|  |  |
| --- | --- |
| **How to change the Dear Mr./Mrs. to Dear Mr. or Dear Mrs. depending on user gender** |  |



In this document we will demonstrate how to change the greeting of a letter in accordance with the defined gender in the user record.

* We will use Mr. for Male.
* We will use Mrs. for Female.
* We will use the default Mr./Mrs. when no gender is defined.

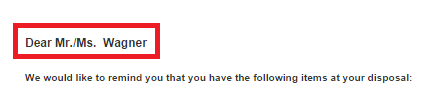
In this document we will change the xsl of the letter letter “Ful User Borrowing Activity Letter”

We will focus on the change to the xsl.

Regarding how to access and change the xsl see [Letters - Letter Configuration in Alma.pptx](https://knowledge.exlibrisgroup.com/@api/deki/files/74499/Letters_-_Letter_Configuration_in_Alma.pptx)

**ONE**

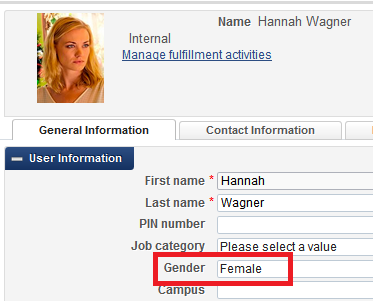
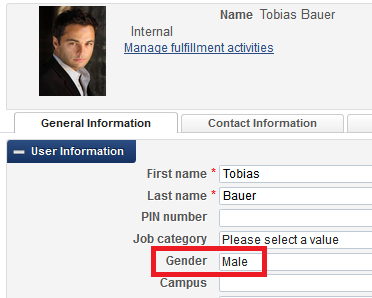
When the borrowing activity letter is sent it appears as follows by default

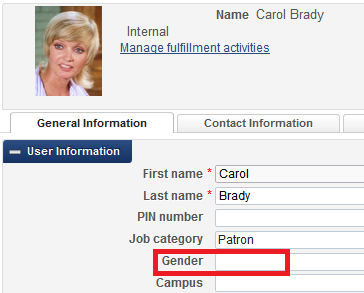


**TWO**

The xsl of the letter does not look at the gender of the users. It is not taken into account

* Hannah Wagner has gender Female
* Tobias Bauer has gender Male
* Carol Brady has gender empty



**THREE**

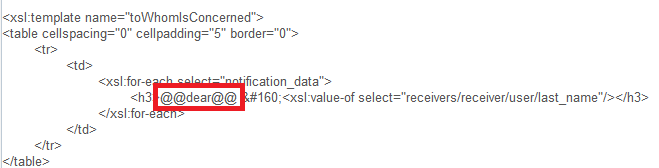
Let’s begin by seeing where the “Dear Mr./Mrs.” Is coming from. If we configure “Ful User Borrowing Activity Letter” we see this:



This means that the greeting “Dear Mr./Mrs.” Is coming from a template called “toWhomIsConcerned” which is defined in mailReason.xsl

**FOUR**

First we can go look at the xsl of “mailReason.xsl Letter” and we see the call to “Dear Mr./Mrs.” It is in @@dear@@ (as stated abive see Letters - Letter Configuration in Alma.pptx for further details)



The @@dear@@ takes the label “Dear Mr./Mrs.”. Then it gives a non-breaking space (&#160;) and then it gives the last name of the user.

**FIVE**

We want to make a condition that it will be

* “Dear Mr. …” if the user gender is male
* “Dear Mrs. …” if the user gender is female
* Leave it as is if the gender is empty

**Therefore we need to see if the gender appears in the raw xml. If it does appear we need to see how. And then we need to make a condition.**

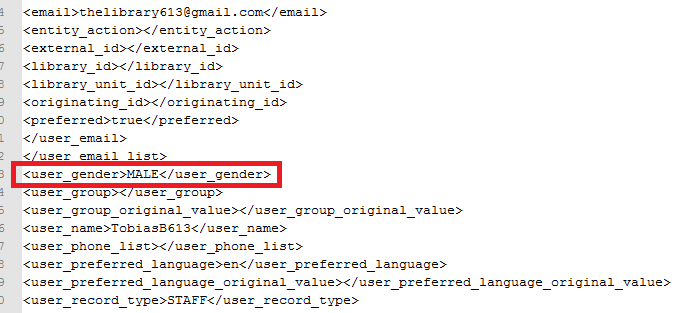
Note that there are several ways to do this in the xsl and here we will do it the simplest way though not necessarily the shortest way.

**SIX**

The gender does appear in the raw xml as follows:

It is in /notification\_data/receivers/receiver/user/user\_gender

And appears as <user\_gender>MALE</user\_gender>



**SEVEN**

Now we need to put it in the “Mail Reason Letter XSL” with the condition

* “Dear Mr. …” if the user gender is male
* “Dear Mrs. …” if the user gender is female
* Leave it as is if the gender is empty

Before we make the actual condition let’s leave the current condition and add our own and make sure the gender appears. We can do this by adding the red section below. The blue section below already existed.

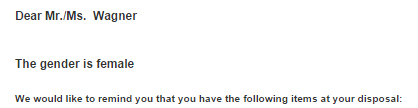
We basically do a test by the gender and printout what it is. This is just a test before we make the real change.

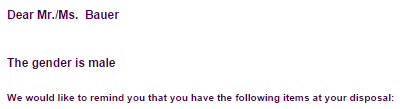
We make a condition according to the **receivers/receiver/user/user\_gender** field

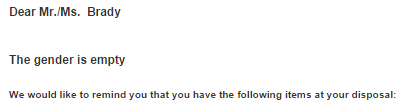
|  |
| --- |
| **<xsl:template name="toWhomIsConcerned">**  **<table cellspacing="0" cellpadding="5" border="0">**  **<tr>**  **<td>**  **<xsl:for-each select="notification\_data">**  **<h3>@@dear@@ &#160;<xsl:value-of select="receivers/receiver/user/last\_name"/></h3>**  **</xsl:for-each>**  **</td>**  **</tr>**  **</table>**  **<table cellspacing="0" cellpadding="5" border="0">**  **<tr>**  **<td>**  **<xsl:for-each select="notification\_data">**  **<xsl:if test="receivers/receiver/user/user\_gender='FEMALE'">**  **<h3>The gender is female</h3>**  **</xsl:if>**  **<xsl:if test="receivers/receiver/user/user\_gender='MALE'">**  **<h3>The gender is male</h3>**  **</xsl:if>**  **<xsl:if test="receivers/receiver/user/user\_gender=''">**  **<h3>The gender is empty</h3>**  **</xsl:if>**  **</xsl:for-each>**  **</td>**  **</tr>**  **</table>**  **</xsl:template>** |

**EIGHT**

Now test it by sending a letter to each of the above three patrons (male, female and no gender) and see if we get the correct “printout” according to our xsl.

****

****

****

**NINE**

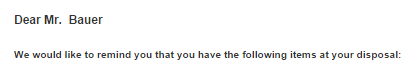
Now we can set it up for real. We will comment out the original table and make our own table.

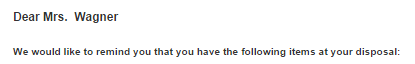
Our table will do “Mr.” for male,” Mrs.” For female, and leave as it was for an empty gender:

|  |
| --- |
| <!--  <table cellspacing="0" cellpadding="5" border="0">  <tr>  <td>  <xsl:for-each select="notification\_data">  <h3>@@dear@@ &#160;<xsl:value-of select="receivers/receiver/user/last\_name"/></h3>  </xsl:for-each>  </td>  </tr>  </table>  -->  <table cellspacing="0" cellpadding="5" border="0">  <tr>  <td>  <xsl:for-each select="notification\_data">  <xsl:if test="receivers/receiver/user/user\_gender='FEMALE'">  <h3>Dear Mrs. &#160;<xsl:value-of select="receivers/receiver/user/last\_name"/></h3>  </xsl:if>  <xsl:if test="receivers/receiver/user/user\_gender='MALE'">  <h3><h3>Dear Mr. &#160;<xsl:value-of select="receivers/receiver/user/last\_name"/></h3></h3>  </xsl:if>  <xsl:if test="receivers/receiver/user/user\_gender=''">  <h3>@@dear@@ &#160;<xsl:value-of select="receivers/receiver/user/last\_name"/></h3>  </xsl:if>  </xsl:for-each>  </td>  </tr>  </table> |

**TEN**

Let’s test it. Works perfectly

****

****

****