
# May 2015

# Pass-Through Authentication Guide

**May 18, 2015**

**Part Number**
**E56156-05**

Oracle Service Cloud User Guide
Part Number E56153-05

# Contents

# Pass-Through Authentication

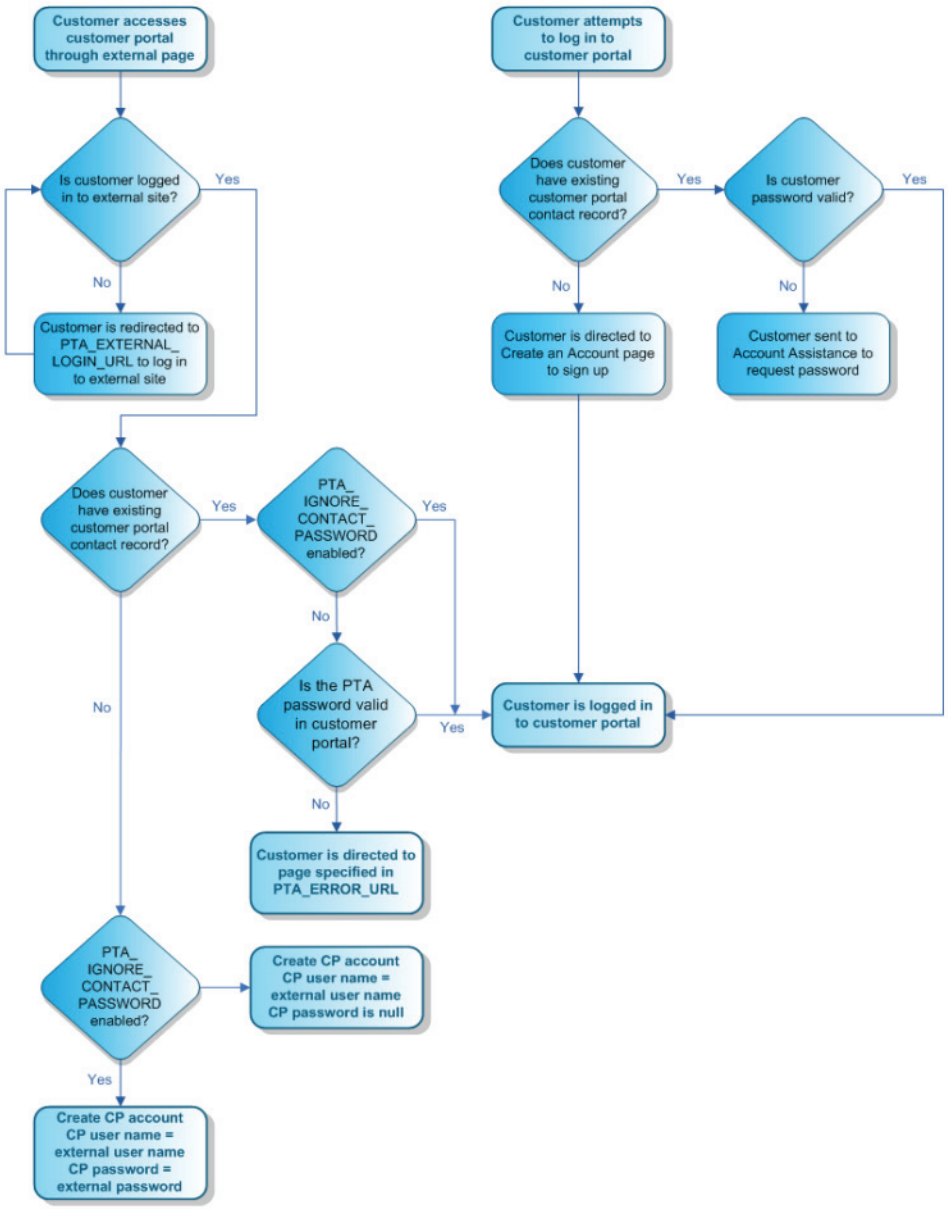
Pass-through authentication (PTA) lets you integrate the Oracle RightNow Customer Portal Cloud Service (Customer Portal) with an external customer validation source, such as your organization's website, so your customers can automatically log in to your customer portal from an external web page. The external source supplies login parameters to the customer portal by placing them in the URL of the customer portal page. This lets your customers log in to your website and then access the customer portal without requiring a second login specifically for the customer portal. Contact information is shared between the external source and the Oracle database since the customer portal uses external login information to create and update contact records.

Data encryption is available to more securely transmit customer information through the URL that accesses the customer portal, and several encryption options exist. Another PTA configuration option allows your customers to log in directly to your customer portal, in addition to logging in with pass-through authentication from your external site. You also have the option of requiring customers to log out through the external site or allowing them to log out from your customer portal.

Although contact records can be created and updated through the PTA integration, they must be deleted through the agent desktop or another integration method, such as the XML API.

**Note** Contact your Oracle account manager for assistance in customizing PTA beyond the procedures detailed in this chapter.

The following flowchart describes pass-through authentication as it is used with the customer portal.


Customer accesses customer portal through external page
Is customer logged in to external site?
Yes
No
Customer is redirected to PTA_EXTERNAL_LOGIN_URL to log in to external site
Does customer have existing customer portal contact record?
Yes
PTA_IGNORE_CONTACT_PASSWORD enabled?
Yes
No
Is the PTA password valid in customer portal?
Yes
No
Customer is directed to page specified in PTA_ERROR_URL
No
PTA_IGNORE_CONTACT_PASSWORD enabled?
Create CP account CP user name = external user name CP password is null
Yes
Create CP account CP user name = external user name CP password = external password
Customer attempts to log in to customer portal
Does customer have existing customer portal contact record?
Yes
Is customer password valid?
Yes
No
Customer is directed to Create an Account page to sign up
No
Customer sent to Account Assistance to request password
Customer is logged in to customer portal

# Configuring pass-through authentication

The first step in configuring pass-through authentication is enabling the feature. Once you have enabled PTA, you must set additional configuration settings and decide on other configuration options. The following steps provide an overview of the process, also shown in the flowchart below.

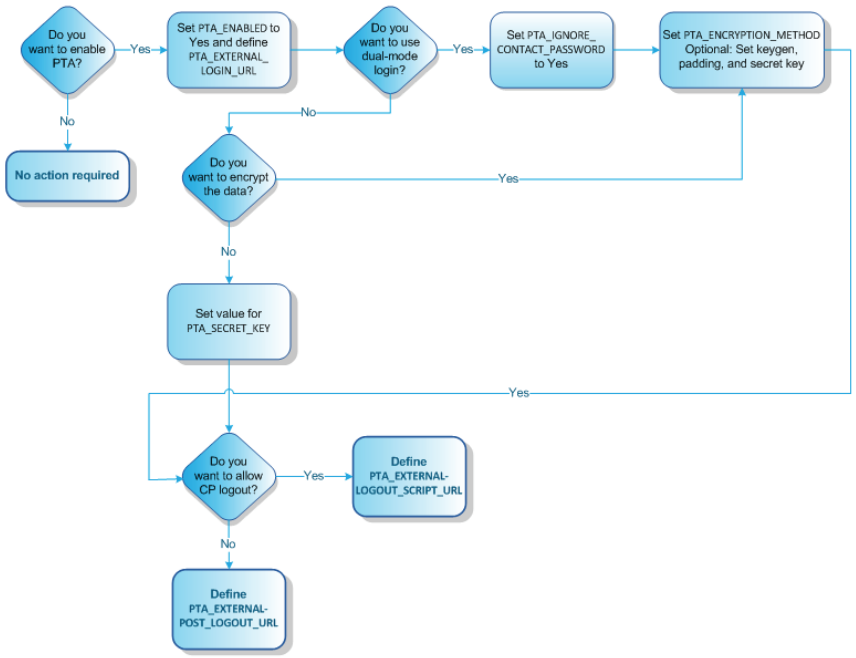
**Step 1**—Enable pass-through authentication. Refer to Enabling pass-through authentication.

**Step 2**—Define the external login page, which is used to pass customer login parameters. Refer to Defining the external login page.

**Step 3**—Decide if you want to use dual-mode login, which lets your customers access the customer portal both with pass-through authentication from your external site and by logging in directly to the customer portal. Refer to Enabling dual-mode login.

**Step 4**—Decide if you want to use data encryption and, if so, set the method and other encryption specifications. Refer to Encrypting data.

**Step 5**—Decide if you want to allow logout from the customer portal. Refer to Enabling customer portal logout.



## Editing configuration settings

A number of configuration settings must be modified to configure pass-through authentication. These settings are listed here and described in detail in the topics referenced in the table.

Table 1: Pass-Through Authentication Configuration Settings

| Setting | Description |
|---|---|
| EGW_AUTO_CONT_ CREATE | This setting, which is enabled by default, allows the creation of new contact records when an email is received from an email address that does not already exist in the database. To avoid potential login issues when using PTA, this setting should be disabled. Refer to Enabling pass-through authentication. |
| EU_CUST_PASSWORD_ ENABLED | This setting, which is enabled by default, enables the display of the contact password field on customer portal pages. This setting should be enabled when using PTA. Refer to Defining the external login page. |
| PTA_ENABLED | This setting enables the use of pass-through authentication. Refer to Enabling pass-through authentication. |
| PTA_ENCRYPTION_IV | This setting specifies the initialization vector you want to use for PTA encryption. Refer to Encrypting data. |
| PTA_ENCRYPTION_ KEYGEN | This setting specifies the keygen method you want to use for PTA encryption. Refer to Encrypting data. |
| PTA_ENCRYPTION_ METHOD | This setting specifies the encryption method you want to use for PTA logins. Refer to Enabling dual-mode login and Encrypting data. |
| PTA_ENCRYPTION_ PADDING | This setting specifies the type of padding you want to use for PTA encryption. Refer to Encrypting data. |
| PTA_ENCRYPTION_ SALT | This setting specifies the salt value you want to use for PTA encryption. Refer to Encrypting data. |
| PTA_ERROR_URL | This setting specifies the URL where customers are redirected when PTA login attempts fail. If this setting is blank, customers are redirected to the URL specified in the PTA_EXTERNAL_LOGIN_URL setting. Refer to Error codes. |

Table 1: Pass-Through Authentication Configuration Settings (Continued)

| Setting | Description |
|---|---|
| PTA_EXTERNAL_ LOGIN_URL | This setting contains the URL to a login page where customers are directed if they try to access a customer portal page that requires authentication. Refer to Defining the external login page. |
| PTA_EXTERNAL_ LOGOUT_SCRIPT_URL | This setting specifies the URL where customers are directed to log out of the customer portal. If this setting has a value, customers can log out of the customer portal. If the setting is blank, customers will not be able to log out from the customer portal since the logout widget will not display. Refer to Enabling customer portal logout. |
| PTA_EXTERNAL_POST_ LOGOUT_URL | This setting contains the URL to the page where you want to redirect customers after they log out of the external system. Refer to Enabling customer portal logout. |
| PTA_IGNORE_ CONTACT_PASSWORD | This setting specifies whether contact passwords are honored during PTA logins. Refer to Enabling dual-mode login. |
| PTA_SECRET_KEY | This setting contains the secret key used to validate login integration parameters when encryption is disabled, or the key to decode the PTA string when encryption is enabled. Refer to Encrypting data. |

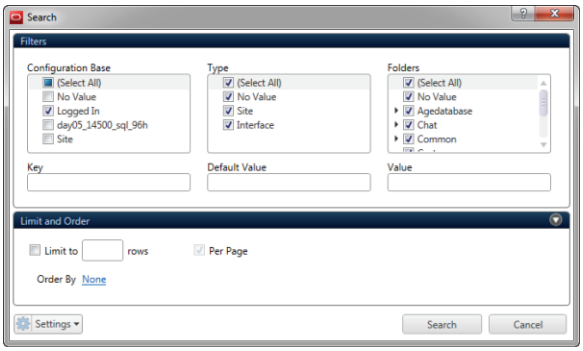
Configuration settings are modified using the Configuration Settings editor. The following procedure describes how to locate and modify configuration settings. For additional information about configuration settings, refer to the Oracle Service Cloud documentation.

*To edit configuration settings*

1 Click the Configuration button on the navigation pane.

2 Double-click Configuration Settings under Site Configuration. The Search window opens.

**Note** The Configuration Settings editor also opens but is inactive at this point. You must perform a search before any data displays.




From the Search window you can filter the configuration settings that display on the editor. Alternatively, you can click the Cancel button to bypass the Search window and perform your search using the buttons on the ribbon or the search fields on the top of the editor.

**3** Type the name of the configuration setting you want to edit in the Key field. You can type a partial name, using the percent (%) or asterisk (*) symbols as wildcard characters. For instance, you can search for configuration settings beginning with PTA_ by typing "pta_%" in the Key field.

**Tip** To display the configuration settings you need, you may need to click the Select All check box in the Configuration Base field.

**4** Select the row that displays the configuration setting you want to edit.

**5** Click the Edit Selection button on the ribbon. Either the Site or <interface name> window opens depending on the whether the configuration setting applies to the entire Oracle Service Cloud site or to specific interfaces.

Or

Double-click the setting to open it on the content pane and edit the value field.

**6** Type or select the new value. Editing options are specific to the field's data type. For example, if a setting can be enabled or disabled, a Yes/No drop-down menu displays in the Value field.

**7** To confirm a value entered in the Site or <interface name> window, click the window's OK button, and then click the Save button on the ribbon.

Or

To confirm a value entered in the editor on the content pane, click the ribbon's Save and Close button.

**Note** You may be required to log out of the administration interface and log back in for your changes to take effect.

## Enabling pass-through authentication

Pass-through authentication is enabled through the PTA_ENABLED configuration setting. When PTA is enabled, the EGW_AUTO_CONT_CREATE setting, which allows creating contact records through email, should be disabled. By default, when an email message is sent to a Service mailbox from an address that does not exist in the Oracle database, a contact record is created automatically. If this functionality remains enabled, it may pose login issues resulting from mismatched user names and passwords.

**Note** When you set EGW_AUTO_CONT_CREATE to No, you should also modify the message base NOT_REG_EMAIL_MSG to direct new customers to your website to register and create an account. For information about editing message bases, refer to the Oracle Service Cloud documentation.

## Defining the external login page

Customers who try to access your customer portal from your external page may be redirected to an external login page when they click the link. If and how this occurs depends on several factors:

- The URL of the link to the customer portal that resides on the external page
- Whether the customer is logged in to the external site
- Whether the customer portal page being accessed requires login

If the link to your customer portal on your external site passes customer information in the URL and the customer is logged in to the external site, the customer data is passed through the Customer Portal login function. The customer is then logged in to the customer portal, and the page opens. The login process is invisible to the customer, who clicks the link on the external page and sees the customer portal page open.

When the customer portal page requires login or the customer clicks the Log In link on the customer portal but is not logged in to your external site, they will be directed to the page defined by the PTA_EXTERNAL_LOGIN_URL configuration setting. Most likely, this will be the login page for your external site. You can pass next page variables and session information in this URL.

**Note** The PTA_EXTERNAL_LOGIN_URL configuration setting also accepts the error code variables you can use in the PTA_ERROR_URL configuration setting to help troubleshoot pass-through authentication login issues. When a value is entered in the PTA_ERROR_URL configuration setting, any error code variables in PTA_EXTERNAL_LOGIN_URL are ignored. Refer to Error codes.

For example, if the customer tries to open the Answers page but you have required login on that page, the URL you specify to redirect the login can contain the `%next_page%` variable. After they have logged in, your login functionality points them back to the customer portal, passing the validated customer information and returning them to the Answers page. The URL looks like this:

```
http://[your_site]/login/nextPage/%nextPage%
```

**Note** The next page parameter gets passed to the page automatically even when you do not configure it, but specifying it lets you control its placement.

You can also pass URL parameters using the following format:

```
http://[your_site]/login.php?nextPage=%nextPage%
```

The customer portal processes the customer information through its login functionality, although the customer does not see this process. If the information passed in the URL is sufficient to identify an existing contact record in the Oracle database, the customer is logged in and sees the customer portal page they originally tried to access. Any new or additional contact information that is passed through the URL is used to update the contact record.

The passed login parameters must provide data for the minimum required fields needed to log in to the customer portal (p_userid and p_passwd) or create a new contact record (p_userid, p_passwd, and p_email.addr). (In most cases, we recommend that you pass back all URL parameters to Oracle Service Cloud that the application passed during the redirection.)

**Important** If additional contact custom fields have been created on the administration interface and are required on the customer portal, the values for these fields must also be passed before a new contact record can be created.

If no contact record in the database matches the login parameters passed to the customer portal, a new contact record is created and the customer is logged in to the customer portal as the new contact. If the contact information that is passed does not contain all the fields required to create a new contact record, you can configure the customer portal to direct the customer to an alternate URL. For example, you might create a web page that lets the customer know that access is denied. Or this URL might be a form for gathering the additional required information that then re-passes the parameters to the customer portal.

**Note** URLs sent to contacts through email (for example, a link to update the incident) use the URL specified in the PTA_EXTERNAL_LOGIN_URL configuration setting.

If you pass a non-blank password via p_passwd in a PTA event and EU_CUST_PASSWD_ENABLED is disabled, the PTA event will fail. We recommend that you do not change the default value of EU_CUST_PASSWD_ENABLED, which is Yes (enabled), when using PTA.

The customer session ID can be automatically appended to the URL when the customer is redirected through the customer portal. The page specified in PTA_EXTERNAL_LOGIN_URL must be configured to accept the session ID.

## Enabling dual-mode login

The PTA_IGNORE_CONTACT_PASSWORD configuration setting lets you configure a site that accepts both PTA login and the normal customer portal login. This configuration, disabled by default, is known as dual-mode login. It lets customers create an account on the customer portal and then log in later using PTA even though the external site does not have access to the customer portal password. It is therefore possible for customers to have one password on the external site and another on the customer portal.

When PTA_IGNORE_CONTACT_PASSWORD is enabled, the customer portal does not evaluate the customer's password when the customer logs in with PTA because the external site has already authorized the login.

**Important** Encryption is enforced when dual-mode login is enabled, so you must enter a valid encryption method in the PTA_ENCRYPTION_METHOD setting or customers cannot log in and an error page will be displayed. Refer to Encrypting data.

## Creating an account with dual-mode login enabled

When PTA_IGNORE_CONTACT_PASSWORD is enabled, customers who access the customer portal directly can create an account as they normally do. The password they enter is stored in their Oracle Service Cloud contact record. They can then log in to the customer portal with the user name and password they defined.

Customers who enter the customer portal from an external site using PTA can also create an account. However, their password for the external site, which was passed through the page URL with encrypted PTA, is ignored, and the customer's contact record will have a blank password. Customers can access the customer portal with PTA through the external site, but they will not be able to log in to the customer portal directly until they have completed the account creation process through the customer portal Account Assistance page.

## Updating an account with dual-mode login enabled

When PTA_IGNORE_CONTACT_PASSWORD is enabled, customers who access the customer portal directly can update their account, including changing their password, as they normally do. To access the customer portal in the future, they must log in with their user name and new password. The new password will not affect their ability to log in through an external site with the setting enabled.

Customers who enter the customer portal from an external site using PTA can also update their account. Although their password is ignored when PTA_IGNORE_CONTACT_PASSWORD is enabled, they have been authenticated through their login to the external site. As a result, the customer portal allows them to update their account, including their customer portal password.

**Note** The contact fields on the standard Account Settings page are read-only because the *allow_external_login_updates* attribute of those input widgets defaults to false. If you want to let customers edit these fields, you must edit the input widgets to set the attribute to true. Regardless of the attribute's setting in the Username field, customers cannot change this field when PTA is enabled. See To let customers edit fields on the Account Settings page.

## Encrypting data

You can use encryption to increase the security of the customer login information passed to the customer portal pages from an external site. By default, encryption is disabled and the data received by the customer portal page URL is Base 64 encoded and then decoded. With encryption enabled, the data is still Base 64 encoded and decoded, but then it is converted to an encrypted string.

**Important** If you do not want to use data encryption, you must define a value for PTA_SECRET_KEY in order to validate login parameters. This value should be passed as a p_li_passwd parameter encoded in the PTA login string.

Four configuration settings are used to configure PTA data encryption. For the procedure to edit configuration settings, refer to To edit configuration settings.

- **PTA_ENCRYPTION_METHOD**—This setting specifies the encryption method you want to use, and is blank by default. The options are des3, aes128, aes192, and aes256.
- **PTA_ENCRYPTION_IV**—This setting lets you specify an initialization vector value to use for PTA encryption. Initialization vectors are optional, but can help you increase the security of the encryption. You can enter up to a 16-byte value, given as a hex-encoded (base 16) list of bytes. The value depends on the type of encryption specified in the PTA_ENCRYPTION_METHOD configuration setting. 16 bytes are required for aes128, aes192, and aes256 encryptions, and 8 bytes are required for des3 encryption.

  Optionally, you can enter a value of ENCODED if the decryption method expects the initialization vector to be read from the encrypted string (after the salt, if salt is used) and before the encrypted value. This option is more secure than hardcoded values if the proper cryptographically random values are sent along in the encrypted data.
- **PTA_ENCRYPTION_KEYGEN**—This setting specifies the keygen method used for PTA encryption. The default value is RSSL_KEYGEN_PKCS5_V20, and the other options are RSSL_KEYGEN_PK55_V15 and RSSL_KEYGEN_NONE.
- **PTA_ENCRYPTION_PADDING**—This setting specifies the padding method used for PTA encryption. The default value is RSSL_PAD_ANSIX923, and the other options are RSSL_PAD_PKCS7, RSSL_PAD_NONE, RSSL_PAD_ZERO, and RSSL_PAD_ISO10126.

- **PTA_ENCRYPTION_SALT**—This setting lets you specify a salt value to use for PTA encryption. Salt values are optional, but can help you increase the security of the encryption. You can enter up to an 8-byte value, given as a hex-encoded (base 16) list of bytes.

  Optionally, you can enter a value of ENCODED if the decryption method expects the salt to be read from the encrypted string before the initialization vector and the encrypted value. This option is more secure than hardcoded values if the proper cryptographically random values are sent along in the encrypted data.
- **PTA_SECRET_KEY**—This setting specifies the key used to decode the encrypted PTA string. The value is blank by default. (Do not include the value of PTA_SECRET_KEY in the string itself. The setting should be used only to encrypt the value sent.)

## Pre_pta_convert hook

You can use a pre_pta_convert hook that runs after the data has been decoded and converted into an array and before it is converted to the correct pairdata structure. This lets you modify the PTA data sent in the URL on the fly without having to decrypt and convert it yourself.

In this hook, a single parameter named "decodedData," which is in the form of an array, is passed. This example shows the data in the URL in the first line and the decoded/decrypted data passed to the hook by reference in the second line.

```
JnBfdXNlcmlkPXVzZXJuYW1lJnBfZW1haWw9dGVzdEBleGFtcGx lLmNvbQ**

[p_userid => 'username', p_email => 'test@example.com']
```

Any modifications to the array are picked up by the PTA controller when the data is converted into contact API pairs.

## Pre_pta_decode hook

The pre_pta_decode hook lets you write custom PHP code, which is executed after accepting the PTA string from the URL and before calling the login routine. After the hook runs, the p_li parameter is processed and passed to the PTA controller. The hook also passes the redirect parameter so you can modify the location where the customer is directed.

It is not possible to return a custom error message from this hook, so if you want to cover the situation of an interrupted PTA login, you need to add a header( ) location redirect and exit( ) in your hook. There are two data formatting options: string or array.

- **String format**—After the hook executes, the login integration data that was passed to the hook is evaluated to see if it is still a string. If it is, the data is assumed to be a standard base_64 encoded string. An algorithm is run to convert the string into an array of contact pairdata in key->value pairs. This allows the pass-through authentication to work as it would if no hook handler is defined and no extra encryption is added.
- **Array format**—After the hook executes, the p_li parameter is evaluated to see if it has been converted to an array. If so, the format of this array is assumed to be the contact pairdata key->value structure. The following is an example of this structure.

```
array(
    [p_passwd]=>
    [p_userid]=>username
    [p_email.addr]=>email@example.com
```

## Enabling customer portal logout

You can require customers who log in to your customer portal from an external site to also log out from the external site. If you do, the Logout link is removed from the customer portal. Customers who log out from the external site must be redirected to the *ci/pta/logout* page, where all cookies are cleared and customers are logged out of the customer portal. Customers do not see this page, but are instead directed to the page defined in the PTA_EXTERNAL_POST_LOGOUT_URL configuration setting after they log out of the customer portal. This might be your external home page, for example, or a page with a message that confirms successful logout.

You can also allow customers to log out from the customer portal even when they have logged in from an external site using pass-through authentication. The PTA_EXTERNAL_LOGOUT_SCRIPT_URL defines the page where customers are directed after logging out of the customer portal, and it allows the display of the Logout link on the customer portal. (If this setting is blank, customers cannot log out of the customer portal because no Logout link is displayed. In this case, they must log out through the external site instead.) When customers click the Logout link, they are logged out of the customer portal and directed to the external URL specified in PTA_EXTERNAL_LOGOUT_SCRIPT_URL. Your code defines what happens next. For example, you might log customers out of your external site automatically when they log out of customer portal. In that case, customers can be directed to the page specified in PTA_EXTERNAL_POST_LOGOUT_URL.

### PTA logout with communities disabled

When Oracle RightNow Social Experience communities (communities) are disabled, customers can log out from the external site or the customer portal. Logging out from the external site works normally: After customers log out from the external site, the customer portal

logout page is invoked, clearing cookies and logging out the customer, who is redirected to the page specified in PTA_EXTERNAL_POST_LOGOUT_URL. This setting must contain the fully qualified URL of the page you want to redirect customers to.

When communities are disabled and customers click the Logout link on the customer portal, they are logged out of the customer portal and redirected to the page defined by PTA_EXTERNAL_LOGOUT_SCRIPT_URL. This page is the page that logs customers out of the external site, and you can pass a source page parameter to send customers back to the page they were on when they logged out. Or you can send them to any other page.

### PTA logout with communities enabled

When communities are enabled, a third logout option is available and customers can log out from the external site, the customer portal, or the community. Logging out from the external site is the same whether communities are enabled or not, except that customers are also logged out of the communities in the process.

When customers click the Logout link on the customer portal, they are directed to the community logout script using the source page parameter in the PTA_EXTERNAL_LOGOUT_SCRIPT_URL setting. Customers are then logged out of the external site and redirected to PTA_EXTERNAL_LOGOUT_SCRIPT_URL. This setting must contain the fully qualified URL of the page you want to redirect customers to.

When customers click the communities logout link, they are logged out of the communities, redirected to a logout page for the customer portal (*ci/social/logout*), logged out of the customer portal, and then redirected to the page specified in PTA_EXTERNAL_LOGOUT_SCRIPT_URL.

**Note** The code for communities must be modified to pass the encoded URL of the page from which the customer logs out.

## Configuring the customer portal for PTA

When you implement PTA integration with the customer portal, you may want to make some changes to your customer portal pages. The following options are available.

- **Require login on customer portal pages**—You can add a login requirement to any customer portal page so that customers must be validated through the external login page before they can open the customer portal page. Refer to Requiring login for customer portal pages.

- **Edit the Your Account pages**—If you do not want customers to edit the fields on the Your Account pages, you can remove them. If you want to let them edit fields, you must edit the input field widgets on the pages containing the fields. Refer to Editing the Your Account pages.
- **Add code to handle service level agreements**—This step adds code to handle pass-through authentication when SLAs (service level agreements) are used to control access to the customer portal pages. Refer to Using PTA with SLAs.

**Important** After you have completed these steps, you must deploy the customer portal. If your profile does not have sufficient permissions to deploy the customer portal, coordinate with your administrator to arrange the deployment.

## Requiring login for customer portal pages

Presumably, your use of pass-through authentication means that you want to require login for at least one of your customer portal pages. You can add a login requirement to any customer portal page, thereby requiring customers to be validated through the external login page before accessing the customer portal page.

**Important** The CP_FORCE_PASSWORDS_OVER_HTTPS configuration setting requires all logged-in customer portal activity to occur over HTTPS. If this setting is enabled, which it is by default, pass-through authentication requests must be sent using HTTPS to ensure that information is being sent securely. For information about configuring the customer portal, refer to the customer portal documentation.

### *To require login for customer portal pages*

1 To require login on the Support Home page, edit the *home.php* file by adding `login_required="true"` to the meta tag line of the page code. Your modified code might look like the following example, where the added code is in bold text.

```
<rn:meta title="#rn:msg:SHP_TITLE_HDG#" template="standard.php"
clickstream="home" login_required="true" />
```

2 To require login on any of the following pages, edit the page's PHP file to add `login_required="true"` to the meta tag line of the page code as you did in step 1. When you do this, customers cannot view any page you require login for unless they have been logged in to the customer portal with pass-through authentication.

- *error.php*—Requiring login on this page prevents customers from seeing PTA-specific error codes.
- *answers/detail.php*—Requiring login on this page prevents customers from viewing answer details.
- *answers/list.php*—Requiring login on this page prevents customers from viewing the Answers page.
- *chat/chat_landing.php*—Requiring login on this page prevents customers from participating in a chat session with an agent.
- *chat/chat_launch.php*—Requiring login on this page prevents customers from requesting a chat session.

**3** If you have specified a value for PTA_EXTERNAL_LOGIN_URL, repeat step 1 for the *utils/login_form.php* file.

**Important** Do not edit the *utils/login_form.php* file if the PTA_EXTERNAL_LOGIN_URL value is blank.

## Editing the Your Account pages

When customers log in to the customer portal through PTA, by default they cannot edit the fields on that page. (That's because the *allow_external_login_updates* attribute of the input widgets on the page defaults to false.) If you want to retain the default behavior, you will probably want to remove the Account Settings and Change Your Password pages and any links to them. If, instead, you want to let customers change their contact information, you can add attributes to the fields you want to be editable.

### *To remove Your Account pages and links*

**1** Edit the *account/overview.php* file to remove the Settings section and the link for updating settings. (By default, the Change Your Password link is hidden when customers log in with PTA.)

**Note** If you enable dual-mode login, you might want to keep this section for customers who log in directly to the customer portal without being validated through an external source. In that case, you might want to create an alternate Your Account page.

**a** Delete the following lines of code.

```
<h2><a class="rn_Profile" href="/app/account/
profile#rn:session#">#rn:msg:SETTINGS_LBL#</a></h2>
<div class="rn_Profile">
    <a href="/app/account/profile#rn:session#">
    #rn:msg:UPDATE_YOUR_ACCOUNT_SETTINGS_CMD#</a><br/>
    <rn:condition external_login_used="false">
        <a href="/app/account/change_password#rn:session#">
        #rn:msg:CHANGE_YOUR_PASSWORD_CMD#</a>
    </rn:condition>
</div>
```

**2** Edit the *templates/standard.php* file to remove the Account Settings option from the drop-down menu on the Your Account tab of the template.

**a** Locate the following line of code.

```
subpages="#rn:msg:ACCOUNT_OVERVIEW_LBL# > /app/account/overview,
#rn:msg:SUPPORT_HISTORY_LBL# > /app/account/questions/list,
#rn:msg:ACCOUNT_SETTINGS_LBL# > /app/account/profile,
#rn:msg:NOTIFICATIONS_LBL# > /app/account/notif/list"/></li>
```

**b** Delete `#rn:msg:ACCOUNT_SETTINGS_LBL#> /app/account/profile,` from the line so the resulting code is:

```
subpages="#rn:msg:ACCOUNT_OVERVIEW_LBL# > /app/account/overview,
#rn:msg:SUPPORT_HISTORY_LBL# > /app/account/questions/list,
#rn:msg:NOTIFICATIONS_LBL# > /app/account/notif/list"/></li>
```

**3** Delete the *account/profile.php* page file.

**4** Delete the *account/change_password.php* page file.

**5** Delete the *utils/submit/password_changed.php* page file.

**6** Delete the *utils/submit/profile_updated.php* page file.

### *To let customers edit fields on the Account Settings page*

**1** Open the *account/profile.php* file.

**2** To let customers edit the First Name and Last Name fields, locate the line of code that defines the ContactNameInput widget and add the *allow_external_login_updates* attribute to it. The code will be as follows.

```
<rn:widget path="input/ContactNameInput" table="contacts" required =
"true" allow_external_login_updates="true" />
```

3 To let customers edit other input fields, locate the line of code that defines the FormInput widget for the field and add the *allow_external_login_updates* attribute to it. For example, the code to let customers change their email address will be as follows.

```
<rn:widget path="input/FormInput" name="contacts.email"
required="true" validate_on_blur="true"
allow_external_login_updates="true" />
```

**Note** Regardless of how you set the attribute for the Username field (*contacts_login*), customers cannot change this field when PTA is enabled.

## Using PTA with SLAs

In addition to requiring login on customer portal pages, you may want to restrict certain pages only to customers who have a specific type of SLA (service level agreement). The following procedure assumes you want to edit the Ask a Question page to require an SLA.

### *To use pass-through authentication with SLA permissions*

1 If you require an SLA to submit incidents, edit the meta tag line of the *ask.php* file. Your modified code might look like the following example, where the added code is in bold text.

```
<rn:meta title="#rn:msg:ASK_QUESTION_HDG#" template="standard.php"
clickstream="incident_create" login_required="true"
sla_required_type="incident" sla_failed_page="/app/error/error_id/2"
/>
```

2 If you require an SLA to request a chat session, edit the *chat/chat_landing.php* and *chat/chat_launch.php* files. Your modified code might look like the following example, where the added code is in bold text.

```
<rn:meta clickstream="chat_landing" include_chat="true"
login_required="true" sla_required_type="incident"
sla_failed_page="/app/error/error_id/2" />
<rn:meta title="#rn:msg:LIVE_CHAT_LBL#" template="standard.php"
clickstream="chat_request" login_required="true"
sla_required_type="incident" sla_failed_page="/app/error/error_id/2"
/>
```

## Using PTA when a chat request is accepted

If you use pass-through authentication (PTA) to log users into your customer portal site, you can use the same login credentials when using the syndicated ProactiveChat widget API, and when a chat request is accepted.

To use PTA, you have to subscribe to the evt_beforeDataRequest event, which is fired after a chat is accepted and after chat agent availability is checked. In the callback method for the event, you can build your PTA string similar to the way it is done within customer portal code. The encoded PTA string can then be added to the data arguments.

### *To perform PTA when a chat request is accepted*

Add code like the following to the web page containing the widget.

```
<script type="text/javascript">
    function addPta(type, args, instance){
        var ptaToken = 'thisIsYourEncryptedPTAString';
        var data = args[0];
        if(data){
            data.pta = ptaToken;


RightNow.Client.Event.evt_beforeDataRequest.subscribe(addPta);
</script>
```

In the example, the callback method addPta is called when the event is fired.

## Finalizing pass-through authentication

After you have made the changes in this section, you must stage and promote the customer portal for your changes to take effect on your production site. For the procedure, refer to the customer portal documentation.

**Important** If you have multiple interfaces, you must modify each of these files for every interface on which you use pass-through authentication with the customer portal.

# Implementing a customer login script

To develop a login-parameters integration, you must embed code within your login script to format a URL that passes data from your external validation source to the customer portal. The embedded code can be written in any scripting language, including PHP, JSP, or ASP. The login parameters from the external validation source must be placed in the customer portal URL and must be encoded using Base 64 encoding. In addition to using the Base 64 function, certain characters must also be replaced in the URL (+ becomes _, / becomes ~, and = becomes *).

**Note** You must use a login script for every link from your website to the customer portal. If contacts exit the customer portal and re-enter later in their session, they are not automatically logged in. Therefore, we recommend that all links to the customer portal contain pass-through data.

URLs use the following format:

```
http://<your domain>/ci/pta/login/redirect/answers/list/p_li/<encoded
login parameters>
```

**Note** You can replace `answers/list` with any customer portal page (for example, `home`), or use the p_next_page parameter to return customers to their original customer portal page.

The PTA controller accepts the p_li encrypted password parameter with either a GET or POST request. The POST parameter is checked only if p_li is not part of the URI (uniform resource identifier).

The parameters that can be passed to the customer portal are detailed in the following table. Each parameter represents the associated field in the *contacts* table of the Oracle database. For additional parameters that can be passed to the customer portal, refer to Additional parameters.

Table 2: Parameter-Field Associations

| Parameter | Field in *contacts* table | Notes |
|---|---|---|
| p_userid | login | This parameter is required to log in and create a contact record. It cannot be updated with pass-through authentication. |

Table 2: Parameter-Field Associations (Continued)

| Parameter | Field in *contacts* table | Notes |
|---|---|---|
| p_passwd | password | This parameter is required to log in and create a contact record or log in as an existing contact. It cannot be updated via pass-through authentication. The value can be null.<br>**Note:** This parameter is ignored when the PTA_IGNORE_CONTACT_PASSWORD configuration setting is enabled. Refer to Enabling dual-mode login. |
| p_email.addr | email | This parameter is required to log in and create a contact record. Its value must be unique. |
| p_title | title | |
| p_name.first | first_name | |
| p_name.last | last_name | |
| p_alt_name.first | alt_first_name | |
| p_alt_name.last | alt_last_name | |
| p_email_alt1.addr | email_alt1 | |
| p_email_alt2.addr | email_alt2 | |
| p_addr.street | street | |
| p_addr.city | city | |
| p_addr.postal_code | postal_code | This parameter must not contain special characters. (For example, 59715-1111 should be passed as 597151111.) |
| p_addr.country_id | country_id | This parameter must be passed as a country's ID number. Refer to Finding code numbers. |
| p_addr.prov_id | prov_id | This parameter must be passed as a state or province's ID number. Refer to Finding code numbers. |
| p_ph_office | ph_office | |

Table 2: Parameter-Field Associations (Continued)

| Parameter | Field in *contacts* table | Notes |
|---|---|---|
| p_ph_mobile | ph_mobile | |
| p_ph_fax | ph_fax | |
| p_ph_asst | ph_asst | |
| p_ph_home | ph_home | |

## Additional parameters

Besides the fields from the *contacts* table, you can also pass the following parameters to the customer portal.

Table 3: Parameter Descriptions

| Parameter | Notes |
|---|---|
| p_ccf_* | This parameter represents a contact custom field in Oracle Service Cloud. The * must be replaced with the number of the cf_id for the contact custom field. If this is a menu custom field, the numbers (not the actual text) for each menu item must be specified as the value in the integration login code. Refer to Finding code numbers. |
| p_chan_* | This parameter represents the contact's social channel. The * must be replaced with the ID number of a valid channel:<br>• 11—Twitter<br>• 12—YouTube<br>The value represents the user name for the channel. For example, if you want to pass the contact's Twitter user name, you would pass the parameter and value **p_chan_11="jane.doe"**, where jane.doe is the user name. |
| p_li_expiry | This parameter is a time stamp that defines how long the PTA login information is valid. When the time expires, the login information is no longer accepted and contacts are redirected to the page specified in the PTA_ERROR_URL configuration setting, and an error code of 16 is passed to the page. Refer to Error codes.<br>**Note:** You can generate the value using a UNIX date/time stamp generator. |

Table 3: Parameter Descriptions (Continued)

| Parameter | Notes |
|---|---|
| p_li_passwd | This parameter represents the string specified in the PTA_SECRET_KEY configuration setting.<br>**Note:** This parameter is required if the PTA_SECRET_KEY configuration setting contains a value and the PTA_ENCRYPTION_METHOD configuration setting does not contain a value. |
| p_org_id | This parameter represents an organization ID to associate with a contact. Refer to Finding code numbers.<br>**Note:** You must manually assign any service level agreements (SLA) that you want to associate with the organization, including those controlling privileged access. |
| p_state.css | This parameter represents the contact's state for Oracle RightNow Cloud Service (Service).<br>• 0—Disabled<br>• 1—Enabled |
| p_state.ma | This parameter represents the contact's state for Oracle RightNow Outreach Cloud Service.<br>• 0—Disabled<br>• 1—Enabled |
| p_state.sa | This parameter represents the contact's state for Oracle RightNow Opportunity Tracking Cloud Service.<br>• 0—Disabled<br>• 1—Enabled |

## Sample code

The following examples show how to generate a form to pass login parameters to Oracle Service Cloud. You can retain all query_string parameters and append key-value pair parameters per the following examples.

To understand these scripts better, replace certain variables with meaningful values. Replace <your_domain> with the domain name used by your Oracle Service Cloud site, <your_interface> with your interface name, and <li_password> with the string specified in PTA_SECRET_KEY.

**Caution** The following example is for illustrative purposes only and will be improperly formatted if you attempt to cut and paste directly from the following text.

```
<?
//Assumption is that user has been validated/logged in and you
//have their contact record available and can access their profile data

//Build up PTA data array
$ptaDataArray = array(
        //Common contact fields (not a complete listing, this is just a
        sampling)
        //The $contact variable is assumed to be the data of the user
        logging in
        'p_userid'     => $contact->login,
        'p_passwd'     => $contact->password, //Only needs to be sent if
        PTA_IGNORE_CONTACT_PASSWORD is disabled
        'p_email.addr' => $contact->emailAddress,
        'p_name.first' => $contact->firstName,
        'p_name.last'  => $contact->lastName,

        //Example of sending in custom field value where the custom field
        ID is 3
        'p_ccf_3'      => $contact->customField3Value,

        //Example of sending in channel field value where the channel
        field ID is 14
        'p_chan_14'    => $contact->channelField14Value
        );

//Add secret key if not using encryption
if(PTA_ENCRYPTION_METHOD config setting IS NOT set)
{
    $ptaDataArray['p_li_passwd'] = Value of PTA_SECRET_KEY config setting;
}
```

```
//Convert PTA data array to string
$ptaDataString = "";
foreach($ptaData as $key=>$value)
{
    $ptaDataString .= ($ptaDataString === "") ? '' : '&';
    $ptaDataString .= "$key=$value";
}

//Optionally encrypt data if using encryption with the method, secret key,
//padding a keygen methods. The function called here is made up. The
actual function
//will vary depending on which language you are using
if(PTA_ENCRYPTION_METHOD IS set)
{
    $ptaDataString = encryptData($ptaDataString, PTA_ENCRYPTION_METHOD,
    PTA_SECRET_KEY, PTA_ENCRYPTION_PADDING, PTA_ENCRYPTION_KEYGEN);
}

//Base64 encode the data
$ptaDataString = base64_encode($ptaDataString);

//Make sure the data is URL safe
$ptaDataString = strtr($ptaDataString, array('+' => '_', '/' => '~', '='
=> '*'));

//Specify which page to take the user to
if(%next_page% URL parameter exists)
    $redirectPage = %next_page% parameter;
else
    $redirectPage = 'home';

//Send the user to the PTA controller to log them in
header("Location: http://<Your CP site>/ci/pta/login/redirect/
$redirectPage/p_li/$ptaDataString");
exit;
```

# Error codes

When a customer attempts to log in using pass-through authentication, there are a number of factors that can cause login failure, including an invalid user name and password, a duplicate email address in the database, and problems with the PTA string, among others. To help you debug login errors or provide informational messages, customers can be redirected to a custom page displaying an error code and session information when an error occurs.

The URL of the page where customers are redirected is specified in the PTA_ERROR_URL configuration setting. To display an error code, the URL must be appended with the `%error_code%` variable. Session information can be provided in the form of a base64 encoded string if the `%session%` variable is also appended. Session information displays only if login tracking cookies are disabled on the customer's computer.

**Important** The PTA_ERROR_URL configuration setting is blank by default. If you do not specify a value for the setting, customers are redirected to the URL in the PTA_EXTERNAL_LOGIN_URL configuration setting when a login error occurs. If you choose not to use the PTA_ERROR_URL configuration setting, you can append the `%error_code%` and `%session%` variables to the end of the URL specified in PTA_EXTERNAL_LOGIN_URL. Refer to Defining the external login page.

For example, assume that you set the configuration setting value to:

```
http://[your_site]/my_login_error_page.php/%error_code%
```

If login fails because the password exceeds the 20-character limit, the URL that is returned is:

```
http://[your_site]/login/nextPage/home/error/15
```

The error code, 15 in this example, lets you know what caused the failure.

The error codes displayed in the URL when an error occurs are described in the following table. These codes are the same whether the `%error_code%` variable is used in the PTA_ERROR_URL or the PTA_EXTERNAL_LOGIN_URL configuration setting.

Table 4: Error Code Descriptions

| Error Code | Description |
|---|---|
| 1 | The PTA string parameter was not found. This parameter contains all of the encoded PTA information in the URL, so it must be present to log in. |

Table 4: Error Code Descriptions (Continued)

| Error Code | Description |
|---|---|
| 2 | PTA information after pre_pta_decode hook was not in the correct format. A string or an array was expected, but something else was received. |
| 3 | PTA string could not be Base 64 decoded. There was an error within the string that caused the decoding process to fail. |
| 4 | One of the PTA string parameters was not well formed. For example, it did not contain "p_" or was missing an "=" separator between the key and value. |
| 5 | The p_userid string was passed in, but it did not have a value. This pair is required for login. |
| 6 | The value of the p_li_passwd pair was incorrect. This error applies only if no value is set for PTA_ENCRYPTION_METHOD. |
| 7 | The specified credentials were invalid. |
| 8 | Login failed because PTA is not enabled for the interface. |
| 9 | Data decryption failed. |
| 10 | Login failed because PTA_ENCRYPTION_METHOD does not contain a valid value. |
| 11 | Login failed because PTA_ENCRYPTION_PADDING does not contain a valid value. |
| 12 | Login failed because PTA_ENCRYPTION_KEYGEN does not contain a valid value. |
| 13 | Login failed because PTA_IGNORE_CONTACT_PASSWORD is enabled, but no encryption scheme has been set in PTA_ENCRYPTION_METHOD. |
| 14 | Login failed because the format of the data after the pre_pta_convert hook was not an array. |
| 15 | Login failed because the password exceeded the 20-character maximum length. |
| 16 | Login failed because the PTA token expired and is no longer valid. A new token must be generated to authenticate the customer. |

Table 4: Error Code Descriptions (Continued)

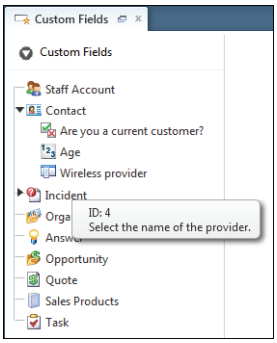
| Error Code | Description |
|---|---|
| 17 | Login failed because two or more email addresses have the same value. |

# Finding code numbers

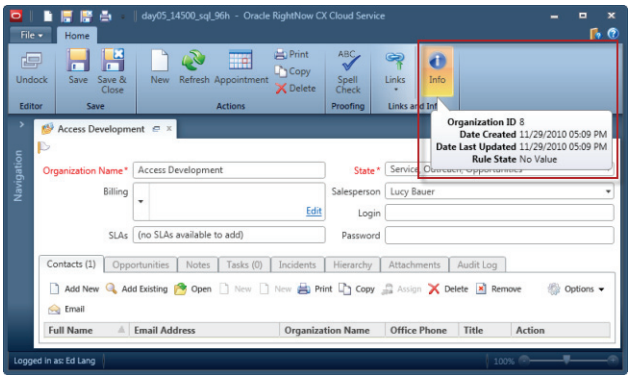
You need to use code numbers (ID numbers) in PTA to specify parameters such as countries, provinces, custom fields, and organization IDs. The administration interface provides three ways to look up the codes for these types of fields: hovering over the field name, displaying report IDs in the Reports explorer, and using the NamedID helper object.

## Viewing code numbers for fields

You can look up many of the code numbers you need by simply hovering over a country, service product, service category, incident disposition, SLA, or custom field on the administration interface. The following figure shows the code number for a custom field.




Many records and items contain an Info button on the Home tab of the ribbon. When you click the Info button, record details are displayed, including the record ID number. The following figure shows the details displayed when you hover over the Info button while editing an organization. In this example, the organization ID is the code number.




## Finding report IDs in analytics

When exploring reports, you can view the report ID (ac_id) by displaying the ID column in the explorer details. For information about customizing explorer details, refer to the Analytics documentation, available here.

## Using the NamedID helper object

You can also use the NamedID helper object to find the ID of an item and return the value by email or in a variable used later in your code. For information about using the NamedID helper object, refer to the Oracle Service Cloud Connect Web Services for SOAP Developer Guide or the Oracle Service Cloud Connect PHP API Developer's Guide.