

Transparent Password Synchronization

Configuration Guide

Software revision: 12.2.4
Document revision: 30072
Last changed: 2022-03-01

Contents

I	INTRODUCTION	2
1	About this document	3
1.1	Conventions	3
1.2	Feedback and help	4
2	Overview of Transparent Password Synchronization	5
2.1	Architecture	6
2.2	Process	8
2.3	Software components	9
2.4	Interceptor compatibility	10
2.5	Use cases	11
2.5.1	Use case: Users continue to change their Active Directory password from their desktop .	11
2.5.2	Use case: Users continue to change their password using the regular password program on Unix/Linux	11
II	IMPLEMENTATION	12
3	Implementing Transparent Password Synchronization	13
3.1	Options	14
3.2	Transparent synchronization and generated passwords	15
3.3	Load Balancing	15
4	Windows Trigger	17
4.1	About the Password Change Notification Module	17
4.2	Installing the software	18
4.2.1	Setting the longid format	21

4.2.2	Testing the connection	21
4.3	Configuring multiple servers	22
4.4	Logging	24
4.5	Filtering password change requests on a Windows trigger system	24
5	Unix Trigger	25
5.1	Configuring the API SOAP service	26
5.2	Installing the psunix installation package	27
5.3	Replacing the native password program	28
5.3.1	Installing pspasswd on the Unix trigger system	28
5.3.2	pspasswd and non-default instances	31
5.4	Configuring the <i>Hitachi ID Bravura Pass</i> PAM	31
5.4.1	Installing the <i>Bravura Pass</i> PAM	31
5.4.2	Editing pam.conf	33
5.4.3	Module options	36
5.5	Unix with NIS or NIS+	37
5.6	Editing psunix configuration files	38
5.6.1	passwd utility configuration	38
5.6.2	API SOAP Service configuration	39
5.6.3	Password Manager Service configuration (idpm)	41
6	LDAP Trigger	42
6.1	Installing the LDAP password filter plugin	43
6.2	Configuring the Password Manager (idpm) service	44
6.3	Configure your LDAP installation to use the LDAP password filter plugin	44
6.3.1	Oracle DSEE, Sun ONE Directory Server (v5.x), or Red Hat Directory Server	44
6.3.2	OpenLDAP	46
6.3.3	IBM Directory Server	46
6.4	Filtering password change requests on a LDAP Directory Service trigger system	47
7	OID-LDAP Trigger	49
7.1	Unix-based OID-LDAP server	49
7.2	Windows-based OID-LDAP server	51

7.3	Troubleshooting	55
8	OS/390 or z/OS (RACF, TopSecret, ACF2) with Mainframe Connector	56
8.1	Configuring the Password Manager service for transparent synchronization	57
9	OS/400 Trigger	58
9.1	Creating and applying a password policy	58
9.2	Installing and configuring pspwdexit	59
9.3	Configuring the Password Manager (idpm) service	61
9.4	Verifying the configuration	61
9.5	OS/400 system components	61
10	User registration	63
10.1	Enabling Transparent Password Synchronization as a self-service user	64
III	REFERENCE	67
11	Password Manager Service (idpm)	68
11.1	Allowing external communication with <i>Bravura Pass</i>	71
11.2	Testing	72
11.3	Monitoring transparent password synchronization	72
11.3.1	Managing the Password Manager Service queue	72
11.3.2	Monitoring transparent password synchronization on Windows servers	72
12	API SOAP Service (idapisoap)	74
12.1	Metadata exchange	76
12.2	Binding	77
12.2.1	IIS .NET versus WWS	77
13	API Service (idapi)	78
14	intcptsvc	80
15	testidpm	87
16	diagutil	90

17	userattrs	92
18	verifycfg	94
A	CIDR notation	97
B	File Locations	99
Index		100

Hitachi ID Systems, Inc.

DISCLAIMER

Although every effort has been made to ensure that the information in this manual is accurate, some information may be inconsistent with the most current software release.

For assistance with installation and configuration, please contact support@Hitachi-ID.com.

Part I

INTRODUCTION

About this document

1

1.1 Conventions

This document uses the following conventions:

This information ...	displayed in ...
Variable text (substituted for your own text)	<angle brackets>
Non-text keystrokes – for example, [Enter] key on a keyboard.	[brackets]
Terms unique to <i>Hitachi ID Bravura Security Fabric</i>	<i>italics</i>
Button names, text fields, and menu items	boldface
Web pages (names)	<i>italics and boldface</i>
Literal text, as typed into configuration files, batch files, command prompts, and data entry fields	monospace font
Wrapped lines of literal text (indicated by the → character)	Write this string as a →single line of text.
Hypertext links – click the link to jump to a section in this document or a web site	Purple text
External document – click the link to jump to a section in another document. The links only work if the documents are kept in the relative directory path.	Magenta text

1.2 Feedback and help

If you have feedback about this document or wish to report an omission or error, please contact doc-feedback@Hitachi-ID.com.

If you require technical assistance with *Hitachi ID Bravura Pass*, contact support@Hitachi-ID.com.

Overview of Transparent Password Synchronization

2

Hitachi ID Bravura Pass can extend the native password management on selected types of systems with *transparent password synchronization*. When this is implemented on a *trigger system*:

- Native password changes on the trigger system are subjected to the *Bravura Pass* password policy, and may be rejected on that basis.
- Successful password changes trigger automatic password synchronization for other accounts, on other systems, that belong to the same user.

Transparent password synchronization can be triggered from native password changes on any of the following systems:

- Windows 2012R2/2016/2019 servers and Active Directory domains (password filter DLL on servers and/or DCs).
- z/OS mainframes with RAC/F, ACF2 or TopSecret security products (security exit in the LPAR with the security products).
- OS/400, iSeries servers.
- Unix/Linux servers (passwd program wrapper binary or privileged access management (PAM)).
- Sun/Oracle and IBM LDAP servers (attribute change filter on the directory server).

Each of these triggers contacts the *Bravura Pass* server twice per password change, over an encrypted TCP/IP socket (shared key handshake, 256-bit AES encryption):

- First connection: validate password quality, possibly reject the user's choice of a new password and block the triggering password change due to policy violation
- Second connection: initiate transparent password synchronization

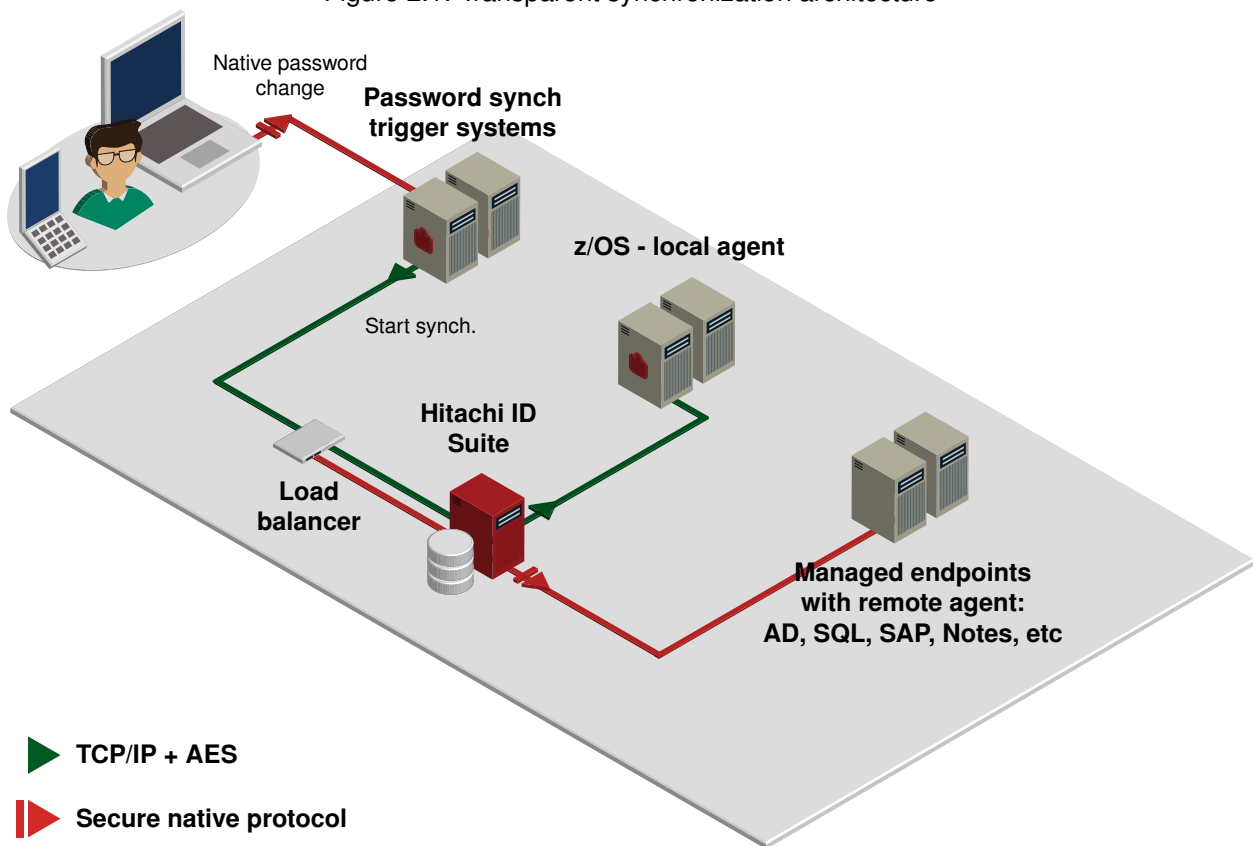
2.1 Architecture

Figure 2.1 shows the basic elements of transparent password synchronization. In the figure, a user changes their password natively on some system (most commonly a Windows workstation connected to an AD domain). A *trigger* installed on this system – in the case of an AD DC, this is a password filter DLL – contacts a *Hitachi ID Bravura Pass* server when the password change happens, typically via a load balancer.

Two calls are made to *Bravura Pass*:

1. User X on system Z wishes to set their password to value Y. Is that acceptable?
 - *Bravura Pass* tests this combination against the password policy that applies to the user.
 - The password, which may have been accepted by the trigger system, may be rejected by *Bravura Pass*. In this case, the original password change is terminated: the new password is not committed anywhere and the user gets an error message.
 - Password quality validation must be performed by *Bravura Pass* as it has access to data, such as password history (unlimited, not just last N) and dictionaries, which enable it to enforce more robust rules than the trigger system.
 - The process is fault-tolerant. Inability to contact *Bravura Pass* is usually configured to cause passwords to be accepted.
2. Password value Y has been committed locally, on system Z, for User X. Please initiate synchronization if required.
 - *Bravura Pass* looks up the user's other accounts (on system Z or elsewhere).
 - *Bravura Pass* connects to each system, resetting X's password administratively to Y.
 - In the event an account password change fails on one of the user's synchronized systems, the password remains changed on all the user's systems that worked and *Bravura Pass* re-queues and retries the password change on any system that failed until it falls out from the queue. Additionally, *Bravura Pass* may be configured to send the user one or more e-mails to notify of the problem and/or create a ticket on an incident management system to alert someone of the synchronization problem.

Figure 2.1: Transparent synchronization architecture



2.2 Process

Transparent password synchronization, triggered by a native password change on a monitored system works as follows:

1. **User:** decides to change their password(s) or has been asked to during the login process (password has expired).
2. **User:** enters their login ID, current password and desired new password.
3. **Login server:** validates password quality internally, then calls a *Hitachi ID Bravura Pass* interceptor library to further validate password quality.
4. **Bravura Pass interceptor:** contacts the *Bravura Pass* server; establishes an encrypted connection; forwards a request for password policy check.
5. **Bravura Pass:** validates password quality; returns result. In the event of an attempted policy violation, *Bravura Pass* may send a message directly to the user by email or a Windows pop-up message.
6. **Login server:** updates the user's password field internally, calls the *Bravura Pass* interceptor to notify it of the successful change. Note that a failure to meet the *Bravura Pass* policy will normally block the initial password change from completing.
7. **Bravura Pass interceptor:** contacts the *Bravura Pass* server; establishes an encrypted connection; forwards a request for password synchronization.
8. **Bravura Pass:** queues up the new password for synchronization.
9. **Bravura Pass:** resolves the single-queued event to a list of passwords that must be set for this user (one per login account).
10. **Bravura Pass:** administratively sets the user's passwords on each system to the new value.
11. **Bravura Pass:** in the event of failure, re-queues and retries; may send the user one or more emails to notify of the problem; may create a ticket on an incident management system to alert someone of an integration problem.

Transparent password synchronization triggers are provided with *Bravura Pass* for Active Directory, Windows servers, LDAP, Linux and Unix (various), iSeries and z/OS (optional component).

2.3 Software components

To implement transparent password synchronization, special software is installed on the trigger system to monitor password changes and verify the strength of new password choices with *Bravura Pass* before permitting changes. This software communicates with the Password Manager service (idpm) on the *Hitachi ID Bravura Pass* server, using an encrypted TCP socket connection.

Note: Although RSA Authentication Manager 7.1/8.2 servers are not capable of being triggers, transparent synchronization can reset PINs, as long as alpha-numeric PINs are allowed on the RSA Authentication Manager 7.1/8.2 server.

Transparent password synchronization involves the following components:

Components	Purpose
Password Manager service (idpm)	This service works in conjunction with trigger programs and libraries on various systems. Over a secure, encrypted TCP connection, the service evaluates a new password selected by a user, determines whether it should be accepted, and if so, synchronizes the password to a new value on all systems where the user has a login account.
Hitachi ID Password Change Notification Module	This local agent intercepts native password changes on Microsoft Active Directory domain controllers and Windows servers, and triggers transparent password synchronization.
Hitachi ID password replacement program (pspasswd) and the Hitachi ID Systems pluggable authentication module	This program intercepts native password changes on Unix servers and triggers transparent password synchronization.
LDAP password filter plugin (psldap) or OID-LDAP password filter plugin (psldap.so)	This local agent intercepts native password changes on LDAP Directory Service servers and triggers transparent password synchronization.
Hitachi ID OS/400 exit program (pspwdexit)	This exit program intercepts password changes on IBM OS/400 and propagates them to the <i>Bravura Pass</i> server for policy validation and to initiate transparent synchronization.

Note: Software components for Windows-based and OS/400 trigger systems are shipped with *Bravura Security Fabric* and installed in the `\<instance>\addon\transparent-synch\` directory. Software for Unix-based trigger systems is shipped with *Connector Pack*. The location depends on whether you install a global or instance-specific connector pack. The OS/390 trigger software is shipped with *Mainframe Connector*. See the [Mainframe Connector Installation Guide](#) for more information.

Optionally, you can enable the *Enable password synchronization* (PSR) module to educate users and en-

force registration for transparent password synchronization.

WARNING!: If using load balancers, do not configure any SSL options for transparent synchronization traffic. SSL options should only be configured on load balancers for WebUI traffic, not transparent synchronization. Transparent synchronization is encrypted using a proprietary encryption algorithm. Contact support@Hitachi-ID.com for more details.

2.4 Interceptor compatibility

Below is a compatibility matrix that should be taken into consideration when upgrading *Hitachi ID Bravura Pass* services (**idpm/pushpass**) or interceptors. **Y** denotes that the versions are compatible and **N** denotes that the versions are not compatible.

Table 2.2: *Bravura Pass* interceptor compatibility

Interceptor version	Service version						
	10.0.x	10.1.x	11.0.x	11.1.x	12.0.x	12.1.x	12.2.x
6.4.9	Y	Y	Y	Y	Y	Y	Y
7.3.1	Y	Y	Y	Y	Y	Y	Y
8.2.8	Y	Y	Y	Y	Y	Y	Y
9.0.x	Y	Y	Y	Y	Y	Y	Y
10.0.x	Y	Y	Y	Y	Y	Y	Y
10.1.x	Y	Y	Y	Y	Y	Y	Y
11.0.x	Y	Y	Y	Y	Y	Y	Y
11.1.x	Y	Y	Y	Y	Y	Y	Y
12.0.x	Y	Y	Y	Y	Y	Y	Y
12.1.x	Y	Y	Y	Y	Y	Y	Y
12.2.x	Y	Y	Y	Y	Y	Y	Y
CP 3.0.x (unix)	Y	N	N	Y	Y	Y	Y
CP 3.1.x (unix)	Y	N	N	Y	Y	Y	Y
CP 3.2.x (unix)	Y	Y	Y	Y	Y	Y	Y
CP 3.3.x (unix)	Y	Y	Y	Y	Y	Y	Y
CP 4.0.x (unix)	Y	Y	Y	Y	Y	Y	Y
CP 4.1.x (unix)	Y	Y	Y	Y	Y	Y	Y

Also review the access control list for the **Comma-delimited list of IP addresses with CIDR bitmask that are allowed to send socket requests** setting for the Password Manager service (idpm). Password synchronization interceptors that need to access **idpm** must be defined in this field.

For more information about Password Manager service see [Password Manager Service \(idpm\)](#).

2.5 Use cases

2.5.1 Use case: Users continue to change their Active Directory password from their desktop

Some organizations require users to change their Active Directory passwords on a regular basis. Often users do this from the login prompt on their desktop. Transparent password synchronization can be set up as follows to allow users to continue changing their passwords using the same, familiar method:

1. Install and configure *Hitachi ID Bravura Pass*.
2. Install the Password Change Notification Module on one Active Directory domain controller.

From now on, when a user changes their password using the familiar method, the Password Change Notification Module will intercept the password change. The Password Change Notification Module will then trigger automatic password synchronization for other accounts, on other systems, that belong to the same user.

2.5.2 Use case: Users continue to change their password using the regular password program on Unix/Linux

Some organizations require users to change their Unix passwords on a regular basis. Often users do this using the regular password program on Unix. Transparent password synchronization can be set up as follows to allow users to continue changing their passwords using the same, familiar method:

1. Install and configure *Hitachi ID Bravura Pass*.
2. Replace the native password program (`/usr/bin/passwd`) with `pspasswd` to intercept password changes made using the standard password command (`passwd`).

From now on, when a user changes their password using the familiar method, the password replacement program (`pspasswd`) will intercept the password change on Unix. The password replacement program will then trigger automatic password synchronization for other accounts, on other systems, that belong to the same user.

Part II

IMPLEMENTATION

Implementing Transparent Password Synchronization

3

This chapter provides an overview of the steps required to implement transparent password synchronization. It assumes you have set up basic requirements for password management, as described in the [Bravura Security Fabric Documentation](#), including email notification and at least one target system that is a source of *Hitachi ID Bravura Pass* profiles.

To implement transparent password synchronization:

1. Add target systems that will be triggers for transparent password synchronization.

See the Connector Pack Integration Guide for detailed information.

2. If required, enable the API SOAP Service (idapisoap) and ensure it is available for access by the host running the interceptor. The API Service (idapi) configuration file requires the URL of the API SOAP Service.

Note: The API SOAP Service is not required for Windows or LDAP Triggers

See [API SOAP Service \(idapisoap\)](#) for more information.

3. Gather the information that you will need when you install the necessary software:

- Trigger system's target system ID
- The communication key (or Master Key)
The CommKey value is encrypted in *Bravura Pass*. If you did not record the key in a secure location, copy the `idmsetup.inf` file from `<instance>\psconfig\` on the *Bravura Pass* server to the same location as the installer. The installer will extract the Communication Key value from the file.
- TCP port number on which the Password Manager service (idpm) is listening for the LDAP interceptor.
- URL of the API SOAP Service, for interceptors other than the LDAP and Windows interceptors.
- DNS host name of each *Bravura Pass* server

4. Install the required software on the trigger system:

- [Windows Trigger](#)
- [Unix Trigger](#)
- [LDAP Trigger](#)
- [OID-LDAP Trigger](#)

- OS/400 Trigger
- OS/390 or z/OS (RACF, TopSecret, ACF2) with Mainframe Connector

5. Educate users.

Inform users that:

- All password changes for users (with a *Bravura Pass* profile ID) will be subjected to the password policies enforced on the *Bravura Pass* server. By default transparent password synchronization is available to all users.
- When users change their passwords on the relevant system (Microsoft Active Directory, LDAP Directory Service, OS/390 mainframe), their new password will be applied automatically to all of their accounts on other systems.

3.1 Options

You can configure the following if required:

- The **Enable password synchronization (PSR)** module

This method of user education requires users to register for transparent synchronization, using the *Enable password synchronization* (PSR) module. This ensures users actively understand and accept the changes. You must enable the *Enable password synchronization* (PSR) module to activate this feature.

See [User registration](#)

- Target system groups

This allows you to apply different password policies and synchronization rules to groups of target systems.

The default target system group is configured to enable transparent password synchronization. Hitachi ID Systems recommends that all target systems belong to a single target system group, and are subject to a single password policy.

- The Password Manager service (idpm).

The Password Manager service is installed and started by default. You can set options for thread count, password change queuing, and integration with older Password Manager service (idpm) services. You can also set the Password Manager service to enforce the password strength policy for non-*Bravura Pass* users. Several synchronization events can be configured to trigger email notification or other external programs.

See [Password Manager Service \(idpm\)](#) for more detail, including command-line and scripting options.

- User notification.

You can use the *Bravura Pass* notification system to warn users of pending password expiry.

3.2 Transparent synchronization and generated passwords

Transparent password synchronization is incompatible with a security policy that mandates that users must select from a set of randomly generated passwords (by enforcing the "Be one of the N suggested passwords" rule), because even with the interceptor, the OS has no way to supply this list.

If users must select from a set of randomly generated passwords, they must use the *Bravura Pass* web interface to change their passwords.

3.3 Load Balancing

By default, the Password Manager service (idpm) service will be running on each *Hitachi ID Bravura Pass* server. However, only one server hostname may be provided to each transparent synchronization interceptor. If multiple *Bravura Pass* servers are operating, it is usually desirable to balance the transparent synchronization load between them dynamically and provide for transparent fail-over.

Round-robin DNS, or assigning multiple address records to a hostname, can be helpful for load balancing. In this configuration, an additional hostname should be set up with a record for each *Bravura Pass* server, and this hostname should be provided to the transparent synchronization interceptor installed on each target system. Target systems will then choose from the list of servers each time they make a request. This method does not provide fail-over.

Transparent synchronization requests can also be handled by a load balancer. Though no specific load balancer is endorsed for this purpose, the following criteria for its configuration apply:

- No heartbeat should be done on either of the ports used by idpm. Use `loadbalancerstatus` to probe the health of nodes. See [Automated node check when using a load balancer](#) in the *Replication and Recovery (replication.pdf)* for more information.
- Persistent or sticky connections are required. Having once connected, a host's traffic should be directed to the same server for considerably longer than the maximum request time. 3-5 minutes is suitable for most environments.
- The traffic must be load balanced as a raw TCP stream. As it is encrypted, the load balancer should attempt no translation or validation on it.
- The load balancer's address facing the *Bravura Pass* server must be configured in the list of IP addresses from which Password Manager service will allow requests.
- A firewall should restrict access to the load balancer so that only those hosts intended to be sources of transparent synchronization events may connect to the Password Manager service service. The CIDR bitmask option provided in the Password Manager service service configuration is ineffective if hosts can connect through a load balancer.

WARNING!: If using load balancers, do not configure any SSL options for transparent synchronization traffic. SSL options should only be configured on load balancers for WebUI traffic, not transparent synchronization. Transparent synchronization is encrypted using a proprietary encryption algorithm. Contact support@Hitachi-ID.com for more details.

Windows Trigger

4

This chapter shows you how to set up transparent password synchronization for a Microsoft Windows server or Microsoft Active Directory based trigger system.

Before you begin, ensure that you have read [Implementing Transparent Password Synchronization](#) and carried out the preparatory steps.

4.1 About the Password Change Notification Module

Hitachi ID Bravura Pass can intercept password changes on a Windows-based trigger system using the Hitachi ID Password Change Notification Module. The Hitachi ID Password Change Notification Module consists of an interceptor service, `intcptsvc`, and the `psintcpt.dll`. The service queues DLL requests and communicates with the Password Manager service (idpm). The DLL captures native password changes.

The installer package also includes testing and maintenance utility programs.

You can install the Password Change Notification Module on a:

- **Microsoft Active Directory domain controller (DC)**

This will affect password changes by users of the Active Directory domain. In order to intercept all password changes in your domain, you must install the Password Change Notification Module on every Active Directory DC on your network.

CAUTION: Do not install the Password Change Notification Module on an Active Directory DC that allows blank passwords. If users change their passwords to a blank password, Active Directory will not send the change to *Bravura Pass*, and the event will not be logged.

- **Windows server**

Password changes local to that server will likewise be subjected to password strength enforcement and synchronization.

The Hitachi ID Password Change Notification Module supports filters that can limit the scope of an Active Directory DC to a specific OU mapped to a target system in *Bravura Pass*. For example, you may want to limit the passwords that are intercepted to accounts in three OUs on an Active Directory DC. Each OU is mapped to a separate target system set up in *Bravura Pass*. The target systems may belong to different target system groups with distinct password policies. You can install the Password Change Notification Module on the DC and configure it with three filters mapped to each OU.

You can also use filters to include or exclude specific account names to be sent to the Password Manager service (idpm) for password strength testing and synchronization.

4.2 Installing the software

Use `intcpt.msi` or `intcpt-x64.msi` from the `\<instance>\addon\transparent-synch\ad\` directory to install the Hitachi ID Password Change Notification Module.

Before you begin:

- Note the communication key (or Master Key) used to encrypt communication between Hitachi ID Systems components on the network.

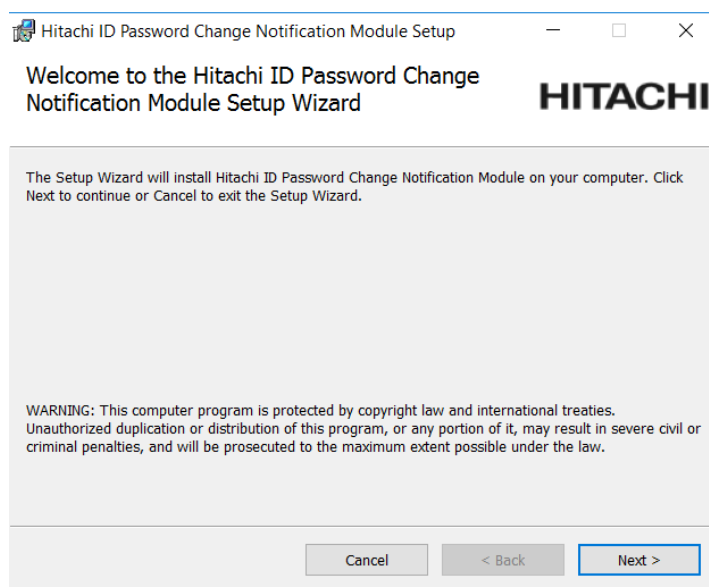
The CommKey value is encrypted in *Bravura Pass*. If you did not record the key in a secure location, copy the `idmsetup.inf` file from `<instance>\psconfig\` on the *Bravura Pass* server to the same location as the installer. The installer will extract the Communication Key value from the file.

This section shows you how to manually install the Password Change Notification Module using the Windows Installer. See:

- The *Bravura Security Fabric* Reference Manual for more information about setting MSI properties in a transform file or from the command line.

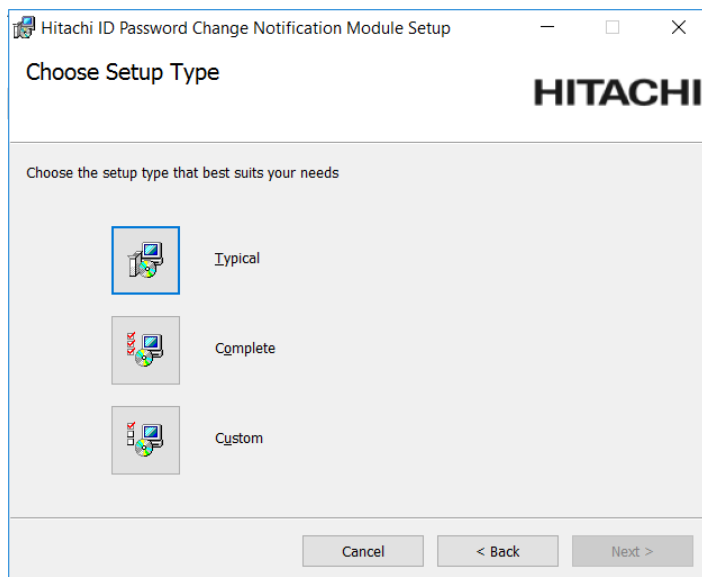
To manually install the Password Change Notification Module:

1. Copy the `intcpt.msi` or `intcpt-x64.msi` installer from the *Hitachi ID Bravura Pass* server to a scratch directory (C:\temp) on the server or domain controller (DC), or to a publicly accessible share.
2. Launch the Windows Installer package.

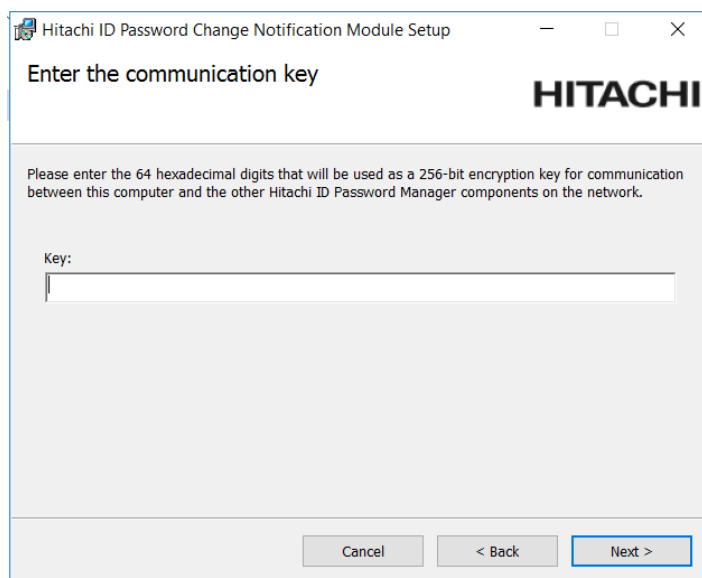


Click **Next**.

3. Read the *Bravura Pass* license. Select **I accept the terms in the License Agreement** if you agree to the terms and click **Next**.



4. Click **Complete** to include the Password Change Notification Module and configuration utility programs.



5. Type the communication key.

Network communication between Hitachi ID components is protected using a secret encryption key. Enter the same key here as you did on the main *Bravura Pass* server (communication key (or Mas-

ter Key)). If you copied the `idmsetup.inf` file from the *Bravura Pass* server the key is entered automatically.

Click **Next**.

Hitachi ID Password Change Notification Module Setup

Interceptor service configuration

HITACHI

Please enter the server name and port used by the Password Manager service on the Hitachi ID Identity and Access Management Suite server.

IDM Suite server name or IP address:

TCP/IP port the service is listening on:

3334

Primary target system ID this machine corresponds to:

Long ID format to send to Password Manager service:

%distinguishedName%

Cancel < Back Next >

6. Configure the service by entering the:

- **IDM Suite server name or IP address**
- **TCP/IP port the service is listening on**
- **Primary target system ID this machine corresponds to**

You must enter the ID of the target system you are installing on as it is configured in the *Bravura Pass* primary server.

This target must be configured as a *Bravura Pass* target system on the primary server before Password Change Notification Module will function properly.

- **Long ID format to send to Password Manager service**

The longid must match the longid on the target system. Choose the format based on the target system setting and how the user's longid is being listed. See [Setting the longidformat \(p21\)](#) for more information.

Click **Next**.

7. Click **Install** to start the installation.

The installer begins copying files to your computer. The **Installation Complete** page appears after the software has been successfully installed.

8. Click **Finish** to exit.

9. Click **Yes** to restart Windows now, or **No** if you will manually restart later.

After you reset Windows, native password changes will be intercepted by the Password Change Notification Module and forwarded to the *Bravura Pass* server for transparent synchronization.

4.2.1 Setting the longid format

The longid format on the target system must match the longid format used by the Password Change Notification Module. By default, the **agtaddn** connector's longid format is set to the NT4 name format.

Complete one or more of the following:

1. Select the matching longid format during the [installation of the Password Change Notification Module \(p18\)](#), or

Note: Versions of the Password Change Notification Module older than 11.0.1 do not prompt for a longid format and instead, set the longid to the distinguished name (DN) format by default

2. Change the longid format on the target system to use the distinguished name (DN) format - see the Connector Pack Integration Guide for detailed information, or

3. Modify the **intcptsvc.cfg** configuration file, located in:

<Program Files path>\Hitachi ID\Password Filter\service\

- (a) Locate the following lines:

```
# LongID = "%sAMAccountName%";
# LongID = "DomainName\\%sAMAccountName%";
LongID = "%distinguishedName%";
```

- (b) Comment out the LongID line:

```
LongID = "%distinguishedName%"
```

- (c) Locate and uncomment the following line:

```
# LongID = "DomainName\\%sAMAccountName%"
```

- (d) Specify the DomainName of the target.

- (e) Save the configuration file.

- (f) Restart the **intcptsvc** service.

4.2.2 Testing the connection

Test connectivity and initialize the API by running the **diagutil** program, located in the following directory on the Windows transparent password synchronization trigger system:

<Program Files path>\Hitachi ID\Password Filter\util\

See also:

- [Password Manager Service \(idpm\)](#) for more information about external server ACLs.
- [intcptsvc](#) for more information about the Password Change Notification Module service, including advanced configuration.
- [diagutil](#) for more information about using the **diagutil** program.

4.3 Configuring multiple servers

You may need to make further configuration changes to the server on which you install the interceptor if you have:

- Installed *Hitachi ID Bravura Pass* on multiple servers with the Password Manager service (idpm) running on each server for load balancing, and
- Configured the DNS servers to resolve the *Bravura Pass* server name in a “round robin” sequence.

Sometimes in this situation, Windows will cache the result and send the request to the same server each time. In this case, you configure the Windows server rotate the list of IP addresses.

You may also want to ensure that the interceptor makes multiple attempts to contact a *Bravura Pass* server before failing, to handle a condition where a single replica server is down. You can configure the Windows DNS Service to make as many attempts as you require.

After you have [installed the interceptor \(p18\)](#):

1. On the server on which the interceptor is installed, run the Windows **nslookup** program with the *Bravura Pass* server's hostname to test whether Windows is caching the result. For example, type:

```
nslookup mercury
```

The **nslookup** program displays all addresses defined for the *Bravura Pass* server, in the order returned from the DNS server. For example:

```
Server:      mercury.example.com
Addresses:   10.0.250.119, 10.0.130.108, 10.0.26.15
```

2. Repeat Step 1 as many times as there are *Bravura Pass* servers. For example, if there are three *Bravura Pass* servers, run the **nslookup** program three times.

If Windows is *not* caching the result, the order of the IP addresses is rotated with each query. For example:

- Try 1:

```
Server:    mercury.example.com
Addresses: 10.0.250.119, 10.0.130.108, 10.0.26.15
```

- Try 2:

```
Server:    mercury.example.com
Addresses: 10.0.130.108, 10.0.26.15, 10.0.250.119
```

- Try 3:

```
Server:    mercury.example.com
Addresses: 10.0.26.15, 10.0.250.119, 10.0.130.108
```

3. If the test shows that Windows *is* caching the result, force the server where the interceptor is installed, to rotate the list of IP addresses. To do this, add the `ManualDNSRotation` registry entry:

Entry name	ManualDNSRotation
Value	1
Data type	REG_DWORD

to the following registry key:

```
HKEY_LOCAL_MACHINE\Software\Hitachi ID\IDM Suite\servertools
```

4. If necessary, set the number of times you want the DLL to retry connecting to a *Bravura Pass* server. To do this, add the `ConnectRetry` registry entry in the same registry key:

Entry name	ConnectRetry
Value	number of times to retry connecting
Data type	REG_DWORD

4.4 Logging

If a connection failure occurs between the interceptor and the Password Manager service (idpm), the error is captured using Windows event logging. The error event is written to the Application log and can be viewed using Windows Event Viewer.

You can use the `logutil` program to enable logging, for debugging purposes, for add-on software. To do so:

1. Copy the `logutil` program, located in the `util` directory on the *Hitachi ID Bravura Pass* server, to the server hosting the add-on tools. It can be placed anywhere on the server.
2. Open a command prompt and invoke `logutil` with:

```
logutil -makekey -instance <instance> -level <loglevel>
```

Note: The `-makekey` option needs to be run once only, to generate an instance name and required registry entries.

See also:

[Monitoring transparent password synchronization](#) for information on monitoring transparent synchronization.

4.5 Filtering password change requests on a Windows trigger system

You can configure the interceptor service, `intcptsvc` to include or exclude certain users when they make password change requests on Windows trigger systems. The excluded requests are not sent to the Password Manager service (idpm), but are instead processed by the Windows password change facility as usual. This can be used to reduce network traffic between the trigger system and `idpm`.

You can configure the Hitachi ID Password Change Notification Module filter using the configuration file, `intcptsvc.cfg`, located in:

```
<Program Files path>\Hitachi ID\Password Filter\service\
```

See the `intcptsvc.cfg` file for basic instructions, and samples located in:

```
<Program Files path>\Hitachi ID\Password Filter\samples\
```

For more detail, and an example of advanced configuration, see [intcptsvc](#).

This chapter shows you how to set up transparent synchronization with a Unix-based trigger system. *Hitachi ID Bravura Pass* can intercept interactive password changes on a Unix server using one of two methods:

- Replacing the native password program (`/usr/bin/passwd`) with **pspasswd** to intercept password changes made using the standard password command (`passwd`).

This method is considered more robust and is less likely to cause side effects due to non-standard configurations.

- Configuring the *Bravura Pass* PAM on Solaris, HP-UX, and Linux systems to intercept password changes made by any authentication program. Normally, all authentication goes through the PAM, including `passwd`.

This method allows *Bravura Pass* to intercept any password change, including password changes where the regular password program is not used.

To set up transparent password synchronization with a Unix-based system:

1. Before you begin, ensure that you have read and performed any required preparatory steps in [Implementing Transparent Password Synchronization](#).
2. [Configure the API SOAP Service \(idapisoap\)](#) (p26).
3. If you are using SSL, ensure that a compatible version of OpenSSL (1.1.x) is installed on the Unix system.
4. If you did not select the **Unix Installation Packages** when you installed the *Hitachi ID Connector Pack*, [install the psunix installation package](#) (p27).
5. Configure transparent synchronization by:
 - [Replacing the native password program \(/usr/bin/passwd\)](#) (p28)
 - or
 - [Configuring the Bravura Pass PAM](#) (p31)

5.1 Configuring the API SOAP service

The **psunix** **pspasswd** and **pspam** interceptors installed with *Hitachi ID Connector Pack 2.1* or later use the API SOAP Service (idapisoap). These interceptors require *Hitachi ID Bravura Pass 8.1* or higher.

To use the API SOAP Service:

1. Ensure the API SOAP Service is set to automatic and started. See the API SOAP Service chapter in the Reference Manual for more information.
2. *Optional:* Ensure the messages are encrypted, by configuring the API SOAP service and IIS server to listen and communicate over SSL.
3. *Recommended:* Create a new administrator with administrative privileges to connect to the *Bravura Pass* API for the remote connections used by the UNIX trigger.

By default, *Bravura Pass* includes the `_API_USER` user. By default, this administrator is only available for shared memory connections to the API that do not require a password.

The new IDAPI caller must be assigned a password before it can be used for remote connections.

- (a) Click **Manage the system** → **Security** → **Access to product features** → **Individual administrators**.
 - (b) Click **Add new....**
 - (c) Enter a value for the ID (for example: UNIXAPI) and Name.
 - (d) Enter a password and confirm the password.
 - (e) Add the IP Address of the trigger system in the CIDR notation to the **Allowed network addresses for remote API access** field.
 - (f) Click **Update**.
4. Add the newly created administrator for the remote connections to the API to the `_EXPLICIT_API_USERS_` user class.
 - (a) Click **Manage the system** → **Policies** → **User classes** → `_EXPLICIT_API_USERS_`.
 - (b) Click the **Explicit users** tab.
 - (c) Click **Select....**
 - (d) Search for the newly added administrator's ID.
 - (e) Click on the administrator and then **Add**.
5. Assign the product administrator 'Change passwords' global help desk privileges to complete transparent synchronization.

5.2 Installing the psunix installation package

The **psunix** archive package contains the Unix Listener and other tools and files used to perform *Bravura Pass* operations on Unix systems. There are versions for each type of supported Unix system. The packages are installed in the unix directory when you choose a complete *Hitachi ID Connector Pack* installation, or select them as part of a custom installation.

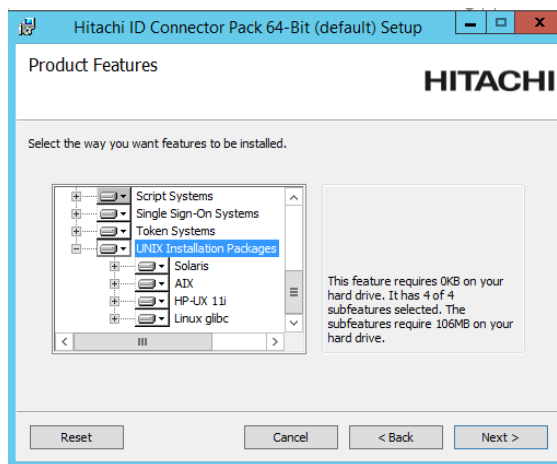
To install the package on a Unix system:

1. If you did not select the **Unix Installation Packages** when you installed *Connector Pack*, run the **setup** from the main software installation to modify your instance.

The **setup** program displays the **Select an instance to configure** page.

- (a) Click **Hitachi ID Connector Pack 64-Bit**.
- (b) Click **Modify**.
- (c) Choose **Change**

Ensure that the select  icon is selected for **Unix Installation Packages**, and the appropriate packages, on the component selection page.



Click **Change**, then complete the installation procedure.

See [Installing Connector Pack](#) in the *Connector Pack Integration Guide* for more details.

2. Copy the **psunix-<os>.<cpu>.tar.gz** file from the unix directory to a scratch directory, such as /tmp/, on the Unix server.
3. Log into the Unix server with administrative privileges, and extract the files from the **psunix** archive. For example, type:

```
cd /tmp
tar -zxvf psunix-solaris10.sparc64.tar.gz
```


5.3 Replacing the native password program

To effectively intercept all password changes made using the regular password program on Unix (/usr/bin/passwd), you can install a replacement password program and a matching configuration file (**psunix.cfg**) on every Unix server where users might change their local or NIS/NIS+ passwords. See [Unix with NIS or NIS+](#) for more information about synchronization in an NIS / NIS+ environment.

The Hitachi ID password replacement program (pspasswd) applies password strength rules defined on the *Hitachi ID Bravura Pass* server to all new password selections. It uses the old, renamed, password program to implement the password change locally on the Unix server, and then forwards a request for synchronization to the *Bravura Pass* server.

Alternately, you can [configure the Bravura Pass PAM](#) (p31) to intercept password changes made by any authentication program, including passwd.

5.3.1 Installing pspasswd on the Unix trigger system

To install Hitachi ID password replacement program:

1. Note the communication key (or Master Key). The CommKey value is encrypted in *Bravura Pass*. If you did not record the key in a secure location, copy the **idmsetup.inf** file from <instance>\psconfig\ on the *Bravura Pass* server to the same location as the installer. The installer will extract the Communication Key value from the file.
2. Note the product administrator and password to connect to the API SOAP Service (idapisoap).
3. Note the URL of the API SOAP Service.
4. If you did not select the **Unix Installation Packages** when you installed the *Hitachi ID Connector Pack*, [install the psunix installation package](#) (p27)
5. Run the installation shell script in:
 - [Interactive mode](#) (p28)
 Or
 - [Non-interactive mode](#) (p30)

5.3.1.1 Installing pspasswd interactively

Installing interactively takes less preparation and allows you to specify settings during installation. You can use the **idmsetup.inf** configuration to pass through some of the information as defaults.

To interactively install pspasswd on the Unix system:

1. Run the shell script **install.sh** from the root of the installation package:

```
sudo sh install.sh [ -inf <path>/idmsetup.inf ] [ -inst <instancename> ]
```

where:

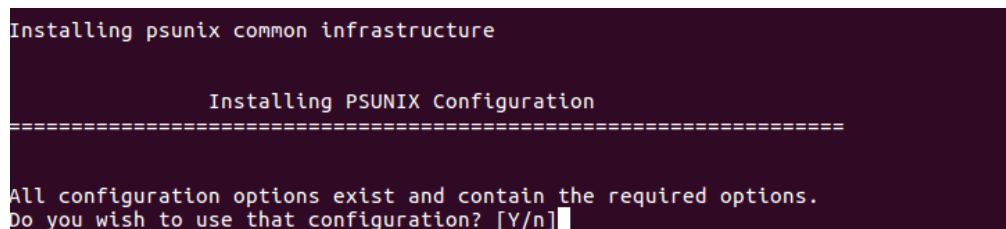
Option	Description
-inf	Specifies the path to the idmsetup.inf file. If omitted, you must enter communication key (or Master Key) and other information when prompted.
-inst	Specifies the instance name for location of the psunix files. If omitted, files are copied to the <code>/usr/local/psunix/default</code> instance.

See [pspasswd and non-default instances](#) for more information about the instance location.

2. Follow the instructions displayed by the installer script.

In the configuration process:

- Allow affected files to be backed up.
- Select the "Passwd Transparent Synch" installation option.
- If you want to use the configuration options that exist in `/etc/psunix.cfg`, type `Y` when asked. If you want to change the values, type `n`.



```
Installing psunix common infrastructure

Installing PSUNIX Configuration
=====
All configuration options exist and contain the required options.
Do you wish to use that configuration? [Y/n]
```

See the Unix / Linux Integration Guide ([unix-linux.pdf](#)) for more information about the Unix listener.

In the installation process:

- Enter the target ID of the target used to target this system.
- Enter the URL of the API SOAP Service.
- Enter a proxy URL if you are using a proxy. Press **Enter** if you are not using a Proxy.
- Enter the proxy user name if you are using a proxy. Press **Enter** if you are not using a Proxy.
- Enter the proxy user name's password if you are using a proxy. Press **Enter** if you are not using a Proxy.
- If you are using SSL, enter the path holding the CA certificate(s). Press **Enter** if you are not using SSL.
- If you are using SSL, enter the certificate details. Press **Enter** if you are not using SSL.
- Define the `[libcurl]` path or press **Enter** to use the system default libcurl library.

- Define the [ignore] or press **Enter** to use the default value.
- Define the user name for login to the IDAPI service.
- Define the password for the IDAPI user.

The installer renames the old password program to *<program name>.bin* and replaces it with **pspasswd**.

5.3.1.2 Installing pspasswd non-interactively

The installer's non-interactive mode allows you to perform unattended installations. This would be advantageous where you want to install on many systems over SSH, for example. This mode requires you to write a *response file* that is used with a command line option.

To install pspasswd non-interactively:

1. Edit the following sections of the **psunix-responsefile.cfg** in the root of the installation package:

```
#####
## general options

# Prior to installing PSUNIX, the installer allows the option to
# backup files affected by the installation process.

pre-backup = "Y";

# By default, if pre-existing configuration file(s) contains all the
# required options, do not replace them.

use-preexisting-cfg = "Y";
```

2. Edit **<psunix-root>/conf/psunix.cfg** to define the communication key (or Master Key) that matches the one set during installation on the *Bravura Security Fabric* server; for example:

```
commkey = "<encrypted commkey value>";
```

Optionally, you can pre-configure other options in this file if you want different behavior from the default. See the Unix / Linux Integration Guide (**unix-linux.pdf**) for details.

3. Edit the **pspasswd** and **pushpass** configuration files as described in [Editing psunix configuration files](#).
4. Run the shell script **install.sh** from the root of the installation package:

```
sh install.sh -c 2 -ni [ -inst <instancename> ]
```

where:

Option	Description
-inst	Specifies the instance name for location of the psunix files. If omitted, files are copied to the <code>/usr/local/psunix/default</code> instance. See pspasswd and non-default instances for more information about the instance location.

5.3.2 pspasswd and non-default instances

The **psunix** local instance name, defined by the `-inst` option when running the `install.sh` script, is not connected to the main *Hitachi ID Bravura Security Fabric* instance name. If specified, it designates a sub-target.

During install/setup, if the instance name is the default, the installer symbolically creates a link from:

- `/usr/local/psunix/<instance>/psunix.d` to `/etc/psunix.d`, and
- `/usr/local/psunix/<instance>/psunix.cfg` to `/etc/psunix.cfg`

The **pspasswd** binary (due to the fact that only one version can be installed in `/usr/bin` or `/bin`) always looks for `/etc/psunix.cfg`.

If you want to install **pspasswd** to run in a non-default instance, you must manually create the symbolic links to `/etc/psunix.d` and `/etc/psunix.cfg`.

5.4 Configuring the Bravura Pass PAM

To effectively intercept all password changes made using PAM-enabled applications, you can install the *Hitachi ID Bravura Pass* PAM (**pspam.so**) on every Unix server where users might change their local or NIS/NIS+ passwords. The *Bravura Pass* PAM, **pspam.so**, is currently available for Solaris, HP-UX, and Linux systems. See [Unix with NIS or NIS+](#) for more information about synchronization in an NIS / NIS+ environment.

The *Bravura Pass* PAM applies password strength rules defined on the *Bravura Pass* server to all new password selections. It intercepts the password change event on the Unix server, and forwards a request for synchronization to the *Bravura Pass* server.

Alternately, you can [replace the native password program \(p28\)](#) with **pspasswd** to intercept password changes made using the standard password command (`passwd`).

WARNING!: The password replacement program (**pspasswd**) is incompatible with **pspam.so**. Installing both can cause undesirable side effects, unless the **pspam.so** is configured to be used on login *only*.

5.4.1 Installing the Bravura Pass PAM

Before you begin, note the communication key (or Master Key). The CommKey value is encrypted in *Bravura Pass*. If you did not record the key in a secure location, copy the `idmsetup.inf` file from `<instance>\psconfig\` on the *Bravura Pass* server to the same location as the installer. The installer will extract the Communication Key value from the file.

To install **pspam.so** on a Unix server:

1. If you did not select the **Unix Installation Packages** when you installed the *Hitachi ID Connector Pack*, install the [psunix installation package](#) (p27).
2. Run the installation shell script in [interactive mode](#) (p32)

5.4.1.1 Installing pspam.so interactively

To interactively install **pspam.so** on the Unix system:

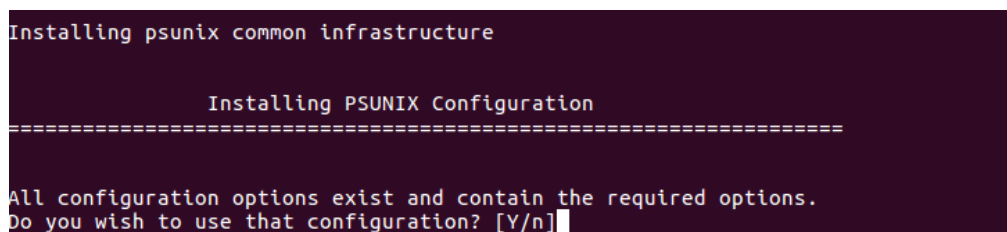
1. Run the shell script **install.sh** from the root of the installation package:

```
sh install.sh 3 [ -inf <path>/idmsetup.inf ]
```

2. Follow the instructions displayed by the installer script.

In the configuration process:

- Allow affected files to be backed up.
- Select the "Pam Transparent Synch" installation option.
- If you want to use the configuration options that exist in `/etc/psunix.cfg`, type **Y** when asked. If you want to change the values (for example, the **idpm** server name and port number), type **n**.



```
Installing psunix common infrastructure

Installing PSUNIX Configuration
=====

All configuration options exist and contain the required options.
Do you wish to use that configuration? [Y/n]
```

In the installation process:

- Enter the target ID of the target used to target this system.
- Enter the URL of the API SOAP Service (`idapisoap`).
- Enter a proxy URL if you are using a proxy. Press **Enter** if you are not using a proxy.
- Enter the proxy user name if you are using a proxy. Press **Enter** if you are not using a proxy.
- Enter the proxy user name's password if you are using a proxy. Press **Enter** if you are not using a proxy.
- If you are using SSL, enter the path holding the CA certificate(s). Press **Enter** if you are not using SSL.
- If you are using SSL, enter the certificate details. Press **Enter** if you are not using SSL.
- Define the `[libcurl]` path or press **Enter** to use the system default libcurl library.
- Define the `[ignore]` or press **Enter** to use the default value.

- Define the user name for login to the IDAPI service.
- Define the password for the IDAPI user.

Note: This install option does not put the module into the PAM stack. You must copy the file to the proper location and correctly declare it in the corresponding PAM configuration files.

3. Check the `/etc/pam.conf` file on your Unix server to determine if your system uses PME (password management extensions).

A machine without PME normally uses only one PAM module, `pam_unix.so`. A machine with PME splits the functions of `pam_unix.so` into several different modules:

- `pam_dhkeys.so`,
- `pam_authtok_get.so`,
- `pam_authtok_check.so`, and
- `pam_authtok_store.so`.

4. Copy the appropriate PAM file from the `psunix` directory to the `/usr/lib/security/` directory and rename it to `pspam.so`:

- If your Solaris system uses PME, copy:
`psunix/pspam/pspam_pme.so.<OS>.<ARCH>`
- If your Solaris, HP-UX, or Linux system does *not* use PME, copy:
`psunix/pspam/pspam.so.<OS>.<ARCH>`

Next:

Edit the `/etc/pam.conf` file as described in [Editing pam.conf](#).

5.4.2 Editing pam.conf

The following sections describe how to configure HP-UX and Solaris (with and without PME (password management extensions)) systems, to use the *Hitachi ID Bravura Pass* PAM.

WARNING!: Be very careful when editing `/etc/pam.conf`, as a misconfiguration can result in your system becoming inaccessible. *Always* keep a login shell open while editing this and test the login mechanism before closing the shell.

5.4.2.1 Configuration for HP-UX

The password management section of `/etc/pam.conf` on HP-UX systems is normally written as follows:

```
login    password required /usr/lib/security/libpam_unix.1
passwd  password required /usr/lib/security/libpam_unix.1
```

```
dtlogin password required /usr/lib/security/libpam_unix.1
dtaction password required /usr/lib/security/libpam_unix.1
OTHER password required /usr/lib/security/libpam_unix.1
```

For each application (login, passwd, dtlogin, dtaction, OTHER) that should use **pspam.so**, edit `/etc/pam.conf` as follows:

1. Append `use_first_pass` to the end of each line.
2. Type a new line above each existing line in the format:

```
<app> password required /usr/lib/security/pspam.so <option>
```

If the application is a login application, include the `get_old_pass` option as follows.

```
login password required /usr/lib/security/pspam.so get_old_pass
```

See [Module options](#) for a list of options that can be specified in the configuration file.

3. Save the file.
4. Ensure that you successfully test the login mechanism before closing the open shell.

For example, a complete password management section of `/etc/pam.conf` appears as follows:

```
login password required /usr/lib/security/pspam.so get_old_pass
login password required /usr/lib/security/libpam_unix.1 use_first_pass
passwd password required /usr/lib/security/pspam.so
passwd password required /usr/lib/security/libpam_unix.1 use_first_pass
dtlogin password required /usr/lib/security/pspam.so get_old_pass
dtlogin password required /usr/lib/security/libpam_unix.1 use_first_pass
dtaction password required /usr/lib/security/pspam.so
dtaction password required /usr/lib/security/libpam_unix.1 use_first_pass
OTHER password required /usr/lib/security/pspam.so
OTHER password required /usr/lib/security/libpam_unix.1 use_first_pass
```

5.4.2.2 Configuration for Solaris 8/9 without PME

The password management section of `/etc/pam.conf` on Solaris systems without PME is normally written as follows:

```
other password required /usr/lib/security/pam_unix.so.1
```

To customize `/etc/pam.conf` to work with the *Hitachi ID Bravura Pass* PAM for Solaris without PME:

1. Append `use_first_pass` to the end of the “other” line as follows:

```
other password required /usr/lib/security/pam_unix.so.1 use_first_pass
```

2. Add a new “other” line for the **pspam.so** module above the first one as follows:

```
other    password required /usr/lib/security/pspam.so get_old_pass
```

See [Module options](#) for a list of options that can be specified in the configuration file.

3. Create a new section for the **passwd** program by inserting the following lines above the “other” lines:

```
passwd  password required /usr/lib/security/pspam.so
passwd  password required /usr/lib/security/pam_unix.so.1 use_first_pass
```

4. Save the file.
5. Ensure that you successfully test the login mechanism before closing the open shell.

For example, a complete password management section of `/etc/pam.conf` appears as follows:

```
passwd  password required      /usr/lib/security/pspam.so
passwd  password required      /usr/lib/security/pam_unix.so.1 use_first_pass
other   password required      /usr/lib/security/pspam.so get_old_pass
other   password required      /usr/lib/security/pam_unix.so.1 use_first_pass
```

5.4.2.3 Configuration for Solaris 8/9 with PME

The password management section of `/etc/pam.conf` on Solaris systems with PME is normally written as follows:

```
other   password required      pam_dhkeys.so.1
other   password requisite      pam_authtok_get.so.1
other   password requisite      pam_authtok_check.so.1
other   password required      pam_authtok_store.so.1
```

For each application that should use **pspam_pme.so**:

1. Replace the line:

```
<app> password requisite pam_authtok_get.so.1
```

with:

```
<app> password requisite pspam.so <options>
```

See [Module options](#) for a list of options that can be specified in the configuration file.

2. Save the file.
3. Ensure that you successfully test the login mechanism before closing the open shell.

For example, a complete password management section of `/etc/pam.conf` appears as follows:

```
other password required pam_dhkeys.so.1
other password requisite pspam_pme.so
other password requisite pam_authtok_check.so.1
other password required pam_authtok_store.so.1
```

5.4.2.4 Configuration for Linux

For systems using `/etc/pam.d/*`

To replace the use of `pam_unix.so` for strength checking, and enable transparent synchronization with all applications that use PAM, edit the `/etc/pam.d/` configuration file. For RHEL-based system this is usually `/etc/pam.d/system-auth-ac`. For Debian-based systems, this is usually `/etc/pam.d/common-password`. The configuration file normally contains a line *similar* to the following:

```
password required pam_unix.so nullok obscure min=4 max=8 md5
```

Add the following below the existing `pam_unix.so` line:

```
password required pspam.so
```

If you only want to affect a particular application, add the `pspam.so` line to the appropriate `/etc/pam.d/<app>` file.

If using a RHEL based system with SELinux, you must ensure that `pspam.so` is labeled with the correct SELinux security context. To relabel the file and then change its default label, run the following commands as root:

```
chcon textrel_shlib_t /lib/security/pspam.so
semanage fcontext -a -t textrel_shlib_t -f -- \
    /lib/security/pspam.so
```

For systems using `/etc/pam.conf`

For Linux systems using `/etc/pam.conf` follow the configuration instructions for Solaris 2.6/7 and Solaris 8/9 without PME.

5.4.3 Module options

The `pspam.so` module supports the following command line options:

Table 5.1: pspam.so arguments

Option	Description
use_first_pass	Do not prompt for passwords. Instead, The module relies upon passwords obtained by prior modules in the module stack. Do <i>not</i> use this option if pspam.so is the first module in an application's stack.
debug	Write all debug-level logging to syslog . The pspam.so module logs to syslog using the LOG_AUTH or LOG_AUTHPRIV facility (see the Unix man page for more information about syslog).
get_old_pass	Always prompt the user for his or her old password. This option is only necessary for login applications. When login requests a password change it runs as root. Normally, if this option is not specified, the PAM module allows root to change a user's password without prompting for the user's old password. Non-root users are always prompted to enter their current passwords.
-conf <conffile>	Use the specified configuration file. If this option is not specified, /etc/psunix.cfg is used.

5.5 Unix with NIS or NIS+

Normally, in an NIS or NIS+ environment:

1. There is a single, master password for every user.
2. Password changes made on an NIS client are automatically propagated to the NIS master server. When this happens, only the password hash, and not the plaintext password, is sent to the NIS master server.
3. Password resets are only possible on the master server.
4. In NIS, password resets must be made by manually editing the passwd map file.

Hitachi ID Bravura Pass allows you to extend the reach of the single NIS password:

1. Password changes made in the NIS environment can be automatically propagated to other, non-NIS systems.
2. Password changes made outside of NIS can be automatically propagated to the NIS environment.
3. Password resets can be made from the command-line of the NIS master, or from a web browser, replacing the manual editing process.
4. Propagation of new passwords between an NIS master and its slaves can be replaced by *Bravura Pass* password propagation, which is much faster.

The installation process is the same as for individual Unix servers, with the following exceptions:

1. The *Bravura Pass* Unix server / listener is only installed on the NIS master server.
2. You must edit additional scripts for an NIS connector. See the Unix / Linux Integration Guide ([unix-linux.pdf](#)).
3. A replacement `passwd` program (`pspasswd`) must be installed on all NIS client machines, and not just on the master. This is required because *Bravura Pass* needs access to plaintext password values to synchronize with other systems, but the NIS master only receives password hashes, and not plaintext passwords.

Alternately, you can replace `passwd` on the NIS clients with a shell script that can:

- Invoke a web browser, prompting users to change their passwords using a *Bravura Pass* web interface.
- Use a program such as SSH to run the `passwd` program on the NIS master server.

5.6 Editing psunix configuration files

Hitachi ID Bravura Pass components that you install on a Unix-based server use a configuration file to define interaction between the component, the Unix server, and the *Bravura Pass* server. By default, this file is `/etc/psunix.cfg`; however, most components allow you to specify an alternate file.

When you install *Bravura Pass* component on Unix using an installer script (for example, `install.sh`), the installer creates a `psunix.cfg` file and configures the required values. You can later edit this file to configure additional options or change your settings.

The `psunix.d` configuration directory contains several configuration files that contain the settings for the various `psunix` components, including the `pushpass` file, which contains settings for transparent password synchronization.

See also:

Unix / Linux Integration Guide ([unix-linux.pdf](#)) for more information about `psunix` scripts.

5.6.1 passwd utility configuration

The `pspasswd` file specifies the `passwd` utility used to perform an operation on a non-*Hitachi ID Bravura Security Fabric* users password. Usually, this option specifies operating system's `passwd` command. The native password operation is executed if the user is contained in the `[restricted-user-list]` option, or is contained in the ignore list on the *Bravura Security Fabric* server. The options are as follows:

passwd-cmd-reset A reset operation is less strict than a change operation since it does not validate the users old password first. Most native `passwd` commands do both change and reset operations depending on who is running the command and the arguments passed on the command line. Generally, running the `passwd` command as superuser is considered a password reset operation.

This option accepts psunix textual replacement strings, notably the "%u" keyword indicating the user-name.

Example:

```
passwd-cmd-reset = "/bin/passwd.bin %u";
```

passwd-cmd-change A change operation is more strict than a reset operation since it validates the users old password first. Most native passwd commands do both change and reset operations depending on who is running the command and the arguments passed on the command line. Generally, running the passwd command as a non-privileged user is considered a password change operation.

This option accepts psunix textual replacement strings, notably the "%u" keyword indicating the user-name.

Example:

```
passwd-cmd-change = "/usr/bin/yppasswd %u";
```

5.6.1.1 Exit status codes

The following table outlines the **pspasswd** exit status codes:

Table 5.2: Exit status codes

Error code	Description
0	Success.
1	Syntax error in PSLANG override script.
2	Failed to acquire password policy from remote idpm/pushpass service (using legacy protocol).
3	Failed to reset password using native command line tool.
4	Failed to reset password.

5.6.2 API SOAP Service configuration

The **idapi** file is used to configure the connection to API SOAP Service (idapisoap). The options are as follows:

targetid If you are using aliasing, this option is used to specify the ID of the target.

Example:

```
targetid = "UNIXSERVER";
```

url The url that API SOAP Service (idapisoap) is listening on.

Example:

```
url = "http://hipmservice/default/idapi";
```

user The product administrator used to connect to the API SOAP Service.

Example:

```
user = "_API_USER";
```

psw The product administrator password used to connect to the API SOAP Service. `idaptool` can be used to provide an encrypted form of the password.

Example:

```
psw = "{AES}xdWShI2f+  
fm7Bd0SRhIi9kHvdhM9Y0fVxvKjpIbHfp4T47X2IAjLakoNitoSfu4Z" ;
```

libcurl In order to communicate to the API SOAP Service over SSL, the libcurl is required. If the full path is specified, then the library can be loaded when connecting over SSL. If no libcurl is available and plain HTTP is used, the value can be set to '0'. If empty, the system default is used.

Example:

```
libcurl = "0";
```

capath When communicating to the API SOAP Service over SSL, a certificate check will be made unless ignore is set to "1". If the CA certificate is not installed on the system default paths, a path can be specified.

Example:

```
capath = "/etc/certs";
```

cert When communicating to the API SOAP Service over SSL, a client-side certificate can be provided. If there is a passphrase as part of the certificate it needs to be specified as well.

Example:

```
cert = "/etc/certs/hipmcert.pem:apassphrase";
```

ignore When communicating to the API SOAP Service over SSL, the certificate check can be ignored. If 0, the check is not ignored. If 1, the check is ignored.

Example:

```
ignore = "1";
```

language The language set in this value will be used when fetching the password rules. By default it is en-us. The language packs must be installed in order to retrieve rules in other languages.

Example:

```
language = "fr-fr";
```

fail-if-unavailable Specifies the action to take if the password operation fails and the Password Manager service (idpm) cannot be contacted. The default behavior is to fail the operation if the Password Manager service is unavailable.

Example:

```
fail-if-unavailable = "true";
```

5.6.3 Password Manager Service configuration (idpm)

The **pushpass** file is used to configure the Password Manager service (idpm). The options are as follows:

targetid If you are using aliasing, this option is used to specify the ID of the target.

Example:

```
targetid = "UNIXSERVER";
```

hostname The hostname option is used to specify the location of the Password Manager service service to be used by pspasswd. This can either be an IP address or a hostname.

Example:

```
hostname = "UNIXSERVER";
```

port The port option is used to specify the port that the Password Manager service is running on. The default value for the Password Manager service is 3333.

Example:

```
port = "3333";
```

timeout Specifies the timeout (in seconds) that should be used when communicating with Password Manager service. The default value is set to 10 seconds.

Example:

```
timeout = "10" ;
```

fail-if-unavailable Specifies the action to take if the password operation fails and the Password Manager service cannot be contacted. The default behavior is to fail the operation if the Password Manager service is unavailable.

Example:

```
fail-if-unavailable = "true";
```

This chapter shows you how to set up transparent password synchronization with an LDAP Directory Service trigger system.

Before you begin, ensure that you have read [Implementing Transparent Password Synchronization](#) and carried out the preparatory steps.

Hitachi ID Bravura Pass can intercept password changes on Unix-based LDAP servers using a pre-change and post-change strength filter, `psldap*`¹

You can install the LDAP password filter plugin (psldap) on the following Unix-based servers:

- Sun ONE Directory Server (v5.x), Oracle DSEE and Red Hat Directory Server
- OpenLDAP v2.2.x
- IBM Directory Server

To set up transparent password synchronization with an LDAP Directory Service trigger system:

1. Before you begin, ensure that you have read and performed any required preparatory steps in [Implementing Transparent Password Synchronization](#).
2. Ensure that a compatible version of OpenSSL (1.1.x) is installed on the LDAP system.
3. [Install the LDAP password filter plugin \(p43\)](#).
4. [Configure the Password Manager \(idpm\) service \(p44\)](#)
5. [Configure your LDAP installation to use the plugin \(p44\)](#).
6. Optional: [Filter password change requests \(p47\)](#) to include certain users, groups and domains.


WARNING!: Ensure that your LDAP client does not hash new passwords before sending requests to the LDAP server. If you do not want passwords to be transmitted in plaintext, it is highly recommended that you enable SSL on the LDAP server.

¹The actual name of the filter varies depending on the target system type. See the appropriate section for details.

6.1 Installing the LDAP password filter plugin

To install the LDAP password filter plugin (psldap) on a Unix-based LDAP server:

1. If you did not select **Unix Installation Packages** when you installed the *Hitachi ID Connector Pack*, run **setup** on the *Hitachi ID Bravura Pass* server to modify your *Connector Pack* installation.

Ensure that the  icon is selected for the appropriate Unix package on the component selection page. Click **Next**, then complete the installation procedure.

See [Installing the psunix installation package](#) for more detail.

2. Copy the **psunix-<os>.<cpu>.tar.gz** file from the unix directory to a scratch directory (such as /tmp) on the Unix server.
3. Log into the LDAP server with administrative privileges, and extract the files from the **psunix** archive. For example, type:

```
cd /tmp
tar -zxvf psunix-solaris9.sparc64.tar.gz
```

4. Run **install.sh** and select LDAP Transparent Synch option.

```
sh install.sh [ -inf <path>/idmsetup.inf ]
```

5. Follow the instructions displayed by the installer script.

In the installation process, follow the instructions and input the information prompted by each input field. To skip a field, press **[Enter]** to use the default value.

6. Verify that the following shared object files are copied to /usr/local/psunix/default/.

The **psldap*** shared object files are named using the format:

psldap-<ldap-type>.so

Where the <ldap-type> is:

sunldap if you are running Oracle DSEE, Sun ONE Directory Server, or Red Hat Directory Server

openldap if you are running OpenLDAP

ibmldap if you are running IBM Directory Server

7. Ensure that the psunix folder and all files and plugins inside are readable and executable.

For example, run the following commands:

```
chmod -R a+rx /usr/local/psunix/
chmod a+rx /usr/local/psunix/default/psldap-openldap.so
```


Also ensure that /etc/psunix.cfg and /etc/psunix.d/ have read and execute permissions:

```
chmod a+rx /etc/psunix.cfg
chmod -R a+rx /etc/psunix.d/
```


8. Stop the LDAP service.
9. Start the LDAP service.

6.2 Configuring the Password Manager (idpm) service

To allow external servers access to the Password Manager service (idpm) on the primary *Hitachi ID Bravura Pass* server you must also add a CIDR mask address for the trigger system.

1. Click **Manage the system** → **Maintenance** → **Services**
2. Select  **Hitachi ID (idpm) Password Manager Service**.
3. Add a CIDR mask address for the trigger system in the following setting:

Comma-delimited list of IP addresses with CIDR bitmask that are allowed to send socket requests

See the [Password Manager Service \(idpm\)](#) for more information about the Password Manager service.

6.3 Configure your LDAP installation to use the LDAP password filter plugin

Refer to the appropriate subsection:

- [Oracle DSEE, Sun ONE Directory Server \(v5.x\), or Red Hat Directory Server \(p44\)](#)
- [OpenLDAP v2.2.x \(p46\)](#)
- [IBM Directory Server \(p46\)](#)

6.3.1 Oracle DSEE, Sun ONE Directory Server (v5.x), or Red Hat Directory Server

Note: The following instructions are intended for Sun ONE Directory Server (formerly Netscape/iPlanet Directory Server) v5.x. Details may vary depending on your version of the software.

Sun ONE Directory Server is more currently known as Oracle Directory Server Enterprise Edition (Oracle DSEE).

CAUTION: For Sun ONE Directory Server, stop the directory server before making these changes, otherwise the changes will be overwritten.

To configure Sun ONE Directory Server to use LDAP password filter plugin (psldap):

1. Find the dse.ldif file.

The file is usually located in `<slapd-servername>/config`.

2. Edit the file by appending the following two sections:

```
dn: cn=Psynch Check Password,cn=plugins,cn=config
objectclass: top
objectclass: nsSlapdPlugin
objectclass: extensibleObject
cn: Hitachi ID Check Password
nsslapd-pluginPath: /usr/local/psunix/default/psldap-sunldap.so
nsslapd-pluginInitfunc: prepasswd_init
nsslapd-pluginType: preoperation
nsslapd-pluginEnabled: on
nsslapd-pluginarg0: -cfg
nsslapd-pluginarg1: /usr/local/psunix/default/psldap.cfg
nsslapd-pluginID: password-preop
nsslapd-pluginVersion: none
nsslapd-pluginVendor: Hitachi ID
nsslapd-pluginDescription: Transparent Password Strength plugin

dn: cn=Psynch Synchronize Password,cn=plugins,cn=config
objectclass: top
objectclass: nsSlapdPlugin
objectclass: extensibleObject
cn: Hitachi ID Synchronize Password
nsslapd-pluginPath: /usr/local/psunix/default/psldap-sunldap.so
nsslapd-pluginInitfunc: postpasswd_init
nsslapd-pluginType: postoperation
nsslapd-pluginEnabled: on
nsslapd-pluginarg0: -cfg
nsslapd-pluginarg1: /usr/local/psunix/default/psldap.cfg
nsslapd-pluginID: password-postop
nsslapd-pluginVersion: none
nsslapd-pluginVendor: Hitachi ID
nsslapd-pluginDescription: Transparent Password Synchronization plugin
```

If the configuration file is not specified for Unix-based servers, the default is `/etc/psunix.cfg`.

3. Save and close the file.
4. Create a new directory, `/usr/local/psunix/default/64/` and copy `psldap-sunldap.so` into the `64/` folder. For example:

```
cd /usr/local/psunix/default/
mkdir 64
cp psldap-sunldap.so 64/
```

Note: On 64-bit systems, the value of the `nsslapd-pluginPath` is entered as `/usr/local/psunix/default/psldap-sunldap.so`; but the actual plugin, however, should be located in the `/usr/local/psunix/default/64` directory.

6.3.2 OpenLDAP

Note: Only OpenLDAP versions 2.2.x or later are supported.

The following instructions are intended for OpenLDAP v2.2.x. Details may vary depending on your version of the software.

Before you begin, note the following:

- When installing OpenLDAP 2.2.x, ensure that the `--enable-modules=yes` and `--enable-slapi=yes` configure options are set.
- The plugin will only work if clear-text passwords are sent to the server.
- The plugin will not work if the LDAPv3 Password Modify (RFC 3062) extended operation is used.

CAUTION: Stop the directory server before making these changes, otherwise the changes will be overwritten.

To configure OpenLDAP for transparent password synchronization:

1. Find the `slapd.conf` configuration file.

The file is usually located in `/etc/openldap/`.

2. Edit the configuration file, and add the following two lines:

```
plugin preoperation "<path-to-psldap>"  prepaswd_init ["-cfg" "<path-to-config-
file>"]
plugin postoperation "<path-to-psldap>"  postpaswd_init ["-cfg" "<path-to-
config-file>"]
```

For example:

```
plugin preoperation "/usr/local/psunix/default/psldap-openldap.so"
  prepaswd_init "-cfg" "/usr/local/psunix/default/psunix.cfg"
plugin postoperation "/usr/local/psunix/default/psldap-openldap.so"
  postpaswd_init "-cfg" "/usr/local/psunix/default/psunix.cfg"
```

If the configuration file is not specified for Unix-based servers, the default is `/etc/psunix.cfg`.

3. Save and close the file.

6.3.3 IBM Directory Server

The following instructions are intended for IBM Tivoli server. Details may vary depending on your version of the software.

To configure IBM Directory Server for transparent password synchronization:

1. Find the **ibmslapd.conf** configuration file.
2. Edit the configuration file, and add the following two lines:

```
ibm-slapdPlugin: preoperation <path-to-psldap> prepaswd_init ["-cfg" "<path-to-config-file>"]
ibm-slapdPlugin: postoperation <path-to-psldap> postpaswd_init ["-cfg" "<path-to-config-file>"]
```

For example:

```
ibm-slapdPlugin: preoperation /usr/local/psunix/default/psldap-ibmldap.so
prepaswd_init "-cfg" "/usr/local/psunix/default/psunix.cfg"
ibm-slapdPlugin: postoperation /usr/local/psunix/default/psldap-ibmldap.so
postpaswd_init "-cfg" "/usr/local/psunix/default/psunix.cfg"
```

If the configuration file is not specified for Unix-based servers, the default is /etc/psunix.cfg.

3. Save and close the file.

6.4 Filtering password change requests on a LDAP Directory Service trigger system

You can configure the LDAP password synchronization to include certain users, groups, and domains when they make password change requests on LDAP Directory Service trigger systems.

To configure the user filters:

1. Modify the **psldap** configuration file in /usr/local/psunix/default/psunix.d/.
2. Uncomment the following lines:

```
# filter-dn-include = {
#     "ou=finance,dc=example,dc=com";
#     "ou=hr,dc=example,dc=com";
# };
```

3. Edit the filters with specific UIDs, OUs, and DCs to include in password change requests.

Filters are in the following syntax:

```
"uid=<userid>,ou=<group>,dc=<domain>,dc=<com>";
```

Multiple filters can be used, with different levels of specificity. Users that pass any one of the filters will be included in password change requests.

For example:

```
filter-dn-include = {
    "dc=mydomain,dc=net";
    "ou=people,dc=example,dc=com";
    "uid=testuser,ou=finance,dc=example,dc=com";
};
```

4. Save the file and restart the LDAP service.

OID-LDAP Trigger

7

This chapter shows you how to set up transparent password synchronization for an Oracle Internet Directory LDAP (OID-LDAP) trigger system.

Before you begin, ensure that you have read and performed the steps in [Implementing Transparent Password Synchronization](#).

Hitachi ID Bravura Pass can intercept password changes on OID-LDAP trigger systems using a pre-change and post-change strength filter, **psldap**¹ for:


- [Unix-based OID-LDAP servers](#) (p49)
- [Windows-based OID-LDAP servers](#) (p51)

7.1 Unix-based OID-LDAP server

Before you start, you should have the encrypted communication key (COMMKEY), or a copy of the **idmsetup.inf** configuration file. The **idmsetup.inf** configuration file is located on the *Hitachi ID Bravura Security Fabric* server in the `\<instance>\psconfig\` directory.

To install the OID-LDAP password filter plugin (**psldap.so**) on a Unix-based OID-LDAP server:

1. If you did not select **Unix Installation Packages** when you installed the *Hitachi ID Connector Pack*, run **setup** on the *Hitachi ID Bravura Pass* server to modify your *Connector Pack* installation.

Ensure that the  icon is selected for the appropriate Unix package on the component selection page. Click **Next**, then complete the installation procedure.

2. Copy the **psunix-<os>.<cpu>.tar.gz** file from the unix directory to a scratch directory (such as /tmp) on the OID-LDAP server.
3. Log into the LDAP server with administrative privileges, and extract the files from the **psunix** archive. For example, type:

```
cd /tmp
tar -zxvf psunix-solaris9.sparc64.tar.gz
```

4. Run **install.sh** and select LDAP Transparent Synch option.

```
sh install.sh -c 4 [ -inf <path>/idmsetup.inf ]
```

¹The actual name of the filter varies depending on the target system type. See the appropriate section for details.

5. Follow the instructions displayed by the installer script.

In the configuration process, verify that the script correctly identifies your operating system type. If not, override it.

In the installation process, follow the instructions and input the information prompted by each input field. To skip a field, press **[Enter]** to use the default value.

6. Verify that the **psldap-oidldap.so** shared object file is copied to `/usr/local/psunix/default/`.
7. Ensure that `/etc/psunix.cfg` and `/etc/psunix.d/` configuration files are readable by the Oracle account:

```
chmod a+rx /etc/psunix.cfg
chmod -R a+rx /etc/psunix.d
```

8. Stop the OID-LDAP Application Server.
9. Copy the **psldap-oidldap.so** file from the `/usr/local/psunix/default/` directory to: `$ORACLE_HOME/lib` on the database server as **psldap.so**. You can place this binary somewhere else, but you must edit files in steps 10 and 11 accordingly.

`ORACLE_HOME` is the destination directory specified during the Oracle Application Server Infrastructure installation.

For example:

```
cp /usr/local/psunix/default/psldap-oidldap.so u01/app/oracle/lib/psldap.so
```

10. Edit the `listener.ora` file in `$ORACLE_HOME/network/admin` to permit access to the shared object. Add the path to the library to the oracle environment variable `EXTPROC_DLLS`. This must match the path to the **psldap.so** binary.

For example:

```
SID_LIST_LISTENER =
(SID_LIST =
(SID_DESC =
(SID_NAME = PLSExtProc)
(ORACLE_HOME = /u01/app/oracle)
(PROGRAM = extproc)
(ENVS="EXTPROC_DLLS=/u01/app/oracle/lib/psldap.so")
)
)
```

11. Edit `psldap-oidldap-plugin.sql`, in the `psunix-<os>.<cpu>/addon/transparent-synch/ldap` directory to set the path to the library in the plugin. This must match the path to the **psldap.so** binary.

For example:

```
CREATE OR REPLACE LIBRARY psldap_lib AS
'/u01/app/oracle/lib/psldap.so';
/
SHOW ERRORS
```

12. Install `psldap-oidldap-plugin.sql`. This file contains stored procedures needed for the plugin. You can install it, for example, by executing the command:

```
sqlplus ods/<odspassword> @<pathto psldap-oidldap-plugin.sql>
```

on the database server, where `ods` is the OID Database Schema owner

For example:

```
$ORACLE_HOME/bin/sqlplus ods/mypass @psldap-oidldap-plugin.sql
```

13. Set up the plugin in the LDAP server. This can be done either from the GUI by hand, or using the supplied `pluginreg.dat`. Using `pluginreg.dat` from the LDAP server, run the command:

```
ldapadd -p <portnum> -h <hostname> -D cn=orcladmin \
-w <orcladminpassword> -v \
-f <pathto psldap-oidldap-pluginreg.dat>
```

where:

- `port` is the port that the OID-LDAP server listens on - default is 389
- `hostname` is the host name of OID-LDAP server - `localhost` can be used.

For example:

```
$ORACLE_HOME/bin/ldapadd -p 389 -h myhost -D cn=orcladmin -w mypass -v -f psldap-oidldap-pluginreg.dat
```

14. Restart the Oracle listener:

```
lsnrctl stop
lsnrctl start
dbstart
```

7.2 Windows-based OID-LDAP server

To install the OID-LDAP password filter plugin (`psldap.dll`) on a Windows Server 2003-based LDAP server:

1. Log into the server hosting the OID-LDAP Application Server with administrative privileges.
2. Stop the OID-LDAP Application Server.
3. Copy `psldap-oidldap.dll` from `addon \transparent-synch\ldap\` on the *Bravura Pass* server to `$ORACLE_HOME/lib` on the OID-LDAP server as `psldap.dll`.

You can place this binary somewhere else, but you must edit files in steps 9 and 10 accordingly.

`ORACLE_HOME` is the destination directory specified during the Oracle Application Server Infrastructure installation.

4. Copy `psldap.cfg` from the `addon transparent-synch\ldap\` directory on the *Bravura Pass* server to `%ORACLE_HOME%\lib` on the OID-LDAP server.
5. Copy the `libidapi.dll` file from the `<instance>\lib\` directory on your *Hitachi ID Bravura Pass* server to `C:\Program Files\Hitachi ID\IDM Suite\<instance>\lib\`.

- Copy the **idapitool.exe** file from the `<instance>\lib\` directory on your *Bravura Pass* server to `C:\Program Files\Hitachi ID\IDM Suite\<instance>\`.

Edit **psldap.cfg** as follows:

comm-key Defines the private key used for encryption. This key must match the one set during installation on the *Bravura Security Fabric* server.

```
comm-key = "<encrypted commkey value>";
```

targetid This option is used to specify the ID of the target system associated with this interceptor.

```
targetid = "ldap";
```

libcurl The full path to the libcurl shared object required when using SSL. An empty value uses the system default location, otherwise the full path can be specified. A value of '0' disables libcurl which ultimately disables SSL and web proxy facilities.

```
libcurl = "0";
```

url The url option specifies the service endpoint of the API SOAP Service (idapisoap).

```
url = "http://host.domain.com/default/idapi";
```

user The user ID the API SOAP Service is configured to use.

```
user = "_API_USER";
```

psw The password the API SOAP Service is configured to use. Use the **idapitool** program to acquire this value from the known plain text value. See the *Bravura Security Fabric* Remote API guide for **idapitool** usage information.

```
psw = "the_encrypted_password_created_by_idapitool";
```

You can generate the encrypted password with the following command:

```
idapitool.exe -url http://host.domain.com/default/idapi -user _API_USER -  
psw Letmein1 -q
```

- Optionally, edit these keys:

proxy The proxy option specifies the address and port.

```
proxy = "http://idapi_proxy.mydomain.com:3128";
```

proxyuser The username to authenticate against the proxy. (optional)

```
proxyuser = "proxyuser";
```

proxypass The password to authenticate against the proxy. (optional)

```
proxypass = "proxypass";
```

capath The CA directory or file holding the root certificates to trust. This value is required if using SSL.

```
capath = "";
```

cert The certificate for client authentication. This value is optional when using SSL and may be used if client verification is required by the server.

```
cert = "ldap.crt";
```

ignore Whether or not to enforce strict name checking of the server certificate.

```
ignore = "0";
```

timeout Specifies the timeout when communicating with IDAPI SOAP service. The default timeout is 300 seconds.

```
timeout = "300";
```

retry-attempts Specifies the retry attempts for failed IDAPI calls. The default retry-attempts value is 2.

```
retry-attempts = "2";
```

retry-delay Specifies the retry delay between IDAPI calls. The default retry-delay is 5 seconds.

```
retry-delay = "5";
```

fail-if-unavailable Specifies if password changes should fail if the IDAPI SOAP service cannot be contacted. The default behavior is to always fail if IDAPI SOAP service is unavailable.

```
fail-if-unavailable = "true";
```

strength-check-only If this option is set to true, the password reset operation will not occur. The default value is set to true.

```
strength-check-only = "true";
```

- On the OID-LDAP server create the following registry keys:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Hitachi ID\IDM Suite\<instance>\
```

Entry name	PsldapCfg
Value	Path to psldap.cfg file
Data type	REG_DWORD

```
HKEY_LOCAL_MACHINE\SOFTWARE\Hitachi ID\IDM Suite\<instance>\
```

Entry name	PsInstallDir
Value	The full directory path of the psldap.cfg file
Data type	REG_SZ

- Edit the listener.ora file in \$ORACLE_HOME/network/admin to permit access to the shared object. Add the path to the library to the oracle environment variable EXTPROC_DLLS. This must match the path to the **psldap.so** binary.

For example:

```
SID_LIST_LISTENER =
(
  SID_LIST =
  (
    SID_DESC =
    (
      SID_NAME = PLSExtProc
      ORACLE_HOME = C:\oracle
      PROGRAM = extproc
      ENVS="EXTPROC_DLLS=C:\oracle\lib\psldap.dll"
    )
  )
)
```

10. Edit `psldap-oidldap-plugin.sql` to set the path to the library in the plugin. This must match the path to the `psldap.so` binary.

For example:

```
CREATE OR REPLACE LIBRARY psldap_lib AS
'C:\oracle\lib\psldap.dll';
/
SHOW ERRORS
```

11. Install `psldap-oidldap-plugin.sql`. This file contains stored procedures needed for the plugin. You can install it, for example, by executing the command:

```
sqlplus ods/<odspassword> @<pathto psldap-oidldap-plugin.sql>
```

on the database server, where `ods` is the OID Database Schema owner

For example:

```
$ORACLE_HOME/bin/sqlplus ods/mypass @psldap-oidldap-plugin.sql
```

12. Set up the plugin in the LDAP server. This can be done either from the GUI by hand, or using the supplied `pluginreg.dat`. Using `pluginreg.dat` from the LDAP server, run the command:

```
ldapadd -p <portnum> -h <hostname> -D cn=orcladmin \
-w <orcladminpassword> -v \
-f <pathto psldap-oidldap-pluginreg.dat>
```

where:

- `port` is the port that the OID-LDAP server listens on - default is 389
- `hostname` is the host name of OID-LDAP server - localhost can be used.

For example:

```
$ORACLE_HOME/bin/ldapadd -p 389 -h myhost -D cn=orcladmin -w mypass -v -f psldap
-oidldap-pluginreg.dat
```

13. Restart the Oracle listener using the Windows Service Control Manager.

7.3 Troubleshooting

If things do not work correctly, execute the following code in sqlplus:

```
declare
errmsg VARCHAR2(255);
result NUMBER;
begin
result := 0;
result := pwd_plugin.check_password('someuser', 'somepasswd', errmsg);
end;
/
```

This will help provide more details about how the shared object is failing.

OS/390 or z/OS (RACF, TopSecret, ACF2) with Mainframe Connector

8

Hitachi ID Mainframe Connector can intercept password changes on OS/390 or z/OS mainframes, with RACF, ACF2 or TopSecret security software. This is done by inserting an exit trap into the security system, and by installing an authorized task which starts at IPL.

The combination of an exit and task apply password strength rules defined on the *Hitachi ID Bravura Pass* server to all new password selections, made using any user interface, natively on MVS or OS390. The task forwards a request for synchronization to the *Bravura Pass* server after every successful mainframe password change.

Before installing the exit and task on your mainframe, be sure to inform your users that:

- All mainframe password changes for users who appear in the *Bravura Pass* server's user database will be subjected to the password policy enforced on the *Bravura Pass* server.
- When users who are defined on the *Bravura Pass* server change their passwords on the mainframe, their new password will be automatically applied to all of their other accounts, on other systems defined on the *Bravura Pass* server.

Refer to the “*Mainframe Connector* Installation Guide” ([mainframe-connector.pdf](#)) for detailed instructions about installing and configuring the exit and task on your security system (RACF, ACF2 or TopSecret).

Note: If you install *Mainframe Connector*, but do not install the password exit in your security product, then *Bravura Pass* will be able to manage mainframe passwords, but transparent password synchronization will not be triggered by native mainframe password changes.

8.1 Configuring the Password Manager service for transparent synchronization

The interceptor installed with *Hitachi ID Mainframe Connector* uses a legacy protocol to communicate with the Password Manager service (idpm). You must configure the Password Manager service (idpm) for backward compatibility:

- Set the following field to use the port configured for this interceptor (default 3333):

Enable this port for backward compatibility (to communicate with older interceptors/triggers). Must be different from Port number above

- Add a CIDR mask address for the trigger system in the following setting:

Comma-delimited list of IP addresses with CIDR bitmask that are allowed to send socket requests

WARNING!: If using load balancers, do not configure any SSL options for transparent synchronization traffic. SSL options should only be configured on load balancers for WebUI traffic, not transparent synchronization. Transparent synchronization is encrypted using a proprietary encryption algorithm. Contact support@Hitachi-ID.com for more details.

See the [Password Manager Service \(idpm\)](#) for more information about the Password Manager service.

Hitachi ID Bravura Pass can intercept password changes on an IBM OS/400 system. This is done by installing an exit program, `pspwdexit_v5r4m0` or `pspwdexit_v7r1m0`, which implements the `QIBM_QSY_VLD_PASSWRD` exit point on the OS/400 system. The exit program informs *Bravura Pass* when a password is changed. It also allows *Bravura Pass* to check who is changing whose password. If a user tries to change another user's password, the attempt is blocked and a warning is sent to a specified administrator.

Bravura Pass ships with two exit programs for OS/400, `pspwdexit_v5r4m0` for the IBM i7.1 operating system and `pspwdexit_v7r1m0` for the IBM i7.2 operating system.

This chapter details how to configure transparent password synchronization on an OS/400 system by:

1. [Creating and applying a password policy \(p58\)](#)
2. [Installing and configuring pspwdexit \(p59\)](#)
3. [Verifying the configuration \(p61\)](#)

This chapter assumes you have set up an os400 target system and tested it's configuration according to the Connector Pack Integration Guide.

9.1 Creating and applying a password policy

Before installing and configuring the *Hitachi ID Bravura Pass* transparent password synchronization software on the IBM OS/400 server, you need to create a password strength policy for the OS/400 server:

1. Configure a password strength policy.
Set the **Maximum number of lowercase letters** to 0. Passwords on the OS/400 system cannot include lowercase letters. Configure other parameters as required.
2. Create a target system group
 - Ensure that the **Use transparent password synchronization** checkbox is selected.
 - Select the password policy that you created.
3. Make the OS/400 server a member of the target system group.

See the [Bravura Security Fabric Documentation](#) for more information about configuring password strength policies and target system groups.

Once you have applied the password policy and installed the exit programs on an OS/400 server, be sure to inform your users that:

- All future password changes are subjected to the password policy enforced by the *Bravura Pass* server.
- When they change their password on the OS/400 server, their new password is automatically applied to all their other accounts managed by the *Bravura Pass* server.
- Their new password must be all uppercase.

9.2 Installing and configuring pspwdexit

The `pspwdexit_v5r4m0` and `pspwdexit_v7r1m0` programs are installed in the `\<instance>\addon\transparent-synch\as400` directory.

To install and configure the `pspwdexit_v5r4m0` for IBM i7.1 or `pspwdexit_v7r1m0` for IBM i7.2 program:

1. From the *Hitachi ID Bravura Pass* server, establish a connection to the IBM OS/400 server using the 5250 emulator software.
2. If the OS/400 server already has another version of the transparent synchronization interceptor installed on it, you must remove it by running the following command:

```
DLTLIB PSYNCH
```

3. Create a PSPWDEXIT save file.

```
CRTSAVF FILE(QGPL/PSPWDEXIT)
```

4. Transfer the `pspwdexit_v5r4m0` or `pspwdexit_v7r1m0` to the OS/400 server, so that it overwrites the placeholder file you created in step 3.

(a) Navigate to the `\<instance>\addon\transparent-synch\as400` directory.

(b) From a Windows command prompt:

```
ftp <OS/400 server>
> binary
> put <exit program> QGPL/PSPWDEXIT (replace
> quit
```

Note that there is no closing parenthesis on the `put` command.

5. Switch back to the 5250 emulator.

6. Restore the PSYNCH library:

```
RSTLIB SAVLIB(PSYNCH) DEV(*SAVF) SAVF(QGPL/PSPWDEXIT)
```


7. Change the following system value:

```
CHGSYSVAL SYSVAL(QPWDVLDPGM) VALUE(*REGFAC)
```

then add the exit program by typing on one line:

```
ADDEXITPGM EXITPNT(QIBM_QSY_VLD_PASSWRD) FORMAT(VLDP0100) PGMNBR(*HIGH) PGM(
  PSYNCH/PSPWDEXIT) THDSAFE(*YES) TEXT('Password Manager Password Exit Program')
```

8. Configure the following data areas:

- Set TARGETID to the target ID of the OS/400 server as it is configured in *Bravura Pass*:

```
CHGDTAARA DTAARA(PSYNCH/TARGETID) VALUE('<target ID>')
```

- Set PSSERVER to the address of the *Bravura Pass* server:

```
CHGDTAARA DTAARA(PSYNCH/PSSERVER) VALUE('<HiPM server address>')
```

- Set PSPOINT to 3334:

```
CHGDTAARA DTAARA(PSYNCH/PSPOINT) VALUE('3334')
```

- Set COMMKEY to the *Bravura Pass* server communication key (or Master Key) value:

```
CHGDTAARA DTAARA(PSYNCH/COMMKEY) VALUE('<commkey value>')
```

- Set MSGUSER to the administrative user who will receive system messages:

```
CHGDTAARA DTAARA(PSYNCH/MSGUSER) VALUE('<user>')
```

9. Modify the PSYNCH library's object authorization.

To modify the authority of the objects in the PSYNCH library:

(a) Type:

```
WRKLIB LIB(PSYNCH)
```


(b) Enter 12 (work with objects).

(c) For each object in the PSYNCH library:

- Select 2 to edit authority.
- Ensure the *PUBLIC user has its object authority set to *USE. Modify accordingly.

9.3 Configuring the Password Manager (idpm) service

To allow external servers access to the Password Manager service (idpm) on the primary *Hitachi ID Bravura Pass* server you must also add a CIDR mask address for the trigger system.

1. Click **Manage the system** → **Maintenance** → **Services**
2. Select  **Hitachi ID (idpm) Password Manager Service**.
3. Add a CIDR mask address for the trigger system in the following setting:

Comma-delimited list of IP addresses with CIDR bitmask that are allowed to send socket requests

See the [Password Manager Service \(idpm\)](#) for more information about the Password Manager service.

You are now ready to verify the installed software.

9.4 Verifying the configuration

Verify that the transparent password synchronization trigger is working as expected. Log into the IBM OS/400 server and change the password of a user that *Hitachi ID Bravura Pass* is managing. Ensure that the password change was captured by *Bravura Pass* and propagated to other target systems.

9.5 OS/400 system components

All the values are type CHAR, so the values should be encased in single-quotes.

Table 9.1: OS/400 system components

Component	Description
PSPWDEXIT	<p>The exit programs work with the QIBM_QSY_VLD_PASSWORD exit point. Use the ADDEXITPGM command to add the exit point.</p> <p>You also need to set the QPWDVLDPGM system value to *REGFAC. The advantage of this, is that it can check who is changing whose password. Users are not allowed to change others' passwords. If this is attempted, a warning message is sent to an administrator MSGUSER.</p>
MSGUSER	<p>The user that administrative messages are sent to. If the user is not specified, messages are sent to QSYSOPR. If a nonexistent user is specified, messages are not sent. Field length is 10.</p>

... continued on next page

Table 9.1: OS/400 system components (Continued)

Component	Description
MSGLEVEL	The administrative message level settings that are logged. The default level is 3. The value can be set to the following: 0 (No logging), 1 (Error), 2 (Warning), 3 (Notice), 4 (Info), 5 (Debug)
PSSERVER	The <i>Hitachi ID Bravura Pass</i> server's network name or IP address. Field length is 50.
PSPORT	The Password Manager service (idpm) port number. Field length is 5.
COMMKEY	The <i>Bravura Pass</i> communication key (or Master Key) in the encrypted format. Field length is 80.
TARGETID	The target ID of the IBM OS/400 server as it is identified in <i>Bravura Pass</i> . Field length is 80.
TIMEOUT	The default timeout value for connecting to Password Manager service is 8 seconds. If the network is slow, a greater value may be needed. Field length is 2.
FAILPPDOWN	The behavior when Password Manager service cannot be contacted. By default it is 0 and the CHGPWD is still permitted if it cannot be contacted. If set to 1, CHGPWD is rejected if Password Manager service cannot be contacted.

User registration

10

When transparent password synchronization is implemented, it is important for users to understand the new password composition rules that *Hitachi ID Bravura Pass* enforces over native password changes made on individual systems. Users also need to understand that password synchronization takes place automatically after they change their own password on a trigger system.

Bravura Pass incorporates a web-based registration module, intended to prompt users for active confirmation that they understand what transparent synchronization does for them. When the *Enable password synchronization* (PSR) module is activated, users are *not* affected by transparent password synchronization until they actively “register” for it.

The *Enable password synchronization* (PSR) module is *disabled* by default. You must enable it to allow users to access this feature.

WARNING!: Hitachi ID Systems recommends that transparent synchronization be applied to all users. Transparent registration can be used to implement transparent synchronization on a user-by-user basis; however, there are incompatibilities between transparent synchronization registration and IVR and the *Bravura Pass* API. If transparent synchronization registration is enabled, IVR and the *Bravura Pass* API may not be able to find users.

It is recommended that transparent synchronization registration only be used as an educational tool. If this module is *not* enabled, all *Bravura Pass* users are automatically subjected to transparent synchronization when it is activated.

To configure transparent password synchronization registration:

1. Click **Manage the system** → **Modules** → **Enable password synchronization (PSR)**.
2. Turn on the **PSR ENABLED** setting.
3. If required, configure event options, listed in [Enable password synchronization \(PSR\) module events that launch interface programs](#), that trigger external programs.
4. Click **Update** to submit the changes.
5. Restart the Password Manager service (idpm) to apply your settings.

CAUTION: The Password Manager service must be restarted after transparent synchronization is enabled. If it is not restarted, users may remain automatically subjected to transparent synchronization despite not being actively registered for it.

Table 10.1: *Enable password synchronization* (PSR) module events that launch interface programs

Option	Description
PSR CANCELLATION SUCCESS	A user disables transparent password synchronization for himself.
PSR REGISTRATION FAILURE	A user tries to register for transparent password synchronization, but fails for some reason.
PSR REGISTRATION SUCCESS	A user registers for transparent password synchronization.

10.1 Enabling Transparent Password Synchronization as a self-service user

When you change your password on certain systems, *Hitachi ID Bravura Pass* can:

- Subject the password to an organization's password policy and reject it if it does not meet security requirements.
- Trigger transparent password synchronization for other accounts on other systems that belong to you. This means you may have the same password for a group of accounts, or possibly all of your accounts.

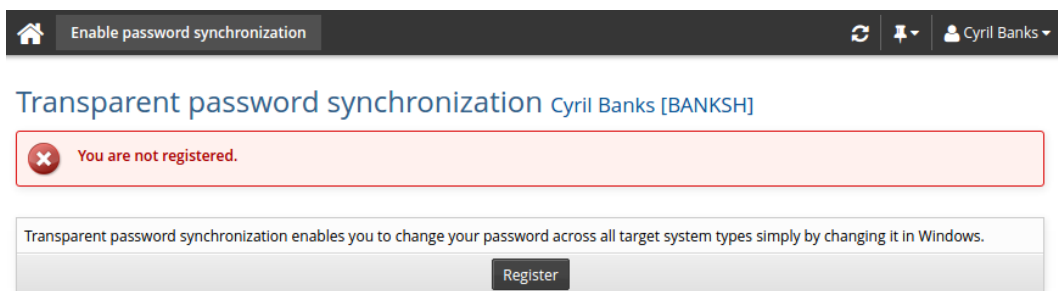
Automatic (or transparent) synchronization is currently available for the following types of trigger systems:

- Windows servers
- Windows domains
- Microsoft Active Directory domains
- OS/390 mainframe with RACF security
- OS/390 with ACF2 security
- OS/390 with TopSecret security
- Unix servers
- Netscape (iPlanet or Directory Server) LDAP servers
- Sun Microsystems (Sun ONE) LDAP servers
- OpenLDAP LDAP servers
- IBM OS/400 servers

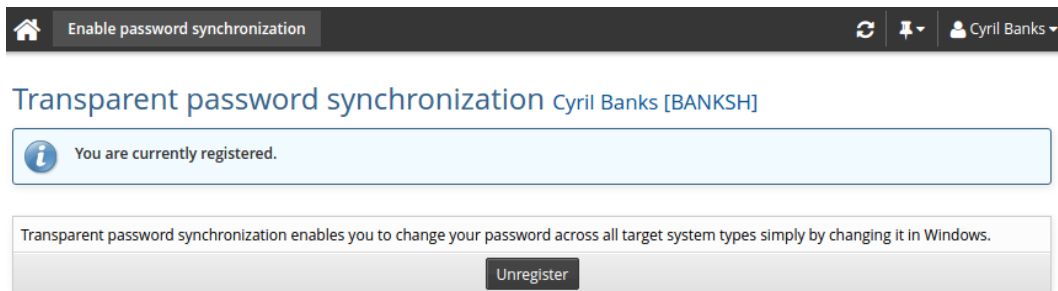
If transparent password synchronization has been set up for a system that you use, you must register before it is activated.

To register for transparent password synchronization:

1. From the main menu , click **Enable password synchronization**.



2. Read the information on the **Enable password synchronization** page and ensure that you understand it.
3. Click **Register**.



To disable transparent password synchronization, click **Remove me** on the **Enable password synchronization** page.

Note: This change only takes affect after the transparent password synchronization service is stopped and restarted, which usually takes place during *auto discovery*. Password changes continue to trigger synchronization until the update.

Once you are registered for transparent password synchronization, you should expect the following changes:

- When you change your own password on a supported system, the new password must meet the security requirements defined on the *Bravura Pass* server.
To see these requirements, access the **Enable password synchronization** page.
If you try to change your password to a new value that violates the security policy, the password change fails. Depending on the type of system that you change your password on, you may or may not get an informative error message.
- After successfully changing your password on the targeted system, *Bravura Pass* automatically changes your password on every other system that it supports to the same new value. This synchronization

process is intended to simplify your passwords, by eliminating the need to remember a different password for every system.

Part III

REFERENCE

Password Manager Service (idpm)

11

Description

The **idpm** service works in conjunction with trigger programs and libraries on various systems, to implement transparent password synchronization.

Trigger systems establish a secure, encrypted TCP connection with the **idpm** service on the *Hitachi ID Bravura Pass* server. Connecting programs may:

- Prompt the **idpm** service to evaluate a new password selected by a user, and determine whether it should be accepted (complies with password strength policy), or rejected.
- Prompt you for a textual description of the current password policy.
- Instruct the **idpm** service to synchronize a user's passwords to a new value on all systems where the user has a login account.

The **idpm** program can also extend the functionality of web-based password management by allowing failed password changes to be queued for automatic retry. Password changes may then be implemented automatically for the accounts when the failed target system becomes available.

By default, the **idpm** service is available to all users when transparent password synchronization is activated.

During auto discovery, **idpm** queues password changes and SESSLOG entries. It will run strength checks for immediate response, but will not write to the database. After the service is taken off hold, it will run through the queued commands and execute them and delete the temporary file.

Configuration

The service is automatically installed and started on the *Hitachi ID Bravura Pass* server during setup. You can modify the following parameters related to this service on the **Service information** page:

Table 11.1: idpm service options

Option	Description
Required parameters:	
Port number this service is running on	<p>This defaults to 3334. This port is used for communication with interceptors installed from <i>Hitachi ID Connector Pack</i> 1.1 and newer. To enable communication with older interceptors, you must set a backward compatibility port, as explained below in this table.</p> <p>The port number selected must not be in use by any other service, including other instances of the Password Manager service (idpm).</p>
Maximum number of concurrent threads the service should run	The number of concurrent password synchronizations the Password Manager service can execute. The default is 8. You should vary this according to the load limit of the <i>Bravura Pass</i> server and the number and type of target systems.
Timeout for connection in seconds	The amount of time the Password Manager service will wait, once it has made a socket connection and sent a synchronization request, before killing the connection. The default is 600.
Comma-delimited list of intervals (in minutes) to wait before retrying failed requests	<p><i>Bravura Pass</i> will retry failed password change requests using the specified time intervals. This is useful in situations where a target system is temporarily down. You can vary the amount of time that <i>Bravura Pass</i> waits before retrying. The default intervals are:</p> <p>2,2,4,4,8,8,16,16,32,32,64,64,128,128,256,256,256,256,256,256,256,256,256,256,256</p> <p>This means <i>Bravura Pass</i> will retry a failed request for a total time of 4604 minutes, or 3.2 days.</p>
Optional parameters:	
Comma-delimited list of IP addresses with CIDR bitmask that are allowed to send socket requests	<p><i>Bravura Pass</i> only accepts socket requests through the authorized IP/CIDR range defined in this field. Password synchronization interceptors that need to access idpm must be defined in this field, otherwise, their requests will be rejected.</p> <p>The default is 127.0.0.1/32,::1/128</p>
Perform password strength check on non-<i>Bravura Pass</i> users	Select the Enable checkbox if you want the Password Manager service to enforce the password strength rules defined in <i>Bravura Pass</i> , when a non- <i>Bravura Pass</i> user (not in the USER table) or a user who is not registered for transparent password synchronization changes his or her password on a trigger system.
Perform transparent password synchronization on locked out users	<p>Select the Enable checkbox if you want the Password Manager service to enforce the transparent password synchronization on locked out users.</p> <p>Note: The locked out users profile status is not affected by selecting or deselecting this option.</p>

...continued on next page

Table 11.1: idpm service options (Continued)

Option	Description
Enable this port for backward compatibility (to communicate with older interceptors/triggers). Must be different from Port number above	<p>This port facilitates communication with interceptors that use legacy protocol. This includes:</p> <ul style="list-style-type: none"> • Unix and LDAP interceptors • OS/400 interceptors installed with <i>Bravura Pass</i> 7.0 or earlier • Interceptors installed with <i>Hitachi ID Mainframe Connector</i> • Any interceptor installed with <i>Bravura Pass</i> version 6.x or older. <p>You must use a different port number than the one specified for Port number this service is running on.</p> <div> <p>Note: If the wrong ports are used, connections are dropped and the passwords are not synchronized.</p> </div>

Password Manager service events that launch interface programs lists Password Manager service events that can trigger email or updates on IT Service Management (Ticket) systems.

Table 11.2: Password Manager service events that launch interface programs

Option	Description
IDPM FINDUSER FAILURE	The Password Manager service attempts to check a password against password strength rules, and does not find the user in the <i>Bravura Pass</i> database.
IDPM GROUP FAILURE	The Password Manager service attempts to synchronize a group of passwords for a user, and fails on at least one of the passwords after the specified sequence of retries.
IDPM GROUP FIRST TRY DONE	All password operations in a request have been attempted once.
IDPM GROUP NOOP	Transparent synchronization request is received for a user that does not have any other accounts to synchronize.
IDPM GROUP SUCCESS	Every password is synchronized for a user by the <i>Bravura Pass</i> interceptor service.
IDPM REQUEUE	A single password change fails, and is queued for retries on the <i>Bravura Pass</i> server.
IDPM SINGLE FAILURE	The Password Manager service attempts to change a single password for a user, and fails after the specified sequence of retries.
IDPM SINGLE SUCCESS	A single password is changed for a user by the Password Manager service.

... continued on next page

Table 11.2: Password Manager service events that launch interface programs (Continued)

Option	Description
IDPM STRENGTH FAILURE	The Password Manager service rejects a user's password, because it failed at least one password policy rule. This exit point is useful for automatically sending the user a reminder describing the password policy.
IDPM STRENGTH SUCCESS	A password strength check is successful.

Table 11.3: idpm command-line arguments

Argument	Description
-h	Displays usage information.
-v	Displays version number only.
-clearqueue	Clears the queue. The service must be manually stopped before using this option. WARNING!: This operation removes <i>all</i> records of outstanding requests.
-config	Displays service configuration information.
-server	Run the service in server mode.
-start	Starts the service.
-stop	Stops the server/service.

11.1 Allowing external communication with *Bravura Pass*

Communication with the *Hitachi ID Bravura Pass* server is controlled by an access control list. When you install the Password Manager service (idpm), it automatically sets the access control for the local server, "127.0.0.1/32,::1/128", with all allowable access so that it can perform its operations with no modification.

To allow external servers access to the Password Manager service on the primary *Bravura Pass* server, you must set up the **Comma-delimited list of IP addresses with CIDR bitmask that are allowed to send socket requests** on the Password Manager service service information page.

The external server ACLs are based on a server's IP address defined using Classless Inter-Domain Routing (CIDR) notation, which allows for address ranges.

See [CIDR notation](#) for more detailed information about CIDR notation.

11.2 Testing

To verify that the service is running, try to connect to the port number it uses with Telnet. For example, from the *Hitachi ID Bravura Pass* server, type:

```
telnet localhost 3334
```

You should see nothing returned.

WARNING! If using load balancers, do not configure any SSL options for transparent synchronization traffic. SSL options should only be configured on load balancers for WebUI traffic, not transparent synchronization. Transparent synchronization is encrypted using a proprietary encryption algorithm. Contact support@Hitachi-ID.com for more details.



11.3 Monitoring transparent password synchronization

11.3.1 Managing the Password Manager Service queue

Monitor transparent password synchronization by running synchronization reports. You may need to remove queued items to improve performance.

By selecting unwanted queued transparent synchronization items and removing them, the items will not be retried to synchronize to their destination targets and will be marked as failed synchronization.

To remove items from the transparent synchronization queue:

1. Click **Manage the system** → **Maintenance** → **Services**.
2. Select  **Hitachi ID (idpm) Password Manager Service**.
3. Select  **Manage work queue** in the bottom table.
4. Enter search criteria and click **Search**.
5. Select the items you want to remove and click **Cancel**.

11.3.2 Monitoring transparent password synchronization on Windows servers

Monitor the health of the Hitachi ID Password Change Notification Module on Windows NT PDCs and Microsoft Active Directory DCs. Run `netstat -an` to see whether there are many (more than 20 or 30) TCP connections pending between the PDC/DC and the *Hitachi ID Bravura Pass* server. If so, there may be a problem with the *Bravura Pass* server.

Sometimes, you may find that the Password Manager service (idpm) appears to be failing to synchronize passwords changed on a Windows server. In many cases this is caused by the Password Change Notification Module interceptor (`psintcpt.dll`) timing out before it has communicated a password change to `idpm`. The default timeout period for `psintcpt.dll` is 60 seconds. You can extend this timeout period in the `intcptsvc.cfg` file. See [intcptsvc](#) for more information.

WARNING!: It is strongly recommended that you edit `intcptsvc.cfg` only under the direction of a Hitachi ID Systems support technician.

API SOAP Service (idapisoap)

12

Description

The API SOAP Service (idapisoap) provides access to the *Bravura Pass* API Service (idapi) with the WWS web service API. It is installed and started on the *Hitachi ID Bravura Pass* server during setup.

Some organizations use the SOAP API to build their own customized front-end to *Bravura Pass*.

Refer to the *Bravura Security Fabric* Remote API manual to learn about the SOAP API.

Requirements

The API SOAP Service (idapisoap) requires:

- Windows Server 2012, or
Windows Server 2012 R2
- Microsoft .NET Framework 4.5+

The API SOAP Service is dependent on the API Service (idapi). The PSLang API, IIS.NET API, and WWS API call into this service.

Configuration

The service is automatically installed and started on the *Hitachi ID Bravura Pass* server during setup.

You can modify the following parameters related to the API SOAP Service on the **Service information** page:

Table 12.1: idapisoap service options

Option	Description
Endpoints for the IDAPI SOAP native service	The URL that the API SOAP Service will listen in on. An optional port can be appended to the host and preceded by a colon. Multiple endpoints can be defined in a comma separated list. To secure the endpoint, configure it to listen on HTTPS.
Use system setting:	Will be set to the following: <code>http://<IP>/default/idapi,http://localhost/default/idapi, http://<IP-fqdn>/default/idapi.</code>

Note: The "server" URL fragment has to be the IP, hostname or FQDN (fully qualified domain name) of the server. If the connection is secured with TLS (https protocol), only a FQDN listed in the TLS server certificate used on the IIS server will allow a secure remote connection.

Table 12.2: idapisoap command-line arguments

Argument	Description
-h	Displays usage information.
-v	Displays version number only.
-binding <binding>	Specify the binding type: 0 – basicHttpBinding 1 – wsHttpBinding (default)
-config	Displays service configuration information
-endpoint <endpoint>	The endpoint that this SOAP service listens on
-mex type <mex type>	The metadata exchange method: 0 – None 1 – MexHttpGet 2 – Mex (only for wsHttpbinding) 4 – HttpGet (default)
-server	Run the service in server mode.

... continued on next page

Table 12.2: idapisoap command-line arguments (Continued)

Argument	Description
-start	Starts the service.
-stop	Stops the server/service.

To configure wsbinding and mex, set the service like this:

```
idapisoap -binding 1 -mextype 2
```

where the API service class can be generated from *<endpoint>*; for example, `http://<IP or fqdn>/hiim/idapi`.

To configure httpbinding and HttpGet, set the service like this:

```
idapisoap -binding 0 -mextype 4
```

where API service class can be generated from the WSDL *<endpoint>/wsdl*; for example, `http://<IP or fqdn>/hiim/idapi/wsdl`.

Note: Changing the SOAP connection type by using the "binding" and "mextype" arguments is only required if the remote SOAP tooling used requires it. Do not change the default connection type if testing the collection of the WSDL using a browser, otherwise you will receive a generic server error (HTTP 500) instead of the WSDL.

12.1 Metadata exchange

The API SOAP Service (idapisoap) has a couple of methods of exchanging metadata. When using the default httpget, the metadata exchange URL is:

```
service_endpoint/wsdl
```

For example:

```
https://<IP or fqdn>/INST/idapi/wsdl
```

If the API SOAP Service uses MexHttpGet, the URL is:

```
https://<IP or fqdn>/INST/idapi/metadata/wsdl
```

If the API SOAP Service uses wsHttpBinding, then Mex metadata exchange can be used, and the metadata exchange URL is same as service endpoint.

Changes to the service are not effective until the service is restarted.

12.2 Binding

If the end point is secured with SSL (https), then the binding for IIS is used to map the certificate to the secure endpoint. The service provides the equivalent functionality with the IIS .NET (via idapiservice.svc).

12.2.1 IIS .NET versus WWS

Differences between .NET and WWS include:

- IIS .NET requires more resources and memory.
The default behavior for IIS applications is to recycle frequently. As a result, initial start up is slow.
- IIS .NET does support additional options such as IP filtering, ws binding with message security.
- WWS requires less resources and still provides IDAPI access when IIS is stopped.

Description

The API Service (idapi) enables client programs to access *Bravura Pass* workflow and provisioning features programmatically. Client programs communicate with the service using:

- SOAP (Simple Object Access Protocol)

Some organizations use the *SOAP API* to build their own customized front-end to *Hitachi ID Bravura Pass*.

- PSLANG functions

Bravura Pass plugins can use PSLANG to retrieve additional information about users, resources, and requests from the API Service (idapi).

Refer to the *Bravura Security Fabric Remote API* manual to learn about the *Bravura Pass* API. The *Bravura Security Fabric Remote API* also contains information about installing and using the SOAP API. Refer to the *PSLang Reference Manual* for more information about PSLANG.

Requirements

The API Service is dependent on the Database Service (iddb). If you restart the database service, you must restart the API Service. See [Database Service \(iddb\)](#) in the *Reference Manual* for information about *iddb* and stopping the service.

Configuration

The service is automatically installed and started on the *Hitachi ID Bravura Pass* server during setup.

You can modify the following parameters related to the API Service on the **Service information** page:

Table 13.1: idapi service options

Option	Description
Timeout for connection in seconds	The timeout to use for connections. The default is 60 seconds.

Table 13.2: idapi command-line arguments

Argument	Description
-h	Displays usage information.
-v	Displays version number only.
-config	Displays service configuration information.
-server	Run the service in server mode.
-start	Starts the service.
-stop	Stops the server/service.

The password interceptor service, **intcptsvc**, is part of the Hitachi ID Password Change Notification Module, which also includes the **psintcpt.dll**. The service queues DLL requests and communicates with the Password Manager service (**idpm**). The DLL is loaded into Windows Local Security Authority (LSA) policy to capture native password changes.

This chapter explains how you can extend functionality of the Hitachi ID Password Change Notification Module to include using different interceptor settings based on user's DN, group and attributes.

This service is installed by **intcpt.msi** or **intcpt-64.msi** on a Windows transparent password synchronization trigger. To learn how to install the service, see [Windows Trigger](#).

You can configure the interceptor service, **intcptsvc** to include or exclude certain users when they make password change requests on Windows trigger systems. The excluded requests are not sent to the Password Manager service (**idpm**), but are instead processed by the Windows password change facility as usual. This can be used to reduce network traffic between the trigger system and **idpm**.

You can configure the Password Change Notification Module filter using the configuration file, **intcptsvc.cfg**, located in:

<Program Files path>\Hitachi ID\Password Filter\service

See the **intcptsvc.cfg** file for basic instructions, and samples located in:

<Program Files path>\Hitachi ID\Password Filter\samples

CAUTION: Do not alter **instsvc.cfg** unless you know what you are doing.

For deployments on Windows NT environments, areas that are commented out should not be edited, due to limitations of Windows NT.

[Sample intcptsvc.cfg file](#) shows a modified configuration file for an Active Directory environment. In this file:

- The **QueryAttributes** group has been edited to specify attributes to query.

Using **QueryAttributes = All** may slow down interceptor performance because it needs to retrieve all user attributes that have values (non blank). This option is good for designing the configuration at the early stage. You can then specify individual attributes once you know what you are looking for.

- `Bypass` defines matching that will *not* be sent to the Password Manager service for password strength checking and password synchronization.

When used with the `NotAny` operation, it *includes* the defined accounts; that is to say: “do not skip these accounts”.

- `BypassNotify` defines matching accounts that will *not* be sent to the Password Manager service for password synchronization.
- The configuration file maps to one Password Manager service only. In the case below, it is mapped to the service at IP address 10.0.5.8, port 3334. This can be the virtual IP of a Network Load Balancer.
- One physical Active Directory DC maps to two logical Target IDs in *Bravura Pass*; `End_Users` and `Admin_Users` logical targets in *Hitachi ID Bravura Pass*. These two targets map to different target groups, which have their own password policies.

The idea here is that the Active Directory accounts meet administrator criteria:

- Is, at least, a member of specified administrator groups; specified by the Bypass operation "`NotAny`" "`memberof`", which has the effect of including defined accounts.
- Is *not* the specified account names; `Guest` and `Krbtgt`.
- Is not a disabled account; specified by `userAccountControl match "([0-1])*10"`

If an administrator is a member of Domain Administrators group, his changing password will be examined by the Password Manager service, but it will not be synchronized to other associated targets. Instead, his password will be changed locally only.

Listing 14.1: Sample `intcptsvc.cfg` file

```

1 # KVGROUP-V2.0
  config "" = {
3   PMServer = {
      Address = 10.0.5.8;
5     Port = 3334;
      ConnectTimeout = 10; # default timeout 10 seconds
7     # How many times retry if connection or communication failed
      MaxRetry = 10; # default maximum retry times is 10;
9     RetryDelay = 5; # default interval between each retry
    };
11  # The total timeout for doing password strength check on all targets
    StrengthCheckTimeoutSeconds = 60;
13  # How many time retry if IDPM server returns recoverable error
    StrengthCheckRetry = 3;
15  # Queue polling time
    QueuePollTimeSeconds = 60;
17  # Queued item will discarded if exceed this setting
    NotificationExpireSeconds = 86400;
19  # Discard this notification if it has been tried max times
    DiscardNotificationAfterTried = 100;
21  # if an exception occurred, 1 -- return StrengthCheck succeeded, 0 -- return
    strength check failed
    BypassStrengthOnException = 1;

```

```

23 MaxSessionLifeSeconds = 60; # default max life time for session is 60

25 # regular expression to bypass both strength check and password change notification
    based on
    # sAMAccountName before retrieving account attributes.
27 # Default setting is for bypassing empty user name and computer account
    sAMAccountNameBypassRegEx = "^\\s*$|^\\.\\.\\$+\\$";

29
31 # ADsPath has the syntax as: LDAP://HostName[:PortNumber][//DistinguishedName]
    # following variables can be used for the HostName and DistinguishedName
    # %PDC%    -- primary domain controller
33 # %DC%      -- default domain controller
    # %DN%     -- default naming context
35 ADsPath = "LDAP://%DC%/%DN%";

37 # LDAP search filter for querying account's attributes, the account name variable %
    USER% can be used in the filter
    # %USER%    -- the account name
39 # ADsSearchFilter = "(&(objectCategory=person)(objectClass=user)(sAMAccountName=%
    USER%))";
    ADsSearchFilter = "(sAMAccountName=%USER%)";

41
43 # QueryAttributes defines attributes that will be used by the PSLang BypassCheck
    # function and the Target-based bypass check for integrating with Active Directory.
    # There is one inherent attribute, '_AccountName_'; this is the only
45 # attribute that can be supported on a non-Active Directory platform.
    # Following are sample attributes for and Active Directory provider.
47 # Specifying "All" as QueryAttributes, instead of a list of attributes,
    # indicates to query all possible attributes for the user.
49 # QueryAttributes = All;

51
53 # Active Directory attributes sample
    QueryAttributes = {
55     "distinguishedName";
57     "userAccountControl";
59     "memberOf";
61     "objectSid";
63     "pwdLastSet";
65     "replPropertyMetaData";
67     "whenCreated";
69     "whenChanged";
71     "logonHours";
73     "lastLogon";
    };

    Targets = {
        End_Users = {
            # IDPM return code:
            # 0 -- Communication failure
            # 1 -- Communication timeout
            # 99 -- IDPM service internal database access failure
            # 100 -- Weak password

```

```

75  # 101 -- Access denied ( ACL )
    # 102 -- User not found
77  # 103 -- User has been locked out
    # 104 -- User has not been registered
79  # 105 -- User has been disabled
    # 106 -- Account not specified
81  # 107 -- TargetID not specified
    # 119 -- Invalid operation code
83  # 120 -- Invalid request version
    # 200 -- Good password
85  CheckStrengthFailIfIDPMReturn = { 100; };
    CheckStrengthOnly = 0;
87  # If the target longid isn't the default sAMAccountName, define the longid as:
    # LongID = "%sAMAccountName%";
89  # LongID = "DomainName\\%sAMAccountName%";
    # LongID = "%distinguishedName%";
91  LongID = "LongIDMatchesPMTarget";

93  # Target based bypass setting is based on 'Condition Group', the 'Condition
    Group' has below definitions:
    #
95  # Defines the conditions to bypass both strength check and password change
    notification
    # Bypass "LogicalOperation" = {
97  #   ConditionGroup1;
    #   ConditionGroup2;
99  #   ...
    # };
101 #
    # ConditionGroup:
103 # LogicalOperation [Attribute] = {
    #   Expression1;
105 #   Expression2;
    #   ConditionGroup1;
107 #   ConditionGroup2;
    # };
109 #
    # LogicalOperations:
111 # "Any", "All", "NotAny", "NotAll"
    #
113 # ComparisonOperators:
    #   Equal, NotEqual, Like, NotLike, Match, NotMatch, Great, Less, GreatEqual,
    #   LessEqual
115 # SpecialOperations: Exists, NotExists
    #
117 # Expression:
    #   ComparisonOperator[:OperationModifier] = Pattern;
119 #   SpecialOperations;
    #
121 # The 'Match' and 'NotMatch' use TR1 Regular Expression standard and ECMA script
    grammar
    # The 'OperationModifier' is an option for the 'ComparisonOperation', specify 'i
    ,
123 # to make comparison case insensitive. The KVG expression treats value and
    # pattern as string by default, use the 'OperationModifier' to specify type or

```



```

transform
125 # both value and pattern before make comparision.
# 'i' -- insensitive case comparison
127 # 'b' -- convert decimal integer to bit string
# 'h' -- convert decimal integer to hexadecimal string
129 # 'B' -- convert hex string to bit string
# 't' -- convert file time integer to yyyyymmddhhmmss UTC time string
131 # 'I' -- comparison as 64 bit integer for arithmetic comparison operators
#
133 # Defines the condition to bypass password change notification
# BypassNotify "LogicalOperation" = {
135 #   ConditionGroup1;
#   ConditionGroup2;
137 #   ...
# };
139 # For example, we would like to bypass password both strength check and
# password change notification on this target for any account name starts
141 # with root or Admin or users in Administrators group or users in Managers
# group:
143 # Bypass "Any" = {
#   "Any" "_AccountName_" = {
145 #     Like = "root*";
#     Like = "Admin*";
147 #   };
#   "Any" "memberOf" = {
149 #     Equal = "Administrators";
#     Equal = "Managers";
151 #   };
# };
153
CheckStrengthFailIfIDPMReturn = { 100; };
155 CheckStrengthOnly = 0;

157 Bypass "Any" = { # Bypass strength check to HiPM
  "Any" "userAccountControl" = {
159     # Disabled accounts are Bypassing HiPM strength check.
    # Disabled accounts control numer is 2 (binary -> 10).
161
    # convert userAccountControl number from decimal to bit
    # string. Then use regular expression for comparison.
163     match:b = "([0-1])*10";
165
  };

167   "Any" "logonHours" = {
    Match:B = "([1])*";
169
  };

171   "Any" "pwdLastSet" = {
    match:t = "1290538([0-9])*";
173
  };

175   "Any" "lastLogon" = {
    Less:I = 128539593944756250;
177
  };

```

```

179     "Any" "_AccountName_" = {
180         # put the computer accounts below that are bypassing HiPM
181         Equal:i = "Guest";
182         Equal:i = "krbtgt";
183         Equal:i = "LethBridgeUser1";
184     };
185
186     "Any" "distinguishedName" = {
187         Like:i = "*OU=Calgary*";
188     };
189
190     "Any" "memberOf" = {
191         # The accounts have membership in the following groups are
192         # bypassing HiPM
193
194         Like:i = "CN=Administrators,CN=Builtin*";
195         Like:i = "CN=Domain Admins*";
196         Like:i = "CN=Enterprise Admins*";
197     };
198 };
199
200 BypassNotify "Any" = { # Bypass password synchronization to HiPM
201     "Any" "_AccountName_" = {
202         Equal = "LethbridgeUser2";
203     };
204 }; #End of BypassNotify
205 }; #End of 'End_Users' Target
206
207
208 Admin_Users = {
209     CheckStrengthFailIfIDPMReturn = { 100; };
210     CheckStrengthOnly = 0;
211     # If the target longid isn't the default sAMAccountName, define the longid as:
212     # LongID = "%sAMAccountName%";
213     # LongID = "DomainName\\%sAMAccountName%";
214     # LongID = "%distinguishedName%";
215     LongID = "LongIDMatchesPMTarget";
216
217     Bypass "Any" = { # Bypassing Strength Check to HiPM
218         "Any" "userAccountControl" = {
219             # Disabled accounts are Bypassing HiPM strength check
220             match:b = "([0-1])*10";
221         };
222
223         "Any" "_AccountName_" = {
224             # put the computer accounts below that are BYPASSING HiPM
225             Equal:i = "Guest";
226             Equal:i = "krbtgt";
227         };
228
229         "NotAny" "memberOf" = {
230             # The accounts have membership in the following groups are
231             # SENDING to HiPM

```

```

233         Like = "CN=Administrators,CN=Builtin*";
234         Like = "CN=Domain Admins*";
235         Like = "CN=Enterprise Admins*";
236     };
237 };
238
239 BypassNotify "Any" = {      # Bypassing Password Synchronization to HiPM
240     "Any" "memberOf" = {
241         Like = "CN=Domain Admins*";
242     };
243 };
244
245 }; #End of BypassNotify
246 }; #End of 'Admin_Users' Target
247
248
249
250
251 };
252 };

```

See also:

The following utilities are shipped with the Password Change Notification Module for testing and maintenance:

- [diagutil](#)
- [userattrs](#)
- [verifycfg](#)

Description

Use the **testidpm** program to test *Hitachi ID Bravura Pass* interceptors, API functions, and CGI programs such as **psa**, that send requests to the Password Manager service (idpm).

Usage

Run **testidpm** with the following arguments:

```
testidpm [-host <host>] [-port <port>] [-targetid <targetid>]
         [-account <account>] [-userid <userid>] [-pass <pass>]
         <-option> "<argument>" [<argument 2>] ...
```

Table 15.1: testidpm arguments

Argument	Description
-account <account>	The account/longID on <TARGETID>.
-block	Adds a blocking record and deletes all old requests in queue for the specified user.
-both	Used with “-strength” and “-reset”. Sends strength first, and if OK, sends reset.
-cgiresetqueued	Resets an account's password asynchronously.
-cgiresetsynch	Resets an account's password synchronously. This is a CGI service function. You must specify the account to reset. The caller uses shared memory, and gets a reply only after the connector run is finished. Calls made for an interceptor use the -reset argument.
-cgiunlocksynch	Unlocks a locked-out account synchronously.
-cgiverifysynch	Verify an account's password synchronously.
-checknchange	Checks the password against strength rules. If passed, it resets/synchronizes a user's password.
-finduser	Finds a <i>Hitachi ID Bravura Pass</i> user based on targetID and account.
-host <host>	Host where idpm is running.
-pass <pass>	Password for the user.
-port <port>	TCP port on which idpm is listening.
-pwrules	Displays the password policy, either default, or for a specified account.

... continued on next page

Table 15.1: testidpm arguments (Continued)

Argument	Description
-reset	Resets/synchronizes a user's password. This is for an interceptor operation, coming in through socket, and all accounts in the same target system group of the trigger account can be reset, depending on the target system group setting. Calls made for a CGI program use the -cgiresetsynch argument.
-sessionid <sessionid>	A GUID produced by the caller of Password Manager service (idpm) CGI functions, such as <code>PSS</code> , <code>PSK</code> or <code>IDA</code> , to identify a round of requests on one or more accounts. Requests with the same sessionid are considered to be in the same group for running GROUP_SUCCESS/FAILURE external interface triggers (exit traps).
-strength	Checks the password against strength rules.
-targetid <targetid>	Target ID for the specified account.
-threads <threads>	Number of threads to send concurrent requests.
-timeout <timeout>	Socket connection timeout value in seconds. Note: The default value is 60 seconds, however the Windows System default for connect timeout is 21 seconds maximum. If not specified, the timeout value defaults to 60 seconds.
-userid <userid>	UserID for the user.

Examples

1. To test the ability of Password Manager service to check a password against *Bravura Pass* strength rules (assuming you are on the *Bravura Pass* server):

```
testidpm.exe -host localhost -port 3333 -account acct1 -pass mypass1234 -targetid w2kserver -strength
```

2. To test resetting a user's password through Password Manager service:

```
testidpm.exe -host localhost -port 3333 -account acct1 -pass mypass1234 -targetid w2kserver -reset
```

3. To print the *Bravura Pass* password rules retrieved by `idpm`:

```
testidpm.exe -host MyPasswordManager -port 3334 -pwrules -timeout 90
```

4. To check if a *Bravura Pass* ID exists:

```
testidpm.exe -host localhost -port 3333 -finduser -user user1
```

5. To check if a user has an account on a target system:

```
testidpm.exe -host localhost -port 3333 -targetid win2kserver -finduser -user  
user1
```

Description

Use the **diagutil** program for troubleshooting the Hitachi ID Password Change Notification Module interceptor. Once executed, it submits real requests to the Password Manager service (idpm) to process according to the **intcptsvc** (p80) configuration file.

This program acts as a real time logging utility to monitor the activities between the interceptor and the Password Manager service (idpm) on the *Bravura Pass* server; for example, if an account password is changed, using `net user <accountName> <Password>` in another command prompt, the activities are displayed in the **diagutil** prompt until **[Ctrl]+[C]** is used to stop logging.

This program is installed by **intcpt.msi** or **intcpt-64.msi** on a Windows transparent password synchronization trigger system and can be found in the following directory:

<Program Files path>\Hitachi ID\Password Filter\util\

Usage

```
diagutil.exe [-l <debug level> ] [ -t <timeout> ]
              -u <userID> -p <userPassword>
```

Table 16.1: diagutil arguments

Argument	Description
-l, --level <N>	The debug level 1-6. The default is 4.
-t, --timeout <N>	The timeout in seconds to receive diagnostic information. The default is 60 seconds.
-u, --user <userID>	The user to be diagnosed.
-p, --password	The user's password.

Examples

For example:

```
diagutil.exe -u qa1000 -p letmein!
```

returns:

```
User account name: qa1000
Checking the password filter Dll system registry setting...passed
Checking the password filter Dll if it has been loaded...passed
Checking the password filter service if it is running...passed
Setting password for user [qa1000]:
2010-02-02 11:15:28.561.2903 - [] psintcpt.dll [388,2172] Info: Logging has been
enabled, Log level: 4
2010-02-02 11:15:28.576.5698 - [] psintcpt.dll [388,456] Info: User:[qa1000], entered
PasswordFilter, sequential number ( SNO ): 1
2010-02-02 11:15:29.675.4688 - [] psintcpt.dll [388,456] Info: User:[qa1000], SNO: 1,
PasswordFilter returned [1]
2010-02-02 11:15:29.717.0218 - [] psintcpt.dll [388,456] Info: User:[qa1000], entered
PasswordChangeNotify
2010-02-02 11:15:29.720.6418 - [] psintcpt.dll [388,456] Info: User:[qa1000],
PasswordChangeNotify finished
2010-02-02 11:15:29.724.6046 - [] diagutil.exe [3244,452] Info: ***** User:[qa1000]'s
password has been reset successfully *****
```

Note: If the debug level is greater than 4 (default), `diagutil` returns more detail on how the account falls into which categories.

Description

Use the **userattrs** program to query account attributes in Microsoft Active Directory; to find specific useful attributes that may be used as search criteria in designing the **intcptsvc** (p80) configuration file.

This program is installed by **intcpt.msi** or **intcpt-64.msi** on a Windows transparent password synchronization trigger system and can be found in the following directory:

<Program Files path>\Hitachi ID\Password Filter\util\

Usage

```
userattrs.exe [-p <ADsPath> ] [-a <attributes>] [-c <admin account> <admin password>]
               [-f <ADsSearchFilter>] -u <accounts>
```

Table 17.1: userattrs arguments

Argument	Description
-p, --adspath <ADsPath>	Specify the domain path (ADsPath). Available macros are %PDC%, %DC%, %DN%. The default is LDAP://%DC%/%DN%.
-a, --attributes <attributes>	Specify the attributes to query, separated by a space. If none are specified, all attributes are queried.
-c, --credential <admin account> <admin password>	Specify administrator account and password separated by space. If no credentials are specified, the security context of the current process is used to bind the object.
-p, --filter <filter>	Specify the account searching criteria (ADsSearchFilter). The default is sAMAccountName=%USER%.
-u, --users	Specify the users' accounts to query, separated by a space.

Examples

For example:

```
userattrs.exe -a DisplayName distinguishedName -u brownwi
```

returns:

```
Open an ADs object: LDAP://%DC%/%DN%...succeeded.  
Retrieving user [brownwi]'s attributes...  
Attribute: [DisplayName]:  
    Brown, William  
Attribute: [_AccountName]:  
    brownwi  
Attribute: [distinguishedName]:  
    CN=brownwi qa,CN=CertCo,DC=example,DC=local  
  
Cost : 437 (ms)
```

Description

Use the **verifcfg** program to verify that a given account will be bypassed by password strength checking and/or synchronization, according to per-target criteria set in the specified **intcptsvc** (p80) configuration file. Neither actual account strength checking, nor synchronization, is performed by this utility. This tool is useful for verifying whether the configuration file is designed properly before putting the interceptor online.

This program is installed by **intcpt.msi** or **intcpt-64.msi** on a Windows transparent password synchronization trigger system and can be found in the following directory:

<Program Files path>\Hitachi ID\Password Filter\util\

Usage

```
verifcfg.exe [-c <file>] [-l <level>] -u <user>
```

Table 18.1: verifcfg arguments

Argument	Description
-c, --cfg <cfg>	Specify the intcptsvc configuration file. The default is intcptsvc.cfg
-l, --level <N>	The debug level 1-6. The default is 4.
-u, --user	The user to be verified (required).

Examples

For example:

```
verifcfg.exe -c ..\service\intcptsvc.cfg -u qa1000 -l 4
```

returns:

```
Loading service configuration file:[..\service\intcptsvc.cfg]...succeeded.
Retrieve user:[qa1000]'s attributes and evaluate settings...
User: qa1000
Session has been created successfully
Target: End_Users -- None bypass          <== It means this account will be strength
       checked and synchronized against 'End_Users' target
Retrieve target: [End_Users], user: [qa1000]'s status...( FindUser )
```

Transparent Password Synchronization Configuration Guide

```
If return code belongs set [( 100 )] will be treated as strength check failed
Target: [End_Users], user: [qa1000]'s status = 200, message = userid=qa1000 username=
qa 1000
Returned [200] [userid=qa1000 username=qa 1000], strength check will be successful
<== The account was found in HiPM through idpm.
Target: Admin_Users -- Bypass both strength check and synchronization <== It means
this account will be NOT strength checked and synchronized against 'Admin_Users'
target
```

Note: If the debug level is greater than 4 (default), **verifycfg** returns more detail on how the account falls into which categories.

Appendices

CIDR notation

A

The purpose of the IP addresses field is to identify specific external server(s) that are allowed to access the *Hitachi ID Bravura Pass* server within the boundaries of the specified access rights. This is accomplished by using Classless Inter-Domain Routing (CIDR) notation to define an IP address block. The address block is represented by an IP address and a prefix size and is written in slash notation `<IP>/<prefix>` where:

- IP is the IPv4 (`<0-255>.<0-255>.<0-255>.<0-255>`) or IPv6 address (`<0000-ffff>:<0000-ffff>:<0000-ffff>:<0000-ffff>:<0000-ffff>:<0000-ffff>:<0000-ffff>:<0000-ffff>`)
- prefix is the prefix size for the mask and must be an integer within the range 0-32 for IPv4 or 0-128 for IPv6

The subnet mask determines the size of the address block (the number of IP addresses belonging to the block), and used in conjunction with the IP address, specifies which particular IP addresses belong to that block. Some planning is required. You need to determine whether the entry is going to be used to represent a single address or a range of consecutive addresses.

The size of the address block, or the number of IP addresses that the block contains is

$$\text{number of entries} = 2^{(32 - \text{subnetmask})}$$

To determine the IP addresses of the servers that will be granted access to the *Bravura Pass* server from the IP address/subnet entry do the following:

1. Turn the IP address into binary notation; for example,

$$128.10.12.1 = 10000000.00001010.00001100.00000001$$

2. Start at the left hand side of the binary representation and mark off the number of binary digits specified by the subnet mask; for example, using a 30 bit subnet mask, the first 30 digits are marked off (bolded).

$$\mathbf{10000000.00001010.00001100.00000001}$$

Valid or matching IP addresses must be based on these first 30 digits.

3. Calculate the valid addresses by creating all possible permutations using the remaining binary digits. In this case, there are only two unmasked digits, and as a result there are only $2^{(32-30)} = 2^2 = 4$ possible addresses that match.

$$\mathbf{10000000.00001010.00001100.00000000} = 128.10.12.0$$

$$\mathbf{10000000.00001010.00001100.00000001} = 128.10.12.1$$

10000000.00001010.00001100.00000010 = 128.10.12.2

10000000.00001010.00001100.00000011 = 128.10.12.3

To restrict access to a single server use a complete prefix (32 for IPv4 or 128 for IPv6). This means that the IP address must match exactly.

To restrict access only to processes (servers) that reside on the *Bravura Pass* server, use the local host address with a complete prefix (127.0.0.1/32 or ::1/128 for IPv4 and IPv6 respectively). External access will be denied.

File Locations

B

There are three main directories that are created when you install *Bravura Pass* instance:

- *<Program Files path>\Hitachi ID\IDM Suite\<instance>*
- *<Program Files path>\Hitachi ID\IDM Suite\Logs\<instance>*
- *<Program Files path>\Hitachi ID\IDM Suite\Locks*

When you install *Hitachi ID Connector Pack*, files are placed in different locations depending on the type of *Connector Pack*.

For an instance-specific connector pack, the installer, **connector-pack-x64.msi**, installs agent connectors in:

<Program Files path>\Hitachi ID\IDM Suite\<instance>\agent\.

For a global connector pack, the installer, **connector-pack-x64.msi**, installs connectors and supporting files in:

- *<Program Files path>\Hitachi ID\Connector Packs\global\agent\.*

See “File Locations” in the [Bravura Security Fabric Documentation](#) for more detail.

Index

A

ACF2
transparent password synchronization, 56
Active Directory
domain controller, 17
transparent password synchronization, 17–24

C

CIDR notation, 97
configuring
psunix.cfg, 38
transparent password synchronization registration, 63
conventions used in this document, 3

D

database tables
SESSLOG, 68
USER, 69
diagutil, 21, 22
documentation
conventions, 3
feedback, 4

E

Enable password synchronization, 9, 14, 63, 64
enabling
transparent password synchronization, 13

F

filtering password change requests, 24, 80

H

HP-UX, 33

I

ibmslapd.conf, 47
idapi, 78
idapisoap, 74
idpm, 24, 32, 68–73, 80, 88
access controls, 71
testing connectivity, 21
idpm service
testing, 87
install.sh, 38
installing
psintcpt.dll, 18
psldap*, 43
pspam.so, 31
pspasswd, 28
psunix.tar, 28
transparent password synchronization on Windows, 18
intcpt-64.msi, 80, 90, 92, 94
intcpt-x64.msi, 18
intcpt.msi, 18, 80, 90, 92, 94
intcptsvc, 17, 80, 90, 92, 94
configuring, 80
intcptsvc.cfg, 21, 24, 80
interceptor service
configuring, 21, 24, 80
IP addresses
rotating, 22
iPlanet, Unix
transparent password synchronization, 44

L

LDAP
plugins, 42
transparent password synchronization, 42–48
LDAP password filter plugin, 42
Linux, 36
login, 37

M

Mainframe Connector
 transparent password synchronization, 56–57
man, 37
 monitoring
 servers, 73
 transparent password synchronization, 72

N

nslookup, 22

O

OID-LDAP
 plugins, 49
 transparent password synchronization, 49–55
 OID-LDAP password strength filter plugin, 49
 openldap, 43
 OS/400 Server
 transparent password synchronization, 58–62

P

pam.conf
 editing, 33
 HP-UX, 33
 Linux, 36
 Solaris 8/9 without PME, 34
 Solaris 8/9 with PME, 35
passwd, 35
 password
 transparent password synchronization, 64
 Password Change Notification Module
 configuring, 80
 queue, 72
 timing out, 72
 password change requests
 filtering requests, 24, 80
 naming format, 21
 password synchronization
 registration, 64
 transparent, 5
 unregistering, 65
 plugins
 LDAP, 42
 OID-LDAP, 49
 PME (password management extensions), 33
 psintcpt
 timing out, 72
psintcpt.dll, 17
 extending timeout, 73

installing, 18

psldap, 47
 psldap*
 installing on Unix-based LDAP servers, 43
psldap*, 42, 43, 49
psldap.cfg, 51
 pspam.so
 installing, 31
 options, 36
pspam_pme.so, 35
pspasswd, 11, 25, 30, 31
pspwdexit_v5r4m0, 58, 59
pspwdexit_v7r1m0, 58, 59
 psr, 14
 configuring, 63
psunix, 27
psunix.cfg, 28, 38

R

RACF
 transparent password synchronization, 56
 Red Hat Directory Server, Unix
 transparent password synchronization, 44
 registry entries
 IDFilters, 24, 80
 replacing the password program, 28
 response file, 30
 rotating IP addresses, 22

S

self-service
 registering for transparent password synchronization, 64
 servers
 monitoring, 73
 services
 idapi, 78
 idapisoap, 74
 idpm, 68
 SESSLOG database table, 68
 setting longid naming format, 21
setup
 psunix, 27
slapd.conf, 46
 Solaris 8/9 without PME, 34
 Solaris 8/9 with PME, 35
 styles used in this document, 3
 sunldap, 43
 Sun ONE Directory Server, Unix

transparent password synchronization, [44](#)
 synchronizing
 passwords, [64](#)
syslog, [37](#)

T

technical support, [4](#)
testidpm, [87](#)
 TopSecret
 transparent password synchronization, [56](#)
 transparent password synchronization, [64–66](#)
 configuring multiple servers, [22](#)
 enabling, [13](#)
 event logs, [24](#)
 failing, [72](#)
 filtering password change requests, [24, 80](#)
 IBM Directory Server, [46](#)
 idpm service, [68](#)
 installing on Windows, [17](#)
 iPlanet, Unix, [44](#)
 LDAP, [42–48](#)
 load balancing, [15](#)
 logging, [24](#)
 Mainframe Connector, [56–57](#)
 managing, [72](#)
 monitoring, [72](#)
 OID-LDAP, [49–55](#)
 OpenLDAP, [46](#)
 OS/400 Server, [58–62](#)
 pushpass configuration file, [38](#)
 queue, [72](#)
 Red Hat Directory Server, Unix, [44](#)
 registering, [64](#)
 registration, [64](#)
 setting the longid naming format, [21](#)
 Sun ONE Directory Server, Unix, [44](#)
 unregistering, [65](#)
 Windows, [17–24](#)
 Windows trigger, [80](#)
 transparent password synchronization registration,
 [63–64](#)
 configuring, [63](#)
 options, [63](#)
 transparent synchronization
 HP-UX, [33](#)
 installing pspam.so, [31](#)
 Linux, [36](#)
 Solaris 8/9 without PME, [34](#)
 Solaris 8/9 with PME, [35](#)
 Unix, [25–41](#)

U

Unix
 transparent synchronization, [25–41](#)
 USER database table, [69](#)

W

Windows
 transparent password synchronization trigger,
 [80](#)
 Windows NT
 transparent password synchronization, [24](#)
 Windows password change requests
 filtering requests, [24, 80](#)
 naming format, [21](#)
 Windows servers
 transparent password synchronization, [17](#)

Transparent Password Synchronization Configuration Guide