data. User data includes user credentials, security question answers, and other user-focused data. Administrative data includes password policies, application definitions, security questions, and other wider-ranging data. When a user signs on, Single Sign-on compares that user's credentials to those stored in the central store. As the user opens password-protected applications or Web pages, the appropriate credentials are drawn from the central store.
- **Single Sign-on component of the Citrix AppCenter.** For this deployment, you can use the Single Sign-on component of the Citrix AppCenter to define password policies, configure Single Sign-on to recognize applications, and create user configurations.
- **Application Definition Tool.** The Application Definition Tool has the same features as the portion of the Single Sign-on component of the Citrix AppCenter that configures Single Sign-on to recognize applications.
- **Single Sign-on Plug-in.** The Single Sign-on Plug-in is the component of Single Sign-on the users interact with. It submits the appropriate credentials to the applications running on the user's client device, enforces password policies, and enables users to manage their credentials with the Manage Passwords window. For this deployment, it is installed on each user device.

This deployment does not include the Single Sign-on Service or any of these optional features it supports:

- Self-Service, which allows users to reset their Windows passwords and unlock their Windows accounts.
- Data Integrity, which protects data from being compromised while in transit from the central store to the Single Sign-on Plug-in.
- Key Management, which provides users with the capability to recover their secondary credentials when their primary password changes, either with automatic key recovery or after answering security questions with question-based authentication.
- Provisioning, which allows you to use the Single Sign-on component of the Citrix AppCenter to add, remove, or update Single Sign-on user data and credential information.
- Credential Synchronization, which synchronizes user credentials among domains using a Web service.

Perform the tasks in this topic in the order the sections appear here.

Plan Your Deployment

- Review the system requirements for the central store, the Single Sign-on component of the AppCenter, the Application Definition Tool, and the plug-in: System Requirements.
- Review the licensing requirements for Single Sign-on and install and upgrade licenses if needed: System Requirements.
- Identify the applications you want to include. For this deployment, choose only Windows and web applications published with XenApp:
  - For Windows applications, use 32-bit Windows applications (including Java applications) such as Microsoft Outlook,

    Lotus Notes, SAP, or any password-enabled Windows application. Single Sign-on categorizes any application launched by a file with an .exe extension as a Windows application.
  - For Web applications, use Web applications (including Java applets and SAP) accessed through Microsoft Internet Explorer. Typically, Single Sign-on categorizes any application that runs in a browser as a Web application. Single Sign-on supports Web applications running on Internet Explorer Versions 6.0, 7.0, 8.0, and 9.0.
- Identify the users you want to include. Ensure that their user devices support the Single Sign-on Plug-in.
- Decide where to install the central store. The cental store for this deployment is a NTFS network share.
- Decide where to install the Single Sign-on component of the Citrix AppCenter. You can use an AppCenter that is already installed or install a new AppCenter.
- Decide whether you will install the Application Definition Tool and where to install it. If the Citrix AppCenter is not installed on the computer running an application you want to include in your deployment, install the Application Definition Tool on that computer. When you configure Single Sign-on to recognize applications, you run the applications and allow wizards within the tool to capture information about the applications.
- Plan your password policies. Password policies are rules that control how passwords are created, submitted, and managed; you apply password policies to all users or to specific groups of applications. Single Sign-on includes two standard password policies named Default and Domain. If the default values for these standard policies meet your needs for this deployment, you can use them without modification. Otherwise, you can create new policies based on them and modify these values.
  - For an overview of password policies, see Password Policies.
  - For guidelines on making your password policies secure and usable, see Password Policies.
  - To understand how Single Sign-on enforces password policies, see Enforcing Password Requirements.
  - To determine whether the default values of the password policy rules are appropriate for your applications and users, review the default values for each setting in the Password Policies reference topic and all its subtopics. The standard password policies (Default and Domain) have these default values.
- Plan your user configurations. A user configuration is a unique collection of settings, password policies, and applications that you apply to users associated with an Active Directory hierarchy (OU or individual user) or Active Directory group. A user configuration enables you to control the behavior and appearance of the plug-in software for users.
  - For an overview of user configurations and to review user configuration settings used in this deployment and their default values, see Single Sign-on 5.0 Settings Reference. Keep in mind that some options and features discussed in that topic are not used in this deployment. The overview includes the following information:
    - Basic Plug-in Interaction
    - Plug-in User Interface
    - Synchronization

      Note: Do not select allow user credentials to be accessed through the Credential Synchronization Module. The user configuration deployment does not include the Credential Synchronization module.
    - Application Support
    - Licensing
  - To protect your users credentials, see Data Protection Methods.

    Note: Use the default values for the secondary data protection settings. Other values require the Key Management module, which is not included in this deployment.

    For this deployment, you can use the default user configuration settings (except for licensing settings) initially in most environments. If your requirements change once the deployment is in use, you can edit the user configuration values.

    Settings for features not used in this deployment are disabled by default.

Create the Central Store

The Single Sign-on central store can be one of two types: Active Directory or NTFS network share. For this deployment, you create a NTFS network share because it is requires fewer permissions to create than an Active Directory central store. For advantages and considerations of a NTFS network share central store, see Choosing an NTFS Network Share.

If necessary, you can migrate users to an Active Directory central store later.

To create the NTFS network central store:

1. Load the XenApp media.
2. From the Autorun menu, select Manually install components > Server Components > Additional Features > Single Sign-on.
3. Select Central Store.
4. Select NTFS network share.

The central store is created as %SystemDrive%\CITRIXSYNC$.

Install the Single Sign-on Component of the AppCenter

The AppCenter includes the Single Sign-on component by default when installed.

To use an existing AppCenter with Single Sign-on, configure and run discovery after the central store is created.

To install a new AppCenter for use with Single Sign-on, ensure that the required Microsoft Visual C++ Redistributable Packages and Microsoft Primary Interoperability Assembles are installed, as described in System Requirements.

To install the AppCenter:

1. Load the XenApp media on the computer.
2. From the Autorun menu, select Manually install components > Common Components > Management Console. Follow the instructions.
3. Select Configure and run discovery and follow the instructions.

After configuration, the Single Sign-on component of the AppCenter is connected to the central store and you can use it to define password policies, configure Single Sign-on to recognize applications, and create user configurations.

Install the Application Definition Tool

If the Citrix AppCenter is not installed on the computer running an application you want to include in your deployment, install the Application Definition Tool to create application definitions for the application.

1. Load the XenApp media on the computer.
2. Locate the ASC_PasswordManager file in the Administration folder and run it.
3. Select Application Definition Tool. Follow the instructions.

Define Password Policies

If you determined that the default values for the standard password policies meet your needs for this deployment, you do not have to define any additional policies. Otherwise, create new policies based on the standard policies.

To create a new password policy:

1. Click Start > All Programs > Citrix > Management Consoles > Citrix AppCenter.
2. Expand the Single Sign-On node and select Password Policies.
3. From the Action menu, click Create new password policy.
4. Follow the instructions in the Password Policy Wizard.

Configure Single Sign-on to Recognize Applications

Single Sign-on recognizes and responds to applications based on the settings identified in application definitions. Application definitions provide the information necessary for the Single Sign-on Plug-in to supply user credentials to applications, and detect error conditions if they occur.

Application definitions consist of form definitions. Form definitions allow the Single Sign-on Plug-in to analyze each application as it is started, recognize certain identifying features, and determine if the starting application requires the plug-in to perform some specific action, such as:

- Submit user credentials at a logon prompt
- Negotiate a credential changing interface
- Process a credential confirmation interface

Although most applications and their corresponding application definitions use only two forms for managing user credentials, you can define as many forms as necessary in a single application definition.

You can create these types of user credential management forms:

- Logon form
  Identifies the logon interface to an application and manages the actions required to gain access to the associated application.

- Password change form
  Identifies the password change interface to an application and manages the actions required to change the user password to the associated application.

- Successful password change form
  Identifies the password change interface to an application and manages the actions required to acknowledge the successful password change for the associated application.

- Failed password change form
  Identifies the unsuccessful password change interface to an application and defines the actions to take when a credential change operation is unsuccessful.

You create application definitions by using the wizards available from the AppCenter or the Application Definition Tool. When the application you want to define is running or available in a browser window, these wizards help capture the information you need for the application definition. To create an application definition, you must be able to access the application from the computer where the application definition is created.

Because application signatures can vary depending on the underlying operating system, test application definitions on all operating systems on which they will run.

Application templates are available for some applications. These templates simplify the process of adding application definitions to your Single Sign-on deployment by supplying most of the information needed to create an application definition. For more information about application templates, see Application Templates.

To create a Windows application definition

To create application definitions for a Windows application, run the application on a computer on which you launch the Application Definition Wizard from the Citrix AppCenter of the Application Definition Tool. You navigate to the form within the application that requires a user credential management event (user logon, change password, successful password change, or failed password change) while running the wizard.

For an overview of considerations for Windows application definitions, see Windows Type Application Definitions.

1. Start the application.
2. Prepare to start the Application Definition Wizard:
   - From the AppCenter: Click Start > All Programs > Citrix > Management Consoles > Citrix AppCenter. Expand the Single Sign-On node and select Application Definitions.
   - From the Application Definition Tool: From the AppCenter: Click Start > All Programs > Citrix > Single Sign-On > Application Definition Tool.
3. Select Create application definition.
4. Ensure that Windows and Create new are selected and click Start Wizard.
5. Enter the name of the application as you want it to appear in the central store. Optionally, enter a description. Click Next.
6. Click Add Form. This launches the Form Definition Wizard.
7. If you haven't already done so, navigate to the application's user logon, change password, successful password change, or failed password change form.
8. From the Identify Form page of the Form Definition Wizard, click Select.
9. In the Window selector that appears, select the application you are creating the definition for. A flashing border appears around the application's prompt.
10. In the Name form page, enter a name for the form and select the form type. Click Next.
11. In the Window selector, click OK.
12. In the Identify form page, click Next.
13. In the Define forms actions page, configure the credential fields and buttons that you want to appear in the form:
    1. Click the Set/Change hyperlink associated with a specific user credential. This action opens the Configure Control Text dialog box used to identify the control to receive the selected credential.
    2. Select the control type candidate to receive the credential. As the different candidates are selected, the associated control type is highlighted on the application with a flashing border.
    3. Repeat this action for all the user credentials required by the form and for the button required to submit the form. Some forms require domains or other user-configurable credentials that must be successfully submitted to process the form. To accommodate these requirements, two custom fields are available. Assign special-requirement credentials to these fields. The names associated with these fields are defined on the Name custom fields page of the Application Definition Wizard after the form is defined.

       Note: Not all the credentials identified in the top of the Define form actions page must be configured.
14. If your application requires additional forms, use the wizards to create them.

To create a Web application definition

To create application definitions for a Web application, run the application on a computer on which you launch the Application Definition Wizard from the Citrix AppCenter of the Application Definition Tool. You navigate to the form within the application that requires a user credential management event (user logon, change password, successful password change, or failed password change) while running the wizard.

1. Start the application.
2. Prepare to start the Application Definition Wizard:
   - From the AppCenter: Click Start > All Programs > Citrix > Management Consoles > Citrix AppCenter. Expand the Single Sign-On node and select Application Definitions.

   - From the Application Definition Tool: From the AppCenter: Click Start > All Programs > Citrix > Single Sign-On > Application Definition Tool.
3. Select Create application definition.
4. Ensure that Web and Create new are selected and click Start Wizard.
5. In the Identify application page that appears, enter the name of the application as you want it to appear in the central store. Optionally, enter a description. Click Next.
6. Click Add Form. This launches the Form Definition Wizard.
7. In the Name form page: Click Next.
   1. Enter a name for the form.
   2. Select the form type.
   3. Ensure that No special action is selected.
   4. Click Next.
8. If you haven't already done so, navigate to the application's user logon, change password, successful password change, or failed password change form.
9. From the Identify form page, click Select. This launches the Web Form Wizard.
10. In the Web page selector that appears, select the application you are creating the definition for. Click OK. A flashing border appears around the web page displaying the application's credential form.
11. Enter a name for the form and select the form type. Click Next.
12. In the Identify form page, two check boxes are available to manage how to interpret identified URLs. Select the appropriate check boxes and click Next.
    - Strict URL matching
      Select this check box to recognize only user credential management events from Web applications that are started using the specified URL(s). Some URLs may contain dynamic data such as session management identifiers, application parameters, or other identifiers that can change for each instance. In these circumstances, using strict matching results in the URL not being recognized.

    - Case-sensitive URL
      Select this check box to use exact case matching URL(s).

13. In the Define forms actions page, configure the credential fields and buttons that you want to appear in the form:.
    1. Click the Set/Change hyperlink associated with a specific user credential. This action opens the Configure Field Text dialog box used to identify the field to receive the selected credential. If the form is already open, this dialog box displays all the possible candidates for the field type associated with the selected user credential or submit option.
    2. If the application credential form is not currently open, start the application and navigate to the correct user credential form. Then select Refresh . After the application form is selected, this dialog box is populated with field type candidates that are appropriate for the selected user credential.
    3. Select the field type candidate to receive the credential. As the different candidates are selected, the associated field type is visibly highlighted on the application to make it easier to identify the field type that is to receive the identified user credential or submit button.
    4. Repeat this action for all the user credentials required by the form and for the button required to submit the form. Some forms require domains or other user-configurable credentials that must be successfully submitted to process the form. To accommodate these requirements, two custom fields are available. Assign special-requirement credentials to these fields. The names associated with these fields are defined on the Name custom fields page of the Application Definition Wizard after the form is defined.

       Note: Not all the credentials identified in the top of the Define form actions page must be configured.
14. If your application requires additional forms, use the wizards to create them.

To add an application definition for an application with an available template

The Application Definition Wizard helps you locate application templates and add them to your deployment.

1. Prepare to start the Application Definition Wizard:
   - From the AppCenter: Click Start > All Programs > Citrix > Management Consoles > Citrix AppCenter. Expand the Single Sign-On node and select Application Definitions.
   - From the Application Definition Tool: From the AppCenter: Click Start > All Programs > Citrix > Single Sign-On > Application Definition Tool.
2. Select Manage templates.
3. View the list of templates to see if the application you want appears. You can also click the link to download more applications from the web and import them to the list.
4. Select the application template you want to add and click Create Application Definition.
5. Use the wizard to edit the forms for the application or accept the default values.

Create User Configurations

1. Click Start > All Programs > Citrix > Management Consoles > Citrix AppCenter.
2. Expand the Single Sign-on node and select User Configurations.
3. Click Add new user configuration.
4. Enter the name of the application as you want it to appear in the central store. Optionally, enter a description.
5. Specify how you will associate this user configuration to users.
   You have two choices: associate users according to Active Directory hierarchy (OU or individual user) or Active Directory Group. If necessary, you can associate the user configuration with a different hierarchy or group later, by clicking Move user configuration in the Action menu.

   Important: How you organize your Active Directory environment might affect how user configurations operate. If you use both (Active Directory hierarchy and group) and a user is located in both containers, the user configuration associated with the hierarchy takes precedence and is the one used. This scheme is considered a mixed environment. Also, if a user belongs to two Active Directory groups and each group is associated with a user configuration, the user configuration with the highest priority takes precedence and is the one used.

   Associating user configurations to groups is supported only in Active Directory domains that use Active Directory authentication.

6. From the Choose applications page, add the applications for the user configuration. When you click the Add button, a dialog box showing the application definitions you created previously appears.
7. Use the Configure Single Sign-on Plug-in interaction page to determine the user experience for all plug-in software users in your environment.
8. Select a license server and licensing model at the Configure licensing page.
9. Use the Select data protection methods page to select the data protection methods to protect user credentials based on the various authentication methods your users are authorized to use.

Install the Single Sign-on Plug-in

The Single Sign-on Plug-in runs on the XenApp server and provides credentials and access to published applications. The plug-in also runs on each user device, submitting credentials to applications and enabling users to manage their credentials.

Installation considerations:

- After installing the plug-in on a supported operating system that uses the Microsoft Graphical Identification and

Authentication (GINA) Windows component, you must restart the device. Ths includes Windows XP, Microsoft Windows XP Embedded, Microsoft Windows Fundamentals for Legacy PCs, Microsoft Windows Server 2003 R2, and Microsoft Windows Server 2003 with Service Pack 2.
WinLogon uses the GINA controls for the dialog box that users see when they press the key combination CTRL+ALT+DEL. The dialog box collects the data needed to perform authentication. XenApp, the Single Sign-on Plug-in, and the Novell NetWare client interact with or require the replacement of the GINA dynamic link library (DLL). You might be required to install or uninstall software in a specific order to preserve proper GINA chaining. By installing the Single Sign-on Plug-in last, you ensure that the Single Sign-on GINA is called first by the Winlogon process.

- After the installation completes (and the device restarts, if needed), the Citrix Receiver icon appears in the system tray.
- After installing the plug-in, if you configure or change Citrix licensing information, restart the plug-in to apply the changes.

To install the Single Sign-on Plug-in on a user device or on a server with XenApp installed:

1. Load the XenApp media on the computer or server.
2. From the Autorun menu, select Manually install components > Server Components > Additional Features > Single Sign-on > Single Sign-on Plug-in.
3. Follow the instructions.

Get User Started Using Single Sign-on

Before end users begin using Single Sign-on, review the end user help available through the Single Sign-on interface. Inform your users how Single Sign-on works and which features are available to them in this deployment.

System Requirements

May 09, 2015

Computers in your Single Sign-on environment require the following system software:

Software Component	Required by	Available from...
Microsoft Windows Installer 3.0 or later (automatically included during Autorun installation)	All	- Support folder on the Single Sign-on installation media - http://www.microsoft.com
Microsoft .NET Framework 3.5 Service Pack 1(automatically included during Autorun installation)	- Single Sign-on Service - Single Sign-on component of the AppCenter - Application Definition Tool	Support folder on the Single Sign-on installation media
Microsoft Internet Explorer Version 6.0, 7.0, 8.0 or 9.0 (non-protected mode)	Users accessing Single Sign-on-enabled Web applications	http://www.microsoft.com
ASP.NET	Single Sign-on Service	http://www.asp.net/
- For 32-bit computers: Microsoft Visual C++ 2005 Redistributable Package (x86) Service Pack 1 - vc80_vcredist_x86.exe - For 64-bit computers: Microsoft Visual C++ 2005 Redistributable Package (x64)Service Pack 1 - vc80_vcredist_x86.exe - vc80_vcredist_x64.exe	Single Sign-on console component, service, or plug-in—When installing the console component, service, or plug-in from a command prompt onto a Windows Vista, Windows Server 2008, or Windows Server 2008 R2 computer	Support folder on the Single Sign-on installation media
- For 32-bit computers: Microsoft Visual C++ 2008 Redistributable Package (x86) Service Pack 1 - vc90_vcredist_x86.exe - For 64-bit computers: Microsoft Visual C++ 2008 Redistributable Package (x86) Service Pack 1 - vc90_vcredist_x86.exe - vc90_vcredist_x64.exe	Single Sign-on console component, service, or plug-in—When installing the console component, service, or plug-in from a command prompt onto a Windows Vista, Windows Server 2008, or Windows Server 2008 R2 computer	Support folder on the Single Sign-on installation media
Microsoft Primary Interoperability	Single Sign-on console component	Support folder on the Single Sign-on

Software Component	Required by	Available from...
Assemblies • vs90_piaredist.exe	—When installing the console component from a command prompt onto a Windows Vista, Windows Server 2008, or Windows Server 2008 R2 computer	installation media
Internet Explorer Enhanced Security Configuration	Single Sign-on Plug-in—Disable Internet Explorer Enhanced Security Configuration when installing the plug-in onto a Windows Server 2003, Windows Server 2008, or Windows Server 2008 R2 computer. If left enabled, the plug-in does not respond to Web application definitions.	

Single Sign-on Component Requirements

Single Sign-on Component	Supported Environment or Microsoft Windows Operating System	Supported Language	Hardware Requirements
Central store	• Active Directory • NTFS File Share	• English • German • French • Spanish • Japanese	30KB disk space per user
Single Sign-on component of the AppCenter	• Microsoft Windows 7 Service Pack 1—32-bit and 64-bit • Microsoft Windows 7—32-bit and 64-bit • Microsoft Windows Vista Service Pack 2 (Business Edition, Ultimate Edition, Enterprise Edition)—32-bit and 64-bit • Microsoft Windows Vista (Business Edition, Ultimate Edition, Enterprise Edition)—32-bit and 64-bit • Windows XP Service Pack 3—32-bit • Microsoft Windows XP Professional, Service Pack 2—32-bit • Microsoft Windows XP Professional x64 Edition—64-bit • Windows Server 2008 R2 Service Pack 1—64-bit • Microsoft Windows Server 2008 R2—64-bit • Microsoft Windows Server 2008 (Standard Edition, Enterprise Edition, Datacenter Edition)—32-bit and 64-bit	• English • German • French • Spanish • Japanese	• 64MB RAM • 60MB disk space

Single Sign-on Component	Supported Environment or Microsoft Windows Operating System	Supported Language	Hardware Requirements
	• Microsoft Windows Server 2003 R2 (Standard Edition, Enterprise Edition, Datacenter Edition)—32-bit and 64-bit • Microsoft Windows Server 2003 with Service Pack 2 (Standard Edition, Enterprise Edition, Datacenter Edition)—32-bit and 64-bit		
Plug-in	• Microsoft Windows 7 Service Pack 1—32-bit and 64-bit • Microsoft Windows 7—32-bit and 64-bit • Microsoft Windows Vista Service Pack 2 (Business Edition, Ultimate Edition, Enterprise Edition)—32-bit and 64-bit • Microsoft Windows Vista (Business Edition, Ultimate Edition, Enterprise Edition)—32-bit and 64-bit • Windows XP Service Pack 3—32-bit • Microsoft Windows XP Professional, Service Pack 2—32-bit • Microsoft Windows XP Professional x64 Edition—64-bit • Microsoft Windows XP Embedded • Windows Server 2008 R2 Service Pack 1—64-bit • Microsoft Windows Server 2008 R2—64-bit • Microsoft Windows Server 2008 (Standard Edition, Enterprise Edition, Datacenter Edition)—32-bit and 64-bit • Microsoft Windows Server 2003 R2 (Standard Edition, Enterprise Edition, Datacenter Edition)—32-bit and 64-bit • Microsoft Windows Server 2003 with Service Pack 2 (Standard Edition, Enterprise Edition, Datacenter Edition)—32-bit and 64-bit	• English • German • French • Spanish • Japanese • Simplified Chinese	• 10MB RAM • 25MB disk space (if optional features are not installed) • 35MB disk space (if optional features are installed)
Service	• Windows Server 2008 R2 Service Pack 1—64-bit • Microsoft Windows Server 2008 R2—64-bit • Microsoft Windows Server 2008 (Standard Edition, Enterprise Edition, Datacenter Edition)—32-bit • Microsoft Windows Server 2003 R2 (Standard Edition, Enterprise Edition, Datacenter Edition)—32-bit • Microsoft Windows Server 2003 with Service Pack 2 (Standard Edition, Enterprise Edition, Datacenter Edition)—32-bit	• English • German • French • Spanish • Japanese	• 128MB RAM • 30MB disk space
Application Definition Tool	Same as plug-in	• English • German	Same as plug-in

Single Sign-on Component	Supported Environment or Microsoft Windows Operating System	Supported Language	Hardware Requirements
		- French - Spanish - Japanese	

Note: Single Sign-on is not supported on Microsoft Windows XP Home Edition.
Hot Desktop is supported only on:

- Microsoft Windows XP Professional, Service Pack 2—32-bit
- Microsoft Windows XP Embedded

Hot Desktop is not supported on 64-bit operating systems or any server operating systems.

Licensing Requirements

Install the license server and add licenses before installing Single Sign-on.

To run this release, ensure that you have the latest version of the license server installed. If you are running an earlier version of the license server, upgrade your license server.
Important: Locally installed instances of the Single Sign-on Plug-in do not require a separate license for users who have access to hosted applications in a Citrix XenApp, Platinum Edition environment.

Disconnected Mode

If you have users who work disconnected from the license server for extended periods of time, such as mobile users with laptops, you must specify a disconnected mode period for these users. The disconnected mode period is specified as part of the licensing settings in the user configuration. The disconnected mode period specifies two aspects of licensing behavior:

- The amount of time the user can be disconnected from the license server without entering the licensing grace period. When the disconnected mode period expires, the users employing the associated user configuration lapse into the 30 day licensing grace period.
- The amount of time until a checked out license which is being used in disconnected mode is returned to the pool of available licenses on the license server, regardless of whether or not the product reconnects to the license server. If a license is checked out and the disconnected mode associated with that license expires before the license is checked in, the license server automatically checks the license back in so the license is available again. For example, if a laptop running Single Sign-on is lost and never reconnects with your organization's network, the license server automatically checks the license back in at the end of the disconnected mode period.

When you set the disconnected mode, you are actually specifying how long you want to wait until the license is returned to the pool of available licenses.

Consider setting long disconnected mode periods for users who do not connect to your organization's network regularly, such as sales personnel who work remotely. However, keep in mind you cannot retrieve any checked out licenses, even from lost or broken equipment, for the duration of this period.

Mixed License Types

Depending on your Single Sign-on environment and enterprise needs, you might be using previously purchased stand-alone Single Sign-on licenses. For example, you might create user configurations based on the named user license model for mobile users who use the Single Sign-on Plug-in through a desktop computer and laptop computer. You might also create user configurations based on the concurrent user license model for Hot Desktop users.

In some cases, all of your named user licenses might be in use, making Single Sign-on unavailable for some users. If so, you can use any available concurrent user licenses in your user configuration to be consumed offline.

Plan

Mar 24, 2011

Before installing Single Sign-on, you should plan your environment. This includes determining the type of central store to use, the Single Sign-on-enabled applications in your enterprise, which Single Sign-on features to use, and establishing password policies.

A Single Sign-on environment can include the following:

- Shared network folders or Active Directory containing the central store.
- One or more computers running the Single Sign-on component of the Citrix AppCenter.
- User computers running the Single Sign-on Plug-in.
- A dedicated server hosting the Single Sign-on service with one or more feature modules installed on it.
- Citrix XenApp environment hosting the Single Sign-on Plug-in.
- Authentication devices such as smart cards.
- Single Sign-on features such as Hot Desktop and key management.

Central Store Types

May 09, 2015

Single Sign-on uses a repository known as the central store to store and retrieve information about your users and your environment. Single Sign-on relies on the data in the central store to perform all default and configured Single Sign-on functions. You can create a central store automatically as part of the Single Sign-on installation process or manually by using the central store setup utilities.

The central store contains user data and administrative data:

- User data in the central store includes user secondary credentials, security questions and answers, service-related data (for example, provisioned data, question-based authentication data, key recovery enrollment, and so on), and user Windows registry data associated with Single Sign-on.
- Administrative data in the central store includes application definitions, password policies, security questions, and other settings made through the console for Single Sign-on features and components.

The central store basically enables the plug-in software running on a user computer or computer running Citrix XenApp to communicate with the central store and services, and to provide user credentials to applications to which the user is granted access.

The plug-in software maintains a local store on the user computer. The local store contains only the user's secondary credentials, key recovery information, and security questions and answers (if applicable). It synchronizes with the central store to allow users to roam throughout the enterprise and always have access to saved user credentials.

The central store can be one of the following types:

- Active Directory
  The central store uses the Active Directory environment and objects to store and update Single Sign-on data.
- NTFS network share
  The central store uses a Windows network file share to store the Single Sign-on data.

If necessary, you can migrate users from one central store type to another.

Choosing an Active Directory Central Store

Choosing to use Active Directory as your central store enables you to leverage the convenience of your existing Active Directory user authentication and object administration. For example, you can apply user-specific settings to any level in a domain—domain, organizational unit, group, or user.

Two new classes and two attributes are added to the Active Directory schema when you create an Active Directory central store:

Class	Description
citrix-SSOConfig	Describes the object containing data for the plug-in software settings, synchronization state, and the application definitions and the first-time plug-in software use behavior.This class includes the following attributes: citrix-SSOConfigData - contains the actual data and citrix-SSOConfigType - specifies the data type
citrix-SSOSecret	Describes the secret data object used to authenticate a Single Sign-on user. This class

Class	Description
	includes the following attribute: citrix-SSOSecretData - contains encrypted credential data for an application and Account Self-Service password reset data

Note: See the CitrixMPMSchema.xml file in the \Tools folder on the installation media for more information about these classes and attributes.

In general, choose Active Directory as your central store if you:

- Can successfully extend your Active Directory schema without affecting your enterprise
- Already implement best practices for Active Directory backup and restore as recommended by Microsoft (although this is not a requirement)
- Prefer the high availability that is built in to Active Directory to be extended to the central store data

Advantages of an Active Directory Central Store

The following are advantages of using an Active Directory central store:

- Active Directory includes built-in failover and redundancy, so additional measures for disaster recovery are not needed
- Active Directory replication helps to distribute central store administrative and user data across your enterprise
- No additional hardware is needed when using an Active Directory central store

Active Directory Central Store Considerations

Consider the following before using an Active Directory central store:

- You must extend your schema when using an Active Directory central store, which requires careful planning and implementation. Extending the schema affects the entire forest.
- You might want to extend the schema and create your Active Directory central store during non-peak usage hours. Your Active Directory replication cycle latency affects how quickly these changes are copied to all domain controllers in the forest.
- Inter-site replication of central store data across large enterprises using WANs requires you to configure replication correctly to reduce latency. (However, intra-site replication typically introduces less latency.)

Choosing an NTFS Network Share

Choosing to use an NTFS network share as your central store enables you to leverage the convenience of your existing Active Directory user authentication and tree structure without having to extend the Active Directory schema. For example, you can apply user-specific settings to any level in a domain—domain, organizational unit, group, or user.

Important: Use a hidden share for the central store in this case.

Single Sign-on creates a shared folder named CITRIXSYNC$ with two subfolders named People and CentralStoreRoot.

The People folder contains a subfolder for each user and includes the appropriate read and write permission properties for the user. The CentralStoreRoot folder contains administrative data.

Advantages of an NTFS Network Share

The following are advantages of using an NTFS network share:

- You can emulate the look and feel of an Active Directory central store without having to extend your Active Directory schema. Yet you can take advantage of your existing Active Directory hierarchy or groups.
  Note: Associating user configurations to groups is supported only in Active Directory domains that use Active Directory

authentication.
- User data is always up-to-date, because it is stored in a central location and avoids any data replication latency associated with Active Directory.
- You can load balance your shares among multiple computers that can each host an NTFS network share for higher availability.
- NTFS network share helps reduce the authentication task workload from your Active Directory environment.
- Single Sign-on enables you to migrate your NTFS shared folder central store to an Active Directory central store if you decide later to implement an Active Directory central store.

NTFS Network Share Considerations

Consider the following before using an NTFS network share:
- You might need additional hardware to host the central store.
- You need to back up central store files and folders (including their related permissions) regularly. Ensure that you also maintain and implement disaster recovery plans where you replicate files and folders for site recovery.
- Your enterprise network topology might require users (and the Single Sign-on Plug-in software) to transfer user data across one or more WAN links. In this case, consider implementing the Distributed File System technology included as part of Microsoft Windows Server 2003 and 2008. The Microsoft Web site http://support.microsoft.com describes the Distributed File System technology in detail.

Using Account Association with Multiple Central Stores and User Account Credentials in a Multiple Domain Enterprise

Administrators can create multiple central stores in enterprises that contain multiple domains. In fact, you can use more than one type of central store in these environments. For example, you can associate user configurations with an NTFS network share central store in one domain and an Active Directory central store in another domain.

Because companies might maintain multiple Windows domains, users might also have more than one Windows account. Single Sign-on includes a feature known as Account Association to allow a user to log on to any application from one or more Windows accounts. Because Single Sign-on typically binds user credentials to a single account, the credential information is not synchronized automatically among multiple accounts that a user owns.

However, administrators can configure Account Association to synchronize user credentials by using the Credential Synchronization Module. Users with Account Association configured have access to all applications from any of their accounts in their Single Sign-on environment. When user credentials are changed, added, or removed from one account, the credentials are synchronized automatically with each of the user’s associated accounts.

Without Account Association, users with multiple Windows accounts are forced to manually change their logon information separately from each Windows account.

To allow users to synchronize credentials by using Account Association, give them access to AccAssoc.exe as a published application.

Advantages of Using Account Association

- Account Association can help increase productivity and reduce support calls by synchronizing user credentials to help reduce logon maintenance or failures.
- Accounts can be synchronized across different central store types. That is, a user account configured to use Active Directory as the central store can synchronize with an associated user account that is configured to use an NTFS network share.

- Accounts can also be synchronized across different user configuration associations. For example, a user configuration can be associated with an Active Directory hierarchy (OU or user) in one domain and associated with an Active Directory group in another domain.
- Accounts can also be synchronized across different user configuration associations in the same domain and within the same central store.
- Trust relationships between domain controllers are not necessary to use Account Association.

Consider the following before configuring Account Association:

- Account Association is not compatible with smart cards when smart cards are used as the primary authentication mechanism to log on to Windows.
  Note: The user configuration in each domain might have different password policies that might block access to a resource, but Account Association synchronizes user credentials only, not user configuration policies. Consider how you compose password policies in your enterprise.
- Each associated domain account must use Single Sign-on.
- Application definition names must be the same in each user configuration for the Account Association feature to synchronize credentials.
- User credentials are shared only for applications specified in application definitions created by the Single Sign-on administrator.
- As part of the Single Sign-on Service, the Credential Synchronization Module is a Web service available through a secure HTTP connection, so this module must be accessible from all computers in your enterprise using Account Association.

Password Policies

May 09, 2015

Password policies are rules that control how passwords are created, submitted, and managed. The Single Sign-on installation includes two standard password policies named Default and Domain, which cannot be deleted. You can copy these policies and make modifications to suit your enterprise policies and regulations.

Default Password Policy

Single Sign-on applies the Default policy to password-enabled applications used in your enterprise (except for those that require user domain credentials). This policy is applied to any application that is not defined by an administrator (by using the application definition feature in the console) or any application that is not part of an application group.

When a user adds credentials to the Manage Passwords window (formerly known as Logon Manager) for an application that does not have a corresponding application definition, Single Sign-on applies the Default policy to manage that application.

Domain Password Policy

Typically, an administrator creates an application group and selects the Domain policy to be applied to the applications in that group. Single Sign-on then applies the Domain policy to those applications that require the user's domain credentials for access. The Domain policy can be modified or copied to reflect your enterprise's Active Directory or NT domain policies for user accounts.

If you want an application group to be treated as a domain password sharing group, you must apply the Domain policy to that application group. An application group is a collection of defined applications associated with one or more user configurations, including the policy to manage the applications.

Custom Password Policies

You can create password policies as needed: you can apply one policy for your domain sharing group, create individual policies to apply to individual groups of applications to secure them further, and so on.

When creating a custom password policy or modifying existing policies, ensure that your enterprise requirements and application requirements match. For example, if you create a policy that does not at least match an application's requirements, your users might not be able to authenticate to that application.

In general, password policies can specify restrictions such as the following:

- A minimum and maximum number of characters for a password
- Alphabetical and numerical character usage
- Number of times a character can be repeated
- Excluding or requiring which characters or special characters can be used
- Whether or not users can view their stored passwords
- How many times users can try entering their password correctly
- Password expiration parameters
- Password history and password exceptions

Password Policy Considerations

Consider the following before establishing password policies:

- Consider your security requirements in the context of ease-of-use for your users. Overly restrictive passwords might be hard for users to create, implement, or recall.
- Because Single Sign-on is secure by design, the Default password policy defines the minimum level of password security recommended by Citrix for securing most Single Sign-on enabled applications. You can modify these settings according to your enterprise policies and regulations.
- Because Single Sign-on applies the Default password policy to user-added applications, ensure that you configure the Default policy to be as broad as needed to accept passwords for those applications for which you allow passwords to be stored.
- When users change their passwords, Single Sign-on can be configured through a user configuration setting to check the old password against the new password. This helps prevent users from reusing passwords for the same application twice in a row.
- Users might have a single password that is used for multiple applications (in a suite of products, for example). This scheme is known as password sharing, where the same authentication authority is used for the applications.
  While the other credentials for those applications (such as user name and custom fields) might be different, the user's password is the same. In this case, create an application group that is a password sharing group to ensure that the plug-in software manages the password for all applications in the group as a single entity. When the password is changed in one of the applications, the plug-in software ensures that the password change is reflected in the stored credentials for all applications in the group.
- Domain password sharing groups differ from other password sharing groups because the user's domain password is used as the master password for the application group. When the user changes the domain password, the plug-in software ensures that the change is reflected in the credentials for all other applications in the group. Only the domain password can be changed; users cannot initiate password changes on any of the other applications in the group unless the administrator removes the application from the domain password sharing group.

Application Definitions

Feb 06, 2011

As the Single Sign-on administrator, you can create an application definition or modify an application definition template for each application that you want Single Sign-on to manage for your users. You create application definitions by using the console or the stand-alone Application Definition Tool that can be installed on non-console workstations.

You can also allow users to add their credentials to Single Sign-on for any of their client-side applications that it detects, according to settings in user configurations. The plug-in software can detect and respond to logon changes for most applications, including the following application types:

Application Types	**Description**
Windows	32-bit Windows applications (including Java applications) such as Microsoft Outlook, Lotus Notes, SAP, or any password-enabled Windows application
Web	Web applications (including Java applets and SAP) accessed through Microsoft Internet Explorer
Terminal Emulator	Applications that you access through a HLLAPI-compliant terminal emulator. (Single Sign-on does not support 64-bit terminal emulator sorftware.)

The plug-in software responds according to application definitions that you create from scratch or copy from existing templates. An application definition:

- Enables the plug-in software to recognize and respond to applications and the forms used by the applications to process user credentials
- Consists of a set of identifiers that establish parameters to accomplish this recognition and response

Within each definition, you create logon and password-related forms required by the application to enable access. The application definition wizards can help you create a definition if you open the application; the wizards can detect the forms and fields of most applications by using Single Sign-on's window-matching capabilities.

Tip: Single Sign-on includes default application definition templates for a variety of Citrix applications or application features. Additional templates are available by searching the Citrix Support Web site.

Smart Cards

Mar 24, 2011

Citrix has tested smart cards that meet Standard 7816 of the International Organization for Standardization (ISO) for cards with electrical contacts (known as a contact card) that interface with a computer system through a device called a smart card reader. The reader can be connected to the host computer by the serial, USB, or PC Card (PCMCIA) port.

Citrix supports the use of PC/SC-based cryptographic smart cards. These cards include support for cryptographic operations such as digital signatures and encryption. Cryptographic cards are designed to allow secure storage of private keys such as those used in Public Key Infrastructure (PKI) security systems.

These cards perform the actual cryptographic functions on the smart card itself, meaning the private keys never leave the card. In addition, smart cards provide two-factor authentication for increased security: the card and the user's pin number. When these items are used together, the cardholder can be proven to be the rightful owner of the smart card.

Smart Card Software Requirements

Consult your smart card vendor or integrator to determine detailed configuration requirements for your specific smart card implementation. The following components are required on the server or client:

- PC/SC software
- Cryptographic Service Provider (CSP) software
- Smart card reader software drivers

Your Windows server and client operating systems might already include PC/SC, CSP, or smart card reader drivers. See your smart card vendor for information about whether these software components are supported or must be replaced with vendor-specific software.

To use smart cards in a Windows Server 2008 or Windows Vista environment, your central store must be created with or updated by a Single Sign-on 4.5 (formerly Password Manager) or later console and Microsoft Data Protection API (requires roaming profiles) must be selected in your user configurations.

Requiring Identity Verification

May 09, 2015

Depending on user configuration settings, you might require users to verify their identities when the following events occur:

- Users change their authentication types; for example, a user might switch between smart card and password authentication (you can create a user configuration that requires initial verification only when switching between authentication types)
- An administrator changes a user's primary password
- Users reset their primary password using Account Self-Service
- Users unlock their domain account using Account Self-Service
- Users change their primary password on a device that does not have the plug-in software installed and then log on to a device where the plug-in software is installed

Single Sign-on can be configured to verify the user's identity to ensure that the user is authorized to use Single Sign-on. You can select one of two identity verification methods:

Method	Description
Previous Password	In this case, users verify their identities by entering their previous primary password.
Security questions (also known as question-based authentication)	In this case, you create a questionnaire that contains as many questions and question groups as you want to make available to users. You can use the default questions Single Sign-on provides or create your own.

Caution: When previous password is the only identity verification method available to your users, users who forget their previous primary password are locked out. An administrator must then use the Single Sign-on component task Reset User Data to enable the users to reenroll. An administrator might also need to reset the passwords in the user's applications.

Verifying User Identity by Using Security Questions (Question-Based Authentication)

Single Sign-on enables you to use question-based authentication to verify user identity. Single Sign-on includes four questions (in English, French, German, Japanese, Simplified Chinese, and Spanish) that you can use for this purpose.

You can use question-based authentication:

- As part of a user's Security Question Registration during the first-time plug-in software enrollment
- After enrollment, if you configured Account Self-Service to allow users to change their primary credentials or unlock their accounts

When users change their primary passwords, you can confirm your users' identities by prompting them to answer security questions in the form of a questionnaire you create. This questionnaire appears the first time your users launch the plug-in software. Users answer the required number of security questions and can be prompted to reenter this information at specific password change events.

To allow users to reregister answers to their security questions without being prompted, give them access to QBAEnroll.exe as a published application.

If you choose not to set up security questions, users are prompted for their previous primary password when they first log on and when they change their primary password. You can also allow users to choose the method they prefer to use when authenticating (previous passwords or security questions).

Recovering or Unlocking User Credentials Automatically

Important: Automatic key management is not as secure as other key recovery mechanisms such as security questions and previous password.
You can configure Single Sign-on to bypass identity verification and retrieve user credentials (that is, encryption keys associated with the user data) automatically by installing the Single Sign-on Service and using the Key Management Module.

The basic workflow to use automatic key management is as follows:

1. Install the Citrix Single Sign-on Service with the Key Management Module.
2. Create or edit user configurations and select the key recovery method that allows automatic key management without identity verification. This option is available as part of the Secondary Data Protection property in the user configuration.

Planning Your Single Sign-on Plug-in User Configurations

Feb 07, 2011

A user configuration is a unique collection of settings, password policies, and applications that you apply to users associated with an Active Directory hierarchy (organizational unit or an individual user) or Active Directory group (except for distribution groups and Domain Local groups in Active Directory mixed mode, which are not supported). A user configuration enables you to control the behavior and appearance of the plug-in software for users.

User configurations set your user information, application definitions, password policies, and identity verification methods. You must also specify license information (license server and license type) in each user configuration. Therefore, your users cannot use the plug-in software until you establish their user configuration settings.

Before you create your user configurations, ensure that you already created or defined the following:

- Your central store
- Optional service modules
- Application definitions
- Password policies
- Security questions (optional)

User configurations consist of the following:

- Users associated with an Active Directory domain hierarchy (organizational unit or individual user) or group.
- Data protection methods.
- Application definitions you created, which you can combine into an application group when you create a user configuration.
- Password policies associated with any application groups. (While creating a user configuration, you can create one or more application groups to associate with a user configuration. You can also add an application group to a user configuration after you create the user configuration.)
- Self-service features (account unlock and password reset) and key management options (use of previous passwords, security questions you create for your users, and automatic key management).
- Settings for options such as Hot Desktop, credential provisioning, and application support.

Associating user configurations to groups is supported only in Active Directory domains that use Active Directory authentication.

Consider the following when planning your Single Sign-on Plug-in user environment:

- If you need to apply the same user configuration settings to a different group of users, duplicate the user configuration in the console and modify the settings accordingly.
- How you organize your Single Sign-on user environment might affect how user configurations operate. That is, you associate user configurations in your Single Sign-on environment with an Active Directory hierarchy (OU or users) or an Active Directory group. If you use both (hierarchy and group) and a user is located in both containers, the user configuration associated with the hierarchy takes precedence and is the one used. This scheme is considered a mixed environment.
- The user configuration information maintained in the central store takes precedence over information stored in the local store (that is, user data stored on a user's computer). The local store user data is mostly used when the central store is

not available or offline.

Enabling the Sharing of the Same Resources or Workstation Among Many Users (Hot Desktop)

Feb 06, 2011

The Hot Desktop feature allows users to share workstations efficiently and securely. With Hot Desktop, you get the convenience of fast user switching in addition to Single Sign-on capability through Single Sign-on.

Before you can implement Hot Desktop, however, you must:

- Create Hot Desktop-related user configurations
- Configure a Hot Desktop shared account
- Edit the scripts that define what applications run on Hot Desktop devices and their start up and shut down behavior

Hot Desktop functionality is not installed by default; you can select it during the initial installation of the plug-in software. You can also upgrade existing deployments to use Hot Desktop.

If you deploy Hot Desktop in an environment where users log on with smart cards and your selected smart card key source is DPAPI with Profile, do not select Prompt user to enter the previous password as the only key recovery method for those users. Users in such an environment cannot enter the correct previous password and, consequently, are irretrievably locked out of the system. To avoid this problem, select the automatic key management option or make question-based authentication available as an option.

Controlling Applications with Hot Desktop

With Hot Desktop, users can authenticate quickly using their Windows account credentials or smart card strong authenticator. As the administrator, you can configure Hot Desktop to launch applications in the Hot Desktop environment so your users do not have to search for and wait for their applications to launch.

You can also configure Hot Desktop to help ensure that all applications terminate properly, leaving behind a clean environment for the next user session.

The Hot Desktop User Experience

When the shared account logs on, it places the device into “fast user switch” mode, which causes a standard Windows authentication prompt to appear on the screen. The shared account remains logged on regardless of Hot Desktop user activity.

When users authenticate, they do not log on to Hot Desktop in the traditional sense. Instead, Hot Desktop uses their Windows credentials to start a Hot Desktop session. Because users are not truly logging on but rather authenticating, time-consuming events normally associated with logging on, such as applying group policy, initializing printers, and so on, do not occur. This creates the “fast-switch” users experience when running Hot Desktop. A user can start a session, perform any job-related tasks, and end the session so the next user can enter the system and do the same. The switch from user to user occurs quickly and efficiently.

The Single Sign-on Plug-in software launches when the Hot Desktop session starts. After the session is established, Hot Desktop accesses the user’s Windows account credentials to launch applications using the standard shell interface. Typically, these lightweight client applications prompt users for their credentials, which can be supplied by the plug-in software using settings associated with their Windows account.

Planning for Optional Single Sign-on Service Features

Feb 06, 2011

The Single Sign-on Service is a Web service that uses Secure Sockets Layer (SSL) to encrypt the data shared by the Single Sign-on Service, the console, and the plug-in software. It uses a dedicated Web server to host the optional features included in Single Sign-on.

Install the Single Sign-on Service if you plan to implement one or more of the following modules:

- Key Management
- Data Integrity
- Provisioning
- Self-Service
- Credential Synchronization

Important: The server that hosts the Single Sign-on Service contains highly sensitive user-related information. Citrix recommends that you use a dedicated server and that you place the server in a physically secure location.

Key Management

Key management allows users to log on to the network and have immediate access to applications managed by Single Sign-on without needing to verify their identities through question-based authentication (also known as automatic key management). To reduce security threats, automatic key management uses key splitting (the process of dividing a private key into two parts).

However, automatic key management does not protect against access by an unauthorized user or administrator impersonating a user because there is no “user secret” to protect the user’s network password. To help prevent this potential problem, implement automatic key management in combination with the Account Self-Service Module and question-based authentication.

Important: Depending on the security policy your organization implements, system administrators might be able to access passwords for applications managed by Single Sign-on. Check your organization’s security policy before allowing Single Sign-on to handle passwords that users want to keep completely private. Clearing automatic key management features in the Data Protection Methods setting in the user configuration can also help prevent this unauthorized access.

Data Integrity

The Data Integrity Module contains the public and private key files used for signing the data. It utilizes RSA public key cryptography to ensure that the plug-in software obtains configuration data provided by an authorized source only. The Data Integrity Module never distributes its private key.

After the console signs the data, the console sends both the data and the signature to the central store. The plug-in software receives the data and signature from the central store during synchronization. The plug-in software then contacts the Single Sign-on Service to obtain a copy of the public key it needs to verify the signature it received from the central store.

Install the Data Integrity Module if you want to ensure that data transmitted among the Single Sign-on components is provided by a trusted and authorized source. This module is optional and is designed for users who have non-trusted networks.

If the plug-in software is configured to use the Data Integrity Module, it never accepts configuration data that failed the data integrity check. If a check fails, the plug-in software logs the event and displays an error message telling users to contact their administrator directly. The plug-in software then defaults to previous configurations or returns to an offline state.

If you already implement a security framework that protects data in transit, such as IPsec (Internet Protocol Security) or SMB (Server Message Block) signing, you do not need to install the Data Integrity Module.

Provisioning

Provisioning (also known as credential provisioning) allows you to automate certain credential management processes. You can:

- Add, modify, and delete credentials in the central store
- Reset user credential information
- Remove users and their application credentials from Single Sign-on

Credential provisioning is achieved by using information about your environment to create a template that you can use to add, remove, or change credential information in your central store.

Self-Service

You can configure the self-service features of Single Sign-on to allow your users to reset their primary password or unlock their Windows domain accounts without intervention by administrative or help desk staff. Depending on your needs, you can implement one or both of the self-service password reset and account unlock features securely in your Single Sign-on environment.
Note: You can use the Account Self-Service feature only in an Active Directory environment to allow your users to reset their primary password or unlock their Windows domain accounts.
These account features are protected by Question-Based Authentication to help ensure that your users are authorized to reset their passwords or unlock their accounts. With Account Self-Service enabled, users must enroll, a process that requires them to answer the security questions you create and select. These security questions are then presented to users when they need to reset their password or unlock their account. When the questions are answered correctly, users are allowed to reset their password or unlock their account.

Credential Synchronization

Credential synchronization (also known as account association) allows a user to log on to any application from one or more Windows accounts. Because Single Sign-on typically binds user credentials to a single account, the credential information is not automatically synchronized among multiple accounts that a user owns. However, administrators can configure Account Association to synchronize user credentials. Users with Account Association configured have access to all applications from any of their accounts in their Single Sign-on environment. When user credentials are changed, added, or removed from one account, the credentials are automatically synchronized with each of the user's associated accounts.

Single Sign-on Plug-in Software Deployment Scenarios

Apr 13, 2011

You can use Single Sign-on in environments that include XenApp hosted applications, locally installed applications, or both.

In a XenApp deployment, you install the Single Sign-on Plug-in software on each server in the XenApp farm that hosts applications requiring credential authentication. Users access these applications through Citrix connections. The plug-in software on the server determines the application type (Windows, Web, or terminal emulator) and retrieves the appropriate credentials from the local credential store in the user's profile.

You can also install a Single Sign-on Plug-in on each user device. For a XenApp deployment, see the considerations described below. If users run applications that are installed locally on their devices, the Single Sign-on plug-in must be installed on the user device to provide credentials and access to the local applications.

Regardless of whether the Single Sign-on Plug-in is installed on the user device, users can reregister answers to their security questions without being prompted, or synchronize credentials by using Account Association only by using published applications you can give them access to after installing the plug-in on a XenApp server.

Single Sign-on can be used with:

- Access Gateway Advanced Edition (applications are available from XenApp through a web browser)
- Citrix XenApp features:
  - Citrix Receiver for Windows
  - Citrix Offline Plug-in
  - Web Interface

Deploying the Single Sign-on Plug-in on User Devices in a XenApp Environment

In a XenApp environment, deciding whether or not to install or publish the Single Sign-on Plug-in on the user device depends on what you want users to be able to do. In all cases, credentials are submitted to published applications.

- If you do not install the Single Sign-on Plug-in on the user device, users can:
  - Register answers for security questions when prompted
  - Store credentials automatically when prompted by Single Sign-on
  - Change the password for a program or Web site when prompted by Single Sign-on
- If you publish the Manage Passwords application (LogonManager.exe, installed when the Single Sign-on Plug-in is installed), users can:
  - Register answers for security questions when prompted
  - Store credentials automatically when prompted by Single Sign-on
  - Change the password for a program or Web site when prompted by Single Sign-on
  - Edit, delete, or reveal passwords stored in Single Sign-on
- If you install the Single Sign-on Plug-in on the user device, users can perform all available Single Sign-on tasks:
  - Register answers for security questions when prompted
  - Store credentials automatically when prompted by Single Sign-on
  - Change the password for a program or Web site when prompted by Single Sign-on
  - Edit, delete, or reveal passwords stored in Single Sign-on
  - Submit credentials manually, when not prompted by Single Sign-on
  - Add additional passwords for programs and Web sites already in Single Sign-on

- Pause Single Sign-on, resume Single Sign-on, or determine whether Single Sign-on is paused
- Use Account Self-Service

Planning for Multiple Primary Authentication and User Credential Protection

Feb 06, 2011

When you create or edit a user configuration, you can select user credential protection methods depending on the authentication schemes you use in your enterprise.

The following user configuration property pages enable you to tune the Single Sign-on Plug-in software behavior and credential data protection method used when users implement one or more primary authentication methods.

Data Protection Methods Page

The user configuration Data Protection Methods properties page enables you to select single or multiple primary authentication data protection methods. Additionally, you can also regulate administrator access to user credential data to help prevent administrators from impersonating a user and gaining unauthorized access to user information.

Secondary Data Protection Page

For added security when users change their primary authentication (for example, a domain password is changed or smart card is replaced), the user configuration Secondary Data Protection properties page enables you to require users to reauthenticate and verify their identities before unlocking their application credentials.

Security Versus Usability

Two key questions to ask when deciding which options to choose on these two user configuration property pages is:

- Which authentication types are used in my environment for the users I am administering in this user configuration?
- How can I balance security requirements for the enterprise and usability for all users?

Consider also that the following choices are not mutually exclusive and that you can use a mix of them in your enterprise (that is, multiple primary authentication). Your decision is ultimately based on your need for security versus ease-of-use for your enterprise users.

User Impersonation

If you want to disallow administrator access to user credentials, select Yes for the Do you need to regulate account administrator access to user data? option. Credentials are protected against administrators seeking to impersonate a user and to gain access to user information.

Yes is the default setting for the Data Protection Methods page. With this configuration, the account or other administrator does not have access to user passwords or user data. This setting helps prevent an administrator from impersonating a user. The administrator cannot log on as the user with this default setting and possibly access data located in the user local credential store.

The Yes setting disables the use of the Microsoft Data Protection API option on this page and the Do not prompt users; restore primary data protection automatically option on the following Secondary Data Protection page. Smart cards and roaming profiles are not allowed in this case, and credentials are not restored automatically upon a password change without authentication or verification.

Select No if you want to allow use of all the multiple authentication features available from this page and the Secondary

Data Protection page (including the ability to restore credentials automatically without reauthentication or identity verification).

User Name and Password

The simplest implementation is the default setting for the Data Protection Methods page: a password-only environment. The default setting lets your users employ their user name and password while protecting their credentials against unauthorized access by administrators.

Important: The security of this setting choice depends on the relative strength of your domain password policy. The stronger (or more complex) the password requirement, the more secure this choice is.

Option	**Description**
Do you need to regulate account administrator access to user data?	See *— User Impersonation* .
Users authentication data	Selected. A user secret is used to access and help protect user data. In this case, the user secret is a password. Password security can be derived from the user's typed domain password or a one-time password from token, proximity, or biometric devices.

Smart Cards with Certificates and User Authentication Data

Use Smart Cards Certificate and Users' authentication data if you combine smart cards with embedded certificates or digital signatures and user authentication data in your enterprise. Combining smart cards with a user name and password for authentication is the most secure choice for protecting user authentication data.

Select the Smart Card Certificate option if you use smart cards with Hot Desktop.

To use smart cards in a Windows Server 2008 or Windows Vista environment, your central store must be created with or updated by a Single Sign-on 4.5 (formerly Password Manager) or later console and Microsoft Data Protection API (requires roaming profiles) must be selected in your user configurations.

Option	**Description**
Do you need to regulate account administrator access to user data?	See *— User Impersonation* .
Users' authentication data	Selected. A user secret is used to access and help protect user data. In this case, the user secret is a password. Password security can be derived from the user's typed domain password or a one-time password from token, proximity, or biometric devices.
Smart Card Certificate	Selected.

Option	Description
	In this case, the user secret is protected by the encryption and decryption provided by the card's security certificate.

Smart Cards with PINs

If you use smart cards that do not support security certificates as the primary authenticator in a Windows domain or you do not use roaming profiles, use the Allow Smart Card PINs option. When you select this option, the encryption keys used to protect secondary credentials are derived from the smart card PIN.

Consider enforcing the use of a strong PIN. In some enterprises, smart card PINs are four-digit numbers that do not provide as strong a level of protection as, for example, an eight-character password and might be more vulnerable to attack. Use the PIN as password option only if your organization enforces a smart card PIN policy that requires a mixture of letters and numbers, and requires a minimum length of eight characters.

Option	Description
Do you need to regulate account administrator access to user data?	See *— User Impersonation* .
Users authentication data	Selected. A user secret is used to access and help protect user data. In this case, the user secret is a personal identification number (PIN).
Allow Smart Card PINs	Selected. Allow the Smart Card PIN to be used as the user secret for protection. Use this only if your enterprise or environment has a "strong PIN" policy

This option is supported by Version 4.1 of the Single Sign-on (formerly Password Manager) plug-in if you select Use data protection as in Single Sign-on 4.1 and previous versions and PIN as password, if you plan to use legacy plug-ins.

Roaming Profiles (Microsoft DPAPI)

Select No in response to Do you need to regulate account administrator access to user data? to enable the use of the roaming profiles and Microsoft Data Protection API in your environment. This option is the next-most secure option after smart cards with certificates and user authentication data.

Select this option if you are using roaming profiles implementing a Kerberos network authentication protocol for users. This option works only if roaming profiles are available. If you are storing roaming profiles on workstations, you must select this option.

Single Sign-on derives the encryption keys that protect secondary credentials from the user's primary password. However, if a user uses a smart card for primary authentication, a primary password does not exist and cannot be used. In this case, the best plug-in option is Microsoft Data Protection API. This option uses the Microsoft DPAPI to derive encryption keys and protect secondary credentials. This encryption mechanism uses the user's Windows or domain credentials to derive the encryption keys.

If users employ passwords to access their computers and a Kerberos network authentication protocol to access XenApp servers, select:

- No in response to Do you need to regulate account administrator access to user data?
- Users authentication data
- Microsoft Data Protection API

This method also allows the use of user credentials and smart cards to log on.

To use smart cards in a Windows Server 2008 or Windows Vista environment, your central store must be created with or updated by a Single Sign-on 4.5 (formerly Password Manager) or later console and Microsoft Data Protection API (requires roaming profiles) must be selected in your user configurations.

This method is supported by Version 4.1 of the Single Sign-on Plug-in and is supported on Windows XP and Windows 2003 Server platforms. Select Use data protection as in Single Sign-on 4.1 and previous versions and DPAPI with Profile if you plan to use legacy plug-ins.

Blank Passwords

Allowing the use of a blank password should be considered a special case and should only be used in low security environments that require extreme ease of use. One scenario is when a common workstation is placed on a factory floor and is accessed by many users. You can still use Single Sign-on to control access to applications but the user credentials to access the workstation include a blank password.

Important: If you do not select this option and a blank password is allowed in your environment, the plug-in software does not derive a user secret or otherwise perform any data protection with the blank password.

Option	**Description**
Do you need to regulate account administrator access to user data?	See *— User Impersonation* .
Users authentication data	Selected. A user secret is used to access and help protect user data. In this case, the user secret is a password.
Allow protection using blank passwords	Selected. When you select this option and the plug-in software detects that the user has a blank password, a user secret for data protection is derived from the user ID.

Install and Upgrade

May 09, 2015

The suggested installation order of Single Sign-on is:

1. Create the central store.
2. Install the Citrix AppCenter, which includes the Single Sign-on console component.
3. Install the Single Sign-on Service if you want to use one or more of the following modules:
   - Key management
   - Self-service
   - Provisioning
   - Credential synchronization
   - Data integrity

     If you decide to install the Data Integrity Module later or after installing the Citrix AppCenter and the Single Sign-on Plug-in, you must digitally sign your existing central store data with the data signing tool CtxSignData.exe. (This tool is available after you install the Data Integrity Module.) Conversely, if you uninstall the Data Integrity Module, you must unsign your central store data.
4. Install the Application Definition Tool on one or more computers in your environment when you need to create application definitions only. (When you install the XenApp server role with its default components, the Application Definition Tool is included.)
5. Install the Single Sign-on Plug-in on each user computer and on the XenApp server.

Important: The server(s) that hosts the Single Sign-on Service and NTFS central store contains highly sensitive user-related information. Use a dedicated server in a physically secure location.
The following installations are not recommended and are not supported:

- Do not install the service and the plug-in on the same computer.
- Do not install the service and the XenApp server role on the same server.
- Do not install Single Sign-on on a domain controller. Installation of the plug-in or service, console, or creation of an NTFS network share central store on a domain controller is not supported.

Upgrading to Single Sign-on 5.0

You can upgrade your entire environment to Single Sign-on version 5.0, or use a phased approach.

To upgrade your entire environment

1. Although it is not required, Citrix recommends that you upgrade to the latest version of Licensing Server and add the required licenses before you upgrade Single Sign-on.
2. If you are using any of the following modules, upgrade the Single Sign-on Service. You can also install additional modules at this time.
   - Key management
   - Self-service
   - Provisioning
   - Credential synchronization
   - Data integrity

   Note: If you decide to install the Data Integrity Module at a later date or after installing the Single Sign-on console component of the Citrix AppCenter and the Single Sign-on Plug-in, you must digitally sign your existing central store data by using the data signing tool CtxSignData.exe. (This tool is available after you install the Data Integrity Module.) Conversely, if you uninstall the Data Integrity Module, you must unsign your central store data.

3. Upgrade the Single Sign-on console component of the Citrix AppCenter (formerly known as Delivery Services Console) on one or more computers in your environment.
   Note:
   - Citrix recommends using the Single Sign-on Service and console component at the same version level.
   - Upgrading the console component to version 5.0 also performs an upgrade at the Single Sign-on central store. After you upgrade one Single Sign-on 4.8 console to version 5.0, other version 4.8 consoles cannot make changes to the central store.
4. If you need to create application definitions only, upgrade or install the Application Definition Tool on one or more computers in your environment. (When you install the XenApp server role with its default components, the Application Definition Tool is included.)
5. Upgrade the Single Sign-on central store.
   - For NTFS network share-based central stores:
     - Back up the network share folder before upgrading the Single Sign-on central store.
     - Select the Single Sign-on node and run the Configure and run discovery wizard from the Citrix AppCenter to automatically upgrade the Single Sign-on central store.
     - In the wizard, specify the UNC path of your existing NTFS network share, typically \\servername\CITRIXSYNC$, where servername is the name of the server computer where you created your central store.
   - For Active Directory-based central stores, select the Single Sign-on node and run the Configure and run discovery wizard from the Citrix AppCenter to automatically upgrade the Single Sign-on central store.
   - If you are upgrading from a version of Citrix Password Manager that supported Novell shared folder (for example, version 4.6), you may need to back up the share and export and import administrative data to continue using settings configured in that central store type. Refer to the Password Manager 4.6 administration and installation documentation for information about moving central store data. Documentation is available at the Citrix Knowledge Center.
6. After configuring Single Sign-on features in the Citrix AppCenter, upgrade or install the Single Sign-on Plug-in on each user device in your environment.

To upgrade in phases

1. Start by adding user devices running the Single Sign-on 5.0 Plug-in into your existing (Single Sign-on 4.8) environment.
2. When you're ready, upgrade both the Single Sign-on Service and the console to version 5.0.
3. Roll out deployment of the Single Sign-on 5.0 Plug-in to the remainder of your user devices.

Setting Security and Accounts Before Installing Single Sign-on

Mar 24, 2011

Before you install the Single Sign-on Service, ensure that the appropriate accounts and components are available to support the service. Also, because the service uses secure HTTP (HTTPS), the service requires a server authentication certificate for Secure Sockets Layer (SSL) communication with the console and plug-in.

Obtaining and Installing a Server Authentication Certificate

Obtain a server authentication certificate for SSL communication from a certificate authority (CA) or, if you have an existing public key infrastructure (PKI), download your own certificate to the server running the service.

An SSL certificate is necessary to ensure secure communication from the service to the console and plug-in, and to guarantee that the plug-in and console are communicating with the correct service server.

- Because this certificate is used for SSL communication, the certificate common name must match the service server's fully qualified domain name (FQDN). Specify a minimum key size of 1024.
- Install the certificate in your local computer certificate store and establish the appropriate trust relationships for the Single Sign-on component of the Citrix AppCenter and the plug-in.
- Install this certificate on the computers running the Single Sign-on component of the Citrix AppCenter, the Single Sign-on Service, and the plug-in.
- In a load balancing or clustered service environment, you can use one certificate for multiple service servers if the common name of the SSL certificate uses a wildcard (typically an asterisk character) in it. For example, you can use an SSL certificate with a common name of server*.mycompanysname.com for an environment with servers named server1.mycompanysname.com, server2.mycompanysname.com, and server3.mycompanysname.com. You could also use an SSL certificate with a common name of *.mycompanysname.com in this case, where the common name does not match the server FQDN.

Important: If you obtain your certificate from an authority that is not trusted by default (such as a certificate authority installed in your company), install the root authority certificate to your local computer's trusted root certificate store to establish the trust relationship.
If users are experiencing SSL failures, it is most likely because the server certificate is not trusted. Refer to the Microsoft Web site for instructions about extracting and deploying CA root certificates.

The signing and validation certificates created during Single Sign-on installation are not related to the SSL certificate.

Accounts Required for Service Modules

The Single Sign-on Service can require up to three system account types to read and write data as it operates in your environment. The number and type of accounts required depend on the service modules you use. The table shows the accounts required by each module of the service. In cases where different modules require the same type of account, you can use the same account for multiple modules or you can specify different customized accounts for each module.

Module	**Accounts Required**		
	Service	**Data Proxy**	**Self-Service**
Data Integrity	Yes	No	No

Module	Accounts Required		
	Service	Data Proxy	Self-Service
Key Management	Yes	Yes	No
Provisioning	Yes	Yes	No
Self-Service	Yes	Yes	Yes
Credential Synchronization	Yes	No	No

Service Account Requirements

On the server running the Single Sign-on Service, use the existing Network Service or Local Service accounts

You cannot specify a local user account as the service account in this version of Single Sign-on. You can specify the built-in Local Service account.

If you choose to create a domain account as the service account, you must register a service principal name for this domain account and the service computer in Active Directory by using the setspn.exe utility. if using a domain user account, the account should be assigned "Logon as services" rights. The computer running the service needs to be trusted for delegation.

See the Microsoft Web site for more information about service principal names.

Data Proxy Account Requirements

On the server running the Single Sign-on Service, create a domain administrator account with the following settings, to be used for data proxy communication with the service.

The account requires read and write access to the central store. The account requirements depend on the central store type you are implementing.

Central Store Type	Account Description
NTFS Network Share	The account: ● Requires read and write access to the central store. ● Is a member of the domain. After you create the central store: ● Grant the account Full Control sharing permissions to the CITRIXSYNC$ share. ● Grant the account Full Control permissions to the CITRIXSYNC folder and its subfolders: CentralStoreRoot folder and People folder ● Grant the account Full Control permissions to all file objects within the CITRIXSYNC folder and its subfolders ● Ensure that the Authenticated Users group has the right to create folders inside the People folder.
Active Directory	The account: ● Requires read and write access to the central store.

Central Store Type	Account Description
	- Is a member of the domain administrator group.

Self-Service Requirements

If you are using the Self-Service Password Reset or Self-Service Account Unlock features of the Account Self-Service Module, use an account that is a member of the domain administrators group.

Accounts Required to Install and Use Single Sign-on

The user installing the Single Sign-on Service and running the Service Configuration wizard must be a member of the domain (a Domain User) and a member of the local Administrators group on the service computer (add a domain user account to the local Administrators group).

The user installing the Single Sign-on console component, performing a discovery and configuration operation, and using the console component must be a domain administrator and a member of the local Administrators group on the console computer. This user account must have read and write access to the central store. A non-administrator user account can be assigned the right to manage the console component and its related functions through Active Directory delegation or constrained delegation.

The user installing the Single Sign-on Plug-in must be a member of the domain (a domain user) and a member of the local Administrators group on the user device. The user installing the plug-in must be a member of the domain (a domain user) and a member of the local Administrators group on the user device. The user running the plug-in must be a member of the domain (a domain user).

Installing the Java Runtime Environment

Aug 20, 2013

Single sign-on supports the Java Runtime Environment (JRE), Versions 1.4.x, 5 (1.5.x), and 6 (1.6.x). Download the current supported version from the Sun Microsystems Web site (http://java.sun.com).

If You Install or Upgrade the JRE after Installing the Single Sign-on Components

If you install or upgrade the JRE after installing the Single sign-on component of the Delivery Services Console, Application Definition Tool, or plug-in, associate the current JRE with the Single sign-on component.

1. In the Control Panel, go to the Programs area and select the Single sign-on component.
2. Click Change.
3. In the setup dialog, select Repair.

Troubleshooting a Java-Related Error Message When Installing or Uninstalling the Plug-in

You might see the following error message when you attempt to install or uninstall the plug-in:

"Citrix Single sign-on has detected that one or more Java software programs or files are currently in use. Please close all programs and stop all Java-related services before continuing. "

Typically, this error occurs if you are installing the plug-in on a computer also running a Web server service such as Apache Tomcat or Apache HTTP server. Also, this error might be seen if you are installing the plug-in on a XenApp server with a License Management Console installed.

In this case, perform the following steps:

1. Stop the service.
2. Install or uninstall the plug-in.
3. Restart the service.

Creating a central store

Aug 16, 2013

1. Load the XenApp media.
2. From the Autorun menu, select Manually install components > Server Components > Additional Features > Single Sign-on.
3. Select Central Store.
4. Select a central store type: NTFS network share or Active Directory.
   - If you select NTFS network share, the central store is created as %SystemDrive%\CITRIXSYNC$.
   - If you select Active Directory:
     1. Select Step 1: Extend Active Directory. The Active Directory schema is extended.
     2. Select Step 2: Create Central Store.
     3. After the Central Store is created, restart the server where the Single Sign-on console is installed. You need to do this so the Central Store is discovered.

     Important: Ensure the current server is part of the Active Directory domain and that the current user is a member of the Schema Administrators group and Domain Administrators group. Ensure that the Active Directory Schema Master is configured to allow updates. Also, if the server you are extending the Active Directory schema from is not the domain controller, ensure the Microsoft Windows utility Ldifde.exe is installed on it before beginning this step. The utility can be found on the Windows installation media or at the Microsoft Web site. You will not be able to complete this process if Ldifde.exe is not installed.

Installing the Console Component

Dec 16, 2011

The Single Sign-on console component is included when you install the Citrix AppCenter.

Important: You must create the Single Sign-on central store before you can successfully complete the Configure and run discovery wizard and use Single Sign-on.

To install the AppCenter (and Single Sign-on console component) when installing XenApp

1. Follow the procedure for installing the XenApp server role. By default, the AppCenter is included in the installation.
2. Select Configure and run discovery and follow the instructions.

To install the AppCenter (and Single Sign-on console component) manually

Ensure that the required Microsoft Visual C++ Redistributable Packages and Microsoft Primary Interoperability Assembles are installed, as described in System Requirements.

1. Load the XenApp media on the computer.
2. From the Autorun menu, select Manually install components > Common Components > Management Console. Follow the instructions.
3. Select Configure and run discovery and follow the instructions.

Installing and Configuring the Service Modules

Jul 29, 2016

The installation and configuration workflow is:

1. Acquire and install the SSL certificate on the computers running the Single Sign-on service, console, and plug-in.
2. Create the account type required by the service modules you will install
3. Install the service modules.
4. Configure the service modules.

The following procedures assume that the installation media is loaded on the computer that you chose to host the Single Sign-on Service modules and that the XenApp autorun screen appears.

To install the service modules

1. Load the XenApp media.
2. From the Autorun menu, select Manually install components > Server Components > Additional Features > Single Sign-on > Single Sign-on Service.
3. Follow the instructions.

To configure the service modules

The Service Configuration wizard starts when the service module installation completes. You can start the wizard later by selecting Start > All Programs > Citrix > Password Manager > Service Configuration.

Follow the directions.

- On the Configuration service page:

Connection Setting	Specify the port number for the service connection; the default is 443. You can use any other available port on the server running the service. If you install one or more service modules later, use the port number that you specified when you first installed the service. The service cannot run on multiple ports; if you specify the wrong port, Single Sign-on might later display “cannot communicate or connect with the Single Sign-on service” type error messages. Specify the correct service port number when using the Data Integrity Signing Tool at the command prompt.
SSL Certificate	Select the SSL certificate installed on the service computer to use for communication with client devices. Select the Display Long Name check box to show the LDAP informationConnection Setting contained in the certificate.
Virtual host name	Use default value is selected by default if the SSL certificate name and virtual host name match. The virtual host name must match the SSL certificate name.

	The virtual host is the machine name visible to users when the certificate was created and might not be the actual machine name. For example, the certificate name might include a wildcard (asterisk), or uppercase or lowercase domain name that does not match the certificate domain name case. This setting is useful in a load-balanced or clustered service environment.
Account Credentials	Select the local computer account to use for the service. Typically, you can select the Network Service account.

- On the Configure domains page:
  1. Select the check box next to each domain to which you want to enable service support.
  2. Select one or more domains and click Properties to open the Edit Configuration dialog box.
  3. If you created an Active Directory central store, select Domain Controller and select the correct domain controller from the list.
  4. Select Data Proxy Account and type the user name, password, and domain of the data proxy account used to communicate with the central store.
  5. If you installed the Self Service module, select Self-Service Features Account and type the credentials for this feature. Select OK to close the Edit Configuration dialog box.
     Important: If the service is running in a Windows Server 2008 or Windows Server 2008 R2 environment with an NTFS central store, you must use CtxFileSyncPrep.exe to add the data proxy account as an administrator to the central store. Type:
     CtxFileSyncPrep [/Admin:accountname]

     If the service is running in a Windows Server 2008 or Windows Server 2008 R2 environment with an Active Directory central store, you also must add the data proxy account as an administrator to the central store.

Configuring the Service for Multidomain Use

Single Sign-on Service can process service requests among users in different trusted domains. An administrator can install the Citrix AppCenter with the Single Sign-on console component on computers in different domains and create one or more user configurations in each domain.

For example, with the Single Sign-on Service computer located in DomainA, users associated with a user configuration in DomainA can use the Account Self-Service features to unlock their accounts. Users associated with a user configuration in DomainB can also use this feature, as provided by the DomainA service computer. In this case, multiple user configurations exist in multiple domains and are using a single service computer for this feature.

Multi-Domain Service Feature Requirements

Before you implement the multi-domain service feature, ensure that you meet the following requirements:

Component	Requirement
Domains	Each domain sharing the service must be part of the same domain forest. The domains within the forest must have a two-way transitive trust agreement.

Component	Requirement
Central store	This feature is available for implementations using Active Directory or NTFS network share central stores. All users sharing the same service computer must be implemented using the same central store type: Active Directory or NTFS shared folder. Multiple central store types are not supported. One NTFS shared folder central store per domain is not supported in this case. However, you can use one NTFS shared folder central store per forest.
Data Integrity feature	The Data Integrity feature must be used consistently across domains. That is, it is either enabled or disabled in the service and Single Sign-on Plug-in configurations for all domains. For example, you cannot enable this feature in the service configuration and disable it when installing the plug-in.
Single Sign-on console component of the Citrix AppCenter	Each console can view one central store only, not multiple central stores. The Single Sign-on administrator should install one console in each domain and install it by using a user account with administrative rights in that domain. Alternatively, the administrator can install a console with the ability to access other domains and, as needed, switch to one of those domains by logging on with credentials for that specific domain.
Data Proxy and Self Service accounts	You can configure one data proxy and self service account that has read and write access to the central store and sufficient privileges to reset user passwords and unlock user accounts. Optionally, you can specify these accounts for each domain in the Service Configuration tool.

To configure the service for multidomain use

1. Log on as an administrator to the computer where the service is installed.
2. Start the Service Configuration tool by clicking Start > All Programs > Citrix > Password Manager > Service Configuration.
3. When the Service Configuration tool appears, click Domain Configurations in the left pane.
4. Select the check box next to each domain to enable service support on that domain.
5. Select one or more domains and click Properties to open the Edit Configuration dialog box.
6. In the Edit Configuration dialog box:
    1. If you created an Active Directory central store, click Domain Controllers and, from the list, select the domain controller you want Single Sign-on to bind to when writing to the central store or select Any writeable domain controller.
    2. Click Data Proxy Account and type the user name, password, and domain of the data proxy account used to communicate with the central store.
    3. If you installed the Self Service module, click Self-Service Features Account and type the credentials for this feature.

Installing the Single Sign-on Plug-in

Apr 15, 2011

The Single Sign-on Plug-in runs on the XenApp server and provides credentials and access to published applications. The plug-in also runs on each user device, providing credentials, access to applications running locally on the device, and the ability to control Single Sign-on operations.

Note: When you use this version of the plug-in on XenApp to publish applications enabled for Single Sign-on, user devices should also have the plug-in installed. If the user device does not have the plug-in installed, Single Sign-on will automatically submit credentials to applications published with XenApp, but the user will be unable to edit, delete, or reveal password information, pause or resume Single Sign-on, determine whether Single Sign-on is paused, or submit passwords manually.
Installation considerations:

- Installing this version of the Single Sign-on Plug-in on a user device upgrades a 4.8 version.
- After installing the plug-in on a supported operating system that uses the Microsoft Graphical Identification and Authentication (GINA) Windows component, you must restart the device. Ths includes Windows XP, Microsoft Windows XP Embedded, Microsoft Windows Fundamentals for Legacy PCs, Microsoft Windows Server 2003 R2, and Microsoft Windows Server 2003 with Service Pack 2.
  WinLogon uses the GINA controls for the dialog box that users see when they press the key combination CTRL+ALT+DEL. The dialog box collects the data needed to perform authentication. XenApp, the Single Sign-on Plug-in, and the Novell NetWare client interact with or require the replacement of the GINA dynamic link library (DLL). You might be required to install or uninstall software in a specific order to preserve proper GINA chaining. By installing the Single Sign-on Plug-in last, you ensure that the Single Sign-on GINA is called first by the Winlogon process.
- After the installation completes (and the device restarts, if needed), the Citrix Receiver icon appears in the system tray.
- After installing the plug-in, if you configure or change Citrix licensing information, restart the plug-in to apply the changes.

To install the Single Sign-on Plug-in on a server when you install XenApp (wizard-based)

1. Follow the instructions in
   — Installing XenApp Using the Wizard-Based Server Role Manager
   . From the Optional Components list, select Single Sign-on Plug-in.
2. When configuring XenApp using the wizard-based Server Configuration Tool, you are prompted to select the type of central store: Microsoft Active Directory (default) or NTFS Network Share and its path.

To install the Single Sign-on Plug-in on a server when you install XenApp (command-line)

1. Follow the instructions in
   — Installing XenApp from the Command Line
   . Include the SSONAgentFeature option (/install:XenApp,SSONAgentFeature).
2. When configuring XenApp from the command line, you can include the /SSOPluginUncPath:path option to specify the UNC path to the NTFS network share central store. If you omit this option, Active Directory is assumed.

To install the Single Sign-on Plug-in on a user device or on a server with XenApp installed

1. Load the XenApp media on the computer or server.
2. From the Autorun menu, select Manually install components > Server Components > Additional Features > Single Sign-on > Single Sign-on Plug-in.
3. Follow the instructions. You are prompted to select the type of central store and the components to install (such as

language packs, Self-Service, and data integrity).

To install the Single Sign-on Plug-in on a user device using Merchandising Server

Follow the procedures for downloading or delivering plug-ins in the Merchandising Server documentation.

Icon Consolidation in the Microsoft Windows Notification Area

When using this version of the Single Sign-on Plug-in for all XenApp sessions and on each user device, the Microsoft Windows notification area on each user device contains only one Receiver icon, with an integrated Single Sign-on menu that consolidates all sessions.

However, if either XenApp or the user device uses an earlier plug-in version, the Windows notification area can also contain Single Sign-on icons. The following table illustrates several scenarios.

User device		**XenApp server**		**Windows Notification area**	**Passwords menu available from Receiver icon?**
Citrix Receiver	**Single Sign-on Plug-in**	**Citrix Receiver**	**Single Sign-on Plug-in**		
Current *	5.0	Current	5.0	One Receiver icon	Yes
Current	-	Current	5.0	One Receiver icon	No
Current	5.0	-	4.8	One Receiver icon and one Single Sign-on icon for each connected XenApp session. **	Yes
Current	4.8	Current	5.0	One Receiver icon and one single Sign-on icon	No
Current	4.8	Current	4.8	One Receiver icon and one Single Sign-on icon, plus one Single Sign-on icon for each connected XenApp session. **	No
Earlier online plug-in	4.8	Current	5.0	One Single Sign-on icon and one online plug-in icon	No

* Current = Receiver for Windows, which contains the Online Plug-in

** If the XenApp servers are running an earlier version of the Single Sign-on plug-in, and the current Receiver is installed on the user device (regardless of whether any Single Sign-on plug-in is installed on the user device), the Windows notification area on the user device will contain a Single Sign-on icon for each of those XenApp servers (running the earlier plug-in version) to which it is connected.

Manage

Mar 29, 2011

You can use password policies to define rules that control the characteristics of users' stored passwords. These rules comprise password policies that you can apply to all users or to specific groups of applications as determined by your organization's needs.

Note: Citrix XenApp provides policy rules that allow you to configure and control which users can access Single Sign-on when they connect to servers and published applications in the server farm. Despite the similar names, these two types of policies are not related.
Single Sign-on includes two standard password policies named Default and Domain. You can use these policies as is, copied, or modified to suit your enterprise policies and regulations. You cannot delete the Default and Domain policies.

When a user adds credentials to Manage Passwords window (formerly known as Logon Manager) for an application not defined by an administrator, Single Sign-on uses the Default policy to manage that application. If you want an application group to be treated as a domain password sharing group, apply the Domain policy to that application group.

Because Single Sign-on applies the Default password policy to user-added applications, configure the Default policy to be as broad as needed to accept passwords for those applications for which you allow passwords to be stored.

You can create as many policies as you need in your enterprise. For example, you can apply one policy for your domain sharing group, and create individual policies to apply to individual groups of applications to define the requirements further. With a password policy, you can:

- Automate password changes for applications.
- Implement security schemes that include complex passwords and application-specific passwords not visible to the users.
- Define password expiration for applications, even if the application does not have a password expiration feature.
- Prevent users from reusing passwords for the same application twice in a row.

Password Sharing Groups

Users might have a single password that is used for multiple applications (in a suite of products, for example). This is known as password sharing, where the same authentication authority is used for the applications.

While other credentials for those applications (such as user name and custom fields) might be different, the user's password is the same. In this case, create an application group that is a password sharing group to ensure that Single Sign-on Plug-in manages the password for all applications in the group as a single entity. When the password is changed in one of the applications, Single Sign-on Plug-in ensures that the password change is reflected in the stored credentials for all applications in the group.

Domain Password Sharing Groups

Domain password sharing groups differ from other password sharing groups because the user's domain password is the master password for the application group. When the user changes the domain password, Single Sign-on Plug-in reflects the change in the credentials for all other applications in the group. Only the domain password can be changed; users cannot initiate password changes on any of the other applications in the group unless the administrator removes the application from the domain password sharing group.

Enforcing Password Policies

Single Sign-on enforces password policies, regardless of whether the password is user-defined or automatically generated by Single Sign-on.

A password policy is not enforced when:

- A user registers with Single Sign-on (during first-time use).
- A user edits a password from the Manage Passwords window (formerly known as Logon Manager).
- An administrator creates an application definition.

Single Sign-on also does not enforce a password policy on existing passwords (those created before Single Sign-on is implemented in the enterprise) because users might be denied access to applications or resources currently in use.

Configuring Single Sign-on to Recognize Applications

May 09, 2015

Single Sign-on recognizes and responds to applications based on the settings identified in application definitions.

Application definitions contain forms that allow the Single Sign-on Plug-in to analyze each application as it is started, recognize certain identifying features, and determine if the starting application requires the plug-in to perform some specific action, such as:

- Submit user credentials at a logon prompt.
- Negotiate a credential changing interface.
- Process a credential confirmation interface.

Application definitions consist of sets of specific user credential form recognition and action characteristics referred to as form definitions, and the set of configuration options that apply to all the forms in the configuration.

The form definition settings define the actions that Single Sign-on performs when an application requests a specific user credential action.

An application definition contains all the user credential management forms associated with a single application.

Although most applications and their corresponding application definitions use only two forms for managing user credentials, you can define as many forms as necessary in a single application definition.

Single Sign-on provides support for a variety of applications including Windows, Web, and terminal emulator-based applications. It works with Java applications; SAP solutions; and applications hosted on a mainframe, AS/400 system, or UNIX server.

Use the provided wizards to create application definitions for applications that do not have predefined application templates. The Application Definition Wizard configures the characteristics associated with all the forms included in the definition. The Form Definition Wizard leads you through a step-by-step procedure to define support for Windows, Web, and terminal emulator-based applications.

Single Sign-on also provides the ability to perform external application discovery and action processing support. This allows third-party implementers to extend the application detection and credential submission tasks associated with a form by providing access to external processes during the application detection and action submission processing phases in the Single Sign-on Plug-in.

These features combine to provide you with a flexible and adaptable application definition development environment to support your user community with secure and flexible Single Sign-on access to critical applications.

Caution: Single Sign-on is dependent on the secure operation of the computers hosting the product's components. If the user device becomes infected with any malicious code, there is a risk that this code could undermine the security provided by Single Sign-on. To reduce this risk, follow standard security best practices to maintain the security of your organization's infrastructure.

Application Templates

Application templates are XML files used to share application definitions between different Single Sign-on environments. Application templates save time and effort because you can convert them to application definitions with minimal intervention or configuration. Templates require you to provide information to complete the application definition, such as a URL or executable file name, password expiration, and any advanced detection settings.

Install application templates using the Single Sign-on node of the Citrix AppCenter or the Application Definition Tool. Both of these include application templates for commonly-used Windows and Web applications.

Important: To write to an Active Directory central store while running in Windows Server 2008, Windows Server 2008 R2, Windows Vista, or Windows 7, grant the Application Definition Tool an integrity level of High. Log onto an account that is a member of the local administrators group to start the tool on the system computer as well as be a member of the domain administrators group or have write privileges to the Active Directory objects in the central store. Provide these credentials when running the tool, either at the User Account Control prompt or when logging on to the system. The tool is assigned a High integrity level and can write to the Active Directory.
When an application template cannot be found for an application, create an application definition using the Single Sign-on node of the Citrix AppCenter or the Application Definition Tool.

How Single Sign-on Plug-in Identifies Applications and User Credential Management Events

Mar 24, 2011

The user interface to an application includes different forms that are used to manage user credential events associated with the application.

For example, one form enters the logon credentials, a second form changes an application password, and a third form confirms a successful change to user credentials.

Depending on the type of application being defined (Windows, Web, or terminal emulator), Single Sign-on uses a variety of identifiers collected in application definitions, to uniquely respond to and identify the forms. These include but are not limited to the application type, window title, and the executable file name.

When Single Sign-on Plug-in identifies the application and form, it prompts users to provide or store their credentials, submits stored credentials, or prompts users to update their credential information, depending on the defined settings.

Create application definitions using the AppCenter or the Application Definition Tool.

A single application definition supports all user credential management events associated with a single application including:

- Authenticating the user.
- Changing user credentials.
- Confirming credential changes.

Application definitions are categorized into three main types which determine the information collected:

- Windows applications (including Java applications and the SAP LogonPad)
- Web applications (including Java applets)
- HLLAPI-compliant terminal emulator-based applications

An application definition consists of:

- Application characteristics that apply to all forms included in the definition. These are defined using the Application Definition Wizard.
- Form-specific data used to recognize each different credential management event associated with the application. Define these forms and events using the Form Definition Wizard. This wizard runs during the Application Definition Wizard operation.

The application characteristics for all types of applications contain similar configuration information. However, form-specific data contained in the application definition varies greatly based on the type of application being defined.

To create an application definition, you must be able to access the application from the computer where the application definition is created. Because application signatures can vary depending on the underlying operating system, test application definitions on all operating system software in your organization.

Test any changes or upgrades to an application after an application definition is deployed to ensure that there are no changes to the application signatures requiring a change to the application definition.

Important: As a security measure, in its default state, Windows Server 2008, Windows Server 2008 R2, Windows Vista, and Windows 7 run with User Interface Privilege Isolation (UIPI) enabled. UIPI prevents applications from sending messages to other applications with a higher integrity level. As a result, the Single Sign-on Plug-in, which operates by default at a medium integrity level, does not detect or submit credentials to applications running at a high integrity level. To maintain the intended security level of these operating systems and Single Sign-on, continue to use these default settings.

Application and Form Definition Wizard Overview

May 09, 2015

All application definitions are initially created using the Application Definition Wizard and the integrated Form Definition Wizard.

The Form Definition Wizard defines the characteristics associated with each user credential management form included in an application definition.

Application Definition Wizard Overview

To start the Application Definition Wizard, select the Application Definitions node in AppCenter and then, from the Action menu, select Create application definition.

The Application Definition Wizard collects information for each type of application (Windows, Web, and terminal emulator-based).

Data Collected	Windows	Web	Terminal Emulator
Identify application	X	X	X
Manage forms	X	X	X
Name custom fields	X	X	X
Specify icon	X		
Configure advanced detection	X	X	X
Configure password expiration	X	X	X
Confirm settings	X	X	X

Managing Forms with the Application Definition Wizard

Most applications have separate forms for logon and password changes. Some applications also have separate forms that notify users if they successfully changed their password.

Use the Manage forms page to add forms to the application definition. You can also edit and delete forms on this page.

Selecting Add Form starts the Form Definition Wizard that is used to collect the form data for a single form. Use the Form Definition Wizard for each form in the application definition.

Naming Custom Fields

Single Sign-on includes the user name and password fields as required for any logon form. Some applications require additional information such as a database name, domain name, or system name to authenticate the user.

You can add up to two custom fields with the Form Definition Wizard. If you do so, when you return to the Application Definition Wizard, use the Name custom fields page to name these fields.

To create a hot key for the custom field name, place an ampersand (&) in the field name immediately before the letter you want to specify as the hot key. If no hot key is identified, Single Sign-on Plug-in dynamically appends a numeric value as the hot key for the control. This appears on the button as (1) or (2) depending on the number of custom fields defined.

Specifying an Icon for Windows Applications

By default, Single Sign-on uses a different icon to denote between Windows, Web, and terminal emulator-based applications in the Manage Passwords window (formerly known as Logon Manager). However, you can specify a custom icon for Windows applications on the Specify icon page to help users identify specific applications. If you chose the custom icon option, store the icon file in the same location as the application.

Preventing Credential Loops

Use the options on the Configure advanced detection page to mitigate credential submission loops and credential change loops.

On occasion, users may find themselves on a Web site with a credential submission loop. In these cases, users log off from an application which returns them to the logon screen. Single Sign-on Plug-in detects the logon screen and submits the users' credentials, automatically logging them back on. Enable Process only the first logon for this application to prevent the automatic submissision.

When a predefined application is launched for the first time and this option is selected, Single Sign-on Plug-in submits credentials on the initial instance of the logon form without any additional user action required. When users log off and the logon screen appears again, a window appears and stays visible for approximately 10 seconds. Users have three options:

- Close the window - no credentials are submitted
- Ignore the window - no credentials are submitted
- Click the link - credentials are submitted

Closing the application terminates the session and Single Sign-on submits the credentials the next time the application opens.

To prevent a credential change loop, enable Process only the first password change for this application. With this option selected, if users attempt to change their passwords multiple times while accessing a specified application, they are asked to verify subsequent password changes.

Configure Password Expiration

The Configure password expiration page includes options to:

- Identify a script to run when the password expires.
- Use the Single Sign-on expiration warning.

You or someone within your enterprise may develop a script to prompt users to change passwords on any or all of their applications at regular intervals, change passwords on any or all of their applications automatically, or a combination of these processes to meet your security and regulatory requirements. To run such a script when the password associated with this application definition expires (as defined in the password policy), enable the run script option and specify the absolute script path. The script path must be accessible to all users. Do not use a Universal Naming Convention (UNC) path.

Typically, the script invokes an associated application using a command prompt interface with a change password parameter.

You can also optionally enable the Use Single Sign-on expiration warning. Enabling this option causes a Single Sign-on password expiration warning to appear when the password policy associated with the application indicates that the password expired. This action displays a re-occurring message that the associated time period has expired but does not force a password change action.

Form Definition Wizard Overview

Use the Form Definition Wizard to:

- Define a form with the Application Definition Wizard.
- Edit an existing form.
- Add a form to an existing application definition.

Use the Form Definition Wizard to define several standard user credential management forms:

- Logon form
  Identifies the logon interface to an application and manages the actions required to gain access to the associated application.

- Password change form
  Identifies the password change interface to an application and manages the actions required to change the user password to the associated application.

- Successful password change form
  Identifies the password change interface to an application and manages the actions required to acknowledge the successful password change for the associated application.

- Failed password change form
  Identifies the unsuccessful password change interface to an application and defines the actions to take when a credential change operation is unsuccessful.

Password Manager Agent Versions 4.0 and 4.1 do not support successful or failed change credentials forms and do not respond to application definitions containing these forms.

The data collected for each form performs two functions:

- Uniquely identifies when an application-specific form is started.
- Performs the appropriate user credential processing actions associated with the form.

The Form Definition Wizard is started from the Manage forms page of the Application Definitions Wizard by selecting Add Form.

The following table shows the form information that is collected for each type of application (Windows, Web, and terminal emulator-based) using the Form Definition Wizard.

Data Collected	Windows	Web	Terminal Emulator

Data Collected	Windows	Web	Terminal Emulator
Name form	X	X	X
Identify form	X	X	X
Define form actions	X	X	
Set field detection rules			X
Configure other settings	X	X	X
Confirm settings	X	X	X

Windows Type Application Definitions

May 09, 2015

Use Windows type application definitions to identify Windows applications, Java applications, and applications started from an SAP Logon Pad.

For the purposes of defining an application definition, categorize any application launched by a file with an .exe extension as a Windows application.

To gather information required for Windows application definitions, launch the application and navigate to the form that requires a user credential management event (user logon, change password, successful password change, or failed password change) while running the Form Definition Wizard from the console or from the Application Definition Tool. The wizard provides instructions for locating and identifying the applicable parts of the application.

Identifying Forms

When creating application definitions for Windows type applications, use the Identify form page to provide the information required for Single Sign-On Plug-in to uniquely recognize the form being defined.

The identifying information includes the Window title and the executable file name. When Single Sign-On Plug-in detects the executable file name, it monitors the application for the defined Window titles.

When a window title is detected, Single Sign-On Plug-in performs the actions defined for the form.

To identify a form

1. If you haven't already done so, start the Windows program and navigate to the user logon, change password, successful password change, or failed password change form.
2. From the Identify Form page of the Form Definition Wizard, click Select.
3. If the desired program is not highlighted, use the Window selector to choose from the other available programs.

Identifying Dynamic Window Titles

While on the Identify form page, you can edit the titles in Window titles for this form to manage dynamic Window title data such as a date or session identifier. To do so, substitute wildcard characters for dynamic data that appears in the Window title as follows:

Wildcard	**Description**
?	Use only for a single dynamic/changing character in a Windows title.
*	Use this value to represent dynamic title data for one or more characters. This value is not recommended for empty Windows titles. Use NULL for these situations.
NULL	Use this value for empty Windows titles (the word “NULL” must be all uppercase).

Identifying Secure Paths

The Executable file names and paths area displays the name of the identified executable file and any secure path information.

Secure paths limit recognition of the application to only those program instances initiated from the paths defined here. If one or more secure paths are identified, Single Sign-on Plug-in submits credentials only when the identified program is run from the defined path and all other defined form identifiers are present.

You can define a secure path by clicking, in the Window selector, Use Full executable path.

If no path information is defined, None provided appears and Single Sign-On Plug-in provides credential information to any program that matches the other form identifiers.

Separate multiple paths with semicolons. You can use absolute paths or environment variables to identify the path.

Note: You can use application definitions that include secure path information to create an application definition template; however, the secure path is not included as part of the template.

Defining Form Actions

The Define form actions page is used to define the actions that must be performed by the Single Sign-on Plug-in to submit the credentials for the specific form being defined.

The top of the page displays the selection of user credentials associated with the specific form:

	Logon Form	**Password Change Form**	**Successful Password Change Form**	**Failed Password Change Form**
Username/ID	X	X	X	X
Password	X		X	X
Old Password		X		
New Password		X		
Confirm Password		X		
Custom Field 1	X		X	X
Custom Field 2	X		X	X
OK	X	X	X	X

The bottom of the page displays the defined action sequence.

The objective of this page is to define the actions to be taken by the Single Sign-on Plug-in to successfully submit the required user credentials to the identified form.

To define form actions

The following procedure is sufficient for most Windows applications:

1. Click the Set/Change hyperlink associated with a specific user credential. This action opens the Configure Control Text dialog box used to identify the control to receive the selected credential.
2. Select the control type candidate to receive the credential. As the different candidates are selected, the associated control type is visibly highlighted on the application to make it easier to identify the control type that is to receive the identified user credential or submit button.
3. Repeat this action for all the user credentials required by the form and for the button required to submit the form. Some forms require domains or other user-configurable credentials that must be successfully submitted to process the form. To accommodate these requirements, two custom fields are made available. Assign special-requirement credentials to these fields. The names associated with these fields are defined on the Name custom fields page of the Application Definition Wizard after the form is defined.

   Note: Not all the credentials identified in the top of the Define form actions page must be configured.

Window Identifier

The Window Identifier page is used to define a Windows control ID that uniquely identifies a form when more than one window can be identified using only the defined Windows title and the executable file name. It is useful only if the Windows control ID can be used to differentiate among the multiple forms that can be identified.

Select the Enable matching by Window Control ID check box and provide the control ID that uniquely differentiates the window for the form being defined from all the other possible forms.

Identification Extensions

Identification extensions are part of the Application Definition Extensions. These extensions provide support for using applications that are external to the plug-in software to recognize the occurrence of a user credential management event and perform the credential submission process.

Although Single Sign-on administrators can generally create application definitions using the Single Sign-on console component and the Application Definition Tool, some applications have special considerations or requirements that require an alternate means of detecting the application and submitting the user credentials or performing other similar actions.

To support these applications, Single Sign-on administrators can use the Application Definition Extensions to provide an abstraction for the application controls and the associated data input mechanisms.

Identification extensions are developed by third-party implementers and implementation is application-specific. Therefore the procedures required to configure their use are application-specific.

Generally, Single Sign-on administrators are not involved in the development of these extensions. Extensions are created by third-party implementers. Because configuration of these extensions is extension-specific, instructions for configuring the extension will most likely accompany the extension.

Defining Action Sequences for Windows Forms by Using the Action Editor

Use the Define form actions page to define actions that must be performed by the plug-in software to submit the credentials for the specific user credential management form being defined.

For many Windows applications, the basic information gathered in the Form Definition Wizard is enough to define the form.

However some forms require more information, steps, special keys, or other actions to successfully complete a user credential management task. For these forms, on the Define form actions page, click Action Editor to open the Action Editor dialog box.

The Action Editor dialog box consists of:

- Select Actions
  Displays all possible action-sequence actions:
- Configure Actions
  Used to define the action-specific options to include in the action sequence.
- Sequence Actions
  Displays the sequence of defined actions to perform to process the specific user credential management form.

At the bottom of the Action Editor dialog box is the Advanced Settings button that is used to access the Advanced Settings dialog box. The Advanced Settings dialog box has two controls:

- Control ordinal numbers
  Select this check box to use control ordinal numbers (often referred to as Z-order) instead of control ID numbers. Control ordinal numbers are independently enumerated during the definition process (and by the plug-in software) to uniquely identify the controls independently of the control ID numbers defined by the application.

  Consider selecting this feature defining .NET applications that dynamically generate control ID numbers or for applications that have duplicate control ID numbers.
- Initial delay
  Select this option and define the amount of time that the plug-in software is to delay processing before beginning the action sequence. A delay can also be configured by starting the action sequence with a delay using the Insert delay action for additional information.

  Unlike using the Insert delay option that is accessed from the Select Actions area on the Action Editor dialog box (defined as a send key operation), any initial delay defined here can be used to avoid creating an application definition that is supported only on Versions 4.5, 4.6, 4.6 with Service Pack 1, 4.8, and 5.0 of the Single Sign-on Plug-in.

To define an action sequence

1. Select an action from among the choices in Select Actions.
2. Configure the action using the Configure Actions options. When you are satisfied with the configuration settings, click Insert. The configured action appears in Sequence Actions.
3. Repeat Steps 1 and 2 for all actions required by the user credential form.
4. Select actions in Sequence Actions and click Move Up or Move Down to arrange them in the correct execution sequence required by the user credential management form being defined.
5. When satisfied that the action sequence is correct and complete, click OK. This action returns you to the Define form actions page with the defined action sequence in the Action sequence area.
6. Click Next to continue the form definition process on the Configure other settings page. If any combination of form actions limits the defined sequence to the plug-in/agent of only Password Manager 4.5, Password Manager 4.6, Password Manager 4.6 with Service Pack 1, Single Sign-on 4.8, and Single Sign-on 5.0, a message appears to allow you to continue or return to modify your configuration.

Considerations for Windows Type Definitions

When defining Windows type application definitions, consider the following:

- Application templates help reduce the effort of creating application definitions.
- Test your application definitions with the plug-in software before you make them available to users.
- Most application definitions work using only the basic information. If an application definition does not work as expected in your test environment, it may be due to unique features such as a dynamic window title, dynamic control IDs, or other special identifiers or actions that were programmed into the application.
- To export application definitions from your test environment to your production environment, use the Export administrative data task from the Single Sign-on component of the Citrix AppCenter.
- Settings that are selected at the application definition level apply to all forms within the application definition.
- Some settings that are selected at the application definition level can be overridden at the form level. For example, for an application with three defined forms, the auto-submit can be enabled at the application definition level. Each time the plug-in software encounters one of these three forms for this application, the user credentials are supplied and submitted automatically. However, auto-submit can be disabled for one of the forms at the form level and the plug-in software will not submit the information for that specific form automatically—in this case the user is required to click Submit or OK for the selected form.
- To create a hot key for the custom field name, place an ampersand (&) in the field name immediately before the letter you want to specify as the hot key.
  If no hot key is identified, the plug-in software dynamically appends a numeric value as the hot key for the control. This will appear on the button as (1) or (2) depending on the number of custom fields that are defined.

  Be sure to test the resulting form to ensure that the defined name does not exceed the amount of space allocated to the custom field name.

Redirect to Windows Application Configuration

When no form is recognized for the Web application in the Web Form Wizard, the form definition must be redirected to use a form definition defined for a Windows application.

Forms may not be recognized when the Web application uses ActiveX controls, Flash-based controls, some types of Ajax controls, or other non-HTML-based controls used to manage user credential management events.

In these cases, ensure that the Redirect to Windows application check box is selected on the Name form page. Click Next to progress through each of the remaining Form Definition Wizard pages, and click Finish on the Confirm settings page.

The form recognition characteristics and credential actions must now be defined using Windows type definitions and send key actions.

Identifying Windows Forms with Advanced Matching

Feb 06, 2011

The Identify forms page of the Form Definition Wizard provides enough form identification matches for most Windows applications. Some user credential management forms require additional identifiers. For these forms, Single Sign-on offers Advanced Matching. You can access this feature from the Identify form page of the Form Definition Wizard by clicking Advanced Matching.

Advanced Matching offers five advanced identifiers for Windows applications:

- Class Information
- Control Matching
- SAP Session Information
- Window Identifier
- Identification Extensions

Ignoring Forms Using Class Information

Using the Class Information page, you can identify forms you want Single Sign-on to ignore. If you type a Window class into the Ignore this window class field, the Single Sign-on Plug-in does not react when a form with that class information appears.

Do not use this type of matching for .NET applications or applications that use Windows class 32770 (default class).

This setting is useful when the Window class is dynamic. In this case, use wildcard characters to match a dynamic Windows class identifier.

Wildcard	**Description**
?	Use only for a single dynamic/changing character.
*	Use this value to represent dynamic identifier data for one or more characters. This value is not recommended for empty Windows class identifiers. Use NULL for these situations.
NULL	Use this value for empty Windows class identifiers (the word “NULL” must be all uppercase).

Use Windows class identifiers when trying to identify one Windows class from among many possible Windows class targets. The following conditions apply:

- The specified window title and associated executable file result in multiple matching candidates. This condition most often occurs when the Windows title contains dynamic data and wild cards are specified.
- The target form must be associated with a unique Window class identifier and all other candidates must use different Window class identifiers.

To identify class information

Start this procedure on the Identify form page of the Form Definition Wizard.

1. Click Advanced Matching and then select the Class Information option.

2. Click Select to choose the target application from among the applications currently open on your computer.
   Note: To expand your choices, select the Show hidden program windows check box or the Show child windows check box.

Defining Matching Criteria With Control Matching When Associated Identifiers are Identical

Some applications assign dynamic information to control labels. In these cases, the window title, its associated executable application, and the control ID (or IDs) can be the same for several different user credential management forms while the text labels or other properties on the form change in response to application-specific events.

For these types of forms, use the control matching configuration options to uniquely identify a form for a specific plug-in action based on the unique class, style, or text values associated with control ID (or multiple control IDs if multiple definitions are required to uniquely identify the form).

To define matching criteria

Start this procedure on the Identify form page of the Form Definition Wizard.

1. Click Advanced Matching and then select the Control Matching option.
2. Click Add Match.
   Note: Define only enough control matching criteria to uniquely identify the user credential management form being defined.
3. From the Define Matching Criteria dialog box, click Select.
4. Right-click a control ID entry.
5. Select Class, Style, or Text to choose a characteristic to be used to qualify the form for the selected control ID.
6. Repeat Steps 4 and 5 for each control ID that is to be used to uniquely identify the form.

Identifying Matches When Using Multiple SAP Sessions

Older versions of SAP are managed using the standard Windows and Web application definitions. However, the Advanced Matching dialog box provides support for SAP applications when multiple SAP systems are defined to use the same SAP GUI user logon interface (such as SAP Logon Pad).

SAP Session Information support requires that the SAP administrator enable GUI scripting on the server. This allows the console and Single Sign-on Plug-in to interrogate the SAP Logon Pad and determine the System ID or Server name (or both) required to uniquely identify the specific user credential management form.

By using the SAP Session Information option, the session information can be extracted from an SAP window to uniquely identify and differentiate one SAP logon window from another.

To manually define SAP session information

The SAP System ID and Server Name field values can be manually entered. Both fields accept regular expressions for their respective values. This is useful for controlling the ability to match multiple servers.

You may also want to manually enter the values to match DNS and NetBIOS names of a server.

Use the following regular expression format to support both DNS and NetBIOS.

```
^servername(\.domain\.com)?$
```

To generate an SAP GUI scripting message

SAP GUI scripting messages can be generated whenever a program attempts to establish a connection to the SAP LogonPad using the SAP GUI. In this case, a registry setting can be changed to prevent the message.

The key is HKEY_CURRENT_USER\Software\SAP\SAPGUI Front\SAP Frontend Server\Security\WarnOnAttach. It is a DWORD. If this key value is set to 0, a message is not shown. The default value is 1.

Web Type Application Definitions

May 09, 2015

Web type application definitions are used to identify Web-based applications, including Java applets.

Typically, any application that runs in a browser is categorized as a Web application for the purposes of defining an application definition. Single Sign-on supports Web applications running on Internet Explorer Versions 6.0, 7.0, 8.0 and 9.0.

Web application definitions are created, in part, by identifying parts of the Web application as it runs. To gather the information required for Web application definitions, launch the application and navigate to the form that requires a user credential management event (user logon, change password, successful password change, or failed password change) while running the Form Definition Wizard from the console or from the Application Definition Tool. The wizard's on-screen text provides instructions for locating and identifying the applicable parts of the application.

Name Form

When creating application definitions for Web type applications, the Name form page of the Form Definition Wizard is used to:

- Assign a user-defined name to the form being created
- Identify the type of form being created
- Identify any special actions

Consider that the name assigned to the form appears on the Manage forms page of the Application Definition Wizard. Assign a name that is meaningful to the type of form being defined.

Several types of standard user credential processing forms can be defined using the Form Definition Wizard including:

- Logon form
  Used to identify the logon interface to an application, and to manage the user credential actions required to gain access to the associated application.

- Password change form
  Used to identify the password change interface to an application, and to manage the user credential actions required to change the user password to the associated application.

- Successful password change form
  Used to identify the password change interface to an application, and to manage the user credential actions required to acknowledge the successful change to a password for the associated application.

- Failed password change form
  Used to identify the unsuccessful password change interface to an application, and to define the actions to take when a credential change operation is unsuccessful.

Password Manager Agent Versions 4.0 and 4.1 do not support successful or failed change credentials forms and do not respond to application definitions containing these forms.

Use the Special actions area to identify any special form treatments for the form being defined:

- No special action

Select this option for normal Web form processing.

- Redirect to Windows application
  Select this option when no form is recognized for the web application in the Web Form Wizard. This occurs when the Web application uses ActiveX controls, Flash-based controls, some types of Ajax controls, or other non-HTML based controls used to manage user credential management events.

- Ignore this form when it is detected by the plug-in software
  Select this option to have the plug-in software ignore the form.

Identify Form

When creating application definitions for Web type applications, the Identify form page is used to provide the information required to have the Single Sign-on Plug-in software uniquely recognize the form being defined.

Web applications are identified using the URL address associated with the user credential management form being defined.

Click Select to open the Web page selector. Use the Web page selector to identify the Web page you want to associate with the form.

After completing the Web page selector, you are returned to this page. Two check boxes are available to manage how to interpret identified URLs:

- Strict URL matching
  Select this check box to recognize only user credential management events from Web applications that are started using the specified URL(s). Some URLs may contain dynamic data such as session management identifiers, application parameters, or other identifiers that can change for each instance. In these circumstances, using strict matching results in the URL not being recognized.

- Case-sensitive URL
  Select this check box to use exact case matching URL(s).

Define Form Actions

The Define form actions page is used to define the actions that must be performed by the Single Sign-on Plug-in to submit the credentials for the specific form being defined.

The top of the page displays the selection of user credentials associated with the specific form:

	Logon Form	**Password Change Form**	**Successful Password Change Form**	**Failed Password Change Form**
Username/ID	X	X	X	X
Password	X		X	X
Old Password		X		
New Password		X		

Confirm Password	Logon Form	X Password Change Form	Successful Password Change Form	Failed Password Change Form
Custom Field 1	X		X	X
Custom Field 2	X		X	X
OK	X	X	X	X

The bottom of the page displays the defined action sequence.

The objective of this page is to define the actions to be taken by the plug-in software to successfully submit the required user credentials to the identified form.

For many Web applications, the following process is all that is required:

1. Click the Set/Change hyperlink associated with a specific user credential. This action opens the Configure Field Text dialog box used to identify the field to receive the selected credential. If the form is already open, this dialog box displays all the possible candidates for the field type associated with the selected user credential or submit option.
   If the application credential form is not currently open, start the application and navigate to the correct user credential form. Then select the Refresh . After the application form is selected, this dialog box is populated with field type candidates that are appropriate for the selected user credential.

2. Select the field type candidate to receive the credential. As the different candidates are selected, the associated field type is visibly highlighted on the application to make it easier to identify the field type that is to receive the identified user credential or submit button.
3. Repeat this action for all the user credentials required by the form and for the button required to submit the form. Some forms require domains or other user-configurable credentials that must be successfully submitted to process the form. To accommodate these requirements, two custom fields are made available. Assign special-requirement credentials to these fields. The names associated with these fields are defined on the Name custom fields page of the Application Definition Wizard after the form is defined.

   Note: Not all the credentials identified in the top of the Define form actions page must be configured.

For many Web applications, after you define which fields on the form are to receive the identified user credential and which button to select to submit the form, you have completed the form action definition process and you can continue with the next page in the wizard.

However some forms require more information, steps, special keys, or other actions to complete a credential management task successfully. For these forms, click Action Editor to open the Action Editor dialog box.

Defining Action Sequences for Web Forms Using the Action Editor

Use the Define form actions page to define actions that must be performed by the plug-in software to submit the credentials for the specific user credential management form being defined.

For many Web applications, the basic information gathered in the Form Definition Wizard is enough to define the form. However some forms require more information, steps, special keys, or other actions to successfully complete a user credential management task. For these forms, on the Define form actions page, click Action Editor to open the Action Editor dialog box.

The Action Editor for Web dialog box consists of:

- Select Actions
  Displays all possible action-sequence actions:
- Configure Actions
  Used to define the action-specific options to include in the action sequence.
- Sequence Actions
  Displays the sequence of defined actions to perform to process the specific user credential management form.

Configure Other Settings

For Web definitions, the Configure other settings page is used to specify if the Web page's submit button is pressed automatically by the plug-in software or if the user is required to manually press the button.

Select Submit this form automatically to submit the form without user intervention.

Advanced Settings Dialog Box for Web Applications

Jan 18, 2010

Some Web applications use dynamic URLs. When this condition is encountered, additional form definition criteria (referred to as detection matching entries) must be used to uniquely identify a specific user credential management form.

These detection matching entries are defined using the Match Detail dialog box and appear on the Advanced Settings dialog box. To access the Match Detail dialog box, click Advanced Matching on the Identify form page to access the Advanced Settings dialog box, then click Add.

Use the Match Detail dialog box options and controls to define the criteria used to uniquely identify a specific user credential management form. It works by looking for specific values in the tagged content of the HTML form presented to manage a specific user credential management action. You need to define only enough match conditions to uniquely identify the user credential management form being defined.

Type the Web element you want to match in the Find box. If the element is not found, expand the Additional Settings section to manually identify the element.

The Additional Settings section is divided as follows:

- Tag
  This field is used to search for the identified HTML tag. If the specific instance of the tag is known, select Match this instance and identify which instance in the document to use. If no specific instance is identified, all instances in the document are evaluated. Only the tag needs to be specified, not the delimiter (for example p rather than <p>). As a guideline, select the tag nearest to the content you are matching.

  Note: Because the Match this instance option can vary from browser to browser, use this feature only when necessary and test your configuration well.
- Value Type
  This area is used to define the criteria to match. Select one of the following criteria:

Criteria	Description
Text	Can be any text found in the HTML code.
HTML	Any specific code found within the specified tag.
Attribute	Any attribute of the HTML code (such as a name attribute of a form tag)

- Value to match
  This field is used to enter the value to match. Select Match the whole value to enforce strict matching of the value (any unspecified text that is in the tag element will cause the match to fail). Include all delimiters and quotes that could be encountered.

  Note: Match the whole value should be selected only when there are multiple instances of similar matching criteria.
- Operator
  This area is used to define the relationship of this match entry to others defined for this form. The options include:

Options	Description
AND	Select this option when this match entry is one of multiple matches that must succeed to identify the form. By selecting this option the current match outcome is compared with the next match outcome. If both are true, the match succeeds.
OR	Select this option when this match alone can successfully identify the form. By selecting this option, the current match outcome is compared with the next match outcome. If either is true, the match succeeds. This option is used for single match definitions.
NOT	Select this operation to apply negative logic to the operator. This operator is used to define match criteria that should not appear on the page to succeed.

Terminal Emulator Type Application Definitions

Feb 14, 2011

Terminal Emulator type application definitions are used to identify terminal emulator-based applications including mainframe, AS/400, OS/390, or UNIX. Single Sign-on provides Single Sign-on functionality to terminal emulator-based applications that implement a High-Level Language Application Programming Interface (HLLAPI), or that have a built-in scripting language that can display a dialog box.

Gathering the Information Required for Terminal Emulator Application Definitions

Usually the best (and simplest) way to gather the information required for terminal emulator (HLLAPI) application definitions is to launch the application.

Terminal emulator-based application definitions are created using the Form Definition Wizard. The wizard is used to identify one or more text strings that must be present (or not present) on the terminal emulator-based application screens for a specific user credential management form (user logon, change password, successful password change, or failed password change).

As you navigate to the user credential management form being defined, record all the user actions required to access the form. These actions must be provided in the form definition for each form while running the Form Definition Wizard from the console or from the Application Definition Tool.

After identifying the correct user credential management form, the coordinates of the data entry fields used for submitting the appropriate user credential information to the application are defined. These are defined by specifying the sequence of actions, or keystrokes required to move between fields or screens and enter text.

Form Definition Process

Mar 24, 2011

The form definition process consists of collecting the form-specific identification information, and action information using the following pages in the Form Definition Wizard for Web applications:

- Name form
- Identify form
- Configure other settings
- Confirm settings

After completing the actions required for a specific page, click Next to proceed through the wizard. The Back button is generally available on each page to return to some previously configured options. However, changing previously configured options may require you to alter subsequent settings.

Name Form

When creating application definitions for terminal emulator (HLLAPI) type applications, the Name form page of the Form Definition Wizard is used to:

- Assign a user-defined name to the form being created
- Identify the type of form being created

Consider that the name assigned to the form appears on the Manage forms page of the Application Definition Wizard. Assign a name that is meaningful to the type of form being defined.

Several types of standard user credential processing forms can be defined using the Form Definition Wizard including:

- Logon form
  Used to identify the logon interface to an application, and to manage the user credential actions required to gain access to the associated application.

- Password change form
  Used to identify the password change interface to an application, and to manage the user credential actions required to change the user password to the associated application.

- Successful password change form
  Used to identify the password change interface to an application, and to manage the user credential actions required to acknowledge the successful change to a password for the associated application.

- Failed password change form
  Used to identify the unsuccessful password change interface to an application, and to define the actions to take when a credential change operation is unsuccessful.

Password Manager Agent Versions 4.0 and 4.1 do not support successful or failed change credentials forms and do not respond to application definitions containing these forms.

If the terminal emulator you are using displays more than one logon or password change page, you must create a form for each page.

Identify Form

When creating application definitions for terminal emulator (HLLAPI) type applications, the Identify form page is used to provide the information required to have the Single Sign-on Plug-in software uniquely recognize the form being defined.

Terminal emulator-based applications are identified by locating text strings that appear at specified row and column locations on the terminal emulator-based application page. Only enough text string matches required to uniquely identify the host need to be defined.

To add a text-match qualification entry

1. Ensure that the terminal emulator-based application is started and that you already determined the text strings to be used to uniquely identify the target application.
2. On the Identify form page of the Form Definition Wizard, click Add to add a new text match entry to the list of text match entries used to qualify the application. This action opens the Text to Match dialog box.
3. Complete the following fields on the Text to Match dialog box:
   - Text string
     Enter the exact text that will be used to identify the application.

   - Row
     Enter the exact row number for the string.

   - Column
     Enter the exact column number for the string.

   Note: When the plug-in software scans a terminal emulator-based application, the screen is searched for the exact text string to appear at the defined row and column location. If the text at the defined coordinates does not match the specified text, the screen is ignored.
4. Click OK. The defined Text to Match entry appears on the Identity form page.

Often, more than one text string must be defined to exactly identify the correct start of the target terminal emulator-based application. If more Text to Match strings are required, repeat Steps 2 through 4 for each string.

Set Field Detection Rules

The Set field detection rules page is used to identify the location and key actions required to manage the user credential form being defined.

The objective is to create field entries that indicate the user credential to process, the location on the screen where the user credential is to be inserted (row and column coordinates), and the keystrokes required to advance the cursor to the next credential or submit action.

To add a field entry

1. Click Add to open the Define Field dialog box.
2. Complete the following fields on the Define Field dialog box:
   - Field function
     Select the user credential to be submitted from among the choices that appear in the drop-down list box.

   - Row
     Enter the exact row number for the string.

   - Column
     Enter the exact column number for the string.

   - Keys after

Enter the key codes required to advance to the next credential field or to perform the submit action

Note: Select the Virtual key codes hyperlink to access help information about the valid key codes.

3. Click OK. The defined field entry appears on the Set field detection rules page.
4. Repeat Steps 1 through 3 for each user credential required by the form being defined.
5. The field entries displayed on the Set field detection rules page are processed from top to bottom as they appear on the page. Use the UP ARROW and DOWN ARROW keys to arrange the entries in the sequence required by the user credential form being processed.

Configure Other Settings

The Configure other settings page is used to access advanced settings options for the form being defined. Advanced settings include:

- Defining an initial form processing delay
- Defining the keystrokes required to access the user credential management form being defined
- Defining text string matching criteria that tells the plug-in software to ignore processing

If any additional advanced configuration is required for the user credential management form being defined, click Advanced to open the Advanced Settings dialog box.

Advanced Settings for Terminal Emulator-Based Applications

Jan 14, 2010

Some terminal emulator-based applications require additional configuration support to ensure that the correct user credential management form is identified. That might include:

- Waiting a defined amount of time for the terminal emulator-based application to start before attempting to identify the application
- Processing a series of keystrokes to navigate to the initial logon page or change password page
- Ignore processing a page when specific text appears

When advanced configuration settings are required for the user credential management form being defined, click Advanced on the Configure other settings page of the Form Definition Wizard to open the Advanced Settings dialog box.

The Advanced Settings dialog box has two configuration pages that are accessed from the left panel on the page:

- Highlight the Host Form Additional Settings option to access the Additional settings options:
  - Delay field entries (ms). Enter the number of milliseconds to delay processing the form while waiting for the application to complete loading.
  - Keys before. Enter the virtual key codes that must be entered to access the first field of the user credential management form being processed. Select the Virtual key codes hyperlink to access the help for the valid virtual key codes.
- Highlight the Ignore Match option to access the Text match to stop credential submission option. This option is used to specify text strings that appear on the application page for forms that are to be ignored.

Considerations for Terminal Emulator Type Definitions

Nov 05, 2009

Consider the following factors when defining terminal emulator (HLLAPI) type application definitions:

- Terminal emulation support must be enabled for each user configuration that uses terminal emulator-based applications.
- Verify that your terminal emulator program is HLLAPI compliant.
- Verify that your terminal emulator program is defined in the plug-in software mfrmlist.ini file.
- Save time by using a terminal emulator that shows the row and column coordinates of the cursor position. This allows you to more easily determine the location of the text and fields used to identify the host application and its logon forms.
- For HLLAPI detection, the terminal emulator must set a short name for each session. The plug-in software cannot detect an application without the terminal emulator's session short name.
- The documentation for your terminal emulator-based application may include unique identifiers, such as screen numbers, for the screens used to submit user logon information. In this case, use the screen number as the unique identifier that ensures the plug-in software identifies and submits credentials to the correct form.

Terminal Emulation Support

Feb 07, 2011

The supported terminal emulators are included in a Mfrmlist.ini file. This file represents all the terminal emulators tested by Citrix.

It is possible to add terminal emulators to this list. However, these definitions should be tested and verified before being introduced into your production environment. A sample section of this file is included below:

```
[Emulators]
Ver=20021101
EMU1=Rumba6
EMU2=Attachmate myExtra!
EMU3=Attachmate Extra! 6.3
EMU4=Attachmate Extra! 6.4
EMU5=Attachmate Extra! 6.5
EMU7=Attachmate Extra! 7.1
EMU8=Reflection7
EMU9=Reflection8
EMU10=Reflection9
EMU11=Reflection10
EMU12=PCOM
EMU13=HostOnDemand 4.1
EMU14=GLink
EMU1EMU16=ViewNow5=Aviva
EMU16=ViewNow
EMU17=ZephyrPC
EMU18=ZephyrWeb
;EMU19=BOSaNOVA
;EMU20=HostExplorer6
;EMU21=HostExplorer8
[Rumba6]
DisplayName=Rumba
RegistryLoc=WALLDATA\Install
ValueName=
DLLFile=SYSTEM\EHLAPI32.DLL
UpdateNotificationHandling=0.FirstLogin
Process=shared
ConvertPosType=long
QuerySessionsType=long
QuerySessionStatusType=long
QueryHostUpdateType=long
StartNotificationType=long
IntSize=16
WindowClass=WdPageFrame
WindowTitle=RUMBA
```

The terminal emulator entries in the [Emulators] section of the Mfrmlist.ini file must be in numeric sequence, from EMU1 up to and including EMU99. Any break in sequence causes the Ssomho.exe process to terminate before reading all of the entries.

Removing or commenting out unused terminal emulators can improve the startup process. Ssomho.exe does not waste resources or time scanning for the location of unnecessary HLLAPI DLLs.

To comment out an entry, move the entry to the bottom of the list, place a semicolon before the entry, and then renumber the remaining EMU entries so no numeric value is skipped.

Single Sign-on cannot globally update this mfrmlist.ini file; you must overwrite the file manually after installing the plug-in. For large deployments, consider using batch files or scripts run through System Management Server (SMS), CA-Unicenter, or Active Directory software installation.

Mfrmlist.ini Field Definitions

Feb 07, 2011

Terminal emulators added to the Mfrmlist.ini file will function only if they follow the HLLAPI standard. The field definitions for the Mfrmlist.ini file are provided below. If you must add a terminal emulator definition, check with the terminal emulator's manufacturer to determine whether or not the terminal emulator supports HLLAPI and to obtain the correct field definition entries. To determine whether or not a terminal emulator works with Single Sign-on, test it outside of your production environment.

Field	Definitions
[EmulatorName]	The value for EmulatorName must match the value used for the EMUnn=EmulatorName line in the [Emulators] section.
GroupName	Internal use only.
DisplayName	The display name of the terminal emulator, which will be one of the two parameters used when spawning a new process to handle the session. Must be unique to the Mfrmlist.ini file.
RegistryLoc	The registry key in HKEY_LOCAL_MACHINE\SOFTWARE that points to the path where the HLLAPI DLL is stored. If the program does not store this information in HKEY_LOCAL_MACHINE\SOFTWARE, use the ExplicitPath setting instead of the RegistryLoc setting. If both RegistryLoc and ExplicitPath settings are defined, the ExplicitPath setting takes precedence.
ExplicitPath	The explicit path of the HLLAPI DLL file used by this emulator. This setting is used in place of the RegistryLoc setting when the emulator program does not store the HLLAPI DLL location in the system registry. If both RegistryLoc and ExplicitPath settings are defined, the ExplicitPath setting takes precedence.
ValueName	The name of the value in the RegistryLoc key that contains the actual path value.
DLLFile	The name of the HLLAPI DLL file.
StripFileName	Indicates the value stored in ValueName contains a backslash \ that must be stripped when assembling the HLLAPI DLL path from ValueName and DLL File entries.
IntSize	Defines the integer size supported by the terminal emulator, 16-bit or 32-bit.
WindowClass	The Window Class name for the terminal emulator. Obtained by using the Single Sign-on console or the Application Definition Tool.
WindowTitle	A portion of the Window Title that can be used by Single Sign-on to ensure this window is associated with the terminal emulator. Must contain at least one word

Field	Definitions
	that will always be in the Windows title. Wildcards are assumed on either side of the text
UseSendKeys	Instructs Single Sign-on to use SendKeys for communicating with the terminal emulator. The option is not the same as the one used for Windows applications.

Creating User Configurations

May 09, 2015

A user configuration enables you to control the behavior and appearance of the plug-in software for users. Creating one or more user configurations is the final step you take before distributing Single Sign-on Plug-in software to users in your environment. Note that you can add new or edit existing user configurations at any time.

A user configuration is a unique collection of settings, password policies, and applications that you apply to users associated with an Active Directory hierarchy (organizational unit [OU] or an individual user) or Active Directory group.

A user configuration consists of the following:

- Users associated with an Active Directory domain hierarchy (OU or individual user) or Active Directory group
  Important: Distribution groups and Domain Local groups in Active Directory mixed mode are not supported.
- License type and related settings associated with the users (concurrent or named user license model)
- Data protection methods
- Application definitions that you created, which you can combine into an application group when you create a user configuration
- Password policies associated with any application groups
- Self-service features (account unlock and password reset) and key management options (use of previous passwords, security questions, and automatic key management)
- Settings for options such as credential provisioning and application support

Before you create your user configurations, ensure that you already created or defined the following:

- Central store
- Application definitions
- Password policies
- Security questions

You must create user configurations before you deploy the Single Sign-on Plug-in software to users. Among other settings, a user configuration contains the license server and licensing information required by the plug-in software for operation.

For user configuration setting defaults and details, see the topics under

— Single Sign-on Settings Reference > User Configurations

.

To specify a domain controller for an existing user configuration

In environments where you use an Active Directory-based central store and have more than one domain controller, you can select the domain controller to bind user configurations to when writing to the central store.

This binding scheme helps to reduce synchronization delays caused by Active Directory replication. Such delays might occur in environments where users access Single Sign-on in multiple Active Directory sites simultaneously.

During the discovery process available through the console, Single Sign-on can discover every domain controller in your domain. You can then bind user configurations that you created to a specific domain controller by selecting that controller when you create a user configuration.

For example, you can require users to be bound to a domain controller within their local network. After you specify a domain controller, users are bound to that domain controller the next time they log on to Single Sign-on.

By default, users bind to any writeable domain controller until you select a domain controller they must bind to. You can change the domain controller setting at any time by updating the user configuration as needed without losing user data integrity.

Note: When choosing a domain controller for binding, verify that the resources available on the domain controller can accept the communication traffic users generate when connecting to the domain controller during peak operational times. If the specified domain controller is unavailable or offline, the plug-in software uses the local store's user data (that is, the user data located on the user's computer). If the domain controller is offline for a long period of time (as defined by you), you can select the Edit User Configuration task from the console and choose another domain controller or the Any writeable domain controller option.

1. Click Start > All Programs > Citrix > Management Consoles > Citrix AppCenter.
2. Expand the Single Sign-on node and User Configurations.
3. Select a user configuration.
4. From the Action menu, select Edit user configuration.
5. Select Domain Controller from the options on the left side of the Edit User Configuration wizard page.
6. Select an available domain controller or select Any writeable domain controller.

To create a user configuration

1. Click Start > All Programs > Citrix > Management Consoles >Citrix Deliver Services Console.
2. Expand the Single Sign-on node and select User Configurations.
3. From the Action menu, click Add new user configuration.

Naming Your User Configuration

The Name User Configuration page of the User Configuration Wizard allows you to name your user configuration as well as choose how you will associate the user configuration to the users.

- Name
  Consider naming the user configuration according to how you plan to group your users and associate them with specific applications. For example, Marketing Users, Software Development Users, North American Users, and so on.

- User configuration association
  You have two choices: associate users according to Active Directory hierarchy (OU or individual user) or Active Directory Group. If necessary, you can associate the user configuration with a different hierarchy or group later, by clicking Move user configuration in the Action menu.

  Important: How you organize your Active Directory environment might affect how user configurations operate. If you use both (Active Directory hierarchy and group) and a user is located in both containers, the user configuration associated with the hierarchy takes precedence and is the one used. This scheme is considered a mixed environment. Also, if a user belongs to two Active Directory groups and each group is associated with a user configuration, the user configuration with the highest priority takes precedence and is the one used.

  Associating user configurations to groups is supported only in Active Directory domains that use Active Directory authentication.

Specifying a Domain Controller

If you are using an Active Directory central store, the Specify Domain Controller page of the User Configuration Wizard enables you to select an available domain controller or select Any writeable domain controller.

Choose Applications and Configure User Settings

Feb 11, 2011

From the Choose applications page of the User Configuration Wizard, add the applications for the user configuration. When you click the Add button, a dialog box showing the application definitions you created previously appears. You can now combine these application definitions in an application group. An application group can contain several applications or as few as one application.

You can also make the application group a password sharing group to automate and simplify the password change process. If the password for an application definition that is part of a password sharing group changes, the plug-in software ensures that the password change is reflected in the stored credentials for all applications in the group.

Password sharing groups enable the plug-in software to manage multiple credentials for applications that use the same authentication authority. For example, if you have two applications that use the same Oracle database to authenticate, such as a financial application and a human resources application, you can place these two applications in the same password sharing group. When your users change their password for either application, the other application's credentials are updated automatically.

Important: For best results, ensure that all passwords in the password sharing group are managed by a common authentication authority. For example, you would implement a password sharing group if the applications in a password sharing group share a common back-end authentication authority like a database, where the user would submit the same credentials to each application to authenticate to the database. You would not group unrelated applications like an email program, a Web application, and a custom Single Sign-on enabled program on your intranet where a user could potentially submit three different sets of credentials, but only by coincidence is using the same credentials for all three applications. In this case, if a user changed the credentials for one application in this password sharing group, it does not necessarily follow that those credentials would be valid for the other two applications.

Configure User Settings

Use the following pages to configure user settings. For setting details, see the topics under

— Single Sign-on Settings Reference > User Configurations

.

- The Configure Single Sign-on Plug-in interaction page of the User Configuration Wizard enables you to determine the user experience for all plug-in software users in your environment.
- Select a license server and licensing model at the Configure licensing page of the User Configuration Wizard. Important: If you edit the user configuration later and change product editions, your license model will change. For example, changing the product edition from Single Sign-on Enterprise to Single Sign-on Advanced will change your licensing model from Concurrent User to Named User.
- The Select data protection methods page of the User Configuration Wizard enables you to select the data protection methods to protect user credentials based on the various authentication methods your users are authorized to use. In some environments, users can use more than one method.
- When users change their primary authentication (for example, a domain password change or a replaced smart card), the Select secondary data protection page of the User Configuration Wizard enables you to specify secondary credential data protection options to use before unlocking user credentials. It also enables you to require that users verify their identity for added security. Alternatively, it also enables you to specify that credentials are restored automatically by implementing the Key Management Module.
- The options available on the Enable self-service features page of the User Configuration Wizard require the installation of the Key Management Module. This feature inserts an Account Self-Service button on the Windows logon and Unlock

Computer dialog boxes and can help reduce costs associated with administrator intervention or help desk support in your enterprise.

- The Key Management Module and Provisioning module pages of the User Configuration Wizard require you to specify the URL and service port of any installed service modules.

Synchronizing Credentials by Using Account Association

Apr 13, 2011

In companies that maintain multiple Windows domains, users might also have more than one Windows account. Single Sign-on includes a service known as Credential Synchronization to enable Account Association.

Account Association allows a user to log on to any application from one or more Windows accounts. Because Single Sign-on typically binds user credentials to a single account, the credential information is not automatically synchronized among multiple accounts that a user owns. However, administrators can configure Account Association to synchronize user credentials. Users with Account Association configured have access to all applications from any of their accounts in their Single Sign-on environment. When user credentials are changed, added, or removed from one account, the credentials are synchronized automatically with each of the user's associated accounts.

Without Account Association, an individual with multiple Windows accounts is forced to manually change their logon information separately from each Windows account.

To configure Account Association, the enterprise Windows domain administrators must perform the following steps in order:

1. Choose a domain in which to install and run the Credential Synchronization Module, which is part of the Single Sign-on Service.
2. Deploy the trusted root certificate to all computers in the enterprise that will use Account Association.
3. Manually synchronize application definitions among domains.
4. Configure the Account Association user settings in other domains to connect to the Credential Synchronization Module.
5. Make the Account Association tool available to users as a published application.

Each user must enable Account Association in the Single Sign-on Plug-in.

Choosing and Configuring a Domain to Host the Credential Synchronization Module

Choose the domain that contains the accounts for all users in your enterprise who will use Account Association. The Credential Synchronization Module acts as the hub for all user credential information in the enterprise. Install this module in this domain as you would any other Single Sign-on Service.

Important: Contact your network administrator to determine if any firewall changes are necessary and if the changes are compliant with your company's policies.
After you install the Credential Synchronization Module, create or edit user configurations from the Citrix AppCenter to authorize individual user accounts to use the Credential Synchronization Module, as follows.

To configure the credential synchronization features in the host domain

Open the console from the domain that is hosting the Credential Synchronization Module. Some domains can access multiple central stores. Ensure that the console you are using is configured to connect to the same central store as the Credential Synchronization Module service.

1. Click Start > All Programs > Citrix > Management Consoles > Citrix AppCenter.
2. Expand the Single Sign-on node and select User Configurations.

3. Select an existing user configuration or create a new one.
   - If you are creating a new user configuration, the following options are available from the Advanced Settings button on the Configure plug-in Interaction page of the User Configuration Wizard.
   - If you are editing an existing user configuration, the following options are available from the Edit User Configuration properties page.
4. Click Synchronization and select Allow user credentials to be accessed through the Credential Synchronization Module.
5. Click OK and repeat Steps 3 and 4 for each existing and new user configuration.

To manually synchronize application definitions among domains

Accounts can also synchronize across different user configuration associations. That is, a user configuration can be associated with an Active Directory hierarchy (OU or user) in one domain and associated with an Active Directory group in another domain. As long as the application definition names are the same in each user configuration, the Account Association feature will synchronize credentials.

User credentials are shared only for applications defined by the Single Sign-on administrator. Administrators must ensure that each application definition on each domain has the same name in each central store.

For example, if the application definition for SAP is named SAP Logon on one domain, SAP on another, and SAP Launch Pad on another, user credentials for these applications will not be synchronized across accounts for these domains.

A best practice when creating a new application definition across domains is to use the Export application definitions and Import administrative data tasks in the console. Use these tasks to export newly-created application definitions to import into each central store. Existing, previously-defined applications must be manually renamed.

To configure Account Association user settings in other domains

Install and open the console from a workstation in each domain that is not hosting the Credential Synchronization Module. Some domains have multiple central stores; therefore, ensure that you configure each central store.

All domain administrators must allow the domain users to associate their accounts with their host domain account. Edit the Account Association section of the appropriate user configurations in the console.

1. Click Start > All Programs > Citrix > Management Consoles > Citrix AppCenter.
2. Expand the Single Sign-on node and select User Configurations.
3. Select an existing user configuration or create a new one.
   - If you are creating a new user configuration, the following options are available from the Advanced Settings button on the Configure plug-in interaction page of the User Configuration Wizard.
   - If you are editing an existing user configuration, the following options are available from the Edit User Configuration properties page.
4. Click Account Association.
5. Select Allow users to associate accounts.
   The following options are not required but help provide a seamless user experience.

6. Select Provide default service address and type the Single Sign-on Service address and port for the domain hosting the Credential Synchronization Module.
7. Clear Allow users to edit service address.
8. Select Provide default domain and type the name for the domain hosting the Credential Synchronization Module. If you do not provide the domain, users might be confused as to which domain account user credentials they should provide.

9. Clear Allow users to edit domain.
10. Depending on your company's security policies, select Allow users to remember password.
11. Click OK and repeat for each user configuration.

Publishing the Account Association Tool

Because this version of the Single Sign-on Plug-in does not provide a menu option that allows users to enable Account Association, you provide users with a tool for enabling Account Association as a published application:

1. Install the Single Sign-on Plug-in on a XenApp server.
2. Locate the AccAssoc.exe file on the XenApp sever.
3. Publish the AccAssoc.exe file and make it available to users.
4. Inform users how to access and use the Account Association tool.

Note: Users running Single Sign-on Plug-in Versions 4.8 and earlier can use a plug-in menu option to enable Account Association. These users do not require access to the Account Association tool as a published application.

To enable Account Association in the Single Sign-on Plug-in

Apr 13, 2011

When logging on to the domain hosting the Credential Synchronization Module, users do not need to perform any action to enable Account Association. These accounts act as a central repository for each user's credential information.

When logging on to other domains, users have two ways to enable Account Association, depending on the version of the Single Sign-on Plug-in they are using:

- For this version of the Single Sign-on Plug-in, users access the Account Association tool as a published application. You publish the Account Association tool and inform users how to access and use it.
- For the Single Sign-on Plug-in versions 4.8 and earlier, users will now see an Account Association option under the Tools menu in the plug-in software's Logon Manager. Users choose this option to enable Account Association.

1. Depending on plug-in version, users access the Account Association tool as a published application or select Tools > Account Association from the Logon Manager. The Account Association dialog box appears.
2. Users select Enable Account Association.
   Note: If you did not provide the service address that is hosting the Credential Synchronization Module, users must type it in the text field. If the field is unavailable, you already provided this service address and users cannot type in this field.
3. Users click OK. The Authenticate for Account Association dialog box appears.
4. Users type the username and password for the user's associated Windows account. If the domain where the Credential Synchronization Module is installed is not shown, users type it in the Domain field.
   Note: If you provided the domain name, users cannot type text in this field.
5. Users click OK. Account Association is now enabled. The user's credentials are synchronized whenever plug-in software synchronization occurs.

Managing User Configurations

May 09, 2015

Single Sign-On allows you to manage user configurations. You can:

- Reset user data
- Delete user data
- Prompt users to register again
- Set the user configuration priority
- Assign the user configuration to different users
- Upgrade the user configuration for existing users

The Reset user data task requires that you install and configure the Provisioning Module.

Reset user data enables you to reset user information in your central store, which results in the selected user being returned to an initial state.

- In Active Directory central stores, the user data (credentials, security questions and answers, and so on) is deleted and the user is flagged as having had their data reset.
- In NTFS network share central stores, the user folders are retained, all user data is deleted, and the user is flagged as having had their data reset.

You can use Reset user data if users forget the answers to their security questions or to reset their credential data if the user's data somehow is corrupted. When the user later uses the plug-in software to contact the central store, the user's local credential store is cleared of all data, and the user must reenroll.

This task is also useful when a user cannot log on to the plug-in software.

Important: Password history is retained on a per-user basis. If you reset the data for a user, the password history is removed and password history cannot be enforced for the deleted passwords.

1. Click Start > All Programs > Citrix > Management Consoles > Citrix AppCenter.
2. Expand the Single Sign-on node and select User Configurations.
3. From the Action menu, click Other Tasks > Reset user data. The Select User dialog box appears.
4. Type a user name in the text field and click Check Names.
5. If the user is found, click OK.
6. Select a user in your central store and click Reset.
7. Click OK. A warning message appears.
8. Verify that any users who might be running Single Sign-on as an application hosted by Citrix XenApp are logged off and click Continue to flag the user's data for reset.
   Note: If users are not logged off, click Cancel, reset their ICA session, and return to this procedure.
9. Click OK in the Reset User Data dialog box when the user information is verified and reset. The user's data is reset the next time the user logs on to Single Sign-on using the plug-in software.

The Delete user data from central store task deletes all user data and information from the central store. You can use Delete user data from central store when a user leaves your enterprise permanently.

The local credential store on the user computer remains intact until it is explicitly deleted by an administrator or operator.

If the plug-in software is run by the now-deleted user, the plug-in software synchronizes its local credential store with the central store unless the local credential store is explicitly deleted by an administrator or operator. To prevent this, delete this user from your enterprise (for example, disable or delete the user from Active Directory).

1. Click Start > All Programs > Citrix > Management Consoles > Citrix AppCenter.
2. Expand the Single Sign-on node and select User Configurations.
3. From the Action menu, click Other Tasks > Delete user data from central store. The Select user dialog box appears.
4. Type a user name in the text field and click Check Names.
5. If the user is found, click OK. Click Yes to confirm. A confirmation message appears.
6. Click OK. The user is now deleted from the central store.

You can prompt one user or all users to reregister answers to their security questions. You would use these features for security purposes or when user data becomes corrupted:

- Revoke security question registration for a user
  Select this option to delete a user's security question data. Any question-based authentication is unavailable until the user reregisters.

- Prompt all users to reregister security questions
  Select this option to prompt all users to reregister their security questions and answers when they launch the plug-in software. Security question data is retained and any feature requiring question-based authentication is still available with the current answers. Users are prompted until they reregister.

If users choose not to reregister their answers by cancelling the Citrix Single Sign-on Registration dialog box when prompted, they will not be able to use features that use question-based authentication such as Account Self-Service until they choose to reregister their answers.

1. Click Start > All Programs > Citrix > Management Consoles > Citrix AppCenter.
2. Expand the Single Sign-on node and select User Configurations.
3. From the Action menu, click Other Tasks and one of the following:
   - Revoke security question registration for a user
     The Select User dialog box appears. Type or select a user. Confirm that you want to revoke that user's security question registration.

   - Prompt all users to reregister security questions
     Click Yes to prompt all users, then click OK.

When you create or edit a user configuration, you can associate users located in Active Directory groups with user configurations. It is possible that a user in a group can be associated with more than one user configuration. In this case, you can set the priority of the user configuration.

Important: How you organize your Single Sign-on user environment might affect how user configurations operate. That is, you associate user configurations in your Single Sign-on environment with an Active Directory hierarchy (OU or users) or an Active Directory group. If you use both (hierarchy and group) and a user is located in both containers, the user configuration associated with the hierarchy takes precedence and is the one used. This scheme is considered a mixed environment.

1. Click Start > All Programs > Citrix > Management Consoles > Citrix AppCenter.
2. Expand the Single Sign-on node and select User Configurations.
3. From the Action menu, click Other Tasks > Set user configuration priority. The Set User Configuration Priority dialog box appears.
4. Select a user configuration and click Move Up or Move Down, according to your preference.

When you edit an existing user configuration, note that you cannot edit the user configuration location. You can perform one of the following procedures:

- Apply a user configuration to an additional set of users by duplicating it
- Apply a user configuration to a different set of users by moving it

To duplicate a user configuration

1. Click Start > All Programs > Citrix > Management Consoles > Citrix AppCenter.
2. Expand the Single Sign-on node and select User Configurations.
3. Select the user configuration.
4. From the Action menu, click Duplicate user configuration.
5. Type a name for the duplication configuration.
6. Specify the OU, user, or group that contains the users to which the user configuration will apply.

To move a user configuration to different users

You cannot move a user configuration that is associated with an Active Directory group. To associate the user configuration with an Active Directory hierarchy (OU or user), duplicate the user configuration and specify the desired association.

1. Click Start > All Programs > Citrix > Management Consoles > Citrix Citrix AppCenter.
2. Expand the Single Sign-on node and select User Configurations.
3. Select the user configuration.
4. From the Action menu, click Move user configuration.
5. Specify the OU, user, or group that contains the users to which the user configuration will apply.

In Password Manager Versions 4.0 and 4.1, you associated users to a user configuration by an Active Directory hierarchy (OU or user). In Password Manager 4.5 and 4.6 and Single Sign-on 4.8 and 5.0, you can choose to associate users by an Active Directory group.

- If you use an existing user configuration organized by hierarchy and now create user configurations organized by group and a user is located in both containers, the user configuration associated with the hierarchy takes precedence and is the one used. This scheme is considered a mixed environment. In this case, your users might experience unintended plug-in software behavior. That is, they will have access to resources associated with the hierarchy-based user configuration instead of resources associated with the group-based user configuration.
- If you want to preserve the settings in your existing hierarchy-based user configurations but change their association, move the user configuration to a different user. This procedure is applicable for Versions 4.1, 4.5, 4.6, 4.8 and 5.0 hierarchy-based user configurations.

Consider the following if you want to upgrade existing user configurations whose users are organized by OU or user:

If you upgrade the Single Sign-on Service and console but do not upgrade the plug-in software, the plug-in software will still provide basic functionality to users whose user configurations are associated with Active Directory hierarchies (organizational units or users). However, your users will not have access to the latest Single Sign-on features. Consider upgrading the plug-in software whenever possible to match the service and console versions.

User Authentication and Identity Verification

May 09, 2015

Two types of authentication exist in Single Sign-on:

- Primary authentication, which occurs when users type their primary user names, passwords, and, optionally, domain name when logging on to Microsoft Windows to access their corporate or enterprise network. The existing Windows security subsystem is responsible for managing network authentication.
- Secondary authentication, which occurs when you configure Single Sign-on to submit credentials that allow users to access protected Single Sign-on enabled resources. These resources can include an enterprise application, a Web application, a protected field in an application, an IP address, a URL, and so on.

After a successful network authentication, Single Sign-on obtains the primary password from the Windows logon and, along with other variables, uses this information to create the encryption key that protects user credentials. The plug-in software uses this key to retrieve and decrypt the credentials as applications or resources request them.

Important: If a user's password is compromised, reset the user's password twice, rather than once, to ensure that the compromised password is removed from the previous password feature. Users need to log on with each of the new passwords so that the plug-in software can capture the changes.

Each time users log on to your environment, they confirm their identity by typing their user name and password or by using a smart card or other authentication device that uniquely identifies who they are.

However, several events require a second layer of authentication to verify that the user initiating the change is the user authorized to do so:

Event	Description
An administrator changes a user's primary password	When administrators change users' primary passwords, users will then be further prompted to confirm their identities to ensure the authorized user is logged on.
Users reset their primary password using Account Self-Service	When users reset their primary password using Account Self-Service, they are prompted to further confirm their identity. Do not use the Prompt user to enter the previous password authentication option if enabling the self-service features.
Users unlock their domain account using Account Self-Service	When users unlock their account using the self-service features, they are prompted to further confirm their identity.
Users change their authentication types	For example, when users switch from smart card authentication to password-based authentication, they are prompted to further confirm their identity.
Password change on a client device not running Single Sign-on	Users who change their primary password on a client device not running the plug-in software are prompted to confirm their identity the next time they log on to a client device running the plug-in software.

Your users can confirm their identity using one or more of the options you can specify to meet your organization's requirements.

Single Sign-on includes two identity verification methods to help ensure that the user is authorized to use Single Sign-on:

- Previous password
- Security questions

You can also choose to bypass identity verification by using the automatic key management feature.

You can allow users to choose the identity verification method (previous passwords or security questions) they prefer to use when authenticating. This option is available as part of Secondary Data Protection property in the user configuration.

Previous Password

With this method, users verify their identities by typing their previous primary password.

Caution: When previous password is the only method available to your users, users who forget their previous primary password are locked out of the system. Their user data must be deleted from the central store and from all client devices on which it is stored, and they must reenter their credentials for all of their applications.

Security Questions

When users change their primary passwords, you can confirm your users' identities by prompting them to answer security questions in the form of a questionnaire you create. This questionnaire appears the first time your users launch the plug-in software. Users answer the required number of security questions and are prompted to reenter this information at specific password change events.

The questions in your questionnaire should be of a nature that ensures the person answering the question is the only person who knows or could easily provide the answer. You can use the default questions Single Sign-on provides or create your own.

Bypassing Identity Verification

Important: Automatic key management is not as secure as other key recovery mechanisms such as security questions and previous password.
If you want Single Sign-on to bypass identity verification and retrieve user encryption keys automatically, you can specify the Secondary Data Protection option Do not prompt users; restore primary data protection automatically over the network.

This method, known as automatic key management, is available when you install the Key Management Module and you create a user configuration with this option selected.

With this method, users log on to the network and have immediate access to applications managed by Single Sign-on. There are no questions to answer. When users change their primary passwords, the plug-in software detects these password changes and recovers the users' encryption keys using the Single Sign-on Service.

Automatic key management provides users with the easiest and fastest access to their applications. However, it does not protect against access by an unauthorized user because there is no user secret to protect the user's network password. To help prevent this potential problem, implement automatic key management in combination with the Self-Service Module.

This module requires question-based authentication to allow your users to confirm their identity when resetting their primary passwords or unlocking their domain accounts.

In Single Sign-on, users can switch among multiple primary authentication methods. Single Sign-on protects user passwords with a unique copy of the security key as a reauthentication method to efficiently unlock the user's data each time the user switches between authentication methods, without the user having to verify identity.

The option to select multiple primary authentication methods is available as part of the Data Protection Methods page in the user configuration.

Consider the following user scenario:

- A call center supervisor logs on to a computer using primary credentials (Windows user name and password). Single Sign-on Plug-in software is installed on the computer and allows the supervisor to use Single Sign-on (SSO) enabled applications.
- The supervisor occasionally uses a smart card with PIN to log on to a shared computer on the call center floor and launch another published application through XenApp. This computer uses Hot Desktop to enable fast user switching among different accounts.

In Citrix Password Manager Versions 4.0 and 4.1, the call center supervisor is required to verify identify before using the SSO-enabled applications when changing primary authentication methods. In this use case, the supervisor used two primary authentication methods: first a user name and password, then a smart card with PIN. Password Manager Versions 4.0 and 4.1 treat the change of authentication method as requiring security key recovery and possibly required the supervisor to verify identity.

Users are required to register or enroll each new authentication method the first time they use or switch to the method. However, later switches do not require a registration or enrollment (that is, a key recovery is not subsequently required).

Managing Question-Based Authentication

May 09, 2015

Question-based authentication allows you to provide secure authentication to users who change their primary password under specific circumstances, change their method of authentication, or have their accounts locked.

The use of security questions and question-based authentication can help protect against access by unauthorized users by requesting information known only to your individual users. The questions you create must request non-public information that would be difficult for anyone other than the authorized users to provide or find (for example, difficult for brute force guessing, dictionary based attacks, and so on).

Important: If you plan to use the password reset or domain account unlock self-service features available from the Single Sign-on Key Management Module, you must use the question-based authentication method to allow your users to confirm their identity when resetting their primary passwords or unlocking their domain accounts.

If you are implementing the password reset or domain account unlock self-service features available from the Single Sign-on Key Management Module, use question-based authentication for user identity verification. You can also choose question-based authentication as a form of secondary data protection if a user's primary authentication changes.

Depending on the user configuration settings in the console, users might be required to verify their identities when the following events occur:

- Users change their authentication types; for example, a user might switch between smart card and password authentication.
- An administrator changes a user's primary password
- Users reset their primary password using Account Self-Service
- Users unlock their domain account using Account Self-Service
- Users change their primary password on a device that does not have the plug-in software installed and then log on to a device where the plug-in software is installed

Note: You can also create a user configuration that does not require subsequent verification when switching among authentication types; see

— If Users Switch among Multiple Primary Authentication Methods

.

If configured, the Single Sign-on Plug-in software prompts users to answer the security questions during first-time use. When one of these events occurs that requires users to verify their identity, the plug-in software launches the questionnaire you created for them. A questionnaire is a preconfigured list of questions you create.

Each question in the questionnaire appears on a separate page. For example, if five questions are in your questionnaire, users will see five separate pages—one for each question. Users must answer every question correctly. Depending on administrator settings, answers must be an exact match, including case and punctuation, to the answers users gave when Single Sign-on was launched for the first time.

The correct combination of questions and answers confirms the user's identity. After a user is confirmed, the plug-in software encrypts the keys again using the new primary password and stores the user's secondary credentials.

Considerations

- If you choose not to configure answers to security questions as required for your users, users are prompted for their previous primary password when they change their primary password and attempt to log on with their new password. You can allow users to choose the identity verification method they prefer to use when authenticating. This option is available as part of the Secondary Data Protection property in the user configuration.

- To prevent user lockout, do not combine the Account Self-Service password reset feature with the Prompt user to enter the previous password option. Users who reset their password are unlikely to recall their previous primary password and cannot retrieve their secondary credentials.
- Multiple questions provide the best data protection.
- By default, Question-Based Authentication is populated with four security questions. While you can use these four questions exclusively, consider adding your own security questions and question groups.

Important: Depending on administrator settings, alphabet case usage, punctuation, and spaces are included in the user's answer and must match exactly when the user is asked to answer the selected security questions at a later date.

Create and make available your security questions before deploying the plug-in software. After a user selects a question, that question must always be available. If you change or remove a question that is in use, those users cannot use the security questions to recover their secondary credentials until and unless you force them to re-enroll.

1. Create your security questions, defining the minimum length and case sensitivity. These questions can be made available in the languages Single Sign-on supports.
2. Optionally, group these questions in security question groups. You can create a number of questions for your users to choose from, giving them flexibility to choose a question to which they are more likely to recall an answer. This allows you to define the number of questions from each group that users are required to answer.
3. Add your questions, or questions and question groups, to your questionnaire.
4. Select one or two questions to be used for key recovery. These questions are used to encrypt the data for key recovery; your users will still be required to provide answers for questions they answered at enrollment.
5. Optionally, enable security questions answer masking. This feature provides you the option to mask user answers to question-based authentication security questions. If enabled, users' answers are protected during answer registration and identity verification.
   Security question answer masking is available on console and plug-in software running Password Manager 4.6 and 4.6 with Service Pack 1 and Single Sign-on 4.8 and 5.0.

Single Sign-on provides four default questions that you can use to manage user registration. These questions are available in all supported languages (English, French, German, Japanese, Simplified Chinese, and Spanish). Citrix recommends you create your own security questions and make them available in each of the languages your environment must support.

Someone trying to gain access to a user password needs to know the answers to all the questions the user originally answered. Consider that requiring users to answer too many questions might make it too difficult for your users to confirm their identity.

Security questions should request non-public information that would be difficult for anyone other than the valid user to provide (for example, difficult for brute force guessing or dictionary-based attacks). The key factor in determining the security of questions is the degree of difficulty involved when someone attempts to guess the answer.

A good questions is one that has high entropy; that is, a question for which:

- The number of unique answers possible is very high
- The probability of guessing any one specific answer is very low

For usability purposes, the question should be easy for a user to remember but difficult for an adversary to determine. For example:

- What is the name of your favorite college professor or high school teacher?
- Where would you go for your ultimate dream vacation? (city, country)
- What is the title of your favorite song and who is the artist?
- What is the title of your favorite book and who is its author?
- What is the name of your favorite work of art, who is the artist, and where did you see it?

However, in these examples, cultural bias could make it more likely for users in the same population to have identical answers to these questions, even if they do not deliberately share the answers. This bias potentially increases the risk of an insider attack.

Avoid creating questions that:

- Return simple answers, such as "What is your favorite color?"
- Request information likely to be known, or change, such as "What is your address?"

Single Sign-on allows your users to change answers to their security questions at any time without intervention of an administrator.

If your environment includes security questions or account self-service features, users who register security questions and answers can use the plug-in software to provide new answers to their available security questions.

After users successfully provide their answers and receive confirmation that the new answers are saved to the central store, their old answers are no longer valid.

Users change their answers to their security questions by accessing the Security Questions Registration wizard.

You provide users access to Security Questions Registration wizard as a published application:

1. Install the Single Sign-on Plug-in on a XenApp server.
2. Locate the QBAEnroll.exe file on the XenApp sever.
3. Publish the QBAEnroll.exe file and make it available to users.
4. Inform users how to access and use the Security Questions Registration wizard.

Note: Users running Single Sign-on Plug-in Version 4.8 can access the Security Question Registration wizard by selecting Tools > Security Questions Registration in the Logon Manager. These users do not require access to the Security Questions Registration wizard as a published application. Users running Single Sign-on Plug-in Version 4.6 Service Pack 1 or earlier cannot access the Security Questions Registration wizard as a published application.

Managing Your Questions

Mar 25, 2011

The Question-Based Authentication node in the Single Sign-on component of the Citrix AppCenter provides you with a central location for managing all security questions associated with identity verification, self-service password reset, and account unlock. You can add your own security questions to the list of default questions and create question groups and target them to specific users.

- If you edit the existing default questions after users register their answers, consider the meaning of the edited questions. Editing a question does not force a user reenrollment; but if you change the meaning of a question, users who answered that question originally might not be able to provide the correct answer.
- Adding, deleting, and replacing security questions after users are enrolled means that all users who were previously enrolled using an older set of questions cannot authenticate and reset their password until they reenroll. Users must answer the new set of questions when they open the plug-in software.
- Individual security questions can belong to multiple security question groups. When you create your security question groups, all questions you create are available for use with any security question group.

Use these steps to access the settings referenced in the following procedures:

1. Click Start > All Programs > Citrix > Management Consoles > Citrix AppCenter.
2. Expand the Single Sign-on node, expand Identity Verification, and select the Question-Based Authentication node.
3. From the Action menu, click Manage Questions.

You can create many different questions and designate a language for each question. You can also provide multiple translations of a single question. The plug-in software presents the user with the questionnaire in the language that corresponds to the language settings of the user's profile. If the language is not available, Single Sign-on displays the questions in the default language.

Note: When you specify a language for a security question, the question appears to users whose operating system settings are configured for that designated language. If the selected operating system settings do not match any of the questions available, users are shown your selected default language.

1. Select Security Questions.
2. From the Language drop-down list, select a language and click Add Question. The Security Question dialog box appears.
3. Create the new question In the Security Question dialog box.

Important: You must use the Edit command to include the translated text of existing questions. If you select Add Question, you are creating a new question that is not associated with the original.

In most instances, users see security questions displayed in the language associated with their current user profile. If the language is not available, Single Sign-on displays the questions in the default language that you specify.

1. Select Question-Based Authentication.
2. From the Default Language drop-down list, select the default language.

Note: The Perform backward compatibility option in this dialog box ensures that Password Manager 4.0 and Password Manager 4.1 plug-in software can continue to display identity verification questions.

Adding, deleting, and replacing security questions after users are enrolled means that all users who were previously enrolled using an older set of questions cannot authenticate and reset their password until they reenroll. Users must answer the new set of questions when they open the plug-in software. Editing a question does not force a user reenrollment; but if you change the meaning of a question, users who answered that question originally may not be able to provide the correct answer.

Important: If you are editing an existing question, be careful not to change the meaning of a question. This might cause a mismatch in user answers during reauthentication. That is, a user might provide a different answer that might not match the stored answer.

1. Select Security Questions.
2. Select a language from the Language drop-down box.
3. Select the question and click Edit. The Security Question dialog box appears.
4. Edit the question in the Security Question dialog box.

You can create a number of security questions that your users answer to confirm their identities. Each question you add to the questionnaire must be answered by your users. However, you can also group these questions together in a security question group.

For example, putting your questions in a group enables you to add a group of six questions to your questionnaire, and allows your users to choose from that group of questions, answering, for example, three of the six. This gives your users flexibility in selecting questions and providing answers to be used for identity verification.

1. Select Security Questions.
2. Click Add Group.
3. In the Security Question Group dialog box, name the group, select the questions, and set the number of questions the user must answer.

1. Select Security Questions.
2. Select the security group you want to edit and click Edit. The Security Question Group dialog box appears, with a list of security questions available to be part of the group. The questions currently in the group are indicated by a check mark. Here you can edit the name of the group, add questions to the group, and select the number of questions from this group a user must answer.

You must select one or two of the questions your users answer to encrypt the data for key recovery. Your users need to provide answers for all of the questions they originally answered when enrolling, but the questions you select are used to provide data to include as part of the encryption and key recovery process.

1. Select Key Recovery.
2. Select the check box next to each question or question groups to use for key recovery during identity verification.
3. Click OK to save your question and settings. A message might appear asking if you want to force users to reenroll answers. Click Yes to force reenrollment.

Security answer masking is available with Password Manager Versions 4.6 and 4.6 with Service Pack 1 and Single Sign-on 4.8 and 5.0.

Security answer masking provides an added level of security for your users when they register their security question answers or provide their answers during identity verification. When this feature is enabled, the answers of users running Password Manager 4.6, Password Manager 4.6 with Service Pack 1, Single Sign-on 4.8 or Single Sign-on 5.0 are hidden. During the answer registration process, these users will be asked to type their answers twice to avoid typing and spelling errors. Users will need to type their answers only once during identity validation because they are prompted to retry if there is an error.

Note: Security question answers registered with Password Manager 4.5 agent software can be masked when your software is upgraded to Single Sign-on Version 5.0. Security question answers for users with agent software for Password Manager 4.5, 4.1, or 4.0 remain visible regardless of the console setting.

1. Select Security Answer Masking.
2. Select Mask answers for security questions.

Backward compatibility mode enables the plug-in software to continue prompting users with identity verification questions you used for Password Manager Versions 4.0 and 4.1. Backward compatibility mode also allows you to continue using the default question, "What is your identity verification phrase?" If you are upgrading from Version 4.1, the identity verification questions and the questions you used for self-service password reset appear as a questionnaire in the Manage Questions dialog box.

Important: When creating and editing user configurations, do not enable backward compatibility if you have a new installation of Single Sign-on because that limits plug-in software functionality to Versions 4.0 and 4.1 of the product. Conversely, do not disable backward compatibility mode if agent software from Version 4.0 or 4.1 is running because that prevents users from performing key recovery and self-service password reset registrations.
If you are using automatic key management, do not enable backward compatibility. Automatic key recovery does not require users to answer identity verification questions.

For Versions 4.0 and 4.1 backward compatibility, the questionnaire must include at least one security question associated with the Account Self-Service password reset feature.

Each security question must include the following settings:

- Case sensitivity disabled.
- Minimum answer length set to one.
- Questions cannot be enabled for key recovery.

You can check for backward compatibility if you are upgrading from a previous version of Single Sign-on/Password Manager:

1. Select Question-Based Authentication.
2. Select Perform backwards compatibility check and click OK.

Single Sign-on performs the backward compatibility check and displays any errors in a dialog box.

Allowing Users to Manage Their Primary Credentials with Account Self-Service

May 09, 2015

You can configure the self-service features of Single Sign-on to allow your users to reset their primary password or unlock their Windows domain accounts without intervention by administrative or help desk staff. Depending on your needs, you can implement the self-service password reset and account unlock features securely in your Single Sign-on environment.

Note: To implement Account Self-Service with Citrix Web Interface, see

— Web Interface

.

The Self-Service Module features are protected by question-based authentication, which ensures that your users are authorized to reset their passwords or unlock their accounts. During the first-time use of the Single Sign-on Plug-in software or first-time use after the Account Self-Service function is configured, users must register answers to security questions you create and select during Single Sign-on setup.

These security questions are then presented to users when they need to reset their password or unlock their account. When the questions are answered correctly, users are allowed to reset their password or unlock their account, avoiding the need to call the help desk or administrator.

Important: The self-service password reset and account unlock features require that you implement question-based authentication. Users must register answers to security questions to use these features. If you choose not to use question-based authentication in your Single Sign-on environment, the self-service password reset and account unlock features are not available to your users.

Factors to consider:

- You can implement the features of the Self-Service Module to allow your users to reset their primary (domain account) password or unlock their Windows domain accounts in an Active Directory environment only.
- When users change their application password by using the Single Sign-on Plug-in software or primary password by using the CTRL+ALT+DEL key combination on a device in which the plug-in software is installed, Single Sign-on automatically captures the password change.
- To prevent user lockout, do not combine the self-service password reset with the Prompt user to enter the previous password option for confirming users' identities exclusively. When the previous password is the only method available to your users, users who forget their previous primary password are locked out of the system. Their user data must be reset or deleted from the central store and from all user devices on which it is stored, and they must reenter their credentials for all of their applications.

To use Account Self-Service functionality, perform the following steps:

1. Install the Self-Service Module and the Key Management Module.
2. Configure your question-based authentication.
3. Create a user configuration with one or both of the self-service password reset or account unlock features enabled.
4. Install and configure the plug-in software.

Combining automatic key management with self-service provides greater ease-of-use to users needing access to password-protected applications handled by the Single Sign-on Plug-in software. For example, if users reset their primary passwords, they do not need to answer security questions after successfully resetting their passwords. (However, they do need to answer security questions during the self-service password reset process.)

With automatic key management, users do not have to verify their identities after unlocking their accounts or resetting their domain passwords.

If users are locked out of their Windows account and cannot remember the answers to their security questions, you must use the Single Sign-on component of the Citrix AppCenter to reset self-service registration for users. After you reset users, the Self-Service Registration wizard appears the next time the users open the plug-on software. Your users can then register answers to their security questions.

1. Click Start > All Programs > Citrix > Management Consoles > Citrix AppCenter.
2. Expand the Single Sign-on node, expand the Identity Verification node and select Question Based Authentication.
3. In the Action menu, click Other Tasks > Revoke security question registration for a user.
4. In the Select User dialog box, type the user or user group name.

After the service and plug-in software are installed and configured, the Self-Service Module modifies the user's Windows logon dialog box and the Unlock Computer dialog box, or the Welcome screen for Windows Vista. Windows 7, Windows Server 2008, and Windows Server 2008 R2 users, (available when users lock their computers with the CTRL-ALT-DELETE key combination) by including an Account Self-Service button.

Before users can access the self-service features, they must log on to their primary domain account and register answers to security features. After successfully enrolling, they can use the self-service password reset and account unlock features.

With automatic key management, users do not have to verify their identities after unlocking their accounts or resetting their domain passwords.

Account Self-Service

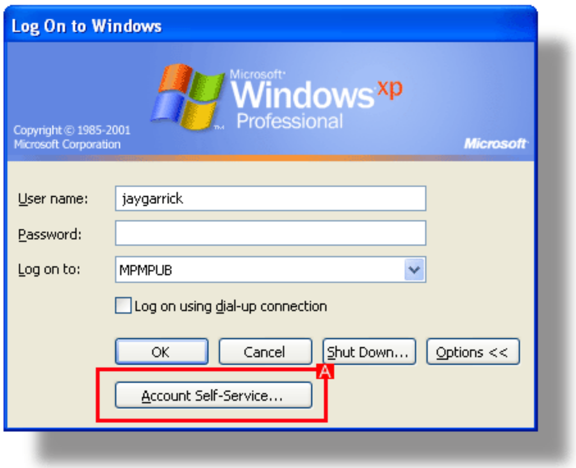
May 09, 2015

Single Sign-on customers have the option of deploying the Account Self-Service features—Self-Service Password Reset and Account Unlock—with no other Single Sign-on features available to users.

The Account Self-Service features of Single Sign-on help reduce calls to your computer help desk by allowing your employees to perform the following tasks on their own:

- Change their Microsoft Windows domain password
- Unlock their Windows domain account

The Account Self-Service features allow you to establish a set of security questions for identity verification. After the question-based authentication is enabled and the Account Self-Service features are made available to them, your users enroll, or register, with the service by answering the series of security questions. Once registered, your users can click Account Self-Service (A), found on the Log On to Windows dialog box, or for Microsoft Windows Vista users, the Welcome screen (B).

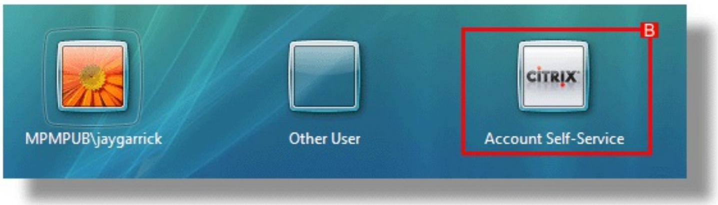


User name: jaygarrick

Password:

Log on to: MPMPUB

☐ Log on using dial-up connection



Administrators can require users to re-register by:

- Revoking a single user's question data
- Prompting all users to re-register
- Changing the existing questionnaire

Enrolled users can also start the re-registration process whenever they want to change their answers to the security questions.

This document describes how to install and configure Single Sign-on to provide users with only the Account Self-Service features.

Note: Account Self-Service does not support user principal name (UPN) logons, such as username@domain.com.

A Single Sign-on license is consumed during the re-enrollment process when users submit new responses for question-based authentication. Using concurrent user licenses ensures maximum license availability within your organization. A concurrent user license is returned to the license pool after the user completes the re-enrollment process. A named user license in the same situation remains with the user, even though it is not in use, for a minimum of two days.

Ratios are used to provide a greater number of Account Self-Service only licenses per Single Sign-on license. Concurrent user licenses use a 10:1 ratio, where 100 concurrent user licenses translates to 1,000 Account Self-Service licenses. Named user license use a 5:1 ratio, with 100 licenses translating to 500 Account Self-Service licenses.

To allow concurrent user licenses to be used offline

1. Create a user configuration.
2. On the Configure Licensing page of the User Configuration Wizard, select Concurrent User Licensing (Enterprise and Platinum Edition Only).
3. Select Allow license to be consumed for offline use and set the amount of time the license can be checked out from the license server.
4. Finish setting the user configuration.

For users associated with this user configuration, the license model is the same as a named user license—it can be consumed by users who might occasionally work remotely and be offline for periods of time. Concurrent user licenses are then consumed on a per-user basis.

Important: Locally installed instances of the Single Sign-on Plug-in do not require a separate license for users who have access to hosted applications in a Citrix XenApp, Platinum Edition environment.

Use the following steps to create a user configuration that allows Account Self-Service functionality without enabling Single Sign-on capability.

Note: Application definitions are not included in this user configuration because the feature does not include Single Sign-on functionality. If users need full Single Sign-on functionality, place them in a user configuration that does not include the Account Self-Service-Only modifications.

1. Click Start > All Programs > Citrix > Management Consoles and select the Citrix AppCenter.
2. To start the wizard, expand the Single Sign-on node and click User Configurations. In the Actions area, click Add new user configuration to open the User Configuration Wizard.
3. On the Name user configuration page:
    1. In the Name field, type the user configuration name.
    2. In the User configuration association area, choose how the user configuration is associated to the users by identifying the Active Directory hierarchy (organizational unit or user) or Active Directory group.
4. On the Select product edition page, select Single Sign-on Enterprise.
5. On the Choose applications page, click Next.
6. On the Configure plug-in interaction page, clear the following check boxes:
    - Automatically detect applications and prompt user to store credentials
    - Automatically process defined forms when Single Sign-on Plug-in detects them

    Click Advanced Settings.
7. In Advanced Single Sign-on Plug-in Settings:

- Select Application Support and clear the Detect client-side application definitions check box.

  Click OK to close Advanced Settings, and click Next.

8. On the Configure licensing page, in the License server address area, type the name of your license server and its port number.
   In the Licensing Model area, select Named User Licensing or Concurrent User Licensing.

   Note: Using concurrent user licenses ensures maximum license availability within your organization. A concurrent user license is returned to the license pool when the user completes the re-enrollment process. A named user license in the same situation remains with the user, even though it is not in use, for a minimum of two days.
9. On the Select data protection methods page, provide information as needed.
10. On the Select secondary data protection page, select Prompt user to select the method: previous password or security questions.
11. On the Enable self-service features page, select one or both of the following options:
    - Allow users to reset their primary domain password
    - Allow users to unlock their domain account
12. On the Locate service modules > Key Management Module page, provide the service address.
13. Finish the wizard without additional changes.

Note: Consider automating the following procedures by using scripts to help increase efficiency and improve accuracy. After the Single Sign-on Plug-in software is installed on the users' computers, you must modify the ssoShell.exe shortcut and the Start menu to provide user access to only the Account Self-Service features.

During the basic installation of the Single Sign-on Plug-in software, the ssoShell.exe shortcut contains the following command-line switch:

```
/background
```

Change this switch to:

```
/qbaenroll /noforceqbaenroll
```

Making this change causes the Single Sign-on Plug-in software on the user's computer, upon user logon, to synchronize with the central store and determine the status of the user's question-based authentication registration. If the registration process is complete and current, the user is not prompted to register. The user is prompted to register if one of the following conditions is discovered during synchronization:

- The user did not complete the question-based authentication registration process
- The administrator reset the user's question-based authentication questions
- The administrator modified the question-based authentication questionnaire

After completing the synchronization and, if necessary, starting the registration process, ssoShell exits automatically.

To update the Single Sign-on ssoShell.exe shortcut

For a desktop installation:

1. Windows Vista computers: Using Windows Explorer, navigate to %SystemDrive%\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup.
   Non-Windows Vista computers: Using Windows Explorer, navigate to %SystemDrive%\Documents and Settings\All

Users\Start Menu\Programs\Startup.

2. From the Startup folder, select Single Sign-on Background Process and select File > Properties.
3. In the Single Sign-on Background Process Properties dialog box, click in the Target field, scroll to the end of the text in that field, and delete /background.
4. In the Target field, following the remaining text, type /qbaenroll /noforceqbaenroll.

For a server installation:
Caution: Editing the Registry incorrectly can cause serious problems that may require you to reinstall your operating system. Citrix cannot guarantee that problems resulting from the incorrect use of Registry Editor can be solved. Use Registry Editor at your own risk. Be sure to back up the registry before you edit it.

1. Open the registry and navigate to HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\windows NT\CurrentVersion\Winlogon\AppSetup.
2. In this subkey, double-click the default entry to open the Edit String dialog box.
3. In the Value Data field:
   change: %SystemDrive%\Citrix\Metaframe Password Manager\WTS\SSOlauncher.exe /no ssoshutdown

   to: %SystemDrive%\Citrix\Metaframe Password Manager\ssoshell.exe /qbaenroll /noforceqbaenroll.

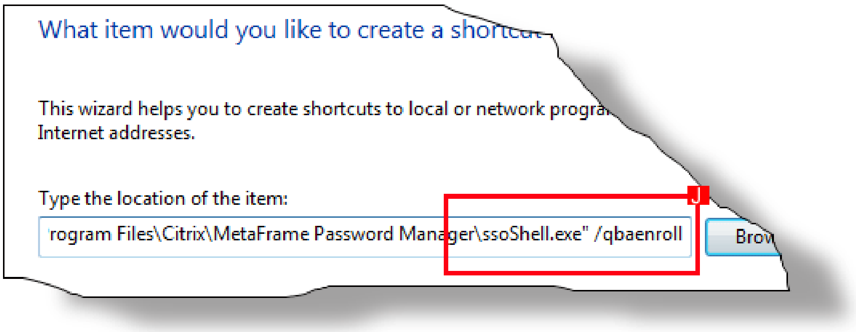
The ssoShell.exe file is modified for Account Self-Service functionality only.

To add a self-service registration shortcut to the Start menu

Add a shortcut to the Start menu to allow users to start the enrollment process on their own. This helps eliminate service calls if users do not provide answers during their initial logon or want to change answers they provided earlier.

1. Windows Vista computers: Using Windows Explorer, navigate to %SystemDrive%\ProgramData\Microsoft\Windows\Start Menu\Programs\Citrix\.
   Non-Windows Vista computers: Using Windows Explorer, navigate to %SystemDrive%\Documents and Settings\All Users\Start Menu\Programs\Citrix\.

2. From the File menu, select New > Shortcut. The Create Shortcut wizard appears.
3. Click Browse.
4. Navigate to %InstallationDirectory%\Program Files\Citrix\Metaframe Password Manager\, select ssoShell.exe, and click OK. The Browse for Folder dialog box closes and the path to ssoShell.exe appears in the Type the location of the item field.
5. In the Type the location of the item field, place the insertion point after ssoShell.exe and type a space followed by /qbaenroll (J).



What item would you like to create a shortcut

This wizard helps you to create shortcuts to local or network progra
Internet addresses.

Type the location of the item:

rogram Files\Citrix\MetaFrame Password Manager\ssoShell.exe" /qbaenroll

6. Click Next.
7. Type Citrix Account Self-Service Registration and click Finish.

The shortcut appears in Start > All Programs > Citrix.

To remove the Single Sign-on shortcut

During installation of the Single Sign-on Plug-in software, a shortcut is placed in the Start menu. If a user configured to use only the Account Self-Service features selects this command, ssoShell.exe launches and, unless there are changes to the user's Question-Based Authentication, exits. This may result in confusion to the user and cause support calls. To avoid this, remove the shortcut from the Start menu.

1. Windows Vista computers: Using Windows Explorer, navigate to %SystemDrive%\ProgramData\Microsoft\Windows\Start Menu\Programs\Citrix\.
   Non-Windows Vista computers: Using Windows Explorer, navigate to %SystemDrive%\Documents and Settings\All Users\Start Menu\Programs\Citrix\.

2. Delete the Single Sign-on shortcut.

The Single Sign-on shortcut is removed from the Start menu.

To remove the Single Sign-on Plug-in shortcut from the Startup folder

Remove the Single Sign-on Plug-in shortcut on the user device to prevent the plug-in software from starting each time the user logs on to the computer. This task prevents the user from unnecessarily consuming a license.

1. Using Windows Explorer, navigate to %SystemDrive%\Documents and Settings\All Users\Start Menu\Programs\Startup.
2. From the Startup folder, delete Single Sign-on Plug-in Background Process.
   Note: If the plug-in software is installed on Citrix Presentation Server or a terminal server environment, the AppSetup registry subkey, located in HKLM\SOFTWARE\microsoft\Windows NT\CurrentVersion\Winlogon\AppSetup must be edited to remove the reference to Password Manager or Single Sign-on.

The Single Sign-on Plug-in shortcut no longer starts automatically upon user logon.

Using Provisioning to Automate Credential Entry

May 09, 2015

Use the Provisioning Module (also known as credential provisioning) to manipulate user credentials associated with applications defined in a user configuration. Provisioning enables you to automate these procedures and apply them to multiple users. If you plan to deploy new software to your users, create an application definition for the application and use credential provisioning to add the credentials for all users who will use the application.

To manipulate credential information in your central store for SSO-enabled applications contained in user configurations, you must perform the following tasks:

1. Install the Provisioning Module of the Single Sign-on Service.
2. Create a user configuration that uses the provisioning service.
3. Generate a credential provisioning template.
4. Populate the template with user credential data and select a command to run.
5. Process your provisioning data.

Important: The XML file you use to provision credentials contains highly sensitive user-related information. Consider deleting the file or moving it to a secure location when credential provisioning is completed.
After the credentials are added, removed, or modified in the central store, the credentials are ready for use in your environment. When users start the plug-in software, the credentials updated in the plug-in software and applications are made available to your users.

Adding, changing, or removing credentials from the central store can consume a large amount of system resources. When possible, perform credential provisioning during off-peak hours.

The Credential Provisioning SDK

If you need to manipulate the credentials of many users, consider using the Credential Provisioning Software Development Kit (SDK). The SDK provides a description of the APIs made available when you install the Provisioning Module of the Single Sign-on Service. Use this SDK and included sample code to create your own provisioning client for use with Single Sign-on.

The following procedure assumes that you created a user configuration consisting of at least one of the following: application definition, application group, password policy (perhaps including an optional password sharing group) and provisioning is enabled in the user configuration.

A provisioning template is an XML document that contains information about the applications included in your selected user configuration:

- Application group
- Application definition name and globally unique identifier number (GUID)
- User information like user name and password

It also includes add, remove, and modify commands that you use when you use the edited template from the console to run provisioning.

The resulting template includes example command information and specific information about the selected user configuration.

To generate a credential provisioning template

1. Click Start > All Programs > Citrix > Management Consoles > Citrix AppCenter.
2. Expand the Single Sign-on node and select User Configurations.
3. Select a user configuration.
4. From the Action menu, click Generate Provisioning Template.
5. In the Generate Provisioning Template dialog box, type a name for the template.

Use the Single Sign-on component of the Citrix AppCenter to perform the provisioning tasks specified in your XML file. Single Sign-on validates the syntax of each command, executes the commands, and adds or modifies the data in the central store.

Caution: Do not close the provisioning process screen until provisioning has fully stopped or fully completed. Closing this screen does not halt the provisioning process. If the screen is closed while the previsioning process is running, there is no way to capture any information or halt the process until it completes.

1. Click Start > All Programs > Citrix > Management Consoles > Citrix AppCenter.
2. Expand the Single Sign-on node and expand User Configurations.
3. Select a user configuration or application group of a user configuration.
4. From the Action menu, click Run provisioning. The Provisioning Wizard appears.
5. Click Next.
6. Type the name of your provisioning XML file or click Browse to locate it, then click Next. Single Sign-on validates the XML file.
   - If no syntax errors are found, a summary of the changes that can be made is shown. You can save the summary.
   - If syntax or other errors are found, an error log appears. You can save the error log and click Cancel to close the wizard.
7. If no errors were found, click Next to execute the commands in the file. As the information is changed in the central store, any errors that occur as a result of provisioning appear. To stop provisioning while it is in process, click Abort. When Single Sign-on reaches the end of the current section of data in process (by default, data is processed in groups of 50 lines of code), provisioning terminates.

When you finish the wizard, you can save the provisioning results.

Caution: This procedure requires you to edit the registry. Using Registry Editor incorrectly can cause serious problems that might require you to reinstall your operating system. Citrix cannot guarantee that problems resulting from the incorrect use of Registry Editor can be solved. Use Registry Editor at your own risk. Always back up a copy of your system registry before continuing

By default, if you use Single Sign-on for credential provisioning, your information is processed in batches of 50 commands with a time-out of 100,000 milliseconds. The following registry keys can be edited to change these default values:

HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\MetaFrame Password Manager\Console\Provisioning\BatchSize

Type: DWORD

```
Default value if left blank: 50

HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\MetaFrame Password Manager\Console\Provisioning\ServiceTimeout

Type: DWORD

Default value in milliseconds if left blank: 100000
```

Editing the Provisioning Template

May 09, 2015

Use a text editor or XML file editor to edit the generated template. The provisioning template uses Service Provisioning Markup Languagex (SPML), an XML-based standard for data interchange. As with XML, ensure that each SPML tag or element (for example, the <add> tag) is well-formed and conforms to XML syntax rules. For example, when removing comment characters such as !-- and --, ensure that you remove any extraneous angle bracket characters (< or >) or errors might result during processing of the provisioning template. For detailed information about XML, see the W3C Web site at http://www.w3.org/. Ensure that you remove comment characters (!-- and --) where applicable.

The generated template includes the following:

- <user> information about the user who generated the template
- <add> command for the application name in the user configuration
- <modify> command with the application definition name

Near the bottom of the XML file is the specific information about the selected user configuration that you can copy and use in your template. For example:

```
<user fqdn="DOMAIN\Fred-Admin">
<!--Application Group: PNA-->
<!--Application Definition: Citrix GoToMeeting-->

<!--<add>
 <application name="Citrix GoToMeeting">0998ac2c-baa5-4103-809a-b2daeea047f3</application>
 <name>Citrix GoToMeeting</name>
 <description>Citrix GoToMeeting Login</description>
 <hidden-description>Citrix GoToMeeting hidden Description</hidden-description>
 <userID>userId</userID>
 <password>password</password>
</add>-->

<!--<modify>
 <credential-id>00000000-0000-0000-0000-000000000000</credential-id>
 <name>Citrix GoToMeeting</name>
 <description>Citrix GoToMeeting Login</description>
 <hidden-description>Citrix GoToMeeting hidden Description</hidden-description>
 <userID>userId</userID>
 <password>password</password>
</modify>-->

</user>
```

For example, you can copy the user information between the <user> and </user> tags, uncomment it, and edit it for each user whose credentials you wish to add.

Note: In the example above, <user fqdn="DOMAIN\Fred-Admin"> is the domain and user name of the user who generated the template. You can comment out this information or delete it if you do not want to store it in the template.

Note that you must include your desired tags and commands within the <cpm-provision> provisioning tag (located around line 70 in the generated XML file):

```
<cpm-provision version="1.0" xmlns="http://citrix.com/Provision/Import">

insert <user> tag and commands here

</cpm-provision>
```

Use the <user> tag to add domain and user name information for each user whose application credentials you wish to provision. You must provide one <user> tag for each user to be provisioned. Each <user> tag will also contain the commands to execute on this user account.

The syntax for this command is as follows.

```
<user fqdn="yourDomain\usrid">
 <command>
</user>
```

where:

yourDomain	Indicates the domain name of the user to be added
userid	Indicates the username of the user to be added
command	Indicates one or more commands that you can execute on this user: ● <add> ● <modify> ● <delete> ● <remove> ● <reset> ● <list-credentials>

The <add> command enables you to add a user name and password required for the applications included in the user configuration.

The syntax for this command is as follows.

```
<add>
```

```
 <application name="%APPNAME%">%APPGUID%</application>
 <name> <description>longDescription</description>%CREDENTIALNAME%</name>
 <description>longDescription</description>
 <hidden-description>%APPNAME% hidden description</hidden-description>
 <userID>userid</userID>
 <password>password</password>
 <custom-field index="1" label="%LABELTEXT%">custom-field1 </custom-field>
 <custom-field index="2" label="%LABELTEXT%">custom-field2 </custom-field>
</add>
```

where:

<application>	Required. The <application> element and its attributes are typically generated automatically when you generate a template. The name= attribute is optional. ● %APPNAME% is the name of the application definition in the selected user configuration. ● %APPGUID% is the GUID of the application and must match
<name>	Required. The <name> element and its attributes are typically generated automatically. ● %CREDENTIALNAME% is the name of the application in the application definition.
<description>	Optional. Type text that describes the user configuration.
<hidden-description>	Optional. Type any text here.
<userID>	Required. userid is the user name of the user to be added.
<password>	Required. password is the password for the user to be added.
<custom-field>	Required if another field is required for authentication (for example, for a field where the user must enter the domain). Use as many custom fields as required by the application.

The <modify> command enables you to modify a user name and password required for the applications included in the user configuration.

Important: This command requires the user's credentials. You can retrieve user credentials by using the <list-credentials> command before using the <modify> command.
Include only those elements you want to modify:

- To leave a value unchanged, delete the line. For example, delete the <name> element to leave the application name as is.

- To change a value, specify the value in the template. For example, include the <name> element to specify a new application name.
- A value is cleared by including the element without a value. For example, use <description></description> to delete the current description.

The syntax for this command is as follows.

```
<modify>
 <credential-id>%CREDENTIAL-ID%</credential-id>
 <name>%CREDENTIALNAME%</name>
 <description>longDescription</description>
 <hidden-description>%APPNAME% hidden description</hidden-description>
 <userID>userid</userID>
 <password>password</password>
 <custom-field index="1" label="%LABELTEXT%">
 custom-field1 </custom-field>
 <custom-field index="2" label="%LABELTEXT%">custom-field2 </custom-field>
</modify>
```

where:

<credential-id>	Required. The credential GUID value %CREDENTIAL-ID% of the user must match the value returned by a <list-credentials> command.
<name>	Optional. The <name> element and its attributes are typically generated automatically. ● %CREDENTIALNAME% is the name of the application in the application definition.
<description>	Optional. Type text that describes the user configuration.
<hidden-description>	Optional. Type any text here.
<userID>	Required. userid is the user name of the user to be modified.
<password>	Required. password is the password for the user to be modified.
<custom-field>	Required if another field is required for authentication (for example, for a field where the user must enter the domain). Use as many custom fields as required by the application.

The <delete> command enables you to delete a user's credentials for a specific SSO-enabled application.

Important: This command requires the user's credentials. You can retrieve user credentials by using the <list-credentials> command before using the <delete> command.

The syntax for this command is as follows.

```
<user fqdn="yourDomain\userid">
 <delete>
  <credential-id>%CREDENTIAL-ID%</credential-id>
 </delete>
</user>
```

where:

yourDomain	Indicates the domain name of the user.
userid	Indicates the user name of the user.
<credential-id>	Required. The credential GUID value %CREDENTIAL-ID% of the user must match the value returned by a <list-credentials> command.

The <remove> command enables you to remove user data and information from the central store. Use this command when a user leaves your enterprise permanently. The local credential store on the user device remains intact until it is explicitly deleted by an administrator or operator.

The syntax for this command is as follows.

```
<user fqdn="yourDomain\userid">
 <remove />
</user>
```

where:

yourDomain	Indicates the domain name of the user.
userid	Indicates the user name of the user.

Note: This command is similar to the Single Sign-on Delete user data from central store task carried out from the Citrix AppCenter.

The <reset> command enables you to reset user information in your central store, which results in the selected user being returned to an initial state. In the case of non-Active Directory central stores, the user folders are retained, but all user data (such as credentials, enrollment questions and answers) is deleted. In Active Directory central stores, the user data is deleted and the user is flagged as having had data reset.

The syntax for this command is as follows.

```
<user fqdn="yourDomain\userid">
```

```
 <reset />
</user>
```

where:

yourDomain	Indicates the domain name of the user.
userid	Indicates the user name of the user.

Note: his command is similar to the Single Sign-on Reset user data task carried out from the Citrix AppCenter.

The <list-credentials> command enables you to retrieve a specific user's credentials for each application in the associated user configuration. The <modify> and <delete> commands require that you use the retrieved credential GUID as the value for the %CREDENTIAL-ID% parameter.

The identification number that this command retrieves is a credential GUID; for example, 634EE015-10C2-4ed2-80F5-75CCA9AA5C11.

The syntax for this command is as follows.

```
<user fqdn="yourDomain\userid">
 <list-credentials />
</user>
```

where:

yourDomain	Indicates the domain name of the user.
userid	Indicates the user name of the user.

Hot Desktop: A Shared Desktop Environment for Users

May 11, 2015

Hot Desktop combines the convenience of fast user switching with security of Single Sign-on capability through Single Sign-on. Hot Desktop functionality is not installed by default; you can select it during the initial Single Sign-on Plug-in installation process. You can also upgrade existing Single Sign-on Plug-in deployments to use Hot Desktop. Before you can implement Hot Desktop, however, you must configure Hot Desktop according to requirements in your environment and enterprise.

Hot Desktop is supported only on:

- Microsoft Windows XP Professional, Service Pack 2—32-bit
- Microsoft Windows XP Embedded

Hot Desktop is not supported on 64-bit operating systems or any server operating systems.

Hot Desktop is not available when Single Sign-on is deployed through Citrix Receiver Updater.

The Single Sign-on Hot Desktop feature allows users to share workstations efficiently and securely. Hot Desktop extends the standard Windows environment by allowing a user to:

- Quickly authenticate to Windows using the standard GINA interactive logon dialog box
- Run SSO-enabled applications in the interactive user shell by using the user's Single Sign-on credentials
- Log off from the Hot Desktop workstation so that other users can run applications

Before you can implement Hot Desktop you must:

- Create a Hot Desktop shared account
- Create user configurations with specific Hot Desktop-related settings to control the Hot Desktop user experience
- Define Hot Desktop startup and shutdown behavior, including:
  - Deciding which applications are launched at startup, and which applications use Hot Desktop User or Hot Desktop shared account credentials and permissions
  - Deciding which applications are persistent and run even when users log off (for fast user switching) and which applications terminate when users log off, including your optional cleanup scripts or applications to delete user information from session to session

Perform the following tasks to configure and enable Hot Desktop:

1. Create a Hot Desktop shared account that is available for each workstation or client device running Hot Desktop.
2. Decide which of your SSO-enabled applications should run in the Hot Desktop environment.
3. Decide how applications run in Hot Desktop and configure the Hot Desktop user environment.
4. Create or modify a user configuration to select Hot Desktop options.
5. Install the plug-in software with the Hot Desktop feature selected.
6. Uninstall Hot Desktop if necessary.

This process flow describes the events associated with Hot Desktop startup and shutdown. When the workstation or client device starts, it is logged on automatically to the shared account, allowing the device to run in shared desktop mode.

Note: The Hot Desktop shared account remains active at all times. Users do not have the permissions to terminate the shared account.

1. A Hot Desktop user logs on to the workstation and enters a user name and password or uses a strong authenticator such as a smart card.
2. When the user is authenticated, the Hot Desktop session starts.
3. Single Sign-on launches. The plug-in software synchronizes its data with the central store. This ensures that the user has the most current application definitions, password policies, and other plug-in software-related settings.
4. The session.xml file is read and any applications that you specified to run under the shared account or Hot Desktop User account launch. These applications can be local or remote applications that are published by XenApp. The user accesses the applications to perform job-related tasks.
5. The Hot Desktop user logs off.
   Note: When users leave a workstation idle, Hot Desktop initiates a session time-out period. Using the Access Management Console, you specify how long a workstation can remain inactive. When the interval is exceeded, Hot Desktop locks the workstation. If additional time passes and the user does not return, Hot Desktop terminates the session.
6. Hot Desktop leaves applications running or terminates them according to settings in process.xml.
7. Single Sign-on exits.
8. Any shutdown scripts specified in session.xml run.
9. The Hot Desktop session terminates.

When a user logs on to a computer running Single Sign-on configured for Hot Desktop, it is possible that the startup scripts specified in the session.xml file might run before Single Sign-on Plug-in software has fully launched.

During its startup, Hot Desktop waits 30 seconds for the plug-in software to start before it begins running the startup scripts. After 30 seconds, these startup scripts run, even if the plug-in software is not yet fully launched.

This situation is most likely to occur during the user's initial logon (also known as first-time user), where the Single Sign-on administrator identified a list of applications requiring logon credential registration or required answers for security questions. The sequence in this case is:

1. The user logs on to the computer or client device running the plug-in software and a prompt appears for the user to register logon credentials for the listed applications or register answers to security questions.
2. While performing these tasks, the 30 seconds pass and Hot Desktop startup scripts run. A series of windows might open and close, depending on the applications specified in the session.xml startup scripts.
3. User frustration might result as the computer keeps moving focus to the startup script windows.
4. When the startup scripts are completed, an error message appears. The error is similar to “One or more errors occurred. Please consult the Event log for more information.”

While this behavior might cause user frustration, it does not damage the user's data, work environment, or Single Sign-on.

Advise users not to register their logon credentials and security question answers until the error message appears. Users can then close the error message and complete the enrollment and registration process.

Following the error message and registration, if any application specified in session.xml does not open, advise the user to log off and then log back on to the account. This restarts any Hot Desktop startup scripts, which run uninterrupted because registration is complete and not delaying the process.

You must create a Hot Desktop shared account for the client devices or workstations on which Hot Desktop will run. This shared account can be a domain account or a local account on the device. When you install Hot Desktop on the client device, you provide credentials for the shared account. When the device or workstation starts, it is logged on automatically to the shared account, allowing it to run in the Hot Desktop shared workstation mode.

User sessions run “on top” of the shared account Windows session (users cannot make changes to the shared account unless you allow them to). Users start a Hot Desktop session by typing their Windows domain credentials. In a Hot Desktop environment, a user’s Windows account is referred to as the Hot Desktop User.

Organizing Hot Desktop Users

If you plan to deploy Hot Desktop, you might want to set up your user environment first. For example, you might group Hot Desktop users under one or more Active Directory organizational units or groups. Also, you can organize users who are Hot Desktop users and also use their own workstations into multiple groups (and prioritize these groups).

This enables you to apply Hot Desktop settings, application definitions, password policies, and other user configuration information to multiple Hot Desktop users in those organizational units.

Restricting User Rights

Because the Hot Desktop device is shared by all Hot Desktop users, it may be necessary to restrict permissions and set them to a minimum required to use their assigned applications. For example, Hot Desktop users should not have the right to shut down the device. Restrict this right to members of the Administrators group.

Hot Desktop, Smart Cards, and Key Recovery

Note: Select the Smart Card Certificate user configuration Data Protection option if users use smart cards in the Hot Desktop environment.
If you deploy Hot Desktop in an environment where users log on with smart cards, do not select Prompt user to enter the previous password as the only key recovery and data protection method for those users. Users in such an environment cannot enter the correct previous password and, consequently, are locked out of the system. To avoid this problem, select the key recovery option for automatic key management or make question-based authentication available as an option.

Follow these guidelines to create a shared account:

- Ensure that the account does not belong to the local or domain Administrators group.
- The shared account can be a local or domain account. Any privileges available to the shared account are available to the Hot Desktop User only for those applications you specify. That is, you can specify those applications that launch with Hot Desktop shared account credentials and those that launch with the user’s Windows domain credentials.
- The Hot Desktop installation process verifies the logon name and domain of the shared account. When you create this account, ensure that you select the Password never expires option. Do not use expired credentials.
- Ensure that the account has limited privileges. Limit permissions to Hot Desktop use only.
- Specify the domain name to which the workstation belongs using the domain’s NetBIOS name and not the fully qualified domain name (FQDN). If you are using a local account, specify the host name of the device.

- As a best practice, name the shared account “Hot Desktop.” This ensures that users see the message “Logoff Hot Desktop” when they log off. If you give the shared account a cryptic name, users see the name as they log off and might get confused. If you have more than one group of Hot Desktop Users, you can name each shared account accordingly; for example, “Hot Desktop Marketing,” “Hot Desktop Accounting,” and so on.

Applications that you use in a Hot Desktop environment must meet the following requirements:

- Applications that require user credentials must be defined for use with Single Sign-on in application definitions and user configurations.
- Applications that are launched by the shared account must be able to run in the Windows interactive environment. In this scenario, the applications (and the Hot Desktop users) must have access to the user profiles, network shares, and other resources associated with the shared account.
- Applications must shut down cleanly when sent the request to do so. Hot Desktop terminates applications using procedures similar to a log off from a Windows interactive session. Graceful application termination is particularly important in a Hot Desktop environment because the application might be used many times before the workstation or client device is shut down.
- Any application that must save sensitive data in the user’s profile or needs access to the user’s profile for settings should run as the Hot Desktop User account. Applications that can share “community” configuration information can run as a shared account. Administrators can use a session shutdown script specified in the session.xml file to ensure that user-specific files are removed at the end of each session.

Important: If you want Single Sign-on to submit credentials in a Hot Desktop environment for terminal emulators that store information in the HKEY_CURRENT_USER registry hive, you must run these applications as the Hot Desktop User account. Specify terminal emulators to run as the Hot Desktop User account in the ShellExecute section of the process.xml file. To run a terminal emulator at session start up, specify it in the start script section of the session.xml file. Terminal emulators must run as the Hot Desktop User account in the start script.

Single Sign-on makes two files available to control the behavior of applications in a Hot Desktop environment: session.xml and process.xml.

Important: You cannot specify that a process runs as the Hot Desktop shared account in the session.xml file and then specify it to run as the Hot Desktop User in the process.xml file. Entries in the session.xml file override any entries you make under the <shellexecute_processes> element in the process.xml file.
Before you begin:

- To log on to the workstation or user device for administrative purposes (for example, to edit the process.xml file), hold down the SHIFT key during the Windows startup process. For more information about bypassing the Windows autologon process, visit the Microsoft Web site.
- When running Hot Desktop session.xml, password expiration scripts, or any other scripts, executable files, or batch files from within a Hot Desktop User session, the following environment variables are not supported: APPDATA, HOMEDRIVE, HOMEPATH, HOMESHARE, and LOGONSERVER. If any of the unsupported variables are used, the script, application, or executable file might fail to run. To avoid this problem, applications should not access unsupported environment variables while running in a Hot Desktop User session.
- You must instruct users to shut down applications that are specified as persistent processes. For example, if a user launches a persistent process, creates a file, and leaves the file open when exiting the Hot Desktop session, the next user who logs on can see the contents of the file.

Important: Instruct users to always shut down sensitive applications that are defined as persistent before they end their Hot Desktop sessions.

When you define an application as persistent in process.xml and specify it in a start script in session.xml, the number of application instances might increase if users do not terminate new application instances during a Hot Desktop session. To prevent this from occurring, limit the number of instances by creating a script or wrapper application that launches the application. You can also modify the application itself to ensure that only one instance is running at any given time.

- Applications launched from a command prompt run as the Hot Desktop shared account even if they are specified as the Hot Desktop User account. To launch applications from a command prompt as the Hot Desktop User, you must specify the command prompt in the <shellexecute_processes> section of the process.xml file. Also, if the command prompt is running as the shared account and the file type association (such as *.txt) is defined in the process.xml file <shellexecute_processes> section, if the user runs a file with a .txt extension, the application launches as the Hot Desktop User.
- Persistent applications that use the 8.3 file format must use the 8.3 format in the path of the executable when specified in process.xml.
- While the XML tags and formatting in process.xml file are case-sensitive, the paths and executable names are not.
- If your users are running SAP Logon for Windows (saplogon.exe), it must run as the Hot Desktop User. In the process.xml file, specify saplogon.exe under the <shellexecute_processes> tag.

User Configuration Settings for Hot Desktop

May 11, 2015

You can control the Hot Desktop user experience through the following user configuration settings.

Caution: Some procedures require you to edit the registry. Using Registry Editor incorrectly can cause serious problems that might require you to reinstall your operating system. Citrix cannot guarantee that problems resulting from the incorrect use of Registry Editor can be solved. Use Registry Editor at your own risk. Always back up a copy of your system registry before continuing.

Locate Hot Desktop settings in a user configuration:

- When you create a new user configuration, these settings are available from the Advanced Settings of the Configure Plug-in Interaction dialog box.
- When you modify an existing user configuration, these settings are available from the Hot Desktop panel of the Edit User Configuration dialog box.
  For setting details, see the HotDesktop topic under
  — Single Sign-on Settings Reference > User Configurations
  .

To configure the session settings script path

1. In the Hot Desktop page of the Edit User Configuration dialog box, in the Session settings script path text field, type the location of the session.xml file. The location can be a network shared folder. For example, if you place your session.xml file on a network share such as \\Citrix\MPM\Share\, type that path here.
2. Restart the Hot Desktop workstation after you save the user configuration and install the session.xml file.

Interaction with Automatic Key Recovery

If your Single Sign-on environment combines the automatic key recovery feature with Hot Desktop, password changes performed by the administrator are not communicated to the plug-in software of affected users with active Hot Desktop sessions. If those users lock and then attempt to unlock their active sessions, they might be prompted unexpectedly to provide their previous passwords. Users should close the previous password dialog box, then terminate and restart the Hot Desktop session by logging off to continue using the plug-in software.

Hot Desktop Screen Saver

To make it easier for users to identify which workstations are running Hot Desktop, a custom screen saver is included in a Hot Desktop installation. The screen saver does not launch until the workstation is idle for 10 minutes.

Note: A locked session is considered active. The screen saver does not launch until 10 minutes of idle time passes and after all users are logged off from the workstation.

Hot Desktop can be installed on with a new or existing installation of Single Sign-on Plug-in.

1. Log on to the user device as a local administrator.
2. From the Control Panel, select Add or Remove Programs.

3. Select Single Sign-on Plug-in and click Change.
4. Select Modify and click Next.
5. Select Hot Desktop and click Next.
6. Click Yes to the confirmation message to disable Terminal Services and Remote Desktop.
7. Specify the location of the central store and click Next.
8. Specify the service server address and click Next.
9. Type the user credentials for the Hot Desktop shared account and click Next. Specify the domain name to which the workstation belongs using the domain's NetBIOS name, not the fully qualified domain name (FQDN).
10. Click Install. Access the installation media so that the install process can find Single Sign-on Plug-in .msi file

After you finish the installation, restart the user device.

If you need to remove the Hot Desktop feature from a workstation, you might also need to perform these procedures after uninstalling the Hot Desktop feature:

- Restore Terminal Services after Uninstalling Hot Desktop
- Enable Multiple Sessions after Uninstalling Hot Desktop

1. To log on to the shared workstation or client device to perform administrator tasks, hold down the SHIFT key during the Windows startup process.
   This prevents the Hot Desktop shared account from logging on and starting the Hot Desktop environment. For more information about bypassing the Windows autologon process, visit the Microsoft Web site.

   Log on as the administrator.

2. Open the Control Panel and select Add or Remove Programs.
3. Select Single Sign-on Plug-in.
4. Click Change to remove the Hot Desktop feature only.
5. On the Application Maintenance page, select Modify.
6. On the Feature Selection page, select Hot Desktop and make the feature unavailable.
7. Follow the prompts to select your central store type and to confirm the plug-in software changes.
8. Restart the workstation.

Hot Desktop is not removed completely until the workstation is restarted.
Important: When uninstalling software that may have disrupted the GINA chain, it is important to uninstall the software in the reverse order in which it was installed on the client device. Failure to uninstall in the reverse order in which GINA-altering software was installed can leave the computer in an invalid state. Do not edit the registry.

The Hot Desktop installation process disables Terminal Services. Perform the following steps to enable Terminal Services.

1. Log on to the workstation as an administrator.
2. Click Start > Run and type regedit.
3. Change the value of the registry key to 1 as follows:
   [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server]TSEnabled=dword:00000001

During a Hot Desktop installation, the installer resets this registry key value to zero. Perform the following procedure to enable multiple sessions.

1. Log on to the workstation as an administrator.
2. Click Start > Run and type regedit.
3. Change the value of the registry key to 1 as follows: [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\Current Version\Winlogon] AllowMultipleSessions =dword:00000001

In a Hot Desktop environment, the shell (explorer.exe) runs as the Hot Desktop shared account. Consequently, the shell does not have the access rights to navigate to the Hot Desktop User profile folder.

1. In the process.xml file, under <shellexecute_processes> section, include Internet Explorer (iexplore.exe) so that it runs as the Hot Desktop User.
2. Log on as the Hot Desktop User and launch Internet Explorer.
3. To view the profiles, in the address bar, type the full path to the Hot Desktop User profile directory. For example: C:\Documents and Settings\All Users\Application Data\Citrix\MetaFrame Password Manager

Some third-party authenticators might not work if the AutoAdminLogon feature is enabled. Some third-party applications disable or remove the AutoAdminLogon value during installation. If this is the case, you must disable Hot Desktop AutoAdminLogon.

1. Restart the shared workstation or user device while holding down the SHIFT key during the Windows start process. This prevents the Hot Desktop shared account from logging on and starting the Hot Desktop environment. For more information about bypassing the Windows autologon process, visit the Microsoft Web site.
2. Log on as an administrator.
3. Edit the registry and set the following values under HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\MetaFrame Password Manager\HotDesktop

Value Name	Type	Value
AutoAdminLogon	REG_SZ	0 to disable

4. After the value is set, restart the workstation and log on manually using the shared account. The Hot Desktop logon page appears, allowing users to use the third-party authenticator.

It might become necessary to change the Hot Desktop shared account password. You first entered the account credentials during the plug-in installation. To change the password, perform the following procedure.

1. Log on to a workstation where Hot Desktop is installed.
   Important: Do not use an administrator account or the Hot Desktop shared account credentials for Step 1.
2. Press the CTRL+ALT+DELETE key combination. The Windows Security dialog box appears.
3. Click Change Password.
4. Type or select the following:
   - Hot Desktop shared account user name
   - Domain name or local computer name

   - Old password
   - New password
5. Click OK.
6. From the Windows Security dialog box, click Shutdown, then Restart to restart the computer.

Because only administrators are allowed to shut down Hot Desktop workstations, the Shut Down option is not available from the Start menu of a Hot Desktop workstation.

To shut down a Hot Desktop workstation for administrative use, press CTRL+ALT+DEL. When the Windows Security dialog box appears, click Shut Down.

Single Sign-on supports the use of Citrix plug-ins with Hot Desktop. Use these general guidelines to consider if you plan to use Hot Desktop with these plug-ins and Web Interface:

- Edit the process.xml file to ensure Citrix Receiver and Citrix Offline Plug-in are transient processes (in case the plug-in is set to be launched by Windows startup program and is running after the first Hot Desktop session starts).
- If you are using the Security Service Provider Interface, you must run the plug-in as the Hot Desktop User. You may also run the plug-in as the Hot Desktop User if you are concerned about security; the ICA files are stored in the profile.
  - Edit the <shellexecute_processes> section of the process.xml file to ensure Citrix Receiver and Citrix Offline Plug-in run as the Hot Desktop User when launched from the Windows shell
  - Edit the session.xml file to specify a start script or executable to launch Citrix Receiver and Citrix Offline Plug-in when the first Hot Desktop session starts

Citrix Receiver

You can configure Citrix Receiver to use the Security Service Provider Interface. Security Service Provider Interface allows Receiver to authenticate to the XenApp server using the Hot Desktop User credentials. You must ensure that XenApp trusts the Windows security authority used to authenticate the Hot Desktop User. For more information about configuring the Security Service Provider Interface for Receiver, see the topics for

— XenApp Administration

.

Web Interface

The Hot Desktop plug-in can submit credentials through the Web Interface to a XenApp server. For more information, see

— Web Interface

topics about configuration.

The Session.xml File

Feb 08, 2011

Use the session.xml file to specify the applications that launch when a Hot Desktop session starts (start script) and remove files or other information left behind by a user session (shutdown script). After you edit this file as needed, put it on a network share or other central location for your Hot Desktop workstations to access. You specify this location of the session.xml file in the user configuration.

You must include your desired tags within the <session_settings> and </session_settings> tags in the file.

Note: A sample session.xml file is located in the \Support folder of the installation media.

Use a shutdown Visual Basic script to clean up any user data left behind at the end of a session. The session_cleanup.vbs script launches as the shared account (named HDSA) and is located in C:\.

```
<shutdown_scripts>
 <script>
  <account>HDSA</account>
  <working_directory>c:\</working_directory>
  <path>c:\session_cleanup.vbs</path>
 </script>
</shutdown_scripts>
```

Launch Internet Explorer with the URL of your mycompany.com intranet. In this case, Internet Explorer runs as a process associated with the Hot Desktop User.

Note that you would enclose your desired tags within the <session_settings> and </session_settings> tags in the file.

```
<startup_scripts>
 <script>
  <account>HDU</account>
  <working_directory>c:\program files\Internet Explorer</working_directory>
  <path>c:\program files\Internet Explorer\iexplore.exe http://www.mycompany.com</path>
 </script>
</startup_scripts>
```

This section of the file is used to specify any applications to launch under the Hot Desktop shared account and the Windows account associated with the Hot Desktop User.

```
<startup_scripts>
 <script>
  <account>account</account>
  <working_directory>wd</working_directory>
  <path>path_options</path>
 </script>
</startup_scripts>
```

where:

account	Indicates the account under which to run the application. Choices are HDU or the Hot Desktop shared account user name.
wd	Indicates the working directory of the application.
path_options	Indicates the fully qualified folder path to the application executable file or script on the local computer and any options to run with the application. For example: c:\program files\Internet Explorer\iexplore.exe http://www.yahoo.com

Edit the session.xml shutdown applications to remove all unused data from the previous user session. Typically, these applications should remove configuration files that might prevent the next user from working, sensitive files such as logs, and documents stored on the system. These applications should ensure that the Hot Desktop environment is clean for the next user session. This part of the file is especially useful for data security.

Note: If necessary, you can initiate administrator programs or scripts to clean up the user environment at logoff. For example, you can write a Visual Basic script using a third-party application to delete user-specific .ini files.

```
<shutdown_scripts>
 <script>
  <account>account</account>
  <working_directory>wd</working_directory>
  <path>path_options</path>
 </script>
</shutdown_scripts>
```

where:

account	Indicates the account under which to run the shutdown application. Choices are HDU and the Hot Desktop shared account user name.
wd	Indicates the working directory of the application.
path_options	Indicates the fully qualified folder path to the application executable file or script on the local computer and any options to run with the application. For example: c:\cleanup.vbs

Consider the following:

- The applications you specify in the session.xml file must already be installed on the workstation.
- Because Hot Desktop is part of the Single Sign-on Plug-in software, the plug-in software starts automatically and does

not need to be specified in this file.

Other applications specified in session.xml can launch under the Hot Desktop shared account shell, which can prompt users for credentials. The plug-in software then performs according to settings in the user configurations.

Important: Save the session.xml file in UTF-8 format. ANSI encoding is acceptable if all characters are in the 0 to 127 (standard English character set) range. If your session.xml file contains special or foreign characters such as Asian language characters, you must save it in UTF-8 format.

The Process.xml File

Feb 08, 2011

Note: The process.xml file is created on each workstation or device where Hot Desktop is installed in the C:\Program Files\Citrix\MetaFrame Password Manager\HotDesktop folder. A sample process.xml file is also located in the \Support folder of the installation media. Therefore, any changes you want to make to this file must be performed on a device-by-device basis. However, refer to the Citrix Support article http://support.citrix.com/article/CTX110394 to learn how to replace each user process.xml file through a Machine Group Policy in Active Directory.

Use the process.xml file to specify which applications continue to run after a Hot Desktop User logs off. These applications are known as persistent applications or persistent processes.

You can also use the process.xml file to specify applications that terminate after a Hot Desktop User logs off. These applications are known as transient applications or transient processes.

Note that you must include your desired tags within the <configuration> and </configuration> tags in the file.

Important: Save the process.xml file in UTF-8 format. ANSI encoding is acceptable if all characters are in the 0 to 127 (standard English character set) range. If your process.xml file contains special or foreign characters such as Asian language characters, you must save it in UTF-8 format.

Use this section of the file to specify any applications or file types to be run as the Hot Desktop User. This setting helps ensure the security of those applications to be run using the credentials of the currently logged on users.

Note: After installation, the plug-in software automatically specifies a shell executable application named ssoshell.exe (the Single Sign-on Plug-in software) in the process.xml file. By default, it is specified as a process to be run as the Hot Desktop User.

While the start script in the session.xml file specifies the applications that launch when a Hot Desktop session first starts up, <shellexecute_processes> lists those applications that users can launch in the context of their Hot Desktop session.

```
<shellexecute_processes>
 <process>
  <name>appname</name>
 </process>
</shellexecute_processes>
```

where:

appname	Indicates the application name only of the process or application to be run. The full path is not required. For example: pnagent.exe.

Note: process.xml allows the use of a wildcard (*) in addition to static file names such as Notepad.exe. Wildcards can be used alone or in combination with file names. For example, *.txt, pnagent.exe, and *.doc are all valid appnames.

Use this section of the file to specify any applications that continue to run after the Hot Desktop User logs off. Specified applications are not terminated on shutdown (logoff) of Hot Desktop sessions, even if they were started during a session.

Specify the full path of the persistent process to ensure that only the correct processes remain running after each session.

```
<persistent_processes>
 <process>
  <name>path_options</name>
 </process>
</persistent_processes>
```

where:

path_options	Indicates the fully qualified folder path to the application executable file or script on the local computer and any options to run with the application. For example: `c:\program files\Internet Explorer\iexplore.exe http://www.yahoo.com`

Note: After installation, the plug-in software automatically creates an entry for a persistent application named activator.exe in the process.xml file. The activator.exe application provides users with their Hot Desktop session indicator. The session indicator is a transparent moveable window users see when they are logged on; it contains information about users and their sessions as defined by the administrator. By default, activator.exe is specified as a persistent process so that it is not restarted when each Hot Desktop User logs on or off.

Use this section of the file to specify any applications that will terminate after the Hot Desktop User logs off.

Note: After installation, the plug-in software automatically specifies a transient application named shellexecute.exe in the process.xml file. By default, it is specified as a transient process so that it is terminated when each Hot Desktop User logs off.

```
<transient_processes>
 <process>
  <name>appname</name>
 </process>
</transient_processes>
```

where:

appname	Indicates the application name only of the process or application to be terminated. The full path is not required. For example: `pnagent.exe`.

Reference

May 11, 2015

This reference describes the settings and setting default conditions available in the Single Sign-on node of the Citrix AppCenter, grouped by their locations in the console.

This section describes the user configuration settings and controls. All navigation hints provided in this section are made to an existing user configuration when performing an edit function. To access the Edit User Configuration dialog box, navigate as follows:

Start > All Programs > Citrix > Management Consoles > Citrix AppCenter > Single Sign-on> User Configurations > [configuration] > Edit user configuration

These controls customize how the Single Sign-on Plug-in works for this user configuration. The user interface preferences are set here.

Start > All Programs > Citrix > Management Consoles > Citrix AppCenter > Single Sign-on> User Configurations > [configuration] > Edit user configuration > Basic Plug-in Interaction

allow users to reveal passwords

This setting controls if users can reveal passwords in the Manage Passwords window. When the setting is not selected, the Reveal Password button is disabled. To restrict the ability to reveal a password to specific applications, select this setting and then use the corresponding password policy setting to control whether users can reveal passwords for applications managed by that policy.

Default setting: selected

force re-authentication before revealing user passwords

This setting controls if users must re-authenticate to Single Sign-on before a reveal password request is honored.

Default setting: selected

automatically detect applications and prompt user to store credentials

This setting controls if the plug-in software prompts the user to add credentials for applications newly detected by the plug-in software.

Clear this option to disable the Single Sign-on Plug-in software's ability to detect any applications that are not associated with this user configuration. If this option is cleared, users must submit credentials manually to these applications. Use this setting to prevent users from adding applications that are not currently part of their assigned user configuration to their set of SSO-enabled applications.

If cleared, this option overrides the Enable users to cancel credential storage when a new application is detected option available on the Advanced Settings > Client-Side Interaction page. Also, if you plan to use provisioning, clearing this option prevents users from being prompted to enter their credentials.

Default setting: selected

automatically process defined forms when Single Sign-on Plug-in detects them

Select this option to permit the plug-in software to submit stored credentials automatically without user intervention. Credential fields in the application will automatically populate if the corresponding setting Submit this form automatically is selected in the application definition associated with this user configuration.

Default setting: selected

time between re-authentication requests

This setting specifies the time between plug-in re-authentication requests. When the specified time expires, the user's device is locked and users must re-authenticate by entering their primary credentials. The minimum allowed value is 1 minute.

Default setting: 8 hours

These controls are used to set the credential submission delay and the columns in the Manage Passwords window.

Start > All Programs > Citrix > Management Consoles > Citrix AppCenter > Single Sign-on> User Configurations > [configuration] > Edit user configuration > Plug-in User Interface

Specify the length of time plug-in delays credential submission

Select this setting to specify the length of time the plug-in software delays credential submission after detecting an allowed application. If selected, specify the length of time (in seconds) to delay credential submission. Use this setting to ensure that the application is ready to receive the credentials. During this time, the plug-in software will show a progress indicator, indicating that the plug-in software is working.

Default setting: not selected (0 seconds)

Set the default columns and column order in Logon Manager

This setting controls which columns are shown in the Details view of the Manage Passwords window (formerly known as Logon Manager). It also controls the order in which the columns are presented.

The default settings are:

- Application Name
- Description
- Group
- Last Used
- Modified

These settings are used to configure plug-in software event logging, registry key retention on shutdown, and credential storage on newly detected applications.

Start > All Programs > Citrix > Management Consoles > Citrix AppCenter > Single Sign-on > User Configurations > [configuration] > Edit user configuration > Client-side Interaction

log Single Sign-on Plug-in events using Windows event logging

Select this control to track plug-in software informational events in the local Windows Event Log. Warnings and error events are always logged, regardless of this setting.

Default setting: not selected

delete user's data folder and registry keys when Single Sign-on Plug-in is shut down

Select this control to delete the user's registry keys and data folder (including encrypted credentials) when the plug-in software is shut down.

Default setting: not selected

enable users to cancel credential storage when a new application is detected

This setting is used to control whether users are prompted to store credentials every time the plug-in software recognizes an application for which no credentials are stored. If selected, users can choose to store their credentials in the Manage Passwords window (formerly known as Logon Manager) now, later, or never. If the setting Automatically detect applications and prompt users to store credentials is not selected on the Configure plug-in interaction page, the plug-in software does not prompt users to store credentials.

Default setting: selected

limit the number of days to keep track of deleted credentials

Use these controls to specify how long the central store tracks credentials deleted from Manage Passwords window (formerly known as Logon Manager). When user credentials are stored on multiple client devices, the plug-in deletes the credentials when it synchronizes with the central store during this time period. If the credentials are still stored on the client device when the time elapses, they are restored when the plug-in synchronizes with the central store.

Default setting: selected / 180 days

These controls are used to allow users to refresh Single Sign-on Plug-in settings, synchronize user configuration information, allow the plug-in to continue to operate if it cannot connect to the central store, and to specify automatic synchronization intervals

Start > All Programs > Citrix > Management Consoles > Citrix AppCenter > Single Sign-on> User Configurations > [configuration] > Edit user configuration > Synchronization

allow users to update Single Sign-on Plug-in settings

Select this setting to allow users to refresh the plug-in software settings in Manage Passwords window (formerly known as Logon Manager). When the setting is not selected, the Manage Passwords window Refresh button is disabled.

Default setting: selected

synchronize every time users launch recognized applications or Logon Manager

Select this setting to have the plug-in software synchronize user configuration information whenever a user launches a recognized application or Manage Passwords window (formerly known as Logon Manager). Frequent synchronization can degrade performance on both the client and server, as well as increase network traffic.

Default setting: not selected

allow Single Sign-on Plug-in to operate when unable to reconnect to central store

This setting controls whether or not Single Sign-on operates when unable to connect to the central store for synchronization. When selected, a licensed instance of Single Sign-on Plug-in continues to operate even if the connection fails. If not selected, the plug-in software operates only when connected to the central store.

Default setting: selected

specify the time between automatic synchronization requests

This control is used to specify the time in minutes between automatic synchronization attempts. Automatic synchronization is independent of user activity and takes place in addition to other events that trigger synchronization.

Default setting: not selected / 0 minutes

allow user credentials to be accessed through the Credential Synchronization Module

Select this setting to allow remote clients to access user credentials through the service. This option is used with the Account Association feature, which allows a plug-in software user to log on to any application from one or more Windows accounts.

Default setting: not selected

Because companies can maintain multiple Windows domains, users can also have more than one Windows account. The Account Association option allows a user to log on to any application from one or more Windows accounts. These controls allow users to associate logon information among multiple Windows accounts.

Start > All Programs > Management Consoles > Citrix AppCenter > Single Sign-on > User Configurations > [configuration] > Edit user configuration > Account Association

allow users to associate accounts

Select this setting to allow users to associate multiple Windows accounts, provide the URL, and port where the Credential Synchronization Module is installed. This option cannot be set when initially configuring a User Configuration. It can be defined only when editing an existing configuration.

Default setting: not selected

provide default service address

Select this setting to allow the default service address and service port to the Credential Synchronization Module to be defined. After defining the settings, you can select the Validate option to validate the address path and service port.

Default setting: <AddressOfYourServer > /MPMService/

service port: 443

allow users to edit service address

If a service address is defined, select this setting to allow the user to edit the settings through the plug-in interface. Select this option if credential synchronization is run in multiple places and users need to be able to switch.

Default setting: not selected

provide default domain

Select this setting to specify the default domain used for authentication when the plug-in software synchronizes with the associated Windows account. If this setting is selected, enter the default domain name in the space provided. If you do not provide the domain, users might be confused as to which user credentials they should provide.

Default setting: not selected

allow users to edit domain

Select this setting to allow users to edit the default domain used for authentication when the plug-in software synchronizes with the associated Windows account.

Default setting: not selected

allow users to remember password

Select this setting to allow users to save their associated Windows account password in the plug-in software.

Default setting: not selected

These controls allow the plug-in software to detect client-side application definitions, enable support for terminal emulator, and specify the minimum number of domain name levels to match for web applications.

Start > All Programs > Management Consoles > Citrix AppCenter > Single Sign-on > User Configurations > [configuration] > Edit user configuration > Application Support

detect client-side application definitions

Select this setting to allow Single Sign-on to detect applications in one of the following ways.

- All applications
  Detects and responds to applications defined by an administrator or a user (in Manage Passwords window, formerly known as Logon Manager) and defined in the default settings at installation.
- Only applications that are defined by users in Logon Manager

Detects and responds to applications defined by an administrator and a user in Manage Passwords window (formerly known as Logon Manager). The plug-in software will not recognize or respond to applications defined in the default settings at installation.

- Only applications that are included with Single Sign-on Plug-in
  Detects and responds to applications defined by an administrator and defined in the default settings at installation. Users cannot create their own application definitions from Manage Passwords window (formerly known as Logon Manager).

Default setting: All applications

enable support for terminal emulators

This setting controls support for terminal emulation programs. When this setting is enabled, the plug-in software runs a process that detects terminal emulators and terminal emulator-based applications.

Default setting: not selected

time interval in which plug-in checks the terminal emulator for changes

This setting is used to specify how much time in milliseconds must pass before the plug-in software checks the terminal emulator for screen changes. Lower values can use more CPU time on the client and increase network traffic.

Default setting: 3000 milliseconds

number of domain name levels to match

This setting is used to specify the minimum number of domain name levels to match for allowed Web applications. A value of 2 or less would match *.domain1.topleveldomain; a value of 3 would match *.domain2.domain1.topleveldomain. Domain name levels beyond the specified number are treated as wild cards. To strictly control URL matching for Web applications, set strict URL matching in your application definitions.

Default setting: 99

These controls specify how Hot Desktop sessions are handled.

Start > All Programs > Management Consoles > Citrix AppCenter > Single Sign-on > User Configurations > [configuration] > Edit user configuration > Hot Desktop

session settings script path

This control specifies the path of the session settings file that defines the scripts to be executed at the start and end of a Hot Desktop session. The start script can be used to start applications. The stop script can be used to perform cleanup tasks such as file removal. The file used must be accessible to all users.

Default setting: [blank]

lock time-out

This control is used to specify the length of time in minutes that a Hot Desktop session will remain active when the workstation is idle. If this interval is exceeded, the desktop is locked.

Default setting: 10 minutes

session time-out

This control is used to specify the length of time in minutes that a Hot Desktop session will run while the desktop is locked. If this time is exceeded, the session is terminated and a new session is started when the desktop is unlocked.

Default setting: 5 minutes

enable session indicator

This setting controls whether a window identifying the Hot Desktop session is enabled. When this setting is selected, a transparent moveable window appears on the desktop during Hot Desktop sessions. This window displays the user's name and the elapsed time of the active session.

Default setting: selected

enable graphic

This control is used to specify the path of the graphic file displayed in the Hot Desktop session indicator. The specified file must be in a location accessible to all users and must be in Windows bitmap (.bmp) file format.

A default bitmap named Citrix.bmp is available from the %ProgramFiles%\Citrix\MetaFrame Password Manager\Hot Desktop folder on each Hot Desktop workstation.

Default setting: [none]

These controls are used to identity the license server and licensing model.

Start > All Programs > Management Consoles > Citrix AppCenter > Single Sign-on > User Configurations > [configuration] > Edit user configuration > Licensing

Important: Locally installed instances of the Single Sign-on Plug-in do not require a separate license for users who have access to hosted applications in a Citrix XenApp, Platinum Edition environment.

license server name

The fully qualified domain name (hostname.domain.tld) associated with the license server must be identified.

Default setting: [blank]

use default value (for license server port number)

Select this setting to use the default access port on the license server. If the license server is listening on a different port than its default port, disable this setting and enter the access port in the provided field.

Default setting: selected

Default port: 27000

named user licensing

This option is selected if you choose Single Sign-on Advanced as the product edition. You can also choose this option if you select Single Sign-on Enterprise as the product edition. With this license type, Single Sign-on can be used only by specific, named users. If this option is selected, you must specify the time period (in days, hours, and minutes) that the license is assigned to the named user before the license expires and the plug-in software reconnects to the license server. The user maintains control of the license for the specified time period even if the user computer shuts down.

Default setting: selected if Single Sign-on Advanced Edition; not available if XenApp Platinum Edition

Default disconnect setting: 21 days

concurrent user licensing (Enterprise and Platinum Edition only)

This option is enabled if you select the product edition as Single Sign-on Enterprise or XenApp Platinum. It is not available if you select Advanced Edition as the product edition.
Note: This license model is enabled if you upgraded from Password Manager Version 4.1. Citrix Systems considers this previous version as equivalent to Single Sign-on 5.0 Enterprise Edition for licensing purposes when you upgrade.
With this license type, a single Single Sign-on license can be shared by different users (although not at the same time; this license type is sometimes also known as a floating license).

Default setting: selected if Single Sign-on Enterprise or XenApp Platinum Edition; not available if Single Sign-on Advanced Edition

Default disconnect setting: 1 hour, 30 minutes if Allow license to be consumed for offline use is not selected; 21 days if Allow license to be consumed for offline use is selected

allow license to be consumed for offline use

This option is available only if Concurrent User Licensing is selected. Select this setting to specify the amount of time that the user can be disconnected (offline) before the license expires and is returned to the pool of available licenses. If specified, the user maintains control of the license for the specified time period even if the user computer shuts down. The default time period is 1 hour 30 minutes; the recommended value is between 2 and 365 days.

Default setting: Not selected

continue without validating licensing information

This setting allows the editing process to continue without requiring a valid license server name and access port.

Default setting: not selected

Data Protection Methods

May 11, 2015

These settings are used to select the primary data protection methods to use to protect the credentials of your users. In some environments, users can use more than one method.

Start > All Programs > Management Consoles > Citrix AppCenter > Single Sign-on > User Configurations > [configuration] > Edit user configuration > Data Protection Methods

Select Yes to disallow administrator access to user credentials. Selecting this option disables the Microsoft Data Protection API options (including the DPAPI with profile selection in the Smart Card key source drop-down menu) and the Do not prompt users; restore primary data protection automatically over the network option on the Secondary Data Protection settings. With this configuration, the account or other administrator does not have access to user passwords or data. This setting helps prevent an administrator from impersonating a user. The administrator cannot log on as the user with the default setting and possibly access data located in the user's local credential store.

Select No to allow use of all the multiple authentication features here and the secondary data protection methods on the Secondary Data Protection configuration settings.

Default setting: Yes

Choose this selection to use the primary authentication features that are made available in the settings described in the following table.

Default setting: selected

Control	Description
Users authentication data	A user secret is used to access and protect user data. The authentication secret can be a user password or PIN-based device used in your environment. Default setting: selected To further protect the user data, you can also select the following: Allow Smart Card PINs Select to allow the smart card PIN to be used as the user secret for protection. Use this only if your enterprise or environment has a "strong PIN" policy. Default setting: not selected Allow protection using blank passwords Select this option only if your domain has low security requirements and allows

Control	Description
	users to have blank domain passwords. If you select this option and the plug-in software detects that the user has a blank password, a user secret is derived from the user ID. If you do not select this option, the plug-in software does not derive a user secret or otherwise perform any data protection with the blank password. If you select Users authentication data and do not select Allow Smart Card PINs and Allow protection using blank passwords, after the user logs on for the first-time enrollment and registration process with a blank password, an error message appears and the plug-in software is disabled. Default setting: not selected
Microsoft Data Protection API	Select this option if you are using roaming profiles implementing a Kerberos network authentication protocol for users. This option works only if roaming profiles are available. For example, select Users authentication data and this option if users are using passwords to access their computers and a Kerberos network authentication protocol to access a farm of computers running Citrix XenApp. This method also allows the use of user credentials and smart cards to log on. Default setting: not selected
Smart Card Certificate	Select to allow users to use cryptographic cards that enable encryption and decryption of authentication data. Citrix recommends that, if possible, you select this option if you are using Hot Desktop in your environment. Default setting: not selected

Select this option and select a method from the Smart Card key source drop-down menu to permit users to use a single primary authentication method and/or if you are using Password Manager Versions 4.0 or 4.1 plug-in software. If you upgraded your central store from Version 4.1 to Version 5.0, this option is selected automatically.

This option is availble only when the Triple DES encryption method is used.

The Smart Card key source choices are:

- PIN as password
- Smart Card Data Protect
- DPAPI with profile (not available if No is selected for Do you need to regulate account administrator access to user data?

Default setting: not selected

These options allow you to specify secondary credential data protection features to use before unlocking user credentials when users change their primary authentication (for example, when a domain password is changed, or a smart card is replaced). Alternatively, it also enables you to specify that credentials are restored automatically when implementing the

Key Management Module.

Start > All Programs > Management Consoles > Citrix AppCenter > Single Sign-on > User Configurations > [configuration] > Edit user configuration > Secondary Data Protection

prompt users to verify identity

Default setting: selected

Choose this button to select one of the following user reauthentication methods:

Control	Description
Prompt user to enter the previous password	If you select this option, note that users who forget their previous password will be locked out and must reenroll their secondary credentials. Do not select this option if your users employ smart cards for their primary authentication. Default setting: selected
Prompt user to select the method: previous password or security questions	If you select this option, users are prompted according to their choice of verification method. This option includes this suboption: Use identity verification as in previous versions of Password Manager Select this option if you upgraded from Password Manager Versions 4.0 or 4.1 and you enabled question-based authentication or identity verification questions. The 4.0 and 4.1 Versions of the plug-in software do not need access to the service in this case. Default setting: not selected

do not prompt users; restore primary data protection automatically over the network

Select this option when implementing the Key Management Service Module to bypass identity verification and automatically unlock user credentials. This method is less secure than other data protection methods but increases ease-of-use for your users by retrieving credentials automatically.

Default setting: not selected

The options available in this section require installation of the Key Management service module. This module inserts a button on the Windows logon dialog box that is used to allow users to reset their passwords.

Start > All Programs > Management Consoles > Citrix AppCenter > Single Sign-on > User Configurations > [configuration] > Edit user configuration > Self-Service Features

allow users to reset their primary domain password

Select this setting to allow users to reset their primary domain password without administrative intervention.

Default setting: not selected

allow users to unlock their domain account

Select this setting to allow users to unlock their domain account.

Default setting: not selected

These controls identify the service location and port for the Key Management Module.

Start > All Programs > Management Consoles > Citrix AppCenter > Single Sign-on > User Configurations > [configuration] > Edit user configuration > Key Management Module

service location (Key Management Module)

This setting is used to identify the service address and port for the Key Management Module. Use the Validate button to ensure the settings are valid.

Default setting: [blank]

service port: 443

The Provisioning Module allows the credentials associated with users in this user configuration to be imported, modified, and removed. This page requires you to specify the location and service port of the Provisioning Module.

Start > All Programs > Management Consoles > Citrix AppCenter > Single Sign-on > User Configurations > [configuration] > Edit user configuration > Provisioning Module

use provisioning

Select this setting to use provisioning.

Default setting: not selected

service location (Provisioning Module)

This setting is used to identify the service address and port for the Provisioning Module. Use the Validate button to ensure the settings are valid.

Default setting: [blank]

service port: 443

Application Definitions

May 11, 2015

This topic describes the application definition settings and controls. All navigation hints provided in this topic are made to an application definition when performing an edit function.

These controls set the rules that govern password length and character repetition.

Start > All Programs > Citrix > Management Consoles > Citrix AppCenter > Single Sign-on > Application Definitions > [definition] > Edit application definition > Application Forms > [defined form] > Edit > Other Settings

submit this form automatically

This setting is used to specify if the submit button is pressed automatically by the plug-in or if the user is required to press it manually. Select the Submits this form automatically check box to automatically submit the form without user intervention.

Default setting: selected

This control is used to identify the icon that is displayed next to the application in the Manage Passwords window (formerly known as Logon Manager).

Start > All Programs > Citrix > Management Consoles > Citrix AppCenter > Single Sign-on > Application Definitions > [definition] > Edit application definition > Application Icon

application icon

This setting controls the application icon that appears next to the application name in the Manage Passwords window (formerly known as Logon Manager). Two options are available:

- Use default icon
- Use custom icon (enter icon path below).

If a custom icon is to be used, use the browse feature to identify the path to the icon file. Any standard Windows icon file can be identified. Microsoft Windows environment variables are supported.

Default setting: use default icon

These controls are used to force the plug-in to ignore subsequent logon or password change forms during an application session when a logon or password change was already processed.

Start > All Programs > Citrix > Management Consoles > Citrix AppCenter > Single Sign-on > Application Definitions > [definition] > Edit application definition > Application Detection

process only the first logon for this application

Select this control to only process the first logon for this application and to ignore subsequent logon requests.

Default setting: not selected

process only the first password change for this application

Select this control to process only the first password change request for this application and to ignore subsequent password change requests.

Default setting: not selected

These controls are used to specify the settings for this application when the password expires. Single Sign-on expiration policy is enforced only if it is selected in the password policy associated with this application.

Start > All Programs > Citrix > Management Consoles > Citrix AppCenter > Single Sign-on > Application Definitions > [definition] > Edit application definition > Password Expiration

run script when password expires

Select this setting and identify the script and its absolute path to run a specific script file when the password expires. Do not use a Universal Naming Convention (UNC) path.

Default setting: not selected

use Citrix Single Sign-on expiration warning

Select this setting to use the Single Sign-on expiration warning when the password expires.

Default setting: not selected

Password Policies

May 11, 2015

This section describes the password policy settings and controls. All navigation hints provided in this section are made to an existing password policy when performing an edit function. To access the Edit Password Policy dialog box, navigate as follows:

Start > All Programs > Citrix > Management Consoles > Citrix AppCenter > Single Sign-on > Password Policies > [policy] > Edit password policy

These controls set the rules that govern password length and character repetition.

Start > All Programs > Citrix > Management Consoles > Citrix AppCenter > Single Sign-on > Password Policies > [policy] > Edit password policy > Basic Password Rules

minimum password length

Specifies the minimum number of characters required in the password. Minimum allowed value = 0. Maximum allowed value = 128.

default setting: 8

maximum password length

Specifies the maximum number of characters allowed in the password. Minimum allowed value = 1. Maximum allowed value = 128.

default setting: 20

maximum number of times a character can occur

Specifies the maximum number of times a character can occur in a password. Minimum allowed value = 1. Maximum allowed value = 128.

default setting: 6

maximum number of times the same character can occur sequentially

Specifies the maximum number of times the same character can occur sequentially. Minimum allowed value = 1. Maximum allowed value = 128.

default setting: 4

These controls set the rules that govern alphabetic character use in passwords.

Start > All Programs > Citrix > Management Consoles > Citrix AppCenter > Single Sign-on > Password Policies > [policy] > Edit password policy > Alphabetic Character Rules

allow lowercase characters

Controls whether or not lowercase alphabetic characters can be used in passwords.

default setting: allow lowercase characters

password can begin with a lowercase character

Controls whether or not passwords can begin with a lowercase character.

default setting: allow passwords to begin with a lowercase character

password can end with a lowercase character

Controls whether or not passwords can end with a lowercase character.

default setting: allow passwords to end with a lowercase character

minimum number of lowercase characters required

Specifies the minimum number of lowercase alphabetic characters required in a password. Minimum allowed value = 0. Maximum allowed value = 128.

default setting: 0

allow uppercase characters

Controls whether or not uppercase alphabetic characters can be used in passwords.

default setting: allow uppercase characters

password can begin with an uppercase character

Controls whether or not passwords can begin with an uppercase character.

default setting: allow passwords to begin with an uppercase character

password can end with an uppercase character

Controls whether or not passwords can end with an uppercase character.

default setting: allow passwords to end with an uppercase character

minimum number of uppercase characters required

Specifies the minimum number of uppercase alphabetic characters required in a password. Minimum allowed value = 0. Maximum allowed value = 128.

default setting: 0

These controls set the rules that govern numeric character (0-9) use in passwords.

Start > All Programs > Citrix > Management Consoles > Citrix AppCenter > Single Sign-on > Password Policies > [policy] > Edit password policy > Numeric Character Rules

allow numeric characters

Controls whether or not numeric characters can be used in passwords.

default setting: allow numeric characters

password can begin with a numeric character

Controls whether or not passwords can begin with a numeric character.

default setting: allow passwords to begin with a numeric character

password can end with a numeric character

Controls whether or not passwords can end with a numeric character.

default setting: allow passwords to end with a numeric character

minimum number of numeric characters required

Specifies the minimum number of numeric characters required in a password. Minimum allowed value = 0. Maximum allowed value = 128.

default setting: 0

maximum number of numeric characters allowed

Specifies the maximum number of numeric characters allowed in a password. Minimum allowed value = 1. Maximum allowed value = 128.

default setting: 20

These controls set the rules that govern special (non-alphabetic and non-numeric) character use in passwords.

Start > All Programs > Citrix > Management Consoles > Citrix AppCenter > Single Sign-on > Password Policies > [policy] > Edit password policy > Special Character Rules

allow special characters

Controls whether or not special (non-alphabetic and non-numeric) characters can be used in passwords.

default setting: allow numeric characters

password can begin with a special character

Controls whether or not passwords can begin with a special character.

default setting: allow passwords to begin with a special character

password can end with a special character

Controls whether or not passwords can end with a special character.

default setting: allow passwords to end with a special character

minimum number of special characters required

Specifies the minimum number of special characters required in a password. Minimum allowed value = 0, Maximum allowed value = 128.

default setting: 0

maximum number of special characters allowed

Specifies the maximum number of special characters allowed in a password. Minimum allowed value = 0, Maximum allowed value = 128.

default setting: 20

allowed special characters list

Specifies the special characters allowed in a password.

default setting: !@#$^&*()_-+=[]\|,?

These controls specify the characters and character strings that are not allowed in passwords.

Start > All Programs > Citrix > Management Consoles > Citrix AppCenter > Single Sign-on > Password Policies > [policy] > Edit password policy > Exclusion Rules

exclude the following list of characters or character groups from passwords

Select the Edit List option to open the Edit Exclusion List dialog box that is used to specify up to 256 individual characters or groups of characters that are not allowed in passwords. Enter one character or group of characters per line. Each group can contain up to 32 characters. Individual characters or groups of characters are not case-sensitive.

default setting: [blank]

do not allow application user name in password

Controls whether or not the application user name is allowed in password. Select this check box if the application user name is allowed in the password.

default setting: not selected

do not allow portions of application user name in password

Controls whether or not portions of the application user name are allowed in a password. This includes all possible character groups that can be taken from the user name. This setting is closely coupled to the Number of characters in portions setting. For example, when this setting is selected and the Number of characters in portions setting is set to four a password that includes character groups of “citr,” “itri,” or “trix” would not be allowed for a user with a user name of “citrix.”

default setting: not selected

do not allow Windows user name in password

Controls whether or not the Windows user name is allowed in password. If not selected, the Windows user name is allowed in the password.This setting is closely coupled to the Number of characters in portions setting. For example, when this setting is selected and the Number of characters in portions setting is set to four a password that includes character groups of “citr,” "itri,” or “trix” would not be allowed for a user with a Windows user name of “citrix.”

default setting: not selected

These controls specify whether or not a new password can be a repeat of a previous password, and the password expiration setting.

Password history is retained on a per-user basis. If you reset the user data for a user, the password history is removed and password history cannot be enforced for the deleted passwords.

The password expiration option notifies users only that a password will or has expired. Your users can use expired credentials, but are shown password change reminders or password change requests until the password is changed in Manage Passwords window (formerly known as Logon Manager).

Start > All Programs > Citrix > Management Consoles > Citrix AppCenter > Single Sign-on > Password Policies > [policy] > Edit password policy > Password History and Expiration

new password must not be the same as previous password

Controls whether or not the new password can be the same as a previous password. Previous passwords are kept in a password history.

default setting: new password can be the same as previous password (check box not selected)

number of previous passwords remembered

Specifies the number of previous passwords that are kept in the password history. Minimum allowed value is 1. Maximum allowed value is 24.

default setting: 1

use the password expiration settings associated with the application definitions

When selected, the settings (Number of days until password expires and Number of days to warn user before password expires) specified here are applied to application definitions associated with this policy. Single Sign-on policy operates independently of any existing password expiration policy built into the application.

default setting: password expiration not specified (check box not selected)

number of days until password expires

Specifies the maximum number of days that a password can remain unchanged. Minimum allowed value is 1. Maximum allowed value is 99999.

default setting: 42

number of days to warn user before password expires

Specifies the number of days before a password expires that a user starts to receive pending password expiration warnings. Minimum allowed value is 0. Maximum allowed value is 99998.

default setting: 14

These controls are used to test a manually generated password to verify compliance with the defined policy, automatically generate a compliant password, and verify that the defined constraints do not restrict the ability to generate enough passwords for your organization.

Start > All Programs > Citrix > Management Consoles > Citrix AppCenter > Single Sign-on > Password Policies > [policy] > Edit password policy > Test Password Policy

test the compliance of a manually created password

This field is used to test the compliance of a manually created password. Enter the manually created password and click Test. The entered password is tested against all the defined criteria.

default setting: none

generate a random policy-compliant password

This control is used to generate a password that complies with the currently defined password criteria. Click Generate to generate a compliant password that can be copied from the field (Ctrl-C).

default setting: none

generate and test a number of unique policy-compliant passwords

It is possible to define a set of password constraints that support a limited number of total password possibilities. This control is used to generate a user-defined number of compliant passwords to determine if the defined policy is flexible enough to meet the password needs of the organization. Click Generate multiple passwords to open a dialog box that allows you to generate a user-defined number of passwords.

default setting: none

These controls are used to define if the Reveal option is available for application definitions that use this policy, mandate that the user reauthenticate before submitting application credentials, set the number of logon retries, and set the amount of time the user has to successfully authenticate after a failed authentication attempt.

Start > All Programs > Citrix > Management Consoles > Citrix AppCenter > Single Sign-on > Password Policies > [policy] > Edit password policy > Logon Preferences

allow user to reveal password for applications

This control is used to determine whether or not the Reveal button in the Manage Passwords window (formerly known as Logon Manager) is available for applications managed by this policy. When users select the Reveal button in Manage Passwords window they can see their password in clear text. If this setting is not selected, users cannot reveal their passwords.

default setting: Reveal button not displayed (check box not selected)

force user to re-authenticate before submitting application credentials

This control is used to determine if users must enter their primary logon credentials before the plug-in submits credentials to the application. When this setting is selected, the Single Sign-on Plug-in immediately locks the workstation when it recognizes an application that is managed by this setting. Users must enter their primary credentials to unlock the workstation. When the workstation is unlocked with the proper credentials, the plug-in submits the user credentials to the application. This setting is useful for applications that access confidential or sensitive information because it forces users to verify their identities before the plug-in submits the credentials to the application.

default setting: User not forced to reauthenticate (check box not selected)

number of logon retries

This control is used to set the number of additional times the plug-in can submit user credentials to the same application within the specified time limit. When set to the minimum value of 0, users get an error message immediately upon a second attempt to submit credentials to the application.

default setting: 0

time limit for number of retries

This control is used to specify the amount of time (in seconds) the user is allowed to submit user credentials to the same application after the initial credential submission failed.

default setting: 30 seconds

This control is used to determine how the Password Change Wizard responds to Password Change Forms. One of four possible options must be configured:

- Allow users to choose a system-generated password or create their own password
- Only allow users to create their own password
- Only allow users to choose a system-generated password
- Generate a password and submit it to the application without displaying the Password Change Wizard

Start > All Programs > Citrix > Management Consoles > Citrix AppCenter > Single Sign-on > Password Policies > [policy] > Edit password policy > Password Change Wizard

allow users to choose a system-generated password or create their own password

Select this option to have the Password Change Wizard allow users to choose a system-generated password or create their own.

default setting: selected

only allow users to create their own password

Select this option to have the Password Change Wizard not allow users to choose a system-generated password, and require users to enter their own password.

default setting: not selected

only allow users to choose a system-generated password

Select this option to have the Password Change Wizard automatically use a system-generated password without allowing users to create their own password.

default setting: not selected

generate a password and submit it to the application without displaying the Password Change Wizard

Select this option to have the Single Sign-on Plug-in automatically submit a system-generated password without displaying the Password Change Wizard to the user. The user can see the fields on the password change screen being filled in and the resulting feedback from the application indicating whether or not the password was changed successfully.

default setting: not selected

Operations

Mar 18, 2011

Single Sign-on logs plug-in or user-generated events in the host computer's Windows event application log. Events are classified as information, warnings, or errors. Warning and error events are always logged. Logging of information events is disabled by default, but you can enable it on the console after you create your user configuration.

Single Sign-on logs events for features such as Hot Desktop, smart cards, licensing, and the Single Sign-on Service. The event log captures and verifies security-related events that you may need to track for regulatory compliance, such as for the Federal Information Processing Standard (FIPS) or for the Health Insurance Portability and Accountability Act of 1996 (HIPAA). Single Sign-on's event log capabilities also help increase your IT security.

If you are using Single Sign-on in a XenApp environment, the event log identifies both user and session information. All logon attempt failures are captured.

To enable information event logging:

1. In the console, find your user configuration and, from the Action menu, click Edit user configuration.
2. In the properties of the user configuration, select Client-Side Interaction.
3. Click Log Single Sign-on Plug-in events using Windows event logging.

The following table contains some of the standard events that Single Sign-on logs:

Standard Event Types	
Logon attempt failure (plug-in software authentication)	
	Logged during unsuccessful user authentication to Single Sign-on. Failure to open the credential store.
Logon attempt success (plug-in software authentication)	
	Logged during successful user authentication and success opening the central store.
Logon attempt (submitting credentials)	
	Logged during attempts to submit credentials to an external application.
Operations with credentials	
	Logged during operations involving passwords, such as change, reveal, and identity verification.
Synchronization failures (communication)	
	Logged during failure to synchronize with the central store due to communication issues.

Standard Event Types	
Synchronization failures (permissions)	
	Logged during failure to synchronize with the central store due to incorrect user credentials.
Smart card DataProtect encrypt/decrypt failure	
	Logged during general failure associated with encrypting or decrypting smart card data.
Smart card DataProtect encrypt/decrypt failure (missing card)	
	Logged when smart card is not available.
Plug-in software start up and shut down	
	Logged when smart card is not available.
Missing or corrupted .dll files	
	Logged when a .dll cannot be loaded correctly.

The following table contains some of the Hot Desktop events that Single Sign-on logs.

Hot Desktop Event Types	
Hot Desktop session logon failure	
	Logged only when there is a fatal error at session start up.
Hot Desktop session logon success	
	Logged when Hot Desktop starts a session due to successful user authentication.
Hot Desktop session logoff failure	
	Logged only when there is a fatal error during session termination.
Hot Desktop logoff success	
	Logged when a session terminates successfully due to user input or session time-out.

Hot Desktop Event Types

Mfrmlist.ini File

Feb 06, 2011

The Mfrmlist.ini file contains a list of the terminal emulators and locations to the HLLAPI dll that the Single Sign-on Plug-in software monitors. The file is located at:

%ProgramFiles%\Citrix\MetaFrame Password Manager\Helper\MFEmu

Single Sign-on Plug-in Does Not Submit Credentials

Mar 24, 2011

Occasionally the Single Sign-on Plug-in does not submit a user's credentials to a configured application. This problem can be caused by a number of reasons, such as:

- Changes in the web application leading to an out-of-date application definition.
- A setting unintentionally configured when creating the application definition.

Do the following initial activities to determine the cause of the submission failure:

- Check all settings for potential conflicts.
- Verify that the plug-in is configured to detect applications.
- Compare the plug-in and Single Sign-on console definitions.
- Remove matching criteria and submission fields one by one until the plug-in begins submitting credentials.

Important: Single Sign-on contains many settings to help you build application definitions, password policies, user configurations, and identification verification methods. It is possible to create contradictory settings where, for example, credentials are not submitted to an application.

If the Single Sign-on Plug-in still fails to submit the user's credentials, try the following troubleshooting techniques for Web-based and terminal emulator-based applications.

- Verifying that StrictURL Is used correctly
  1. In the Single Sign-on component of AppCenter, select the Web-based application you want to view.
  2. From the Action menu, click Edit application definition.
  3. Click Application Forms, select an application form and then click Edit.
  4. Click Form Identity. From here, you can enable Strict URL matching as well as URL case- sensitivity.
  5. Make sure that pages use HTML-compliant field types. Web application definitions require HTML-compliant field types. Undefined and user-defined field types are not detected.

- When using InPrivate Browsing in Internet Explorer 8, ensure Disable toolbars and extensions when InPrivate Browsing starts is not selected. See Microsoft's website for details of Internet Explorer privacy features.

Create terminal emulator-based application definitions using the Application Definition Wizard and Form Definition Wizard. When adding the application definition to a user configuration, be sure to enable support for terminal emulators.

- Verify that the terminal emulator is configured in the Mfrmlist.ini file

  The Ssomho.exe process that controls Single Sign-on interaction with terminal emulators recognizes only emulators defined in the Mfrmlist.ini file. If the terminal emulator is not defined in this file, the Ssomho.exe process does not attempt to communicate with the terminal emulator.

- Verify that a session short name is specified

  The Ssomho.exe process uses the session short name to communicate with the HLLAPI dll. Without a session short name, Ssomho.exe loads, but cannot monitor the screen activity. Configure the session short name on the terminal emulator on your client device.

- Verify that the Ssomho.exe process is running

  Follow these instructions to make sure Ssomho.exe is running:

1. On the computer running the Single Sign-on Plug-in software, open Task Manager and select the Processes tab.
2. Click the Image Name heading to sort the processes by image name.
3. Verify that Ssomho.exe is listed.

If the Ssomho.exe process is not listed, the process could be failing to locate any HLLAPI dlls, or it could be terminating prematurely because of third-party terminal emulator-related issues.

Note: Even if the Ssomho.exe process is listed, it may not be communicating with the HLLAPI dll successfully. Verify the session short name is correct before pursuing further troubleshooting alternatives.

- Test each terminal emulator individually

  If you installed multiple supported emulators on the same system, Ssomho.exe attempts to communicate with all of them. Occasionally, one of the HLLAPI dll implementations may cause Ssomho.exe to be unstable. Test each terminal emulator individually by removing the other host emulators or by commenting out and resequencing the entries in the Mfrmlist.ini file.

  This step helps verify that the ssomho process is not inadvertently connecting to an emulator other than the one you are attempting to troubleshoot.

Supporting Terminal Emulators

May 11, 2015

To enable HLLAPI support for any terminal emulator in Single Sign-on, you must enable support for terminal emulators in the console.

When terminal emulator support is enabled, SSOShell starts the Ssomho.exe process. This process first reads the Mfrmlist.ini file located at %Program Files%\Citrix\MetaFrame Password Manager\Helper\MFEmu, then looks for all configured emulators and attempts to load the HLLAPI-compliant .dll assigned in the file.

The Mfrmlist.ini file can be extended to accommodate additional HLLAPI-compliant emulators.

The Ssomho.exe process looks in the HKEY_LOCAL_MACHINE\SOFTWARE registry hive for the location of the HLLAPI-compliant .dll unless otherwise specified in the Mfrmlist.ini file.

Some terminal emulators place the location in the HKEY_CURRENT_USER hive. For those emulators, manually specify the location of the DLL file using the explicit path setting in the Mfrmlist.ini file.

To configure emulator support

Configuring Single Sign-on to work with the tested emulator programs is a multistep process. This process requires installing the emulator software, creating an emulator session to be used with Single Sign-on, and configuring Single Sign-on with a terminal emulator-based application definition that uses text matching so it can recognize a particular emulator session.

1. Install the terminal emulator software and restart the computer.
2. Start the terminal emulator software and create a new session, defining the display and the connection.
3. Set the session short name.
4. Enable HLLAPI API support.
   Note: A separate terminal emulator application definition is required for each unique session that will be used with Single Sign-on. The plug-in software detects sessions by matching text on the terminal emulator-based application screen with text in a specified row and column provided in the application definition. Single Sign-on submits the credentials based on row and column information provided in the application definition. Therefore, each unique session requires its own host application definition.
5. Save and close your session.
6. Exit the terminal emulator.
7. Create an application definition for the host application.
8. Open the console and verify that support is enabled in the appropriate user configurations.
9. Run the emulator and open the session.
10. Start or refresh the Single Sign-on Plug-in software.

The plug-in software recognizes the connection screen and displays a form for credentials to be entered and saved.

Single Sign-on Plug-in Software Does not Start

Feb 07, 2011

The Single Sign-on Plug-in software should be the last GINA-altering software installed on your non-Windows Server 2008, Windows Server 2008 R2, Windows Vista, or Windows 7 devices. If the Single Sign-on Plug-in software is installed but does not start as expected, it might be caused by a broken GINA chain. This happens when software installed or upgraded after the Single Sign-on Plug-in software alters the Windows GINA chain. Software packages that support smart card authentication, Symantec, and XenApp are all known to alter the Windows GINA chain.

If Single Sign-on is already installed and you plan to install or upgrade software that alters the Windows GINA chain, first uninstall the Single Sign-on Plug-in software. When the Single Sign-on Plug-in software is uninstalled, install the new software (or upgrade), then reinstall the Single Sign-on Plug-in software. This ensures that the correct .dll file is installed and registered for use with Single Sign-on.

Recommended Reinstallation Steps

1. Uninstall any third-party software that alters the GINA chain.
2. Uninstall the plug-in software.
3. Install the third-party software.
4. Install the plug-in software.

If you recently upgraded or installed third-party software and you suspect that it may have altered the Windows GINA chain, check the Windows registry entry and the client device to verify the presence and the location of the GINA chain.dll files appropriate to your installation. If the files are not located on the computer, uninstall and reinstall the Single Sign-on Plug-in software.

Important: When uninstalling software that may have disrupted the GINA chain, it is important to uninstall the software in the reverse order in which it was installed on the user device. Failure to uninstall in the reverse order in which GINA-altering software was installed can leave the computer in an invalid state. Do not edit the registry.

Creating a New Signing Certificate

Feb 07, 2011

The Single Sign-on Service generates event log alerts just prior to and upon signing certificate expiration. Create a new certificate to stop event log alerts. Use CtxCreateSigningCert.exe to create a new certificate. Use the Data Signing Tool, CtxSignData.exe, to sign the data (using keys supplied by the new certificate) in your central store.

You do not need to create a new signing certificate after you first configure the Single Sign-on Service unless one of the following statements is true:

- Your signing certificate is about to expire or has expired
- You believe your signing certificate is compromised

To create a new certificate, you must run CtxCreateSigningCert.exe, available from the %ProgramFiles%\Citrix\MetaFrame Single Sign-on\Service folder. At a command prompt of the computer running the Single Sign-on Service, type `CtxCreateSigningCert.exe`.

Enter the public key file name, the private key file name, and the time, in months, before the signing certificate expires. The new certificate is created.

CtxCreateSigningCert	
Usage:	CtxCreateSigningCert <name_of_public_cert> <name_of_private_cert> <expiration_period_in_months>
Where:	<name_of_public_cert> = File name to use for the public certificate <name_of_private_cert> = File name to use for the private certificate <expiration_period_in_months> = Number of months before the certificate expires
Example:	`ctxcreatesigningcert “C:\PublicKeyCert.cert” “C:\PrivateKeyCert.cert” “12”`

Signing, Unsigning, Re-signing, and Verifying Data

May 11, 2015

The Data Signing Tool, CtxSignData.exe, allows you to sign, re-sign, unsign, and verify in your central store. It is a command-line driven tool available from the installation media under \Service. CtxSignData.exe is also installed on the server hosting the service at %ProgramFiles%\Citrix\MetaFrame Password Manager\Service\SigningTool\CtxSignData.exe.

Note: The Data Signing Tool is installed with the Data Integrity Module of Single Sign-on Service. This module can be installed at a later time if it was not part of the initial Single Sign-on installation.
To start the data signing tool, at a command prompt of the computer running Single Sign-on Service, type `CtxSignData.exe` and use the appropriate command line parameter (-s, -r, -u, -v).

Signing Data (-s)

Use the sign command-line parameter to enable data integrity in an environment with existing unsigned data.

Note: If you have a Single Sign-on environment that is running without data integrity implemented and you later decide to use data integrity, you must use the Data Signing Tool to sign data in the existing central store.
You must supply the signing certificate file name, the Single Sign-on Service Uniform Resource Identifier (URI), the location of the central store, and central store type (NTFS network share or Active Directory). All data is read and signed using the new signing certificate.

The syntax for the CtxSignData command with the -s parameter is:

```
CtxSignData [-s service_path certificate_file centralstore_location NTFS|AD]
```

where:

-s	Signs data files in the central store
service_path	Indicates the Single Sign-on Service path in URI format
certificate_file	Indicates the filename of the certificate to use for signing or resigning data
centralstore_location	Indicates the Universal Naming Convention (UNC) path to the location of the file share or Domain Name System (DNS) of the Active Directory domain controller
NTFS | AD	NTFS | AD = Central store network directory service type, where ● NTFS = Microsoft NTFS file share ● AD = Microsoft Active Directory

The following are examples of the CtxSignData command with the -s parameter:

```
ctxsigndata -s “mpmserver.mycompany.com/MPMService” “C:\priv12mos.cert” “\\MPMCentralServer\citrixsync$” NTFS

ctxsigndata -s mpmserver.mycompany.com/MPMService “C:\priv12mos.cert” DC1.mycompany.com AD
```

Re-signing Data (-r)

Use the re-sign command-line parameter when the existing signing certificate is nearing expiration, has expired, or is compromised. You must supply the new signing certificate file name, the Single Sign-on Service URI, the location of the central store, and central store type (NTFS network share or Active Directory). All data is read and verified and then signed using the new certificate. No setting changes are necessary in the console or plug-in software because they already have data integrity enabled.

Use the following steps to re-sign corrupt data:

1. Open the Single Sign-on component of the Citrix AppCenter and locate the user configuration that is affected.
2. Open the user configuration to verify the data can be read from the central store.
3. Close the user configuration to save new corruption-free data in the central store.

4. Use the signing tool (ctxsigndata) to re-sign the data in the central store.

Note: If the corruption appears to be caused by a security breach, perform this procedure for all user configurations before re-signing the data to avoid inadvertently signing unsecured data.
The syntax for the CtxSignData command with the -r parameter is:

```
CtxSignData [-r service_path certificate_file centralstore_location NTFS|AD]
```

where:

-r	Re-signs data files in the central store (includes -v)
service_path	Indicates the Single Sign-on Service path in URI format
certificate_file	Indicates the filename of the certificate to use for signing or re-signing data
centralstore_location	Indicates the Universal Naming Convention (UNC) path to the location of the file share or Domain Name System (DNS) of the Active Directory domain controller
NTFS|AD	NTFS|AD = Central store network directory service type, where ● NTFS = Microsoft NTFS file share ● AD = Microsoft Active Directory

The following are examples of the CtxSignData command with the -r parameter:

```
ctxsigndata -r “mpmserver.mycompany.com/MPMService” “C:\priv12mos.cert” “\\MPMCentralServer\citrixsync$” NTFS

ctxsigndata -r mpmserver.mycompany.com/MPMService “C:\priv3mos.cert” DC1.mycompany.com AD
```

Unsigning Data (-u)

Use the unsign command-line parameter when you disable data integrity. You must supply the signing certificate file name, the Single Sign-on Service URI, the location of the central store, and central store type (NTFS network share or Active Directory). All data is read without verification and the signatures are removed.

The syntax for the CtxSignData command with the -u parameter is:

```
CtxSignData [-u centralstore_location NTFS|AD]
```

where:

-u	Unsigns all the data files in the central store
centralstore_location	Indicates the Universal Naming Convention (UNC) path to the location of the file share or Domain Name System (DNS) of the Active Directory domain controller
NTFS|AD	NTFS|AD = Central store network directory service type, where ● NTFS = Microsoft NTFS file share ● AD = Microsoft Active Directory

The following are examples of the CtxSignData command with the -u parameter:

```
ctxsigndata -u “\\MPMCentralServer\citrixsync$” NTFS

ctxsigndata -u DC1.mycompany.com AD
```

Verifying Data (-v)

Use the verify command-line parameter to check that all data in the central store is signed and verified. You must supply the signing certificate file name, the Single Sign-on Service URI, the location of the central store, and central store type (NTFS network share or Active Directory). All data is read with verification and signed.

The syntax for the CtxSignData command with the -v parameter is:

```
CtxSignData [-v service_path centralstore_location NTFS|AD]
```

Where:

-v	Verifies signatures on the data files in the central store
service_path	Indicates the Single Sign-on Service path in URI format
centralstore_location	Indicates the Universal Naming Convention (UNC) path to the location of the file share or Domain Name System (DNS) of the Active Directory domain controller
NTFS|AD	NTFS|AD = Central store network directory service type, where ● NTFS = Microsoft NTFS file share ● AD = Microsoft Active Directory

The following are examples of the CtxSignData command with the -v parameter:

```
ctxsigndata -v “mpmserver.mycompany.com/MPMService” “\\MPMCentralServer\citrixsync$” NTFS
```

```
ctxsigndata -v mpmserver.mycompany.com/MPMService “https://mpmserver.mycompany.com/MPMService” DC1.mycompany.com AD
```

Displaying Help (-h)

Use the help command-line parameter to display help for the CtxSignData command.

The syntax for the CtxSignData command with the -h parameter is:

```
CtxSignData [-h]
```

Where:

-h	Displays the help

The following is an example of the CtxSignData command with the -h parameter:

```
ctxsigndata -h
```

Enabling and Disabling the Data Integrity Service on Single Sign-on Plug-in Software

Feb 06, 2011

The following registry key can be edited to enable or disable the Data Integrity Service for Single Sign-on Plug-in software.

HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\MetaFrame Password Manager\Extensions\SyncManager\PerformIntegrityCheck

Type: DWORD

Values:

0=Data Integrity Validation Disabled

1=Data Integrity Validation Enabled

Moving Data to a Different Central Store

May 11, 2015

There are several reasons why you may need to migrate password policies, application templates, application definitions, security questions, and other types of Single Sign-on administrative data:

- The user moves to a new domain
- A new server is added to the Single Sign-on environment
- A new domain is added so users can use Single Sign-on's Account Association feature
- Users begin using Account Association across existing domains
- Single Sign-on is moved from a test environment to a production environment

Migration is a two-step process performed through the Single Sign-on component of the Citrix AppCenter: Step 1. Export the existing administrative data; Step 2. Import the administrative data into the new environment. In most instances, you must also redirect users to the new central store.

The following table lists the data that does and does not migrate when you use the Export command:

Migrates	**Does not migrate**
Password policies (except for the Default and Domain policies)	User configurations
Application templates	People folders
Application definitions	Application groups
Security questions and security question groups used as part of question-based authentication	User credentials
	Questionnaires
	Single Sign-on Service data

Single Sign-on Service does not migrate from one central store to another. To successfully complete the migration if you are using a service, you will need to install Single Sign-on Service in a new location and have both the existing and new Service available temporarily after the migration.

Caution: Additional steps are required to ensure successful migration if the Self Service or Data Integrity service modules are installed or if Delete user's data folder and registry keys when Single Sign-on Plug-in is shut down is enabled in a user configuration.
User configurations do not migrate from one central store to another automatically. Instead, you must recreate user configurations and redirect users to the new central store. When Single Sign-on Plug-in synchronizes its data with data in the original central store, it recognizes that the values changed; the plug-in then copies the credentials to the new central store.

Migrating Data to a New Central Store

The Export Admin Data Wizard allows you to export all application definitions, application templates, password policies, and security questions and groups in the central store. You can choose to export or leave entire types of data, but this wizard does not allow you to act on a subset of data: for example, you must export all password policies or leave them in the old central store.

Unlike the other types of administrative data, you can choose which application definitions to export by using the Export application definition command.

Caution: Manual steps are required to ensure successful migration if the Self Service or Data Integrity service modules are installed or if Delete user's data folder and registry keys when Single Sign-on Plug-in is shut down is enabled in a user configuration.

To export administrative data

1. In the Citrix AppCenter, while connected to the original central store, click the Single Sign-on node and, from the Action menu, click Export administrative data.
2. Follow the on-screen directions for the Export Admin Data Wizard.

To import administrative data

1. On the new machine, install and start the Single Sign-on console component, completing the Configure and Run Discovery process.
   Note: The Configure and Run Discovery process allows you to identify the central store to which you want to connect.
2. In the Citrix AppCenter, while connected to the new central store, click the Single Sign-on node and, from the Action menu, click Import administrative data.
3. Follow the on-screen directions for the Import Admin Data Wizard.
4. Create new user configurations.
5. On the Citrix AppCenter, while connected to the original central store, select a migrated user configuration, from the Action menu, select Redirect users, and then

identify the location of the new central store. Repeat as needed.
6. Ensure all users log on to Single Sign-on at least once. It is now safe to shut down the original central store and service.

To migrate to a new central store if Delete user's data folder and registry keys when Single Sign-on Plug-in is shut down is enabled

If your enterprise enables Delete user's data folder and registry keys when Single Sign-on Plug-in is shut down in user configurations, complete the following steps to migrate your users' administrative data to a new central store. Failure to do so forces the migrated users to re-enroll, either through question-based authentication or automatic key recovery each time they log on to their computer. This is because the users' administrative data is deleted each time they log off or exit Single Sign-on Plug-in.

1. Migrate the administrative data to the new central store.
2. On the Citrix AppCenter, while connected to the new central store, create new user configurations. Do not enable Delete user's data folder and registry keys when Single Sign-on Plug-in is shut down.
3. On the Citrix AppCenter, while connected to the original central store, select a migrated user configuration, from the Action menu, select Redirect users, and then identify the location of the new central store. Repeat as needed.
4. Ensure all users log on to Single Sign-on at least once.
5. Write and run a script to update the type and location of the central store in the registry of users' computers. The following table provides the registry settings based on central store type.

Central Store Types	Old Settings	New Settings
NTFS to NTFS	HKEY_Local_Machine\SOFTWARE\Citrix\Metaframe Password Manager\Extensions\SyncManager\Syncs\DefaultSync\SSOSyncType = FileSyncPath HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\MetaFrame Password Manager\Extensions\SyncManager\Syncs\DefaultSync\Servers\Server1 = <OLD UNC path>	HKEY_Local_Machine\SOFTWARE\Citrix\Metaframe Password Manager\Extensions\SyncManager\Syncs\DefaultSync\SSOSyncType = FileSyncPath HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\MetaFrame Password Manager\Extensions\SyncManager\Syncs\DefaultSync\Servers\Server1 = <NEW UNC path>
NTFS to Active Directory	HKEY_Local_Machine\SOFTWARE\Citrix\Metaframe Password Manager\Extensions\SyncManager\Syncs\DefaultSync\SSOSyncType = FileSyncPath HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\MetaFrame Password Manager\Extensions\SyncManager\Syncs\DefaultSync\Servers\Server1 = <OLD UNC path>	HKEY_Local_Machine\SOFTWARE\Citrix\Metaframe Password Manager\Extensions\SyncManager\Syncs\DefaultSync\SSOSyncType = ADSyncPath
Active Directory to NTFS	HKEY_Local_Machine\SOFTWARE\Citrix\Metaframe Password Manager\Extensions\SyncManager\Syncs\DefaultSync\SSOSyncType = ADSyncPath	HKEY_Local_Machine\SOFTWARE\Citrix\Metaframe Password Manager\Extensions\SyncManager\Syncs\DefaultSync\SSOSyncType = FileSyncPath HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\MetaFrame Password Manager\Extensions\SyncManager\Syncs\DefaultSync\Servers\Server1 = <NEW UNC path>

Caution: Editing the registry incorrectly can cause serious problems that may require you to reinstall your operating system. Citrix cannot guarantee that problems resulting from the incorrect use of Registry Editor can be solved. Use Registry Editor at your own risk. Be sure to back up the registry before you edit it.

6. On the Citrix AppCenter, while connected to the new central store, select the new user configurations and enable Delete user's data folder and registry keys when Single Sign-on Plug-in is shut down. It is now safe to shut down the original central store and service.

Exporting Application Definitions

You can export single application definitions or any number of application definitions to an .xml file.

To export a single definition application

1. In the Citrix AppCenter, while connected to the original central store, expand the Single Sign-on node and then expand Application Definitions.
2. Select the application definition to be exported and, from the Action menu, click Export application definition.
3. In the Export application definition dialog box, save the application definition to a location you can access from the new console's computer.

To export multiple application definitions

1. In the Citrix AppCenter, while connected to the original central store, expand the Single Sign-on node and then click Application Definitions.
2. From the Actions menu, click Export application definitions.
3. Follow the on-screen directions for the Export Application Definitions Wizard.

To back up the service

When you back up important files, be sure to include the central store and its contents, certificates, and personal and private keys in your company's regular backup procedures.

Important: You must modify the permissions for these files in Windows if your central store is in an NTFS network share for them to be accessible to your backup program.

1. Take note of the settings you make when running the Service Configuration Tool to set up your service.
2. Export the service data to a secure share or disk using CtxMoveServiceData.exe:
   1. From a command prompt, go to %ProgramFiles%\Citrix\Metaframe Password Manager\Service\Tools.
   2. Type CtxMoveServiceData.exe –export \\server\share\backupfile.
      Note: Do not use environment variables in your path.
   3. When asked, type a password of your choice. Make note of the password.
      Important: The service data you save to your backup file will be encrypted using this password. Do not lose your password.
   4. When asked to confirm your password, type it again.
   5. Verify your backup file was created.

To restore the service

1. Install the service from the installation media.
2. Configure the service with the proper settings, using the notes you made prior to back up.
   Note: If you are using data integrity, make sure you configure the data integrity server location properly, whether the data integrity server location has changed or stayed the same.
3. Finish the configuration and allow the service to start. After the service starts, you can immediately stop the service if you choose.
4. Import the service data from a secure share or disk, using CtxMoveServiceData.exe:
   1. At a command prompt, go to %Program Files%\Citrix\Metaframe Password Manager\Service\tools.
   2. Type CtxMoveServiceData.exe –import <\\server\share\backupfile>.
   3. Enter the correct password when prompted.
   4. When asked if you want to overwrite AKR.DAT, select Yes.
5. Restart the service. The service is now ready for use.

Removing Deleted Objects from Your Central Store

Use the CtxFileSyncClean tool to delete orphaned configuration data files from NTFS Share central stores. These files became orphaned when the objects they pointed to were deleted. The CtxFileSyncClean tool does not delete user data files, even if that user was deleted. Run CtxFileSyncClean.exe from the \Tools directory of your installation media.

Application Definition Extensions

May 11, 2015

Although Single Sign-on administrators can generally create application definitions using the Single Sign-on component to the Citrix AppCenter and the Application Definition Tool, some applications have special considerations or requirements that need an external process to determine if an application started or to submit user credentials using Single Sign-on Plug-in.

To support applications that have these types of requirements, third-party implementers that create processes to satisfy these external processing requirements can use Application Definition Extensions in the Single Sign-on component of the Citrix AppCenter and the Application Definition Tool to configure when and how these processes are initiated.

Single Sign-on Plug-in Software Operation

There are two different types of application definition extensions:

- Identification Extensions
  Use external processes to determine if the target application is a form that requires user credential management actions. These external processes can be used instead of or in conjunction with other window detection algorithms defined in the form definition
- Actions Extensions
  Use external processes to perform the required user credential management actions. These external processes can be used instead of or in conjunction with other window action algorithms defined in the form definition

A single form definition can be configured to use application definition extensions to perform either or both of these operations.

Identification Extensions

The Single Sign-on Plug-in uses listener hooks to detect events on the desktop such as application instantiation, URL loading, HTML page document complete notices, and other similar events.

As these events occur, the plug-in determines if the target application requires any user credential management action (such as ignore, logon, change password, and so on). The determination is made by comparing the characteristics exposed by application against the defined characteristics that uniquely identify a form. These characteristics include the Windows title and the executable file name (at a minimum) and, if required, other advanced matching characteristics that can include using an external process to identify the form (identification extension).

If an external identification process is required, the process or processes are identified in the form definition. The form definition includes information about the identification extension and any associated parameters. These are directly associated with a registry setting.

After the plug-in successfully processes the minimum matching and advanced matching algorithms, identification extensions that use an external process are evaluated.

When multiple identification extensions are defined to evaluate a form, the extensions are executed in the order that they appear in the identification extensions page (from top to bottom).

For each identification extension, the plug-in waits the specified amount of time (defined in the registry setting) for the external process to exit before it analyzes the process exit code.

If the minimum matching, advanced matching, and external matching processes complete with a zero return code, the target application is considered a match. If any matching process exits with any other value, the evaluation process stops and the application is considered not a match.

If a negative value is returned, an error is logged to the Windows Event Viewer. Positive values are written to a log file, if enabled.

The subsequent user credential management action can be performed by using any combination of standard Windows form actions, action sequences, or action extensions.

To define an identification extension

Configure identification extensions using the Form Definition Wizard during the application definition development process.

1. From AppCenter, expand the Single Sign-on node, select Application Definitions, and, from the Action menu, click Create application definition.
2. In the Application Definitions Wizard, continue to the Manage forms page, and select Add Form to start the Form Definition Wizard.
3. Advance through the definition process until the Identify form page appears.
4. On the Identify form page, click Advanced Matching. The appears.
5. In the Advanced Matching dialog box, click Identification Extensions.
6. On the Identification Extensions page, click Add to open the Add Identification Extension dialog box. The Add Identification Extension dialog box is used to define the following:

Extension ID	The extension ID identifies the ExtensionName to look for in the registry settings.
Description	A user-defined description of the identification extension being defined.
Parameters	Any name/value pairs (parameter name/parameter value) that are used to pass implementer-defined parameters to the external process that is launched by this extension.

The ExtensionName identifies a registry key name. This key name and its associated key values define the external identification process executable and its operating characteristics. The registry key name and its associated keys are located at:

[HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\MetaFrame Password Manager\Extension\{ExtensionName}]

Where the ExtensionName value is identified using the Extension ID value in the Add Identification Extension dialog box.

On 64-bit platforms the registry key name and its associated keys are located at:

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\MetaFrame Password Manager\Extension\{ExtensionName}]

The following table defines the key value characteristics.

Key	Type	Value
Type	REG_SZ	Must be EXECUTABLE

Key	Type	Value
Timeout	REG_DWORD	0 to wait forever for application to complete. Any other value is time to wait in milliseconds.
TerminateProcess	BOOL implemented as a REG_DWORD	(optional) On time-out, terminate process. TRUE—(default) Terminate process. FALSE—(0) Do not terminate process.
Executable	REG_EXPAND_SZ	The executable process and its fully qualified path.
Arguments	REG_SZ	Parameters for the executable.

The Executable value is the full path to the executable file. Environment variables are allowed. If the extension is implemented as a script, the script interpreter must be used for the Executable, and the script name as part of the Arguments. External processes can be developed using any editor/language or IDE of your choosing.

The Arguments value supports parameters that the plug-in software can replace with run-time parameters or the parameter name/value pairs specified in the Add Identification Extension dialog box. Each parameter that needs substitution must be prefixed and suffixed with a dollar sign ($) delimiter. For example, the following command-line arguments:

```
/h $_HANDLE$ /s $SAPSERVER$ /t $SAPTYPE$
```

appear to the executable as:

```
/h 1275366 /s “Houston, TX” /t 43
```

The Microsoft Windows handle associated with the application is a supported internal parameter defined as $_HANDLE$.

All internal parameters use $_ as a preface to avoid naming conflicts. Implementer parameters are not allowed to use underscores in key names.

Substitution precedence is defined to preserve parameter values after they are written. The precedence is defined as internal parameters (such as $_HANDLE$), followed by implementer parameters, followed by environmental variables.

All implementer parameters are permitted to use lowercase and uppercase letters and numbers in key names. Key names are case-insensitive.

If the extension identification executable requires parameters to be presented in a specific sequence, the Argument must support the required sequence. The parameter name/value pairs defined in the Add Identification Extension dialog box can be in any sequence.

Action Extensions

Action extensions use an external process to manage user credential management actions. The extension definition process has the ability to pass user credentials to the external application.

After a user credential management form is successfully identified (see

— Identification Extensions

), the subsequent user credential management action can be performed using any combination of standard Windows form actions, action sequences, or action extensions.

The Single Sign-on Plug-in supports the same features described in

— Identification Extensions

.

The plug-in executes the external process and waits the specified time for the process to exit (if WaitForCompletion is set to TRUE) and then analyzes its Process Exit Code. If the process exits with a zero return value, the extension executed successfully. Any non-zero return indicates an error.

If a negative value is returned, the error is logged to the Windows Event Viewer. Positive values are written to a log file, if enabled (see

— Enabling Logging

for additional information).

To define an action extension

Configure action extensions using the Form Definition Wizard during the application definition development process.

1. From the Citrix AppCenter, expand the Single Sign-on node, select Application Definitions, and, from the Action menu, click Create application definition.
2. In the Application Definitions Wizard, continue to the Manage forms page, and select Add Form to start the Form Definition Wizard.
3. Advance through the definition process until the Define form actions page appears.
4. On the Define form actions page, click Action Editor.
5. From the Action Editor dialog box, select Launch action extension. The Configuration Actions panel appears. This panel is used to view, edit, or add Launch action extension entries to the action sequence.
6. To add an action extension to the action sequence, provide the following information and click Insert:

ID	The ID identifies the ExtensionName to look for in the registry settings.
Description	A user-defined description of the action extension being defined.
Parameters	Any name/value pairs (parameter name/parameter value) that are used to pass implementer-defined parameters to the external process that is launched by this extension.

As with the identification extensions, the ExtensionName identifies a registry key name. This key name and its associated key values define the action processing executable and its operating characteristics. The registry key name and its associated keys are located at:

[HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\MetaFrame Password Manager\Extension\{ExtensionName}]

Where the ExtensionName value is identified using the ID value in Action configuration panel.

On 64-bit platforms the registry key name and its associated keys are located at:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Citrix\MetaFrame Password Manager\Extension\
{ExtensionName}]
```

The following table defines the key value characteristics.

Key	Type	Value
Type	REG_SZ	Must be EXECUTABLE
Timeout	REG_DWORD	0 to wait forever for application to complete. Any other value is time to wait in milliseconds.
TerminateProcess	BOOL implemented as a REG_DWORD	(optional) On time-out, terminate process. TRUE—(default) Terminate process. FALSE—(0) Do not terminate process.
WaitForCompletion	BOOL implemented as a REG_DWORD	(optional) Plug-in waits for process to exit. TRUE—(default) Wait. FALSE—(0) Do not wait.
Executable	REG_EXPAND_SZ	The executable process and its fully qualified path.
Arguments	REG_SZ	Parameters for the executable.

The Executable value follows the same conventions as the identification extensions.

The Arguments value supports parameters that the plug-in can replace with run-time parameters or the parameter name/value pairs specified in the Launch action extension view of the Action Editor. Each parameter that needs substitution must be prefixed and suffixed with a dollar sign ($) delimiter. For example, the following command-line arguments:

```
/h $_HANDLE$ /s $SAPSERVER$ /t $SAPTYPE$
```

appear to the executable as:

```
/h 1275366 /s “Houston, TX” /t 43
```

The Microsoft Windows handle associated with the application is a supported internal parameter defined as $_HANDLE$.

All internal parameters use $_ as a preface to avoid naming conflicts. Implementer parameters are not allowed to use underscores in key names.

In addition to the Windows handle, the following internal parameters are supported to manage user credentials:

- Username ($_USERNAME$)
- Password ($_PASSWORD$)
- Custom1 ($_CUSTOM1$)
- Custom2 ($_CUSTOM2$)
- Old Password ($_OLDPASSWORD$)

Substitution precedence is defined to preserve parameter values after they are written. The precedence is defined as internal parameters, followed by implementer parameters, followed by environmental variables.

All implementer parameters are permitted to use lowercase and uppercase letters and numbers in key names. Key names are case-insensitive.

If the extension identification executable requires parameters to be presented in a specific sequence, the Argument must support the required sequence. The parameter name/value pairs defined in the Action configuration panel can be in any sequence.

Implementer Requirements

The external process that performs advanced matching or credential management actions is defined as any process or application that can be initiated using a command-line interface. Any required or optional arguments for identification extensions or action extensions must also be able to be specified in-line using a command-line interface.

For action extensions, the implementer must support the same features as previously described for the Windows detection implementation. The Username, Password, Custom1, Custom2, and Old Password credentials can be passed to the executable.

For identification extensions and action extensions, the implementer is responsible for:

- Deploying all executable, support modules and files to support the extension on the Single Sign-on Plug-in.
- Maintaining all deployed modules.
- Adding all the specified registry entries on the plug-in.
- Maintaining extension name uniqueness in their domains.

The recommended extension naming schema is a reverse domain naming schema (that is, com.citrix.cpm.ext4).

Enabling Logging

To activate debug tracing for the Single Sign-on Plug-in, a registry modification must be made.

The registry key name and its associated keys are located at:

[HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\MetaFrame Password Manager\Log]

The following table defines the key value characteristics.

Key	Type	Value
Enabled	REG_DWORD	Default value is 0. 0—disabled. 1—enabled.

Key	Type	Value
Filter	REG_DWORD	Bitmask that dictates what is to be logged. 0x00000001—Windows application flag used to log identification extension errors. 0x00000004—Windows password filling used to log action extension errors.
MaxSizeInBytes	REG_DWORD	Maximum size of the log file in bytes. Maximum theoretical value can be 4GB (2^32). Default—819200

Log file data is recorded in an sso_<username>.log file in:

```
%LocalAppData%\Citrix\MetaFrame Password Manager
```

Virtual Key Codes for Windows, Web and Terminal Emulator-based Applications

May 11, 2015

Single Sign-on supports virtual key codes for Windows, Web and terminal emulator-based applications. These codes are used to send specific keystrokes to logon or password change form fields.

Codes for VTabKeyN (Windows and Web)

Use the following identifiers to create a key code sequence for Windows and Web-based applications.

Code	Description
‘DELAY=N’	N is the number of milliseconds to delay.
‘VKEY=N’	N is the virtual key code to send.

For example, to send a Tab, End, Space, a 1.5 second delay, Logon username, Space, the username/ID, Home, a 0.35 second delay, Tab, and then the password use the following:

```
VTabKey1= ‘VKEY=9’‘VKEY=35’ ‘DELAY=1500 ‘Logon username‘VKEY=32’
VTabKey2=‘VKEY=36’‘DELAY=350’‘VKEY=9’
```

Codes for VirtualKeyCode and VKEY (Windows and Web)

Key	Code	Key	Code	Key	Code	Key	Code
Break	3	5	53	V	86	F5	116
Backspace	8	6	54	W	87	F6	117
Tab	9	7	55	X	88	F7	118
Clear	12	8	56	Y	89	F8	119
Enter	13	9	57	Z	90	F9	120
Shift	16	A	65	Left (window)	91	F10	121
Ctrl	17	B	66	Right (window)	92	F11	122
Alt	18	C	67	NumPad 0	96	F12	123

Key	Code	Key	Code	Key	Code	Key	Code
Caps Lock	20	D	68	NumPad 1	97	F13	124
Esc	27	E	69	NumPad 2	98	F14	125
Spacebar	32	F	70	NumPad 3	99	F15	126
Page Up	33	G	71	NumPad 4	100	F16	127
Page Down	34	H	72	NumPad 5	101	F17	128
End	35	I	73	NumPad 6	102	F18	129
Home	36	J	74	NumPad 7	103	F19	130
Left	37	K	75	NumPad 8	104	F20	131
Up	38	L	76	NumPad 9	105	F21	132
Right	39	M	77	Asterisk(*)	106	F22	133
Down	40	N	78	Plus (+)	107	F23	134
Print Screen	44	O	79	Minus (-)	109	F24	135
Help	47	P	80	Period (.)	110	Num Lock	144
0	48	Q	81	Slash (/)	111	Scroll Lock	145
1	49	R	82	F1	112	Left Shift	160
2	50	S	83	F2	113	Right Shift	161
3	51	T	84	F3	114	Left Ctrl	162
4	52	U	85	F4	115	Right Ctrl	163

Virtual Key Codes for HLLAPI-Compliant Terminal Emulators

Char/Cmd	Code	Char/Cmd	Code	Char/Cmd	Code
Alt Cursor	@$	Local Print	@P	PF12/F12	@c
Backspace	@<	Reset	@R	PF13/F13	@d
@	@@	Shift	@S	PF14/F14	@e
Alt	@A	Dup	@S@x	PF15/F15	@f
Field -	@A@-	Field Mark	@S@y	PF16/F16	@g
Field +	@A@+	Tab (Right)	@T	PF17/F17	@h
Field Exit	@A@E	Cursor Up	@U	PF18/F18	@i
Alt Cursor	@$	Cursor Down	@V	PF19F19	@j
Erase Input	@A@F	Cursor Left	@L	PF20/F20	@k
Sys Request	@A@H	Cursor Right	@Z	PF21/F21	@l
Insert Toggle	@A@I	Page Up	@u	PF22/F22	@m
Cursor Select	@A@J	Page Down	@v	PF23/F23	@n
Attention	@A@Q	End	@q	PF24/F24	@o
Print Screen	@A@T	Home	@0	PA1	@x
Hexadecimal	@A@X	PF1/F1	@1	PA2	@y
Cmd/Func Key	@A@Y	PF2/F2	@2	PA3	@z
Print (PC)	@A@t	PF3/F3	@3	PA4	@+
Back/Left Tab	@B	PF4/F4	@4	PA5	@%
Clear	@C	PF5/F5	@5	PA6	@&

Char/Cmd	Code	Char/Cmd	Code	Char/Cmd	Code
Delete	@D	PF6/F6	@6	PA7	@’
Enter	@E	PF7/F7	@7	PA8	@(
Erase EOF	@F	PF8/F8	@8	PA9	@)
Help	@H	PF9/F9	@9	PA10	@*
Insert	@I	PF10/F10	@a		
New Line	@N	PF11/F11	@b		

Single Sign-on Provisioning Software Development Kit (SDK)

May 11, 2015

The Single Sign-on Provisioning Software Development Kit (SDK) allows you to fully manage users' secondary credentials. Secondary credentials are application-specific credentials that Single Sign-on submits on the user's behalf after the primary domain authentication has occurred.

Credential provisioning allows you to automate many of the tasks associated with managing user credentials. Whether you are rolling out a new installation of Single Sign-on, adding new users, adding new applications, or clearing out unneeded information, credential provisioning provides you with the tools to complete these tasks quickly and efficiently.

This online help describes the high-level design of Single Sign-on's credential provisioning feature and provides a summary of API functions that you can use to define actions in the provisioning XML file.

The Provisioning Module

The Provisioning Module is part of the Single Sign-on Service and is a standard Web Service that exposes a Simple Object Access Protocol and Service Provisioning Markup Language (SOAP/SPML) interface for receiving provisioning commands. All communications between the client and the Provisioning Module occur over a Transport Layer Security (TLS) channel.

When sending provisioning commands to the queue, ensure that the data is stored securely and is not transmitted over an insecure network connection.

The Provisioning Module must have read and write access to the Single Sign-on central store to be able to queue the incoming provisioning commands until the Single Sign-on Plug-in executes the commands.

Commands sent to the Provisioning Module cannot be recalled. Once sent, commands remain queued until they are executed by the SIngle Sign-on Plug-in. If you need to remove a command from the queue, send the opposite command for each user, application, and credential object that must be removed from the queue.

Note: The Provisioning Module uses an interface that conforms to SPML 2.0. Only the core operations required for conformance are supported.

The SPML 2.0 Model

The provisioning XML file and any third party components that issue SPML requests are called requesting authorities (RA).

The Provisioning Module is a Provisioning Service Provider (PSP). This PSP supports a single Provisioning Service Target (PST) that performs per-user queuing of Single Sign-on provisioning commands.

Providing users with secondary credentials is the action that is performed when running provisioning. This means that end users and secondary credentials are the Provisioning Service Objects (PSO) of the provisioning service target. The unique identifier (PSO-ID) for each end user is a fully qualified domain name (FQDN). The unique identifier (PSO-ID) for each secondary credential is the GUID assigned to the credential when it is created. Since the secondary credentials are associated with a particular user, the user PSO acts as a container for the credential PSOs. This is represented by the containerID element in an SPML request.

Strictly speaking, Single Sign-on does not add, modify, or delete users; however, Single Sign-on does add, modify, and delete data associated with a user.

Provisioning and the Single Sign-on Plug-in

Because the Single Sign-on Plug-in is ultimately responsible for protecting a user's secondary credentials with user-specific encryption keys, the execution of provisioning operations is a two-step process. First, you must issue the provisioning command to a Provisioning Module. Then, on behalf of its current user, the Single Sign-on Plug-in applies any queued provisioning commands to the user's secondary credential store.

The Single Sign-on Plug-in detects the presence of queued provisioning operations during its regular synchronization process, which occurs at start up. The plug-in executes queued provisioning commands before resuming normal activity. This ensures that in the First-Time-Use scenario, the plug-in performs the provisioning actions first, thus minimizing the end user's First-Time-Use configuration actions.

All communications between the Single Sign-on Plug-in and Provisioning Module occur over a TLS-secured connection.

The client provisioning application must define the mapping between the applications listed as available for provisioning and the client-side representation of the application.

Provisioning Secondary Credentials

Secondary credentials are associated with a specific application definition that was created using the Single Sign-on component of AppCenter; therefore, the addRequest operation must include data that binds the user details in the request to a specific application definition. This means that the requesting authority must determine the list of applications available for provisioning on a per-user basis, and provide an ID of an application definition as part of the addRequest operation. This burdens the requesting authority with determining the association between the Single Sign-on application definitions and your external identification (such as the application name) of the applications you are provisioning.

Since the person who administers Single Sign-on and the person who performs provisioning tasks may not be the same, there is a potential for "human induced confusion." For example, a Single Sign-on administrator may define the "Microsoft Outlook" application, while a provisioning administrator creates "Microsoft Exchange" accounts. Single Sign-on allows multiple secondary credentials for a given application definition. For example, a user might have multiple MSN Hotmail accounts for which Single Sign-on has stored credentials. This ability means that it is valid for an administrator to issue multiple addRequests with identical parameters. In this case, multiple secondary credentials are created. Similarly, the administrator may want to provision multiple different credentials for the same application; however, the secrets of the credential (user ID, password, and custom fields) are encrypted by Single Sign-on Plug-in and are not recoverable by the Provisioning module to aid the requesting authority in distinguishing the credentials at a later date.

To help address these issues an optional requesting authority private data field, provision description is available in the addRequest and modifyRequest operations. This provides the requesting authority with the ability to add an ID or descriptive data to help it distinguish the credentials. This field is not modified or displayed by the Single Sign-on Plug-in or the Provisioning module. It is retained and returned to the requesting authority when a credential list is requested through a lookupRequest.

The Single Sign-on Plug-in has complete edit-access to all the secondary credentials. This includes actions such as duplicating, deleting, and modifying the credentials. This means that users can change their data so that it no longer matches the state created by provisioning operations.

Also, users can define applications at will, which means they can add credentials for applications not defined in the Single Sign-on console. This ability can lead to ownership issues such as whether or not an administrator can delete or modify a secondary credential or should these credentials be listed in a lookupResponse.This release of Single Sign-on does not support any ownership restrictions; all credentials can be modified by either the administrator or the end user.

Application Groups

Single Sign-on allows you to group applications. An attribute of this grouping is whether or not to use the same password for all credentials defined for applications in the group. Whenever a user edits a credential associated with a group, the change is applied to all credentials of all applications in the group.

This behavior persists when changes are made through the provisioning API. More specifically, if a credential is added to an application group, the new password provided as a parameter to the add command becomes the new password for each application in the group. So, the add command has the net effect of an add and several modify commands. Similarly, a modify command modifies all the applications in a group and therefore has the net effect of several modify commands.

Error Codes

Code	Description
101	One or more required credential fields are missing in provisioning request
102	Invalid user name specified; user name is either missing or format is not correct
103	Specified user not found
104	Invalid application definition; either application definition is missing or has an invalid structure
105	Credential identifier is not in valid format.
106	Specified credential not found.
107	Invalid authorization security token.
108	Unauthorized access token. Specified token is not allowed to perform requested operation.
109	Storage mechanism is accessed by another process. Please try after some time.
110	Error occurred while consuming provisioning commands.
111	User is not authorized to access provision command queue.
112	Error occurred while getting provisioning secret key.
113	Unable to allocate memory for encryption.

Code	Description
114	Unable to allocate entropy data buffer.
115	Encryption failed.
116	Unable to allocate cipherText buffer.
117	Decryption failed.
118	Failed to format windows error code to error message.
119	Pso Id is either missing or not in proper format.
120	Referenced application could not be found.
121	User configuration for requested user could not be found.
122	"join" attribute is missing for credential in password sharing group.
123	"use-new-password" attribute is missing for credential in password sharing group.
124	Password is missing for credential in password sharing group.
125	Credential name is invalid or missing. Please specify valid credential name.
126	Invalid application id specified.
127	Credential cannot rejoin password sharing group.
128	Provisioning is not enabled for the specified user account.

Summary of API Functions

Feb 07, 2011

The API functions provide a method you can use to define actions in the provisioning XML file. In addition to the sample code in these topics, more sample code is available on your product installation media.

All Single Sign-on specific elements and attributes introduced are prefixed with the "ctxs" namespace indicator. The XML snippet in each text box lists both a request and a corresponding response.

Only the synchronous execution mode is supported. Any requests to use asynchronous execution results in unsupportedExecutionMode errors.

For brevity, the following descriptive placeholders are used instead of example values:

Placeholder Text	**Interpretation**
FQDN	The user's Fully Qualified Domain Name.
application-GUID	The GUID assigned to an application definition when it is created using the Single Sign-on component of the Citrix AppCenter.
credential-GUID	The GUID assigned to the provisioned secondary credential by the Provisioning Service upon completion of an addRequest.
RA-generated-ID	A unique ID for a request created by the Requesting Authority. This is used in the requestID optional attribute of the request elements. It is only relevant if support for asynchronous execution is added.
AuthToken	The authentication-token element is mandatory, but is not used at this time.

Provisioning a Single Application - addRequest

Feb 06, 2011

Use the addRequest operation to add credentials to an application for a user.

An addRequest operation requests that a new object (the credential) be added to the specified container object (the user's data store). A containerID (user's fully qualified domain name (FQDN)) must be specified and the psoID (credential GUID) for the newly created object is returned. The data of the request are the specifics of the credential to be created.

If the application definition assigned to the new credential is a member of a password sharing group, then all of the credentials associated with members of that group are updated to use the new password.

Syntax

```
<addRequest requestID='optional-client-generated-ID'
targetID='CPM Provisioning 1.0' returnData='identifier'
executionMode='synchronous'>
<ctxs:authentication-token xmlns:ctxs='http://citrix.com/Provision'>AuthToken
</ctxs:authentication-token>
    <containerID ID='userFQDN'/>
    <data>
        <ctxs:credential xmlns:ctxs='http://citrix.com/Provision'>
            <ctxs:name>Credential name</ctxs:name>
            <ctxs:provision-description>Admin Text</ctxs:provision-description>
            <ctxs:description>Credential description</ctxs:description>
            <ctxs:application>
                <ctxs:id>appdefGuid</ctxs:id>
                <ctxs:group join='true' use-new-password='true'>Domain</ctxs:group>
                <ctxs:fields>
                    <ctxs:userID>salima</ctxs:userID>
                    <ctxs:password>pass123</ctxs:password>
                    <ctxs:custom-field index='1'>domain</ctxs:custom-field>
                    <ctxs:custom-field index='2'>database</ctxs:custom-field>
                </ctxs:fields>
            </ctxs:application>
        </ctxs:credential>
     </data>
 </addRequest>
```

Parameters

requestID (mandatory)	This is the client-generated ID that associates the return values with this request.
targetID (mandatory)	This is the ID of the Provisioning Module, identified with the targetID 'CPM Provisioning 1.0.'
returnData (mandatory)	data — details of a secondary credential identifier — list of credentials for a user

	name — not supported in Single Sign-on everything — application definitions available to the specified user
executionMode (mandatory)	Single Sign-on supports synchronous execution mode.
authentication-token (mandatory)	The authentication-token element is mandatory, but is not used at this time. .
containerID (mandatory)	The containerID provides the FQDN of the user who owns the credential.
data (mandatory)	Data is the description of the data being modified. This is the credential element and may include any child elements of the credential and application elements.
ctxs:credential (mandatory)	The credential element is used to describe a single secondary credential. The name and description children of the credential element are optional. If not provided, the plug-in uses the name and description from the application definition.
ctxs:application (mandatory)	The application element is used both to describe an application definition and to describe details of a credential. The application element must correspond to one previously obtained from a lookupApplicationsRequest operation.

Syntax for Return Values (addResponse)

```
 <addResponse status='success' requestID='client-generated-ID' >
   <pso>
       <psoID ID='credential-GUID'>
          <containerID ID='userFQDN'/>
       </psoID>
       <data/>
   </pso>
</addResponse>
```

Parameters for Return Values (addResponse)

status (mandatory)	Possible values: Success, Failure, Pending
requestID (mandatory)	This is the client-generated ID that associates these return values with the associated request.
pso (mandatory)	The data of the pso is a credential as described in ctxs:credential.
psoID (mandatory)	The psoID is a unique identifier for each end user; PSOID is the credential's GUID returned by the lookupResponse.
containerID	The containerID provides the FQDN of the user who owns the credential.

(mandatory)	
data (mandatory)	Data is the description of the data being modified. This is the credential element and may include any child elements of the credential and application elements.

Group Element Attributes

The join and use-new-password attributes of the group element control how the new credential affects the existing group members. If the application group has not been configured to share passwords, the group element is ignored.

Join value	Use-new-password value	Effect
False	False	The new credential is disassociated from the existing credentials in the group. There is no effect on the existing group.
False	True	The new credential is disassociated from the existing credentials in the group. There is no effect on the existing group.
True	False	The new credential is joined to the existing group. The password of the new credential is set to the password shared by the existing group members. If there are no existing group members, the password value is used.
True	True	The new credential is joined to the existing group. The password included in the command is used for the new credential and also assigned to all of the existing group members.

The credential GUID returned as the psoID in the response is the same one that will be listed in the lookupResponse operation and may also be used to identify this secondary credential in a modifyRequest or deleteRequest operations.

batchRequest - Running a Batch

Feb 07, 2011

The batchRequest operation acts as a container for a list containing other operations (requestnameRequest). Single Sign-on supports sequential processing mode only. A batchRequest that specifies parallel processing does not result in an error, but is processed sequentially.

Syntax

```
<batchRequest processing="sequential" onError="resume">
  <addRequest requestID=&rsquo;client-generated-ID1&rsquo;
  targetID='CPM Provisioning 1.0'
  returnData='identifier' executionMode='synchronous'>
  <ctxs:authentication-token
  xmlns:ctxs='http://citrix.com/Provision'>AuthToken
  </ctxs:authentication-token>
  <containerID ID='userFQDN'/>
  <data>
    <ctxs:credential xmlns:ctxs='http://citrix.com/Provision'>
      <ctxs:name>Credential name</ctxs:name>
      <ctxs:application>
        <ctxs:id>appdefGuid</ctxs:id>
        <ctxs:fields>
          <ctxs:userID>janed</ctxs:userID>
          <ctxs:password>pwd123</ctxs:password>
        </ctxs:fields>
      </ctxs:application>
    </ctxs:credential>
  </data>
</addRequest>

<addRequest requestID=&rsquo;client-generated-ID2'
  targetID='CPM Provisioning 1.0'
  returnData='identifier' executionMode='synchronous'>
  <ctxs:authentication-token
  xmlns:ctxs='http://citrix.com/Provision'>AuthToken
  </ctxs:authentication-token>
  <containerID ID='userFQDN'/>
  <data>
    <ctxs:credential xmlns:ctxs='http://citrix.com/Provision'>
      <ctxs:name>Credential name</ctxs:name>
      <ctxs:application>
        <ctxs:id>appdefGuid2</ctxs:id>
        <ctxs:fields>
          <ctxs:userID>salima</ctxs:userID>
          <ctxs:password>pass123</ctxs:password>
        </ctxs:fields>
      </ctxs:application>
    </ctxs:credential>
  </data>
</addRequest>
</batchRequest>
```

Parameters

processing (mandatory)	This is the processing mode. Valid values are 'sequential' and 'parallel'; however, Single Sign-on supports only sequential mode. When parallel processing mode is specified, Single Sign-on processes the request sequentially.
onError	This is the action you want Single Sign-on to take when there is an error during processing. Valid values are 'resume' and 'exit.'
requestnameRequest (mandatory, variable)	List each request you want to process in this batch, using the syntax and parameters specified for that request.

Syntax for Return Values (batchResponse)

```
<batchResponse>
  <addResponse status="success" requestID="client-generated-ID1">
    <pso>
      <psoID ID='credential-GUID'>
    </pso>
  </addResponse>
  <addResponse status="success" requestID="client-generated-ID1">
    <pso>
      <psoID ID='credential-GUID'>
    </pso>
  </addResponse>
</batchResponse>
```

Parameters for Return Values (batchResponse)

requestnameResponse (variable)	The name of each request that was specified to process in this batch request. For the return values syntax related to each request, refer to the documentation for that request.

Deleting a Credential - deleteRequest

Feb 07, 2011

Use the deleteRequest operation to delete a single credential. The credential GUID specifies the credential to be deleted.

Syntax

```
<deleteRequest requestID="RA-generated-ID"
executionMode="synchronous">
  <ctxs:authentication-token>AuthToken
  </ctxs:authentication-token/>
  <psoId ID='credential-GUID'>
    <containerID ID='userFQDN'/>
  </psoId>
</deleteRequest>
```

Parameters

requestID (mandatory)	This is the client-generated ID that associates the return values with this request.
executionMode (mandatory)	Single Sign-on supports synchronous execution mode.
authentication-token (mandatory)	The auth-token element is mandatory, but is not used at this time.
psoID (mandatory)	The psoID is a unique identifier for each end user; PSOID is the credential's GUID returned by the lookupResponse.
containerID (mandatory)	The containerID provides the FQDN of the user who owns the credential.

Syntax for Return Values

```
<deleteResponse status="success" requestID="RA-generated-ID">
</deleteResponse>
```

Parameters for Return Values

status (mandatory)	Possible values: Success, Failure, Pending
requestID	This is the client-generated ID that associates these return values with the associated request.

Deleting a User - deleteRequest

Feb 06, 2011

Use the deleteRequest operation to remove all data associated with a user from the central store.

Syntax

```
<deleteRequest requestID="RA-generated-ID" executionMode="synchronous">
  <ctxs:authentication-token>AuthToken</ctxs:authentication-token/>
  <psoId ID='userFQDN'/>
</deleteRequest>
```

Parameters

requestID (mandatory)	This is the client-generated ID that associates the return values with this request.
executionMode	Single Sign-on supports synchronous execution mode.
authentication-token	The authentication-token element is mandatory, but is not used at this time.
psoID (mandatory)	The psoID is a unique identifier for each end user; PSOID is the credential's GUID returned by the lookupResponse.

Syntax for Return Values (deleteResponse)

```
<deleteResponse status="success" requestID="RA-generated-ID">
</deleteResponse>
```

Parameters for Return Values

status (mandatory)	Possible values: Success, Failure, Pending
requestID (mandatory)	This is the client-generated ID that associates these return values with the associated request.

Remarks

You may choose to completely remove data associated with specific users when they leave the enterprise. Also, if users forget critical information and are unable to access their credentials, you may choose to reset their Single Sign-on state so they can start over (see resetRequest).

These two scenarios, complete removal of data and data reset, need to be differentiated because the Single Sign-on Plug-in behaves differently in each case. Depending on administrator settings, there may be a local copy of the user's Single Sign-on data in the user's profile. If there is no data for the user in the central store, the plug-in runs a registration wizard and copies the user's local data to the central store.

In the reset user scenario, the plug-in software discards the local data then runs the registration wizard.

Querying for Targets - listTargetsRequest

Feb 07, 2011

The listTargetsRequest operation queries for the targets configured on the system. The Single Sign-on Service supports a single, unique target—the Provisioning Module— identified with the targetID 'CPM Provisioning 1.0.'

Syntax

```
<listTargetsRequest requestID='client-generated-ID'
executionMode='synchronous'>
  <ctxs:authentication-token
  xmlns:ctxs='http://citrix.com/Provision'>AuthToken
  </ctxs:authentication-token>
 </listTargetsRequest>
```

Parameters

requestID (mandatory)	This is the client-generated ID that associates the return values with this request.
executionMode (mandatory)	Single Sign-on supports synchronous execution mode.
authentication-token (mandatory)	The authentication-token element is mandatory, but is not used at this time.

Syntax for Return Values

```
 <listTargetsResponse requestID='client-generated-ID' status='success' >
  <target targetID='CPM Provisioning 1.0'>
    <schema>
      <!-- the schema of the ctxs namespace elements will be here -->
    </schema>
  </target>
</listTargetsResponse>
```

Parameters for Return Values

requestID (mandatory)	This is the client-generated ID that associates these return values with the associated request.
status (mandatory)	Possible values: Success, Failure, Pending
targetID (mandatory)	This is the ID of the Provisioning Module: targetID='CPM Provisioning 1.0'.
schema (mandatory)	The response to this operation contains a unique ID of the module and a schema describing the objects the Provisioning Module manages, such as users and their secondary credentials.

Obtaining a List of Applications Available to a User - lookupApplicationRequest

Feb 07, 2011

Use the lookupApplicationRequest operation to obtain a list of the applications (including their application IDs) available to a specific user. In Single Sign-on, the set of application definitions available to a user is determined by the user configuration associated with the user in the console. These application definitions are not owned by a user and cannot be edited outside of the console.

Syntax

```
<lookupApplicationRequest requestID=RA-generated-ID'
returns='everything' executionMode='synchronous'>
  <ctxs:authentication-token>AuthToken
  </ctxs:authentication-token>
  <psoID ID='userFQDN'/>
</lookupApplicationRequest>
```

Parameters

requestID (mandatory)	This is the client-generated ID that associates the return values with this request.
authentication-token (mandatory)	The authentication-token element is mandatory, but is not used at this time.
psoID (mandatory)	The psoID is a unique identifier for each end user; PSOID is the user's FQDN.

Syntax for Return Values — lookupApplicationResponse

```
<lookupApplicationResponse status="success"
requestID="client-generated-ID">
<pso>
<psoID ID='userFQDN'/>
<data>
  <ctxs:application xmlns:ctxs="http://citrix.com/Provision">
    <ctxs:id>app-GUID1</ctxs:id>
    <ctxs:name>Outlook</ctxs:name>
    <ctxs:description>Outlook 2003</ctxs:description>
    <ctxs:group password-sharing='true'>Domain</ctxs:group>
    <ctxs:fields>
      <ctxs:userID/>
      <ctxs:password/>
      <ctxs:custom-field index='1' label='Domain'/>
      <ctxs:custom-field index='2' label='Exchange Server'/>
    </ctxs:fields>
  </ctxs:application>
  <ctxs:application xmlns:ctxs="http://citrix.com/Provision">
```

```
    <ctxs:id>app-GUID2</ctxs:id>
    <ctxs:name>Vantive</ctxs:name>
    <ctxs:description>Bug Database</ctxs:description>
    <ctxs:group password-sharing='false'>SAP</ctxs:group>
    <ctxs:fields>
      <ctxs:userID/>
      <ctxs:password/>
      <ctxs:custom-field index='1' label='Domain'/>
    </ctxs:fields>
  </ctxs:application>
</data>
</pso>
</lookupApplicationResponse>
```

Parameters for Return Values

status (mandatory)	Possible values: Success, Failure, Pending
requestID (mandatory)	This is the client-generated ID that associates these return values with the associated request.
psoID (mandatory)	The psoID is a unique identifier for each end user; PSOID is the user's FQDN.
data (mandatory)	Data is the description of the data being modified. This is the credential element and may include any child elements of the credential and application elements.
ctxs:application (mandatory)	The application element is used both to describe an application definition and to describe details of a credential. The application element must correspond to one previously obtained from a lookupApplicationRequest operation. There is exactly one application element for each application element for each application definition available in the user configuration. See ctxs:application for more information.

Remarks

A lookup of this type of data is an anomaly not covered in standard SPML semantics. A custom capability is used to obtain the list of application definitions available to a user.

Obtaining a List of Applications for which Credentials are Stored - lookupRequest

Mar 24, 2011

Use the lookupRequest operation to obtain the list of applications for which a user has stored credentials. The value of the returnData attribute determines the level of detail returned.

Syntax

```
<lookupRequest requestID='optional-client-generated-ID'
returnData='identifier' executionMode='synchronous'>
  <ctxs:authentication-token
  xmlns:ctxs='http://citrix.com/Provision'>
  AuthToken</ctxs:authentication-token>
  <psoID ID='userFQDN'/>
</lookupRequest>
```

Parameters

requestID (mandatory)	This is the client-generated ID that associates the return values with this request.
returnData (mandatory)	data - details of a secondary credential identifier - list of credentials for a user name - not supported in Single Sign-on everything - application definitions available to the specified user
executionMode (mandatory)	Single Sign-on supports synchronous execution mode.
authentication-token (mandatory)	The authentication-token element is mandatory, but is not used at this time.
psoID (mandatory)	The psoID is a unique identifier for each end user; PSOID is the credential's GUID returned by the lookupResponse.

Syntax for Return Values - lookupResponse

```
<lookupResponse status='success' requestID='client-generated-ID'>
<pso>
  <psoID ID=ID='userFQDN'/>
  <data>
  <ctxs:user xmlns:ctxs="http://citrix.com/Provision">
  <ctxs:credential ctxs:status="queued" ctxs:pendingAction="modify"
  xmlns:ctxs="http://citrix.com/Provision">
    <ctxs:id>credential-GUID1</ctxs:id>
    <ctxs:name>Aviva</ctxs:name>
    <ctxs:description>Aviva 5250 Demo</ctxs:description>
    <ctxs:provision-description>Aviva 5250
    </ctxs:provision-description>
      <ctxs:application>
        <ctxs:id>app-GUID1</ctxs:id>
        <ctxs:name>Aviva 5250 Demo</ctxs:name>
        <ctxs:group password-sharing='false'>AppGroup
        </ctxs:group>
        <ctxs:fields>
          <ctxs:userID/>
          <ctxs:password/>
        </ctxs:fields>
      </ctxs:application>
    </ctxs:credential>

<!-- Example of a return value for a credential
    added by an end user in Manage Passwords -->

    <ctxs:credential ctxs:status="active"
    xmlns:ctxs="http://citrix.com/Provision">
      <ctxs:id>credential-GUID2</ctxs:id>
      <ctxs:name>Dynamic App1</ctxs:name>
    </ctxs:credential>
  </ctxs:user>
  </data>
  </pso>
</lookupResponse>
```

Parameters for Return Values

status (mandatory)	Possible values: Success, Failure, Pending
requestID (mandatory)	This is the client-generated ID that associates these return values with the associated request.
pso (mandatory)	The data of the pso is a credential as described in ctxs:credential.
psoID (mandatory)	The psoID is a unique identifier for each end user; PSOID is the user's FQDN. According to Single Sign-on's SPML model, the data of the pso is a credential as described in ctxs:credential. This would be included if returnData attribute was set to data or everything. There is exactly one pso element for each secondary credential. The ID attribute of the psoID provides the credential's GUID.
data (mandatory)	Data is the description of the data that was looked up. This is the credential element and may include any child elements of the credential and application elements.
ctxs:credential (mandatory)	The credential element is used to describe a single secondary credential. The name and description children of the credential element are optional. If not provided, the plug-in uses the name and

	description from the application definition. See ctxs:credential for more information.
ctxs:application (mandatory)	The application element is used both to describe an application definition and to describe details of a credential. The application element must correspond to one previously obtained from a lookupApplicationRequest operation. There is exactly one application element for each application definition in the user's user configuration. See ctxs:application for more information.

Remarks

When a lookupRequest operation specifies a credential, the response contains the details of the credential. In general, the secrets of each credential are encrypted by the plug-in software and cannot be accessed by the Provisioning module. That means the character data of the specific field elements is empty for credentials already managed by the plug-in software.

Provisioning is a two-step process. First, the Provisioning Module queues provisioning commands. Next, the plug-in software executes the queued commands. To enable you to verify an action you have just performed, the credential list returned must account for the queued commands. Since the queued commands are protected by the Provisioning Module and not the plug-in software, the Provisioning Module is able to decrypt the command parameters. The credentials that have queued add or modify commands also have the accessible command parameters listed in the lookupResponse operation. Note that the command parameters may include the userID, password, and custom-field values.

Retrieving Secondary Credentials - lookupRequest

Feb 06, 2011

Use this operation to retrieve details of a secondary credential.

Syntax

```
<lookupRequest xmlns:ctxs='http://citrix.com/Provision'
requestID='optional-client-generated-ID'
returnData='data' executionMode='synchronous'>
  <ctxs:authentication-token>AuthToken</ctxs:authentication-token>
  <psoID ID='credential-GUID'>
    <containerID ID='userFQDN'/>
  </psoID>
 </lookupRequest>
```

Parameters

requestID (mandatory)	This is the client-generated ID that associates the return values with this request.
returnData (mandatory)	data - details of a secondary credential identifier - list of credentials for a user name - not supported in Single Sign-on everything - application definitions available to the specified user
executionMode (mandatory)	Only the synchronous execution mode is supported. Any requests to use asynchronous execution results in unsupportedExecutionMode errors.
authentication-token (mandatory)	The authentication-token element is mandatory, but is not used at this time.
psoID (mandatory)	The psoID is a unique identifier for each end user; PSOID is the user's FQDN
containerID (mandatory)	The containerID provides the FQDN of the user who owns the credential.

Syntax for Return Values - lookupResponse

```
<lookupResponse status='success' requestID='xsd:ID optional'>
<pso>
  <psoID ID='credential-GUID'>
    <containerID ID='userFQDN'/>
  </psoID>
  <data>
    <ctxs:credential xmlns:ctxs='http://citrix.com/Provision'
    ctxs:status='queued'>
      <ctxs:name>Credential-name</ctxs:name>
      <ctxs:provision-description>Admin text</ctxs:provision-description>
      <ctxs:description>Credential description</ctxs:description>
      <ctxs:application>
        <ctxs:id>app-GUID</ctxs:id>
        <ctxs:name>Outlook</ctxs:name>
        <ctxs:description>description from app-def</ctxs:description>
        <ctxs:group password-sharing='true'>Domain</ctxs:group>
        <ctxs:fields>
          <ctxs:userID/>
          <ctxs:password/>
        </ctxs:fields>
      </application>
    </ctxs:credential>
  </data>
  </pso>
 </lookupResponse>
```

Parameters for Return Values

status (mandatory)	Possible values: Success, Failure, Pending
requestID (mandatory)	This is the client-generated ID that associates these return values with the associated request.
psoID (mandatory)	The psoID is a unique identifier for each end user; PSOID is the user's FQDN. According to Single Sign-on's SPML model, the data of the pso is a credential as described in ctxs:credential Element. This would be included if returnData attribute was set to data or everything. There is exactly one pso element for each secondary credential. The ID attribute of the psoID provides the credential's GUID.
containerID (mandatory)	The containerID provides the FQDN of the user who owns the credential.
data (mandatory)	Data is the description of the data that was looked up. This is the credential element and may include any child elements of the credential and application elements.
ctxs:credential (mandatory)	The credential element is used to describe a single secondary credential. The name and description children of the credential element are optional. If not provided, the Single Sign-on Plug-in uses the name and description from the application definition. See ctxs:credential Element for more information.
ctxs:application (mandatory)	The application element is used both to describe an application definition and to describe details of a credential. The application element must correspond to one previously obtained from a lookupApplicationRequest operation. There is exactly one application element for each application definition in the user's user configuration. See ctxs:credential Element for more information.

Modifying a Credential - modifyRequest

Apr 22, 2011

Use the modifyRequest operation to change a previously provisioned credential. If the application definition associated with the changed credential is a member of a password sharing group, then all of the credentials associated with members of that group are updated to use the new password.

Syntax

```
<modifyRequest requestID='client-generated-ID'>
<ctxs:authentication-token xmlns:ctxs='http://citrix.com/Provision'>
AuthToken</ctxs:authentication-token>
  <psoID ID='credential-GUID'>
    <containerID ID='userFQDN'/>
  </psoID>
  <modification modificationMode='replace'>
    <data>
    <ctxs:credential xmlns:ctxs='http://citrix.com/Provision'>
    <ctxs:name>New Credential Name</ctxs:name>
    <ctxs:application>
    <ctxs:fields>
      <ctxs:userID>username</ctxs:userID>
      <ctxs:password/>
    </ctxs:fields>
    </ctxs:application>
    </ctxs:credential>
    </data>
  </modification>
</modifyRequest>
```

Parameters

requestID (mandatory)	This is the client-generated ID that associates the return values with this request.
ctxs:authentication-token	The authentication-token element is mandatory, but is not used at this time.
psoID (mandatory)	The credential ID is a GUID (created by the Single Sign-on system and stored in your central store). It must match the value returned by the lookupRequest and is used to locate the credential being modified.
containerID (mandatory)	The containerID provides the FQDN of the user who owns the credential.
modification (mandatory)	modificationMode (optional) add: To add credentials. This produces the same result as an addRequest. If modificationMode is add, the restrictions on the psoID and data elements are the same as for the addRequest. The psoID must only specify a container (as in deleteRequest) and the data must contain a

	credential element (as in addRequest). replace: To replace a field value, put the new value inside the tag. delete: To clear a field value. The contents of the data element are ignored.
data (mandatory)	Data is the description of the data being modified. This is the credential element and may include any child elements of the credential and application elements.
credential (mandatory)	The credential element is used to describe a single secondary credential. The name and description children of the credential element are optional. If not provided, the plug-in uses the name and description from the application definition. See ctx:credential for more information.
name	The name is the application definition name as it appears in your Single Sign-on component of AppCenter.
application (mandatory)	The application element is used both to describe an application definition and to describe details of a credential. The application element must correspond to one previously obtained from a lookupApplicationsRequest operation. See ctxs:application for more information. If an id child of an application is provided, it must match the value stored in the credential.
group	Default values are provided if the group element is not part of the add request. This element describes the relationship between the new credential and existing credentials associated with the group. See the information about Group Element Attributes.
fields (mandatory)	Each child element of fields listed in the lookupResponse operation must be included in the addRequest operation or an error is returned.
userID (mandatory)	userID provides the user's account for this credential.
password (mandatory)	Password provides the user's password associated with this credential.
custom-field	Custom fields provide the custom values for this credential. Single Sign-on supports two custom fields in addition to the user name and password fields.
psoID (mandatory)	The psoID is a unique identifier for each end user; PSOID is the user's FQDN and is used to specify the container for the credential being modified.

Syntax for Return Values - modifyResponse

```
<modifyResponse status='success'requestID='client-generated-ID'>
</modifyResponse>
```

Parameters for Return Values

status (mandatory)	Possible values: Success, Failure, Pending
requestID (mandatory)	This is the client-generated ID that associates these return values with the associated request.

Remarks

The modifyRequest can be used to request that a disassociated credential join the group by setting the attribute join='true' (see addRequest). The group element is subject to the same constraints and has the same effect as described under addRequest.

Note that any of the ctxs:fields sub-elements defined for the application may be included in a modifyRequest. The available fields are listed in the

— *lookupResponse*

.

Group Element Attributes

Join value	Use-new-password value	Effect
False	True	The new credential is disassociated from the existing credentials in the group. There is no effect on the existing group.
True	False	The new credential is joined to the existing group. The password of the new credential is set to the password shared by the existing group members. If there are no existing group members, the password value is used.
True	True	The new credential is joined to the existing group. The password included in the command is used for the new credential and also assigned to all of the existing group members.

The credential GUID returned as the psoID in the response is the same one that will be listed in the lookupResponse operation and may also be used to identify this secondary credential in a modifyRequest or deleteRequest operations.

Resetting a User - resetRequest

Feb 07, 2011

Use the resetRequest operation to reset users' Single Sign-on state when they are unable to access their credentials.

Syntax

```
<resetRequest requestID="RA-generated-ID"
executionMode="synchronous">
  <ctxs:authentication-token>AuthToken</ctxs:authentication-token>
  <psoId ID='userFQDN'/>
</resetRequest>
```

Parameters

requestID (mandatory)	This is the client-generated ID that associates the return values with this request.
executionMode	Single Sign-on supports synchronous execution mode.
authentication-token	The authentication-token element is mandatory, but is not used at this time.
psoID (mandatory)	The psoID is a unique identifier for each end user; PSOID is the user FQDN.

Syntax for Return Values - resetResponse

```
<deleteResponse status="success" requestID="RA-generated-ID">
</deleteResponse>
```

Parameters for Return Values

status (mandatory)	Possible values: Success, Failure, Pending
requestID (mandatory)	This is the client-generated ID that associates these return values with the associated request.

Namespace Elements

May 11, 2015

All Single Sign-on custom elements used in SPML commands are members of the http://citrix.com/Provision namespace. This namespace is also referred to as ctxs prefix. There are three top-level elements in this namespace that occur in SPML commands: authentication-token, application, and credential.

Authentication-Token Element - ctxs:authentication-token

The authentication-token element is used as a container for the authentication token (AuthToken). This element is mandatory, but is not used. There are no child elements of the authentication-token element.

Syntax

```
<ctxs:authentication-token xmlns:ctxs='http://citrix.com/Provision'>
AuthToken
</ctxs:authentication-token>
```

Application Element - ctxs:application

The application element may occur as a top-level element or as a child of the credential element.

The application element is used both to describe an application definition (see lookupApplicationRequest) and to describe details of a credential (see addRequest).

Syntax

```
<ctxs:application xmlns:ctxs='http://citrix.com/Provision'>
  <ctxs:id>app-GUID</ctxs:id>
  <ctxs:name>Outlook</ctxs:name>
  <ctxs:description>description from app-def
  </ctxs:description>
  <ctxs:group password-sharing='true'>Domain</ctxs:group>
  <ctxs:fields>
    <ctxs:userID/>
    <ctxs:password/>
    <ctxs:custom-field index='1' label='Domain'/>
    <ctxs:custom-field index='2' label='Exchange Server'/>
  </ctxs:fields>
</ctxs:application>
```

Note: None of the children of the fields element contain character data in this sample.

Parameters

ctxsID (mandatory)	The GUID assigned to the application definition when it is created in the console
name	The administrator-defined name for the application definition
description	The administrator-defined description for the application definition

group (mandatory if password sharing is used)	The application group this definition is assigned to in the console. The password-sharing attribute is a boolean value used to indicate if this group has been configured to share passwords. For more information, see addRequest.
fields (mandatory)	Lists the data fields to be configured for credentials using this application definition. Any subset of the fields listed may be defined for any particular application definition. Children of the fields element: • userID corresponds to the user id • password corresponds to the user's password • custom-field corresponds to the custom fields that may be included in a definition; the index attribute indicates the particular field (either '1' or '2') and the label attribute contains the optional label text.

See ctxs:credential for an example of an application element as a child of a credential element.

Credential Element - ctxs:credential

The credential element is used to describe a single secondary credential. Most credentials are associated with a particular application definition; this is expressed by a child application element. Credentials that users enter manually do not contain an application element.

Syntax

```
<ctxs:credential xmlns:ctxs='http://citrix.com/Provision'
ctxs:status='available' ctxs:pendingAction='delete'>
  <ctxs:name>Credential Name</ctxs:name>
  <ctxs:description>user visible description
  </ctxs:description>
  <ctxs:provision-description>optional-RA provided-description
  </ctxs:provision-description>
  <ctxs:application>
    <ctxs:id>appdefGuid</ctxs:id>
    <ctxs:fields>
      <ctxs:userID>johnd</ctxs:userID>
      <ctxs:password>pass123</ctxs:password>
      <ctxs:custom-field index='1'>mydomain
      </ctxs:custom-field>
    </ctxs:fields>
  </ctxs:application>
</ctxs:credential>
```

Parameters

status (mandatory)	The status attribute of the credential element indicates the state of this credential from the Single Sign-on Plug-in's perspective. The status is either active or queued. A value of active means that the credential is currently available for the Single Sign-on Plug-in to use. A value of queued means that a

	command to add the credential has been queued but the Single Sign-on Plug-in has not yet processed that command.
pendingAction	The pendingAction attribute of the credential element indicates if there are any queued commands that affects this credential. The pendingAction values are add, modify, and delete. A value of delete indicates that a delete command has been queued for this credential. A value of modify indicates that a modify command has been queued for this credential. This attribute is optional and is omitted if no commands are queued for the credential.
name	The name attribute of the credential element is the value displayed by the Single Sign-on Plug-in in the Manage Passwords window (formerly known as Logon Manager). This value can be edited by the user using the property page of the credential.
description	The description value of the credential element is the value displayed by the Single Sign-on Plug-in in the Manage Passwords window (formerly known as Logon Manager). This value can be edited by the user using the property page of the credential.
provision-description	The provision-description is administrator data that cannot be viewed or edited by the Single Sign-on Plug-in. This is provided solely for the convenience of the Provisioning Administrator.
application	The application element indicates the id of the application definition and the character data for the userID, password, and custom-field elements provides the user's details for this credential.