

Appian Step-by-Step #4

Exercise to Accompany
Expressions: Transform Your Data

The Appian Step-by-Step series consists of 12 exercises that accompany the courses in the Appian Developer learning path. Exercises build upon each other. Complete exercises in order and keep the app and all objects until you are done with the project.

- 1** Welcome to the Appian Developer Learning Path
- 2** Create an Application
- 3** Manage Users and Groups
- 4** **Expressions**
- 5** Records Part 1: Accessing Your Data
- 6** Records Part 2: Record Type Relationships
- 7** Query Entities
- 8** Process Modeling 101: Part 1
- 9** Process Modeling 101: Part 2
- 10** Reports
- 11** Sites
- 12** Task Report

Exercise 4: Expressions

Create Constants

Create Expression Rule

3

3

3

Notice of Rights

This document was created by Appian Corporation, 7950 Jones Branch Dr, Tysons, Virginia 22102. Copyright 2022 by Appian Corporation. All rights reserved. Information in this document is subject to change. No part of this document may be reproduced or transmitted in any form by any means, without prior written permission of Appian Corporation. For more information on obtaining permission for reprints or excerpts, contact Appian Training at academyonline@appian.com.

Exercise 4: Expressions

In this exercise, you will create two types of Rule objects that you need for your application. Specifically, you will create six constants and one expression rule.

Create Constants

A constant is a literal value that can be called from any expression, and they can be reused across multiple objects throughout your application. In this section, you will create constants that point to the groups and folders you just created. Once you create a constant that points to a group or folder, you will be able to call it from any expression. Follow the steps below to create a constant.


1. From within the **AX Constants** folder, click **New**, and select **Constant** from the dropdown menu.
2. Type **AX_ADMINISTRATORS_POINTER** for the **Name**, and add a description: "A pointer to the AX application administrators group."
3. In the **Type** field, select **Group**.
4. In the **Value** field, type **AX Administrators**, and then click **Create**.
5. In the same manner, create constants for the remaining four groups: **AX All Users**, **AX Supervisors**, **AX Registrars**, and **AX Maintenance**.
6. Now, you will create a constant that points to the AX Documents folder. Click **New**, and select **Constant** from the dropdown.
7. Name the constant **AX_DOCUMENTS_FOLDER_POINTER**, and provide a description: "Points to the AX Documents folder."
8. Select **Folder** for the **Type** field, **AX Documents** for the **Value** field, and **AX Constants** for the **Save In** field.
9. Click **Save**.

Create Expression Rule

An expression rule is a statement that evaluates to return a value. In this section, you will create an expression that will help format a user's name. This expression will be used in a later exercise to help you format the name field correctly. Follow the steps below to create this expression rule.

1. From within the **AX Expressions** folder, click **New**, and select **Expression Rule** from the dropdown menu.
2. Type **AX_DisplayUser** for the **Name**, and add a description: "Formats a user's name for the AX app." Click **Create**.

3. In the **Rule Inputs** pane, click the **plus sign** to add a rule input:
 - **user** (Text)

RULE INPUTS 					
Name	Type	Array			
user	 Text 	<input type="checkbox"/>			

4. Enter the following expression into the **Expression Editor** on the left:

```
proper(  
  if(  
    isnull(ri!user),  
    "",  
  
    isusername(ri!user),  
    user(ri!user, "firstName") & " " & user(ri!user,  
    "lastName"),  
  
    joinarray(split(ri!user, "."), " ")  
  )  
)
```

This expression first checks whether the user's username is null or not. If the user's username is not null, the expression then checks whether the input is an active user or not. Since the process for formatting users and non-users is different, the last two lines in the expression are both ways to reformat the input as First and Last Name.

5. Click the **Format** icon  in the toolbar to automatically format your expression.

6. Click **Save Changes** (or press ctrl/cmd + s). Test this expression by typing your ACE email (wrapped in quotes) in the **Test Inputs Expression** field. Click **Test Rule** to see the formatted name:

The screenshot shows the Appian test rule configuration interface. At the top, there are tabs for 'Ad Hoc Test' and 'Test Cases (0)'. Below this is a section titled 'Test Inputs' which contains a table with three columns: 'Rule Input Name', 'Expression', and 'Value'. The table has one row with the following data:

Rule Input Name	Expression	Value
user (Text)	1 "test.email@gmail.com"	"test.email@gmail.com"

Below the table, there is a 'Save as Test Case' link and a 'TEST RULE' button. The next section is 'Test Output', which shows 'Time 11 ms (View Performance)' and 'Type Text'. Under 'Value', there are three radio buttons: 'Formatted' (selected), 'Raw', and 'Expression'. The output value is '"First Last" (Text)', which is circled in red in the original image.