

Hitachi ID Password Manager

Frequently Asked Questions
for Network Architects



Contents

1	How does Password Manager reset passwords?	1
2	How does Password Manager synchronize passwords?	1
3	What kind of database does Password Manager use?	2
4	What systems does Password Manager support?	3
5	On what platform does Password Manager run?	3
6	In what ways can Password Manager be customized?	4
7	How does Password Manager compare to the “password reset disk” in Windows XP and .NET?	6

1 How does Password Manager reset passwords?

Hitachi ID Password Manager resets passwords by signing into the target system with its own privileged password, looking up the relevant login account, setting the password attribute for that user and logging off from the target system.

At least one privileged ID/password is encrypted into the Password Manager database for each target system.

On systems that support it, Password Manager's own credentials can be given limited privileges – the right to list users, to search for users, to reset passwords and to set/clear flags such as intruder lockout.

Password Manager is web based. Client communication to the web server is HTTPS, while the server communicates with the managed systems directly using their various native protocols or via a Password Manager proxy server (128-bit AES encrypted TCP socket) or using a server-side agent (Unix, z/OS RSA Authentication Managers) with the same TCP socket encryption.

2 How does Password Manager synchronize passwords?

Since passwords are typically hashed on each system in a non-reversible, fashion and since different systems use incompatible password hashes, password synchronization must be an active process that takes place whenever users change their passwords.

There are really just two ways to synchronize passwords. Hitachi ID Password Manager supports both of the possible mechanisms for password synchronization:

- **Transparent synchronization:**

Password Manager can be configured to intercept native password changes on certain systems and:

- Apply a password policy beyond the one built into the system where a native password change first happened and potentially reject the initial password change
- Automatically synchronize the user's other passwords, on other systems, to the same value

Systems that can trigger password synchronization are Windows server or Active Directory (32-bit, 64-bit), Sun LDAP, IBM LDAP, Oracle Internet Directory, Unix (various), z/OS and iSeries (AS/400).

- **Web-based synchronization:**

Users authenticate to the Password Manager web GUI, using any browser, by keying in their NOS or directory ID and password. They can then set a single password on one or more of their own IDs on one or more systems.

3 What kind of database does Password Manager use?

Hitachi ID Password Manager must be configured with a SQL-based relational database. The Password Manager replicating data service can be configured to use any of the following SQL database engines as its physical data store:

- Oracle 10g, Enterprise Edition, R2.
- Oracle 11gR1, Enterprise Edition, so long as the 10gR2 client is used.
- Microsoft SQL Server 2005, Enterprise Edition.
- Microsoft SQL Server 2008, Enterprise Edition, so long as the SQL 2005 client is used.
- Oracle 10g, Express Edition, R2 (free download from <http://oracle.com/>).
- Microsoft SQL Server 2005, Express Edition, with Advanced Services (free download from <http://microsoft.com/>).

Password Manager maintains an identity cache in the database, which contains data about users, identity attributes and group memberships drawn from target systems nightly. This cache significantly improves the run-time performance of Password Manager, as it eliminates the need to repeatedly connect to target systems or to an external directory, to look up the same identity attributes again and again during the course of a workflow request or interactive user session.

The identity cache built into Password Manager:

- Is not an authoritative source of data – it is updated on a scheduled basis, generally nightly.
 - Stores data in a clearly documented SQL schema, available to 3rd party reporting programs.
 - Includes automatic data replication between multiple Password Manager servers. This supports both scalability and high availability.
-

4 What systems does Password Manager support?

Directories:	Servers:	Databases:
Any LDAP, AD, NDS, eDirectory, NIS/NIS+.	Windows 2000, 2003, 2008, Samba, Novell, SharePoint.	Oracle, Sybase, SQL Server, DB2/UDB, ODBC.
Unix:	Mainframes:	Midrange:
Linux, Solaris, AIX, HPUX, 24 more.	z/OS with RAC/F, ACF/2 or TopSecret.	iSeries (OS400), OpenVMS.
ERP:	Collaboration:	Tokens, Smart Cards:
JDE, Oracle eBiz, PeopleSoft, SAP R/3, Siebel, Business Objects.	Lotus Notes, Exchange, GroupWise, BlackBerry ES.	RSA SecurID, SafeWord, RADIUS, ActivIdentity, Schlumberger.
WebSSO:	Help Desk:	HDD Encryption:
CA Siteminder, IBM TAM, Oracle AM, RSA Access Manager.	BMC Remedy, BMC SDE, HP Service Manager, CA Unicenter, Assyst, HEAT, Altiris, etc.	McAfee, CheckPoint.

5 On what platform does Password Manager run?

Hitachi ID Password Manager must be installed on a Windows 2003, Windows 2008 or Windows 2008R2 server.

Installing on Windows 2003 or Windows 2008 allows Password Manager to leverage client software for most types of target systems, which is available only on the “Wintel” platform. In turn, this makes it possible for Password Manager to manage passwords and accounts on target systems without installing a server-side agent.

The Password Manager server must also be configured with a web server. Since the Password Manager application is implemented as CGI executables, any web server will work. The Password Manager installation program can detect and automatically configure IIS or Apache web servers, but other web servers can be configured manually.

Password Manager is a security application and should be locked down accordingly. Please refer to the Hitachi ID Systems document about hardening Password Manager servers to learn how to do this. In short, most of the native Windows services can and should be removed, leaving a very small attack surface, with exactly one inbound TCP/IP port (443):

1. IIS is not required (Apache is a reasonable substitute).
2. No ASP, JSP or PHP are used, so these engines should be disabled.
3. .NET is not required on the web UI and in most cases can be disabled on IIS.

4. No ODBC or DCOM are required inbound, so these services should at least be filtered.
 5. File sharing should be disabled.
 6. Remote registry services should be disabled.
 7. Inbound TCP/IP connections should be firewalled, allowing only port 443 and possibly terminal services (if required for some configuration tasks).
-

6 In what ways can Password Manager be customized?

The entire Hitachi ID Password Manager user interface is customizable and translatable. This includes graphical changes, text changes, layout changes, language translations, etc. No user interface elements are hard-coded into Password Manager.

User interface customization is simple to implement. Common elements, such as page layout and HTML preambles, are factored out into standard macros using an open source macro language (M4). Modifications made to M4 macros are propagated across the entire user interface.

Note that M4 (at least as it is used in Password Manager) is really just 3 keywords: `include`, `define` and `ifelse`. It is not something that administrators need to learn. Rather, the complexity is in the information architecture (which UI elements are defined where). To customize the Password Manager UI, all that is needed is an understanding of HTML and CSS, plus a bit of patience to find the right macro to edit – so that a change will propagate to the entire UI.

UI customizations are defined separately from the core UI, using a macro override scheme. This allows most customizations to survive Password Manager version upgrades with minimal intervention. For example, customers may define a new markup for HTML tables. This markup is placed in an override file and takes precedence over the default HTML table code. When Password Manager is upgraded, the customized markup will continue to take precedence over default HTML code.

In addition to modifying HTML and CSS code, customers can change the values of a number of system variables which alter Password Manager behavior. For example, password policy, intruder lockout frequency and duration, non-password authentication rules and more can all be adjusted from the Password Manager administrative web UI. System variables also survive version upgrades.

Password Manager behavioral modifications are made using plug-in points, rather than (as is common with many other applications) by modifying the source code of Password Manager itself.

Plug-ins are scripts or executables installed on the Password Manager server. Password Manager components call plug-in programs to make business policy decisions or to look-up information. Examples include:

- Look up a user's known, existing login accounts.
 - Helpful for integration with an existing meta directory.
 - Plug-ins are provided for LDAP directories and SQL databases.

- Look up a user's security questions.
 - Can be used to leverage existing authentication data.
 - Plug-ins are provided for LDAP and SQL implementations.
- Assign a new login ID to a newly created user.
 - A sample script is provided that implements popular ID schemes.
- Validate form inputs for workflow requests.
 - Is normally used to validate form inputs, such as checking that a new hire's home address has mutually-consistent city, state and area code fields.
 - Can also populate hidden fields (e.g., directory OU) and assign IDs (e.g., e-mail address) based on business policy.
- Assign appropriate authorizers to workflow requests.
 - May be based on the requester, recipient, entitlements or operations involved.
 - Global authorization logic is easier to manage than assigning static authorizers to every conceivable kind of request.
- Escalate from non-responsive authorizers to alternates.
 - A default implementation is provided, to escalate to the previous authorizer's manager.

This architecture, which encapsulates business logic into stand-alone scripts or executables, has two important benefits:

- It is significantly easier for organizations to adjust Password Manager behavior, since all such modifications are made in simple, self-contained files.
- Business logic implemented in this way survives Password Manager version upgrades, reducing the cost and delay associated with major upgrades.

Password Manager includes over 189 exit points.

Exit points may be triggered by many events, including:

- Attempts to sign into Password Manager (successful or failed).
- One user looking up the profile of another.
- Changes to a user's profile, such as creating a new account or changing attributes or group memberships for an existing account.
- Assigning a role to a user or removing a user from a role; changing Password Manager's configuration.
- Running a report.
- Triggering an intruder lockout.

Example uses of exit points include sending e-mails to users or administrators and creating, updating or closing incident records in an incident management application, notifying an IT infrastructure management system of an integration problem or recording a security event to a security incident event management (SIEM) or intrusion detection (IDS) system.

Various pre-built interface programs designed for use with exit points are included with Password Manager. They are generally scriptable and simplify the process of creating help desk incidents (e.g., BMC Remedy, HP Service Manager and the like) and sending e-mails.

For clarity, it should be noted that exit programs and plug-in programs in Password Manager are distinct components that serve different functions. Whereas plug-in programs are bidirectional – Password Manager sends data to the plug-in, the plug-in responds with data that alters Password Manager’s behavior – exit programs are uni-directional and are used strictly to pass information outbound from Password Manager to other applications. .

7 How does Password Manager compare to the “password reset disk” in Windows XP and .NET?

Starting with Windows XP, users can create a “password reset disk” whenever they change their passwords.

If a user forgets his login password, he can log into his workstation by typing his login ID but leaving the password field blank and instead inserting a previously-created password reset disk.

This feature is helpful for home users, but is significantly less useful than self-service password reset with Hitachi ID Password Manager:

- **Does not work for domain users:** The password reset disk feature does not work for domain passwords – only local ones.
- **Inconvenient:** Users must create a new disk whenever they change their passwords. In comparison, users register with Password Manager just once.
- **Inconvenient:** Mobile must carry the password reset disk with them. In comparison, users can access Password Manager from any computer, at any time.
- **Insecure:** Anyone who can touch the password reset disk can steal or copy it and subsequently log into the user’s account. There is no comparable vulnerability in Password Manager.