

# **Selling and Fulfillment Foundation**

---

## **Password Policy Management**

**Release 9.0**

**March 2010**

***Sterling Commerce***  
*An IBM Company*

© Copyright 2010 Sterling Commerce, Inc. All rights reserved.  
Additional copyright information is located on the documentation library:  
<http://www.sterlingcommerce.com/Documentation/MCSF90/CopyrightPage.htm>

---

# Contents

Password Policy Overview .....	5
Guidelines for Creating Passwords .....	5
Guidelines for Controlling User Access and Login Attempts .....	5
Guidelines for Resetting Passwords .....	6
Guidelines for Defining Rules of a Password Policy .....	7
Configuring Rules for a Password Policy .....	7
Rule Definition Interfaces .....	8
Application Interfaces .....	8
Custom Interfaces .....	8
Generating Passwords .....	9
Generating Random Passwords .....	9
Generating Custom Passwords .....	9
Resetting Passwords .....	11
Configuring Secret Questions .....	11
Resetting Passwords Through E-mail .....	11
Changing User Passwords .....	12
Resetting Passwords through Other Protocols .....	12
Controlling User Access and Login Attempts .....	13
Controlling Invalid Login Attempts by a User .....	13
Controlling Incorrect Answers by a User .....	13
Blocking an IP Address After Invalid Login Attempts .....	13
Extracting an IP Address .....	14
Passing the IP Address to the Interface .....	14
Configuring a Password Policy .....	15
Defining Rules for a Password Policy .....	15
Create an Assignment Rule .....	16
Create a Login Rule .....	17
Create a Password Change Rule .....	18
Creating a Password Reset Rule .....	19
Configure a Rule to Send a Confirmation to Users on Password Reset .....	20
Create a Secret Answer Rule .....	21
Modify a Rule Definition .....	22
Modify a Rule Parameter Definition .....	22
Modify a Parameter Value .....	22
Delete a Rule .....	23
Delete a Rule Parameter Definition .....	23
Configure Password Policy Parameters .....	23
Delete a Password Policy .....	24
Configuring Questions .....	25

Create a Question . . . . .	25
Modify a Question . . . . .	25
Delete a Question . . . . .	26
Assigning a Password Policy . . . . .	27
Assigning a Password Policy to a User . . . . .	27
Assigning a Password Policy to an Enterprise . . . . .	27

**Index** **29**

---

---

## Password Policy Overview

The application provides an in-built and flexible password policy management for controlling password use and behavior. A password policy is a set of rules to define, control and manage user passwords. You can configure your own rules for the password policy, as applicable. The password policy is set at the organization or enterprise level.

The password policy broadly governs the following password characteristics:

- ◆ **Password strength:** Password strength controls the length of the password (minimum and maximum length), special characters in the password and password reuse.
- ◆ **Password generation:** Password generation controls generating a password during user creation, frequency of password expiration, failed login attempts and user roles that may affect the password policy.
- ◆ **Password reset:** Password reset controls resetting of the password through different protocols such as e-mail, SMS or any other.
- ◆ **User authentication:** User authentication includes authenticating users by using secret questions and answers whenever a user password is reset or changed.

Additionally, password policy configuration can be used to deny access to users in case of repeated invalid login attempts.

**Note:** If LDAP is used, the password management must be handled externally.

## Guidelines for Creating Passwords

A password can either be provided by the application or randomly generated by the system.

You can configure the password policy for:

- ◆ Stipulating minimum and maximum lengths for a password
- ◆ Password History - Store recently used passwords (including invalid passwords) to prevent a user from entering the same password again. The number of recently used passwords can be configured.
- ◆ Password Strength - Ability to enforce usage of special characters for a strong password generation.

## Guidelines for Controlling User Access and Login Attempts

The password policy can control:

- ◆ User Access: Each user in an organization can be associated with a password policy.
- ◆ Login Attempts: Ability to deny access to users in case of repeated failure to login. Number of incorrect login attempts allowed can be configured.
- ◆ To prevent hacking, the IP address of users can be blocked after a configured number of invalid login attempts repeatedly from the same IP.

## Guidelines for Resetting Passwords

A password can be reset when a user forgets a password or requests a password change. The password policy defines the behavior in case of password resets.

You can reset passwords by configuring:

- ◆ Secret questions and answers. Questions can be configured in the password policy for user authentication and on success, password can be reset or changed. Repeated wrong answers lock users out. Secret questions act akin to passwords for handling user authentication.
- ◆ Using protocols such as e-mail, SMS or any other protocol. You can configure any protocol as applicable.

An e-mail can be sent to users when:

- ◆ A password is changed or reset
- ◆ An answer to a secret question is given or changed.

---

## Guidelines for Defining Rules of a Password Policy

A password policy is governed by a set of rules, called *rule definitions*. Rule definitions must be configured to provide the actual values of rule parameters.

Each rule definition has an implementation class that validates the rule definition parameter values set in the configuration. You can add your own custom classes to implement the rules.

Each rule must belong to a rule type. Following predefined rule types are available in the system:

- ◆ Password policy assignment
- ◆ Login
- ◆ Password change
- ◆ Password reset
- ◆ Password secret answer

You cannot add or modify rule types. Each rule type has an associated interface, which is implemented by the class you specify.

## Configuring Rules for a Password Policy

A rule configuration consists of:

- ◆ Rule definition
- ◆ Parameter Definition
- ◆ Parameter Configuration

Each rule definition is identified by a unique indicative name. For example, PWD\_RULE\_DEFN.

A rule definition must include a class name for implementation. For example, `com.yantra.ycp.passwordpolicy.defaultimpl.passwordchange.YCPPasswordHistValidator`

Each rule can have any number of parameters. For example, you can define the following parameters for the rule, PWD\_RULE\_DEFN:

- ◆ MIN\_LEN and MAX\_LEN: They determine the maximum and minimum lengths of a given password.
- ◆ NUM\_SPL: Password must contain at least NUM\_SPL special characters.
- ◆ NUM\_ATTEMPTS: After NUM\_ATTEMPTS of invalid login attempts, user is locked out.
- ◆ DIC\_WORD: Password must not be based on a dictionary word (password history).

You must configure these parameters with actual values for the password policy. For example, MIN\_LEN=8, NUM\_SPL=3 etc.

You can configure any number of rules using the Applications Manager. For more information on configuring rules and parameters, refer to the topic “*Configuring a Password Policy*”.

---

## Rule Definition Interfaces

Each rule type has a corresponding interface associated with it. Classes specified for each rule definition implement the related interface.

The following sets of interfaces are provided:

- ◆ Application interfaces
- ◆ Custom interfaces

You can use either of these interfaces as applicable for implementing the password policy.

## Application Interfaces

The following rule definition application interfaces (corresponding to each rule type) are provided by the application:

- ◆ `IPasswordPolicyForAssignment` - This interface must be invoked when a policy is first assigned to the user.
- ◆ `IPasswordPolicyForPasswordChange` - This interface must be invoked when a password is changed.
- ◆ `IPasswordPolicyForLogin` - This interface must be invoked during login to check for login attempts.
- ◆ `IPasswordPolicyForSecretAnswers` - This interface must be invoked before login for validating answers to secret questions.
- ◆ `IPasswordPolicyForReset` - This interface must be invoked when there is a password reset request.

## Custom Interfaces

For every rule definition application interface, a corresponding custom interface is also provided. You can use your own class for implementing these interfaces.

An API template, `getUserForPasswordPolicy.xml` controls the *input.xml* for all implementations:

- ◆ `IPasswordPolicyDOM`
- ◆ `IPasswordPolicyForPasswordChangeDOM`
- ◆ `IPasswordPolicyForLoginDOM`
- ◆ `IPasswordPolicyForSecretAnswersDOM`
- ◆ `IPasswordPolicyForResetDOM`



---

## Generating Passwords

There are two ways of creating user passwords in the system:

- ◆ System generates a random password
- ◆ You can generate a custom password (A user exit `YCPGeneratePasswordUE` is provided for this purpose).

The password policy controls the following parameters:

- ◆ Generating a password during user creation
- ◆ Password length
- ◆ Frequency of password expiration
- ◆ Number of login attempts
- ◆ Checking for commonly used passwords
- ◆ Ability to store recently used passwords

The above parameters are configured as rules. For more information about rules, refer to the topic “Defining Rules for a Password Policy” .

## Generating Random Passwords

To generate random passwords for each user during user creation:

- ◆ An attribute `GeneratePassword` is provided in the `createUserHierarchy` API.
- ◆ Set the attribute to “`true`” to generate a random password for the user. A random 8-digit password is generated using a mix of characters, special characters and numbers.

Once generated, this password is available in the reset password event.

## Generating Custom Passwords

You can use the `YCPGeneratePasswordUE` user exit to generate custom passwords. A custom password can be generated when you require longer passwords, more number of special characters in the password or, any other.

The user exit is implemented only if the attribute `GeneratePassword` in `createUserHierarchy` API is set to “`true`” . If this flag is set to `true` and the user exit is also implemented, the system generates a custom password.

If the user exit is not found but `GeneratePassword` is set to “`true`”, the system generates a random password.

```
package com.yantra.ycp.japi.ue;
public interface YCPGeneratePasswordUE {
Document generatePassword(YFSEnvironment env, Document inXML);
}
```

`inXML` is the input XML template containing a list of required parameters to be passed to the user exit.

The user exit API generates a password based on the parameters and stores the result in `<User`

`GeneratedPassword="..." />`.

When a user password changes or is reset, an event `RESET_PASSWORD.ON_SUCCESS` is triggered. You can configure this event to send an e-mail to users when there is a change in the password or when a random password is generated.

---

## Resetting Passwords

Passwords can be reset following password expiry login failure due to invalid login attempts or in the case of a forgotten password. The password can be reset only if the password policy of a user allows it. For more information, refer to the topic “Resetting Passwords through E-mail” below.

Resetting a password requires user authentication, which is done using secret questions and answers. You can configure the questions as required.

An event `RESET_PASSWORD.ON_SUCCESS` is triggered whenever a random password is generated or a password is changed.

You can configure the system to send an e-mail once the password is changed. Any other protocol such as SMS can also be configured.

## Configuring Secret Questions

A secret question and answer pair can be used for effective user authentication. Each secret question is configured and set at the organization level. Application access is denied to users in case of repeated incorrect answers. Custom questions pertaining to a user can also be configured.

Answers are encrypted before they are stored.

For more information on configuring questions, refer to the topic “Configuring Password Policy”.

## Resetting Passwords Through E-mail

Use the `rule type=password reset` and the associated `IPasswordPolicyForReset` interface to reset passwords.

To reset a password:

- ◆ Invoke the API `requestPasswordReset` with the attribute `ResetType` as *Email*:

```
<ResetPassword UserKey="" ResetType="Email">
  <User Loginid=""/>
</ResetPassword>
```

- ◆ The `ResetType` is passed to the `allowPasswordReset` method of `IPasswordPolicyForReset` interface. If the `ResetType` is blank, password reset is not allowed.

The API generates a random character set and stores it as `RequestId`. The `RequestId` is passed to `RESET_PASSWORD.ON_REQUEST` event.

The event `RESET_PASSWORD.ON_REQUEST` is triggered once a password reset is requested. If the attribute `ResetType` is set to “Email” for the given user ID, then the `RequestId` is sent to the user through e-mail.

- ◆ The `changePassword` API must be called to reset or change the actual password.

## Changing User Passwords

Users are allowed to change their passwords once their password expires, which is determined by the password policy configuration. For more information on configuring the frequency of password expiration and password resets, refer to the “Configuring Password Policy”.

To change the password:

- ◆ An API, `changePassword` is provided to change the actual password. :

```
<User UserKey="" Loginid="" >
  <ResetPassword RequestId="" ResetType=""/>
</User>
```

The attribute `RequestId` is a randomly generated string that stores the password reset request ID, which is passed to the `changePassword` API to authenticate the request.

- ◆ A combination of `RequestId` and `ResetType` is used for request validation, and on successful validation, user password is changed.

## Resetting Passwords through Other Protocols

You can reset passwords by configuring other protocols such as SMS. You must configure a new password policy to allow password reset through the new protocol.

To enable this:

- ◆ Configure the `RESET_PASSWORD.ON_REQUEST` event to send a message.
- ◆ Set the `ResetType` to an appropriate protocol. For example, `ResetType=SMS`.
- ◆ Call the `requestPasswordReset` API with `ResetType=SMS`.
- ◆ The `RESET_PASSWORD.ON_REQUEST` event processes the request and sends the message to the user along with the request ID.
- ◆ Call `changePassword` API with `ResetType=SMS` and the request ID as given in the message to change the password.

**Note:** A Password Request purge agent is provided to clear the `PLT_PWD_REQ` table. For details on purge agents, refer to the Sterling Distributed Order Management: Configuration Guide

---

## Controlling User Access and Login Attempts

The password policy enables you to control:

- ◆ Number of invalid login attempts by a user
- ◆ Number of incorrect answers to questions
- ◆ Number of invalid login attempts from a particular URL (IP address)

### Controlling Invalid Login Attempts by a User

The number of login attempts allowed for a user can be configured. For more information on configuring this number, refer to the topic “Configuring a Password Policy”.

Failed login attempts are stored in the table `PLT_USER_LOGIN_FAILED`. An attribute `FAILURE_TYPE` in the table is used to determine the type of login failure.

In case of login failure, `FAILURE_TYPE` is set to “LOGIN”.

**Note:** A User Login Failed purge agent is provided to clear the `PLT_USER_LOGIN_FAILED` table. For details on purge agents, refer to the Sterling Distributed Order Management: Configuration Guide.

### Controlling Incorrect Answers by a User

By default, the system locks out a user after a number of failed answer attempts to secret questions.

In case of failed answers, the `FAILURE_TYPE` is set to “ANSWER” in the `PLT_USER_LOGIN_FAILED` table.

- ◆ The `IPasswordPolicyForSecretAnswers` interface associated with the *rule type = password secret answer* can be used to check for failed answers.
- ◆ Use the following parameters in the implementation class for validation. You can configure the values for these parameters in the password policy as required:
  - ◆ `MaxFailedAnswers`: Number of allowed incorrect answers within the interval specified in the attribute `CheckIntervalMinutes`.
  - ◆ `CheckIntervalMinutes`: Time interval (in minutes) allowed for incorrect answers, after which the answers are not validated and the user is locked out for this duration. For example, if the interval specified is 180 mins and the number of allowed wrong answers exceeds this limit, the user is locked out for 3 hours. The user can login again after the interval lapses.

### Blocking an IP Address After Invalid Login Attempts

In case of repeated login failure from a particular IP address, you can configure the system to block the IP address from accessing the application.

To block an IP, perform the following steps:

1. Extract the IP address from the header
2. Pass the IP address to the interface associated with the rule type, *password secret answer*

## Extracting an IP Address

To determine the authenticity of an IP address and whether it is using a proxy, a property is provided in `yfs.properties` file:

```
yfs.clientip.reader=<classname>
```

where `<classname>` is the name of the class that is used to read and store the client IP address from the request.

To extract an IP address, perform the following steps:

- ◆ You can provide your own client IP reader class. This property checks the proxy-specific headers and extracts the appropriate IP address as:

```
yfs.clientip.httprequest.ipHeaderAttribute
```

- ◆ An interface `YFSClientIPReader` is provided, which must be implemented in the given class, `<classname>`:

```
public interface YFSClientIPReader{
    String readClientIP(HttpServletRequest req);
}
```

## Passing the IP Address to the Interface

Perform the following steps to pass the extracted IP address to the interface:

- ◆ Use the `IPasswordPolicyForSecretAnswers` interface to check for failed answers from a particular IP.
- ◆ The IP address is available in the `onAnswerFailure` method of this interface.
- ◆ Use the following parameters in the implementation class for validation. You can configure the values for these parameters in the password policy as required:
  - ◆ `MaxFailedLogins`: Number of allowed invalid password attempts within the interval specified in the attribute, `CheckIntervalMinutes`.
  - ◆ `CheckIntervalMinutes`: Time interval (in minutes) allowed for invalid login attempts, after which the IP is blocked for this duration.

---

## Configuring a Password Policy

You can set up and configure a password policy for your organization and associate it to user accounts in the organization from the Applications Manager. For details on using the Applications Manager, refer to the *Sterling Distributed Order Management: Configuration Guide*.

**Note:** The password policy is set at the enterprise level. You can use policy configurations of any other enterprise through enterprise inheritance.

A password policy is a set of rule configurations. Each rule configuration provides values for the parameters of a rule.

Configuring a password policy broadly consists of the following functions:

- ◆ Defining rules
- ◆ Configuring rule parameters for implementation and validation
- ◆ Defining a password policy that implements the defined rules
- ◆ Setting up secret questions for user authentication

## Defining Rules for a Password Policy

You can define and configure rules for the password policy in the Application. Each rule has parameters for specifying one or more policy definitions (password change, password length or password reset).

Each rule you define must belong to one of the predefined rule types. The following table lists some of the common rules that you can create, the rule type they must belong to and the associated interface that must be implemented:

Rule	Rule Type	Interface
Login	Password Rule Type Login	IPasswordPolicyForLogin
Policy assignment	Password Rule Type Assignment	IPasswordPolicyForAssignment
Password length	Password Rule Type Password Change	IPasswordPolicyForPasswordChange
Password strength	Password Rule Type Password Change	IPasswordPolicyForPasswordChange
Password reset	Password Rule Type Password Reset	IPasswordPolicyForReset
Failed login attempts	Password Rule Type Login	IPasswordPolicyForLogin
User authentication	Password Rule Type Secret Answer	IPasswordPolicyForSecretAnswers

Each rule type has an associated interface, which is implemented by the class you specify when creating the rule. For details on implementing these interfaces, refer to the topic “Guidelines for Defining Rules and Rule Definition Interfaces”.

You can define any number of rules for a given rule type and each rule definition can contain any number of rule parameters.

A rule is configured when you:


- ◆ Create the rule.
- ◆ Define the parameters for the rule.
- ◆ Configure the rule parameters by specifying values.

The following sections explain how to create a rule belonging to each of the predefined rule types.


## Create an Assignment Rule

Assignment rules are applied when a password policy is assigned to a user or an enterprise.

To create an assignment rule, perform the following steps:

1. Open the Applications Manager and select **Password Policy Management > Password Rule Definition**. The **Password Rule Definition List** window is displayed.
2. In the list window, click . The **Rule Definition Details** window is displayed.
3. Specify the following:


Field	Description
Rule Name	Specify a name that is indicative, for the new rule. There is no restriction as to the length, case and type.
Rule Description	Provide a description for the rule.
Class Name	Assignment rules must implement the <code>IPasswordPolicyForAssignment</code> interface. Each rule definition has its own class for implementation. Specify the class to be used for validating the parameters provided in the policy configuration. This class must implement the interface associated with the specified rule type. For example, <code>com.yantra.ycp.passwordpolicy.defaultimpl.policyassign.YCPolicyAssignValidator</code>
Rule Type	Select <b>Assignment</b> from the drop-down list. The interface associated with this rule is specified in <b>Class Name</b> .

4. After you define a rule, you must define the rule parameters. Click  in the **Rule Parameters** panel. A pop-up window is displayed.
5. Specify the following:

Field	Description
Name	Specify a parameter name that is indicative. The parameter name must be unique as this value is used in the class that validates the rule.




Field	Description
Description	Provide a description for the parameter.
Data Type	Specify the data type (number, string, long etc) of the parameter.

- Click **OK** to close the pop-up window.
- Repeat steps 5-6 to add more rule parameters, as applicable.
- Click  in the **Rule Definition Details** panel to save the rule definition details.
- Proceed with configuring the policy parameters. Refer to the topic “Configure Password Policy Parameters”.


## Create a Login Rule

Login rules are applied when a user logs into the application. Login rules can be used to validate either the number of failed login attempts by a user or the number of failed login attempts from a specific URL. The login rule can also be used to check expiry of passwords.


To create a login rule, perform the following steps:

- Open the Applications Manager and select **Password Policy Management > Password Rule Definition**. The **Password Rule Definition List** window is displayed.
- In the list window, click . The **Rule Definition Details** window is displayed.
- Specify the following:

Field	Description
Rule Name	Specify a name that is indicative, for the new rule. There is no restriction as to the length, case and type.
Rule Description	Provide a description for the rule.
Class Name	Login rules must implement the <code>IPasswordPolicyForLogin</code> interface. Each rule definition has its own class for implementation. Specify the class to be used for validating the parameters provided in the policy configuration. This class must implement the interface associated with the specified rule type. For example, <code>com.yantra.ycp.passwordpolicy.defaultimpl.login.YCPFailedLoginValidator</code>
Rule Type	Select <b>Login</b> from the drop-down list. The interface associated with this rule is specified in <b>Class Name</b> .

- After you define a rule, you must define the rule parameters. Click  in the **Rule Parameters** panel. A pop-up window is displayed.
- Specify the following:


Field	Description
Name	Specify a parameter name that is indicative. The parameter name must be unique as this value is used in the class that validates the rule.
Description	Provide a description of the parameter.
Data Type	Specify the data type (number, string, long etc) of the parameter.

- Click **OK** to close the pop-up window.
- Repeat steps 5-6 to add more rule parameters, as applicable.
- Click  in the **Rule Definition Details** panel to save the rule definition details.
- Proceed with configuring the policy parameters. Refer to the topic “Configure Password Policy Parameters”.


## Create a Password Change Rule

Password change rules are applied when a password is changed by a user. Password change rules can be used to validate password length, strength, and the history that tracks previously used passwords.


To create a password change rule, perform the following steps:

- Open the Applications Manager and select **Password Policy Management > Password Rule Definition**. The **Password Rule Definition List** window is displayed.
- In the list window, click . The **Rule Definition Details** window is displayed.
- Specify the following:

Field	Description
Rule Name	Specify a name that is indicative, for the new rule. There is no restriction as to the length, case and type.
Rule Description	Provide a description for the rule.
Class Name	<p>Password change rules must implement the <code>IPasswordPolicyForPasswordChange</code> interface.</p> <p>Each rule definition has its own class for implementation. Specify the class to be used for validating the parameters provided in the policy configuration. This class must implement the interface associated with the specified rule type. For example, <code>com.yantra.ycp.passwordpolicy.defaultimpl.passwordchange.YCPPasswordHistValidator</code></p>
Rule Type	Select <b>Password Change</b> from the drop-down list. The interface associated with this rule is specified in <b>Class Name</b> .

- After you define a rule, you must define the rule parameters. Click  in the **Rule Parameters** panel. A pop-up window is displayed.
- Specify the following:


Field	Description
Name	Specify a parameter name that is indicative. The parameter name must be unique as this value is used in the class that validates the rule.
Description	Provide a description of the parameter.
Data Type	Specify the data type (number, string, long etc) of the parameter.

- Click **OK** to close the pop-up window.
- Repeat steps 5-6 to add more rule parameters, as applicable.
- Click  in the **Rule Definition Details** panel to save the rule definition details.
- Proceed with configuring the policy parameters. Refer to the topic “Configure Password Policy Parameters”.

## Creating a Password Reset Rule


Password reset rules are applied when a password is reset due to invalid password entry or password expiry. Password reset rules can be used to allow password resets for a user or an enterprise, and modes of password resets, such as, e-mail, SMS or any other protocol.

To create a password reset rule, perform the following steps:


- Open the Applications Manager and select **Password Policy Management > Password Rule Definition**. The **Password Rule Definition List** window is displayed.
- In the list window, click . The **Rule Definition Details** window is displayed.
- Specify the following:

Field	Description
Rule Name	Specify a name that is indicative, for the new rule. There is no restriction as to the length, case and type.
Rule Description	Provide a description for the rule.
Class Name	Password reset rules must implement the <code>IPasswordPolicyForReset</code> interface. Each rule definition has its own class for implementation. Specify the class to be used for validating the parameters provided in the policy configuration. This class must implement the interface associated with the specified rule type. For example, <code>com.yantra.ycp.passwordpolicy.defaultimpl.passwordreset.YCPasswordResetValidator</code>

Field	Description
Rule Type	Select <b>Password Reset</b> from the drop-down list. The interface associated with this rule is specified in <b>Class Name</b> .

- After you define a rule, you must define the rule parameters. Click  in the **Rule Parameters** panel. A pop-up window is displayed.
- Specify the following:

Field	Description
Name	Specify a parameter name that is indicative. The parameter name must be unique as this value is used in the class that validates the rule.
Description	Provide a description of the parameter.
Data Type	Specify the data type (number, string, long etc) of the parameter.

- Click **OK** to close the pop-up window.
- Repeat steps 5-6 to add more rule parameters, as applicable.
- Click  in the **Rule Definition Details** panel to save the rule definition details.
- Proceed with configuring the policy parameters. Refer to the topic “Configure Password Policy Parameters”.

## Configure a Rule to Send a Confirmation to Users on Password Reset


This section applies to Foundation only.

You can define a password policy rule to send a confirmation to the user when the user’s password is reset.

Perform the following steps:

- Open the Applications Manager and select **Password Policy Management > Password Policy Rules**. The **Password Policy Rules** window is displayed.
- Select the **Confirmation Is Required On Password Reset** check box if you want to send a confirmation to the user when the user’s password is reset.


Field	Description
Confirmation Is Required On Password Reset	Select this check box if you want to send confirmation to the user when the user’s password is reset.

- Click  to save the changes.


## Create a Secret Answer Rule

Secret answer rules are applied during user authentication, whenever a password is changed or reset. Secret answer rules can be used to validate user answers against secret questions that are defined for the organization. Secret answer rules can also be used to validate and control the number of incorrect answers allowed to secret questions.


To create a secret answer rule, perform the following steps:

1. Open the Applications Manager and select **Password Policy Management > Password Rule Definition**. The **Password Rule Definition List** window is displayed.
2. In the list window, click . The **Rule Definition Details** window is displayed.
3. Specify the following:

Field	Description
Rule Name	Specify a name that is indicative, for the new rule. There is no restriction as to the length, case and type.
Rule Description	Provide a description for the rule.
Class Name	Secret answer rules must implement the <code>IPasswordPolicyForSecretAnswers</code> interface. Each rule definition has its own class for implementation. Specify the class to be used for validating the parameters provided in the policy configuration. This class must implement the interface associated with the specified rule type. For example, <code>com.yantra.ycp.passwordpolicy.defaultimpl.secretanswer.YCPSecretAnswerIPValidator</code>
Rule Type	Select <b>Password Secret Answer</b> from the drop-down list. The interface associated with this rule is specified in <b>Class Name</b> .

4. After you define a rule, you must define the rule parameters. Click  in the **Rule Parameters** panel. A pop-up window is displayed.
5. Specify the following:

Field	Description
Name	Specify a parameter name that is indicative. The parameter name must be unique as this value is used in the class that validates the rule.
Description	Provide a description of the parameter.
Data Type	Specify the data type (number, string, long etc) of the parameter.



6. Click **OK** to close the pop-up window.
7. Repeat steps 5-6 to add more rule parameters, as applicable.
8. Click  in the **Rule Definition Details** panel to save the rule configuration details.

9. Proceed with configuring the policy parameters. Refer to the topic “Configure Password Policy Parameters”.

## Modify a Rule Definition



Once all the rules and parameters are created, they are displayed in the **Password Rule Definition List** panel.

To modify a rule:

1. Open the Applications Manager and select **Password Policy Management > Password Rule Definition**. The **Password Rule Definition List** is displayed.
2. In the list, select the rule to be modified. Click .
3. The **Rule Definition Details** window is displayed. Modify the details as required. You cannot modify the **Rule Name**.
4. Click  to save the details.



## Modify a Rule Parameter Definition



To modify a rule parameter:

1. Open the Applications Manager and select **Password Policy Management > Password Rule Definition**. The **Password Rule Definition List** is displayed.
2. Select the rule. The **Rule Definition Details** window is displayed. The **Rule Parameters** panel displays all the parameters associated with the selected rule.
3. In the panel, select the parameter to be modified. Click .
4. The **Rule Parameter Details** pop-up window is displayed. Modify the details as applicable. You cannot modify the **Parameter Name**.
5. Click **OK** to close the window.
6. Click  to save the details.

## Modify a Parameter Value


To modify a parameter value:

1. Open the Applications Manager and select **Password Policy Management > Password Policy**. The **Password Policy List** is displayed.
2. Select the policy and click . The **Password Policy Details** window is displayed, along with the associated parameters.
3. In the **Policy Configurations** panel, select the parameter to be modified. Click . The **Configuration Details** pop-up window is displayed along with the available configuration parameters.

4. In the **Configuration Parameters** panel, select the parameter to be modified. Click .
5. Modify the value as required. You cannot modify the **Parameter Name**. You can only change the parameter value.
6. Click **OK** to close the window.
7. Click  to save the details.


## Delete a Rule

To delete a rule:

1. Open the Applications Manager and select **Password Policy Management > Password Rule Definition**. The **Password Rule Definition List** window is displayed.
2. Select the rule to be deleted and click  in the panel.

## Delete a Rule Parameter Definition

To delete a parameter:


1. Open the Applications Manager and select **Password Policy Management > Password Rule Definition**. The **Password Rule Definition List** window is displayed.
2. Select the rule. The **Rule Definition Details** window is displayed. The **Rule Parameters** panel displays all the parameters associated with the selected rule.
3. Select the parameter to be deleted and click .

## Configure Password Policy Parameters

Creating the rule sets up the framework for rule parameters. Afterwards, you configure the password policy in order to specify the actual value of the parameters that are required by the rule. This process includes the following actions:



1. Define a password policy that governs the rule you want.
2. Select the rule type associated with the password policy.
3. Select the parameters for the rule.
4. Specify the parameter values.

To define a password policy and configure parameters, follow these steps:


1. Open the Applications Manager and select **Password Policy Management > Password Policy**. The **Password Policy List** window is displayed.
2. In the list window, click . The **Password Policy Details** window is displayed.

3. Specify the following:

Field	Description
Policy Name	Specify a unique name for the password policy. This policy is associated to the user account. Once the name is specified, it cannot be modified.
Policy Description	Provide a brief description of the password policy.
Policy Status	Select the policy status from the drop-down list. The available options are: <ul style="list-style-type: none"><li>◆ <b>Disabled</b> - If this option is set, the user account will not be validated against the rules defined in the policy during implementation.</li><li>◆ <b>Enabled</b> - If this option is set, the user account will be associated with the password policy and the account will be validated against all the rules in the policy.</li></ul>

4. Click  in the **Policy Configurations** panel. The **Rule Parameter Details** pop-up window is displayed.
5. Select the rule type you want for this policy. A list of rules that have been defined for this rule type is displayed in the **Rule Name** drop-down list.
6. Select the rule name for which the parameter values must be configured. A list of available parameters for this rule is displayed in the **Configuration Parameters** panel.
7. Select the parameter to be configured and click . The **Parameter Details** window is displayed.
8. Specify the following:

Field	Description
Parameter Name	The parameter name defined in the <b>Rule Parameters</b> panel is displayed here. You cannot modify the parameter name once specified.
Parameter Value	Specify the actual value for the parameter that drives the rule associated with it. This value is used by the rule definition class for validation.


9. Click **OK** to close the window.
10. Repeat steps 7-12 to configure values for other parameters associated with this rule type and name.
11. Click **OK** to close the **Configuration Details** window.
12. Click  in the **Password Policy Details** panel to save the rule configuration details.

## Delete a Password Policy

To delete a password policy:


1. Open the Applications Manager and select **Password Policy Management > Password Policy**. The **Password Policy List** window is displayed.



2. Select the policy to be deleted and click .

## Delete a Password Policy Parameter

To delete a password policy parameter:

1. Open the Applications Manager and select **Password Policy Management > Password Policy**. The **Password Policy List** window is displayed.
2. Select the policy. The **Password Policy Details** window is displayed.
3. In the **Policy Configurations** panel, select the parameter to be deleted and click .


## Configuring Questions

The system allows you to define a set of questions for user authentication during a new password request following password expiry or a password reset. The questions are set at the organization or enterprise level.


You can define any number of questions for the organization.

## Create a Question

To create a question:

1. Open the Applications Manager and select **Password Policy Management > Question**. The **Authentication Question List** window is displayed.
2. In the list, click . The **Question Details** window is displayed.
3. Specify the question and the sequence in which it should appear:

Field	Description
Question Text	Provide a question here. For example, <i>Which car do you own?</i>
Sequence	Specify the sequence in which the question must appear.



4. Repeat steps 1-4 to add more questions.
5. Click  to save the information.

## Modify a Question

Once a question is specified, it cannot be modified. You can only change the sequence in which it appears.


To modify the sequence:

1. Open the Applications Manager and select **Password Policy Management > Question**. The **Authentication Question List** window is displayed.

2. Select the question to be modified and click  . The **Question Details** window is displayed.
3. Modify the sequence as applicable.
4. Click  to save the details.

## Delete a Question

To delete the question:

1. Open the Applications Manager and select **Password Policy Management > Question**. The **Authentication Question List** window is displayed.
2. Select the question to be deleted and click .

---

## Assigning a Password Policy

A password policy can be assigned to each user or a group of users in an organization or could be set as a default at the organization or enterprise level. The password policy assigned to a user takes precedence.

Before you assign a password policy to a user, user group or an organization, you must first define and configure the password policy with rules for implementation.



**Note:** When a user's password policy is changed, the rules governing the policy take effect only when the user's password is subsequently reset or changed.

You can assign a password policy to a user, user group or an organization in the Applications Manager from the **User Details**, **Group Details** and **Organization Details** windows respectively.

### Assigning a Password Policy to a User

You can assign a password policy to a user from the **Primary Info** tab of the **User Details** window.

To assign a password policy to a user:


1. Open the Applications Manager and select **Security > Users**. The **User Search** window is displayed.
2. Select the applicable search criteria and choose . The results are displayed in the **Search Results** panel.
3. Double-click a user from the **Search Results** panel. The **User Details** window for the selected user is displayed.
4. From the **Password Policy** drop-down list, select the password policy that you want to associate with the user.
5. Click  to save the changes.


For additional information about the **User Details** window, refer to the *Sterling Distributed Order Management: Configuration Guide*.

### Assigning a Password Policy to an Enterprise

You can assign a password policy to an enterprise from the **Roles & Participation** tab of the **Organization Details** window.

To assign a password policy to an enterprise:

1. Open the Applications Manager and select **Participant Modeling > Participant Setup**. The **Organization Search** window is displayed.
2. Select the applicable search criteria and choose . The results are displayed in the **Search Results** panel.

3. Double-click an organization from the **Search Results** panel. The **Organization Details** window for the selected organization is displayed.
4. In the **Roles & Participation** tab, select the **Enterprise Attributes** tab.  
**Note:** The **Enterprise Attributes** tab is displayed depending on the roles selected in the **Roles & Participation** tab.
5. From the **Password Policy** drop-down list, select the password policy that you want to associate with the enterprise.
6. Click  to save the changes.

For additional information about the **Organization Details** window, see the *Sterling Distributed Order Management: Configuration Guide*.

## A

Application Interfaces 8

## C

Creating Passwords  
guidelines 5

Custom Interfaces 8

## I

IP Address  
extracting 14

## L

Login Attempts  
IP, blocking of 13

## P

Parameter Values  
modifying 22

Password Policy 15  
configuring 15  
deleting 24, 25  
rules 15

Passwords  
changing 12  
generating 9  
resetting 11, 12

Passwords, custom  
generating 9

Passwords, random  
generating 9

print documents  
creating 16, 17, 18, 19, 21, 25  
deleting 23  
modifying 22

## Q

Questions  
configuring 25  
deleting 26  
modifying 25

queue management 15

## R

Resetting Passwords  
guidelines 6

Rule Definition Interfaces 8

Rule Parameters  
deleting 23  
modifying 22

Rules  
configuring 7  
creating 16, 17, 18, 19, 21  
deleting 23  
guidelines 7  
modifying 22

## S

Secret Questions  
configuring 11

## U

User Access  
controlling 13