

Reusing Components Across Letters

To configure components, you must have one of the following roles:

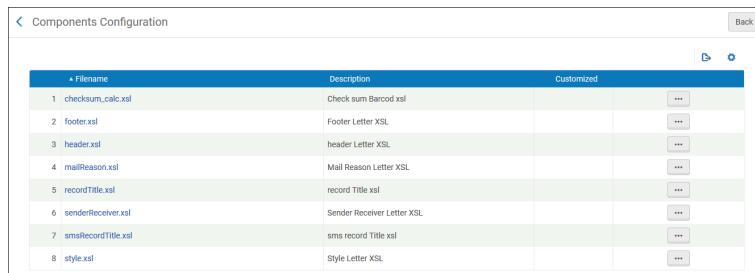
- General System Administrator
- Letter Administrator

Each letter can contain common components that are used across letters, such as headers, footers, etc. This enables you to define (and update) the common components in one place. For example, you can define a common header across all letters, that includes a corporate logo. In the future, if you change the formatting of the logo (such as its alignment) in the Components Configuration page, that change is automatically applied to all the letters that use that component. For information on configuring letters in general see [Configuring Rapido Letters](#).

Changes made to components affect all letters. It is therefore recommended to review changes carefully before saving.

Accessing the Components Configuration Page

The Components Configuration page ([Configuration > General > Letters > Components Configuration](#)) enables you to customize templates and other components shared between letters. Like letters, components are made up of labels (strings of static text) and the template (XSL style sheet for formatting and dynamic data).



Filename	Description	Customized	
1 checksum_calc.xml	Check sum Barcode xml		...
2 footer.xml	Footer Letter XSL		...
3 header.xml	header Letter XSL		...
4 mailReason.xml	Mail Reason Letter XSL		...
5 recordTitle.xml	record Title xml		...
6 senderReceiver.xml	Sender Receiver Letter XSL		...
7 smsRecordTitle.xml	sms record Title xml		...
8 style.xml	Style Letter XSL		...

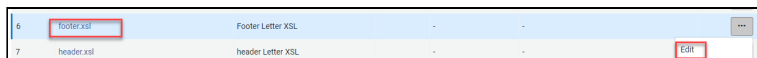
Components Configuration

Configuring Component Labels

You can use the component labels to edit a letter without needing to edit the actual XSL template. For example, if you want the letter to state "Contact the Library" instead of "Contact Us", you can change that in the labels.

How to update a label:

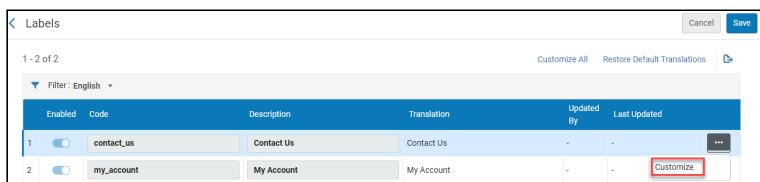
1. In the Components Configuration page, select **Edit** on the row action list of the component.



2. Select the **Labels** button. Note that if the component does not have a label, the **Labels** button is disabled.



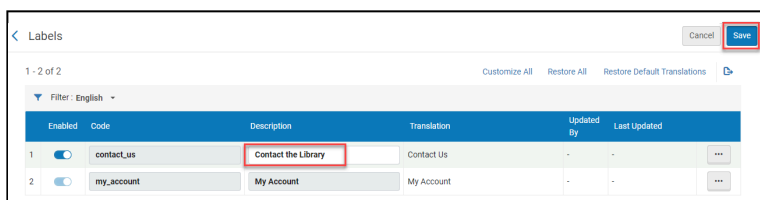
3. Select **Customize** for the label to edit.



4. Update the label as needed. You also have the following options:

- **Language change** - If another language is configured, the **Translation** field opens for editing.
- **Restore** - If the line was modified, the **Enabled** and **Description** fields are restored to their default.
- **Restore Translation** - If another language is configured and the **Translation** label was modified, the field is restored to the default.

5. When done, select **Save**.



6. You can enable or disable the label by selecting the toggle in the **Enabled** column.

Customizing Component Templates

You can view and edit the template (XSL style sheet) that is used to format the component and determine which fields and text appear in the letter. Configure the template to customize what labels and data appear in the letters, as well as how they display. In addition, you can use XSL conditional statements to make decisions about what appears in a letter, and even whether the letter is sent. For an example, see [How to use the if condition to configure a letter to not be sent under certain circumstances.docx](#).

To customize a component's template (XSL style sheets):

1. On the Components Configuration page, select **Edit** on the row actions menu for the relevant component.
2. Edit the information in the XSL pane as needed.



The available fields on this page are in the labels associated with the component and the XML fields that are generated by Rapido.

- Use the substring and string-length functions to output the last four digits of a code:

```
<b>@@requested_for@@ :*****
<xsl:value-of select="substring (notification_data/user_for_printing/
identifiers/code_value/value, string-length(notification_data/user_for_printing/
identifiers/code_value/value) - 3)"/></b>
```

- If the following snippet follows the address line in the `SenderReceiver.xml` file, you can ensure that the user's phone number is included in all the letters that use this template.

```
<tr>
<td><xsl:value-of select="notification_data/user_for_printing/phone"/></td>
</tr>
```

- Have the name of the style sheet appear as hidden text in the output by adding the following after the `<html>` tag.

```
<html> <xsl:attribute name="data-filename"> <xsl:value-of
select="/notification_data/general_data/letter_type"/> </xsl:attribute>
```

The output will include something like:

```
<html data-filename="FulLoanReceiptLetter"> ...
```

3. You can use **Ctrl+Shift+f** to implement Prettyprint (convert and present the XSL source code with line breaks and indentations to make the code easier to read).
4. Select **Save Draft** to save your customizations as a draft. Changes done in the draft will not be applied to the component until selecting **Save**. The draft watermark displays in the XSL pane while in draft mode. Note that once Save is selected, the changes will be applied to all letters using this component. Select **Remove Draft** to erase the draft and display the last saved XSL. Select **Restore** to restore the XSL to its original state. Note that while a draft is active, the **Edit Draft** and **Remove Draft** options are available in the row actions menu for the component.
5. To preview the changes, view a letter that uses this component.
6. To preview changes that were made to a **component draft**:
 - i. , Navigate to a letter that uses this component.
 - ii. In the letter, update the XSL by adding `_DRAFT` to the relevant `<xsl:include href="" />` statement, as shown in the following example:

```
<xsl:include href="footer.xsl_DRAFT"/>
```

- iii. The preview pane will now display the draft component (the updated footer in this example), so that you can review the changes.
- iv. When the changes are complete, remove the `_DRAFT` suffix from the include statement:

```
<xsl:include href="footer.xsl"/>
```

7. Select **Save** to apply the changes to the component i.e., override the current XSL with the updates. This also removes the draft watermark. The changes are now applied to all the letters using this component. Note that after saving, the letter remains in the preview window. To return to the full list of letters, select **Cancel**.

Component List

The following table lists the components available in Rapido.

Letter Components

Letter	Description						
checksum_calc.xsl	For internal use						
footer.xsl	<p>Defines the text used in the footer for all letter emails. Define the following in General > General Configuration > Other Settings:</p> <table border="1"> <thead> <tr> <th>Parameter Value</th> <th>Parameter Key</th> </tr> </thead> <tbody> <tr> <td>contact_us</td> <td>email_contact_us</td> </tr> <tr> <td>my_account</td> <td>email_my_account</td> </tr> </tbody> </table>	Parameter Value	Parameter Key	contact_us	email_contact_us	my_account	email_my_account
Parameter Value	Parameter Key						
contact_us	email_contact_us						
my_account	email_my_account						
header.xsl	Defines the text (such as your institution's logo, the letter name, and the date) used in the header for all letter emails.						
mailReason.xsl	<p>Defines the greeting (such as Dear Sir/Madam) used in most letter emails. The following would print Dear John Smith for a user named John Smith:</p> <pre><xsl:template name="toWhomIsConcerned"> <table cellpadding="5" cellspacing="0" border="0"> <tr> <td> <xsl:for-each select="notification_data"> <h3>@@dear@@ &#160;<xsl:value-of select="receivers/receiver/user/first_name"/> &#160;<xsl:value-of select="receivers/receiver/user/last_name"/></h3> </xsl:for-each> </td> </tr> </table></pre>						
recordTitle.xsl	Defines the text used for bibliographic details in letter emails.						
senderReceiver.xsl	Defines the address information for all letter emails. This component uses <code><xsl:value-of select="notification_data/user_for_printing/name"/></code> , which is populated according to the						

Letter	Description
	<p>configuration in User Name Display. However, user_for_printing also includes the name and preferred name values, so it can be configured differently. For example, from the activity report XML's user_for_printing:</p> <pre data-bbox="358 302 1507 814"> <first_name>Ford</first_name> <identifiers><code_value><code>Primary Identifier</code> <value>fprefect</value> </code_value> <code_value><code>BARCODE</code> <value>11223344</value> </code_value> </identifiers> <last_name>Prefect</last_name> <middle_name></middle_name> <name>Prefecta, Forda</name> <phone></phone> <postal_code></postal_code> <preferred_first_name>Forda</preferred_first_name> <preferred_last_name>Prefecta</preferred_last_name> <preferred_middle_name></preferred_middle_name> </pre>
SMSRecordTitle.xsl	Defines the text used for bibliographic details in SMS letters.
style.xsl Letter	Defines the CSS used for all letter emails.