IBM Security Directory Integrator
Version 7.2.0.1

Password Synchronization Plug-ins

SC27-2708-03

IBM Security Directory Integrator
Version 7.2.0.1

Password Synchronization Plug-ins

SC27-2708-03

Note
Before using this information and the product it supports, read the general information under "Notices" on page 99.

Edition notice

Note: This edition applies to version 7.2.0.1 of *IBM Security Directory Integrator* **licensed program (5724-K74) and to all subsequent releases and modifications until otherwise indicated in new editions.**

Contents

About this publication

This publication contains the information that you require to develop solutions by using components that are part of IBM® Security Directory Integrator.

IBM Security Directory Integrator components are designed for network administrators who are responsible for maintaining user directories and other resources. It is assumed that you have practical experience with installation and usage of both IBM Security Directory Integrator and IBM Security Directory Server.

The information is also intended for users who are responsible for the development, installation, and administration of solutions by usingIBM Security Directory Integrator. The reader must familiar with the concepts and the administration of the systems that the developed solution would connect to. Depending on the solution, these systems might include, but are not limited to, one or more of the following products, systems, and concepts:

- IBM Security Directory Server
- IBM Security Identity Manager
- IBM Java™ runtime environment (JRE) or Oracle Java runtime environment
- Microsoft Active Directory
- Windows and UNIX operating systems
- Security management
- Internet protocols, including HyperText Transfer Protocol (HTTP), HyperText Transfer Protocol Secure (HTTPS) and Transmission Control Protocol/Internet Protocol (TCP/IP)
- Lightweight Directory Access Protocol (LDAP) and directory services
- A supported user registry
- Authentication and authorization concepts
- SAP ABAP Application Server

Access to publications and terminology

Read the descriptions of the IBM Security Directory Integrator Version 7.2.0.1 library and the related publications that you can access online.

This section provides:

- A list of publications in the "IBM Security Directory Integrator library."
- Links to "Online publications" on page vi.
- A link to the "IBM Terminology website" on page vii.

IBM Security Directory Integrator library

The following documents are available in the IBM Security Directory Integrator library:

- *IBM Security Directory Integrator Version 7.2.0.1 Federated Directory Server Administration Guide*

  Contains information about using the Federated Directory Server console to design, implement, and administer data integration solutions. Also contains

information about using the System for Cross-Domain Identity Management (SCIM) protocol and interface for identity management.

- *IBM Security Directory Integrator Version 7.2.0.1 Getting Started Guide*

  Contains a brief tutorial and introduction to IBM Security Directory Integrator. Includes examples to create interaction and hands-on learning of IBM Security Directory Integrator.

- *IBM Security Directory Integrator Version 7.2.0.1 Users Guide*

  Contains information about using IBM Security Directory Integrator. Contains instructions for designing solutions using the Security Directory Integrator designer tool (the Configuration Editor) or running the ready-made solutions from the command line. Also provides information about interfaces, concepts and AssemblyLine creation.

- *IBM Security Directory Integrator Version 7.2.0.1 Installation and Administrator Guide*

  Includes complete information about installing, migrating from a previous version, configuring the logging functionality, and the security model underlying the Remote Server API of IBM Security Directory Integrator. Contains information on how to deploy and manage solutions.

- *IBM Security Directory Integrator Version 7.2.0.1 Reference Guide*

  Contains detailed information about the individual components of IBM Security Directory Integrator: Connectors, Function Components, Parsers, Objects and so forth – the building blocks of the AssemblyLine.

- *IBM Security Directory Integrator Version 7.2.0.1 Problem Determination Guide*

  Provides information about IBM Security Directory Integrator tools, resources, and techniques that can aid in the identification and resolution of problems.

- *IBM Security Directory Integrator Version 7.2.0.1 Message Guide*

  Provides a list of all informational, warning and error messages associated with the IBM Security Directory Integrator.

- *IBM Security Directory Integrator Version 7.2.0.1 Password Synchronization Plug-ins Guide*

  Includes complete information for installing and configuring each of the five IBM Password Synchronization Plug-ins: Windows Password Synchronizer, Sun Directory Server Password Synchronizer, IBM Security Directory Server Password Synchronizer, Domino® Password Synchronizer and Password Synchronizer for UNIX and Linux. Also provides configuration instructions for the LDAP Password Store and JMS Password Store.

- *IBM Security Directory Integrator Version 7.2.0.1 Release Notes*

  Describes new features and late-breaking information about IBM Security Directory Integrator that did not get included in the documentation.

Online publications

IBM posts product publications when the product is released and when the publications are updated at the following locations:

IBM Security Directory Integrator Library
: The product documentation site (http://www-01.ibm.com/support/knowledgecenter/SSCQGF/welcome) displays the welcome page and navigation for this library.

IBM Security Systems Documentation Central
: IBM Security Systems Documentation Central provides an alphabetical list of all IBM Security Systems product libraries and links to the online documentation for specific versions of each product.

IBM Publications Center

The IBM Publications Center site (http://www-05.ibm.com/e-business/linkweb/publications/servlet/pbi.wss) offers customized search functions to help you find all the IBM publications you need.

Related information

Information related to IBM Security Directory Integrator is available at the following locations:

- IBM Security Directory Integrator uses the JNDI client from Oracle. For information about the JNDI client, see the *Java Naming and Directory Interface™ Specification* at http://download.oracle.com/javase/7/docs/technotes/guides/jndi/index.html .
- Information that might help to answer your questions related to IBM Security Directory Integrator can be found at https://www-947.ibm.com/support/entry/myportal/over-accesspubsview/software/security_systems/tivoli_directory_integrator.

IBM Terminology website

The IBM Terminology website consolidates terminology for product libraries in one location. You can access the Terminology website at http://www.ibm.com/software/globalization/terminology.

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

For additional information, see the Accessibility Appendix in *Configuring Directory Integrator*.

Technical training

For technical training information, see the following IBM Education website at http://www.ibm.com/software/tivoli/education.

Support information

IBM Support provides assistance with code-related problems and routine, short duration installation or usage questions. You can directly access the IBM Software Support site at http://www.ibm.com/software/support/probsub.html.

Troubleshooting provides details about:

- What information to collect before contacting IBM Support.
- The various methods for contacting IBM Support.
- How to use IBM Support Assistant.
- Instructions and problem-determination resources to isolate and fix the problem yourself.

Statement of Good Security Practices

IT system security involves protecting systems and information through prevention, detection and response to improper access from within and outside your enterprise. Improper access can result in information being altered, destroyed, misappropriated or misused or can result in damage to or misuse of your systems, including for use in attacks on others. No IT system or product should be considered completely secure and no single product, service or security measure can be completely effective in preventing improper use or access. IBM systems, products and services are designed to be part of a comprehensive security approach, which will necessarily involve additional operational procedures, and may require other systems, products or services to be most effective. IBM DOES NOT WARRANT THAT ANY SYSTEMS, PRODUCTS OR SERVICES ARE IMMUNE FROM, OR WILL MAKE YOUR ENTERPRISE IMMUNE FROM, THE MALICIOUS OR ILLEGAL CONDUCT OF ANY PARTY.

Chapter 1. Introduction to password synchronization plug-ins

The IBM Security Directory Integrator provides the infrastructure and ready-to-use components to implement the solutions that synchronize user passwords in a heterogeneous software environment.

The password synchronization solution, which is built by using IBM Security Directory Integrator, can intercept password changes on a number of software systems. You can direct the intercepted changes into:

- The same software system.
- A different set of software systems.

You can synchronize passwords by using the IBM Security Directory Integrator AssemblyLines. Configure the AssemblyLines to propagate the intercepted passwords to the preferred systems.

The password synchronization solution includes the following components:

Password Synchronizers
Deployed on the system where password changes occur. The Password Synchronizers are responsible for intercepting unencrypted values of the passwords when they change.

Java Proxy process
Receives passwords from the Password Synchronizers and forwards them to a Password Store.

Password Stores
Receive and encrypt the intercepted passwords. Password Stores store the intercepted passwords in the locations where IBM Security Directory Integrator can access them.

Connectors
Connect to the locations where the intercepted and encrypted passwords are stored. You can use the standard or specialized IBM Security Directory Integrator connectors to retrieve and decrypt the passwords.

AssemblyLines
Use the connectors to get the intercepted passwords and then build the custom logic to send passwords to other software systems.

Solution building

You must configure and deploy the ready-to-use components such as Password Synchronizers, Password Stores, and Connectors to implement the password synchronization solution. The solution intercepts the passwords and makes them accessible from IBM Security Directory Integrator.

Implement a custom AssemblyLine for the solution, which consolidates passwords that are intercepted from different sources and feeds them into the systems to be synchronized. Design of the AssemblyLine depends on the custom environment and the specific solution requirements. The IBM Security Directory Integrator does not include these customized AssemblyLines. You must implement the AssemblyLines.

Password synchronization AssemblyLine uses the Iterator Connector to retrieve the passwords from the password stores. Then, the AssemblyLine uses other standard IBM Security Directory Integrator connectors to set these passwords into other systems. If the synchronized systems have custom requirements to set the passwords, address these requirements in the AssemblyLine and in the connectors that set these passwords. Such customization requires you to set certain connector parameters. For example, you must turn on the **Auto Map AD Password** option in the LDAP Connector to set user passwords in the Active Directory. For more complex cases, scripting is necessary.

To automate the synchronization process, the password synchronization solution includes the IBM Security Directory Integrator AssemblyLines with connectors in `Server Mode`. For example:

- An AssemblyLine listens for changes in the repository where a password store component stores the intercepted passwords. The AssemblyLine triggers the synchronization AssemblyLine whenever a new password is intercepted.
- Using an AssemblyLine with a Timer loop that starts the synchronization AssemblyLine on a schedule.

Each of the components of Password Synchronizer provides interfaces that you can use to tune the behavior. You can also combine various components with each other to create the custom solutions. These key features provide flexibility to build solutions that meet custom requirements and limitations. The password synchronization suite consists of specialized components that intercept the passwords and make them accessible for IBM Security Directory Integrator. The IBM Security Directory Integrator can access the intercepted passwords through its connectors. You can use the flexibility and openness of the architecture to organize the password retrieval process and propagation to other systems.

Password Synchronizer deployment limitation

For password synchronization that involves Chapter 5, “Sun Directory Server Password Synchronizer,” on page 29 and Chapter 6, “IBM Security Directory Server Password Synchronizer,” on page 35, use simpler methods to deploy the synchronizers.

Deploy the Password Synchronizer only if hashed password values are unusable outside the directory. The IBM Security Directory Server and the Sun Directory Server support password encryption where the password values are encrypted before they are stored in the directory. Password encryption uses either a one-way or a two-way cryptographic transformation. One-way transformations, for example, hashing with `SHA-1` or `MD-5`, are not reversible. You cannot obtain the plaintext value from the one-way encrypted password. The Password Synchronizer catches the plaintext password before it is hashed and stored in the directory. If hashed values are used by the destination repository, the synchronization is achieved through LDAP. For example, when both of the source and the destination systems support the same hashing schemes.

You do not require a Password Synchronizer to synchronize the passwords between instances of the IBM Security Directory Server and the Sun Directory Server. Both the products support the same set of hashing algorithms for the passwords. In such cases, you copy the passwords between the two instances through LDAP. Alternatively, if you are required to authenticate against IBM Security Directory Server with the credentials stored in Sun Directory Server, use the `pass-through` authentication option.

Issues with IBM Security Directory Server replication

In a replication topology, deploy the Password Synchronizers on all the master instances. When you configure the replication, changes are propagated to replication consumers through the LDAP operations. If a Password Synchronizer is deployed on a consumer, it intercepts LDAP operations that are triggered by the replication. If the password synchronizer rejects a password that originates from replication, the replication fails. To avoid such a situation, deploy the Password Synchronizers on all the replication masters to reject passwords before they are saved into the directory.

When passwords are set on the supplier node in a replication topology, the synchronizers on the associated consumer nodes synchronize password value to the Password Store. As a result, the same password is sent to the password store multiple times. To avoid this condition, configure IBM Security Directory Server to use the hashed passwords. Password Synchronizers ignore hashed passwords. Therefore, Password Synchronizers on consumers ignore the already hashed password value, which is received from the replication supplier.

Hashed passwords

Password Synchronizer ignores the hashed password values and only the plaintext passwords are synchronized. The Password Synchronizer receives hashed passwords in the following cases:

- If an LDAP client sends a password value that is already hashed, the IBM Security Directory Server accepts it. However, the Password Synchronizer cannot obtain a plaintext password and ignores it. For example, if an LDAP client sends `{SHA}5yfRRkrhJDbomacm2lsvEdg4GyY=` instead of `mypass`, the Password Synchronizer does not send password to the Password Store.
- If the password encryption is set to one-way transformation, for example, `crypt`, `MD5`, `SHA-1`, passwords are stored in hashed form in the directory.

Specialized components

You can use the various specialized components that are available with IBM Security Directory Integrator to build and implement the password synchronization solution.

Password Synchronizers

Password Synchronizer for Windows
Intercepts the Windows login password changes. See Chapter 4, “Windows Password Synchronizer,” on page 17.

Password Synchronizer for Sun Directory Server
Intercepts the Sun Directory Server password changes. See Chapter 5, “Sun Directory Server Password Synchronizer,” on page 29.

Password Synchronizer for IBM Security Directory Server
Intercepts the IBM Security Directory Server password changes. See Chapter 6, “IBM Security Directory Server Password Synchronizer,” on page 35.

Password Synchronizer for Domino
Intercepts changes of the HTTP password for Lotus® Notes® users. See Chapter 7, “Domino HTTP Password Synchronizer,” on page 39.

Password Synchronizer for UNIX and Linux
Intercepts changes of the UNIX and Linux user passwords where Pluggable Authentication Module (PAM) is enabled. See Chapter 8, "Password Synchronizer for UNIX and Linux," on page 63.

Password Stores

LDAP Password Store
Provides function to store the intercepted user passwords in the LDAP directory servers. See Chapter 9, "LDAP Password Store," on page 67.

JMS Password Store
Formerly known as IBM WebSphere® MQ Everyplace® Password Store. The JMS Password Store stores the intercepted user passwords in a JMS Provider Queue from where any JMS client can read them. For example, IBM Security Directory Integrator. See Chapter 10, "JMS Password Store," on page 75.

Log Password Store
Logs any action that is taken by a normal password store. This password store is useful to verify whether the Java Proxy and the plug-ins are communicating correctly.

Specialized connectors

JMS Password Store Connector
Retrieves password update messages from a JMS Password Store, and sends them to IBM Security Directory Integrator. See *Reference*, and Chapter 10, "JMS Password Store," on page 75 for more details.

IBM Security Identity Manager integration

You can integrate the IBM Security Identity Manager and the following Password Synchronizers:

- Sun Directory Server Password Synchronizer
- IBM Security Directory Server Password Synchronizer
- Windows Password Synchronizer
- Password Synchronizer for UNIX and Linux

For more information, see Chapter 13, "IBM Security Identity Manager integration," on page 91.

Password synchronization architecture and workflow

The IBM Security Directory Integrator password synchronizer architecture consists of four layers, which you can combine to build the required password synchronization solutions.

There are several layers in the IBM Security Directory Integrator Password Synchronizer architecture as shown in the following picture.

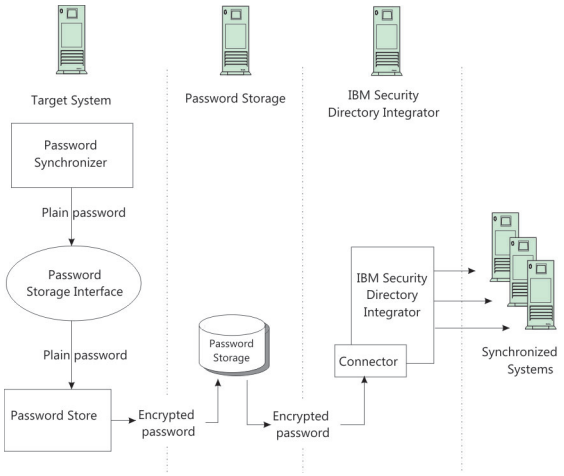

Figure 1. IBM Security Directory Integrator password synchronizer architecture

The `Target System` in the diagram depicts the software system where you want to intercept password changes. The `Password Synchronizer` component hooks into the `Target System` with custom interfaces provided by the `Target System`. The `Password Synchronizer` component intercepts password changes as they occur in the `Target System` and before the password is hashed irreversibly.

The Java Proxy component is a proxy, which receives passwords from the server plug-in and redirects them to the `Password Storage` component. The proxy acts as a container for the `Password Storage` component. This component manages the lifecycle of the `Password Storage` component and handles inter-process communication with the IBM Security Directory Integrator plug-ins.

The Java Proxy logs errors, if any, in the configured log file. If an initialization error is raised, the Java Proxy fails to load it. If a run time error occurs, the error is logged for later investigation. However, the server continues to run and thus provides high availability in a temporary environment change or failure.

The `Password Storage` component is deployed on the `Target System`. When the Password Synchronizer intercepts a password change, it sends the password to the `Password Store` by using the Java Proxy process. The `Password Store` encrypts the password and sends it to a `Password Storage`.

The `Password Storage` component is the second layer in the architecture. It represents a persistent storage system, for example, an LDAP directory, or the IBM WebSphere MQ Everyplace. In the storage system, the intercepted and the encrypted passwords are stored in a form and location that are accessible from IBM Security Directory Integrator. You can have the `Password Storage` on the `Target System` or on another network system.

The IBM Security Directory Integrator, the third layer of the architecture, uses a `Connector` component to connect to the `Password Storage` and retrieves the stored passwords. In the IBM Security Directory Integrator, the passwords are decrypted and made available to the AssemblyLine. The AssemblyLine synchronizes passwords with other systems. You can deploy the IBM Security Directory Integrator on a system different from the `Target System` and the`Password Storage` systems.

The systems where passwords are synchronized with the Target System represent the next layer in the architecture, in the data flow direction. The password synchronization AssemblyLine is responsible to connect to these systems and to update the passwords.

Java Proxy process authentication

Java Proxy receives passwords from the Password Synchronizers and redirects them to the Password Store component. The Java Proxy manages lifecycle of the Password Storage component and handles the inter-process communication with IBM Security Directory Integrator plug-ins.

The proxy and the directory plug-in share a common binary command protocol. The communication happens over the sockets. The proxy acts as a server, listening for commands. The directory plug-in connects to the proxy, transmits a command, and reads the response.

Depending on the configuration, the Java Proxy can also do a preliminary validation on the password strength. You can validate the password policies that are defined only in a remote IBM Security Identity Manager server. The Java Proxy is responsible for storing password changes that are received by the plug-in in the configured Password Store.

The communication between the various plug-ins and the Java Proxy happens over sockets. It is restricted only to the loopback network interface. A two-way authentication takes place each time a connection between the client plug-in and the Java Proxy is established. Authentication is based on the file system permissions. The authentication procedure uses the Authentication Folder, the place where the pwsync.props file is located. You must protect the *Authentication Folder* with file system permissions because the authentication process creates one-time-passwords and stores them as files in the folder.

You must secure the Authentication Folder after the password synchronizer is set up. To secure the folder, make it readable or writable only by the user who runs the process by loading the plug-in. For example, for the Domino HTTP Password Synchronizer, the user notes runs the Domino Server. The user must have full control over the Authentication Folder for the Password Synchronizer to work.

Note: The Java Proxy process automatically starts from the plug-in side and thus is run with the same privileges as the plug-in. If the Java Proxy is started manually by another user, you must grant the read and write access to the Authentication Folder. For example, if the user has full control over the *Authentication Folder*, run the commands with the privileges of that user for the authentication.

On Windows

```
runas /user:user startProxy.bat configuration_file
runas /user:user stopProxy.bat configuration_file
```

On Linux and UNIX

```
su - user
startProxy.sh configuration_file
stopProxy.sh configuration_file
```

Restricting file access on Windows

You must restrict access to the Authentication Folder for the Windows Password Synchronizer. Grant folder access only to the administrator group.

About this task

The Windows Password Synchronizer plug-in runs in the Local Security Authority (LSA) process owned by the local system account. Since the user account is part of the administrators' group, grant access only to that group. If you are setting up a different password synchronizer, grant the required privileges to the appropriate user or the group.

Procedure

1. In Windows Explorer, right-click the `Authentication Folder`.
2. From the menu, select **Properties**.
3. In the Properties window, click the **Security** tab.
4. Click **Advanced**.
5. Clear the check box that allows propagation of parent permissions and select the check box that replaces all the child permissions.
6. Remove all the records from the **Permission entries** list.
7. Click **Add**.
8. Add the administrators group and grant full control access.
9. Click **OK**.

Restricting file access on Linux and UNIX

You must restrict access to the `Authentication Folder` for the PAM Password Synchronizer. The authentication folder contains the `pwsync.props` plug-in configuration file.

Procedure

1. Change the ownership of the folder.

   ```
   chown -R root:root auth_dir
   ```

   The authentication process takes place in the *auth_dir* folder.
2. Change the permissions for the folder.

   ```
   chmod -R 700 auth_dir
   ```

Password store interface

Password Store is the place where Java Proxy stores the received passwords.

You can change the Password Store that is used by a Password Synchronizer when necessary. For example, a Password Synchronizer for IBM Security Directory Server is deployed and configured to use the LDAP Password Store. If you want to use the JMS Password Store, configure the Password Store, change a single property of the Password Synchronizer, and restart the IBM Security Directory Server. New password changes are stored in your designated JMS Password Store. Installation of the synchronization solution again is not required.

Architecture options

You can build a password synchronization solution to intercept password changes on several `Target Systems` by using a layered password synchronization architecture.

The layered password synchronization architecture brings the following values in terms of scalability and customization options:

- You can configure the Password Store components of several Target Systems to store the intercepted passwords in the same Password Storage. The IBM Security Directory Integrator AssemblyLine uses a single connector to connect to the Password Storage. The AssemblyLine is not affected by the number of Target Systems whose passwords are intercepted and stored in this Password Storage.
- You can configure the AssemblyLine to connect to many Password Storages with several Iterator Connectors. This configuration is useful when you use different Password Storages, or distinction of the Target Systems on the IBM Security Directory Integrator is necessary.

In either or both of these previous approaches, you can add, remove, or change Target Systems in an existing solution by focusing on the new function. Rest of the solution is not affected.

In the data flow, where the passwords are updated in the systems that you want to keep synchronized, the password synchronization architecture benefits from the inherent scalability of the IBM Security Directory Integrator. Updating passwords on yet another system is like adding a connector in the password synchronization AssemblyLine.

When the Target System is also one of the systems that is updated, with the intercepted passwords from other systems, you must avoid circular updates. To implement the solution on the IBM Security Directory Integrator, you must build the logic that does not update a system with passwords intercepted on the same system.

Security

You can use the public-private key infrastructure to provide secure transport and intermediate storage of password data.

The Password Store components use a public key to encrypt password data before you send it on the wire and storing it in the Password Storage. The IBM Security Directory Integrator AssemblyLine or specialized connectors have the corresponding private key. This key is used to decrypt password data that are retrieved from the Password Storage.

The Password Store components, which support SSL, add a layer of security.

Set the appropriate file system permissions to protect the installation folder of each password synchronizer and its files against non-trusted users on the host operating system. Restrict read, write, or execute access to the installation folder or the files of the Password Synchronizer for non-trusted users and groups.

Reliability

Functions for preventing and dealing with possible password de-synchronization is built into the password synchronization workflow.

The Password Synchronizer and the Password Store components together provide functions to deal with the conditions where an external storage system is not available or malfunctions.

The Password Store reports to the Password Synchronizer to check whether the password was successfully stored in the Password Storage. The Password Synchronizer component can use the following techniques to prevent or handle possible password de-synchronizations:

- The Password Synchronizer, when enabled, cancels the password change in the Target System. The cancellation occurs when the Password Store reports that the password is not stored in the Password Storage because of non-availability or other reasons.
- Failure is logged when the Target System cannot enable the cancellation or rollback of the password change, which is required on unsuccessful storage. The failure is logged with user information whose password is not stored in the Password Storage. An administrator can inspect the log and resolve de-synchronized passwords.

Chapter 2. Installation of password synchronization plug-ins

You must install the password synchronization plug-ins by using the standard IBM Security Directory Integrator installer.

On Windows platform, the user ID to install the IBM Security Directory Integrator password synchronization plug-ins must be that of the administrator or a member of the administrators' group. On UNIX platform, the installer requires that the user must be the root user. The installer fails if the user does not have these privileges.

For more information about platform requirements for each password synchronization plug-in, see the "Supported Platforms" section in each of the password synchronization plug-in section.

You can install the IBM Security Directory Integrator password synchronization plug-ins by using the standard IBM Security Directory Integrator installer. See *Installing and Administering* for the installation instructions.

Instead of installing the main product, you must choose a custom installation, and select the **Password Synchronization plug-ins** option.

1. Start the installer either from the command line, or double-click the installer (on Windows).
2. Select the installation directory for password synchronization plug-ins.
3. Select **Custom**.
4. Select **Password Synchronization Plug-ins**. All the plug-ins are installed in the *TDI_install_dir*/pwd_plugins directory.

After the installation of password synchronization plug-ins, you must complete several post-installation tasks.

Upgrading and migrating password synchronization plug-ins

You must save the existing configuration files of the password synchronization plug-ins before you install the new version.

About this task

An upgrade or migration of the existing plug-in configuration files is not possible. You must uninstall the existing version and install the newer version of password synchronization plug-ins.

Procedure

1. Save the following existing configuration files:
   - Windows Password Synchronizer registry entry HKEY_LOCAL_MACHINE\SOFTWARE\IBM\<*previous Security Directory Integrator version* >\Windows Password Synchronizer
   - Password Store configuration files. For example, mqepwstore.props or idipwstore.props
   - The IBM WebSphere MQ Everyplace Password Store configuration file. For example, mqeconfig.props
2. Uninstall the existing version.

a. Go to the IBM Security Directory Integrator password synchronization plug-ins `_uninst` directory.
b. Run the uninstaller.

Platform	Executable
Windows	`uninstall.exe`
All other platforms	`uninstall.bin`

c. Restart the system.

3. Install the new version of password synchronization plug-ins.
4. Restore the configuration files from Step 1.

Chapter 3. Common configuration and utilities of password synchronization plug-ins

The password synchronization plug-ins and the Java Proxy share the `pwsync.props` configuration file. You can use the command-line utilities to control the configuration of Password Synchronizers and the data flow process.

Configuration file parameters

You must specify the path to the `pwsync.props` configuration file when you register the plug-in. Configuration file path is then passed to the Java Proxy on startup by the plug-in, or by the command-line utility that starts the proxy.

Note: The standard `java.util.Properties` class parses the configuration file and replaces control-like characters with actual control characters. For example, the `\\n` character is converted to `\n` character. Therefore, when you set a path in the configuration file on the Windows platform, you must set the `\` character with another slash `\\`.

Common parameters for all password plug-ins in the configuration file are as follows:

proxyStartExe
: This string parameter holds the path for an executable file, binary or shell script, and is used to start the Java Proxy. The default value is *TDI_install_dir*`/pwd_plugins/bin/startProxy.bat(sh)`.

  Note: The password plug-in automatically starts the Java Proxy if it is not already running. Comment out the **proxyStartExe** parameter to manually control the Java Proxy startup. The password plug-in rejects all the password changes if Java Proxy is not running.

serverPort
: This integer property specifies the port number that the Java proxy listens to. This property is read by the client plug-in to establish a connection to the Java Proxy. The default value is 18001.

logFile
: This string parameter configures the log file of the client plug-in. If this parameter is not set, logging is not possible.

  Note: The PAM plug-in logs use the UNIX syslog daemon and do not use this property.

checkRepository
: This Boolean property enables turning on or off of the function that checks for availability of the `Password Storage`.

  When this property is set to true, the `Password Synchronizer` checks whether the `Password Storage` is available. If available, the password is changed in the directory, and then the password is sent to the `Password Storage`. If the check indicates that the storage is not available, the LDAP operation, which is the password update, is rejected on the `Target System`.

  When the **checkRepository** property is set to false, the `Password Synchronizer` does not check for the storage availability. The password

update is made in the directory, and then stored in the Password Storage. If the password cannot be stored, a message is logged in the log file (pointed to the **logFile** property) to indicate the password synchronization failure.

The default value is true.

Note: The check for availability of the Password Storage works with all the Password Store components.

syncClass
This required property defines the full name of the Java class of the Password Store component. The default value is com.ibm.di.plugin.pwstore.log.LogPasswordStore. The available parameters are:

- **com.ibm.di.plugin.pwstore.log.LogPasswordStore**
- **com.ibm.di.plugin.pwstore.jms.JMSPasswordStore**
- **com.ibm.di.plugin.pwstore.ldap.LDAPPasswordStore**

javaLogFile
This string parameter configures the log file of the Java Proxy. If this parameter is not set, logging is not possible.

customData
This parameter specifies the custom string that is sent with each password change. Use this parameter to uniquely identify the system or the application that generates changes. For example, system IP, application name, and version.

Note: The Java Proxy sends the same custom data for every password change it processes.

debug This Boolean property turns on or off the debugging. Both the client plug-in and the Java Proxy check this property. The default value is true.

ProxyRetryAttempt
This property is used to specify the number of retry attempts to made before timeout. The default value is 15.

This property is available from IBM Security Directory Integrator Version 7.2 onwards.

The parameters from the configuration file are set as Java system properties. Set the following properties in the configuration file if SSL is required for the communication with any of the stores or with the IBM Security Identity Manager servlet:

Table 1. SSL Java Properties

Property	Value
javax.net.ssl.trustStore	Specifies the truststore for the JVM.
javax.net.ssl.trustStorePassword	Specifies the password of the truststore. **Note:** You must encrypt this password by using the encryptPasswd utility.
javax.net.ssl.trustStoreType	Type of the truststore. For example: jks
javax.net.ssl.keyStore	Specifies the keystore of the JVM.

Table 1. SSL Java Properties (continued)

Property	Value
javax.net.ssl.keyStorePassword	Specifies the password for the keystore. **Note:** You must encrypt this password by using the `encryptPasswd` utility.
javax.net.ssl.keyStoreType	Type of the keystore. For example: `jks`

Any additional parameters in the configuration file are specific to the actual password plug-in.

Command-line utilities

The following utilities are available to control certain aspects of configuration and flow process of the Password Synchronizers:

TDI_install_dir/**pwd_plugins/bin/encryptPasswd.bat(sh)**
Encrypts passwords before you set them in the various configuration files.

Note: This utility uses a symmetric algorithm to encrypt the passwords. The passwords can be easily decrypted by a skilled user. Make sure that you allow the reading of the configuration files only by the trusted users.

TDI_install_dir/**pwd_plugins/bin/startProxy.bat(sh)**
Starts the Java Proxy manually. This utility automatically searches for the default `jars` folder and creates the class path of the Java Proxy. The default folder is *TDI_install_dir*`/pwd_plugins/jars/`. For example, if you configure the JMS Password Store to work with the IBM WebSphere MQ, add the required IBM WebSphere MQ JAR files to the `pwd_plugins/jars/` folder before you start the Java Proxy.

TDI_install_dir/**pwd_plugins/bin/stopProxy.bat(sh)**
Sends a stop request to the running Java Proxy process. The Java proxy waits until all operations are complete and then exits normally.

When a task, which is calling one of the Password Synchronizers is shut down, the Java Proxy process is not automatically terminated. The Password Synchronizer connects to the proxy process if it is already running and therefore termination of the proxy is not required.

*TDI_install_dir***pwd_plugins\windows\pwsync_admin.exe**

Starts or stops the Java Proxy and you can also use this utility to pause or resume the Windows plug-in. This utility is for the 32-bit version. For a Windows 64-bit installation, use the `pwsync_admin_64.exe` file.

*TDI_install_dir***jvm\jre\bin\keytool and** *TDI_install_dir***jvm\jre\bin\ikeyman**
Manages the keystore/truststore that are used during the plug-ins setup. For more information, see the "Keystore and truststore management" topic in the *Installing and Administering*.

Java Proxy with IBM Tivoli Monitoring

You can use the Agent Management Services of the IBM Tivoli® Monitoring Version 6.2.2, Fix Pack 2.0 to manage the Java Proxy process of password synchronizers.

The monitoring behavior of the Agent Management Services towards a particular agent is governed by the settings in an XML-based policy file. This policy file is referred to as a Common Agent Package (CAP) file. These services are available in

the IBM Tivoli Monitoring OS Monitoring Agent for the Windows, Linux, and UNIX platforms. The Agent Management Services are designed to keep the Java Proxy process available and to provide the information about its status to the Tivoli Enterprise Portal. For more information about Agent Management Services, see http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/topic/com.ibm.itm.doc_6.2.2/itm_agentmgmtsvcs_intro.htm.

Managing the Java Proxy with IBM Tivoli Monitoring is optional.

Each Password Synchronizer has an associated CAP file that describes its Java Proxy process. After the installation, all the CAP files are available in the *TDI_install_dir*/pwd_plugins/cap/ directory. The following table lists the available CAP files:

Password Synchronizer	**CAP File**
Windows Password Synchronizer:	tdi_ad_plugin_default.xml
IBM Security Directory Server Password Synchronizer	tdi_tds_plugin_default.xml
Sun Directory Server Password Synchronizer	tdi_sun_plugin_default.xml
PAM Password Synchronizer	tdi_pam_plugin_default.xml
Domino HTTP Password Synchronizer	tdi_domino_plugin_default.xml

You must copy the appropriate CAP file to the correct directory for IBM Tivoli Monitoring to recognize that the proxy is available to be managed.

On UNIX or Linux, the directory is: /opt/IBM/CAP

On Windows, the directory is: %ALLUSERSPROFILE%\ApplicationData\IBM\CAP

Before you can use the CAP files, modify them to contain the correct path to the installation of IBM Security Directory Integrator.

Chapter 4. Windows Password Synchronizer

The Windows Password Synchronizer intercepts password changes of user accounts on the Windows operating systems.

Overview

Password changes are intercepted in all of the following cases:

- When a user changes the password through the Windows user interface.
- When an administrator changes the password of a user through the Windows administrative user interface.
- When a password change request to the Active Directory is made through LDAP.

The IBM Security Directory Integrator password synchronizer plug-in propagates the changes to a repository such as password store before the Windows system changes the password.

The IBM Security Directory Integrator Password Synchronizer stores the user password in a password store such as LDAP server or JMS Password Store.

The change is later propagated to other servers by the IBM Security Directory Integrator AssemblyLine. After the password is stored, control is returned to the Windows system and you can modify the user password.

Synchronization from a single system

To synchronize passwords from a single system, install the Windows Password Synchronizer on the Windows stand-alone system.

Synchronization from a Windows 2008 domain

Password Synchronizer can synchronize password changes from the Windows 2008 domain. You must install the Password Synchronizer on all domain controllers to synchronize.

Sample scenario

Bob logs on to the Windows systems, presses **Ctrl**+**Alt**+**Delete**, and requests a password change. The password change is intercepted by the Windows Password Synchronizer, then delegated to the associated Password Store such as LDAP Password Store or JMS Password Store. If the Password Store confirms that the password was successfully stored, the password change takes place on the local Windows system. The changes can happen on a stand-alone systems or a domain controller. If the Password Store indicates that the password was not stored, then the password change on the local Windows systems is denied.

The password change requests to the Active Directory through LDAP and JNDI are also intercepted and handled by the Windows Password Synchronizer.

Windows Password Synchronizer workflow

The Windows Password Synchronizer intercepts a password change before the change is committed internally by the Windows and the Active Directory. The Password Synchronizer passes the new password to the Password Store.

If the Password Store indicates that the password is stored successfully, the Password Synchronizer enables the password change to be committed in the Windows system.

If the Password Store indicates that the password is not stored, the password change is rejected on the Windows system. If you make the password change from the Windows user interface, the following error message is displayed:

```
Windows cannot complete the password change for user_name because:
The password does not meet the password policy requirements.
Check the minimum password length, password complexity and password history
requirements.
```

The log files of the Password Synchronizer and the Password Store component indicate the reason why the password cannot be stored in the Password Storage.

On each successful password change, the Password Synchronizer sends the full name of the user and the **displayName** attribute from Active Directory to the Password Store. The JMS Password Store ignores this data. The LDAP Password Store writes more information to the extended data attribute of the user entry. By default, the extended data attribute is named as **ibm-diExtendedData**.

The Password Synchronizer returns the **sAMAccountName** attribute of the user after the password is changed. This name is unique for each Windows domain but it is not unique for the Domain Forest Model. To retrieve the rest of the user attributes, more lookups are required through the specified **sAMAccountName** attribute as the link criteria.

Windows Password Synchronizer filtering

The Windows Password Synchronizer provides filtering feature. Filtering affects only when a password change is sent to the password store and not when the Windows domain accepts or rejects the password change. If the user filter accepts a user, the password changes for that user are sent to the Password Store.

The user filter works based on the following two criteria:

- Group membership
- DN matching (LDAP subtree location matching)

Group membership indicates whether a user is a member of some Windows group. The user filter does not recognize nested groups. If a user is a member of group A, which is nested into group B, then the user cannot be a deemed member of group B.

DN matching deals with whether a DN suffix matches the distinguished name of a user. For example, if the user has a distinguished name `cn=myuser,ou=myou,dc=mydc,dc=com`, it is matched by the DN suffix `dc=mydc,dc=com` but not by `dc=mydc`.

The user filter allows `include` and `exclude` rules of both group membership and DN matching. For example, you can configure the user filter to accept all users

who are the members of a certain Windows group (include form). However, the users are not the members of other Windows group (exclude form).

You can combine group membership and DN matching in both rules, include or exclude. However, there is a limitation. The exclude rules always have higher priority than include rules. For example, if a user is included by DN matching but excluded by group membership, the user filter cannot accept the user.

To preserve compatibility with an earlier version, if no include form is specified for group membership and DN matching, the default form is include all. If no exclude form is specified for group membership and DN matching, the default form is exclude none.

Examples of filtering mechanism:

- If no configuration is provided to the user filter, it accepts all users (compatibility with an earlier version).
- If the user filter is provided with some include rules and no exclude rules, it accepts users who are matched by the specified include rules.
- If the user filter is provided with some exclude rules and no include rules, it accepts users who are not matched by any of the exclude rules.
- If the user filter is provided with some include rules and some exclude rules, it accepts users:
  - Matched by some of the include rules
  - Not matched by any of the exclude rules

The user filter of the Windows Password Synchronizer is configured by using the following four string values in the plug-in configuration file to define the include and exclude rules.

includeGroups
: A list of Windows groups. If the user is a member of a group on the list, the filter accepts the user. The assumption is that the user is not excluded by some of the **exclude** lists.

excludeGroups
: A list of Windows groups. If the user is a member of a group on the list, the filter cannot accept the user.

includeDNs
: A list of DN suffixes. If the distinguished name of a user matches some suffix on the list, the filter accepts the user. The assumption is that the user is not excluded by some of the **exclude** lists.

excludeDNs
: A list of DN suffixes. If the distinguished name of the user matches some suffix on the list, the filter cannot accept the user.

All of the above property string values must be lists with tokens separated by semicolons. Redundant white spaces are not allowed. The group lists must include only names of the existing Windows groups. Matching of a DN suffix against a distinguished name is achieved by a non-case sensitive string comparison. No special treatment is provided for white spaces. For example, the DC=COM suffix matches the cn=myuser,dc=mydc,dc=com distinguished name, but the dc = com suffix does not.

If the user filter encounters an issue, an error message is logged and the Windows Password Synchronizer operates as the user is accepted by the filter. For example, an invalid group name in its configuration. If the user filter decides not to accept a user, a message about the logging is displayed.

The configuration of the user filter is read again on each password notification. Therefore, changes to the configuration has an immediate effect. Restarting of the Windows operating system is not required for the changes to the user filter to be effective.

Note: The user filter configuration of the Windows Password Synchronizer is sensitive to modifications of the Windows groups, which are involved in the configuration. If some of the following changes occur, the Windows Password Synchronizer must be restarted, which requires restart of the operating system:

- When the Windows name of a group is modified. The change corresponds to the `sAMAccountName` attribute in the Active Directory.
- When the distinguished name of a group is modified. For example, the group is moved to another container.

This restriction is applicable for all the groups, which are in the configuration of the Windows Password Synchronizer during its lifetime.

Attention: The user filter feature of the Windows Password Synchronizer functions properly only on the systems that are part of a Windows domain. Workgroup systems are unable to use the user filter. If you configure the user filter on a workgroup system, the plug-in logs the error message as shown in the following example on every password change. The change is sent to the password store irrespective of the specified configuration:

```
User filtering failed: The specified domain either does not exist or could not
be contacted.
```

If you do not supply the configuration of user filter, the plug-in functions normally and no error is logged because there is no filtration.

Supported platforms

The following platforms are supported for the IBM Security Directory Integrator Windows Password Synchronization plug-in:

- Windows 7 (x86)
- Windows (x86/x86 - 64)
- Windows 2008 Standard Edition (x86/x86 - 64)
- Windows 2008 Enterprise Edition (x86/x86 - 64)
- Windows 2008 Datacenter Edition (x86/x86 - 64)
- Windows 2008 R2 Standard Edition (x86/x86 - 64)
- Windows 2008 R2 Enterprise Edition (x86/x86 - 64)
- Windows 2008 R2 Datacenter Edition (x86/x86 - 64)
- Windows 2012 R2

Deployment and configuration

Before you deploy the Windows Password Synchronizer, you must complete the post-installation configuration steps to register the Password Synchronizer for password change notifications.

Post-install configuration

1. From the `TDI_install_dir\pwd_plugins\windows` directory, copy the `tdipwflt.dll` DLL of the Windows Password Synchronizer.
2. Paste the DLL file to the `System32` folder of the Windows installation folder. On 64–bit Windows operating systems, you must paste the 64–bit DLL of the Password Synchronizer in the `System32` folder.
3. Add the name of the Windows Password Synchronizer DLL, `tdipwflt` to the `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\LSA\Notification Packages` Windows registry key. Do not delete any of the existing data from the Notification Packages.
4. From the *`TDI_install_dir`*`\pwd_plugins\windows` directory, run the `registerpwsync.reg` file, which is shipped with the Password Synchronizer. The following key is created for the Windows Password Synchronizer in the Windows registry:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\IBM\Security Directory Integrator\Windows
   Password Synchronizer
   ```

   Also, a string value `ConfigFile` is set and it contains the absolute file name of the configuration file of the Windows Password Synchronizer. See "Configuration parameters in the Windows registry" for a list of parameters that are added to the Windows registry.
5. Restart the system.

Password Stores setup information

The IBM Security Directory Integrator installer by default configures the Password Synchronizer to use the Log Password Store.

For information about setting up the Password Stores, see:

- Chapter 10, "JMS Password Store," on page 75
- Chapter 11, "Log Password Store," on page 85

Configuration parameters in the Windows registry

You must register the Windows Password Synchronizer in the Windows LSA to receive password change notifications. You must also register the external library name in the specific registry key.

Store the external library in one of the directories that is specified by the PATH environment variable. You must restart the operating system to load the external library.

Note: If the external library file is registered but you cannot load it, the Windows operating system becomes unstable.

When the native module of the Windows Password Synchronizer is initialized, it reads from the registry key folder:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\IBM\Security Directory Integrator\Windows Password
Synchronizer]
```

The following registry key contains the location of the configuration file of the Password Synchronizer:

Table 2. Primary registry key

Key name	Type	Description	Required?
`ConfigFile`	`REG_SZ`	Specifies full path of the configuration file of the Windows Password Synchronizer.	true

The following table lists the optional registry keys, which affect the behavior of the Windows Password Synchronizer. Use the Administration Tool to set the keys.

Note: You must not set the keys manually.

Table 3. Optional registry keys

Key name	Type	Description	Default	Required?
`disabled`	`REG_SZ`	Specifies whether the password change can be propagated to the Java Proxy process.	false	false
`reconfigure`	`REG_SZ`	Specifies whether the plug-in can reload its configuration file on the next password change notification.	false	false

Register the password filter module by editing the key in the following registry key folder:

`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\LSA`

Ensure that the following key is present:

Table 4. Optional registry keys

Key name	Type	Description	Default	Required?
`Notification Packages`	`REG_MULTI_SZ`	Specifies the external libraries to register for notifications.	unknown	true

Note: Do not delete any values of this key. Include the library name in the last line. Do not include the `.dll` extension to the name you enter.

Restart the Windows system.

Configuration parameters in the configuration file

The Windows Password Synchronizer plug-in has a template configuration file that is installed at `TDI_install_dir /pwd_plugins/windows/pwsync.props`.

Many of the configuration parameters in the configuration file are common to all the password plug-ins, see Configuration file parameters in Chapter 3, "Common configuration and utilities of password synchronization plug-ins," on page 13.

includeGroups
: An optional list of Windows groups. If the user is a member of any group in the list, the filter accepts the user. The assumption is that the user is not excluded by any of the exclude lists.

excludeGroups
: An optional list of Windows groups. If the user is not a member of any group in the list, the filter does not accept the user.

includeDNs
: An optional list of DN suffixes. If the distinguished name of a user

matches any suffix on the list, the filter accepts the user. The assumption is that the user is not excluded by any of the exclude lists.

excludeDNs
A list of DN suffixes. If the distinguished name of a user matches any suffix on the list, the filter does not accept the user.

accountTypes
Specifies the type of account for which the password changes are reported. Format of the parameter is a space-delimited list of account types.

The Password Synchronizer plug-in can report password changes to the following Windows account types:

NORMAL_ACCOUNT
Default account type that represents a typical user.

TEMP_DUPLICATE_ACCOUNT
Account for users whose primary account is in another domain.

INTERDOMAIN_TRUST_ACCOUNT
Permit to trust the account for a domain that trusts other domains.

WORKSTATION_TRUST_ACCOUNT
Computer account for a computer that is a member of this domain.

SERVER_TRUST_ACCOUNT
Computer account for a backup domain controller that is a member of this domain.

An example value for this key is:

```
"NORMAL_ACCOUNT WORKSTATION_TRUST_ACCOUNT"
```

Note: The Password Synchronizer always reports password changes to the accounts of type `NORMAL_ACCOUNT` regardless of whether `NORMAL_ACCOUNT` is specified in the **`AccountTypes`** parameter.

Enabling Local Security Policy

Before you deploy the Windows Password Synchronizer, you must modify the Local Security Policy settings.

About this task

Change the Local Security Policy as follows:

Procedure

1. Select **Control Panel** > **Administrative Tools** > **Local Security Policy**.
2. Select **Account Policies** > **Password Policy**.
3. Select **Passwords must meet complexity requirements** > **enabled**.

Results

Note:

1. Restart the system for this change to take place. Make sure that you set up the Password Store properties file before you restart the system.

2. If the Windows Server is configured as a domain controller, you must apply the **Passwords must meet complexity requirements** setting to the Active Directory Domain. Therefore, you must use the `Domain Security Policy` tool to modify the settings.

Plug-in administration tool

The plug-in administration tool, `pwsync_admin.exe`, is a command-line tool to run the administrative tasks.

You can find the plug-in administration tool in the *TDI_install_dir*`\pwd_plugins\windows` directory.

The primary purpose of this tool is to allow reconfiguration of the Windows Password Synchronizer without restarting the Windows system. For example, this tool enables changing of the password store without rebooting the Windows.

Note: Without rebooting the system, the Windows replaces the `tdipwflt.dll` plug-in in the Windows `System32` directory.

Administration tool usage

From the command line:

`pwsync_admin.exe` – command for 32-bit Windows
`pwsync_admin_64.exe` – command for 64-bit Windows

This administration tool takes a single command-line parameter, which can have one of the following values:

suspend_plugin
Writes a Boolean value to the Windows registry. See Windows registry settings. This value indicates that the subsequent password changes must not be propagated to the Java Proxy. This command causes subsequent password changes to be skipped until a **resume_plugin** command is issued.

resume_plugin
Writes a Boolean value to the Windows registry. See Windows registry settings. This value indicates that the subsequent password changes must be propagated to the Java Proxy. This command causes subsequent password changes to be synchronized until a **suspend_plugin** command is issued.

reconf_plugin
Writes a Boolean value to the Windows registry. See Windows registry settings. This value indicates that the plug-in must reload its configuration file. Reloading occurs only on the next password change. If there are any errors in the new configuration, they are not evident immediately. You can trigger a password change of a test account to enforce the reconfiguration. If the plug-in is suspended, the reconfiguration is postponed.

query_plugin
Queries the status of the plug-in to check whether the plug-in is loaded and its last initialization was successful.

stop_proxy
Causes the administration tool to connect through a socket to the command socket port of the Java Proxy and sends a stop request to the proxy. This stop request terminates the proxy.

start_proxy
Starts the Java Proxy, which causes the proxy configuration to be reloaded.

restart_proxy
Equivalent to **stop_proxy** command and **start_proxy** command.

query_proxy
Determines whether the Java Proxy is running or not.

Operational Windows registry settings

There are a number of Windows registry keys that are associated with the Windows Password plug-in and its operations:

Note: The keys are not present in the Windows registry after the plug-in is installed. These keys are not required for the normal operation of the plug-in.

Enable or disable plug-in
The **suspend_plugin** and **resume_plugin** commands use the following registry key:

```
[HKEY_LOCAL_system\SOFTWARE\IBM\Security Directory Integrator\Windows
Password Synchronizer] "disabled"="true"
```

If the key has a `true` value, the plug-in cannot synchronize the passwords. If this key is missing or has a value other than `true`, the plug-in synchronizes the passwords. The plug-in administration tool creates this key on the first use.

Reload plug-in configuration
The **reconf_plugin** command uses the following registry key:

```
[HKEY_LOCAL_system\SOFTWARE\IBM\Security Directory Integrator\Windows
Password Synchronizer] "reconfigure"="true"
```

If the key is set to `true`, on the next password change the plug-in reloads its configuration file. The plug-in also sets the value to `false` to reload the file only once.

Logging

The administration tool logs messages both to the console and to a log file named `pwsync_admin.log`, which is in the installation directory of the plug-in. You can use the log file to analyze errors that are encountered during the tool operations. You can also use the log file as a historical reference for the operations that are run by using this tool.

Considerations to use the administration tool

- When the plug-in is suspended, password changes are skipped and are not propagated by the plug-in. The plug-in suspension can result in inconsistencies such as password changes that are lost in the target synchronization system.
- The plug-in attempts to restart the Java Proxy only when the reconfiguration is requested and the proxy is not already running. See the **reconf_plugin** administration tool command for more information.
- When the Java Proxy is started, it loads the password store configuration file. The file is loaded when you restart the system or when the plug-in is not suspended. The Java Proxy is stopped when the password change occurs. If the user is editing the configuration file, the Java Proxy might load a possibly corrupted configuration.

- When the plug-in is not suspended and the Java Proxy is not running, if a password change is issued with the **Active Directory Users and Computers** user interface tool, the plug-in is notified by the Windows of the password change. The result is that the same password update is propagated two or three times. This change occurs because the plug-in starts the proxy on the next password change, which takes some time. This change causes the Windows to notify the plug-in several times of the same password change. This multiple reporting occurs at the first instance when the Java Proxy is not running.
- When the plug-in is configured with the LDAP Password Store and the LDAP Store is set for asynchronous storing (`waitForStore=false` specified in the LDAP Store configuration file), and when the plug-in is not suspended, it is possible that a **stop_proxy** command causes some password changes to be skipped.

You can troubleshoot the problems by using the following details:

- Suspend the plug-in by using a **suspend_plugin** command before you run the **stop_proxy** or **restart_proxy** command.
- Make a copy of the configuration file for editing purposes. Replace the old configuration file with the new one when all edits are complete.
- Make the necessary configuration changes at a low usage time to ensure that only a few password changes are skipped and not propagated.

Changing configuration without rebooting Windows system

You must change the plug-in configuration settings at a low usage time, when the password changes are unlikely.

About this task

Note: After these steps are completed, the plug-in, the Java Proxy, and the Password Store use the new configuration settings. During the short window, when the plug-in is suspended, you can skip the password changes. The changes occur in the Windows domain controller, but they are not propagated by the plug-in.

Procedure

1. Copy the configuration file to a temporary location.
2. Edit the file in this temporary location.
3. Copy the edited file back to the original location.
4. Run the **pwsync_admin.exe suspend_plugin** command.
5. Run the **pwsync_admin.exe reconf_plugin** command.
6. Run the **pwsync_admin.exe stop_proxy** command.
7. Run the **pwsync_admin.exe start_proxy** command.
8. Run the **pwsync_admin.exe resume_plugin** command.

Results

Alternatively, if you want to change only some Password Store settings, you can skip the reconfiguration command in the preceding steps. However, the settings must not be related to the plug-in or the proxy.

Reliability and availability of Password Synchronizer

You can use the plug-in administration tool and the error log to analyze the reliability and availability of the Password Synchronizer.

Initialization failure

If the Password Synchronizer fails to initialize, the Windows cannot send notifications about password changes to the Password Synchronizer. For example, non-availability of the configuration file. The password changes can take place, but the Windows Password Synchronizer cannot intercept them.

The most reliable way to determine whether the Password Synchronizer is initialized successfully is to check its error log. Additionally, you can use the **query_plugin** command of the plug-in administration tool.

Reconfiguration failure

If the reconfiguration fails, the Password Synchronizer reaches a non-initialized state and rejects all password changes. Check the error log of the Password Synchronizer to see whether the reconfiguration is successful. See "Plug-in administration tool" on page 24.

Chapter 5. Sun Directory Server Password Synchronizer

The Sun Directory Server Password Synchronizer intercepts changes to LDAP passwords in the Sun Directory Server.

Components of Sun Directory Password Synchronizer

You can build a solution that synchronizes passwords, but without using the Sun Directory Server plug-in. For more information about solution building, see "Solution building" on page 1.

The Sun Directory Password Synchronizer consists of the following parts:

Sun Directory Server plug-in
The plug-in is a native binary, which uses the plug-in API of the Sun Directory Server. It runs in the Sun Directory Server process.

Java Proxy
A separate Java process, which is started or stopped by the server plug-in. The main purpose of this process is to host the Password Storage component and communicate with the plug-in. For more information about the Java Proxy, see "Password synchronization architecture and workflow" on page 4.

Password Storage component
A Java component, which runs inside the Java Proxy and stores passwords in a particular Password Store such as LDAP directory or message queue. For more information about the Password Storage components, see "Specialized components" on page 3.

Passwords in the Sun Directory Server are stored in the `userPassword` LDAP attribute. The Password Synchronizer intercepts updates of the `userPassword` LDAP attribute.

The Sun Directory Server Password Synchronizer intercepts modifications of the `userPassword` attribute of entries of any object class.

Password updates are intercepted for the following types of entry modifications:

- When a new entry is added in the directory, the entry contains the `userPassword` attribute.
- When an existing entry is modified, one of the modified attributes is `userPassword`. The entry includes the following cases:
  - The `userPassword` attribute is added. For example, the entry did not have a `userPassword` attribute.
  - The `userPassword` attribute is modified. For example, the entry had this attribute and its value is now changed.
  - The `userPassword` attribute is deleted from the entry.

Note:

1. Deletion of the entries is not intercepted by the Sun Directory Server Password Synchronizer even when the entry contains the `userPassword` attribute.

2. The `userPassword` attribute in the Sun Directory Server is multi-valued. Users can have several passwords. The Sun Directory Server Password Synchronizer intercepts and reports any change in any of the password values.

Hashed Passwords

The Password Synchronizer ignores hashed password values. Only the plaintext passwords are synchronized. The Password Synchronizer receives hashed passwords in the following cases:

- If an LDAP client sends a password value that is already hashed, the Sun Directory Server accepts it. However, the Password Synchronizer cannot obtain a plaintext password and ignores it. For example, if an LDAP client sends `"{SHA}5yfRRkrhJDbomacm2lsvEdg4GyY="` instead of `"mypass"`, the Password Synchronizer sends no password to the Password Store.
- If password encryption is set to one-way transformation, for example, `"crypt"`, `"MD5"`, or `"SHA-1"`, passwords are stored in hashed form in the directory. The replication operations work with hashed password values. The Password Synchronizers on replication consumers receive the already hashed password values.

Supported platforms

The Sun Directory Server Password Synchronizer is available for the Sun Directory Server on the following platforms:

- Solaris 10 SPARC (32/64-bit), Sun ONE 5.2, Sun Java System Directory Server 7.0 (32/64-bit)
- Solaris 11 SPARC (32/64-bit), Sun ONE 5.2, Sun Java System Directory Server 7.0 (32/64-bit)

Deployment and configuration of Sun Directory Server Password Synchronizer

To configure the Sun Directory Server Password Synchronization plug-in, use the template configuration file that is installed at *TDI_install_dir*`/pwd_plugins/sun/pwsync.props`.

When the Sun Directory Server plug-in is initialized, the configuration file is set as the last parameter of the registration line of the plug-in. The plug-in then reads the file. Some of the parameters in the configuration file are shared between the plug-in and the Java proxy. For a complete list of the supported properties, see theChapter 3, “Common configuration and utilities of password synchronization plug-ins,” on page 13 section.

The following property is specific to the Sun Directory Server Password Synchronizer:

syncBase

This optional property enables restriction of the part of the directory tree where the passwords are intercepted. The specified string value is the LDAP distinguished name (dn) of the root of the tree where the entry passwords are to be intercepted. For example, when you specify `"o=ibm, c=us"` it results in

- Intercepting the password update `"cn=Kyle Nguyen, ou=Austin, o=IBM, c=US"`.

- Skipping the password update "cn=Henry Nguyen, o=SomeOtherCompany, c=US".

Setting no value for this property results in interception of the password updates in the entire directory tree.

Registering Sun Directory Server Password Synchronizer with Sun Directory Server

Use the Directory Server Management Console to register the Sun ONE Directory Server. To register the Sun Java System Directory Server, you must use the **dsconf** command-line tool that is shipped with the Sun Directory Server.

Registering Sun ONE Directory Server 5.2

About this task

To register the plug-in, stop the Sun Directory Server. Add the following lines to the dse.ldif configuration file of the Sun Directory Server, by using the Directory Server Management Console:

```
dn: cn=IBM DI PassSync,cn=plugins,cn=config
nsslapd-pluginPath: TDI_install_dir/pwd_plugins/sun/sunpwsync.dll
nsslapd-pluginEnabled: on
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
cn: IBM DI PassSync
nsslapd-pluginType: object
nsslapd-pluginInitfunc: PWSyncInit
nsslapd-pluginarg0: TDI_install_dir/pwd_plugins/sun/pwsync.props
nsslapd-pluginId: ibmdi.pwsync
nsslapd-pluginVersion: 7.2
nsslapd-pluginVendor: IBM
nsslapd-pluginDescription: IBM Security Directory Integrator plug-in for password
synchronization
```

Note: The 64-bit Sun Directory Server, which is running on Solaris, searches for the 64-bit libraries in a directory under the specified path.

For example, if you set the value of nsslapd-pluginPath in the configuration entry as

```
nsslapd-pluginPath: TDI_install_dir/pwd_plugins/sun/libsunpwsync_64.so
```

, then a 64-bit Directory Server running in Solaris Operating Environment searches for a 64-bit plug-in library named: *TDI_install_dir*/pwd_plugins/sun/64/libsunpwsync_64.so

Therefore, on Solaris, the 64-bit binary for the Sun Directory Server Password Synchronizer is shipped in that folder.

Note: You must avoid manually modifying the dse.ldif configuration file of the Sun Directory Server. Use the following steps to register the plug-in by importing the LDIF statements in the Directory Server console:

1. Save the LDIF content in an LDIF file.
2. Open the Directory Server instance in the Directory Server console.
3. Go to the **Tasks** tab.
4. Select **Import LDIF**.
5. Browse to the location of the file.

6. Select the **Add only** check box.
7. Clear the **Continue on error** check box. Click **OK**.
8. Restart the Directory Server to load the plug-in.

Sun Java System Directory Server Enterprise Edition 7.0

About this task

Ensure that the Directory Server is running. Use the following steps to register the plug-in:

Procedure

1. Register the plug-in binary by changing the name of the binary depending on the platform such as `sunpwsync.dll` or `libsunpwsync.so`.

   ```
   dsconf create-plugin <access options> -H
   "TDI_install_dir/pwd_plugins/sun/sunpwsync.dll"
    -F PWSyncInit -Y object -G "TDI_install_dir/pwd_plugins/sun/pwsync.props"
   "IBM DI PassSync"
   ```

2. Enable the plug-in.

   ```
   dsconf enable-plugin access-options "IBM DI PassSync"
   ```

3. Restart the Directory Server to load the plug-in.

Results

Notes:

- You must replace the *access-options* placeholder with the access details and credentials that you use to connect to the Directory Server.

  For example, you can use the `–p 1389 ––unsecured` option if:

  – The Directory Server is on the localhost.
  – It accepts non-SSL connections on the port 1389.
  – It uses the default administrator DN `cn=Directory Manager`.

  For a list of options that the **dsconf** command-line tool supports, see http://www.oracle.com/technetwork/indexes/documentation/index.html.
- To unregister the plug-in, you can use the following command:

  ```
  dsconf
  delete-plugin access-options "IBM DI PassSync"
  ```

Enabling Sun Directory Server logging for plug-ins

The Sun Directory Server Password Synchronizer logs messages in the error log of the Sun Directory Server. The messages from server plug-ins are not included in the error log for performance reasons.

Enabling Sun ONE Directory Server 5.2

Procedure

1. On the Directory Server console, select the **Configuration** tab.
2. In the navigation tree, expand the `Logs` folder and select the **Error Log** icon.
3. The error log configuration attributes are displayed in the pane on the right side of the window.
4. Select **Enable Logging** to log errors.
5. Select **Plug-ins** in the **Log Level** list box.
6. Click **Save**.

Enabling Sun Java System Directory Server Enterprise Edition 7.0

Procedure

1. Ensure that the Directory Server is running.
2. Run the following command by using the **dsconf** tool of the Directory Server:
   `dsconf set-log-prop` *`access-options`* `error level:err-plugins`

   For more information about access-options, see the "Registering Sun Directory Server Password Synchronizer with Sun Directory Server" on page 31 topic.

Results

To query the current level of the error log, run the following command:
`dsconf get-log-prop` *`access-options`* `error level`

Chapter 6. IBM Security Directory Server Password Synchronizer

The IBM Security Directory Server Password Synchronizer intercepts changes to the LDAP passwords in the IBM Security Directory Server.

Components

You can build a solution that synchronizes passwords, but without using the IBM Security Directory Server plug-in. For more information about solution building, see "Solution building" on page 1.

The IBM Security Directory Integrator Password Synchronizer consists of the following parts:

IBM Security Directory Server plug-in
The plug-in is a native binary, which uses the plug-in API of the IBM Security Directory Server. The plug-in runs in the process of the IBM Security Directory Server.

Java Proxy
A separate Java process, which is started or stopped by the server plug-in. The main purpose of the process is to host the Password Storage component and communicate with the plug-in. For more information about the Java Proxy, see "Password synchronization architecture and workflow" on page 4.

Password Storage component
A Java component, which runs inside the Java Proxy process and stores passwords in a particular Password Store such as LDAP directory or message queue. For more information about the Password Storage components, see "Specialized components" on page 3.

Passwords in the IBM Security Directory Server are stored in the `userPassword` LDAP attribute. The Password Synchronizer intercepts updates of the `userPassword` LDAP attribute.

The IBM Security Directory Server Password Synchronizer intercepts modifications of the `userPassword` attribute of entries of any object class.

Password updates are intercepted for the following types of entry modifications:

- When a new entry is added in the directory, the entry contains the `userPassword` attribute.
- When an existing entry is modified, one of the modified attributes is the `userPassword` attribute. The entry includes the following cases:
  - The `userPassword` attribute is added. For example, the entry did not have a `userPassword` attribute.
  - The `userPassword` attribute is modified. For example, the entry had this attribute and its value is now changed.
  - The `userPassword` attribute is deleted from the entry.

Note:

1. Deletion of the entries (users) is not intercepted by the IBM Security Directory Server Password Synchronizer even when the entry contains the `userPassword` attribute.
2. The `userPassword` attribute in the IBM Security Directory Server is multi-valued. Users can have several passwords. The IBM Security Directory Server Password Synchronizer intercepts and reports the changes in any of the password values.

Supported platforms

The IBM Security Directory Server Password Synchronizer is available for the IBM Security Directory Server on the following platforms and for the following versions:

- Windows 2008 Standard Edition (x86/x86 – 64), IBM Security Directory Server 6.1, 6.2, and 6.3 (32/64–bit)
- Windows 2008 Enterprise Edition (x86/x86 – 64), IBM Security Directory Server 6.1, 6.2, and 6.3 (32/64–bit)
- Windows 2008 Datacenter Edition (x86/x86 – 64), IBM Security Directory Server 6.1, 6.2, and 6.3 (32/64–bit)
- Windows 2008 R2 Standard Edition (x86/x86 – 64), IBM Security Directory Server 6.1, 6.2, and 6.3 (32/64–bit)
- Windows 2008 R2 Enterprise Edition (x86/x86 – 64), IBM Security Directory Server 6.1, 6.2, and 6.3 (32/64–bit)
- Windows 2008 R2 Datacenter Edition (x86/x86 – 64), IBM Security Directory Server 6.1, 6.2, and 6.3 (32/64–bit)
- AIX® 6.1 (64 bit), IBM Security Directory Server 6.0 (64 bit), IBM Security Directory Server 6.1, 6.2, and 6.3 (64–bit)
- AIX 7.1 (64 bit), IBM Security Directory Server 6.0 (64 bit), IBM Security Directory Server 6.1, 6.2, and 6.3 (64–bit)
- Solaris 10 SPARC (64–bit), IBM Security Directory Server 6.1, 6.2, and 6.3 (64–bit)
- Solaris 11 SPARC (64–bit), IBM Security Directory Server 6.1, 6.2, and 6.3 (64–bit)
- RHEL ES/AS 5.0 (x86/x86 – 64), IBM Security Directory Server 6.1, 6.2, and 6.3 (32/64–bit)
- RHEL ES/AS 6.0 (x86/x86 – 64), IBM Security Directory Server 6.1, 6.2, and 6.3 (32/64–bit)
- SLES 10 (x86/x86 – 64), IBM Security Directory Server 6.1, 6.2, and 6.3 (32/64–bit)
- SLES 11 (x86/x86 – 64), IBM Security Directory Server 6.1, 6.2, and 6.3 (32/64–bit)
- RedFlag Data Center 5.0 SP1/Asianix 2.0 SP1, IBM Security Directory Server 6.1, 6.2, and 6.3 (32–bit)

Deployment and configuration

You must register the IBM Security Directory Server Password Synchronizer with the IBM Security Directory Server before you deploy and configure the plug-in.

Plug-in registration with IBM Security Directory Server

To register the plug-in, edit the *ids_dir*`/etc/ibmslapd.conf` configuration file of the IBM Security Directory Server.

Note: Before you edit the file, ensure that the server is not running.

1. Find the section `dn: cn=Directory, cn=RDBM Backends, cn=IBM Directory, cn=Schemas, cn=Configuration` and add the following configuration details:

   Win32 ibm-slapdPlugin: preoperation "*TDI_install_dir*\pwd_plugins\tds\idspwsync.dll" PWSyncInit "*TDI_install_dir*\pwd_plugins\tds\pwsync.props"

   AIX64 ibm-slapdPlugin: preoperation "*TDI_install_dir*/pwd_plugins/tds/libidspwsync_64.a.so "PWSyncInit "*TDI_install_dir*/pwd_plugins/tds/pwsync.props"

   Linux32 ibm-slapdPlugin: preoperation "*TDI_install_dir*/pwd_plugins/tds/libidspwsync.so" PWSyncInit "*TDI_install_dir*/pwd_plugins/tds/pwsync.props"

2. Restart the IBM Security Directory Server.

Configuration of IBM Security Directory Server Password Synchronizer

The IBM Security Directory Server plug-in has a template configuration file that is installed at *TDI_install_dir*/pwd_plugins/tds/pwsync.props. When the plug-in is initialized, the configuration file is set as the last parameter in the registration line of the plug-in. Some of the parameters in the configuration file are shared between the plug-in and the Java Proxy. For a list of the supported properties, see Chapter 3, “Common configuration and utilities of password synchronization plug-ins,” on page 13.

The **syncBase** property is specific to the IBM Security Directory Server Password Synchronizer:

syncBase

This optional property enables restriction of the part of the directory tree where the passwords are intercepted. The specified string value is the LDAP distinguished name (dn) of the root of the tree where entry passwords are to be intercepted. For example, when you specify `"o=ibm, c=us"` it results in

- Intercepting the password update `"cn=Kyle Nguyen, ou=Austin, o=IBM, c=US"`.
- Skipping the password update `"cn=Henry Nguyen, o=SomeOtherCompany, c=US"`.

Setting no value for this property results in the interception of password updates in the entire directory tree.

Chapter 7. Domino HTTP Password Synchronizer

The Domino HTTP Password Synchronizer intercepts changes of the Internet password, which is also known as HTTP password, for the Notes users.

Overview

You can use the Domino HTTP Password Synchronizer to intercept the following types of password changes:

Administrative password resets

Users, for example, administrators, with the necessary rights, can change their password or the password of other users without being prompted for the old password.

You can change the HTTP password by editing the **Internet password** field of the Person document of any user by using the:

- Lotus Domino Administrator client
- Web browser interface

Normal user password changes

A user changes the password and is prompted for the old password. Users can change the password from:

- Web browser by using the `Change Password` form from the Domino web server configuration database `domcfg.nsf`.
- iNotes®

Deployment on Domino Servers

You can deploy the Domino HTTP Password Synchronizer in the following modes:

- Both administrative password resets and normal user password changes are intercepted.
- Only normal user password changes are intercepted.
- Only administrative password resets are intercepted.

Supported platforms

The Domino HTTP Password Synchronizer is supported on the following platforms:

- Windows Server 2008 Standard Edition (x86/x86-64), Domino 8.0 and Domino 8.5.x
- Windows Server 2008 Enterprise Edition (x86/x86-64), Domino 8.0, and Domino 8.5.x
- Windows Server 2008 Datacenter Edition (x86/x86-64), Domino 8.0, and Domino 8.5.x
- Windows Server 2008 R2 Standard Edition (x86/x86-64), Domino 8.0 and Domino 8.5.x
- Windows Server 2008 R2 Enterprise Edition (x86/x86-64), Domino 8.0, and Domino 8.5.x
- Windows Server 2008 R2 Datacenter Edition (x86/x86-64), Domino 8.0, and Domino 8.5.x

- AIX 6.1 (32/64 bit), Domino 8.0 and Domino 8.5.x
- AIX 7.1 (PPC_64), Domino 8.0 and Domino8.5.x
- Solaris 10 SPARC (32/64 bit), Domino 8.0 and Domino 8.5.x
- SLES 10 (x86), Domino 8.0 and Domino 8.5.x
- SLES 10 (x86–64), Domino 8.0 and Domino 8.5.x
- SLES 11 (x86), Domino 8.0 and Domino 8.5.x
- SLES 11 (x86–64), Domino 8.0 and Domino 8.5.x
- RHEL ES/AS 5.0 (x86), Domino 8.0 and Domino 8.5.x
- RHEL ES/AS 5.0 (x86–64), Domino 8.0 and Domino 8.5.x
- RHEL ES/AS 6.0 (x86), Domino 8.0 and Domino 8.5.x
- RHEL ES/AS 6.0 (x86–64), Domino 8.0 and Domino 8.5.x
- Red Flag Data Center 5.0 SP1 /Asianix 2.0 SP1, Domino 8.0 and Domino 8.5.x

Installation and configuration file options

The Domino HTTP Password Synchronizer is installed by using the standard IBM Security Directory Integrator installer wizard.

Configuration file options

The Domino plug-in has a template configuration file that is installed at *TDI_install_dir*/pwd_plugins/domino/pwsync.props. When the Domino Server plug-in is initialized, the configuration file must be at *domino_data_dir*/idipwsync/pwsync.props on UNIX. On Windows, the file must be at *domino_program_dir*\idipwsync\pwsync.props. Some of the parameters in the configuration file are shared between the plug-in and the Java Proxy. For a list of the supported properties, see Chapter 3, "Common configuration and utilities of password synchronization plug-ins," on page 13.

The Domino Password Synchronizer supplies an option to synchronize password changes by using a unique user identifier. The user identifier that is supplied by the Domino, uniquely identifies users within their corresponding Domino Server.

The following common properties are ignored by the Domino plug-in:

proxyStartExe
: The Java Proxy is started as a Domino task when the Domino Server is started. The Java Proxy automatically stops when the Domino Server shuts down. You can also manually shut down by using the stopProxy script. To start the Java Proxy, the Domino Server instantiates the **com.ibm.di.plugin.domino.ProxyLoader** class, which is a native Domino add-in, a separate Domino task. The Domino Server is configured to start another JVM dedicated for that Domino task. You can configure it while you edit the notes.ini file and adding the runjava line.

logFile
: This property is ignored because the Domino plug-in uses three log files instead of one.

In addition to the common configuration properties, the Domino plug-in recognizes the following properties:

admin.logFile
: Sets the file that the admin agent login to. If this file is not set, the agent cannot log any information. The default value is idipwsync/admin.log.

client.logFile
Sets the file that the client agent login to. If this file is not set, the agent cannot log any information. The default value is `idipwsync/client.log`.

web.logFile
Sets the file that the web agent login to. If this file is not set, the agent cannot log any information. The default value is `idipwsync/web.log`.

useUniqueID
Turns on or off the unique user ID. If this property is set to true, the plug-in sends the unique ID instead of the actual user name. The default value is false.

ignoreMissingUniqueID
If this property and the `useUniqueID` property are set to true, the plug-in skips the synchronization for the users for whom `useUniqueID` cannot not be found. The default value is false.

usernamePrefix
If the:

- `usernamePrefix` property is set to true
- `useUniqueID` property is set to true
- `ignoreMissingUniqueID` property is set to false

The plug-in prefixes the distinguished name of the user with the value of the `usernamePrefix` property.

Post-install configuration

After the plug-in installation, you must copy the necessary configuration files to the data directory of the Domino Server. You must also create a signer for the Password Synchronizer agents.

Copy the following files to the data directory of the Domino Server:

- Copy all the files from the *TDI_install_dir*`/pwd_plugins/jars` folder to the *domino_jvm_directory*`/lib/ext` folder on the Domino Server and to the `Lotus\Notes\jvm\lib\ext` folder on the system where the Lotus Domino Designer is installed.
- Copy the files `idipwsync.nsf` and `pwsync_install_r8.nsf` from the *TDI_install_dir*`/pwd_plugins/domino` folder to the data directory of the Domino Server, *domino_data_dir*.
- Copy the Domino and Java Proxy configuration file to *domino_data_directory*`/idipwsync/pwsync.props` on UNIX. The configuration file template is shipped in *TDI_install_dir*`/pwd_plugins/domino/pwsync.props`. On Windows, copy to the *domino_program_directory*`\idipwsync\pwsync.props` file.

Note: On Linux and UNIX based platforms, install the Password Store with the Domino user (`notes` by default) to provide the necessary privileges to the Domino JVM to run the Password Store. Also, make sure that the Domino user has the necessary privileges to read the files that are copied to the Domino Server.

You must restart the server to load the new files.

Signer for the Password Synchronizer agents

Creating a signer for the Password Synchronizer agents involves the following process:

- Creating a person as an agent signer.
- Downloading the ID file of the newly generated person.
- Providing manager access to the `pubnames.ntf` and `admin4.ntf`templates
- Providing the `sign or run unrestricted methods and operations` privilege to the signer.

Creating a signer for the Password Synchronizer agents

You must create a signer for the IBM Security Directory Integrator Domino HTTP Password Synchronizer agents with necessary privileges before you deploy the plug-in.

Procedure

1. Open the Domino Administrator.
2. Click the **People & Groups** tab.
3. On the right panel, select **People** > **Register**.
4. In the Register Person wizard, type `IIDIPWSyncSigner` in the **Last name** field.
5. Type the password in the **Password** field.
6. In the **Mail system** field, select **None**.
7. Select the **Create a Notes ID for this person** check box. The ID file is used to sign the agents.
8. Click **Register**.

What to do next

"Downloading the ID file"

Downloading the ID file

You must download the ID file of the newly generated person from the Person document in the Domino directory.

Procedure

1. Open the Domino Administrator.
2. Click the **People & Groups** tab.
3. In the left navigation panel, open the **People** node.
4. Select the `IDIPWSyncSigner` person.
5. Click **Edit Person** to open the Person document.
6. Right-click the `UserID` file attachment and click **Save**. The `UserID` file is attached at the lower left corner of the Basics window.
7. Click **Cancel** to close the Person document without changes.

What to do next

"Providing manager access"

Providing manager access

The signer must have `Manager` access to the `pubnames.ntf` and `admin4.ntf` templates.

Procedure

1. Open the Domino Administrator.
2. Click the **Files** tab.
3. From **Show me**, select **Templates only**.
4. From the list of templates, select **admin4.ntf**.
5. Right-click **admin4.ntf** and select **Access Control** > **Manage**.
6. Click **Add**.
7. Select **IDIPWSyncSigner**.
8. From the **User Type** list, select **Person**.
9. From the **Access** list, select **Manager**.
10. Click **OK** to close the Access List window.
11. Follow Step 1 to Step 10 for the `pubnames.ntf` template.

What to do next

"Providing required privileges to the signer"

Providing required privileges to the signer

You must provide the `Sign or run unrestricted methods and operations` privilege to the signer.

About this task

Note: If you have multiple Domino Servers, you must run the following steps on each of the server.

Procedure

1. Open the Domino Administrator.
2. Click the **Configuration** tab.
3. Select **Server** > **All Server Documents**.
4. Select the server document.
5. Click **Edit Server**.
6. Click the **Security** tab.
7. In the **Programmability Restrictions** section, add the signer person in the **Sign or run unrestricted methods and operations** field.
8. Click **Save & Close**.

Deployment on a single Domino Server

You must run the necessary configuration steps to deploy the plug-in on a single Domino Server.

To install the Domino HTTP Password Synchronizer on the Domino, run the installer on the system where the Domino Server is installed. The installer places all required files in the appropriate directory structures.

The file paths of the Domino Server data directories are as follows:

- The Domino Server program folder is known as *domino_program_directory*. For example, `C:\Program Files\IBM\Lotus\Domino` on the Windows platform. On the Linux and UNIX based platforms, `/opt/ibm/lotus`.

- The Domino Server data folder is known as *domino_data_directory*. For example, `C:\Program Files\IBM\Lotus\Domino\Data` on the Windows platform. On the Linux and UNIX based platforms, `/local/notesdata`.
- The Domino Server JVM folder is known as *domino_jvm_directory*. For example, `C:\Program Files\IBM\Lotus\Domino\jvm` on the Windows platform. On the Linux and UNIX based platforms, `/opt/ibm/lotus/notes/80000/linux/jvm`.

Notes:

1. The Domino HTTP Password Synchronizer ships with the *TDI_install_dir*`/pws_plugins/domino/pwsync.props` template configuration file that has all the required properties preset by default, and ready for use.
2. The default Password Store, which is configured in the shipped `pwsync.props` file is the Log Password Store. This Password Store logs all the captured passwords in the log file of the proxy. You must use this Password Store only for diagnostic purposes.

The following table explains the deployment steps.

Step	Description
1	Make sure that the new files that are copied during the post-install phase are read by the Domino Server.
2	The external databases that are shipped with the IBM Security Directory Integrator must be signed by the Domino Server to be able to vouch for their integrity. See "Signing databases with Server ID" on page 45.
3	By editing the `pubnames.ntf` template, you can change the behavior of the `names.nsf` database. A code is placed on several key places to intercept the plain password. When the password is captured, it is passed to the appropriate Java agent such as `IDIPWSyncClientAgent` or `IDIPWSyncWebAgent`. See "Updating pubnames.ntf template design" on page 46.
4	By editing the `admin4.ntf` template, you can change the behavior of the `admin4.nsf` database. The copied Java agent `IDIPWSyncAdminRequestAgent` is responsible to periodically process the administration requests, posted by various users when they change the passwords. See "Updating the admin4.ntf template design" on page 49.
5	Agents are run with the rights of their signer. The agents of the Password Synchronizer must run restricted operations such as network access or file system access. Therefore, they can be signed by someone who has the `sign or run unrestricted methods and operations` privilege. See "Signing the agents with a signer" on page 50.
6	Refreshing the design of the `names.nsf` database applies the changed template to the existing database. See "Refreshing names.nsf database design" on page 51.
7	Refreshing the design of the `admin4.nsf` database applies the changed template to the existing database. See "Refreshing the design of the admin4.nsf database" on page 51.

Step	Description
8	The various Java agents use the idipwsync.nsf database to store the documents that need further processing. You must encrypt the documents to protect them in this database. The secret key that is created in this step is used in the database encryption process. See "Setting up secret key encryption infrastructure" on page 51.
9	Port encryption encrypts the communication between Lotus Domino Administrator and the Domino Server, bringing an additional layer of security to the network communication. See "Setting up port encryption" on page 53.
10	SSL is necessary to secure the communication between the web browser and the Domino HTTP Server. If SSL is not set up, the password is transferred over the network in plain text. See "Setting up SSL for Domino HTTP Server" on page 53.
11	The Java Proxy runs in the JVM that is shipped with the Domino. The process starts as a server task when you start the Domino Server. See "Configuring Domino Server to automatically start and stop Java proxy" on page 54.
12	Configure each Lotus Domino Administrator client to enable administrative password changes. See "Configuring execution control list of Lotus Domino Administrator clients" on page 54.
13	The IDIPWSync group contains a list of users who has the rights to change the password of other users. Typically, only the administrators are present in this group. Regular users can still change the passwords through iNotes even if they do not belong to this group. Only members of this group can access the idipwsync.nsf database. The idipwsync.nsf database is used to transfer data between Lotus script and the Password Synchronizer agents. The signer of the Password Synchronizer agents must also be added to the IDIPWSync group so that the agents can access the idipwsync.nsf database. Agents are run with the rights of their signer. See "Configuring Access Control List" on page 55.
14	The pwsync_install_r8.nsf database is used only to distribute the required template objects. When the Domino HTTP plug-in is set up, the database is no longer required and you can delete the database. See "Deleting pwsync_install_r8.nsf database" on page 55.

Signing databases with Server ID

You must sign the pwsync_install_r8.nsf and idipwsync.nsf databases with the Active Server ID.

Procedure

1. Start the Lotus Domino Administrator.
2. Click **Files**.
3. Right-click the pwsync_install_r8 database and select **Sign**.
4. In **Sign Database**, under **Which ID do you want to use?**, select **Active Server's ID**.

5. Right-click the **IDIPWSync** database and select **Sign**.
6. In **Sign Database**, under **Which ID do you want to use?**, select **Active Server's ID**.
7. Click **OK**.

What to do next

"Updating pubnames.ntf template design"

Updating pubnames.ntf template design

You must edit the `pubnames.ntf` template to make the necessary changes to the `names.nsf` database.

Procedure

1. Start the Lotus Domino Designer.
2. Open the following items:
   a. Open the `pwsync_install_r8.nsf` database.
   b. Open the `pubnames.ntf` template.
3. Copy Agents:
   a. In the `pwsync_install_r8.nsf` database, select **Code/Agents**.
   b. Select the `IDIPWSyncClientAgent` and `IDIPWSyncWebAgent` agents.
   c. Right-click the selected agents and select **Copy**.
   d. In **pubnames.ntf**, select **Code/Agents**.
   e. Select **Edit** > **Paste** to paste the two agents.

   Note: If the `Person` form is not modified with user-customized logic, the `Person` form from the Password Synchronizer is used.
4. Rename the `Person` form in `pubnames.ntf`:
   a. In the `pubnames.ntf` database, select **Forms**.
   b. Open the **Person** form.
   c. Select **Design** > **Form Properties**.
   d. Edit the **Name** field. Change the name to **original_Person**, or the name of your choice, other than **Person**.

      Note: Make sure that the default alias Person can be unset from that field.
   e. Save the form.
   f. Close the form.
5. Copy the `Person` form:
   a. In `pwsync_install_r8.nsf`, select **Forms**.
   b. Right-click on the **Person** form and select **Copy**.
   c. In the `pubnames.ntf` database, select **Forms**.
   d. Select **Edit** > **Paste** to paste the form.
   e. Select **Edit -> Paste** to paste the form.

      If the `Person` form is modified with user-customized logic that you want to retain, manually copy the Password Synchronizer source code for the Person form.
6. Copy the `Person` form source code:
   a. Copy `WebQuerySave` event code:

1) In the pwsync_install_r8.nsf database, select **Forms**.
2) Open the Person form.
3) Select the WebQuerySave event.
4) Copy the lines that start with

```
REM {start of IDI Password Synchronizer code};
```

and end with

```
REM {end of IDI Password Synchronizer code};
```

5) In the pubnames.ntf database, select **Forms**.
6) Open the Person form.
7) Select the WebQuerySave event.
8) Paste the copied source code. Make sure that the pasted code is displayed before any other code in this event.
9) Save the form.

b. Copy Querysave event code:
1) In pwsync_install_r8.nsf, select **Forms**.
2) Open the Person form.
3) Select the Querysave event.
4) Copy the lines that start with

```
'start of Password Synchronizer code
```

and end with

```
'end of Password Synchronizer code
```

.
5) In the pubnames.ntf database, select **Forms**.
6) Open the Person form.
7) Select the Querysave event.
8) Paste the copied source code. Make sure that the pasted code is displayed just before the end of the Querysave procedure.
9) Save the form.

c. Copy SyncPass event code:
1) In pwsync_install_r8.nsf database, select **Forms**.
2) Open the Person form.
3) Select the SyncPass event.
4) Copy all code for the **SyncPass** function.
5) In the pubnames.ntf database, select **Forms**.
6) Open the Person form.
7) Select the Querysave event.
8) Paste the copied source code. Make sure that the pasted code is displayed after all code in the event. A new event named SyncPass is created, and the pasted code is transferred there.
9) Save the form.

If the $PersonInheritableSchema subform is not modified with user-customized logic, the $PersonInheritableSchema from the Password Synchronizer is used.

7. Rename the $PersonInheritableSchema subform in pubnames.ntf:
a. In the pubnames.ntf database, select **Shared Elements/Subforms**.

b. Open the $PersonInheritableSchema subform.
c. Select **Design** > **Subform Properties**.
d. Edit the **Name** field. Change the name to **original_$PersonInheritableSchema**, or the name of your choice other than **$PersonInheritableSchema**.
e. Save the form.
f. Close the form.

8. Copy the $PersonInheritableSchema subform:
   a. In the pwsync_install_r8.nsf database, select **Shared Elements/Subforms**.
   b. Right-click the $PersonInheritableSchema form and select **Copy**.
   c. In the pubnames.ntf database, select **Shared Elements/Subforms**.
   d. Select **Edit** > **Paste** to paste the subform.

      If the $PersonInheriableSchema subform is modified with user-customized logic that you want to retain, you must manually copy the Password Synchronizer source code.
9. Copy the $PersonInheritableSchema subform code:
   a. Copy the **HTTPPassword** field code:
      1) In the pwsync_install_r8.nsf database, select **Shared Elements/Subforms**.
      2) Open the $PersonInheritableSchema subform.
      3) Select the **HTTPPassword** field (near the bottom of the form).
      4) Select the Input Translation event.
      5) Copy the lines that start with

         ```
         REM {start of IDI Password Synchronizer code};
         ```

         and end with

         ```
         REM {end of IDI Password Synchronizer code};
         ```
      6) In the pubnames.ntf database, select **Shared Elements/Subforms**.
      7) Open the $PersonInheritableSchema form.
      8) Select the **HTTPPassword** field.
      9) Select the Input Translation event.
      10) Paste the copied source code. Make sure that the pasted code is displayed before any other code in this event.
      11) Save the form.
   b. Copy the Enter Password button code:
      1) In the pwsync_install_r8.nsf database, select **Shared Elements/Subforms**.
      2) Open the $PersonInheritableSchema subform.
      3) Select **Enter Password** (near the bottom of the form).
      4) Select the **Click** event and make sure that the **Run** field is set to client.
      5) Copy the lines that start with

         ```
         REM {start of IDI Password Synchronizer code};
         ```

         and end with

         ```
         REM {end of IDI Password Synchronizer code};
         ```
      6) In the pubnames.ntf database, select **Shared Elements/Subforms**.
      7) Open the $PersonInheritableSchema form.

8) Select **Enter Password**.
9) Select the **Click** event. Set the **Run** field on the right side to `client`.
10) Paste the copied source code. Make sure that the code is displayed:
    - After the piece of code where the received password `tmpPassword` gets verified.
    - Before the code that refreshes all the document fields. For example: `@Command([ViewRefreshFields]);`
11) Save the form.

c. Copy the**FullName** field code:
1) In the `pwsync_install_r8.nsf` database, select **Shared Elements/Subforms**.
2) Open the `$PersonInheritableSchema` subform.
3) Select **FullName** field.
4) Select the `Input Validation` event.
5) Copy the lines that start with
```
REM {start of IDI Password Synchronizer code};
```
and end with
```
REM {end of IDI Password Synchronizer code};
```
6) In the `pubnames.ntf` database, select **Shared Elements/Subforms**.
7) Open the `$PersonInheritableSchema` form.
8) Select **FullName** field.
9) Select the `Input Validation` event.
10) Paste the copied source code before any other code in this event.
11) Save the form.

What to do next

"Updating the admin4.ntf template design"

Updating the admin4.ntf template design

You must edit the `admin4.ntf` template to make necessary changes to the `admin4.nsf` database.

Procedure

1. In the Lotus Domino Designer, open the `admin4.ntf` template database and `pwsync_install_r8.nsf` database.
2. Copy the **IDIPWSyncAdminRequestAgent** agent:
   a. In `pwsync_install_r8.nsf`, select **Code/Agents**.
   b. Select **IDIPWSyncAdminRequestAgent**.
   c. Right-click the selected agent and select **Copy**.
   d. In `admin4.ntf`, select **Code/Agents**.
   e. Select **Edit** > **Paste** to paste the selected agent.
3. Configure the **IDIPWSyncAdminRequestAgent**:
   a. Open the **IDIPWSyncAdminRequestAgent** agent.
   b. Select **Edit** > **Properties**.
   c. Click **Edit settings** from the Run time section of the Agent dialog box.
   d. In the **Run on** field, select the name of the current Domino Server.

e. Click **OK**.
f. Close the Agent dialog box.
g. Select **File** > **Save** to save the new agent settings.

   You might get a `You do not have execution access privileges for agent 'IDIPWSyncAdminRequestAgent' on 'TDITest/IBM'; it will not run` warning message. The Domino account that you use currently in the Domino Designer cannot `sign or run unrestricted methods and operations` on the Domino Server. Therefore, you must sign the agents with a dedicated signer.

What to do next

"Signing the agents with a signer"

Signing the agents with a signer

The agents of the Password Synchronizer must run restricted operations such as network access or file system access. Therefore, agents must be signed by the person who has the `sign or run unrestricted methods and operations` privilege.

Procedure

1. Find the signer name that is listed in the **Sign or run unrestricted methods and operations** field on the **Security** tab of the Server document. To access the Server document:
   a. Start the Domino Administrator.
   b. Select **Configuration**.
   c. In the left navigation panel, select **Server** > **Current Server Document**.

   If there are no existing accounts with this privilege, you must add a signer. For more information, see "Creating a signer for the Password Synchronizer agents" on page 42. Provide the `Sign or run unrestricted methods and operations` privilege only to the trusted accounts. The signer must have `Manager` access to the `pubnames.ntf` and `admin4.ntf` templates.
2. Open the Domino Designer.
3. To switch to the ID of the signer, go to **File** > **Security** > **Switch ID**.
4. Open the `pubnames.ntf` template.
5. Select **Code** > **Agents** to open the list of all agents.
6. From the list of agents, select **IDIPWSyncClientAgent**.
7. Click **Sign** to sign the agent with the current ID.
8. From the list of agents, select **IDIPWSyncWebAgent**.
9. Select **Code** > **Agents** to open the list of all agents.
10. From the list of agents, select **IDIPWSyncAdminAgent**.
11. Click**Sign**.
12. To switch to the ID that you were using previously, select **File** > **Security** > **Switch ID**.

What to do next

"Refreshing names.nsf database design" on page 51

Refreshing names.nsf database design

You must refresh the `names.nsf` database to apply the changes from the template to the existing database.

Procedure

1. From the Domino Administrator, click the **Files** tab.
2. Select the `names.nsf` database.
3. Go to **File** > **Application** > **Refresh Design**.
4. Select the name of your server from the **With Design from Server** list.
5. Click **OK**.
6. Click **Yes** to continue.

What to do next

"Refreshing the design of the admin4.nsf database"

Refreshing the design of the admin4.nsf database

You must refresh the `admin4.nsf` database to apply the changes from the template to the existing database.

Procedure

1. From the Domino Administrator, click the **Files** tab.
2. Select the `admin4.nsf` database.
3. Select **File** > **Application** > **Refresh Design**.
4. Select the name of your server from the **With Design from Server** list.
5. Click **OK**.
6. Click **Yes** to proceed.

What to do next

"Setting up secret key encryption infrastructure"

Setting up secret key encryption infrastructure

The Java agents use the `idipwsync.nsf` database to store documents that are required for further password processing. You must encrypt the documents to protect them in this database.

Procedure

1. Generate a secret key:
   a. From the Lotus Domino Administrator, select **File** > **Security** > **User Security**.
   b. Select **Notes Data** > **Documents** from the left navigation panel.
   c. Click **New Secret Key**.
   d. Enter **IDIPWSync** as secret key name and click **OK**.
   e. Click **Other Actions** and select **Export Secret Key**.
   f. Enter a password to protect the exported secret key.

      Note: This step is optional.
   g. Save the key in a file named `idipwsync.key`.

h. Click **Close**.

2. Import the secret key in the Domino Server ID file:
   a. Stop the Domino Server.
   b. In Lotus Domino Administrator, select **File** > **Security** > **Switch ID**.
   c. Open the `server.id` file for the Domino Server. You must use either a Lotus Domino Administrator that is installed on the Domino Server system, or copy the `server.id` file to the system where the Lotus Domino Administrator is installed. The `server.id` file is saved at: *`domino_data_directory`*
   d. Select **File** > **Security** > **User Security**.
   e. Select **Notes Data** > **Documents** from the left navigation panel.
   f. Click **Other Actions** and select **Import Secret Key**.
   g. Open the `idipwsync.key` file.
   h. If the file is protected by a password, enter the password that was created when you exported the secret key. For more information about the password, see Substep f of Step 1.
   i. Click **Accept** to import the secret key.
   j. Click **Close**.
   k. Select **File** > **Security** > **Switch ID** and switch back to the administrator ID file.
   l. If you edited a copy of the `server.id` file, copy it over the original `server.id` file in the *`domino_data_directory`* directory. Back up the original `server.id` before the file is overwritten with the new one.
   m. Start the Domino Server.
3. Import the secret key in the ID files of all the administrators or users to edit the `Person` documents and to change the HTTP passwords. For each of these administrators or users, do the following steps:
   a. From the Lotus Domino Administrator, select **File** > **Security** > **Switch ID**.
   b. Open the ID file of the administrator or user.
   c. Select **File** > **Security** > **User Security**.
   d. Select **Notes Data** > **Documents** from the left navigation panel.
   e. Click **Other Actions** and select **Import Secret Key**.
   f. Open the `idipwsync.key` file.
   g. If the file is protected by a password, enter the password that was created when you exported the secret key. See the Step 1 for information about how to generate a secret key.
   h. Click **Accept** to import the secret key.
   i. Click **Close**.

   Note: Administrator and user ID files must not contain the secret encryption key to change the **HTTP Password** field of `Person` documents.

What to do next

"Setting up port encryption" on page 53

Setting up port encryption

Port encryption encrypts the communication between the Lotus Domino Administrator and the Domino Server, bringing an additional layer of security to the network communication.

About this task

Port encryption is optional. The password is encrypted before you send it over the network with the secret key, regardless of whether the port encryption is used or not.

The available options are:

- Set up the Domino Server to encrypt the communication ports. You can set up by using the Lotus Notes clients. Only the server settings are configured. However, the setting affects the communication with all clients, including the regular users.
- Set up the Lotus Domino Administrator clients to encrypt the communication ports. This set up requires configuration of each of the Lotus Domino Administrator client that is used. However, this setting does not affect the other Notes clients if encryption is not necessary. Run the following steps to encrypt:

Procedure

1. Encrypt the Domino Server communication ports:
   a. From the Lotus Domino Administrator, select the **Configuration** tab.
   b. Select **Server** > **Setup Ports**.
   c. For each communication port in use, select the port from the **Communication ports** list and select **Encrypt network data**.
   d. Click **OK**.
   e. Restart the Domino Server for changes to take effect.
2. Encrypt the Lotus Domino Administrator communication ports. Run the following steps for each of the Lotus Domino Administrator client that is to be used for password changes:
   a. From the Lotus Domino Administrator, select **File** > **Preferences** > **User Preference**.
   b. Select **Ports** from the left navigation panel.
   c. For each communication port in use, select the port in the **Communication ports** list and select **Encrypt network data**.
   d. Click **OK**.
   e. Restart Lotus Domino Administrator for changes to take effect.

What to do next

"Setting up SSL for Domino HTTP Server"

Setting up SSL for Domino HTTP Server

SSL is necessary to secure the communication between the web browser and the Domino HTTP Server. If SSL is not set up, the password is transferred over the network in plain text.

About this task

For more information about setting up SSL, see the Lotus Domino Administrator help documentation.

What to do next

"Configuring Domino Server to automatically start and stop Java proxy"

Configuring Domino Server to automatically start and stop Java proxy

The Java Proxy runs in the JVM that is shipped with the Domino. The proxy starts as a Server Task when you start the Domino Server.

Procedure

1. Open the *domino_program_directory*/notes.ini file and find the ServerTasks property.
2. Add the following value at the end of the ServerTasks property:

   ```
   runjava com.ibm.di.plugin.domino.ProxyLoader
   ```

   The following is a sample ServerTasks property in notes.ini:

   ```
   ServerTasks=Update,Replica,Router,AMgr,AdminP,CalConn,Sched,HTTP,runjava
   com.ibm.di.plugin.domino.ProxyLoader
   ```

What to do next

"Configuring execution control list of Lotus Domino Administrator clients"

Configuring execution control list of Lotus Domino Administrator clients

You must configure each of the Lotus Domino Administrator client to enable the administrative password change.

Procedure

1. From the Lotus Domino Administrator, select **File** > **Security** > **User Security**.
2. Select **What Others Do** > **Using Workstation** in the left navigation panel.
3. In the **When code is signed by** list, select the name of your Domino Server, for example, serverName or certifierName. If the name of your Domino Server is missing, add it to this list.
4. Under **Allow access to:**, select **Current database**.
5. Under **Allow ability to:**, select **read other databases** and **Modify other databases**.
6. Click **OK**.

What to do next

"Configuring Access Control List" on page 55

Configuring Access Control List

You must create the IDIPWSync group in the Domino Directory and update the Access Control List (ACL) of theidipwsync.nsf database. Only members of the IDIPWSync group can access the idipwsync.nsf database.

Procedure

1. Create **IDIPWSync** group in the Domino Directory:
   a. From the Lotus Domino Administrator, click the **People & Groups** tab.
   b. In the left navigation panel, select **Domino Directories/ *your_domain's*Directory/Groups** where *your_domain* is the name of the Lotus Domino domain.
   c. Click **Add Group**.
   d. Type **IDIPWSync** in the **Group name** field.
   e. In the **Members** field, add all administrators or users who can change passwords by editing the Person documents.
   f. In the **Members** field, add the signer who signs the agents of the Password Synchronizer.
2. Update Access Control List of the idipwsync.nsf database:
   a. From the Lotus Domino Administrator, click the **Files** tab.
   b. Select the idipwsync.nsf database.
   c. Select **Database/Manage ACL** from the right panel.
   d. Click **Add** and select the **IDIPWSync** group.
   e. Select **Editor** from the **Access** list.
   f. Set the following options under **Attributes**:
      1) Select the **Delete documents** check box. You must also select the **Create documents**, **Read public documents**, and **Write public documents** check boxes. This selection is done automatically when the **Editor** access is selected.
      2) Clear the **Create private agents**, **Create personal folders/views**, **Create shared folders/views**, **Create LotusScript/Java agents**, **Replicate or copy documents** check boxes.
   g. Select **Default** from **Access Control List**.
   h. Set **Access** to **No Access**.
   i. Click **OK**.

   Note: After the idipwsync.nsf database ACL is changed, you cannot change the ACL from the Domino Server. For security reasons, the most restrictive settings are used. If a change of the ACL is necessary, the database must be opened locally and you must change the ACL as per the requirements.

What to do next

"Deleting pwsync_install_r8.nsf database"

Deleting pwsync_install_r8.nsf database

The pwsync_install_r8.nsf database is used only to distribute the required template objects. When the Domino HTTP plug-in is set up, the database is not required and you can delete the database from the Domino Server.

Procedure

1. From the Lotus Domino Administrator, click the **Files** tab.
2. Right-click the `pwsync_install_r8` database and select **Delete Database**.
3. Click **OK**.

Deploying on a Domino domain with multiple Domino Servers

On multiple Domino Servers, the Password Synchronizer is installed on all the Domino Servers, which is a Primary Domino Directory Server in the Domino domain.

About this task

The Password Synchronizer is not installed on the Domino Servers that are `configuration only` Directory Servers.

Procedure

1. On the Primary Domino Directory Server, which is the Administration Server for the Domino Directory, run full installation of the Password Synchronizer. For installation instructions, see "Deployment on a single Domino Server" on page 43.
2. For all the other Primary Domino Directory Servers, run the following steps:
   a. Run the Password Synchronizer installer to install the necessary files.
   b. Force replication with the first Primary Domino Directory Server where a full setup is run:
      1) From the Lotus Domino Administrator, click the **Server** tab.
      2) Select **Status**.
      3) In the right panel, select **Server** > **Replicate**.
      4) In the **Which server do you want to replicate with?** field, enter the name of the first Primary Domino Directory Server where a full setup is run.
      5) Click **Replicate**.
      6) Click **Done**.
   c. For this step and the following setup instructions, see the setup steps from the "Deployment on a single Domino Server" on page 43 topic:

      Skip steps 1, 2, 3 and 6. The Domino Directory replication propagates the design updates from the first Primary Domino Directory Server where a full setup is run.
   d. Skip steps 4 and 7. The `IDIPWSyncAdminRequestAgent` is triggered only on the Administration Server for the Domino Directory.
   e. Run step 8, but skip the creation of a secret key step. Use the secret key that is created when you set up the Password Synchronizer on the first Primary Domino Directory Server.
   f. Run steps 9, 10, and 11.
   g. Skip step 12.
   h. Run step 13, but skip the creation of the **IDIPWSync** group step.
   i. Run step 14.

Partial deployment of the Password Synchronizer

You can install two features of the Domino Password Synchronizer such as intercepting administrative password resets and user password changes, which are independent of each other.

The Domino Password Synchronizer intercepts both:

- Administrative password resets when an administrator edits a `Person` document of the users.
- Normal password changes when a user changes the password by using any of the following methods:
  - Through the **Change Password** web form from the `domcfg.nsf` file
  - Through the iNotes)

For the partial deployment of the Password Synchronizer:

1. Install the Domino Password Synchronizer that intercepts only the administrative password resets, by using the Lotus Domino Administrator or through the web browser interface.

   To install a Password Synchronizer that intercepts only the administrative password resets, run the steps from "Deployment on a single Domino Server" on page 43, except steps 4 and 7.

   On step 5, skip the task of opening the `admin4.ntf` file and signing the **IDIPWSyncAdminRequestAgent** agent.

   Steps 4 and Step 7 installs the agent that intercepts normal user password changes.

   When you install the solution on a Domino domain with multiple Domino Servers, follow the instructions in "Deploying on a Domino domain with multiple Domino Servers" on page 56. Skip the steps 4 and 7 when you install the synchronizer on the Administration Server.
2. Install the Domino Password Synchronizer that intercepts only the normal user password changes by using the **Change Password** web form from `domcfg.nsf` or through iNotes.

   To install a Password Synchronizer that intercepts only the normal user password changes, run the following steps from the "Deployment on a single Domino Server" on page 43 topic: 1, 2, 4, 5, 7, 10, 11 and 14. On step 5, skip opening the `pubnames.ntf` and signing the **IDIPWSyncClientAgent** and **IDIPWSyncWebAgent** agents. The steps 3, 6, 8, 9, 12, and 13 are skipped because they are necessary only for interception of administrative password resets.

   When you install the solution on a Domino Domain with multiple Domino Servers, run the previous subset of installation steps on the Primary Domino Directory Server. The Primary Domino Directory Server is the Administration Server for the Domino Directory. No installation on the other Domino Servers in the Domino Domain is necessary.

Deployment procedure without a dedicated agent signer

In the versions prior to IBM Security Directory Integrator 7.1, signer account is not available. Instead, you must provide the `Sign or run unrestricted methods and operations` privilege to the `IDIPWSync` group.

To minimize the scope of required privileges in the IBM Security Directory Integrator V 7.1, the deployment procedure is modified to assign a dedicated

signer account. Dedicated signer can sign the agents of the Password Synchronizer. The pre- V 7.1 deployment procedure did not have such signer account. The pre- V 7.1 deployment procedure is still supported.

You must make the following modification to the "Deployment on a single Domino Server" on page 43 deployment procedure:

1. Skip Step 5, the step for signing the agents.
2. Skip the task of adding the signer account to the `IDIPWSync` group in Step 13.
3. After Step 13, run the following steps:

   Note: In a multi-server topology, apply these steps on all the servers, where you deploy the Password Synchronizer.

   a. From the Domino Administrator, click the **Files** tab.
   b. In the **Run unrestricted methods and operations** field, add the `IDIPWSync` group.
   c. Click **Save & Close**.

Usage of the Password Synchronizer

The Domino HTTP Password Synchronizer modifies the `names.nsf` database and the `admin4.nsf` database to manage the password retrieval and the password change administration requests.

The Domino HTTP Password Synchronizer modifies the `names.nsf` database, adding custom Java agents and custom code in certain hooks.

The code in the hooks is run by the Domino when a Person document is saved in `names.nsf`. The code retrieves the HTTP password before it is hashed and sends the value to the Password Synchronizer proxy process by using the custom Java code.

The Domino HTTP Password Synchronizer modifies the `admin4.nsf` database by adding a custom Java agent. The agent is configured as a scheduled agent that is triggered after documents are created or modified in the administration requests database `admin4.nsf`. The agent is not triggered immediately after a document is created or modified in the`admin4.nsf`database, but after a 5- minutes to 30-minutes interval, depending on the decision of the Agent Manager process in Domino. When triggered, the agent searches the admin request for successfully processed `Change HTTP password in Domino Directory`administration requests. The agent retrieves the new passwords from the requests and sends the password data to the Password Synchronizer proxy process.

The proxy process starts a Password Store component to encrypt and store the password data so that it can be retrieved by the IBM Security Directory Integrator.

Password change mechanisms

When you use the Domino HTTP Password Synchronizer, only the following password change mechanisms are intercepted by:

- Editing the Person document through the Lotus Domino Administrator
- Editing the Person document through the web browser
- Using the Change Password web form from `domcfg.nsf`
- Using iNotes

Note: Password changes that are achieved through any other interfaces are not intercepted. For example, if passwords are changed through LDAP or iNotes with password synchronization enabled, the Domino HTTP Password Synchronizer is not triggered. And also, the password changes are not synchronized.

Secure password transfer

Secure communication is achieved by enabling SSL for the web-based mechanisms for the password change. You can edit the Person documents through the browser by using the Change Password web form or the iNotes.

When you edit the Person documents through the Lotus Domino Administrator client, communication is secured by enabling port encryption in the Domino.

For instructions on how to configure port encryption for the Domino, see "Deployment on a single Domino Server" on page 43.

Solution workflow

You must configure the proxy process, which starts when you start the Domino Server, to instantiate a Password Store such as LDAP and JMS. The proxy process accepts the TCP/IP connections, receives user ID and password data, and starts the Password Store to store data.

Person document change through the Lotus Domino Administrator

The Person document contains the custom code in the `names.nsf` database. When you save the Person document, the code is run on the Lotus Domino Administrator client.

If the HTTP password is changed, the following sequence of actions is run:

1. The password is retrieved before it is hashed.
2. A new document is created and the password is stored in this document. The document is saved in a database on the server.
3. An agent is started on the server to pass the ID of the newly created document. The agent reads and deletes the password data from the document. The agent then sends the document to the proxy process, which in turn sends data to the Password Store.
4. The Password Store returns the password that is not successfully stored. When the password is returned, all changes that are made to the Person document are rejected, including the HTTP Password field change.

Person document change through the Domino web browser interface

The Person document contains the custom code in the `names.nsf` database. When you save the Person document, the code is run on the Domino Server.

If the HTTP password is changed, the following sequence of actions is run:

1. When the Person document save is requested and the HTTP password value is changed, the custom Lotus formula code intercepts the plain text password. The password is then stored in a custom hidden field in the document.
2. The Lotus formula code starts an agent just before the document is saved.
3. The agent reads the password value from the hidden field. It deletes the value of this field, and sends the password to the Proxy Process, which in turn sends it to the Password Store.

4. The Password Store returns the password that is not successfully stored. When the password is returned, all changes that are made to the Person document are rejected, including the HTTP Password field change.

Note: In this scenario, the plain text password value is sent from the browser to the Domino web server when the web form is submitted. To protect the password on the wire, SSL is enabled on the Domino web server and the users use the HTTPS protocol from the browser.

Password change through the Password Change web form or through iNotes

Change of the HTTP password through the Password Change web form or the iNotes results in a `Change HTTP password in Domino Directory` admin request posted in the `admin4.nsf` database. The Admin Process processes this requests and changes the password in the Person document of the user.

The Password Synchronizer adds a custom Java agent in the `admin4.nsf` database. After an administration request document or a reply to an administration request document is added to the `admin4.nsf` database, the Java agent is scheduled to start by the Agent Manager. The Java agent is not started immediately but after some configurable interval chosen by the Agent Manager. Typically, the interval is 5-minutes to 30- minutes after a document is posted. When the agent is run, the following actions are taken:

- Retrieves processing of all admin requests, which are:
  - Of type `Change HTTP password in Domino Directory`.
  - Processed successfully by the Domino Admin Process. It has an attached reply document, which confirms that the password is changed by the Domino.

    If a password change request is not processed yet or is not successfully processed, the password change cannot be applied. Thus, there is no need for the Password Synchronizer to report it.
  - Processed successfully by the agent on a previous run of the agent.
- For each successfully processed password change admin request, the user identifier and the new password are retrieved and sent to the proxy process. The proxy process in turn sends data to the Password Store.

  If the Password Store returns the password that is successfully stored, the admin request is marked as processed. The request is not processed again by the agent on the next run. If the password is not successfully stored, the document is not marked as processed, so the agent processes it again on the next run.

Note: In this scenario, the plain text password value is sent from the browser to the Domino web server when the web form is submitted. To protect the password in transit, SSL is enabled on the Domino web server and users use the HTTPS protocol from the browser.

Migrating from version 7.1.1 to version 7.2

You must modify the configuration settings of the Domino HTTP Password Synchronizer and the Domino Server before you migrate.

About this task

You can skip the following steps if the IBM Security Directory Integrator version 7.1.1 is configured with a dedicated signer who has the `Sign or run unrestricted`

methods and operations privilege. For pre- version 7.2 deployment procedure, see "Deployment procedure without a dedicated agent signer" on page 57.

In the IBM Security Directory Integrator version 7.2, the Password Synchronizer agents are signed by a dedicated signer with the Sign or run unrestricted methods and operations privilege. The IDIPWSync group is not required to have this privilege.

Procedure

1. Sign the agents of the Password Synchronizer. See Step 5 in the "Deployment on a single Domino Server" on page 43 topic.
2. Refresh the designs of names.nsf and admin4.nsf. See Step 6 and Step 7 in the "Deployment on a single Domino Server" on page 43 topic.
3. Add the signer of the agents to the IDIPWSync group. See Step 13 in the "Deployment on a single Domino Server" on page 43 topic.
4. Remove the privilege to sign or run unrestricted methods and operations from the IDIPWSync group:
   a. From the Domino Administrator, click the **Configuration** tab.
   b. Select **Server** > **All Server Documents**.
   c. Select the document of the Server. If you have multiple Domino Servers, you must run all the steps for each of the servers.
   d. Click **Edit Server**.
   e. Click the **Security** tab.
   f. In the **Programmability Restrictions** section, remove the IDIPWSync group from the **Sign or run unrestricted methods and operations** field.
   g. Click **Save & Close**.

Chapter 8. Password Synchronizer for UNIX and Linux

The Password Synchronizer for UNIX and Linux intercepts password change events that originate from the tools, which are based on the UNIX and PAM enabled applications.

Overview

The Pluggable Authentication Modules (PAM) architecture on the UNIX systems, provides an extendable design to enable customized behavior, which is based on user authentication. The PAM Password Synchronizer plug-in uses the UNIX PAM architecture to enable password change notifications to propagate to the IBM Security Directory Integrator plug-in Password Store.

The primary purpose of the PAM Password Synchronizer plug-in is to intercept password change events that originate from the tools that are based on the UNIX and PAM enabled applications, such as the **passwd** command.

Supported platforms

The PAM Password Synchronizer is available on the following platforms:

- Solaris 10 SPARC (32-bit and 64-bit)
- Solaris 11 SPARC (32-bit and 64-bit)
- AIX 6.1 (PPC-64)
- AIX 7.1 (PPC-64)
- RHEL ES/AS 5.0 (x86/x86 - 64)
- RHEL ES/AS 6.0 (x86/x86 – 64)
- SLES 10 (x86/x86 - 64)
- SLES 11 (x86/x86 - 64)
- RedFlag Data Center 5.0 SP1/Asianix 2.0 SP1

Notes:

1. On 64-bit x86 Linux, problems with the bundled JRE are experienced if the plug-in installation is attempted before the **prelink** utility is run by the **cron** utility for the first time. The plug-in installation fails with a message that states no JVM was found. Run the `/etc/cron.daily/prelink` script to resolve the issue and to allow the plug-in installation to proceed.
2. RHEL 5.0 has SELinux enabled by default. The SELinux keeps the host secure from malicious attacks. However, the default settings prevent some of the plug-in libraries from loading. To fix this problem, run the following command:

   ```
   find TDI_install_dir/jvm/jre/bin TDI_install_dir/pwd_plugins/PAM -name
   '*.so' -exec chcon -t textrel_shlib_t {} \;
   ```

Deployment and configuration

Use the template configuration file at `TDI_install_dir/pwd_plugins/pam/pwsync.props` to configure the PAM Password Synchronizer.

You can install the Password Synchronizer by using the IBM Security Directory Integrator installer wizard. After the installation is complete, use the instructions in the following sections for the deployment steps that are required for the PAM Password Synchronizer.

Password Synchronizer registration for UNIX and Linux plug-in within PAM

To register the plug-in, edit the PAM configuration file. The following table shows the standard location of PAM configuration files on various platforms. Your individual PAM configuration causes the PAM password module configuration to be a different file. Check with your system administrator if either these files do not exist, or if the added Password Synchronization module is not started.

Note: The `/etc/pam.conf` configuration file is used in the older versions of PAM on UNIX. This file is now deprecated and all PAM configuration files are now in `/etc/pam.d` for the modules that rely on PAM. You must store the PAM configuration file for the password change module in this directory.

The primary component of external system configuration is the PAM configuration file. Since the purpose of the plug-in is to intercept password events, add a registration line as shown in the following table to the PAM configuration file. If the PAM module is being stacked with other PAM modules, the Security module is the last module in the stack. This way, The module can be sure that previous `required` modules returns a success status before PAM calls the Security module.

Operating System	PAM Configuration File	PAM plug-in registration line
AIX 6.1 or greater	`/etc/pam.conf`	`passwd password is required. TDI_Plugin_Root/pwd_plugins/pam/libpamtivoli.so use_first_pass TDI_Plugin_Root/pwd_plugins/pam/pwsync.props`
Solaris 10	`/etc/pam.conf` or `/etc/pam.d/system-auth`	`Other password is required. TDI_Plugin_Root/pwd_plugins/pam/libpamtivoli.so use_first_pass TDI_Plugin_Root/pwd_plugins/pam/pwsync.props`

Linux	`/etc/pam.confor /etc/pam.d/system-auth (RHEL 5)`
	
	`/etc/pam.confor /etc/pam.d/password (SLES 9)`
	
	`/etc/pam.conf or /etc/pam.d/common-password (SLES 10)`
		`Password is required TDI_Plugin_Root/pwd_plugins/pam/libpamtivoli.so use_first_pass TDI_Plugin_Root/pwd_plugins/pam/pwsync.props`

Note: If the system is 64 bit and the applications that rely on PAM such as `passwd` are also 64 bit, use `libpamtivoli_64` instead of `libpamtivoli`.

Note: The preceding table lists `system-auth` as the PAM configuration file in the `/etc/pam.d` directory. The `/etc/pam.d/passwd` file is the main configuration file to set password and change password. On most operating systems, the standard PAM installations set up the`/etc/pam.d/passwd` file to use the `/etc/pam.d/system-auth` file. This set up defines the actual PAM modules to set the password and change the password. On RHEL 4, the delegation in the `/etc/pam.d/passwd` file can be as shown in the following example:

```
password required pam_stack.so service=system-auth
```

If your PAM /etc/pam.d/passwd configuration file is delegated to system-auth, you must add the configuration entry into the /etc/pam.d/system-auth file.

The exceptions to the placement of the Security module last in the stack are:

- If there are modules above the Security module, and are marked as sufficient, you must change the module to required. This change ensures that the Security module is called. For example, on RHEL 4 Linux, the pam_unix module is marked as sufficient in the standard installation. If the result of the pam_unix module is successful, no proceeding password modules is started. To ensure that the Security module is called, you must change the pam_unix to required and it must come before the Security module in the stack.
- If you have modules only for error processing, such as pam_deny, modules must follow the Security module, and the Security module must be marked as sufficient.

The PAM pluggable architecture allows the modules to be stacked. You can create the custom solution that allows several PAM Password Synchronizers to be installed on the same system. Each PAM plug-in requires a separate Java Proxy process. Each Java Proxy must listen on a separate port. Use the different pwsync.props configuration files. The files must be in a different folder because authentication is taking place in that folder.

Configuration of PAM Password Synchronizer

The PAM plug-in has a template configuration file that is installed at *TDI_install_dir*/pwd_plugins/pam/pwsync.props. When the PAM plug-in is initialized, the configuration file is set as the last parameter of the registration line of the plug-in. Some parameters of the configuration file are shared between the plug-in and the Java Proxy. The plug-in recognizes some of the properties that are described in theChapter 3, “Common configuration and utilities of password synchronization plug-ins,” on page 13 topic.

The syncBase and **logFile** properties are irrelevant to the plug-in, and are ignored. The reason for ignoring thesyncBase property is that the PAM cannot always provide a dn-like naming of arrived users. The reason for ignoring the **logFile** property is that the PAM plug-in logs by using the native UNIX syslog daemon.

Select the Password Store of your choice by setting the correct class name in the **syncClass** parameter.

Chapter 9. LDAP Password Store

The LDAP Password Store stores the intercepted user passwords in an LDAP Directory Server.

Supported directories

The LDAP Password Store is available on the following directories:

- IBM Security Directory Server
- Microsoft Active Directory
- Sun Directory Server

Installation of LDAP Password Store

The IBM Security Directory Integrator LDAP Password Store provides the function necessary to store the intercepted user passwords in an LDAP Directory Server (repository or data source).

You can create the LDAP Password Store component to support a number of IBM Security Directory Integrator plug-ins that intercept password changes for various products or platforms.

The following Password Synchronizers are available to intercept password change request from a user:

IBM Security Directory Integrator Password Synchronizer for Windows
Intercepts the Windows login password change.

IBM Security Directory Server Password Synchronizer for Windows, UNIX, and Linux Intercepts the IBM Security Directory Server password change.

Sun Directory Server Password Synchronizer for Windows, UNIX, and Linux
Intercepts the Sun Directory Server password change.

Domino Password Synchronizer for Windows, UNIX, and Linux
Intercepts changes of the HTTP password for Lotus Notes users.

IBM Security Directory Integrator Password Synchronizer for UNIX, and Linux
Intercepts changes of the UNIX and Linux user passwords.

All the plug-ins use the LDAP Password Store function for secure propagation of the change to another LDAP Server. In the LDAP Server, the password change is manipulated by the IBM Security Directory Integrator AssemblyLine.

You can configure the LDAP Password Store by using the properties files that enable:

- Specification of keystore files, certificates, and credentials for SSL connections.
- Asymmetric encryption of password data.

The property files also accommodate control of trace log and limited control of attributes that are used to store the captured passwords.

Prerequisites

- The LDAP Password Store requires a minimum JRE 1.5. IBM Security Directory Integrator bundles Java 7.0.4 JRE.

- Use the IBM Security Directory Integrator product installer to install the password synchronization plug-ins.

Setting up the LDAP Server

You can use the IBM Security Directory Server to set up a sample environment. To set up, identify a container where the object class with user ID and password is found or created.

Procedure

1. Define the suffix.
   a. Select **Start** > **Programs** > **IBM Security Directory Server** *x.x* > **Directory Configuration**.
   b. Select **Manage suffixes** from the left pane.
   c. In the **Suffix DN** field add the suffix under which you store the password information. For example, `o=ibm,c=us`.
   d. Click **Add**. The new suffix is shown in the **Current suffix DNs** list.
   e. Click **OK**.
   f. Close the **Directory Configuration** tool.
2. Add the suffix data.
   a. Restart the IBM Security Directory Server.
   b. From the IBM Security Directory Server Web Administration Tool, select **Directory management** > **Manage entries**.
   c. Click **Add**.
   d. Select **organization** from the structural object class list.
   e. Click **Next**.
   f. From the Select auxiliary object classes window, click **Next**.
   g. From the Enter the attributes window, clear the value in the **Parent DN** field.
   h. Specify the suffix name into the **Relative DN** field. For example, `o=ibm,c=us`.
   i. Enter the organization name into the **o** field (**ibm** in the previous example).
   j. Click **Finish**.
3. Add the domain object.
   a. From the IBM Security Directory Server Web Administration Tool, select **Directory management** > **Manage entries**.
   b. Select the suffix that are previously created in the previous step, that is, **o=ibm, c=us**.
   c. Click **Add.**
   d. Select **domain** from the **structural object class** list.
   e. Click **Next**.
   f. In the Select auxiliary object classes window, click **Next**.
   g. Enter the domain name in the **Relative DN** field. For example, **dc=mydomain**.
   h. Enter the domain name in the **dc** field (**mydomain** in the previous example).
   i. Click **Finish**.

Note: The domain and suffix entered must also be included in the `pwsync.props` file along with the other information. For configuration details, see "Configuration of LDAP Password Store" on page 70.

4. Define the `ibm-diPerson` object. From a system with IBM Security Directory Server client, run the following command from the *install_directory* as one line:

   ```
   ldapmodify -c -h LDAP Hostname -D admin DN -w admin PW  -f
   TDI_install_dir/pwd_plugins/etc/ibm-diPerson_oc.ldif
   ```

   Note: You might see the following messages:

   - `Attribute type '1.3.18.0.2.4.155' already exists, add operation failed.`
   - `Attribute type '0.9.2342.19200300.100.1.1' already exists, add operation failed.`

   You can ignore these messages. The messages indicate that the `secretKey` and `uid` attributes are already defined in your schema.

Modifying the schema of zLDAP

You must use a Technical Database Management (TDBM) server when you configure the LDAP Server on z/OS® to facilitate loading of the required LDIF files.

About this task

Note: The z/OS operating system is not supported in IBM Security Directory Integrator Version 7.2 onwards.

Note: For setup instructions, see *z/OS Integrated Security Services LDAP Server Administration and Use* in the IBM z/OS online product library.

Procedure

1. Definition of a suffix involves generating a new LDAP config and server JCL jobs. The LDAP administrator and the system programmers are responsible to generate the jobs.
2. While the suffix data is not required to be added, the base schema is not required to be added to the defined suffix. The two base schema LDIF files in the `/usr/lpp/ldap/etc` directory, `schema.IBM.ldif` and `schema.user.ldif`, must be customized with the suffix from Step 1, and then loaded.
3. If required, you can define a domain by creating and loading an LDIF file that defines the domain.
4. Before the `ibm-diPerson_z.ldif` file can be loaded into the LDAP server, it must be customized to include the suffix created in Step 1. This process involves adding the suffix to the end of DN. For example, if your suffix is `o=ibm,c=us`, the DN lines changes from `dn:cn=schema` to `dn:cn=schema,o=ibm,c=us`.

Modifying the schema of Sun Directory Server and Active Directory

You must modify the schema of the Sun Directory Server and the Active Directory with necessary configuration before you install the LDAP Password Store.

Procedure

1. Modify the LDAP schema of the Sun Directory Server. Run the following command as one line:

   ```
   ldapmodify -c -h LDAP Hostname -D admin DN -w admin PW-f
   TDI_install_dir/pwd_plugins/etc/ibm-diPersonForSunDS.ldif
   ```

2. Modify the LDAP schema of the Active Directory:
   a. Enable the Active Directory schema modification by editing the Windows registry key:

      ```
      HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\NTDS\Parameters
      ```

      Add a `REG_DWORD` value named `Schema Update Allowed` with a value of `1` or any value greater than `0`.
   b. Run the following command to update the LDAP schema:

      ```
      ldifde -i -f TDI_install_dir/pwd_plugins/etc/ibm-diPersonSchemaForAD.ldif
      ```

   c. Open the Microsoft Management Console.
   d. Create a new Organizational Unit to store the changed passwords.
   e. Get the Distinguished Name of the Organizational Unit by using one of the following tools: `ldifde.exe`, `csvde.exe`, or `dsquery.exe`. The names are used when you configure the suffix of the LDAP Password Store in the `pwsync.props` file.

Configuration of LDAP Password Store

You must set the properties of LDAP Password Store in the `pwsync.props` configuration file.

By default, there is one configuration file for each of the plug-ins. For example, *TDI_install_dir*`/pwd_plugins/tds/pwsync.props` for the IBM Security Directory Server Password plug-in. The LDAP Password Store is therefore configured in the `pwsync.props` file of the plug-in.

Note: In the configuration file, you must manually encrypt each password property. You can use the `encryptPasswd` utility for encryption. This utility uses a symmetric algorithm for encryption of the passwords. Make sure that the `pwsync.props` file is readable only by the trusted system users.

The `encryptPasswd` utility requires that the password is passed as a parameter. The encrypted password is printed on the standard output.
For a complete list of the configuration parameters and their explanation, see Chapter 3, “Common configuration and utilities of password synchronization plug-ins,” on page 13.

The class for the LDAP Password Store is:
`com.ibm.di.plugin.pwstore.ldap.LDAPPasswordStore`.

The following example shows a completed properties file for an SSL connection and password encryption:

```
#IBM Directory Integrator LDAP Password Store Settings with Encoded Passwords
#Tue Jul 30 08:21:20 EDT 2002
ldap.hostname=gbdthst1
ldap.port=636
ldap.waitForStore=true
ldap.admindn=cn=root
ldap.password=0c0bf0e3146b
ldap.ssl=true
```

```
ldap.suffix=dc=carnd11,o=ibm,c=us
encrypt=true
encryptKeyStoreFilePath=c:\sync\cryptokeys.jks
encryptKeyStoreFilePassword=0c0bf0e3146b
encryptKeyStoreCertificate=cryptoCertName
encryptKeyPassword=0c0bf0e3146b
```

Notes:

1. To disable SSL, select a non-SSL port, for example, 389, and set `ssl=false`.
2. To disable asymmetric password encryption, set `encrypt=false`. When `encrypt=false`, any value in `encryptKeyStoreFilePath`, `encryptKeyStoreFilePassword`, `encryptKeyStoreCertificate`, and `encryptKeyPassword` is ignored.
3. The suffix keyword is used to identify the container where the objects that contain the user ID and new password value are found.
4. There are some additional optional keywords that you can use to override the default object class and attribute definitions. You can add the following properties name in the `pwsync.props` files and their associated default values:

   ldap.schemaPersonObjectName
   ibm-diPerson

   ldap.schemaUseridAttributeName
   ibm-diUserId

   ldap.schemaPasswordAttributeName
   ibm-diPassword
5. Another optional attribute, **ldap.delayMillis**, is used when the **ldap.waitForStore** property is set to false. When **ldap.waitForStore=false**, the **ldap.delayMillis** specifies the number of milliseconds of delay before the storage. A deadlock can occur when the:
   - IBM Security Directory Integrator Password Synchronizer for the Windows system is configured to use the LDAP Password Store.
   - LDAP Password Store is configured to store into the Active Directory on the same system where the Password Synchronizer is installed.

   To avoid the deadlock, use this asynchronous mode of operation. In an asynchronous mode **ldap.waitForStore=false**, the password catcher code that communicates with the Windows system returns control to the Windows. After a short delay, the password store code that is running a separate thread attempts to store the password update into the Active Directory. If **ldap.waitForStore=false** and no value is specified for **ldap.delayMillis**, then a default of **ldap.delayMillis=2000** is used. In this configuration, any Password Store failures are reported by using the log file, which is specified in the **logFilePath** property.

Password encryption

Encryption of password values is supported by both the LDAP Password Store and the JMS Password Store.

By default, the encryption is disabled. To turn it on, set the **encrypt** property to true.

When encryption is used, the **encryptKeyStoreFilePath**, **encryptKeyStoreFilePassword**, and **encryptKeyStoreCertifcate** property values must also be set. The **encryptKeyPassword** property must be set if you are using the

LDAP Password Store. The **encryptKeyPassword** property is irrelevant for the rest of the Password Stores. The password encryption and decryption functions use the RSA algorithm. The following example shows configuration properties for the encryption function:

```
encryptKeyStoreFilePath=path to the key store file
encryptKeyStoreFilePassword=password of the key store file; encoded with
the "encryptPasswd" tool
encryptKeyStoreCertifcate=the alias of the public key certificate in
the key store
encryptKeyPassword=password of the private key; encoded with
the "encryptPasswd" tool
```

You can create and manage keystore files and the public or private keys with the `keytool` and `iKeyman` JRE utilities.

For more information about keystores and keytool, see:

- "Keystore and truststore management" section in the *Installing and Administering*.
- http://www-128.ibm.com/developerworks/websphere/techjournal/0502_benantar/0502_benantar.html#sec2
- http://docs.oracle.com/javase/1.6.0/docs/tooldocs/windows/keytool.html

The `java.security` file in the *install_directory*`/jvm/jre/lib/security` directory is set up to contain a reference to the security provider `com.ibm.crypto.provider.IBMJCE`. The following example shows the relevant portion of the file:

```
:
:
:

# List of providers and their preference orders :
#
security.provider.1=sun.security.provider.Sun
security.provider.2=com.ibm.crypto.provider.IBMJCE

:
:
:
```

An example AssemblyLine, which demonstrates the decryption of captured passwords, is included in the IBM Security Directory Integrator installation. The AssemblyLine and a readme file are stored in the *TDI_install_dir*`/examples/pwsync_decryption/` directory where *TDI_install_dir* is the installation directory of the IBM Security Directory Integrator.

Notes:

1. RSA is an asymmetric encryption algorithm, which uses a public key to encrypt and its associated private key to decrypt. Because you need the public key for encryption, distribute only the public key in the keystore file of the Password Store. This information is not relevant to the LDAP Password Store because it decrypts the already stored password values to determine which password to delete. Therefore, the private key is also required.
2. The keystore files contain sensitive data and must be properly protected by using file system permissions.

Password Store usage

For each user, whose password is intercepted, the LDAP Password Store maintains an LDAP entry in the storage LDAP directory. The directory is the container where the storage entries are added and modified and is specified by the suffix property of the LDAP Password Store.

The entry in the storage directory always contains the passwords that are currently used by the original user on the target system. The LDAP Password Store updates the state of the entry in the directory whenever the LDAP Password Store receives notification for password update from the Password Synchronizer.

The LDAP Password Store receives the following data from the Password Synchronizer:

- User identifier (a string)
- Type of the password modification
- A list of password values

User identifier

The user identifier is used for the relative distinguished name of the entry, which is stored in the LDAP directory. For example, if the user identifier is `john` and the suffix property value is `dc=somedc, o=ibm, c=us`, the distinguished name of the entry that is stored is `ibm-diUserId=john, dc=somedc, o=ibm, c=us`.

The Password Synchronizer reports the LDAP distinguished name of the user for which the password is changed. For example, `cn=john, o=somecompany, c=us`. The LDAP Password Store takes the first element of the distinguished name `john` to construct the distinguished name of the entry on the storage LDAP directory. For example, `ibm-diUserId=john, dc=somedc, o=ibm, c=us`. Therefore, the context information such as department, company, and country is lost. If there are two individuals on the target system with same names but in different departments, the Password Store cannot distinguish the names. However, the Password Store works as if the names represent the same person. For example, `cn=Kyle Nguyen, ou=dept_1, o=ibm, c=us` and `cn=Kyle Nguyen, ou=dept_2, o=ibm, c=us`.

Type of password modification and list of password values

The type of password modification indicates whether the password values are replaced, or new values are added, or certain values are deleted. Using this information and the list of passwords that represents the change, the Password Store duplicates the change on the entry in the storage directory.

The type of password modification is valid only when the password has multiple values, for example, the IBM Security Directory Server or the Sun Directory Server. When the passwords on the target system are single-valued, for example, Windows, the password modification type is always: `replace`

When the password, with all its values, is deleted from the target system, the entry in the storage directory is modified. The password cannot have value for the LDAP attribute that is used to store the passwords.

Possible password retrieval from IBM Security Directory Integrator

You can retrieve passwords that are stored in an LDAP Server by the LDAP Password Store. A `Changelog Connector` is configured to listen to changes in the LDAP Directory that is used for the storage. Whenever the connector detects that an entry is added or modified in the Password Store container, it starts an AssemblyLine. You can start the AssemblyLine by passing the identification of the modified entry. The AssemblyLine uses an LDAP Connector to read the modified entry, decrypts the updated password values, and propagates the values to systems that must be synchronized.

Chapter 10. JMS Password Store

You can use the JMS Password Store to store the intercepted user passwords in a JMS queue from where the JMS clients read the passwords. The JMS Password Store was formerly known as the IBM WebSphere MQ Everyplace Password Store.

The JMS Password Store provides the necessary function to store the intercepted user passwords in a JMS provider Queue from where any JMS client can read them. For example, the IBM Security Directory Integrator.

The JMS Password Store package consists of the Storage Component and the `JMS Password Store Connector`. For more information about the JMS Password Store Connector, see *Reference*. The storage component is actually the Password Store, which is started by the Password Synchronizer. The `JMS Password Store Connector` is a specialized connector on the IBM Security Directory Integrator side that can retrieve passwords that are stored by the configured JMS Provider.

The class for this password store is:
`com.ibm.di.plugin.pwstore.jms.JMSPasswordStore`

Apache ActiveMQ driver

To use ActiveMQ as the JMS provider for the JMS Password Store component, set the **`jmsDriverClass`** property in **`pwsync.props`** to `com.ibm.di.plugin.pwstore.jms.driver.ActiveMQ`. The ActiveMQ driver has the following parameter.

`jms.broker` - the ActiveMQ server address such as protocol, IP address, and TCP port number. For example, `tcp://activeMQhost:61616` or `ssl://activeMQhost:61617` to use SSL connection.

IBM WebSphere MQ Everyplace driver

The IBM WebSphere MQ Everyplace driver is responsible to create the IBM WebSphere MQ Everyplace Queue Manager and to retrieve the required connection objects.

To use IBM WebSphere MQ Everyplace as the JMS provider for the JMS Password Store component, the **`jmsDriverClass`** property in the `pwsync.props` file must be set to `com.ibm.di.plugin.pwstore.jms.driver.IBMMQe`.

Note: You must create an IBM WebSphere MQ Everyplace Queue Manager. To create it, use the IBM WebSphere MQ Everyplace `Configuration` utility, which is bundled with the Password Synchronizer. For more information about the IBM WebSphere MQ Everyplace `Configuration` utility, see "MQe Queue Manager setup" on page 82.

IBM WebSphere MQ driver

The IBM WebSphere MQ driver is responsible to establish the connection with the IBM WebSphere MQ JMS provider. To use IBM WebSphere MQ as the JMS provider for the JMS Password Store component, set the **`jmsDriverClass`** property in the `pwsync.props` to: `com.ibm.di.plugin.pwstore.jms.driver.IBMMQ`

The IBM WebSphere MQ driver has the following parameters:

jms.broker
The IBM WebSphere MQ server address such as IP address and TCP port number. For example, 192.168.113.54:1414

jms.serverChannel
The name of the server channel that is configured for the IBM WebSphere MQ server instance.

jms.qManager
The name of the Queue Manager that is defined for the IBM WebSphere MQ server instance.

jms.sslCipher
The cipher suite name that corresponds to the cipher selected when you configure the IBM WebSphere MQ server channel. For example, SSL_RSA_WITH_RC4128_MD5

jms.sslUseFlag
Specifies whether SSL is used on the connection to the IBM WebSphere MQ Server instance. Valid values are `true` and `false`.

If you use SSL with the IBM WebSphere MQ driver, use the properties in the SSL Java Properties table inChapter 3, "Common configuration and utilities of password synchronization plug-ins," on page 13 to establish a relationship between JMS Password Store and IBM WebSphere MQ.

See the IBM WebSphere MQ Server documentation for configuration information.

Microbroker driver

For users with existing Microbroker installations, a Microbroker driver is provided. This driver is responsible to establish a connection with the Microbroker provider. To use Microbroker as the JMS provider for the JMS Password Store component, set the **`jmsDriverClass`** property in the `pwsync.props` file to: `com.ibm.di.plugin.pwstore.jms.driver.IBMMB`.

The Microbroker driver has the following parameters:

jms.broker
The Microbroker server address such as IP address and TCP port number. For example, 9.126.6.120:1883.

jms.clientID
The client ID that is required,

Note: To use Microbroker as the JMS Password Store, a few Microbroker JAR files are needed. For a sample list of the required JAR files, see the "JMS Connector" section in *Reference*.

JMS script driver

The user-defined JMS script drivers are not supported by the JMS Password Store. The JavaScript engine is not bundled with the Password Synchronizer.

Password message security

You can transfer the messages that contain passwords between JMS Password Store and JMS Password as plain text messages, PKI encrypted messages, or PKCS7 encapsulated messages.

The JMS Password Store stores the password as a message on the JMS provider queue. You can send the message that contains the password as:

- Plain text messages

  The messages are transferred between JMS Password Store and JMS Password Store Connector as plain text. Therefore, no message-based security is applied.
- Pre- IBM Security Directory Integrator 6.1.1 PKI encrypted messages

  This feature is optional. When this option is used, a certificate from a `.jks` file is used to:
  - Encrypt the received messages by the JMS Password Store
  - Decrypt the messages by the JMS Password Store Connector

  Note: Starting from IBM Security Directory Integrator 6.1.1, this encryption is deprecated because PKCS7 encapsulation offers a more secure way to transfer messages, containing encryption.
- PKCS7 encapsulated messages

  Starting from IBM Security Directory Integrator 6.1.1, the JMS Password Store, and the JMS Password Store Connector support PKCS7, which includes both signing and encryption.

Using PKCS7 for encapsulation is optional. By default, it is turned off. If you want to use PKCS7, configure both JMS Password Store and JMS Password Store Connector to use PKCS7. However, when PKCS7 is used, the PKI encryption is not allowed because the PKCS7 supports encryption.

JMS Password Store configuration

You must set the properties of JMS Password Store in the `pwsync.props` configuration file.

The JMS Password Store properties are set in the `pwsync.props` general configuration file of the plug-ins. By default, there is one file per each plug-in, for example, *`TDI_install_dir`*`/pwd_plugins/tds/pwsync.props`.

Note: In the general configuration file, you must manually encrypt each password property. You can use the `encryptPasswd` utility to encrypt the password. This utility uses a symmetric algorithm for encryption of the passwords. Make sure that the `pwsync.props` file is readable only by trusted system users.

In the `encryptPasswd` utility, pass the password as a parameter. The encrypted password is printed on the standard output.
For more information about configuration parameters and `encryptPasswd` utility, see Chapter 3, “Common configuration and utilities of password synchronization plug-ins,” on page 13.

In the `pwsync.props` file:

- For a plain text message, set `encrypt=false`

- For a pre- IBM Security Directory Integrator 6.1.1 PKI encrypted message, set `encrypt=true`
- For a PKCS7 encapsulated message, set `pkcs7=true` and `encrypt=false`

  Note: Setting of `encrypt=true` and `pkcs7=true` is not valid. You can set either `encrypt` or `pkcs7` to true.

For more information about password message security, see "Password message security" on page 77.

The following example shows an extract of the JMS Password Store configuration section of the `pwsync.props` file:

```
#
# This is the configuration file of the Password Synchronizer.
# It is used by all parts of the Password Synchronizer: the Plug-in,
# the Proxy and the Password Store component.
#
# Enter (name)=(value) to set configuration properties.
#
# Follow the Java properties file format. Backslashes must be escaped:
# e.g. instead of 'c:\myfile.txt' type 'c:\\myfile.txt'.
#



# Executable (binary or shell script) used to start the Java Proxy.
# If this property is set, both 'jvmPath' and 'jvmClassPath' will be ignored.
proxyStartExe=C:\\Program Files\\IBM\\TDI\\V7.2/pwd_plugins/bin/startProxy.bat

# Port number, on which the Java Proxy listens for commands.
serverPort=18001

# The log file of the Plug-in part of the Password Synchronizer.
# If empty, no logging will be done.
logFile=C:\\Program Files\\IBM\\TDI\\V7.2/pwd_plugins/windows/plugin.log

# Whether to reject password changes if the Password Store is down.
checkRepository=true

# The log file of the Java Proxy part of the Password Synchronizer. If empty,
# no logging will be done.
javaLogFile=C:\\Program Files\\IBM\\TDI\\V7.2/pwd_plugins/windows/proxy.log

# Turn on debug logging for all parts of the Password Synchronizer.
debug=true

# Custom data that will be send with each password change.
# This string can be used to uniquely identify the machine or product that generates
# the changes (e.g. machine IP, application name and version).
#customData=machine1



#
# User filtering configuration
#

# A list of Windows groups. If a user is a member of some group on the list,
# the user will be accepted # by the user filter (assuming the user is not
# excluded by some of the exclude lists).
# Group names must be separated by semicolons. Redundant white-spaces are not allowed.
# includeGroups=



# A list of Windows groups. If a user is a member of some group on the list, the user
# will not be accepted
# by the user filter.
# Group names must be separated by semicolons. Redundant white-spaces are not allowed.
# excludeGroups=

# A list of DN suffixes. If a user's Distinguished Name matches some suffix on the list,
# the user will be accepted by the user filter
# (assuming the user is not excluded by some of the exclude lists).
```

```
# DN suffixes must be separated by semicolons. Redundant white-spaces are
# not allowed.
# includeDNs=

# A list of DN suffixes. If a user's Distinguished Name matches some suffix on
# the list, the user will not
# be accepted by the user filter.
# DN suffixes must be separated by semicolons. Redundant white-spaces are not allowed.
# excludeDNs=

# Types of the accounts for which password changes will be reported.
# It is a space-delimited list of account types. Recoginzed account types are:
# NORMAL_ACCOUNT
# TEMP_DUPLICATE_ACCOUNT
# INTERDOMAIN_TRUST_ACCOUNT
# WORKSTATION_TRUST_ACCOUNT
# SERVER_TRUST_ACCOUNT
#
# accountTypes=NORMAL_ACCOUNT




#
# The Password Store component
#
# Specify the full name of the Java class.
# Choose one of the following:
# com.ibm.di.plugin.pwstore.log.LogPasswordStore
# com.ibm.di.plugin.pwstore.jms.JMSPasswordStore
# com.ibm.di.plugin.pwstore.ldap.LDAPPasswordStore
#
# LogPasswordStore is for testing purposes only - you should NEVER use it in
# production environment.
#
syncClass=com.ibm.di.plugin.pwstore.log.LogPasswordStore




#
# Public key encryption of passwords
#
# encrypt=true
# encryptKeyStoreFilePath=
# encryptKeyStoreFilePassword=
# encryptKeyStoreCertificate=

# 'encryptKeyPassword' is required by the LDAP Password Store (the rest do not need it)
# encryptKeyPassword=




#
# PKCS7 encapsulation of passwords
#
# pkcs7=true
# pkcs7KeyStoreFilePath=
# pkcs7KeyStoreFilePassword=
# pkcs7MqeStoreCertificateAlias=
# pkcs7MqeConnectorCertificateAlias=




#
# SSL configuration properties
#
# javax.net.ssl.trustStore=
# javax.net.ssl.trustStorePassword=
# javax.net.ssl.trustStoreType=
# javax.net.ssl.keyStore=
# javax.net.ssl.keyStorePassword=
# javax.net.ssl.keyStoreType=



#
```

```
# LDAP Password Store Configuration
#

# LDAP server host
# ldap.hostname=localhost

# LDAP server port
# ldap.port=389

# LDAP bind dn
# ldap.admindn=cn=root

# LDAP bind password
# This field must be encoded. Use the 'encryptPasswd' utility.
# ldap.password=0c0bf0e3146b

# If set to true, password changes will be committed synchronously
# to the Password Store when a password change notification is received.
# The source of the password change will be blocked
# until the password change is written to the Password Store.
#
# If set to false, the commit will be asynchronous.
# Use the 'ldap.delayMillis' property to configure
# the time to wait before committing the password change.
# ldap.waitForStore=true

# Time to wait (in milliseconds), before committing the password change to the
# Password Store. Will be ignored if 'waitForStore' is set to true.
# ldap.delayMillis=2000

# Use SSL for LDAP communication.
# If set to true, JSSE must be configured (set the javax.net.ssl.trustStore and
# javax.net.ssl.keyStore properties).
# ldap.ssl=false

# Location in the LDAP directory tree, where the Password Synchronizer
# will store data.
# ldap.suffix=dc=carnd11,o=ibm,c=us

# Name of an LDAP object class used to hold information for a given user.
# ldap.schemaPersonObjectName=ibm-diPerson

# Name of an LDAP attribute which represents user identifier.
# This attribute must be a member of the object class specified by the
# 'ldap.schemaPersonObjectName' property.
# ldap.schemaUseridAttributeName=ibm-diUserId

# Name of an LDAP attribute which represents user password.
# This attribute must be a member of the object class specified by
# the 'ldap.schemaPersonObjectName' property.
# ldap.schemaPasswordAttributeName=ibm-diPassword




#
# MQe Password Store Configuration
#

# JMS driver, used to establish connecton to the message broker.
# jmsDriverClass=com.ibm.di.plugin.pwstore.jms.driver.IBMMQe

# The path to the .ini file of the MQe QueueManager.
# mqe.file.ini=

# The TCP/IP port that is used when the MQe Connector sends notifications to the
# Storage Component.
# mqe.notify.port=41002


#
# ActiveMQ Password Store Configuration
#

# JMS driver, used to establish connecton to the message broker.
# jmsDriverClass=com.ibm.di.plugin.pwstore.jms.driver.ActiveMQ

# JMS Server address (jms.broker=tcp://<activeMQhost>:61616 or
# jms.broker=ssl://<activeMQhost>:61617)
# jms.broker=
```

```
#
# Websphere MQ Password Store Configuration
#

# JMS driver, used to establish connecton to the message broker.
# jmsDriverClass=com.ibm.di.plugin.pwstore.jms.driver.IBMMQ

# The ID of this client. This value is used when connecting to a broker.
# Most brokers do not allow clients to have the same ID.
# jms.clientId=

# JMS Server address (ip host and tcp port number).
# jms.broker=<host>:<port>
# Login username for the password queue.
# jms.username=

# Login password for password queue.
# This field must be encoded. Use the 'encryptPasswd' utility.
# jms.password=

# MQ Server Channel
# jms.serverChannel=

# Queue Manager Name
# jms.qManager=

# Turn on SSL
# jms.sslUseFlag=false

# SSL cipher suite
# (See the WebSphere MQ documentation for a full list of supported cipher suites).
# jms.sslCipher=SSL_RSA_WITH_RC4_128_MD5




#
# IBM Security Identity Manager Integration
#
# Passwords will be be verified by an IBM Security Identity Manager Server's
# Password Strength Servlet prior to synchronization.
# To enable TIM integration, set the 'syncClass' property to one of the following:
# com.ibm.di.plugin.pwstore.log.LogPasswordStoreITIMDecorator
# com.ibm.di.plugin.pwstore.jms.JMSPasswordStoreITIMDecorator
# com.ibm.di.plugin.pwstore.ldap.LDAPPasswordStoreITIMDecorator

# URL of the IBM Security Identity Manager hosted Password Strength Servlet.
# Note: If https is used, the javax.net.ssl.trustStore* properties must be set.
# Where the specified truststore contains the IBM Security Identity Manager Server's
# certificate.
# itimPasswordUrl=https://<host>:<port>/passwordsynch/synch

# IBM Security Identity Manager user account permitted to perform a password check.
# itimPrincipalName=

# The password for the IBM Security Identity Manager user acount specified by
# the 'itimPrincipalName' property.
# itimPrincipalPassword=

# The IBM Security Identity Manager service name against which the password check
would be performed.
# itimSourceDN=erservicename=TDIPasswordService, o=IBM, ou=IBM, dc=com
```

Note:

mqe.file.ini

Required only if you are using the IBM WebSphere MQ Everyplace driver. If not, this parameter is ignored and the **jms.broker** property is used instead.

The path to the `.ini` file is generated by the `MQe Configuration` utility. For example, `C:\\Program Files\\IBM\\TDI\\V7.2\\pwd_plugins\\tds\\ MQePWStore\\pwstore_client.ini`.

mqe.notify.port
Required only if you are using the IBM WebSphere MQ Everyplace driver. If not, this parameter is ignored.

The TCP/IP port that is used when the JMS Password Connector sends notifications to the IBM WebSphere MQ Everyplace driver on behalf of the JMS Password Store. Default value is 41002.

Note: For more information about this parameter, see the section "Force transfer of accumulated messages from the JMS Password Store with IBM WebSphere MQ Everyplace" in *Reference*.

MQe Queue Manager setup

You must run the IBM WebSphere MQ Everyplace `Configuration Component` utility, which is in the*tdi_install_dir*`/pwd_plugins/jars` directory to automatically create and configure IBM WebSphere MQ Everyplace `Queue Manager`.

- Before you run the IBM WebSphere MQ Everyplace Configuration Component, open the *TDI_install_dir*`/pwd_plugins/etc/mqeconfig.props` properties file and set values for the following properties:

  clientRootFolder
  The folder where you want to store the IBM WebSphere MQ Everyplace Queue Manager. For example, on Windows, `C:\\Program Files\\IBM\\TDI\\V7.2\\pwd_plugins\\tds\\MQePWStore`.

  Note: When you specify the Windows file paths in the property files, the backslash file separator (`\`) must be escaped with a second backslash (`\\`).

  serverIP
  IP address of the system where the IBM Security Directory Integrator and the JMS Password Store are deployed.

  communicationPort
  The TCP/IP port that is used for communication between the two IBM WebSphere MQ Everyplace Queue Managers.

  clientRegistryType
  Optional. Required only for authenticated IBM WebSphere MQ Everyplace access deployments. If used, value must be set to **`PrivateRegistry`**. The **`PrivateRegistry`** stores the certificates that are issued by the IBM WebSphere MQ Everyplace Mini-Certificate server.

  clientRegistryPin
  Optional. Required only for authenticated IBM WebSphere MQ Everyplace access deployments. If used, this value represents the `PIN` access code, which is used by the IBM Security Directory Integrator JMS Password Store to access the **`PrivateRegistry`**. This value is stored as plain text in the IBM WebSphere MQ Everyplace `.ini` file.

  clientKeyRingPassword
  Optional. Required only for authenticated IBM WebSphere MQ Everyplace access deployments. This value is used when you request a certificate from the IBM WebSphere MQ Everyplace Mini-Certificate server. The value is the seed value for certificate generation. This value is stored as plain text in the IBM WebSphere MQ Everyplace `.ini` file.

  certServerReqPin
  Optional. Required only for authenticated IBM WebSphere MQ

Everyplace access deployments. This value is used as a one time authentication PIN by this Queue Manager when you request certificates from the IBM WebSphere MQ Everyplace Mini-Certificate server. This value must match the `Request PIN` value from the Mini-Certificate server setup.

certServerIPAndPort
Optional. Required only for authenticated IBM WebSphere MQ Everyplace access deployments. This value is used as the destination address for IBM WebSphere MQ Everyplace Mini-Certificate server requests. The format of the value is "`FastNetwork:<host>:<port>` where *host* is the system name or TCP/ IP address where the IBM WebSphere MQ Everyplace Mini-Certificate server is running. Port value must match the *port* value from the Mini-Certificate server setup.

debug Specify `true` or `false` to turn on or off the debug information.

Specifying the value `true` collects binary trace information that is logged by the IBM WebSphere MQ Everyplace. Enabling the flag generates a single trace file in the current directory, that is, solution directory. The size of the trace file is unrestricted. The file size increases until trace collection is stopped, or there is no disk space left. Name of the file is `mqe0.trc`.

The following example shows a sample `mqeconfig.props` configuration file:

```
clientRootFolder=C:\\Program Files\\IBM\\TDI\\V7.2\\pwd_plugins\\tds\\MQePWStore
serverIP=127.0.0.1
#clientRegistryType=PrivateRegistry
#clientRegistryPin=<Private client registry access PIN>
#clientKeyRingPassword=<Seed value for certificate generation>


# Properties used for setting up MQe Queue Manager as server

serverRootFolder=C:\\Program Files\\IBM\\TDI\\V7.2\\MQePWStore
#serverRegistryType=PrivateRegistry
#serverRegistryPin=<Private client registry access PIN>
#serverKeyRingPassword=<Seed value for certificate generation>

#certServerReqPin=<One time certificate request PIN>
#certServerIPAndPort=FastNetwork:<Mini-Certificate server hostname or IP>:<port>
#certRenewalEntityName=<QueueManager name or QueueManager+Queue name>

communicationPort=41001

#disableQueueRegistry=
debug=true
```

Note: When you specify the Windows file paths in the property files, the backslash file separator (\ ) must be escaped with a second backslash (\\ ).

The `serverRootFolder` property is not used when you configure the Storage Component. The property is used to configure the IBM WebSphere MQ Everyplace Queue Manager in the IBM WebSphere MQ Everyplace Connector and its value is ignored.

- To create and automatically configure the IBM WebSphere MQ Everyplace Queue Manager for the Storage Component, open a command prompt in the *TDI_install_dir*`/pwd_plugins/bin` folder and enter the following command as one line:

  ```
  .\mqeconfig.bat ..\etc\mqeconfig.props create client
  ```

  The log of this command is displayed on the console. After successful completion, the `Client MQe configuration successfully completed` message is displayed. If the `mqeconfig.props` file contains the optional parameters for IBM

WebSphere MQ Everyplace authenticated access, this step automatically requests the necessary certificates from the IBM WebSphere MQ Everyplace Mini-Certificate server.

Tip: If you run the IBM WebSphere MQ Everyplace certificate authenticated access deployment, certificates are requested only once per authenticate-able entity. If the following exception message is reported during configuration, re-enable certificate issue for that entity by using the Mini-Certificate server GUI.

```
[MQeConfig] [28/07/05 10:10:01]: Action failed:
Code=351;com.ibm.mqe.MQeException: Registration exception =
com.ibm.mqe.MQeException: certificate request failed[PWStoreClient 4]
 (code=8)[PWStoreClient 8] (code=351) [MQeConfig] [28/07/05 10:10:01]:
Error: Server MQe configiration failed; exception:java.lang.Exception:
Code=351;com.ibm.mqe.MQeException: Registration exception = com.ibm.mqe.
MQeException: certificate request failed[PWStoreClient4] (code=8)
[PWStoreClient 8] (code=351)
```

Note: To change the configuration of the IBM WebSphere MQ Everyplace Queue Manager, the following two options are available:

– Delete the IBM WebSphere MQ Everyplace Queue Manager from the disk and create it by following the preceding procedure.
– Install the IBM WebSphere MQ Everyplace admin tool compatible with IBM WebSphere MQ Everyplace 2.0.1.7 Queue Managers and use it to change the Queue Manager settings. For example, IBM WebSphere MQ Everyplace Explorer.

WebSphere MQ setup

When the IBM WebSphere MQ is used as JMS provider, you must copy the JAR files from the IBM WebSphere MQ installation to the class path of the Password Synchronizer.

Include the following JAR files in the class path of the Password Synchronizer:

For IBM WebSphere MQ 6.0:

- com.ibm.mqjms.jar
- com.ibm.mq.jar
- jms.jar
- connector.jar
- dhbcore.jar
- jta.jar

For IBM WebSphere MQ 7.1:

- com.ibm.mqjms.jar
- com.ibm.mq.jmqi.jar
- jms.jar
- dhbcore.jar

Chapter 11. Log Password Store

You can use the Log Password Store to verify the communication between the Java Proxy and the native plug-ins.

You can use the Log Password Store to log any actions that a normal Password Store takes. This Password Store is useful to verify whether the Java Proxy and the native plug-ins are communicating correctly.

Note: This Password Store logs the user names and the passwords in the log file of the Java Proxy. You must use the Password Store only for testing purposes. For example, during the configuration of the plug-ins and development.

The class for this password store is:
`com.ibm.di.plugin.pwstore.log.LogPasswordStore`

Chapter 12. Troubleshooting problems with the Password Synchronizers

You must check the log files of the plug-ins and the Java proxy components to diagnose problems with the Password Synchronizer.

Any problem that occurs during the operation of the plug-ins and Java Proxy is logged as an error message. You can use the time stamp and the severity level of the error message to analyze the problem.

Troubleshooting the problems with plug-ins

Based on the plug-in type, the log messages of the plug-in component are written to UNIX syslog file or LDAP Server log file.

To troubleshoot, check for the following details:

- Check the initialization status. Each Password Synchronizer logs a status message on initialization:
  - Verify whether the log exists. Relevant only for the plug-ins that log in to a file.
  - Verify whether the log contains the message of successful initialization.
  - Verify whether the timestamp of the initialization message is recent. There can be some messages that are left in the log from previous run of the plug-in.

  Non-availability of recent initialization status in the log indicates that the plug-in is not running or failed on initialization before it writes to its log. Possible causes are:
  - The plug-in is not registered correctly into the target system. See the relevant section in the Password Synchronizer for registration steps.
  - The plug-in cannot find its configuration file, `pwsync.props`. See the relevant section in the Password Synchronizer for information about how to specify the configuration file to the plug-in.
- Check for execution errors.

Troubleshooting problems with the Java Proxy

The Java Proxy component and the Password Store component of the Password Synchronizer logs messages to the file that is specified by the **`JavaLogFile`** parameter. The default log file is `proxy.log`.

For more information about the **`javaLogFile`** parameter, see javaLogFile configuration parameter.

- Check the initialization status.

  The plug-in starts the Java proxy on initialization. When the Java Proxy starts, it logs a status message.
  - Verify whether the log file exists, `proxy.log`.
  - Verify whether the log contains the message of successful initialization.
  - Verify whether the timestamp of the initialization message is recent. There can be some messages that are left in the log from previous run of the proxy.
  -

Non-availability of recent initialization status in the log indicates that the proxy is not running or failed on initialization before it can write to its log. A general Java Proxy standard or error log is available in the *authentication_folder*/proxy.stdout.log file. The *authentication_folder* is the folder that contains the plug-in configuration file pwsync.props. If the log file has a message such as java.lang.NoClassDefFoundException, possible reasons are:

– The class path of the Proxy is incomplete:

  Verify whether all the third-party libraries that are required to run the Password Store are added to the jars folder of the plug-ins. The default path is *TDI_install_dir*/pwd_plugins/jars/. If the class path generated by the startProxy script is longer than the length of a shell command, which is allowed by your operating system, the JavaProxy might not run.

– Errors that you encounter when you run the startProxy.bat or startProxy.sh manually:

  Verify whether the startProxy script runs correctly from a command prompt.

  On Linux or UNIX, make sure that the configuration of the scripts path is valid for your operating system environment.

  On Windows, the startup scripts of the plug-ins use JScript to gather the class path. Make sure that the Windows Script Host (WSH) is enabled. If the WSH is disabled on your system, you receive the following message:

  ```
  Windows Script Host access is disabled on this machine.
  Contact your administrator for details.
  ```

  when you double-click the *TDI_install_dir*/pwd_plugins/bin/worker.js file.

– When an unexpected error occurs on initialization:

  Check whether the log file of the plug-in for error messages that are related to the startup of the Java Proxy process.

  Locate the command string that the plug-in uses to start the Java Proxy in the plug-in log, and run it in a command shell.

• Check for execution errors.

Troubleshooting problems of PAM password plug-in with the IBM Security Identity Manager integration

Use this information to troubleshoot issues when the user password is changed even though it is rejected by IBM Security Identity Manager and PAM password plug-in.

Problem

The PAM password plug-in is integrated with IBM Security Identity Manager for password policy validation. The following problem is encountered when users try to change their password.

1. If the new password does not meet the requirements in the IBM Security Identity Manager password policy, the following error appears on the command line:

   ```
   testuser1@iapp2 ~]$ passwd
   Changing password for user testuser1.
   Changing password for testuser1.
   (current) UNIX password:
   New password:
   Retype new password:
   passwd: Authentication token manipulation error
   ```

2. The proxy.log shows that the password is being rejected by IBM Security Identity Manager because it does not meet the password rules. The password is also rejected by IBM Security Directory Integrator Java proxy for storing the password change in the password store for which it is configured.

```
[6/18/13 9:55 AM] {Proxy} DEBUG:
                  CTGDKN026I Received operational code: '2'.
[6/18/13 9:55 AM] {LDAPStore} WARN:
                  CTGIME012E  The password does not meet
                  the requirements of the password rule.
                  The following error occurred.
                  Error: CTGIMH023E A user name cannot be
                  part of a password.
[6/18/13 9:55 AM] {Proxy} WARN:
                  CTGDKN028I Rejecting operation.
```

3. Yet, the user password still does get changed, which is not acceptable. The user can log in with the changed password.

Solution

Update the /etc/pam.d/system-auth setting as shown here.

Mark the IBM Security Directory Integrator plug-in module and the operating system plug-in module as requisite. This setting ensures that the error that is returned by the previous plug-in module is not ignored.

```
password    requisite    pam_cracklib.so try_first_pass retry=3 type=
password    requisite    /opt/IBM/TDI/V7.2/pwd_plugins/pam/libpamtivoli_64.so
                         use_first_pass
                         /opt/IBM/TDI/V7.2/pwd_plugins/pam/pwsync_ioc.props
password    requisite     pam_unix.so md5 shadow use_authtok
```

Also, check the /etc/pam.d/passwd settings. Ensure that the password module is marked as substack.

```
#%PAM-1.0
auth       include      system-auth
account    include      system-auth
password   substack     system-auth
```

Password plug-in enhancements

Use this section to see the enhanced capabilities of the Password plug-ins.

Plug-in enhancements

The following enhancements have been implemented to the IBM Security Directory Integrator Password plug-ins:

Enhanced Java Proxy to support provisioning of custom attributes in password synchronization message

Normally, the synchronization data includes only the user name, password values and the types of the password change (for example, add, delete, or update). A few possible options for the content of a custom attribute are, hard-coded string, hostname or IP address of the system, where the Password Synchronizer is deployed, or the timestamp when the Java Proxy has received the notification. Java Proxy has been enhanced to support custom attributes in password synchronization data.

Chapter 13. IBM Security Identity Manager integration

The IBM Security Identity Manager integration for the Password Synchronizer allows verification of synchronized passwords by the Password Strength servlet of the IBM Security Identity Manager Server.

Overview

The Password Synchronization incorporates password complexity checking by using the IBM Security Identity Manager Password Policies. You can use one of the following IBM Security Identity Manager Decorator Password Synchronizer classes to enable the IBM Security Identity Manager Integration:

- `com.ibm.di.plugin.pwstore.ldap.LDAPPasswordSynchronizerITIMDecorator`
- `com.ibm.di.plugin.pwstore.jms.MQePasswordStoreITIMDecorator`
- `com.ibm.di.plugin.pwstore.log.LogPasswordStoreITIMDecorator`

  Note: The `com.ibm.di.plugin.pwstore.log.LogPasswordStoreITIMDecorator` Password Store logs the user name and the password in the log file of Java proxy. Use this password store only for testing purposes. For example, during deployment testing of the plug-ins.

Supported synchronizers

The IBM Security Identity Manager Password Synchronizer Decorator classes are supported by the following Password Synchronizers:

- Password Synchronizer for Windows
- Password Synchronizer for IBM Security Directory Server Synchronizer
- Password Synchronizer for Sun Directory Server
- Password Synchronizer for UNIX and Linux

Note: The Domino HTTP Password Synchronizer does not support integration with the IBM Security Identity Manager. Custom Password Policies can be created on the Domino Server. Using those Password Policies, you can validate the passwords before they are stored.

IBM Security Identity Manager password strength validation communication

External applications must create an XML request for a password strength validation from the IBM Security Identity Manager Server. The request is sent through HTTPS, a servlet hosted by the IBM Security Identity Manager Server. The following sample shows an XML request for password strength validation:

```
<PSWD_REQ_MSG>
 <CREDENTIALS principal="",pswd="" />
 <REQUEST op="check", srcDN="", userDN="", pswd="" />
</PSWD_REQ_MSG>
```

Credentials tag

The credentials represent the user name and password of an IBM Security Identity Manager Principal. The principal and the password values are used to enable a client, that is, Password Store decorator, to authenticate the IBM Security Identity

Manager Server. The IBM Security Identity Manager Principal must exist in the IBM Security Identity Manager Server, and the authority must be provided to run the password check. These credential values are passed to the IBM Security Directory Integrator client component through the configuration properties.

Request tag

The element attributes are:

- `op` – the operation to be run. The default value is `check`. However, you can use the value `synch` to synchronize the password with the IBM Security Identity Manager.
- `srcDN` - holds the pseudo distinguished name of the service (resource), which is the source of the password strength check. The distinguished name is in the `&<service RDN®>,<bu RDN>,<org RDN>,<tenant DN>` format. An `RDN` is in the `attribute=value` format. The `service RDN` uniquely identifies a service within a section of the organization chart. The `bu RDN` uniquely identifies a container in the organization chart within another section of the organization chart. There might be zero or several `bu RDN` depending on the org-chart structure. The `org RDN` uniquely identifies the organization within a tenant. The `tenant DN` is the physical distinguished name of the tenant. The following distinguished name identifies a service named `Test` within the `IT` organizational unit within the `Acme` organization:

  ```
  erservicename=Test,ou=IT,o=Acme,ou=Acme,dc=com
  ```

  `ou=Acme, dc=com` is the physical DN of the tenant, or the root section of the Directory Server in a single-tenant deployment.
- `userDN` – holds the distinguished name of the user account within the scope of the source service. For example, the distinguished name of the UNIX user with user ID `jdoe` is `eruid=jdoe`.
- `pswd` – holds the password value to check the strength.

Configuration of Password Synchronizers for IBM Security Identity Manager integration

You must set the `syncClass` property value in the`pwsync.props` configuration file to configure the Password Synchronizers for the IBM Security Identity Manager integration.

Configure the Password Synchronizer to use an IBM Security Identity Manager Decorator to one of the Decorator class names that are shown in the following list:

- `com.ibm.di.plugin.pwstore.ldap.LDAPPasswordStoreITIMDecorator`
- `com.ibm.di.plugin.pwstore.ldap.JMSPasswordStoreITIMDecorator`
- `com.ibm.di.plugin.pwstore.log.LogPasswordStoreITIMDecorator`

Specify the following required properties in the `pwsync.props` configuration file to configure for IBM Security Identity Manager integration

Note: Property names are case-sensitive.

itimPasswordUrl
URL of the IBM Security Identity Manager hosted Password Strength servlet. For example:

```
https://host:port/passwordsynch/synch
```

itimPrincipalName
The IBM Security Identity Manager user name to run a password check.

itimPrincipalPassword
Password for the IBM Security Identity Manager user name that is specified in the **itimPrincipalName** property.

itimSourceDN
The IBM Security Identity Manager service name against which you must run the password check. For example:

```
erservicename=TDIPasswordService, o=IBM, ou=IBM, dc=com
```

Note: When the IBM Security Identity Manager Integration is enabled, set the **checkRepository** property to `true` in the configuration file of the Password Synchronizer.

Appendix. IBM Software Support

IBM Software Support provides assistance with product defects.

Before you contact IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have:

- For IBM distributed software products (including, but not limited to, Tivoli, Lotus, and Rational® products, and also DB2® and WebSphere products that run on Windows or UNIX operating systems), enroll in Passport Advantage® in one of the following ways:
  - **Online**: Go to the following Passport Advantage web page and click **How to Enroll**:

    http://www.lotus.com/services/passport.nsf/WebDocs/Passport_Advantage_Home
  - **By phone**: For the phone number to call in your country, go to the IBM Software Support website (http://techsupport.services.ibm.com/guides/contacts.html) and click the name of your geographic region.
- For IBM eServer™ software products (including, but not limited to, DB2 and WebSphere products that run in zSeries, pSeries, and iSeries® environments), you can purchase a software maintenance agreement by working directly with an IBM marketing representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage web page (http://www.ibm.com/servers/eserver/techsupport.html).

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States or, from other countries, go to the contacts page of the IBM Software Support Handbook on the web http://techsupport.services.ibm.com/guides/contacts.html. Click the name of your geographic region for phone numbers of people who provide support for your location.

Follow the following steps to contact IBM Software Support:

1. "Determine the business impact of your problem"
2. "Describe your problem and gather background information" on page 96
3. "Submit your problem to IBM Software Support" on page 96

Determine the business impact of your problem

When you report a problem to IBM, you are asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem you are reporting. Use the following criteria:

Severity 1	**Critical** business impact: You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.
Severity 2	**Significant** business impact: The program is usable but is severely limited.

Severity 3	**Some** business impact: The program is usable with less significant features (not critical to operations) unavailable.
Severity 4	**Minimal** business impact: The problem causes little impact on operations, or a reasonable circumvention to the problem has been implemented.

Describe your problem and gather background information

When explaining a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Software Support specialists can help you solve the problem efficiently. To save time, know the answers to these questions:

- What software versions were you running when the problem occurred?
- Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.
- Can the problem be recreated? If so, what steps led to the failure?
- Have any changes been made to the system? (For example, hardware, operating system, networking software, and so on.)
- Are you currently using a workaround for this problem? If so, please be prepared to explain it when you report the problem.

Submit your problem to IBM Software Support

You can submit your problem in one of two ways:

- **Online**: Go to the "Submit and track problems" page on the IBM Software Support site (http://www.ibm.com/software/support/probsub.html). Enter your information into the appropriate problem submission tool.
- **By phone**: For the phone number to call in your country, go to the contacts page of the IBM Software Support Handbook on the web http://techsupport.services.ibm.com/guides/contacts.html. Click the name of your geographic region.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Software Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Software Support provides a workaround for you to implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the IBM product support web pages daily so that other users who experience the same problem can benefit from the same resolutions.

For more information about problem resolution, see “Searching knowledge bases” and “Obtaining fixes” on page 97.

Searching knowledge bases

If you have a problem with your IBM software, you want it resolved quickly. Begin by searching the available knowledge bases to determine whether the resolution to your problem is already documented.

Search the product documentation on your local system or network

IBM provides extensive documentation that can be installed on your local system or on an intranet server. You can use the search function of this product documentation to query conceptual information, instructions for completing tasks, reference information, and support documents.

Search the Internet

If you cannot find an answer to your question in the product documentation, search the Internet for the latest, most complete information that might help you resolve your problem. To search multiple Internet resources for your product, expand the product folder in the navigation frame to the left and select **Support on the Web**. You can search for various resources that include:

- IBM technotes
- IBM downloads
- IBM Redbooks®
- IBM DeveloperWorks
- Forums and newsgroups
- Google

Obtaining fixes

About this task

A product fix might be available to resolve your problem. You can determine what fixes are available for your IBM software product by checking the product support website:

1. Go to the IBM Software Support website (http://www.ibm.com/software/support).
2. Under **Products A - Z**, select your product name to open the product-specific support site.
3. Under **Self help**, follow the link to **All Updates**, where you find a list of fixes, fix packs, and other service updates for your product. For tips on refining your search, click **Search tips**.
4. Click the name of a fix to read the description and optionally download the fix.

To receive weekly email notifications about fixes and other news about IBM products, follow these steps:

1. From the support page for any IBM product, click **My support** in the upper-right corner of the page.
2. If you are already registered, skip to the next step. If you are not registered, click register in the upper-right corner of the support page to establish your user ID and password.
3. Sign in to **My support**.
4. On the My support page, click **Edit profiles** in the left navigation pane, and scroll to **Select Mail Preferences**. Select a product family and check the appropriate boxes for the type of information you want.
5. Click **Submit**.
6. For email notification for other products, repeat Steps 4 and 5.

For more information about types of fixes, see the *Software Support Handbook* http://techsupport.services.ibm.com/guides/handbook.html.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law :

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to

IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information in softcopy form, the photographs and color illustrations might not be displayed.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, Acrobat, PostScript and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Index

W

Z

Printed in USA

SC27-2708-03