HITACHI
Inspire the Next

Bravura Security Fabric

Migration Reference Manual

- ☑ Hitachi ID Bravura Pass
- ☑ Hitachi ID Bravura Privilege
- ☑ Hitachi ID Bravura Identity
- ☑ Hitachi ID Bravura Group

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

Contents

About the Documentation 1

1.1 This document

This document shows you how to migrate files and data from an existing instance of *Hitachi ID Bravura Security Fabric* to another instance.

About Migrations About Migrations explains the use cases for a migration and when it might be required rather than a simpler upgrade. In general, it is recommended that you carry out an in place upgrade, where possible, because it is much simpler. Read the *Bravura Security Fabric* Upgrade Reference Manual (**`upgrading.pdf`**) to determine if this method is suitable for your situation.

Research and Analysis for Migration Research and Analysis for Migration describes important information that you need to gather, and analysis that is crucial before starting a migration.

Planning for Migration Planning for Migration provides an outline for planning the migration.

WARNING!: It is strongly recommended that you do not skip any of these chapters and read from the beginning.

All migrations can be complicated, especially where the solution includes integration with target systems that are harder to configure.

For *Bravura Privilege* and *Bravura Identity* migrations in particular, we strongly recommend taking advantage of our Professional Services, due to the catastrophic impact that a wrongly configured solution could have on an organization's environment.

Migrating Between Instances Migrating Between Instances describes the process of the most common cases of migrating between two unrelated instances.

Chapters idfilerep to updinst provide detailed usage information on utilities that are commonly used during migration.

This guide compliments the *Bravura Security Fabric* Documentation and the *Bravura Security Fabric* Reference Manual shipped with the latest version of *Bravura Security Fabric*.

Note: This document may be updated after a release, after migrations have been carried out. Check the Hitachi ID Systems portal or contact support@Hitachi-ID.com for the latest version.

1.2 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 *Hitachi ID Bravura Security Fabric*	*italics*
Button names, text fields, and menu items	**boldface**
Web pages (names)	***italics and boldface***
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

1.3 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 Security Fabric*, contact support@Hitachi-ID.com.

Part I

MIGRATION

About Migrations 2

This document uses the following terms:

Migrating copying configuration files and raw data from one instance to another.

Upgrading deploying a newer version of *Bravura Security Fabric* in place of an older version using **`setup`**. For information about the upgrade process, see the "Hitachi ID Suite Upgrade Guide" (**`upgrade.pdf`**).

This document is about the migration process for supported versions of *Hitachi ID Bravura Security Fabric*. To check the latest support status see:

`https://hitachi-id.com/support/support-for-older-releases.html`.

2.1 Migration use cases

Migration of data can be considered for the following use cases:

- Deployment migrations (p4) – when maintaining development, test and product servers
- Configuration migrations (p5) – when synchronizing instances of the same version
- Database migrations (p5) – when the backend database server is upgraded or moved, or after the database is restored from backup

2.1.1 Deployment migrations

Deployment migrations involve creating new instances with the same version and initial configuration as existing instances. This is typically done in the context of maintaining development, test and production *Bravura Security Fabric* servers. This is a recommended best practice.

The production server provides reliable services to users while new configurations are developed and tested.

The test server is a replica of the production server. The test server allows configuration changes to be tested without interrupting or affecting the operation of the production server. Once testing is completed, the tested configuration parameter changes need to be migrated to the production server.

The development server provides a facility to experiment with new features, targets, versions, and so on, without affecting the test or production servers. Migration from development to test is performed when the configuration changes have passed initial testing on the development server.

If you currently do not have a development or testing version of your production instance, it is highly recommended that you set up one prior to starting any migration.

2.1.2 Configuration migrations

Configuration migrations involve synchronizing configuration parameters between existing instances of the same version.

For example; when you want to have a development version to test user interface changes without affecting the main instance. After testing, you migrate those changes to production.

In some cases, more than one instance may be required on the primary server. When multiple instances are maintained, there are often configuration parameters that are common to all instances. When a configuration change to all instances is required, it is necessary to migrate the configuration parameter change from one instance to all other instances.

2.1.3 Database migrations

Database migrations are needed if the backend database server is upgraded or moved, or when the database is restored from backup after disaster recovery.

The majority of configuration parameters are stored in the database, and upgrades require consideration of differences in tables, schema, and data encoding. These considerations are described further in Databases and Migrating data.

2.2 When not to migrate

Before migrating any data, ensure you have taken the following into consideration:

- Should your data be migrated? (Do you fit one of the use cases above?)
- Is it worth retaining the data in the old instance?
- Are your back-ends compatible? (Is it possible to migrate your data from one environment to another?)
- Do you have custom code?

A clean install can avoid the problems of unwanted or obsolete data being transferred from the old instance, and prevent any unexpected results in the conversion process. Obvious configuration items like target systems, templates, roles, and so on are easily observable and can be duplicated in the new instance. Implementing a new version without carrying forward the old version's configuration provides a good opportunity to leverage new features and re-engineer components in a better way.

If the old instance contains unique data such as profile attributes that cannot be constructed from target system lists, security question data from user registrations, password reset history, or product usage statistics in the session log, then that can strengthen the argument for mass migration of the old instance. However, pinpoint migration of only the unique data is possible and should also be considered.

Research and Analysis for Migration

3

Before you start, gather information about your environment to ensure the feasibility of a migration and to help you plan the change.

Your research and analysis should include:

- An inventory of affected systems and installed *Hitachi ID Bravura Security Fabric* components
- An analysis of server requirements
- An analysis of *Bravura Security Fabric* configuration and files, including customizations

Any migration can be complicated, especially if it involves a major version upgrade at the same time or where the solution includes integration with target systems that are harder to configure.

For *Hitachi ID Bravura Privilege* and *Hitachi ID Bravura Identity* upgrades/migrations in particular, we strongly recommend taking advantage of our Professional Services, due to the severe impact that a wrongly configured solution could have on an organization's environment. *Bravura Privilege* holds the keys to the entire IT infrastructure.

An *Bravura Identity* migration requires understanding the entire IT infrastructure, all business logic implemented, an understanding of the solution implemented by the original Solution Architect, and of the issues present in the version of Python that the Python IDMLib library uses. Furthermore, *Bravura Identity* upgrades usually require refactoring of the solution, because the IDMLib and configuration infrastructure may change a lot from a minor version to the next.

As you can see in this document, there are several ways to migrate, and the correct path to take has to be considered based on the components of each individual environment on which the migration is performed.

Newer versions of *Bravura Security Fabric* include innovations to simplify maintenance of the system, to solve issues discovered in older solutions, and to adhere to advances in security, Web UI requirements and new standards, hardware and OS changes, scaling and performance. As a result, though we try to preserve backward compatibility, it is not always possible.

3.1 Taking inventory

Carry out a complete inventory of potentially affected systems to determine the location of all *Hitachi ID Bravura Security Fabric* components that need to be migrated, including:

- ☐ *Bravura Security Fabric* servers
  - ☐ Primary server
  - ☐ Replica servers
  - ☐ Proxy servers
- ☐ Target systems
  - ☐ Targets with listeners, such as, Unix, OS/390 mainframe
  - ☐ Targets with transparent password synchronization triggers, such as, Windows, LDAP Directory Service, Unix, OS/390, IBM OS/400
- ☐ Managed resources – Local service installations
- ☐ Systems that have *Bravura Security Fabric* software components installed on them. Examples include:
  - ☐ *Bravura Security Fabric* API installations
  - ☐ Local Kiosk software installed on user workstations and laptops
  - ☐ Domain Kiosk software installed on domain netlogon shares, called from domain policies
  - ☐ GINA or Credential Provider software installed on user workstations
  - ☐ Password expiry client or notification client software installed on domain netlogon shares
  - ☐ Password Manager Local Reset Extension installed on user workstations
  - ☐ Lotus Notes Extension DLL installed on user workstations
- ☐ Other technologies that support *Bravura Security Fabric*
  - ☐ Network load balancers deployed for high availability and/or redundancy
  - ☐ Reverse web proxy servers to expose the user interface to other networks
  - ☐ Backup servers for disaster recovery

3.2 Server considerations

Migration is sometimes required because the operating system or the hardware on which *Hitachi ID Bravura Security Fabric* is currently running is reaching or past the end of its life cycle. In some cases, older target system components and client components may work with newer versions of *Bravura Security Fabric*. During testing, you can confirm if this is possible in your environment; however, it is recommended that all components should be of the same version.

Hitachi ID Systems recommends you test at least one target of each type that you are planning to migrate. This will require installing all clients used in production. If you have scripted targets, test all scripts to make sure they work in the new environment. Ensure they work with the Python IDMLib library installed with the upgraded version.

The primary reason for this test is to discover and correct the issues that are unique to your *Bravura Security Fabric*.

After all functional testing is complete, you can leverage the existing test *Bravura Security Fabric* server to ease the deployment into production.

There are two complications:

- The dataset in the databases is constantly changing. Users are changing the data in the databases through normal use of *Bravura Security Fabric*.
- The target and client components in production are highly distributed and in constant use. It is highly recommended that the target and client components version match the version of the production *Bravura Security Fabric* server. Check the Connector Pack Integration Guide to determine requirements for each type of target.

These complications make a *Bravura Security Fabric* service outage necessary, to freeze the dataset in the databases. The outage also puts the new version of the *Bravura Security Fabric* server, target and client components, into production at the same time.

To minimize the length of the service outage, perform as much work as possible on the test server, in a test environment with test target and client components, and develop a clear step-by-step change control plan.

After successful testing, the service outage can start while the dataset from the databases is frozen and migrated and the target and client tools are deployed. This will require coordination with all target system and client system administrators.

In some cases, older target system components and client components may work with newer versions of *Bravura Security Fabric*. During testing, you can confirm if this is possible in your environment; however, it is recommended that all components should be of the same version.

Note: Verify that the new server has the same time and timezone configured as the old server. This will prevent artificial time gaps or overlaps in the timestamps recorded in the backend database and in the product logs. If possible, configure the same NTP server for all servers which make up the old and new instance, including application nodes, backend database servers and proxies.

3.3 Product considerations

Carefully analyze configuration parameters and files to determine what will be affected. Consider the following:

- ☐ Databases
- ☐ Replication configuration
- ☐ Logging during migration
- ☐ Scripts, plugins, and configuration files
- ☐ Product customizations and fixes
- ☐ The Windows registry
- ☐ Client tools
- ☐ Licensing
- ☐ Web interface modifications
- ☐ The web server configuration files
- ☐ Supporting systems
- ☐ Python
- ☐ Release notes
- ☐ Operating system updates
- ☐ Product password
- ☐ *Bravura Privilege*

3.3.1 Databases

Each significant version of Hitachi ID Systems software is likely to have different requirements for its database tables, table schema or data encoding.

Hitachi ID Bravura Security Fabric works with any of the following database management systems:

- Microsoft SQL Server 2019
- Microsoft SQL Server 2016 SP2
- Microsoft SQL Server 2014 SP3

Both 32-bit and 64-bit versions of these databases will work.

Note: The **Compatibility level** on the Microsoft SQL Server database must be set to a minimum value of **SQL Server 2012 (110)**.

Note: If you are installing SQL Server Reporting Service (SSRS) to use the *Analytics* app, ensure the server is not a Domain Controller.

Express editions should *only* be used for evaluation purposes. Hitachi ID Systems strongly recommends that, whenever possible, you use an enterprise or standard edition, rather than the express database edition.

Database I/O performance on a virtualized filesystem (e.g., VMDK or equivalent) is slow. If the database server software runs on a VM, please use a fast, nearby NAS or SAN to store the actual data files.

WARNING!: Clustered backend databases can lose data during or after cluster failover. Hitachi ID Systems recommends using *Bravura Security Fabric*'s application-level replication rather than clustered databases whenever possible. If your company policy requires the use of clustered databases, have database cluster nodes available as close as possible on the network to the *Bravura Security Fabric* nodes to target directly. See Installing with a shared schema for setting up the *Bravura Security Fabric* nodes in shared schema.

Data migrations from Standard to Enterprise editions of MSSQL will require adding partitions to some of the database tables. Contact support@Hitachi-ID.com if you are in this situation.

Migrating to Datacenter or Developer editions of MSSQL, as well as to database clusters or other database-replicated configurations that lock down the database schema are not supported by default. Please contact your account manager if you would like our Professional Services to do the migration for you.

3.3.2 Replication configuration

The easiest way to migrate an entire cluster is to migrate the primary node, then install new replication node/proxy servers in the new environment and resynchronize from the primary. See Adding a node details in the *Replication and Recovery (**replication.pdf**)* to learn how to add a replication node into a cluster.

This procedure only synchronizes data in the backend database. Any component prerequisites (such as IDMLib or target system requirements) must still be installed on each node before resynchronizing. After resynchronization, you need to run **updinst** (p62) to synchronize files and registry settings.

Migrating just the primary node may result in some data loss from the nodes that are not migrated. Alternatively, you could migrate all nodes to ensure data preservation, but this may result in some synchronization problems.

3.3.3 Logging during migration

You should leave the logging service running, on the server on which you are running migration utilities, in order to capture the events generated during the data migration process.

To capture event run-time errors which are not captured by the logging service, redirect output to a text file. For example, to log the output of **fixdb** run:

```
fixdb > fixdb.txt 2>&1
```

If you get pop-up errors, copy the text inside and add it to the text file, or take a screenshot. If you get any errors (other than uppercase failures from fixdb), stop the process and send the text files to the support representative you have contacted for your migration.

3.3.4 Scripts, plugins, and configuration files

When configuring Hitachi ID software, various files may have been added or modified in order to implement various features or customizations.

Carefully analyze and test all scripts, plugins, and configuration files; for example:

- Customized auto discovery scripts
- Target integration configuration files
- External interface program configuration files
- Plugin scripts
- Language and user interface modifications
- Customized password strength dictionary file
- Filters
- Access controls, including user classes
- Service configuration

Note the following for custom scripts, plugins, and configuration files:

- The directory structure may have changed between versions; verify any hard-coded directory PATH details. See File Locations for current file locations.
- The names of *Hitachi ID Bravura Security Fabric* programs may have changed; verify all references to those programs.
- The command line usage of *Bravura Security Fabric* programs may have changed; verify all arguments being passed into those programs.
- Scripts written in the PSLANG language are being migrated to Python scripts. You may want to examine any scripts to determine if new build-in functions or versions written in Python are available to optimize your code.
- The capability of connectors and external interface programs may have been enhanced (possibly by changes to the target address line or additional options in its configuration file); you may need to review your integrations to reflect those changes.

- New built-in connectors may have been written for integrations you previously scripted using flexible connectors; you may want to re-implement those targets using the new connector.
- Multiple connectors may have been consolidated into a single connector (for example, the Domino and Lotus Notes connectors were combined in connector pack 1.3); you will need to redo any integrations using either of those connectors.
- If you have your own compiled binaries for custom interfaces, plugins or authentication chains, you must recompile them as 64-bit.
- Changes to wfreq plugins as of 12.0.0 mean almost any custom wfreq extensions likely need to be re-written or have logic changed to meet new restrictions on when the plugin runs.
- Custom loaddb or idtrack scripts written prior to 12.0.0 should be tested. Care has been taken to maintain backwards compatibility but some edge cases may not work or require refactoring; for example, differential listing works differently to older versions.

3.3.5 Reports

Reports in *Hitachi ID Bravura Security Fabric* version earlier than 9.0 were saved as flat files. In 9.0 or later they are SQLite databases. Saved reports will not be preserved when upgrading from earlier versions. After the upgrade, scheduled reports can be viewed on the **Manage the system** → **Maintenance** → **Scheduled jobs** page, but cannot be run or modified.

3.3.6 Product customizations and fixes

In addition to configuration files, it is possible that your *Hitachi ID Bravura Security Fabric* instance may contain custom binaries and/or schema. These could include web modules, connectors, plugin programs, external interface programs, and so on.

Identify custom binaries by right clicking the binary in Windows Explorer and check the versions tab. Alternatively you can check it using the **getfileinfo** program (p30).

Once custom binaries are identified, the purpose for them should be determined in order to decide whether they are still necessary. For example, if the custom binary was created to resolve a product deficit, it is likely the deficit was resolved in the base product. Similarly, if the purpose for the binary was to add custom functionality, it is also possible that feature was added to the base product.

Read the release notes to determine whether a custom binary is necessary, or list all the binary versions by running the following command from each of the folders below:

```
for
 %i in (*.exe) do (%~ni -v>>versions.txt & echo %~ni>>versions.txt)
```

- <*Program Files path*>\Hitachi ID\IDM Suite\<*instance*>\agent
- <*Program Files path*>\Hitachi ID\IDM Suite\<*instance*>\cgi-bin
- <*Program Files path*>\Hitachi ID\IDM Suite\<*instance*>\interface

- <*Program Files path*>\Hitachi ID\IDM Suite\<*instance*>\lib
- <*Program Files path*>\Hitachi ID\IDM Suite\<*instance*>\report
- <*Program Files path*>\Hitachi ID\IDM Suite\<*instance*>\service
- <*Program Files path*>\Hitachi ID\IDM Suite\<*instance*>\util

If it is still not obvious, contact support@Hitachi-ID.com for assistance.

Only in very specific cases will an older binary be usable in a newer product. A previous service engagement or product fixes on an existing instance (including binary and/or schema) might require assistance from Support to either initiate a new service engagement, verify the fixes provided in a previous release, or indicate new feature/functionality that supersedes fixes in older functionality.

3.3.7 The Windows registry

A careful examination of the registry in both the source and destination instance subkeys will reveal manually entered subkeys.

Like custom binaries, each subkey identified as manually entered should be evaluated on a case by case basis before choosing to export it to another instance. This is especially true of version migrations. See the release notes to determine whether the key needs to be migrated or whether product enhancements have been made to toggle these keys from the administration interface. Contact support@Hitachi-ID.com for details about any specific registry subkeys.

If a key needs to be migrated, you can make the change directly using the Windows registry editor (regedit.exe) utility on the new instance, or by exporting from the old and importing to the new server using the same utility.

Note: One key of vital importance in older versions of the product is the "MASTERKEY". This key was used for communication between various components of the software. For example, transparent password synchronization triggers use it to communicate with the **`pushpass`** service, and some connectors use it to communicate with their associated agent installed on the target system. In new product versions, this key is now known as the "CommKey" (short for communications key.) This key *cannot* be copied and renamed as it is now stored in an encrypted format. To set the key, use the **`resetkey`** program.

3.3.8 Client tools

As of 9.0, *Bravura Security Fabric* is completely 64-bit, so it requires any target tools (for example database clients, SDKs or any software our product loads directly in order to integrate with targets), to also be installed as 64-bit versions. This means that for fresh installs (including replication migrations), the 64-bit clients must be installed, and for cloned migration, the 32-bit clients must be uninstalled, and the 64-bit client installed. Refer to the Connector Pack Integration Guide for details on any targets which require the use of target clients or tools.

3.3.9 Licensing

When migrating instances, you must use an existing license (which is installed by default with the product). When you migrate using a fresh installation (rather than replication or cloning), the license will be the same as when you originally installed *Hitachi ID Bravura Security Fabric*. If you have updated your license since first installing *Bravura Security Fabric*, ensure that you manually copy the most recent license.

3.3.10 Web interface modifications

All custom web interface modifications should be reviewed. Some existing modifications will require modification or deletion, while new modifications may need to be added.

The best method to address this issue is to go through every file in the custom directory in the source instance and confirm the same file exists in the destination instance src directory. If a file does not exist in the new instance, the custom version of the file can probably be deleted. If a file does exist, go through the entire file and search for each tag or KVGroup in the new instance. If the tag or KVGroup does not exist, it can probably be deleted from the file. If it does exist, compare the custom version to the default version in the new instance and figure out what if any modification will need to be made to the custom version.

Finally, test all the web pages and verify that the desired modifications have been migrated properly. Also look for new web interface constructs that may require new modifications.

See the *Bravura Security Fabric* Documentation for more information.

3.3.11 The web server configuration files

The *Bravura Security Fabric* server must have a running web server. Microsoft Internet Information Services (IIS) is supported for automatic configuration by *Bravura Security Fabric*'s installer.

IIS URL Rewrite module is also required. If it was not installed as part of your IIS web server install, install it from `http://www.iis.net/downloads/microsoft/url-rewrite`.

By default, the **`setup`** program of your new instance would have configured the web server so that the necessary virtual directories are created, the CGIs are all registered, and whether clients navigating to the base URL will get redirected to a specific instance. If you have customized the behavior of the web server on your old instance in some way, you will have to manually do the same thing to the new instance.

3.3.12 Supporting systems

As mentioned in Taking inventory, there are a number of potentially related supporting systems to consider in addition to the *Bravura Security Fabric* servers themselves. These systems fall into two categories; systems with *Bravura Security Fabric* software components installed on them, and other technologies that support the *Bravura Security Fabric* server.

In particular, each local service will require redeployment for workstations to allow reintegrating to the new instance.

3.3.13 Python

The latest *Bravura Security Fabric* requires Python 3.7.3+ 64-bit. Python must be installed for all users.

The upgrade in support to Python 3.7 as of *Bravura Security Fabric* 12.0.0 may cause issues with some scripts.

3.3.14 Release notes

Review the release notes to identify changes that might affect expected product behavior.

Pay particular attention to:

- Script changes for plugins and exit traps.
- Access controls.
- Features that have been added or revoked.

3.3.15 Operating system updates

Verify there are no pending Windows updates to be installed, and verify that no server restarts are scheduled before starting the upgrade or patching process.

3.3.16 Product password

Ensure that you know the *Bravura Security Fabric* service user (psadmin) password. This will be required to upgrade existing systems.

For information resetting the service user password, with **`serviceacct`**, see the Reference Manual.

3.3.17 *Bravura Privilege*

Review the following:

- Managed system policies. Ensure that no single account belongs to more than one policy.

  WARNING!: As of version 12.0.0, *Bravura Privilege* does not allow for a managed account to belong to more than one policy. If an account is a member of multiple policies at the time of the upgrade, it will be removed from all policies, except from the managed system's primary policy.

- Session monitoring privileges, for both managed system policys, and self-service rules.

WARNING!: Session monitoring privileges have changed as of 12.0.0. All managed system policy and self-service rules will be cleared upon upgrade. You will need to reconfigure them after the upgrade.

- *Hitachi ID Bravura Pattern: Privileged Access Edition* trustee privileges. Trustee privileges changed in 12.0.0. The upgrade script should maintain any existing rules so they work how they used, but authmod rules should be reviewed in case different behavior is desired.

Planning for Migration 4

Use the documented information you gathered in Research and Analysis for Migration to develop:

- ☐ A *test plan* that can measure whether or not the old and new instances behave as expected
- ☐ A *backup and recovery plan*
- ☐ A *change control plan* to minimize downtime of the production system
- ☐ A *communications plan* to prevent calls to the help desk during the process

4.1 Test plan

Write a test plan that systematically identifies use cases covering the required capability of the system. This test plan should cover all automated processing expectations, all end-user self-service cases, all help desk assisting user cases, and so on.

Test the plan on the current production system to ensure it is operating as expected.

The Hitachi ID Systems suite of products has the potential to interact with nearly every other piece of computing equipment in the enterprise, and as such, the test plan ought to be comprehensive and well maintained.

After a migration, systematically test the use cases on the newly upgraded system.

Develop test cases around all the various components of the system that you have implemented; for example:

- ☐ End-user use of the self-service web interface, and help-desk-user use of the web interface to assist others to perform all possible operations, such as:
  - ☐ Identification of the profile during login
  - ☐ Authentication of the profile using each method
  - ☐ Resetting passwords for yourself and others
  - ☐ Unlocking accounts for yourself and others
  - ☐ Claiming unassociated accounts to your profile
  - ☐ Managing tokens, SmartCard, and/or hard disc encryption keys
  - ☐ Requesting access to a privileged account

    - ☐ Approving a request for access to a privileged account
    - ☐ Accessing a privileged account
    - ☐ Checking in access to a privileged account
    - ☐ Randomizing a password
    - ☐ Randomizing a local service
    - ☐ Assigning access and membership via user classes
    - ☐ Requesting a new account on a target system
    - ☐ Requesting attribute changes
- ☐ *Phone Password Manager* functionality
- ☐ Telephone interface for performing operations
- ☐ Target system integrations

  There should be a test for each operation that is possible on each target system.
- ☐ Ticket / issue tracking system integrations

  There should be a test for each operation that is possible on each ticketing or issue tracking system.
- ☐ Technologies deployed on user workstations, such as:
    - ☐ Self Service, Anywhere (SSA) / *Login Assistant* (formally Credential Provider / GINA) interface for domain attached users
    - ☐ Other local kiosk solutions for remote and/or local users
    - ☐ Lotus Notes ID file delivery mechanism
    - ☐ Cached credential controls for external users
    - ☐ *Login Manager* client software
- ☐ Technologies deployed on other servers
    - ☐ Transparent synchronization triggers
    - ☐ Target system agent listeners
    - ☐ Hitachi ID Systems proxy servers that run connectors for targets
    - ☐ Notification service clients that run in user domain logon scripts
    - ☐ Reverse proxy servers that allow the web interface to be reached from external networks
- ☐ High availability technologies
    - ☐ Load balancers that direct traffic to multiple Hitachi ID Systems servers either based on load or simple round robin
- ☐ Redundancy technologies
    - ☐ Hitachi ID Systems replication servers
    - ☐ Third party systems that make periodic backups of the files and registry on the servers
    - ☐ Third party systems that make backups of the databases that the Hitachi ID Systems servers use
- ☐ Automation and scheduled events

- ☐ Nightly scheduled update
- ☐ Automatically scheduled tasks such as auto discovery and log rotation.
- ☐ Automated report generation and delivery
- ☐ Notification of soon-to-expire and other bulk email events
- ☐ Notifications delivered via plugin points or "exit traps" which trigger when certain conditions are met or actions are performed.

Migrate the instance data (both configuration and user data) from production into test. As you do this, document all the steps. This information will be useful later when you are building the production change control plan.

After the migration, systematically test the cases on the newly migrated test environment. Where a test case has problems, make the necessary changes to the system. Document the changes that were made to make a case work in the new system.

A comprehensive test plan could potentially take a very long time to run. Breaking it down into multiple smaller test plans for different purposes may have benefits; for example, you could use a comprehensive test plan while building a complete working system in the lab, then use a subset of the full plan, that tests only critical use cases, during the implementation of the production change control plan to minimize service disruption.

4.2 Production change control plan

Consider the following points to help reduce the production service outage:

- Since production change control windows are usually small, and are typically scheduled during off-hours, extra attention to detail is required to compensate for the late hours.and disruption of routine. Upgrade plans should should be both explicitly documented and easy to follow. For each step:
  - Describe exactly what to do in clear, plain language, so the person implementing that step does not have to take time to analyze the wording.
  - Include a quick test to verify that this step was completed.
  - Include contingency procedures describing how to troubleshoot failure, back out the change, or redo the change.
  - Describe conditions to help the team decide if the entire migration has reached a critical impasse and must be backed out.
- If you are performing a version migration, upgrade all components of the software to the same version.

  You can test 12.2.4 with older components such as transparent synchronization triggers, notification client, and so on, to determine whether it is backwards compatible with them. If it is, then you can choose to postpone the upgrade of those components to another time, possibly in another change control situation.
- You do not *need* to stop the old production system until you copy data from it, but it is recommended that you limit or stop access and usage of the old instance during migration.

Depending on the steps that follow data export, you might want to immediately start it up again to handle certain transactions while the new production system is being finalized. Transactions on the old server, between the time you start it up again until the new system fully takes over, will not be known to the new system; therefore you need to carefully consider what transactions are necessary.

For example, you might decide that it is more important for password strength checking to be enforced by transparent synchronization triggers than it is for users to have access to the web interface. In this case, the migration plan should indicate stopping all services on the old server (include the web server) before exporting its data, then once the export is complete, *only* starting the **`pushpass`** service so that transparent synchronization is operational, but the web interface is not.

- If you are performing a production change control on a Hitachi ID Systems product instance that leverages authorization workflow, do *not* let the old server handle transactions between the time that the data was exported from the old server and the new server comes online.

  If the new server is not aware that the old server handled workflow-based transactions, it may attempt to run those transactions again, which could result in failures, or it might ask authorizers to perform actions that they already performed.

- If your production change control plan requires that the web interface is *not* used, then consider solutions other than simply stopping it outright; for example, redirect the web traffic to a static page which explains that the outage was planned, when the site will be available again, and gives advice on what alternatives are available to them. Remember to add a task to the plan to eliminate the redirection when the new server is ready for use.

4.3 Communications plan

If the migration you are performing is going to affect the user interface in a dramatic way, or if it is going to alter functionality significantly, consider informing users well in advance of pushing changes into production, so that they do not swamp the help desk with questions once the change is in production.

> **Note:** Due to time format changes from Unix time to ISO date strings, there may be time skews in reports and active access requests.

Migrating Between Instances

5

This chapter explains migrations between two unrelated instances. The most common case for doing this is synchronizing changes between test and production environments in order to update or upgrade the production version, or to add or change the capability in an existing deployment.

By validating changes on a test environment prior to pushing changes to production, you can reduce the chances that a change will negatively impact the production environment.

Ideally the test environment is identical to the production environment in all ways; the same hardware characteristics, same operating system, same installed software, and the same networking architecture. If it is not possible to have identical environments, be certain that you are not including test expectations that are unrealistic; for example, load testing on a test environment which uses weak hardware and no replica servers can not be compared to a production environment using powerful hardware and multiple replicas.

Maintaining a test environment requires moving data in both directions between the test and production environments. In simple cases, synchronizing environments could require copying files and exporting the registry from one environment to another. In more complex cases where data in databases must be synchronized, the process is complex enough to require some server downtime. The typical process for pushing a change into production might go like this:

1. Migrate the current production system into a test environment.
2. Implement the changes in the test environment and then test everything that might be affected by the change.
3. Migrate the changes into production.

CAUTION: Avoid a situation where the same systems are targeted by two instances, which will split the history and account data. This may lead to password expiry notices being sent from one instance even after on the other instance the password was already reset. Account data (like profile attributes) will exist on one instance, not on the other. Passwords on managed systems will be reset by both instances, resulting in race conditions and each instance losing control over the managed systems that the other instance has randomized. One way to work around this is to target a different set of accounts from the test instance; for example, create a test OU on an Active Directory system and target that.

Many migrations can be complicated, involving complex components that require an in-depth knowledge of *Hitachi ID Bravura Security Fabric*. It is highly recommended you contact support@Hitachi-ID.com for assistance.

5.1 Migrating files and registry

Generally, migrating files and registry between instances is simple. In the case of files, you are simply copying them from one system to another. In the case of the registry, you are exporting from one server and importing on the other. There are a few things to keep in mind:

- If the hardware, operating system, file system, or installed software (which includes the Hitachi ID Systems software and web server) is configured differently between the instances, then be certain the content of the files or registry export is valid for the server you are copying it to. This can mean checking hard coded drive letter, file path, and file name in your files and registry imports.
- The regularly scheduled file replication task only pushes new or changed files from the primary server to the replicas and proxies. As such, when copying files, they *must* at least be copied to the primary server. Files can optionally be copied to the replica servers as well if you don't want to wait for, or force, the scheduled task.
- As with the handling of files, the registry replication also only pushes changes from the primary to the other servers. As such, all registry changes must at least be made to the primary server and optionally to the replicas and proxies.
- If the file or registry entry being changed or added is somehow related to a running service, that service likely needs to be restarted in order to recognize the change.

For more details on the capability and limits of the file and registry replication service (such as maintaining white lists and black lists that apply limits on what should and should not be replicated) see:

- idfilerep for details on **`idfilerep`**.
- updinst for details on **`updinst`**.

5.2 Migrating data

There are some things to keep in mind when migrating data:

- When exporting data, Hitachi ID Systems recommends that you export from the primary server (the server running auto discovery). If the primary server is not recoverable, use the server that has been in replication the longest.
- When exporting data, be certain that the entire system is not in active use, to export referentially intact data. This usually means arranging for a service outage for all instances so you can stop the necessary services (including the web server). Ensure that you disable the health check task from Windows Task Scheduler.
- Similarly, when bulk importing data, be certain that the system is not in active use, to ensure that you import referentially intact data.
- If you bulk import data to a Hitachi ID Systems server, the database service will not be doing the work, and as a result, the changes will not be sent to replica servers. Either you must import identically to each server, or you must rebuild the replica databases manually.
- Migration of configuration or user data between two relational database versions requires the use of the **`importdata`** utility and an associated configuration file that defines the behavior and identifies the limits.
- The database user on the new database must have read (select) access to the old database.
- The **`smonmove`** program is used to change the location of session monitoring data in the database from one node to another. See smonmove in the *Reference Manual* for more information about this utility.

5.3 Using idmsetup.inf to install *Bravura Security Fabric*

When you install *Hitachi ID Bravura Security Fabric* on the main server, an **`idmsetup.inf`** file is created in the \<*instance*>\psconfig\ directory. You can use the file to aid in the installation of proxy servers, backup servers, and add-on software. It is a recommended best practice to use this file during migration projects. It acts as an "answer file" for the installer, by populating instance specific configuration parameters and encryption keys at every step during the setup process.

It is highly recommended that you:

- Copy the **`idmsetup.inf`** file to each new server and place it in the same directory from which you will run the installer (**`idm.msi`**).
- Ensure the architecture of the new server matches the architecture of the primary server.

See Installing *Hitachi ID Bravura Security Fabric* using idmsetup.inf in the *Replication and Recovery (**`replication.pdf`**)* for an example of using **`idmsetup.inf`** to install a replication node.

Migrating Component Configuration and Data 6

You can migrate component configuration and data using the following methods:

6.1 Replacing an instance's configuration using environment files

Solution architects can work with clients to create environment files that store specific component information such as target configuration, team setup, authentication setup and workflow. Using these files saves time needed to reconfigure the components again when migrating between environments. For example:

1. A client sets up a *Hitachi ID Bravura Security Fabric* test environment with all of the components and settings required.
2. The client conducts testing and makes changes where required.
3. When the test environment is complete, the solution architect can help create environment files that will capture the existing component settings.
4. *Bravura Security Fabric* is installed in the production environment.
5. The environment files from the test environment are copied to production.

   Some settings in these files will need to be adjusted for the new environment, such as target addresses and credentials.
6. Each component is then installed and during the installation, the components will use the environment files and preconfigure the environment in the same way it was in test.

Environment files can also be loaded at a later date, after a component is installed; for example, a client can test changes to a component, then when they are ready for production, copy and load the file in the production environment.

Contact support@Hitachi-ID.com for assistance.

6.2 Migrating using export_data_components.py

The **export_data_components.py** script is used to export product configurations as components and an evironment file. The resulting data components and an environment file can be applied to a different instance.

This section shows you how to export current configurations into components and environment file using the **export_data_components.py** script.

WARNING!: Consult with support@Hitachi-ID.com before using this script.

6.2.1 Setting up the configuration export command-line program

The configuration export program (**export_data_components.py**) is a Python executable script located in the \<*instance*>\script\ directory. In order to run it from the command line, you need to configure a number of environment variables. To do this:

1. Launch a command prompt as an Administrator and navigate to the \<*instance*>\script\ directory.
2. Run the command:

   ```
   ..\instance.bat
   ```

   You should now be able to run the configuration export program. Ensure that you always run it as an Administrator.

6.2.2 Export configurations

The export option is used export all the configurations since the installation of the product. It will generate an environment file that contains all the changes that have occurred in the installed components. It will also generated data components for all other configurations.

```
export_data_components.py export
```

They can also be specified in the output option.

```
export_data_components.py --output env export
```

6.2.3 Export configurations to components

The export option can also be used to export a complete set of components which includes installed components, with updated configuration settings, and components for all other configurations.

To create data components of the configuration changes, specify the output option to **component**.

```
export_data_components.py --output component export
```


6.2.4 Optionally export to a specific directory

A specific location can be specified for the script to export to. The default is the instance directory. The following will export to the specified folder. If the folder does not exist it will create it.

```
export_data_components.py --dir c:\Temp export
```

6.2.5 Additional export options

The script can filter the results using the following options.

Option	Meaning
--audit	Try to calculate changes from audit table.
--comparison_set	A json configuration to compare product configuration against. You can generate complete comparison sets using product_json mode.
--ignore_filesystem	Ignore existing components in component\Default and component\Custom.
--type	The idmconfig Type of the desired object. If none given, all types will be searched.
--field	Key/Value pair of form: key=value. Specify a single key/value pair to match against. Can be specified multiple times.
--user or **--not_user**	Specify audit user(s) to search or filter out configs for.
--module or **--not_module**	Specify audit module(s) to search or filter out configs for.
--start_time	A start time to search for configs from. Format is SQL format YYYY-MM-DD hh:mm:ss
--end_time	A end time to search for configs to. Format is SQL format YYYY-MM-DD hh:mm:ss

6.2.6 Encrypted fields

The script will not decrypt encrypted fields in the component configurations. When moving components to a new environment, the script can be used to ensure the encryption is valid. If the script determines that the encrypted fields are not valid it will give the user an opportunity to update the fields.

In order to use this option, the new components will be copied into the Custom directory of the new environment. To validate the encrypted fields, specify the **check_encrypted** option.


```
export_data_components.py check_encypted
```

6.2.7 Product configuration dump

The script can output a complete product configuration into a single json file. This json configuration can be used when exporting with the **`comparison_set`** option.

```
export_data_components.py product_json
```


Part II

TOOLS

getfileinfo 7

Description

Use the **getfileinfo** program to return the build information for Hitachi ID-created binary files. This information is useful during support calls, or when diagnosing problems.

The program returns information about whether the file has been customized or upgraded, or contains debugging information.

Usage

```
getfileinfo.exe <binary file name>
```

Examples

The following is an example of the return for a regular build:

```
FileName:              <file path>\<filename>
MajorVersion:          4
MinorVersion:          0
BuildNumber:           1
RevisionNumber (QFE): 6552
FileFlags: 0x0
FileVersion:           4.0.1.06552
SpecialBuild:           not found
```

The following is an example of the return for a custom build:

```
FileName:               <filepath>\<filename>
File has been customized.
MajorVersion:          6
MinorVersion:          3
BuildNumber:           0
RevisionNumber (QFE): 2066
FileFlags: 0x20
FileVersion:           6.3.0.02121
SpecialBuild:          Custom build by <developer's name>
```

idfilerep 8

Description

The File Replication Service (idfilerep) receives data from a master instance in a replication environment, and is used in conjunction with the **updinst** utility to synchronize files and registry keys between multiple instances.

When **updinst** is used from the master instance, the File Replication Service (idfilerep) adds, modifies, and removes files and registry settings on the server. Configuration options for file replication are explained further in this section.

> **CAUTION:** Do *not* attempt to replace Database Service files using **updinst** or the File Replication Service. Updating the Database Service and related files (such as **iddbmssql.dll**) must be done manually on all instances. This only applies to the Database Service service. All other services can be updated using the File Replication Service. To update the Database Service files manually, shut down all services on the instance, back up all services, and then replace the Database Service files.

Requirements

You must set up a database replication environment in order for the File Replication Service to identify replication servers with which to synchronize files. See Replication and Recovery (**replication.pdf**).

The File Replication Service uses the **updinst** utility to initiate the file replication process. See updinst for more information about this utility.

Configuration

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

Table 8.1: idfilerep service options

Option	Description
Port number this service is running on	Specifies the port or the shared memory ID to listen on. The default is 2380.

The File Replication Service archives existing files before overwriting them. By default, the archived files are stored in the Logs directory for the instance (<*Program Files path*>\Hitachi ID\IDM Suite\Logs\→<*instance*>\). You can change the archive directory by using the **Manage the system** → **Maintenance** → **Options** → **FILE REPLICATION ARCHIVE DIR** setting. This directory will be automatically created on the other instances during file replication if it does not already exist.

The **Manage the system** → **Maintenance** → **Options** → **FILE REPLICATION TIMEOUT** setting is used to specify a timeout value (in seconds) before the File Replication Service disconnects. The default value is 300 seconds. This timeout only applies if servers lose their connection while backing up or deleting files; an error occurs immediately if the servers are unable to maintain a connection while replicating files.

To manually perform file synchronization:

1. Click **Manage the system** → **Maintenance** → **File synchronization**.
2. Select all file replication servers that you want to synchronize. You can choose file replication servers and proxy servers.
3. Click **Synchronize**.

If any nodes are missing from the ***File synchronization*** page (**Manage the system** → **Maintenance** → **File synchronization**), verify that the missing nodes have network connectivity, then restart their File Replication Services. Reload the ***File synchronization*** page. The missing nodes should be displayed after restarting their File Replication Services.

If the server on which you are running the File Replication Service cannot access the other replication servers using the hostname (that is, database replication has to use the node's IP address to connect with other nodes), you can set the "serveraddress" string value in the instance's registry to broadcast the node's IP address to other replication nodes. This address can be used to set the file replication information.

You can control whether or not to archive existing files by adding the following registry entry in:

`HKLM\SOFTWARE\Hitachi ID\IDM Suite\<instance>\`IDFileRep

Entry name backups

Value 0|1 Set to 0 to disable backups

Data type DWORD

Default 1

The File Replication Service is used in conjunction with **updinst**. By default, **updinst** replicates all files and registry settings in *Bravura Security Fabric* instance. You can write an **updinst.cfg** file to provide additional configuration, including a white list and black list of files and settings to replicate.

A sample of **updinst.cfg** is located in the samples\ directory. This configuration file must be placed in the \<*instance*>\psconfig\ directory before it can be used by the File Replication Service. Use this configuration file to control which files and registry settings are replicated to other instances (white list) and which are not replicated (black list). The white list settings override black list settings.

WARNING!: All file and configuration modifications should be done on the same server (the primary). When attempting to run **updinst** from a node other than the primary, an error will occur, and the operation will be aborted. In extreme circumstances there is an option to force external data store replication (`-extdb -forcerun`) from a secondary node; however that should be done only when that database was corrupted on the primary (and its backups that are created every time the external data store is updated, were also corrupted) but the database, or a backup, survived on a secondary node. If **updinst** is run from more than one server, or if file or registry changes are made on secondary nodes, it is possible for it to overwrite newer files or settings that exist on secondary nodes. If a server with missing files runs **updinst**, that will remove those same files on all other instances.

If a problem occurs during file replication then a notification email is sent to the administrator, and the FILE REPLICATION FAILURE event is triggered.

See also:

- updinst for information about **updinst** usage.

iddbadm 9

Description

Use the **iddbadm** program to modify and configure the credentials used by **iddb** to connect to the database backend.

Usage

Run **iddbadm** with the following arguments:

```
iddbadm.exe [-database <database>] [-dbserver <dbserver>]
[-iddbport <iddbport>]
-dbtype <dbtype> -dbuser <dbuser> -instance <instance> -password <password>
[-servicename <servicename>] [-dbversion <dbversion>]
```

Table 9.1: iddbadm arguments

Argument	Description
-database <*database*>	The database name (required for MSSQL database only)
-dbserver <*dbserver*>	The database server (required for MSSQL database only)
-iddbport <*iddbport*>	The database service TCP port
-dbtype <*dbtype*>	The database type (MSSQL)
-dbuser <*dbuser*>	The database server user ID
-instance <*instance*>	The *Bravura Security Fabric* instance name
-password <*password*>	The database server user password
-dbversion <*dbversion*>	The version of the database (required for MSSQL database only)
-showconfig <*showconfig*>	Show current DBMS backend configuration

Examples

1. To change the DBMS credentials for a MSSQL server:

```
iddbadm -dbtype MSSQL -dbuser mssqluser -instance idminstance -password
dbuserpassword -database dbname -dbserver dbserver.com -dbversion 2008
```

importdata 10

Description

> **CAUTION:** **importdata** and **upgradedb** were deprecated in version 8.1.0 in favor of **migratedata**. Refer to migratedata.

Use the **importdata** program to import data from an existing *Hitachi ID Bravura Security Fabric* database schema on the same database type. Script files, defining SQL commands to be executed, are provided for migrating data for specific older versions of *Bravura Security Fabric*, or from instances of the same version. There are also different script files for migrating configuration data or user data.

The **importdata** program can also be used to apply SQL scripts to patch or upgrade the existing database schema. Please contact Hitachi ID support for the latest upgrade scripts.

Requirements

Data import can only occur on the same database server. The utility does *not* use the new instance's Database Service (iddb), but it does look up new database connection information from the registry to perform the SQL commands included in the script file specified. As such, this program needs to be run from a command prompt within the instance. Prior to running **importdata**, the database credentials used by the new instance must be configured so that they can read data from the old/temporary database.

Install the new instances of *Hitachi ID Bravura Security Fabric* and *Hitachi ID Connector Pack* before importing data.

> **CAUTION:** It is highly recommended that you shut down all services and web access while migration occurs, to prevent interference.

Usage

```
importdata.exe [-olddb <olddb>] -script <script> [-verbose]
```

Table 10.1: importdata arguments

Argument	Description
-olddb <*olddb*>	The prefix of the old database from which to import data.
-script <*script*>	The script file to use for importing data or applying to the database (required). See Database upgrade script files for migrating data.
-verbose	Describe what is being run.

The olddb prefix represents the database context and schema that holds the tables; in the format `<database>.<schema>`.

Examples

1. To execute **importdata** on the configuration script "upgradedb-7.2to8.1-mssql-initial.sql", run the following to migrate data:

   ```
   importdata.exe -olddb OLDPREFIX -script upgradedb-7.3to8.1-mssql-initial.sql -
    verbose >import-initial.txt
   ```

   This will also redirect the output to a text file - `import-initial.txt` to help with troubleshooting.

2. To use **importdata** to apply a patch to the existing database schema, run the following:

   ```
   importdata.exe -script <path to the SQL script> -verbose
   ```

10.1 Database upgrade script files for migrating data

Scripts used with **importdata** to migrate data are located on the Hitachi ID Systems support portal. Contact support@Hitachi-ID.com for assistance.

Migration from *Hitachi ID Bravura Privilege* 5.2.x requires the **upgradedb-5.2to7.1-<*db type*>.sql** script. This script migrates the following information:

- Target systems and credentials
- Profile and account attribute definitions
- Scheduled jobs
- Exit traps
- Plugins
- Objects
- ACLs
- Attribute groups
- Managed systems
- Managed groups
- User classes
- Requests
- Users
- Accounts
- Attribute values
- Authorizers
- Group memberships
- Session data
- Authorizations

Use **importdata** to run this script *once* only.

You must execute the scripts in two stages:

1. **upgradedb-<*old*>to<*new*>-<*db type*>-initial.sql**

   This script migrates configuration information, including:

   - Target systems and credentials
   - Profile/request and account attribute definitions
   - Templates
   - Roles
   - Scheduled jobs
   - Exit traps
   - Plugins
   - Objects
   - Locations
   - ACLs
   - Attribute groups
   - Managed groups
   - User classes
   - Network resources.

   Use **importdata** to run this script *once* only.

2. **upgradedb-<*old*>to<*new*>-<*db type*>-data.sql**

   This script migrates information including:

   - Certification campaigns
   - Requests
   - Users
   - Accounts
   - Attribute values
   - Authorizers
   - Group memberships
   - Roles assigned
   - Session data
   - Authorizations

   Use **importdata** to run this script as many times as required; for example, where you are running two instances in parallel, you migrate the basic configuration using the *-initial.sql script, then as you do your acceptance testing and validation you progressively synchronize the user data between the two instances before making the final switch. The data is reloaded from the older schema tables into the newer schema on each execution of this stage.

instdump 11

Description

The **instdump** program is run at the end of the auto discovery process and writes a configurations summary to a file named config-<*yyyy*>-<*mm*>-<*dd*>.kvg in the <*Program Files path*>\Hitachi ID\IDM Suite\Logs\<*instance*>\ directory. The file can be used by Hitachi ID Systems support to help provide assistance.

The **psupdate** program runs the **instdump** program when the **Maintenance** → **Options** → **PSUPDATE INSTDUMP** setting is enabled (disabled by default).

> **Note:** Some arguments are dependent on the Hitachi ID Systems product license.

Usage

```
instdump.exe [ <options> ] -outfile <outfile>
```

Table 11.1: instdump arguments

Argument	Description
-acl	Output ACL information.
-attrgrp	Output attribute group information. Requires *Bravura Identity*.
-authenidlist	Output authentication target system list information.
-authorizer	Output authorizer information. Requires *Bravura Identity*.
-binaryversion	Output binary version information.
-consoleuser	Output product administrator information.
-customfile	Output custom file information.
-exittrap	Output event action (exit trap) information.
-inventory	Output inventory information. Requires *Bravura Identity*.
-managedgrp	Output managed group information. Requires *Bravura Identity*.
-outfile <*outfile*>	File name for output (required).
-plugin	Output plugin information.
-profileattr	Output attribute information.
-registry	Output registry information.

. . . continued on next page

Table 11.1: instdump arguments (Continued)

Argument	Description
-role	Output role information. Requires *Bravura Identity*.
-service	Output service information.
-strength	Output strength information.
-system	Output system configuration variables.
-target	Output target information.
-targetattr	Output account attribute information.
-template	Output template information. Requires *Bravura Identity*.
-userclass	Output userclass information.
-verbose	Use a more readable format for output.

licviewer 12

Description

Use the **licviewer** program to view the details of your *Hitachi ID Bravura Security Fabric* license. It presents the license expiry date, licensed modules, and numbers of licensed users and target systems.

Usage

```
licviewer.exe <license file> [-o <output file>]
```

Table 12.1: licviewer arguments

Argument	Description
<*license file*>	License file
-o <*output file*>	File to output the license details to

Example

To use **licviewer** to read the idmsuite.lic license file and output the results to licensedetails.txt, type:

```
licviewer ..\license\idmsuite.lic -o licensedetails.txt
```

loadplatform 13

Description

Use the **loadplatform** program to query connectors for their abilities and populate PLATFORM,OBJOPER and OBJREL database tables with the information; Also it sets the default attributes for the connector platforms by populating ATTRDEF and ATTRDEFVAL tables. This is particularly useful when custom connectors have been created and the target type needs to be available in the user interface.

Hitachi ID Bravura Security Fabric target systems types are displayed in **Type** drop-down list on the ***Target system information*** page. Target types in this list are displayed according to target system category.

13.1 Requirements

The client software required by the specified target systems must be installed or else the platform data for the connector will not be imported to the database.

The **loadplatform** program loads a binary executable (.exe), or a scripted platform definition file (.con) (p43) that calls a binary agent. If you do not specify an .exe or .con extension, the program looks for files with either extension. If both exist, **loadplatform** loads the .exe file.

To load and list official scripted connectors, both the scripted platform definition file (.con) and the configuration script specified within the .con file must be located in the agent directory.

13.2 Usage

```
loadplatform.exe -a <connector name> [-dry-run]

loadplatform.exe -target [-dry-run]
```

Table 13.1: loadplatform arguments

Argument	Description
-v, --version	Print out version and exit
-a, --agent <*connector name*>	Load the specified connector.
-target	Load all target system connectors.
-d, --dir <*directory path*>	Changes the directory to look for the agents and connectors in the specified path.
-32bit	Load 32-bit connectors.
-dry-run	Query the specified connectors but do not write the information to the database.
-force	Forcibly update attribute information if conflict exists.
-list-db-inserts	Include a list of all inserted database values.

13.2.1 Examples

1. To import information about the Unix connector into the database, type:

   ```
   loadplatform.exe -a agtunix.exe
   ```

2. To see the operations supported by the Active Directory DN connector, type:

   ```
   loadplatform.exe -dry-run -a agtaddn.exe
   ```


13.3 Loading a new scripted target system type

Some target system types listed on the **Target system information** page are defined by scripted platform definition files that call a binary agent such as the SSH scripted agent (**agtssh**) and specify a configuration script that defines supported operations.

Scripted platform definition files are written in the following format:

```
# KVGROUP-V2.0
 <name> = {
   agent = <binaryToRun>;
   script = <script>;
   category = <category>;
   platform = <platformId>;
   description = <languageTagName>;
 }
```

for example:

```
 # KVGROUP-V2.0
 agtssh-sample = {
   agent = agtssh.exe;
   script = sampleScript.cfg;
   category = SCRIPT;
   platform = AGTSSH-SAMPLE;
   description = !!!AGTSSH-SAMPLE-DESC;
 }
```

> **Note:** Official scripted connectors are only compatible with *Bravura Security Fabric* 10.0 and above.

The keys in the scripted platform definition file are all required, and are all case sensitive. The "category" must be a valid platform category. These are described in **platcat.csv** in the agent\dat directory.

To load a new scripted target system type:

1. Write a configuration script in the format described in SCRIPT TYPES in the *Script Systems Integration Guide (**script-systems.pdf**)*.
2. Write a scripted platform definition file in the format described on this page.
3. Add both the configuration script and the .con file to the agent directory.
4. From the util directory, run:

   ```
   loadplatform -a <con filename>.con
   ```

   This loads the new target system type into the instance database.

Scripted platform definition files and configuration scripts can also be loaded from other directories outside the agent directory. To do this, place both the configuration script and the .con file into the desired directory and run loadplatform with the absolute or relative path to the .con file. For example:


```
loadplatform -a <con filename>.con -d "C:\path\to\agent"
```

SQL scripted connectors also support defining managed identities by using a configuration file in following format:

```
# KVGROUP-V2.0
 <name> = {
   agent = <binaryToRun>;
   script = <script>;
   category = <category>;
   platform = <platformId>;
   description = <languageTagName>;
   objects = <object type>;
 }
```

for example:

```
 # KVGROUP-V2.0
 agtoracustom = {
   agent = agtorascript.exe;
   script = agtoracustom.cfg;
   category = ATTAP;
   platform = ORACUSTOM;
   description = "Custom oracle target";
   system = false;
   objects = {ACCT;ASSET;GRP;ROLE;};
}
```

Providing managed identities in the configuration file allows connectors to be loaded with only operations related to the specified objects.

13.4 Loading a new platform category

Hitachi ID Bravura Security Fabric can load a new platform category dynamically. *Hitachi ID Connector Pack* 1.1 or later is required.

To load a new platform category:

1. Modify the **`platcat.csv`** from the agent\dat directory.
2. Add a new category to the **`platcat.csv`** file.
3. Add the language tag for this new category to the **`en-errmsg.kvg`** file.
4. Generate and install a new set of skins.
5. From the util directory, run:

   ```
   loadplatform -target
   ```

   This loads the new platform category into the platcat table.

13.5 Loading default attributes

Hitachi ID Bravura Security Fabric can load default attributes for connectors from attribute files located in agent\dat directory. The default attributes for connectors are defined in different files which may include account attributes file, group attributes file and object attributes file depending on the connector supported operations.

To load the default attributes:

1. Modify or create a new attribute file(s) in the agent\dat directory.
2. From the util directory, run:

   ```
   loadplatform -a <agent>
   ```

   This loads or modifies the default attributes for the connector platforms by populating ATTRDEF and ATTRDEFVAL tables.

   You should be able to find default attributes for account and group under **Manage the system** → **Resources** → **Account attributes** or **Manage the system** → **Resources** → **Group attributes** then select the connector target system type or the connector target system.

13.6 Loading a new or modified discovery template

Hitachi ID Bravura Security Fabric can load target system discovery templates dynamically. *Hitachi ID Connector Pack* 3.1.0 or later is required. By default, they are located in the agent\dat directory.

Discovery template files are written in the following format:

```
KVGROUP-V2.0
templates = {
  <TARGET_TEMPLATE> = {
        name = <PSLang name description>
        address = <PSLang address description>
        <key> = <value>; # target system option
        ...
        ...
        <key> = <value>;
           Resources = {
           };
           TargetAttrs = {
           };
  };
}
```

for example:

```
# KVGROUP-V2.0
templates = {
  NT_TEMPLATE = {
    name = "$comp[\"dNSHostName\"][0]";
```



```
  address = "\"{server=\" + $comp[\"dNSHostName\"][0] + \";}\"";
  runlist = true;
  listattributes = true;
  listgroups = true;
  idarchivepush = true;
  adminresethide = true;
  selfresethide = true;
  adminunlockhide = true;
  selfunlockhide = true;
  adminclaimhide = true;
  selfclaimhide = true;
  selfmanagehide = true;
  listmembertype = A;
    Resources = {
      ls_scmacct;
      ls_taskacct;
      ls_iisacct;
      ls_comacct;
      ls_normacct;
     };
   TargetAttrs = {
     ADDR_ATTR = "DNSHOSTNAME";
     DESC_ATTR = "DNSHOSTNAME";
   };
};
```

The keys in the discovery template file are all required, and are all case sensitive. The name and address keys are written using PSLang expression based on account attributes discovered using auto discovery.

To load a discovery template:

1. Modify or create a new <target>-template.cfg file in the agent\dat directory.
2. From the util directory, run:

   ```
   loadplatform -a <agent>
   ```

This loads the agent with the new/modified discovery template onto the instance. You should be able to find the discovery template under **Manage the system** → **Resources** → **Target systems** → **Discovery templates**.

loadreports 14

Use the **loadreports** program to regenerate <*instance*>\report\reportsdata.dat which includes information on the available executable reports.

This utility automatically runs and generates this file when installing an instance of *Hitachi ID Bravura Security Fabric*, during the post-installation tasks. The **loadreports** program is only required after installation if custom or additional reports are installed.

migratedata 15

Use the Hitachi ID Systems Data Migration Utility to migrate user data from an older version of *Hitachi ID Bravura Pass* to a current version.

15.1 Prerequisites for the older version

- Users have security questions populated for their question sets including user-defined as well as pre-defined question sets (built-in and optionally custom).
- Password policies are updated for password history enforcement if password history is being exported. Users need to have reset their passwords previously in order to have values populated for the password history.
- If attributes are to be exported, users have profile attributes populated, where those profile attributes are not listed from any target systems, although they may be associated with attributes on a target system.
- Users may have their profiles locked out (too many bad authentication attempts) or disabled (a help desk administrator has disabled a user's profile). For testing purposes these accounts should be noted and revisited after the migration.
- Users have mobile devices that have been registered. The mobile device registration data may be exported so that the users are not required to re-register their devices.
- Target system credentials can be exported from *Hitachi ID Bravura Security Fabric* version 8.2.0 or newer.
- Team management data can be exported from *Bravura Security Fabric* version 10.1.4, and versions 11.1.1 or newer.
- The following *Bravura Privilege* data cannot be exported using the utility:
  - managed accounts and systems not onboarded using team management
  - import rules and related data
  - managed system policies
  - archived data (*_hst tables)
  - inactive managed systems and accounts
  - active managed account check-outs

15.2 Prerequisites for the current version

- Target systems should be set up prior to data import. Accounts should be discovered, and profiles should already be created.
- Target systems should reflect existing configuration between the older version and the current version of *Hitachi ID Bravura Security Fabric*. For example you should use newer agents such as the Active Directory DN (**`agtaddn`**) rather than the older and mostly obsolete Active Directory (**`agtad`**).
- Target system credentials will only be imported for target systems that match the IDs from the older version. This will override any existing credentials already defined on the current version.
- Password policies are updated to the intended policy. Password history enforcement must be enabled if password history is being imported.
- Profile attributes must be created for any profile attribute values that will be imported.
- A dry run might generate some entries in log with !!!TAG complains, but they are minor and only related to a multilingual UI. Any !!!TAG (multilingual) values should be recreated on imported target systems to avoid missing translations if you are deploying in additional languages.
- Hitachi ID Bravura One is configured on the *Bravura Security Fabric* server and the Mobile Worker Service (mobworker) is configured for mobile access with the Hitachi ID Bravura One proxy server.

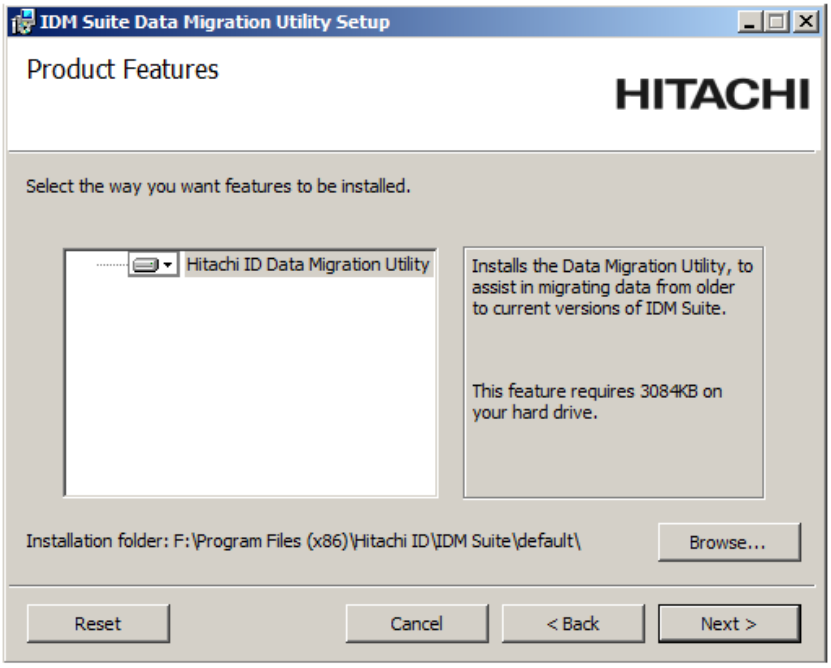
  When migrating the mobile device registrations and the instance name has changed for *Bravura Security Fabric* and/or the company name has changed for the Hitachi ID Bravura One proxy server Apache configuration, the **`Proxy server URL`** address for the Mobile Worker Service may remain the same, however the rewrite rule in the Apache configuration on the Hitachi ID Bravura One proxy server will need to be modified.

  This is to ensure that when one or both of these names have changed, that users will not be required to re-register their mobile devices. See the Hitachi ID Bravura One Configuration Guide for more information on the Apache configuration requirements.
- Randomization must be disabled for all accounts. It is also recommended that all checked out accounts are checked in prior to data export.
- The same team management components installed in the older version must also be installed in the current version. This includes the following:
  - *Bravura Privilege* refbuild components (pam_team_management, pam_team_vault_management)
  - system type scenario components (pam_system_type_winnt, pam_system_type_linux)
  - subscriber validation scenario component (pam_subscriber_validation)
  - personal admin scenario component (pam_personal_admin_management)
- The same source of profiles target used in the older version must also be added and configured in the current version to list users and manage groups.
- If the target system credentials of onboarded systems are associated with a *Bravura Privilege* managed account, you will need to manually add the managed system the account is a part of. As well, you will also need to manually create the managed system policy the managed account was originally added to (if it doesn't already exist), and bind the account to the managed system policy.

15.3 Data migration process

15.3.1 Exporting data from the older (source) version

1. Obtain a copy of the **`migratedata.msi`** Windows installer from Hitachi ID.
2. Copy **`migratedata.msi`** to the computer where the older (source) instance is installed.
3. On the computer where the source instance is installed, run **`migratedata.msi`**.
4. On the **Product Features** page, ensure that **Installation folder** is set to the directory where the source instance is installed; for example, the following source instance is installed under `F:\Program Files (x86)\Hitachi ID\IDM Suite\default`:



5. Continue with the remaining wizard pages to install **`migratedata`**.

   If you do not have at least SQL Server Native Client 2008 installed, you will receive a warning and installation will not proceed until it is installed. Upon completion of the installation process, **`migratedata.exe`** will be available in the `\util` sub-directory of the location specified for **Installation folder**.
6. Open a command prompt and navigate to the directory where **`migratedata`** is located.
7. Run **`migratedata`** using the following parameters:

   ```
   migratedata.exe -action export -file <dbfile> <datatype>
   ```

   where:

   - *<dbfile>* is the SQLite database file that will be created if it does not exist already, or opened if it already exists.

- `<datatype>` is one or more of the data types listed in Table 15.1.

8. Enter a password.

   The password will be used to encrypt the data key of the instance.

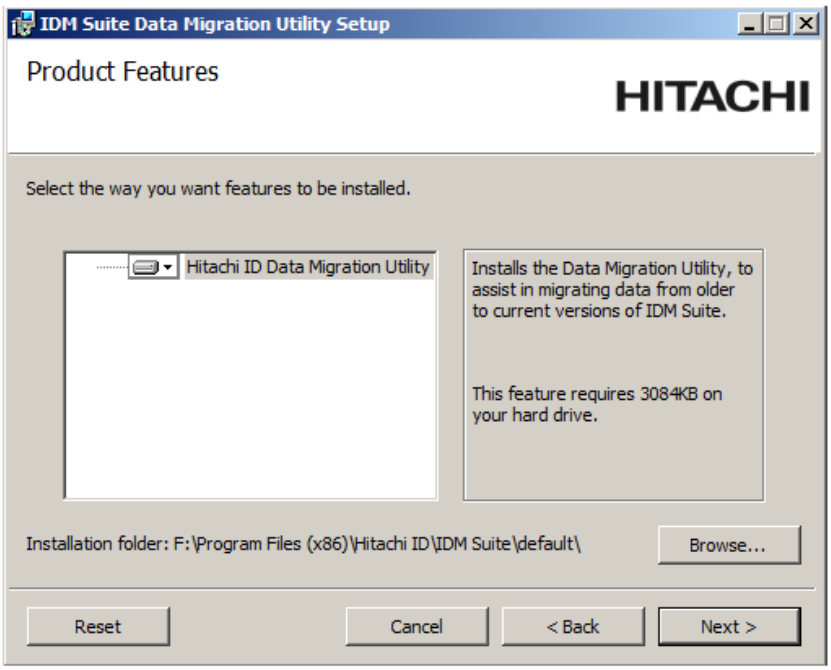
The **`migratedata`** utility will create the export SQLite database. If the SQLite database file already exists, then it will be updated but the password must match the one used to create the database file.

Table 15.1: migratedata data type arguments

Argument	Description
-qaconfig	Question set configuration (qdef+qset tables) Security question configuration settings and lists of custom questions.
-qadata	Question set data (response+responseqd tables) Answers provided by the users for the security questions. **Note:** -qaconfig is required when -qadata is specified.
-madmin	Target system credentials (madmin table) Administrator credentials configured for target systems.
-mobilereg	Mobile registrations (usermobiledevice table) Mobile device registration data for users that have registered a Hitachi ID Bravura One App with the *Bravura Security Fabric* server.
-pamteam	*Bravura Privilege* team management data Configuration of teams, including team groups, members, privileges, systems and accounts
-force-pamteam-export	Force export of *Bravura Privilege* team management data, even if there are checked out managed accounts. Checked out managed accounts will be considered checked in.
-pwhistory	Password history (history table) List of passwords reset by users when password history is enforced in the password policy rules.
-userstat	Userstat tags (userstat table) Userstat tags/records and their values set for users for specific actions.
-userattr	Profile attributes that didn't come from a target (userattr table) Profile and request attributes that have values stored in the database and are not read in from any targets.
-userauth	Profile lockout/disabled-ness (userauth table) User profiles that have been locked out or disabled by the help desk.
-all	All of the above data types

15.3.2 Importing data to the current (destination) version

1. Locate a copy of the **`migratedata.msi`** Windows installer from Hitachi ID.
2. Copy **`migratedata.msi`** to the computer where the current (destination) instance is installed.
3. On the computer where the destination instance is installed, run **`migratedata.msi`**.
4. On the **Product Features** page, ensure that **Installation folder** is set to the directory where the destination instance is installed; for example, the following destination instance is installed under `F:\Program Files (x86)\Hitachi ID\IDM Suite\default`:



5. Continue with the remaining wizard pages to install **`migratedata`**.

   If you do not have at least SQL Server Native Client 2008 installed, you will receive a warning and installation will not proceed until it is installed. Upon completion of the installation process, migratedata.exe will be available in the `\util` sub-directory of the location specified for **Installation folder**.

6. Copy the SQLite database file you created in Exporting data from the older (source) version to the computer where the *Hitachi ID Bravura Security Fabric* instance is located for the current version.
7. (Optional) Perform a dry run (i.e. not actually import data into the backend database used by the instance).

   You can do this by running **`migratedata`** in the destination instance's `\util` sub-directory with the following parameters:

   ```
   migratedata.exe -action dryrun -file <dbfile> <datatype> -log <logfile>
   ```

   where:

   - *<dbfile>* is the SQLite database file from Step 6.

- *<datatype>* is one of the data type values listed in migratedata data type arguments to be imported.
- *<logfile>* is the file used to log the results.

8. (Optional) Enter the password used to encrypt the data key in the database file.
9. (Optional) Review the dry run results for any errors.
10. To import the data, run **migratedata** utility in the destination instance's \util sub-directory with the following parameters:

    ```
    migratedata.exe -action import -file <dbfile> <datatype> -log <logfile>
    ```

    where:

    - *<dbfile>* is the SQLite database file from Step 6.
    - *<datatype>* is one of the data type values listed in migratedata data type arguments to be imported.
    - *<logfile>* is the file used to log the results.
11. Enter the password used to encrypt the data key in the database file.
12. Review the results of the migration and then proceed to verify the upgrade.

Note: If **migratedata** seems to hang on import, wait for its timeout (30min) before killing the process.

Note: Do *not* left-click on the command prompt screen when a console app is running, especially for apps like **migratedata** which takes a long time to complete.

If something is selected in the command console while **migratedata** runs, it may seem that the process is stuck, but in fact the command prompt is paused, waiting for the user to do something with the selection; in that case, press any key (e.g. **[Delete]**) to remove the selection and allow the prompt to continue processing.

15.4 Items to verify after the data migration

Question set configuration

Confirm that:

- Any customized settings for the question sets such as for user-defined and pre-defined are present.
- Any custom pre-defined questions that existed in the older version are also now present in the current version after the data migration.

Security questions

Confirm that:

- A user's security questions and answers match those from the imported data. For example, user-defined as well as pre-defined question sets for both built-in and custom security questions. Previously defined question sets for the account that existed on the current instance will be replaced.
- A user is able to authenticate to self-service using the security questions and answers from the imported data.
- A user is able to use the "Test mode" for all of their security questions successfully when valid answers from the imported data are provided.
- If configured to do so, a help desk administrator may see the list of the user's security questions and they are from the imported data.

Password history

Confirm that:

- A user is unable to reset their password to a value that is in the password history from the imported data following the password policy.
- A user is able to reset their password to a new value that is not in the password history.
- A help desk administrator is unable to reset the password for a user to a value that is in the password history from the imported data.
- A help desk administrator is able to reset the password for a user to a new value that is not in the password history.

Profile attributes

Confirm that:

- Values for a user's profile attributes are replaced with the values from the attributes in the imported data.

Statuses for profile lockout and profile enabled/disabled

Confirm that:

- The status for whether or not a user's profile is locked out when a user has provided too many bad authentication attempts to self-service and has locked out their profile is replaced with the status from the imported data.
- The user will be unable to login to self-service if their profile is locked out.
- A help desk administrator is able to see whether or not a user's profile is locked out or not.
- The status for whether or not a user's profile has been disabled for when a help desk administrator has disabled their profile is replaced with the status from the imported data.
- The user will be unable to login to self-service if their profile is disabled.
- A help desk administrator is able to see whether or not a user's profile is disabled or not.

Userstat tags

From the instance after the import is complete, locate and run the Userstat report. The Userstat report will include records for the users from the imported data. For example, the report output will show the PSQDONE userstat tag for users from the imported data that had completed their security questions profile.

Target system credentials

Confirm that:

- Target system credentials are imported, assuming that the target systems exist on the current version and matches the target system IDs of the older version. If a target system exists on the current version and already have credentials defined, its password will be overwritten during import.
- Target system credentials are not exported if the target system does not exist on the current version.
- Target system credentials of onboarded systems are associated with the same *Bravura Privilege* managed account, if associated in the previous version.

Mobile device registrations

Confirm that:

- A user is able to use the Hitachi ID Bravura One App on their mobile device to login to and access *Bravura Security Fabric* for devices that were previously registered.
- If configured, a user is also able to still authenticate successfully for Computer Login on the desktop for a mobile authentication chain to scan a QR code using the mobile device.
- The mobile devices are not required to be re-registered.

Team management

Confirm that:

- The team admin from the previous version will still be able to create teams, manage group membership of teams, and delete teams.
- In versions 12.0 and newer, the trustee privilege is broken down into several trustee privileges, including Team trustee, System trustee, Account trustee, Vault trustee, OTP trustee, LC trustee and Subscriber trustee. By default, the trustee user from the previous version will have all of these privileges.

  Refer to the *Bravura Security Fabric* Documentation for more information about trustee privileges.
- The same systems and accounts are present in the current version.
- The same team vaults and vault accounts are present in the current version.
- The same OTP API user accounts are present in the current version.
- Account settings, such as disclosure method, session monitoring, and randomize/override settings are unchanged.
- Managed account and vault account passwords are unchanged.
- Vaulted files can be downloaded as normal.
- Managed accounts previously checked out before migration are checked in, and can be requested and checked out as normal.
- Personal admins continue to have unrestricted access to the managed accounts they are entitled to.
- Users with Requesters, Approvers, Auto-approved, and Credential manager privileges in the previous version will continue to have the same access in the current version.
- Users that were help desk trustees before migration continue to have help desk privileges in the current version.
- Target system credentials of onboarded systems are associated with the same *Bravura Privilege* managed account, if associated in the previous version.

resetkey 16

Description

Use the **resetkey** program if you need to reset the communication key (or Master Key), Connector encryption key, or IDMLib encryption key in the registry.

For example, if you've forgotten any of the encryption keys, or if you have a policy which requires you to change it on a regular basis.

Updating the registry *must* occur on all servers, including listeners, proxy servers, application instance servers, IDDB replication nodes, and transparent password synchronization triggers. If this change is *not* completed on all servers, then communication between these servers can fail. You can export the reset encryption keys to the **idmsetup.inf** file, which is used to load information during the initial installations on these servers.

Usage

```
resetkey.exe -type <keytype> -value <keyvalue>

resetkey.exe -type <keytype> -export [-value <keyvalue>] [-file <file>]
```

Table 16.1: Resetkey arguments

Argument	Description
-type <*keytype*>	The key type for the encryption key that is being reset. Valid types are: • commkey • connectorkey • idmlibkey
-value <*keyvalue*>	The hexadecimal key to set for the new value for the key as specified for -type in the registry, **idmsetup.inf** file, or other specified file. The specified <*keyvalue*> must be 64 hexidecimal characters in length.
-export	Copies the registry value for the specified encryption key, encodes it, and writes it to the INF file.
-file <*file*>	Allows you to specify the name and location of an alternate INF file to which to export the encrypted key value. The default name and location is <*instance*>\psconfig\idmsetup.inf. The encryption key value can be taken from the registry, or specified using the -type and -value arguments.

Examples

1. To update the communication key (or Master Key), Connector encryption key, or IDMLib encryption key in the registry with a specified hexidecimal key, type:

```
resetkey -type commkey -value <64-character-key-value>
resetkey -type connectorkey -value <64-character-key-value>
resetkey -type idmlibkey -value <64-character-key-value>
```

2. To update the communication key (or Master Key) in a file named idm-copy.inf, located in the current directory, using the communication key (or Master Key) value from the registry, type:

```
resetkey -type commkey -export -file idm-copy.inf
```

> **Note:** The specified INF file must be a valid setup file that follows the format of **`idmsetup.inf`**. It is recommended that you backup the INF file before exporting the current communication key (or Master Key) to the file.

See also:

- idmsetup.inf in the *Reference Manual* for details on **`idmsetup.inf`**

sqlutil 17

Description

Use the **sqlutil** program to run SQL scripts to apply database fixes, procedure replacements, and so on. The utility automatically handles schema string replacement and replication. This eliminates the need to manually connect to multiple databases (multi-node setups) via MSSMS and run the same script for each.

17.1 Usage

Before running **sqlutil**, navigate to the \<*instance*>\ directory and run **instance.bat** to configure necessary environment variables.

```
sqlutil.exe <script> [-force]
```

Table 17.1: sqlutil arguments

Argument	Description
-force	Forces the SQL script to run if it has been run before.

See also:

You can run a report on script execution history. See SQL utility history in the *Reports User Guide (**reports.pdf**)*.

update_db_crypto 18

Description

Use the **update_db_crypto** program to convert database passwords with AES-128 encryption to AES-256.

This utility can convert passwords from the following database tables:

- did
- importadmintest
- madmin
- piqueue
- ppqueue
- reqacct
- response
- responseqd
- smevent
- wstnpwd
- wstnpwdhis
- wstnsyncpwd
- wstnsyncpwdhis

Usage

```
update_db_crypto.exe -listall | -tables <tables> | -updateall [-numupdates <1-100000>]
```

Table 18.1: update_db_crypto arguments

Argument	Description
-listall	List all database tables passwords can be converted on.
-tables <*list of tables*>	Specify a space-delimited list of tables on which passwords will be converted.
-updateall	Update passwords from all supported database tables.
-numupdates <*1-100000*>	The number of passwords to be updated before data is committed to the database. This can range between 1 to 100000 passwords.

Examples

1. To convert all passwords with AES-128 encryption from the supported database tables to AES-256:

   ```
   update_db_crypto.exe -updateall
   ```

2. To convert the first 100 passwords with AES-128 encryption from the supported database tables to AES-256 before committing the data to the database:

   ```
   update_db_crypto.exe -updateall -numupdates 100
   ```

3. To convert passwords with AES-128 encryption from the wstnsyncpwd and wstnsyncpwdhis table to AES-256:

   ```
   update_db_crypto.exe -tables wstnsyncpwd wstnsyncpwdhis
   ```


updinst 19

Description

The **updinst** program synchronizes files and registry settings between servers in a multiple-instance environment, or a replication environment. The program is run during auto discovery when the **Maintenance** → **Options** → **PSUPDATE FILE REPLICATION** setting is enabled. This is the default setting.

This program also collects and synchronizes proxy log files onto the instance server.

This program is used in conjunction with the File Replication Service (idfilerep). See idfilerep for more information about this service.

By default, **updinst** replicates all files and registry settings in *Bravura Security Fabric* instance. You can write an **updinst.cfg** file to provide additional configuration, including a white list and black list of files and settings to replicate.

A sample of **updinst.cfg** is located in the samples\ directory. This configuration file must be placed in the \<*instance*>\psconfig\ directory before it can be used by the File Replication Service. Use this configuration file to control which files and registry settings are replicated to other instances (white list) and which are not replicated (black list). The white list settings override black list settings.

WARNING!: All file and configuration modifications should be done on the same server (the primary). When attempting to run **updinst** from a node other than the primary, an error will occur, and the operation will be aborted. In extreme circumstances there is an option to force external data store replication (`-extdb -forcerun`) from a secondary node; however that should be done only when that database was corrupted on the primary (and its backups that are created every time the external data store is updated, were also corrupted) but the database, or a backup, survived on a secondary node. If **updinst** is run from more than one server, or if file or registry changes are made on secondary nodes, it is possible for it to overwrite newer files or settings that exist on secondary nodes. If a server with missing files runs **updinst**, that will remove those same files on all other instances.

CAUTION: Do *not* attempt to replace Database Service files using **updinst** or the File Replication Service. Updating the Database Service and related files (such as **iddbmssql.dll**) must be done manually on all instances. This only applies to the Database Service service. All other services can be updated using the File Replication Service. To update the Database Service files manually, shut down all services on the instance, back up all services, and then replace the Database Service files.

If a problem occurs during file replication then a notification email is sent to the administrator, and the FILE REPLICATION FAILURE event is triggered.

Usage

```
updinst.exe [-list] | [-showconfig] | [-synchfile] [-syncreg]
[-globalcp]
 → [-serverid <serverid>...] [-dry-run] |
[-extdb[-forcerun]] | [-getlogs] [-proxyfile <proxylist.csv>] [-removelogs]
```

Table 19.1: updinst arguments

Argument	Description
-dry-run	Only show what would be done without making changes.
-getlogs	Retrieve and synchronize all logs from all proxies configured onto the instance server. Logs will be under their respective \Logs\proxy_<proxy_IP> directory.
-globalcp	Synchronize the global connector pack in addition to the current instance. Used in conjunction with the -syncfile argument.
-list	List file replication services involved and exit. Display the status of each service, node ID, address, and port. If the service is unreachable, check the connection of the Database Service for each server.
-proxyfile <proxylist.csv>	Retrieve and synchronize all logs from proxies listed in the \<*instance*>\psconfig\proxylist.csv file. Logs will be under their respective \Logs\proxy_<proxy_IP> directory. Where proxylist.csv contains a list of proxy IP addresses and their ports. This is only used in conjunction with -getlogs.
-serverid <serverid> . . .	Synchronize to specified servers; can be used with multiple server IDs. The server ID can be found using the -list argument. If no server is specified, all servers are updated.
-showconfig	Show built-in configuration and exit.
-syncfile	Synchronize files to other servers.
-syncreg	Synchronize registry settings to other servers.
-extdb	Synchronize the **extdb** tables.
-forcerun	Forces the current node to synchronize the **extdb** tables.
-removelogs	Remove all previously fetched logs from all proxies. Can be used in conjunction with -getlogs to remove logs that no longer exist on the remote server.
-filedir <filedir>	Specific file path from which files are synchronized to other servers. This is the relative path under the instance's main file folder.
-registrydir <registrydir>	Specific registry path from which settings are synchronized to other servers. This is the relative path under the instance's main registry folder.

Examples

1. To show the default configuration file:

   ```
   updinst -showconfig
   ```

2. To show a list of all file replication services that are running:

   ```
   updinst -list
   ```

3. To synchronize all files and registry settings on all servers:

   ```
   updinst -syncfile -syncreg
   ```

4. To synchronize a specific file directory on all servers:

   ```
   updinst -syncfile -filedir <filedir>
   ```

5. To synchronize multiple file directories on all servers:

   ```
   updinst -syncfile -filedir <filedir1> -filedir <filedir2> ...
   ```

6. To synchronize a specific registry setting on all servers:

   ```
   updinst -syncreg -registrydir <registrydir>
   ```

7. To run a dry-run on a specific server:

   ```
   updinst -serverid <server id (guid)> -syncfile -syncreg
   ```

8. To update global *Connector Pack* files:

   ```
   updinst.exe -globalcp -syncfile
   ```

9. To collect and synchronize logs from all proxies:

   ```
   updinst.exe -getlogs
   ```

10. To collect and synchronize logs from all proxies listed in the proxylist file:

    ```
    updinst.exe -getlogs -proxyfile <proxylist.csv>
    ```

    Where the proxylist.csv contains:

    ```
    ::proxy, port::
    10.0.39.111, 3344
    10.0.39.121, 3344
    ```

11. To synchronize the **extdb** tables:

    ```
    updinst.exe -extdb
    ```

12. To remove all previously fetched logs:

    ```
    updinst.exe -removelogs
    ```

13. To remove logs that no longer exist on the remote server:

    ```
    updinst.exe -removelogs -getlogs
    ```


19.0.1 Use Case: updinst config file

Most of these changes are system variables listed in the registry section at **Manage the system** → **Maintenance** → **System variables**.

In the updinst config file:

- All file paths are rooted not at the instance directory, but at the backslash immediately before it: (instead of script\somefile.py it has to be:

  ```
  ^\\\\script\\\\somefile.py.
  ```

- Here is an example, that adds three blacklisted registry entries (meaning that they will not be sent from the primary to secondary application nodes):

  This example shows that blacklisting these entries ensures that the SENDER_EMAIL and ServerAddress are not propagated from the primary to the secondaries. These two entries will be unique to each server.

  The IDAPISOAP entry is an example of an entry that already exists and is hard coded in the configuration file.

  ```
    # KVGROUP-V2.0
  "updinst" "cfg" = {
  }

    # Registry path blacklist.  Registry paths that match the regular
    # expressions here are not considered for replication, unless they
    # appear in regWhitelist.  All paths are relative to the instance
    # registry root.  Registry keys (and value names) are separated by
    # double-escaped backslashes (\\\\).
    "regBlacklist" = { "^IDAPISOAP\\\\endpoints",
                       "SENDER_EMAIL",
                       "ServerAddress" };
  ```


Part III

APPENDICES

File Locations A

This chapter provides details of the location and purpose of files installed by:

- *Hitachi ID Bravura Security Fabric* (p67)
- *Hitachi ID Connector Pack* (p72)

When you install any Hitachi ID Systems product, the default path for program files is:

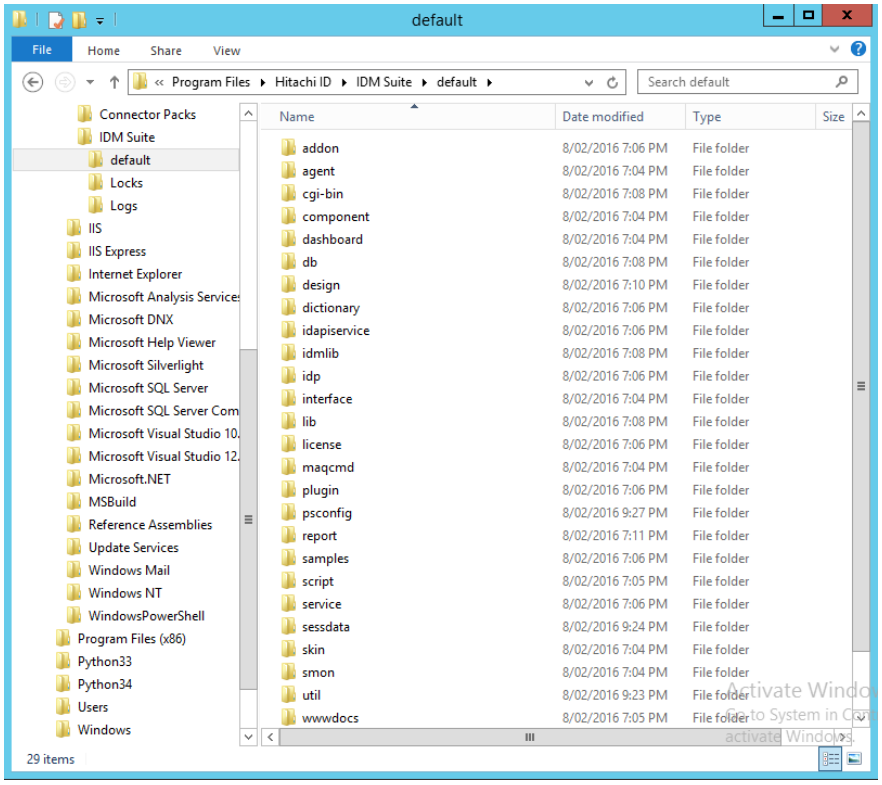
C:\Program Files\Hitachi ID\

A.1 *Bravura Security Fabric* directories and files

There are three main directories that are created when you install *Bravura Security Fabric* 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\

The contents of those directories are detailed in the following subsections.



It is recommended that you do *not* change these directory locations during the setup process. You cannot install any of the directories required for *Bravura Security Fabric* on a mapped drive.

A.1.1 Instance directory

Instance directory files describes the function of directories that are created when an instance of *Bravura Security Fabric* is installed.

Note: Directories marked with ⋆ include files installed by *Connector Pack*.

Directories marked with ⋆⋆ include folders and files installed with the optional *Analytics* app.

Directories marked with † include optional files. They are only installed in a complete installation or if selected in a custom installation.

Table A.1: Instance directory files

	Directory	Contains
†⋆	addon	Files required for add-on software, such as Password Manager Local Reset Extension and secure kiosk account (SKA). Some files, required to target Netegrity SiteMinder, are installed by *Connector Pack*. If you installed a global *Connector Pack*, these files are contained in the *Connector Pack* global directory.
⋆	agent	Instance-specific user management connectors (agents). If you installed a global *Connector Pack*, user management connectors are contained in the *Connector Pack* global directory.
⋆⋆	analytics	*Analytics* app specific folders
⋆⋆	analytics\DataSets	Contains **`*.rsd`** files which are Shared Dataset Definitions. These files are only used by SQL Server versions higher than Express. They contain datasets that are shared between reports.
⋆⋆	analytics\Hidden	Contains **`*.rdl`** files which are Report Definitions. These files are the drillthrough reports used by other reports. They are not visible to the end-user.
⋆⋆	analytics\ReportItems	This folder contains other folders. Each folder in this folder is a category in the *Analytics* app. Within these folders are **`*.rdl`** files which are Report Definitions. The folders need to be added to the **CUSTOM ANALYTIC CATEGORIES** system variable to be visible. These reports are then visible to the end-users in the *Analytics* app.
	cgi-bin	The user web interface modules (**`*.exe`** CGI programs).
	db	The *Bravura Security Fabric* database SQL scripts.
	db\cache	Search engine temporary search results. These files are cleaned up nightly by **`psupdate`**.
	db\replication	Stored procedure replication queues, and temporary replicated batch data.
⋆	design	Files necessary to make modifications to the GUI. Some files are installed by *Connector Pack*. If you install a global *Connector Pack*, files related to connectors are located in the global design directory. See the *Bravura Security Fabric* Documentation for details.
	dictionary	A flat file, **`words.dat`**, that contains dictionary words. *Bravura Security Fabric* uses this file to determine if new passwords fail dictionary-based password-policy rules.
	idapiservice	Files required to use the SOAP API.
⋆	interface	Instance-specific ticket management connectors (exit trap programs). If you installed a global *Connector Pack*, ticket management connectors are contained in the *Connector Pack* global directory.
	lib	Contains the **`pslangapi.dll`**.
	license	The license file for *Bravura Security Fabric*.
	plugin	Plugin programs executed by *Bravura Security Fabric*.

. . . continued on next page

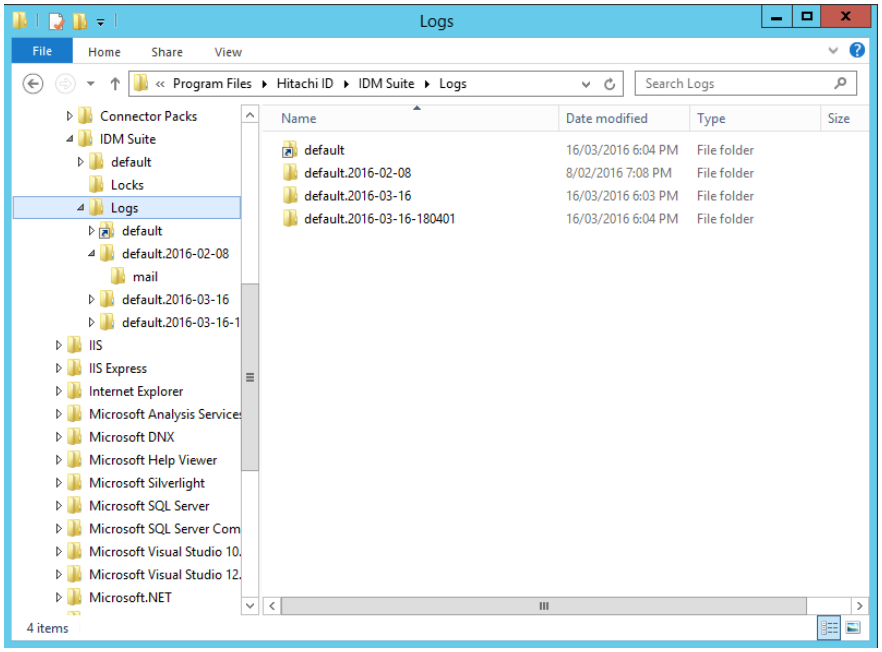
Table A.1: Instance directory files (Continued)

	Directory	Contains
	psconfig	List files produced by auto discovery and the **`idmsetup.inf`** file.
	report	Files and programs for report generation.
† ⋆	samples	Instance-specific sample scripts and configuration files. If you installed a global *Connector Pack*, connector-related sample files are contained in the *Connector Pack* global directory.
	script	Configuration files and scripts used by connectors, **`psupdate`**, plugins and interface programs.
	service	Service programs.
	sessdata	Session data. A scheduled program removed old data files nightly.
	skin	Compiled GUI files used at run-time (HTML and *.z).
	smon	Monitored session data. This location can be changed by *Recorded session management* (SMON) module options.
⋆	util	Command-line programs and utilities. If you install a global *Connector Pack*, tools related to connector configuration are located in the global util directory.
⋆	unix	The **`psunix`** archive, which is required to install the Unix Listener and supporting files on a Unix-based target system. If you installed a global *Connector Pack*, this directory is created in the *Connector Pack* global directory.
	wwwdocs	Images and static HTML pages used by *Bravura Security Fabric*.

A.1.2 Log directory

Any operation that is run by *Bravura Security Fabric* is logged. Those logs are invaluable when debugging an issue. The log directory by default is C:\Program Files\Hitachi ID\IDM Suite\Logs\. Each instance of *Bravura Security Fabric* that is installed will have at least one sub-directory within this directory.

The rotatelog scheduled job, which runs on a nightly basis, rotates the logs in to a new folder, to reduce disk space usage.



See the *Bravura Security Fabric* Documentation for more information.

A.1.3 Locks directory

Certain target systems can only be accessed serially, such as Lotus Notes. This is a limitation of the API used to access the target system. In these cases *Bravura Security Fabric* drops a *lock file* in the locks directory when an operation is being performed that should only be performed serially. For this reason the locks directory *must* be the same for all instances of *Bravura Security Fabric* that are installed on the same server.

See the *Bravura Security Fabric* Documentation for more information.

A.2 Connector pack directories and files

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

For an instance-specific connector pack, the installer, **`connector-pack-x64.msi`**, installs connectors and supporting files in:

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

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\

Connector Pack directory files describes the function of directories that are created when a *Connector Pack* is installed:

Table A.2: Connector Pack directory files

Directory	Contains
addon	Files required to target Netegrity SiteMinder systems
agent	User management connectors (agents)
design	*Connector Pack*-related files necessary to make modifications to the GUI; for example target system address help pages. See the *Bravura Security Fabric* Documentation for details.
interface	Ticket management connectors (exit trap programs)
samples	Sample scripts and configuration files
unix	The **`psunix`** archive, which is required to install the Unix Listener and supporting files on a Unix-based target system
util	Tools to support the configuration of various target systems

Glossary

Database Service

responsible for connecting *Bravura Security Fabric* to its backend database.

File Replication Service

receives data from a master instance in a replication environment, and is used in conjunction with the **`updinst`** utility to synchronize files and registry keys between multiple instances.

Local Reset Extension

Resets passwords and clears cached credentials on users' local workstations.

migration

copying configuration files and raw data from one instance to another.

Mobile Worker Service

works in conjunction with the Hitachi ID Bravura One proxy server to allow the Hitachi ID Bravura One App on mobile devices to access *Bravura Security Fabric* servers.

SKA

is a specially constructed and locked-down account defined on a network operating system or a local workstation. It is typically used to allow users, who forgot or otherwise disable their login password, access to a self-service password reset facility.

SSA

A collection of features that allow users to resolve problems with their passwords, smart cards, tokens or full disk encryption software both at the office and mobile, from any endpoint device.

upgrade

deploying a newer version of *Bravura Security Fabric* in place of an older version using **`setup`**.

Index

Hitachi ID Systems, Inc.
500, 1401 - 1 Street SE, Calgary AB Canada T2G 2J3 Tel: 1.403.233.0740 E-Mail: sales@hitachi-id.com