# *Bravura Security Fabric* Implementation: Password policies

*Hitachi ID Bravura Security Fabric* provides administrators the flexibility to configure password policies to specify the complexity of both randomly chosen and manually selected passwords.

This document contains:

## 1 Requirement

Organizations require user's passwords to meet set guildelines.

## 2 Solution

*Hitachi ID Bravura Security Fabric* can enforce a wide variety of rules as to what constitutes a valid new password. Use these rules to:

- Enforce an enterprise-wide password security policy, or different policies for groups of target systems, or for classes of user, or different managed systems.
- Ensure that passwords are strong enough to be accepted on, and compatible with, all target systems and managed systems.
- Ensure that the passwords used by console-only product administrators are strong. Console-only product administrators, such as superuser, do not have an account on any target system; their passwords are validated by *Bravura Security Fabric*.
- Control what passwords users may select when they reset their own forgotten passwords.
- Control what passwords users may select when they create new accounts.

- Generate random passwords.

  Users can select from the list of random passwords when they type new password values. *Bravura Security Fabric* selects random passwords when performing resets for managed systems.

*Bravura Security Fabric* password policy is enforced when new passwords are created, or passwords are reset (using the web interface or via transparent synchronization).

There are three default password policies used by *Bravura Security Fabric*:

- DEFAULT – applied to all users, including superusers
- PAM_DEFAULT – applied to managed system policies, when using *Bravura Privilege* features
- PERSONAL_VAULT – applied to personal vaults

These default policies are defined in *Manage the system* (PSA) module (**Policies** → **Password policies**).

For privileged access features, you can create a separate password policy for each managed system policy. Once created, it is used instead of the default policy for that group. See Modifying the password policy to learn how to do this.

# 3 Global (default) password policy

A global password policy provides the most clear and understandable experience to users.

Essentially, a global password policy strikes a bargain with users – only one password to remember, and they can easily change it without calling the help desk; in exchange the password will be more complex and they will have to change it more often.

When *Hitachi ID Bravura Security Fabric* is configured to enforce a global password policy, it will never accept or attempt to propagate a password that will not meet the strength rules of every integrated system. For instance, in the case of an organization where users may enter very long passwords on Microsoft Active Directory but only 8 characters on an OS/390 mainframe mainframe, *Bravura Security Fabric* can require that passwords be 8 characters long, at most. Alternatively, *Bravura Security Fabric* can support longer passwords, but truncate them when it updates the mainframe. Users generally prefer the maximum length rule, as it is easier to understand than automatic truncation.

In general, systems enforce one of two types of password rules:

- Complexity requirements ensure that users do not select easily-guessed passwords. Example rules are: disallowing any permutation of the user's login ID, password history, requiring mixed letters and digits, forbidding dictionary words, etc.
- Representational constraints limit what can be physically stored in a password field on a given system. Usually there are just two such rules: maximum length and allowable character set.

By combining the requirements from each system affected by a global policy, *Bravura Security Fabric* forces users to select passwords that are accepted on every system.

The alternative, of defining different password policies for every target system or for groups of target systems, is user-unfriendly. To update their passwords, users must select a system, choose a password, wait for the password update to complete, possibly re-authenticate, choose another system, choose a different password, and so on. Users must then remember multiple passwords and will continue to experience many password problems. It has been shown that users with many passwords have a strong tendency to write down their passwords.

# 4 Privileged access password policy

You can use *Bravura Security Fabric* to create credentials on target systems when managing them. The default privilegd access password policy is used for this and can be modified at **Manage the system** → **Policies** → **Password policies** → **PAM_DEFAULT**. This password policy is also used when creating a new managed system policy, which can be modified individually.

# 5 Personal vaults passphrase policy

When enabled, the *Personal vault* app allows users to store account information and passwords. This information is stored securely and can only be decrypted with the user's passphrase.

The default personal vault passphrase policy is simple; it requires a phrase of at least eight characters. Product administrators can modify the policy at **Manage the system** → **Policies** → **Password policies** → **PERSONAL_VAULT**. This policy can also be configured so that having a passphrase is optional. This is not recommended as personal vault data (excluding passwords) will not be encrypted when there is no passphrase for the personal vault.

BEST PRACTICE

A reasonable password policy, that works on most systems (but not mainframes), is as follows:

1. Have from 8 to 32 characters (32 max for compatibility with some systems)
2. Include both uppercase and lowercase letters
3. Have at least 3 letters and 1 digit
4. Not contain the profile ID or name forwards or backwards
5. Have at most 2 pairs of repeating characters
6. Not be an old password
7. Not contain spaces (again, for compatibility reasons)

Expire passwords after 80 days.

## 5.1 Multiple password policies

By default, *Hitachi ID Bravura Security Fabric* is configured to support a single, global password policy, to ensure that all new passwords are acceptable to every system. You can create additional password policies to apply to:

- Target system groups

  You may want to set up multiple *target system groups* and password policies, for example, if subsets of target systems have incompatible password strength rules, or you want a user's passwords to vary on two or more target systems.

- Users defined by user class

  You can use user classes to apply different password policies for segments of the user population on the same target system group; for example to apply stricter rules to Microsoft Active Directory administrators than to regular users on the same domain.

See Target system groups for more information about target system groups and user-class-selected policies.

### 5.1.1 The case for alternative password policies across systems

*Hitachi ID Bravura Security Fabric* allows you to create multiple alternative password policies that you can apply to subsets of target systems, defined by a target system group.

For example, you may want to create multiple policies for incompatible systems. In some cases, it is impossible to formulate a single, consistent password policy that works across two different systems. Typically this happens when one system requires strong security, and complex passwords, while another system simply cannot support complex passwords.

Examples of weak systems include legacy applications that use very short passwords or numeric PINs, voice mail passwords, and so on.

Systems with a moderate password complexity capability typically include mainframes and DBMS servers.

Systems with a strong password complexity capability typically include Novell NetWare, Windows Active Directory, LDAP directories, and modern implementations of Unix.

### 5.1.2 The case for alternative password policies across user classes

In some cases you may want to use different password policies on the same target for different users; for example administrative users on an Active Directory domain may have a stricter password policy than regular users on the same domain. You can implement this by defining user class points and password policy associations in target system groups, in addition to the default password policy for each target system group. *Bravura Security Fabric* uses these associations to find which password policy should be applied to a given user's password changes.

# 6 Use case: Configure global password policy

This use case shows you how to view the default global password policy.

### Requirements

This use case assumes that:

- *Hitachi ID Bravura Security Fabric* and *Hitachi ID Connector Pack* are installed.
- An Active Directory system has been targeted as a source of profiles.

### Review the default policy rules

To review the existing password policy rules:

1. Log in to *Bravura Security Fabric* as `superuser`.
2. Click **Manage the system** → **Policies** → **Password policies**.
3. Select ▶ the **DEFAULT** policy.
4. Click the **Password policy** tab.
5. Review the policy rules.

Note that the samples showing in the drop-down list at the top of the **Password policy** change when you update password rules. Users are also shown a list of random passwords to help them when changing passwords.

Table 1: Password strength rules

| Rule name | Type | Description |
|---|---|---|
| **have at least N characters** | Req | The smallest number of characters that a legal password must have. |
| **have at most N characters** | Req / Warn | The largest number of characters that a legal password can have. |
| **include both uppercase and lowercase characters** | Req / Warn | Enable if passwords should have both uppercase and lowercase characters. |
| **have at most N lowercase letters** | Req / Warn | The largest number of lowercase letters that a legal password can contain. |
| **have at most N uppercase letters** | Req / Warn | The largest number of uppercase letters that a legal password can contain. |
| **have at least N special characters (not letters or digits)** | Req / Warn | The smallest number of non-alphanumeric characters that a legal password must contain. Spaces are treated as non-alphanumeric characters. |

. . . continued on next page

Table 1: Password strength rules (Continued)

| Rule name | Type | Description |
|---|---|---|
| **Have at most N special characters (not letters or digits)** | Req / Warn | The largest number of non-alphanumeric characters that a legal password can contain. Spaces are treated as non-alphanumeric characters. |
| **have at least N special characters (not letters or digits) not at the beginning and end** | Req / Warn | Same as minimum non-alphanumeric characters, but not counting the first or last character of the password. Spaces are treated as non-alphanumeric characters. |
| **have at least N letters** | Req / Warn | The smallest number of letters that a password must contain. |
| **begin with a letter** | Req / Warn | Enable to require all passwords to start with a letter. Useful for compatibility with some systems. |
| **have at least N digits** | Req / Warn | The smallest number of digits that a legal password must contain. |
| **have at least N digits not at the beginning and end** | Req / Warn | Same as minimum digits, but not counting the first or last character of the password. |
| **have up to 8 characters, only @,#,$ special characters allowed (mainframe compatible)** | Req / Warn | Intended for mainframe compatibility (can have up to 8 chars; alpha/num or @$#). |
| **have password rules apply to the first N characters** | Disabled / Enabled | This forces the first N characters of the password to comply with the password rules, and only the first N characters are used to validate the password. The number of characters must be set. |
| **not be a dictionary word** | Req / Warn | The password, stripped of non-letter characters, may not match a word (consisting of two or more letters) from the dictionary (p9). For example, the passwords `word123` and `pa9sswor*d` are *not* valid. The dictionary search is case-insensitive. |
| **not be an exact dictionary word match (e.g. word)** | Req / Warn | A password may not exactly match a dictionary word consisting of four or more letters. For example, the passwords `w1o2r3d` or `word123` are valid. The password `word` is *not* valid. The dictionary search is case-insensitive. |
| **not contain an exact dictionary word match (e.g. xyzword123)** | Req / Warn | A password may not contain a dictionary word. For example, the password `xyzword123` is *not* valid. The dictionary search is case-insensitive. |
| **not contain a dictionary word (e.g. xyzw1o2r3d)** | Req / Warn | A password, stripped of non-letter characters, may not *contain* a dictionary word. For example, the password `xyzw1o2r3d` is *not* valid. The dictionary search is case-insensitive. |

. . . continued on next page

Table 1: Password strength rules (Continued)

| Rule name | Type | Description |
|---|---|---|
| **not be a dictionary word rearranged (e.g. rdow123)** | Req / Warn | A password, stripped of non-letter characters, may not be a dictionary word rearranged. For example, the password `w1o2r3d4xyz` is valid. The password `rdow123` is *not* valid. |
| **not be the profile ID or name** | Req / Warn | The user's profile ID or name may not be used as the new password. This applies to both the full name and each word in the name. |
| **not be the profile ID or name reversed** | Req / Warn | Same as above but with the letters in the name reversed. This applies to both the full name and each word in the name. |
| **not contain the profile ID or name** | Req / Warn | The user's profile ID or name may not form part of the new password. This applies to both the full name and each word in the name. |
| **not contain the profile ID or name reversed** | Req / Warn | Same as above but with the letters in the name reversed. |
| **not be the profile ID or name rearranged** | Req / Warn | Same as above but with the letters in the name rearranged in any way. This applies to both the full name and each word in the name. |
| **not contain rearranged profile ID or name** | Req / Warn | The password cannot contain the user's profile ID or name rearranged in any permutation and mixed with any number of other characters, numbers, or special characters. This is a more restrictive form than "Not a rearranged user name?". It applies to both the full name and each word in the name.<br>The length checked against the full name and each word in the name is decided by the **MIN DICTWORD LENGTH** setting in the **Manage the system** → **Policies** → **Options** menu. , and the punctuation marks like '.' are also stripped<br>For example, with user name = Bob Jones, profile ID = JonesB the following passwords will be rejected:<br>• obbonjes 1 (with the space)<br>• bsenoj2<br>• obbonjes3<br>• bbo sdfd4<br>• sdf4 snoje |

. . . continued on next page

Table 1: Password strength rules (Continued)

| Rule name | Type | Description |
|---|---|---|
| **not begin with the first N characters of the profile ID or name** | Req / Warn | The new password may not contain the specified number of characters that begin the profile ID name. |
| **require the password to be approved by this plugin** | Disabled / Enabled | An external program is called to verify that a password is acceptable. |
| **generate random passwords using this plugin** | Disabled / Enabled | Specify a plugin to generate random passwords instead of the built-in password generator. Used with **Offer the user N random passwords**. |
| **warn if the password is not approved by this plugin** | Disabled / Enabled | A warning will be generated if the password does not pass the password policy of the specified plugin. |
| **have at most N pairs of repeating characters** | Req / Warn | The maximum number of pairs of the same character appearing consecutively in new, legal password values.<br>**Note:** The total possible pairs are counted in a sequence; for example, `annno` includes two pairs of 'n's (the first two and the last two), and `annnno` includes three pairs; however `Uuno` contains zero pairs, since upper and lower cases letters are treated as different. |
| **be one of the N suggested passwords** | Req / Warn | Display some randomly-selected passwords that the user may choose as a new password value.<br>If disabled, no suggested passwords will be displayed. It is strongly recommended that this rule is set to 'Warning'. This should only be set to 'Required' in cases where corporate policy disallows non-computer-generated passwords.<br>**WARNING!:** Setting this rule to 'Required' is *not* compatible with transparent password synchronization. See Transparent password synchronization for more information. |
| **contain only characters available on a standard English (US) keyboard** | Req / Warn | The password is rejected or a warning is issued if the password contains non-printable ASCII characters. Non-printable ASCII characters can create problems with character encoding translation. The ***Password policy rules*** web form provides a link to a page that lists valid characters. |

. . . continued on next page

Table 1: Password strength rules (Continued)

| Rule name | Type | Description |
|---|---|---|
| **not have N occurrences of the same character** | Req / Warn | The password is rejected or a warning is issued if it contains any character occurring N times. N must be datafilled. |
| **not be an old password** | Req / Warn | New passwords may not be the same as old passwords for the selected targets. |
| **allow old passwords after N days** | Disabled / Enabled | Change the history rule, so that new passwords can be the same as old ones (in the history file), if they are over N days old. Ensure that this value is greater than the value of **password must be changed every N days**, if set. |
| **password must be changed every N days** | Disabled / Enabled | Prompt the user to change passwords every N days. Ensure this value is *less* than the value for **allow old passwords after N days**, if set. |
| **not be one of last N passwords** | Req / Warn | New passwords may not be the same as one of the last N passwords. |
| **be different by at least N characters from the previous password** | Req / Warn | The password is rejected or a warning is issued if the password does not contain N characters that do not already exist in the previous password. |
| **not have been changed by you in the last N hours** | Req / Warn | The password is rejected or a warning is issued if the password was changed in the last N hours. |
| **current password may be reused for password resets for N days after its first use** | Disabled / Enabled | Allow password reuse within limited days when used in conjunction with **not be an old password**. |

### About the dictionary

*Hitachi ID Bravura Security Fabric* uses a flat file, **`words.dat`** to determine if new passwords fail dictionary-based password policy rules. This file is located in \\<*instance*>\\dictionary\\. You can customize this file to suit your needs. Alternatively, you can also create a new dictionary file called **`custom.dat`** located in \\<*instance*>\\dictionary\\. If this file presents, *Bravura Security Fabric* will use this file, instead of **`words.dat`**, as dictionary. The main advantage of using **`custom.dat`** file is that this file will be retained after upgrade while a customized **`words.dat`** will be replaced by the default file.

# 7 Use case: Add a rule using a regular expression

This use case shows you how to set password strength rules using a regular expression.

Regular expressions provide a way to identify patterns of text in strings. They can be used in a number of ways in *Hitachi ID Bravura Security Fabric*, including defining password strength rules. See the appendix “Using Regular Expressions” for more information about regular expressions.

To add a rule using a regular expression:

1. At the bottom of the **Password policy** page, in the **Regular expressions** section, click **Add new. . .**
2. Enter the following information:

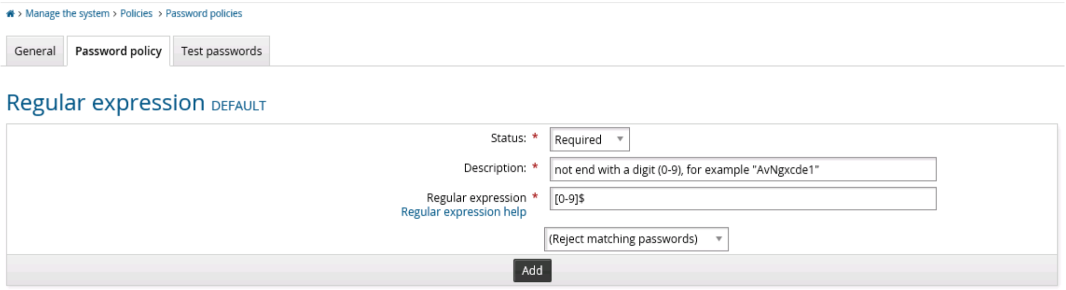
   **Status** Required

   **Description** `not end with a digit (0-9), for example "AvNgxcde2"`

   **Regular expression** `[0-9]$`

   > **Note:** When this rule is displayed, it will be prefixed with "The password must" or "Your password must". Thus the wording of your description must take this into account.

3. Leave the default "(Reject matching passwords)" value in the drop-down list.



Regular expression DEFAULT

Status: * Required

Description: * not end with a digit (0-9), for example "AvNgxcde1"

Regular expression * Regular expression help [0-9]$

(Reject matching passwords)

4. Click **Add**.

### Test the policy

1. Click the **Test passwords** tab.
2. In the **Password** field, type `2Abcdefg`.
3. Click **Test**.

   This password should be accepted.
4. This time, in the **Password** field, type `Abcdefg2`.
5. Click **Test**

   This password should be rejected because it ends in `2`.

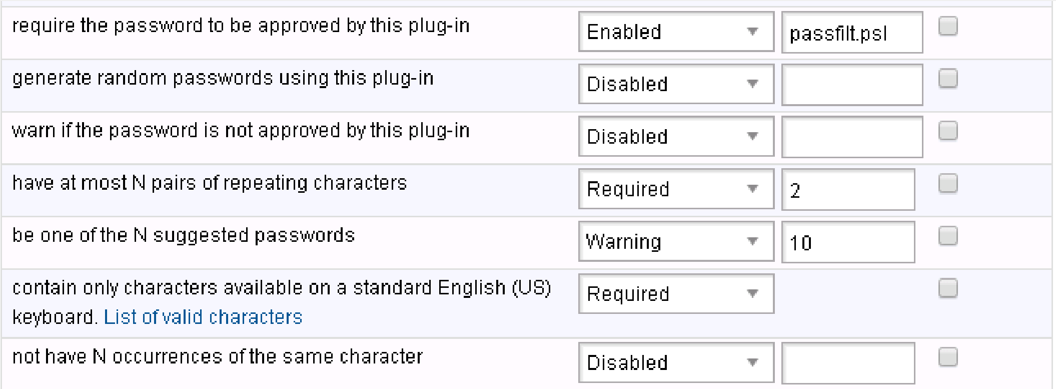
# 8 Use case: Add new rules with a plugin

*Hitachi ID Bravura Security Fabric* ships with a built-in plugin called "Password must meet complexity requirements" (**`passfilt.psl`**), that is pre-configured to match the Active Directory Default Domain Policy Password Policy. It states that passwords can:

- Not contain your user name or any part of your full name
- Contain characters from three of the following four categories:
  - English uppercase characters (A through Z)
  - English lowercase characters (a through z)
  - Base 10 digits (0 through 9)
  - Non-alphabetic characters (for example, !, $, #, %)

To add these rules with a plugin:

1. Click the **Password policy** tab.
2. At **require the password to be approved by this plugin** select "Enabled" and type `passfilt.psl`.



| | | | |
|---|---|---|---|
| require the password to be approved by this plug-in | Enabled ▾ | passfilt.psl | □ |
| generate random passwords using this plug-in | Disabled ▾ | | □ |
| warn if the password is not approved by this plug-in | Disabled ▾ | | □ |
| have at most N pairs of repeating characters | Required ▾ | 2 | □ |
| be one of the N suggested passwords | Warning ▾ | 10 | □ |
| contain only characters available on a standard English (US) keyboard. List of valid characters | Required ▾ | | □ |
| not have N occurrences of the same character | Disabled ▾ | | □ |

3. Click the **Update** button at the bottom of the **Rules** table.

### Test the policy

Now test this policy to ensure it is working as expected.

1. Click the **Test passwords** tab.
2. At **Password**, type `123Abcd`.

   This password should be accepted.
3. At **Password**, type `123ABCD`.

   This password should be rejected because it only has characters from 2 out of 4 types of required categories.

# 9 Use case: Test the random password generator

The random sample passwords are generated by default by the built-in **`randpasswd`** plugin. If your password policy includes complex regular expression rules, or if you are using your own password strength plugin, you should use a custom random password generator, rather than **`randpasswd`**. You would then use the name of the plugin to define the **use this plugin to generate random passwords** rule.

To test whether *Hitachi ID Bravura Security Fabric* can generate enough random passwords based on the current password policy:

1. Click the **Password policy** tab again.
2. Scroll to the bottom of the **Password policy** page.
3. In the **Number of random passwords to try** field, ensure the number of passwords you want to generate is `100`.
4. Click **Test**.

   *Bravura Security Fabric* displays the success rate of the generator with the password policy.

Generally, you need upwards of 30% of the passwords to be successful, in order to create enough passwords to select from. This is critical for *Bravura Privilege*, where passwords are auto-selected as part of randomization.

If your password policy includes complex regular expression rules, or if you are using your own password strength plugin, you should use a custom random password generator, rather than **`randpasswd`**. You would then use the name of the plugin to define the **use this plugin to generate random passwords** rule. See the *Bravura Security Fabric* Reference Manual for information on writing custom plugins.

# 10 Use case: Rules for password history

A particularly useful strength rule, **not be an old password** prevents or warns users against reusing old passwords. This ensures that if a user's password was divulged in the past, it will not constitute a threat in the future.

To set rules for password history:

1. Click the **Password policy** tab again for the default password policy.
2. Set **not be an old password** to "Required".
3. Set **allow reuse of old passwords after N days** to "Enabled" and type `420`.

   This value matches the default Active Directory setting.
4. Set **password must be changed every N days** to "Enabled" and type `42`.

   This value match the default Active Directory password expiry setting (see the note below).

5. Click **Update**.

> **WARNING!:** The number of days for **allow old passwords after N days** *must* be greater than the number of days for **password must be changed every N days**.
>
> The recommended setting is that N = 6 x maximum age; for example, **password must be changed every N days** set to 30 days, and **allow old passwords after N days** set to 180 days.
>
> If configured incorrectly, users are able to reset and “change” their password using their existing password.

*Bravura Security Fabric* can list users with soon-to-expire passwords based on both target system password expiry and *Bravura Security Fabric* password policies. If both target password expiry and *Bravura Security Fabric* password history are in effect, the earliest expiry time is used.

> **Note:** By default Active Directory expires passwords every 42 days, and does not allow users to use the last 10 passwords. This means users will not be able to reuse a password until the 11th reset minimum, assuming they only change their password when it expires. The setting **password must be changed every N days** only prompts users to change their passwords when they login to *Bravura Security Fabric*. For use cases where *Bravura Security Fabric* is only accessed when users lock themselves out or forget their password, this setting is not practical. This might be the case, for example, when password synchronization is configured to be triggered from Active Directory (transparent synchronization).

### Adding new rules using a white list

You can add additional rules that use a white lists of characters that must be used in a password.

To add a password rule using a white list:

1. On the **Password policy** tab, type **Valid characters** in the **White list** form below the standard strength rules; for example `aeiouAEIOU`.
2. Type the **Number of valid characters** that must be included in a password.

   This value will be included in the rule’s description.
3. Select the **Status** (required or warning).

   Depending on the status, *Hitachi ID Bravura Security Fabric* prepends “The password must” or “The password should” to the message to display to users; for example:

   ```
   Your password must:
    • have at least 3 characters in [aeiouAEIOU]
   ```
4. Click **Update**.

   *Bravura Security Fabric* automatically adds a row so that you can add another white list, or you can click **More** to add more rows.

   New passwords will be generated and tested against a combination of white lists and other rules.

To delete a white list, select the checkbox next to the list and click **Update** in the white list table.