



Birthday App Tutorial

Metavine Genesis **Tutorial 1**



Metavine

www.metavine.com

© 2017 Metavine, All rights reserved | Birthday App Tutorial v2.0

Tutorial Introduction and Setup

About This Tutorial	1
Overview of Tutorials	1
What Will the Birthday App Do?	2
No Coding Skills Required	2
First Time Using Metavine Genesis	2
The Login Process	3
Quick Start Wizard	3
Main Navigation Menu Overview	4
Rename the Application	5

Tutorial One - Building the Application Base Structure

Basic Application Structure	8
Input - Packages and Attributes	8
The Design Studio Workspace	9
Organizational Explorer	9
The Stage	10
Create the Birthday Data Package	10
Editing Attribute Properties	13
Edit the Birthday Date Attribute	14
Edit the Reminder Date Attribute	15
Set Attributes to Mandatory	16
Modifying Package Settings	17
Set the Sort Order	18
Creating a User Input Activity	19
Creating Folders and Patterns	20
Create a Folder	20
Create a Pattern	21
Add a User Input Activity to a Pattern Flow	22
Define the Adding Birthdays Activity Content	23
Defining a Package in an Activity	26
Define the Birthday Data Package	26
Enable Access to the All Birthdays Pattern	31
Create a Looping Connector	33
Input Data to the Application	34
Learning Summary	36

Tutorial Two – Designing a Layout with Page Designer

About the Page Designer	37
Access the Page Designer	37
The Page Designer Workspace	38
Right Side Panel	38
Left Side Panel - Top	39
Left Side Panel - Bottom	40
Using Repeaters	41
Create a Repeater	41
The Three Repeater Components	44

Selecting the Repeater Container - Not the Repeated Item	45
Name the Objects Using the Properties Tab	45
Sizing the Repeater and Repeater Items	46
Repeat Items Horizontally in the Birthday Repeater	48
Create a Border	48

Tutorial Three - Creating User Input Actions

About User Input Actions	50
Create a UI Action for Adding the Birthday Package Repeater	50
Adding and Modifying Buttons	53
Add a Button in Page Designer	53
Modify a Button Using the Styles Tab	54
Add a Button Name and Attach a UI Action	55
Create a button for the UI action Delete a Birthday	57
Modify the Submit Button	58

Tutorial Four – Defining Application Rules

Defining Rules	61
Structure of This Rule	62
Access the Rule Builder	62
Understanding the Rule Builder Workspace	63
Define a Rule	65
Add the Reminder Date	65
The Complex Expression Builder Workspace	66
Build a Rule - Part 1	68
Build a Rule - Part 2	71
Test the Reminder Date Rule	75

Tutorial Five - Adding an Email Reminder Pattern

About the Email Reminder Pattern	77
Create the Email Contents Package	78
Create the Email Reminder Pattern	80
Creating an Automated Activity	81
Create an Automated Activity called Create Email Reminders	81
Create the All Birthday People Package	82
Transfer the Email Contents Package	85
Create the Upcoming Birthday Data Package	88
Create a Rule to Generate Email Reminders	92
Create the Send Emails Activity	102
Defining the Send Email Properties	104
Set up a Splitting Connector	106
Setting an Event in Calendar	109
Set Up a Daily Email Reminders Event	109
Set a Schedule on the Email Reminder Pattern	111
Test the Email Reminder Pattern	112

Tutorial Six - Adding an Email Invite Feature

About the Email Invite Feature	114
Adding Two New Attributes	114
Create Two New Attributes	114
Transferring New Attributes	116
Transfer the New Attributes to Birthday Data	116
Add the Invite Attribute to the Page Design	117
Creating the Gather Invitees Automated Activity	119
Create the Gather Invitees Activity	119
Setting a Restriction Rule	120
Set Up a Restriction Rule in the Birthday Data Package	120
Setting the Invite? Attribute Manually	123
Creating a Rule to Set the Email Sent Attribute	124
Create a Rule for Email Sent	124
Create a Connector Between Activities	126
Create the Invite Emails Activity	126
Define the Data Packages for Invite Emails	129
Create a Rule for Email Contents Body	131
Create the Email Body Rule.	131
Defining the Email Properties	133
Set up a Splitting Connector	136
Test the Email Invite Feature	139

Tutorial Introduction and Setup

About This Tutorial

In this set of tutorials you are going to use the Metavine Genesis Design Studio to build an application called the Birthday App. It is a birthday management application that is designed to track the birthdays of your friends and colleagues.

The application you are going to design will be very simple, and it's meant to be. We want to introduce you to Metavine's concepts while enabling you to create something that is easy to use and is actually useful.

You will discover new terms and learn the fundamentals of application creation on Metavine Genesis which is the Metavine platform. On completion of this set of tutorials, you will have created a birthday management application. You can then apply these new skills to design and build a business application for your organization.

Overview of Tutorials

There are six tutorials to guide you through the build process and help you understand the operation of the Metavine Genesis Design Studio. You should complete the tutorials in the sequence shown:

- Tutorial One - Building the Application Base Structure
- Tutorial Two - Designing the Application Layout in Page Designer
- Tutorial Three - Creating User Input Actions in Page Designer
- Tutorial Four - Defining Application Rules
- Tutorial Five - Adding an Email Reminder Pattern
- Tutorial Six - Adding an Email Invite Feature

What Will the Birthday App Do?

The Birthday App will allow you to enter in data of other people's birthday events in one convenient place. Once you add in a person's birthday and email address, you will have the option to email this person and invite them to use your Birthday App too. This enables the other person to supplement the existing birthday data you have entered for them. They will also be able to add in other people's birthdays that you may not know the details for.

You will also configure this application to send you reminder emails prior to a person's birthday. This way, you'll never send a belated birthday card again!

No Coding Skills Required

Metavine Genesis is designed to enable business professionals with no coding knowledge to build business applications. That's right! If you are a business analyst or technical specialist with an understanding of business logic, this tutorial is for you. Follow the steps in these tutorials to learn the fundamentals of creating your own application in a zero-code environment.

First Time Using Metavine Genesis

Whenever you start to use a new platform there is new terminology to learn, different layouts to get used to and new tools to discover. Metavine Genesis is no different. Before you start Tutorial One, read the topic **Main Navigation Menu** at the end of this section to familiarize yourself with how to access the main functional areas in the Genesis Design Studio. The tutorial will introduce you to other features and tools as you progress through the stepped procedures.

To assist you we have developed a mix of videos, articles and step-through tutorials which you will find on the Metavine Knowledge Base.

The Login Process

After verifying your registration request, follow these steps to access your Metavine Genesis account:

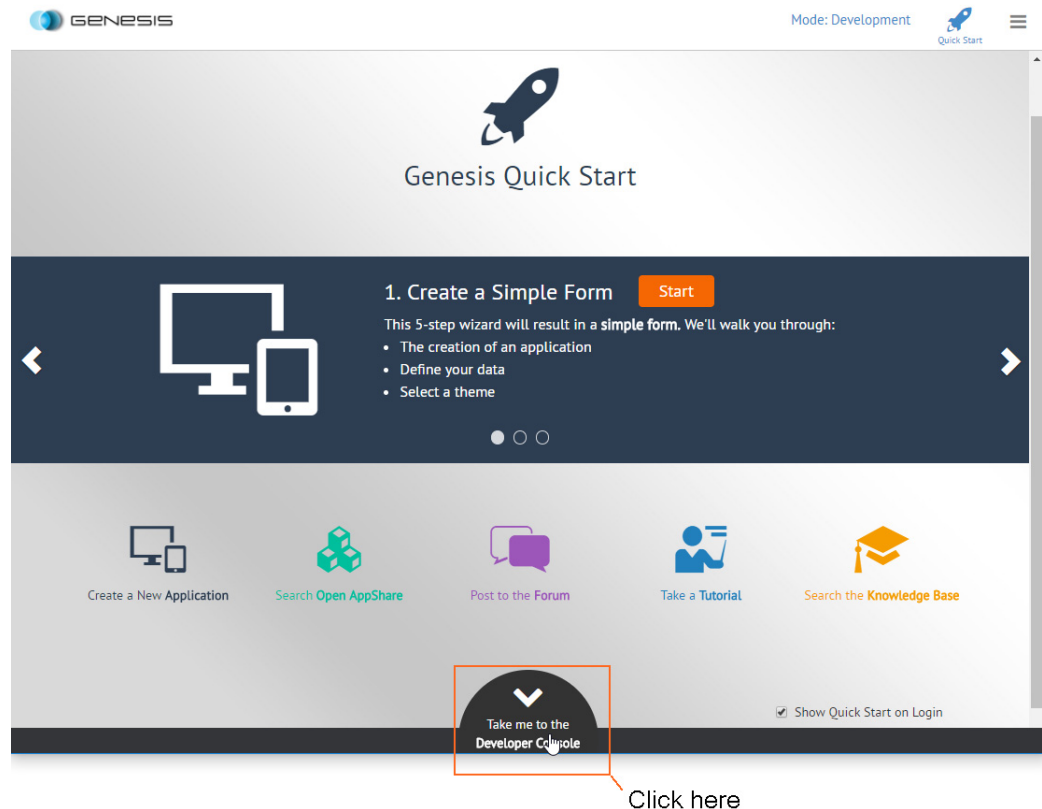
1. Check your email Inbox for the Application Designer Login Details email from Metavine.
2. Click the link to display the login screen.
3. Log in to Metavine Genesis.

You will be prompted to set a new password as part of the login process.

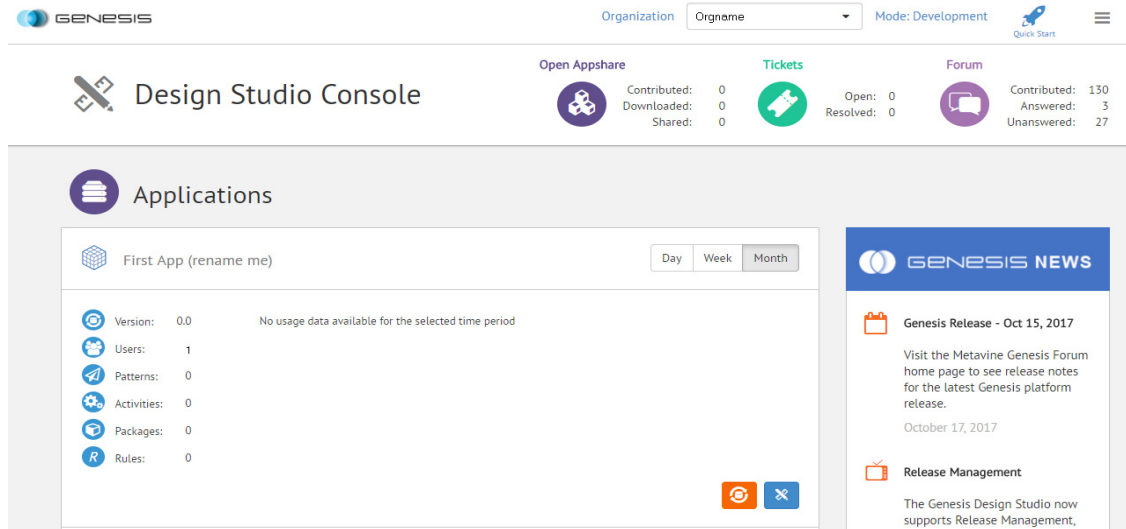
Quick Start Wizard

When you log in to Genesis, the Genesis Quick Start wizard displays. This wizard is a great productivity feature, however we will not be using it in this tutorial.

1. In the lower section of the panel, click **Take me to the Developer Console**.



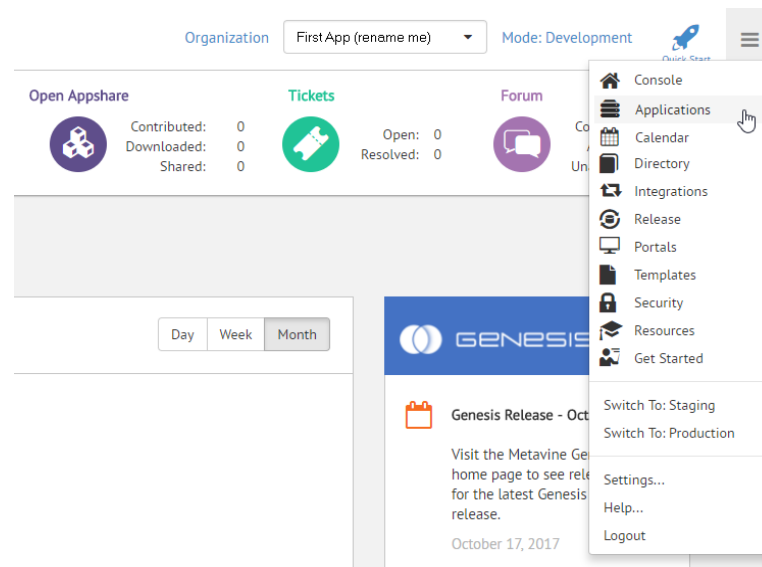
The Developer Console displays:



The Developer Console is like a dashboard that provides easy access to the Genesis Design Studio Main Navigation Panel, your applications, your account details, and a news panel.

Main Navigation Menu Overview

The Main Navigation menu is selected from the 3-bar menu icon in the top right of the screen. It provides easy access to the functional areas where you design, build and configure applications. In this tutorial, you will be mainly working in **Applications**.



You can access the following functions and resources from the Main Navigation menu:

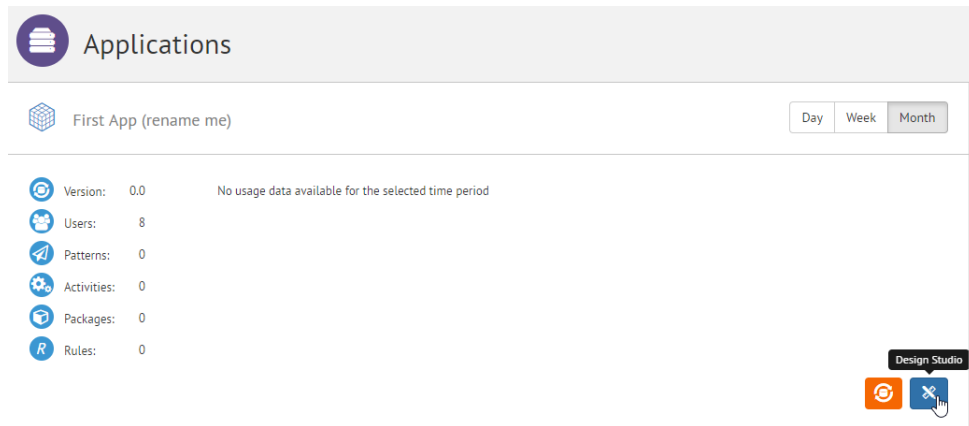
- **Console** provides an overview of Organizations and Applications that you have access to. You can monitor the usage of your Applications, release changes from one environment to another, and access the latest Genesis news from Metavine.
- **Applications** provides a graphical interface to define your application Patterns by configuring inputs, process and outputs.
- **Calendar** manages time-related parameters specific to your organization such as time zones, working hours and holidays.
- **Directory** contains Directory Data Packages, which are used to model the behavior and structure of a hierarchy within an Application.
- **Integrations** facilitates the integration of external systems with your application.
- **Portals** enables the generation of portals that provide access to your application.
- **Templates** enables the creation of predefined templates for Attribute and user interfaces.
- **Security** provides controls to set security parameters for your applications.
- **Resources** links to resources and support pages.
- **Get Started** links to Metavine tutorials.

Rename the Application

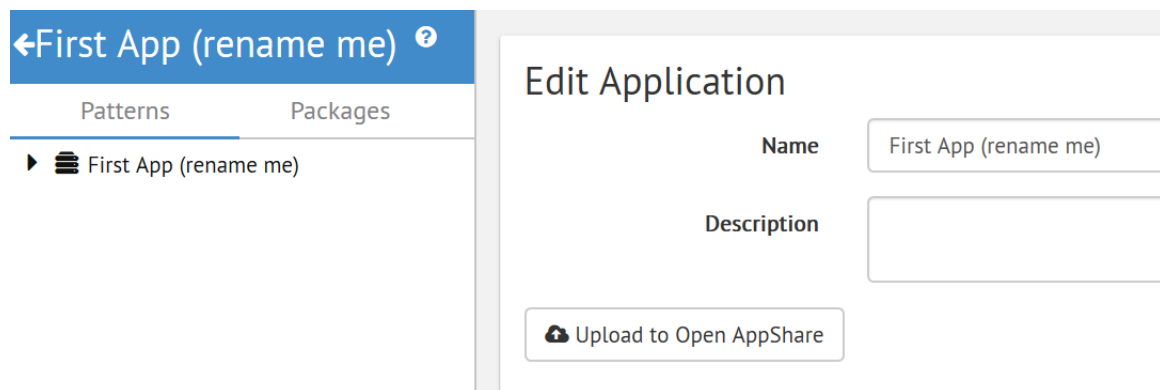
As part of your registration process, Metavine created your first application called **First App (rename me)**. You will need to rename the application to **Birthday App Tutorial** as shown in the following steps.

Steps

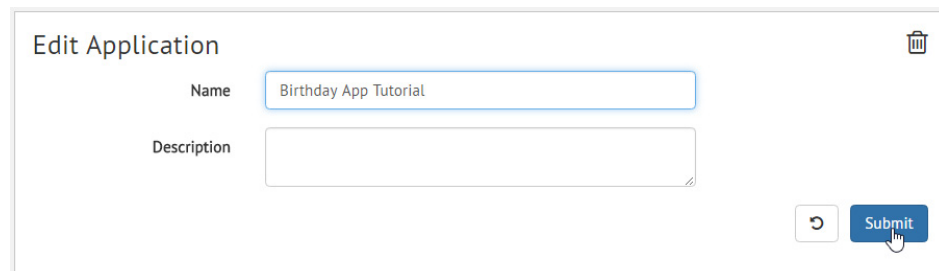
1. In the Applications panel, locate the **First App (rename me)** and click the **Design Studio** button.



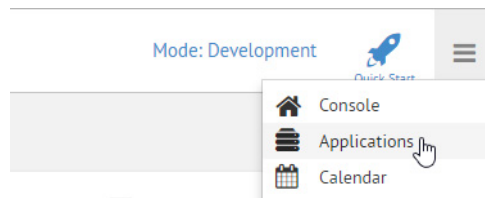
The Edit Application panel displays:



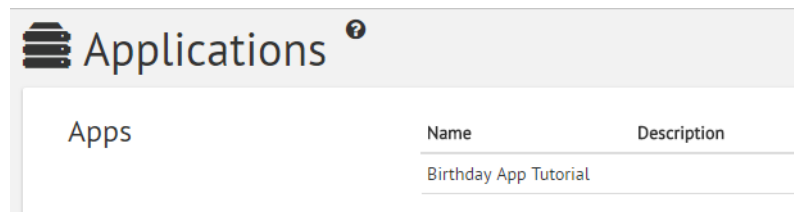
2. Click in the Edit Application **Name** field and delete **First App (rename me)**.
3. Type the name **Birthday App Tutorial**.



4. Click **Submit**.
5. Click on the Main Navigation menu in the top right of the screen and select **Applications**.



The name now appears in the **Apps** section.



Congratulations! You have completed your first tutorial task. Now proceed to *Tutorial One - Building the Base Structure*.

Tutorial One - Building the Application Base Structure

Learning Objectives



- ▶ Metavine Application Concepts
- ▶ The Design Studio Workspace
- ▶ Applications and Packages
- ▶ Activities, Folders and Patterns
- ▶ Working with a Pattern Flow

Basic Application Structure

First, let's talk about application structure. At the most basic level, the function of any application is to take data, process that data, and then produce an output specific to the application's purpose.

In order to build our Birthday App we will need to fulfill these three basic requirements: input, process and output.

Input - Packages and Attributes

In this first tutorial, you will begin the application build by constructing the first element - input. You will build out the base structure for the application to collect the data it needs. To do this, you are going to start by building Packages and Attributes.

- **Packages** are like database tables.
- **Attributes** are the table columns.

Another way to think about this concept is that Packages are a group of related pieces of data, the Attributes.

Client Package				
	Name (attribute)	Age (attribute)	Phone (attribute)	Gender (attribute)
Client Package Row 1				
Client Package Row 2				
Client Package Row 3				

Once you have made Packages and Attributes, you will then create a basic Activity.

An **Activity** is a task that participates in the application. It is the processing part and in some cases, dependent on the Activity type, can also be the output part of an application.

There is more to learn, but let us get started on the building and we will continue to explain other concepts along the way.

The Design Studio Workspace

Throughout these tutorials we refer to the workspace to describe the main work area in the Design Studio user interface. The workspace is the main work area on the screen and it is subdivided into two smaller areas known as the **Organizational Explorer** and the **Stage**. Both are described below.

Organizational Explorer

In this tutorial you will use the Organizational Explorer to build components for your App.

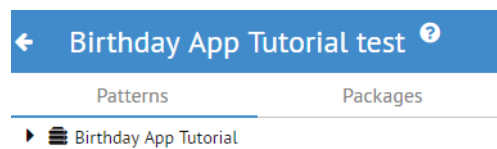
- To access the Organizational Explorer from within Applications , click on the **Birthday App Tutorial** in the list of applications.

- Alternatively, you can also access the Organizational Explorer from within your console  by clicking on the Design Studio button  to the right of the listed Birthday Application.

The Organizational Explorer displays Birthday App Tutorial in the panel heading. The Organizational Explorer panel displays on the left of the workspace and contains two tabs:

- **Patterns** contains Folders and Patterns for an application.
- **Packages** contains data Packages for an application.

Click on a tab to display the items stored within.



The Stage

The area to the right of the Organizational Explorer is called the Stage. The Stage is the main display area and it is context sensitive. This means that the information it displays is dependent on the selection you have made in the Organizational Explorer.

This tutorial will guide you through the process of using the Organizational Explorer to build a Package and its Attributes for birthday data. We will prompt you with screens along the way so you can be sure that what you are doing in your application matches up with the steps in the tutorial.

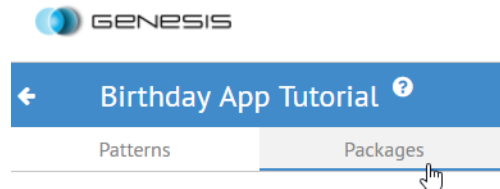
Create the Birthday Data Package

Let us begin by building the Packages and Attributes for this application. We'll start off with a single Package that contains multiple Attributes. This Package will collect data for the birthdays that you create.

In the main navigation panel, the Packages icon is where you create and modify Packages. And in case you've forgotten what a Package is, it is Metavine's organization system for data.

Steps

1. In the Organizational Explorer, click **Packages**.



2. In the lower-right corner of the panel, hover over the **Plus** button.
3. Click New **Package**.



The New Package **Details** pane displays on the Stage.

4. In **Name**, type **Birthday Data**.
Next you will create a set of related Attributes in the Attributes pane (located directly below **Details**). We'll step you through the creation of the first two Attributes and then you can add the rest on your own. The first Attribute to create is for the Birthday Person.
5. In **Attribute name**, type **Birthday Person**.
6. In **Data type**, check that Text is displayed. If not, click **Text**.

Details

Name: Birthday Data

Name in UI: Name in UI

Package type: ☒ Data ☐ Reporting

Parent package: None

Remote package: ☐

Attributes

Attribute name	Data type
Birthday Person	Text

Add attribute

7. Click **Add Attribute** to add another Attribute.

Repeat steps 5 and 6 to add a **Birthday Date** and set the **Data type** as **Date/Time**.

Add New Attribute

Details

General

Name: Birthday Date

Attribute based on: None

Data type: Date/Time

8. Got the hang of it now? Go ahead and finish off adding the rest of the Attributes needed for this Package from the list in the table below.

Attribute Name	Data Type
Birthday Person's Email	Text
Reminder Date	Date/Time

9. Once you have completed step 8, click the blue **Submit** button in the upper right-corner of the page to save the Birthday Data Package.

New Package

Attributes

Attribute name	Data type
Birthday Person	Text
Birthday Date	Date/Time
Birthday Person's Email	Text
Reminder Date	Date/Time

Add attribute

Congratulations! You have created your first data Package using the Design Studio.

Editing Attribute Properties

To make your Package accurately reflect the application requirements, you can edit the properties of each individual Attribute. We'll get you started by showing you how to edit a few different Attributes. Once you feel comfortable with this procedure, you can have a go at customizing Attributes on your own.

The Birthday Data Package and its Attributes should still be displayed on the Stage of the Design Studio. If they aren't, select the Birthday Data Package from the Organizational Explorer to display its details.

In this section we will complete the following tasks:

- Edit the Birthday Date and Reminder Date Attributes so that their date output is set to **Month, Day, Year** in the form 01/22/2001.
- Set the Birthday Date and Birthday Person Attributes to **mandatory** so that a user must enter in data for these fields before submitting the Activity.

Edit the Birthday Date Attribute

In the following steps we will change these Attributes:

- Change Birthday Date order from Day/Month/Year to **Month/Day/Year** so that the month comes first.
- Change the Month format from Aug to **08**.
- Add a separator (*/*).

Steps

1. From the **Attributes** list, click the **Birthday Date** Attribute.

The Attribute details display on the Stage.

The screenshot shows the 'Attributes' list on the left with 'Birthday Date' selected. The 'Edit Birthday Date' form on the right has the following fields:

- Name:** Birthday Date
- Attribute based on:** None
- Data type:** Date/Time

2. In **Edit Birthday Date**, scroll down to **Output format** and click the blue **Edit** button.

Output format

A text input field for the output format, currently empty.



3. Hover over the **Month** label of the date to display a grey outline.

4. Drag and drop the **Month** in front of the **Day**.

Format Date & Time

5. Click the **Month** drop-down list and then click **08**.

6. Directly below, click the **Date separator** drop-down list and then click **/**.
7. Click **Submit**.

You have now changed the format of the Birthday Date output.

Edit the Reminder Date Attribute

Now we will edit the **Reminder Date** Attribute.

Steps

1. From the **Attributes** list, click the **Reminder Date** Attribute.
The Attribute details display on the Stage.
2. Repeat steps 2 to 7 above to edit the **Reminder Date** Attribute.

Now the Birthday Date and Reminder Date outputs have the same output format.

Set Attributes to Mandatory

We will now set Birthday Date and Birthday Person Attributes to mandatory.

Steps

1. From the **Attributes** list, click the **Birthday Date** Attribute.
2. In **Edit Birthday Date**, scroll to **User Interface** and click the **Mandatory** check box so that it is set.
3. Ensure the **Editable** check box (beside the Mandatory check box) is also set.

The screenshot shows the 'Edit Birthday Date' configuration interface. On the left, the 'Attributes' list includes 'Birthday Person', 'Birthday Date' (selected), 'Birthday Person's Email', and 'Reminder Date'. The main panel is titled 'Edit Birthday Date' and contains a 'Details' section. Under the 'General' tab, the 'Name' field is 'Birthday Date'. The 'Attribute based on' dropdown is set to 'None'. The 'Data type' dropdown is set to 'Date/Time'. The 'Default' section has the 'Time at load' checkbox checked, with a date/time picker showing 10:09 AM. The 'Encrypt data' checkbox is unchecked. Under the 'User Interface' section, the 'Display as' dropdown is set to 'Date Selector'. The 'Mandatory', 'Editable', and 'Visible' checkboxes are all checked.

4. From the **Attributes** list, click the **Birthday Person** Attribute.
5. In **Edit Birthday Person**, scroll to **User Interface** and click the **Mandatory** check box so that it is set. Ensure that the **Editable** and **Visible** check boxes are also set.

The Birthday Date and Birthday Person Attributes are now set as mandatory so that a user must enter data into these boxes before submitting an Activity.

6. To save all changes that you have made for these Attributes click **Submit**.

► **NOTE:** The Attribute names display in orange as you edit the settings. When you click Submit, the names revert to black to indicate that they are saved.

You can continue to explore the other possibilities for editing Attribute properties. There are different property settings dependent on the data type that you select for an Attribute. Examine each Attribute to see the setting options that are available.

Modifying Package Settings

Next, we will make some modifications to the Package settings so that our birthdays are sorted by ascending date. There are three main settings on a data Package as follows:

- Package Key Attributes
- Sort Order

- Search Attributes

Right now, we're going to set up the **Sort Order** of the Birthday Data Package. We want our birthday's to be sorted via the birthday date, so that the birthdays are sorted and listed in ascending order based on the date chosen for each Package row.

Set the Sort Order

We will set the Sort Order to list birthdays in ascending order.

Steps

1. In the **Package - Birthday Data** settings, scroll down to **Sort Order**.
2. Click the blue **Plus** button.

Sort Order

List of attributes that determine the order of package instances. Click on the arrow to flip the direction.

Order attribute	Direction
    	

A list displays the Package Attributes.

3. Click the **Birthday Date** Attribute in the list.

4. Ensure **Direction** is set to the **Up Arrow** (Ascending).

Attributes

Select one or more attributes to add.

Attribute name	Direction	Package path
 Birthday Person		Birthday Person
 Birthday Date	↑	Birthday Date
 Birthday Person's Email		Birthday Person's Email
 Reminder Date		Reminder Date




5. Click **Submit**.

The Birthday Date Attribute is now listed in the Sort Order section for the Package.

6. To save all Package changes, in the upper-right of the Stage, click **Submit**.

► **TIP:** In these tutorial steps, if you make a mistake when editing a form, you can

click the **Undo**  (beside the **Submit** button) to cancel the input and return to the previous state.

Creating a User Input Activity

Next, we will create a User Input Activity to allow a user to enter birthday information. A **User Input** (UI) Activity is specifically designed to visually represent Packages and their Attribute fields. In this case, it will allow users to add a birthday and edit the birthday data in the Attribute fields.

We will create a User Input Activity in five stages:

Stage 1 - Creating Folders and Patterns

Stage 2 - Adding User Input Activity to a Pattern Flow

Stage 3 - Defining the Activity Content

Stage 4 - Modifying a Package Definition in an Activity

Stage 5 - Creating a Connector

Creating Folders and Patterns

Before an Activity is created, you must first create a Folder and Pattern.

- **Patterns** are a collective group of Activities that perform a specific action or a set of actions.
- **Folders** are the filing system for Patterns. They provide a way to group Patterns that perform similar tasks based on their Activity content.

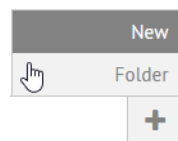
The benefits of Folders will become obvious when you build large complex applications that contain a vast number of Patterns. In the previous steps, you clicked Submit to set the Birthday Date sort order. The Organizational Explorer should now display on the left of the Stage - we will use it to create a folder.

Create a Folder

We will create a Folder called **Birthday Management** in the following steps.

Steps

1. In the Organizational Explorer, click **Patterns**.
2. Click on **Birthday App Tutorial**.
3. In the lower-right corner of the Organizational Explorer, hover over the **Plus** button and click New **Folder**.



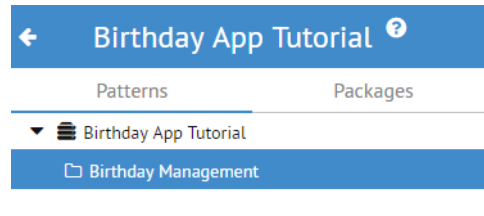
The **Details** pane for the new Folder displays on the Stage.

4. In the **Name** field, type **Birthday Management** and click **Submit**.

Details

Name

The Birthday Management Folder is now listed in the Organizational Explorer.

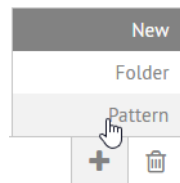


Create a Pattern

We will create a Pattern called **All Birthdays** in the following steps.

Steps

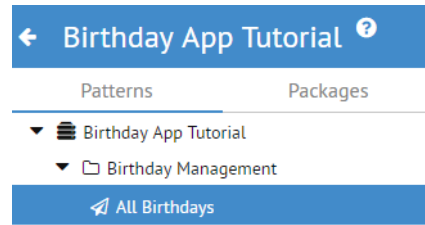
1. Click the **Birthday Management** folder, so that it is highlighted.
We've made sure that you select the **Birthday Management** folder before creating the Pattern. This is so that it tells the system to store the new Pattern being created within the selected folder.
2. Hover over the **Plus** button in the lower-right corner of the Organizational Explorer and click **New Pattern**.



The Pattern details are displayed on the Stage.

3. In the **Name** field, type **All Birthdays** and in the upper-right corner, click **Submit**.

The Pattern is now listed in the Organizational Explorer below the Birthday Management folder. A paper plane icon represents a Pattern.



Add a User Input Activity to a Pattern Flow

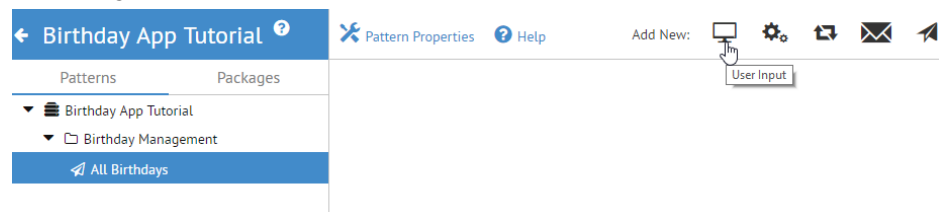
Whenever a Pattern is selected in the Organizational Explorer, its Pattern flow is displayed on the Stage. The Pattern flow is the combination of Activities that are contained inside a single Pattern. Think of the Pattern flow like a storyboard of tasks. At this point in the application build process, we need to add a **User Input** Activity to the **All Birthdays** Pattern flow.

In the following steps we will add a User Input Activity to the All Birthdays Pattern flow.

Steps

The Activity type icons are located in the upper-right corner of the Stage.

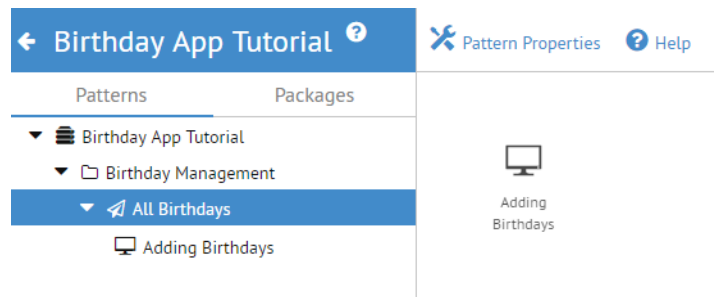
1. Drag and drop the **User Input** Activity icon (computer monitor) onto the **All Birthdays** Pattern flow.



A form displays the basic details for the new Activity

2. In **Activity name**, type **Adding Birthdays**, and click **Submit**.

You will now see your Activity in the Pattern flow represented by the User Input Activity icon and labeled with the Activity title beneath it.



Define the Adding Birthdays Activity Content

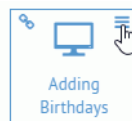
Next, we will define the contents of each Activity based on Packages created in **Packages**. Only one Package has been created so far and this will be the only Package needed for this Activity to begin with.

In the steps below, we will transfer the **Birthday Data** Package into the **Adding Birthdays** Activity and leave the default settings. We'll explain how to change the default settings later in the tutorial.

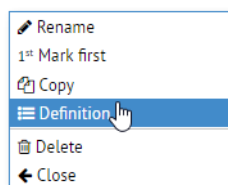
Steps

1. Hover over the **Adding Birthdays** Activity so its outline is highlighted in blue.

Two icons appear in the upper corners of the Activity - the one on the right is known as the **List** icon.



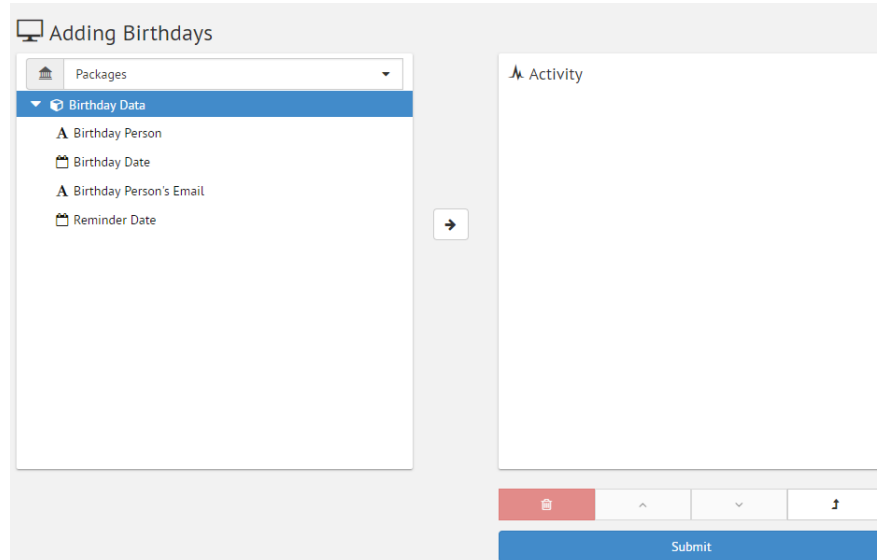
2. Click the **List** icon to see a drop-down list of options.
3. Click **Definition**.



The **Activity Definition Transfer** page displays.

Below Packages, the **Birthday Data** Package displays.

4. Click on the **Birthday Data** package to highlight it.



The tree expands to display the Attributes.

5. To transfer the Package into the Activity, click the **Right-arrow** between the two columns.

The **Source Package Details** form displays.

This is where you can set a new Package name (if required) and define what contents of the Package are added to the Activity. In this case we are going to leave the default settings.

Source Package Details

Package name	Birthday Data
Name in UI	Name in UI
	☐ No name in UI ☐ Suppress display of attributes in UI
Packages to include	☒ Linked packages ☒ Sub-packages
Attributes to include	☐ Only key attributes ☐ Only search attributes ☒ All attributes

←
Submit

- Click **Submit**.

The Package transfers over into the Activity column.

- In the lower-right corner of **Activity**, click **Submit**.

The screenshot shows the 'Adding Birthdays' interface. On the left, under the 'Packages' tab, the 'Birthday Data' package is selected. It contains four attributes: 'Birthday Person', 'Birthday Date', 'Birthday Person's Email', and 'Reminder Date'. An arrow button points from the Packages column to the Activity column. On the right, under the 'Activity' tab, the 'Birthday Data' package is now listed, containing the same four attributes. At the bottom right, there is a 'Submit' button.

You have now saved the addition of the Package to the Adding Birthdays Activity. Well done! Now that you have reached this point, it's time to complete the final stage by modifying the Package definition in the Activity.

Defining a Package in an Activity

Package definitions can be modified in an Activity to alter how the Package is participating in that Activity. It is important to know the purpose of the Activity before creating it, as this will help you better define its Package contents.

The purpose of the Adding Birthdays Activity is to create, edit, and delete birthday data. Therefore this Package needs to be set up to allow a user to do this.

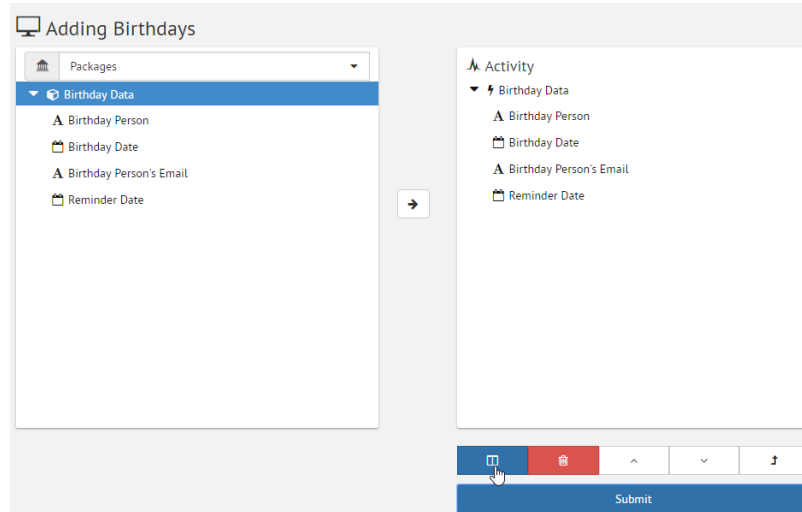
The Package already has some settings defined for it. Each time a new Activity is created and a Package is added to it, the Design Studio tries to intelligently configure the Activity's Package settings. Often this configuration is close to the correct settings, however some changes are usually necessary. The following steps describe how to modify the Package definition in an Activity.

Define the Birthday Data Package

We will access the **Activity Definition main page** to define the Birthday Data Package for the Adding Birthdays Activity.

Steps

1. On the Activity definition transfer page, in the lower-right corner, click the blue **Open Book** button.



The Activity Definition main page displays the Activity content on the left and the **Birthday Data** Package on the right.

2. On the right, click on the tile options to toggle between blue (selected) and clear (not selected). The display form of these options will change from squares to rectangles if you have reduced your browser window.
3. On the right, ensure that the **CREATE NEW ROW(S)**, **UPDATE ROW(S)** and **DELETE ROW(S)** settings are all selected (blue).

These three settings are required in the Adding Birthdays Activity to enable a user to **add** (create new rows), **edit** (update rows) and **delete** (delete rows) birthdays.

4. Ensure that **SELECT ROW(S)** and **REQUIRE ROW(S)** are not selected (clear).

Select Row(s) is *not required* in the Adding Birthdays Activity because a user does *not require* a control (such as a drop-down list) to select birthday data rows. This will become clear when you see the application at runtime.

Require Row(s) is a validation flag that is not required for this tutorial.

5. Ensure that the **Multiple rows?** check box is set.

This setting is required in the Adding Birthdays Activity because a row is required for *each birthday* that is added. There will be multiple birthdays rows added in this Activity. If this check box was not set, then only one birthday could be added to the Birthday App!

6. Ensure that the **Retrieve existing rows** check box is set.

This setting is required so that, when the Adding Birthdays Activity loads, any existing birthday rows are retrieved. Existing birthdays are displayed and can be edited or deleted.

7. Click **Restrict to: Rows most recently referenced in Pattern** to clear the check box.

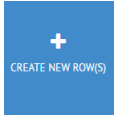
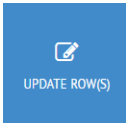
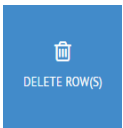
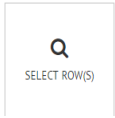
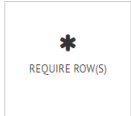
This check box setting is not required because there should be no restriction applied to birthday rows that are retrieved. We would only apply a restriction

if we wanted to change the behavior of the Birthday Data Package in this Activity.

8. Below the **Activity** content, click **Submit**.

- **NOTE:** You will see that when you edit the details of a Package, the Package name in the Activity content displays in orange. Orange text indicates that a save is required. When you click **Submit**, the Package name reverts to black.

You have now defined the Package data for the Adding Birthdays Activity. An explanation of each setting is given in the table below.

Selected?	Setting	Function	Purpose of this setting in the Adding Birthdays Activity	User Interface Control
YES		Inserts a new Package row of data.	Enables a user to add a new birthday.	
YES		Updates an entire Package row of data.	Enables a user to edit a birthday.	
YES		Deletes an entire Package row of data.	Enables a user to delete a birthday.	
NO		Selects a Package row of data using a drop-down list, on-page list, grid etc.	Not required for Adding Birthdays - no selection method is required for the Adding Birthdays Activity. A user does not need to select rows from a list, grid etc for this Activity.	Drop-down list, on-page list, grid, etc
NO		Enables a data construct for creating structured Activities.	Not required in this tutorial.	No user interface control.
YES	Multiple rows? ☒	Enables the creation of multiple Package rows.	The Adding Birthdays Activity allows the creation of multiple Package rows because a Package row is required for each birthday that is added.	No user interface control.
YES	Retrieve existing rows ☒	Ensures that when an Activity is loaded, any Package rows that have previously been created are displayed to enable editing or deletion of data.	The Adding Birthdays Activity will retrieve existing birthday Package rows that have been created so that a user can view and modify birthdays.	No user interface control.

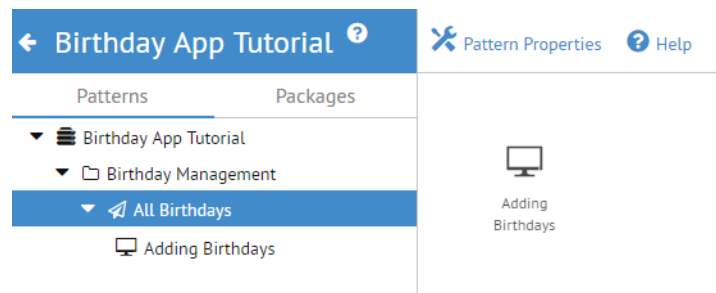
Selected?	Setting	Function	Purpose of this setting in the Adding Birthdays Activity	User Interface Control
NO	Restrict to ☐ Row(s) most ☐ Row(s) creat ☐ Row(s) that f	Restricts the behavior of a retrieved Package row when Retrieve existing rows is set.	Not required for Adding Birthdays - no restrictions are required on birthday Package rows for the Adding Birthdays Activity.	No user interface control.

Enable Access to the All Birthdays Pattern

We are going to enable access to the Adding Birthdays Activity by allowing access to the entire Pattern via URL.

Steps

1. In the Organizational Explorer, click the **All Birthdays** Pattern.
The Pattern flow displays on the Stage.



2. Click the **Pattern Properties** button located in the top-left corner of the Stage.
The **Details** pane displays.

- Below URL Access, click the check box **Allow access via URL** so that it is set.

ll Birthdays [Pattern Flow](#) [View in Runtime](#) [Upload to Open AppShare](#) [Help](#) [Submit](#)

Details

Name: All Birthdays

Parent folder: Birthday Management

Schedule: ☒ None ☐ Specific date & time ☐ Relative to a specific event

Notes:

URL Access

☒ Allow access via URL

- In the upper-right of the page, click **Submit**.
A **Development URL** is generated for you.
- Scroll down to **Access**, click **Anyone**.
The default access setting is **Authenticated Users Only**. This setting provides access to users that are registered for this organization and that are logged in. Changing it to **Anyone** ensures that later on, when the application is shared with others, they will be able to gain access to the Pattern too.
- Click **Reset Permissions on all activities of this Pattern** to set this check box.

Access

Select the access type:

- ☒ Anyone
 - ☐ Allow audit
- ☐ Authenticated users only
- ☐ Specific directory entities and/or rules
- ☐ Allow parent folder permissions to propagate to this pattern
- ☒ Reset permissions on all activities of this pattern

- In the upper-right of the page, click **Submit**.
Your changes are saved.

Create a Looping Connector

There is one final stage of Tutorial One to perform. We need to create a Connector for this Activity. A Connector is a path that links two Activities to each other, or it can link an Activity to itself. When linked to itself, it is called a **Looping Connector**. A Looping Connector can be used where there is a single Activity in the Pattern. When this Activity is submitted, the expected behavior is for the Activity to reload, so that a user can input data and submit the Activity again.

To create a Looping Connector for the Adding Birthdays Activity, you will need to be in the Pattern flow of the Activity as shown in step 1.

Steps

1. Click the **All Birthdays** Pattern to display the flow on the Stage.
2. Hover over the **Adding Birthdays** Activity and in the top-left corner, click the chain **Link Connector** icon.
3. Drag the link from the top-left corner to the lower-right corner.



The **New Connector** form displays.

4. In this case, we will leave the default settings as they are. This is so that the Connector path is followed every time the Activity completes successfully.

New Connector

If the activity completes ☒ Successfully ☐ With an exception

Under the following circumstances ☒ Always ☐ By default ☐ Under the following condition

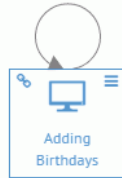
Link to activity

Subsequent action ☒ No change ☐ Create multiple instances of the next activity

Connector Label

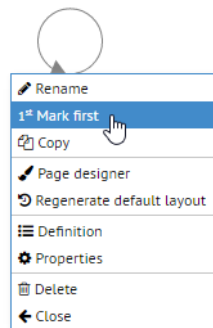
5. Click **Submit**.

The Connector appears in the Pattern flow, looping back over the Activity.

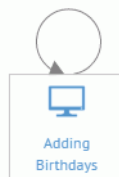


The final step is to mark this Activity as the *first Activity* in the Pattern. You may wonder why we would do this if there's only one Activity in the Pattern. Well, with the addition of the Looping Connector over this Activity, you need to tell the Pattern that it should run the Adding Birthdays Activity first when it initially loads.

6. Hover over the Activity and click on the **List** icon (in the top-right corner).
7. From this drop-down list, click **1st Mark first**.



A message confirms your selection and a **grey square** surrounds the Activity icon to indicate it is marked first.



Well, that's it! You're finished for now. Design Studio will automatically generate a page layout for this Activity (you will see how to customize this page layout later in the tutorial). So now, all you have to do is navigate to the page and start inputting data.

Input Data to the Application

To begin adding data to your application:

1. Above the Stage, click **Pattern Properties**.
2. In **Details**, click on the **Development URL**.

The user input, Adding Birthdays Activity displays in runtime.

If you do not see the screen below and instead get an access denied message, check that you have set the Access settings correctly.

Adding Birthdays

Birthday Data

Birthday Person Birthday Date Birthday Person's Email Reminder Date

The **Reset permissions on all activities of this Pattern** should be selected - see Enable Access to the All Birthdays Pattern.

The Birthday Data Package appears as a Table where you can add multiple rows, and delete or edit an individual row. Right now our page design looks a little basic, but we will fix this in the next tutorial. For now though, you have created your first Activity in the application.

Learning Activity - Multiple Rows

This is an *optional* learning activity to show you how the **Multiple Rows** option works in the Adding Birthdays Activity. If you prefer, you can go straight to Tutorial Two right now and return to this learning Activity at a later stage.

We will temporarily clear the **Multiple rows?** check box so that the option is deselected. Once Multiple rows is deselected, we will lose the ability to add multiple birthdays to the application. Follow the steps below:

To remove Multiple rows

1. Return to the Activity Definition main page for Adding Birthdays. To get there, in the Organizational Explorer, select the **Adding Birthdays** Activity.
2. Click the **Multiple rows?** check box to clear it.
3. Click **Submit**.

To view the result

4. In the Organizational Explorer, click the **All Birthdays** Pattern.
5. Above the stage, click **Pattern Properties**.
6. In **Details**, click on the URL.

The user input Adding Birthdays Activity displays.

► Note: If you previously entered more than one row of data, you will get an error.

7. Compare the result to the previous time Adding Birthdays was displayed (when Multiple rows? was selected).

You will see that the Add/Edit/Delete controls are removed.



By deselecting Multiple rows? you have rescinded the ability to add multiple birthdays.

8. Repeat steps 1 to 3 above and at step 2, ensure that the check box **Multiple rows?** is set.

The Adding Birthdays Activity is returned to its previous state.

Learning Summary

In this tutorial you have learned how to create the fundamental elements of an application. These include Packages, Attributes, Folders and Patterns. You created an Activity to generate a simple application that allows for data input.

Congratulations! You have completed the first part of the Birthday App Tutorial.

Tutorial Two – Designing a Layout with Page Designer

Learning Objectives



- ▶ The Page Designer Workspace
- ▶ Repeaters

About the Page Designer

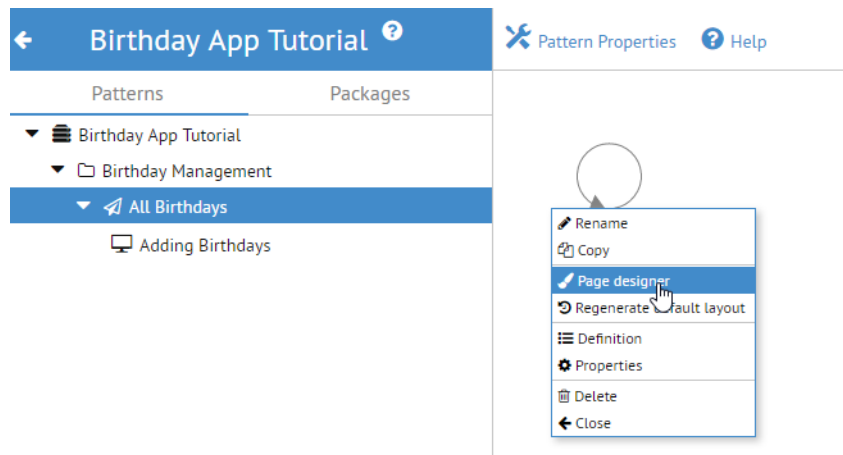
The application that you created in the first tutorial gives you a simple layout of a list of birthdays. Now we will modify that simple layout using the Page Designer, so that your application interface looks professional and is optimized for device screens.

In this next tutorial we've come up with a basic page design for the Birthday App. The layout that we will create is similar to a set of cards or Post-it notes on a board.

Once you know a little more about how to use the Page Designer, you can begin to add your own unique ideas to the layout. You will be introduced to some new concepts here, as well as an entirely new area of the platform.

Access the Page Designer

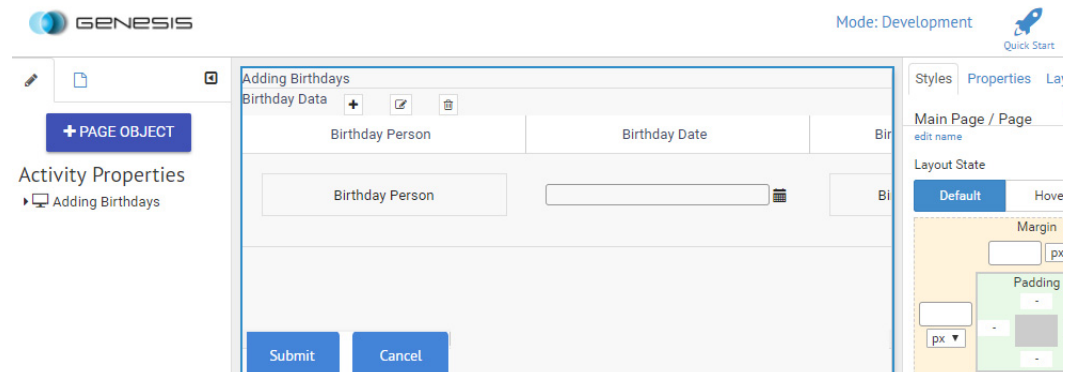
1. In the Organizational Designer, click the **All Birthdays** Pattern flow.
2. Hover over the **Adding Birthdays** Activity.
3. Click the **List** icon in the upper-right corner.
4. In the drop-down menu, click **Page designer**.



The Page Designer opens. The Page Designer workspace is described next.

The Page Designer Workspace

The central area of the workspace presents the page layout for an application.



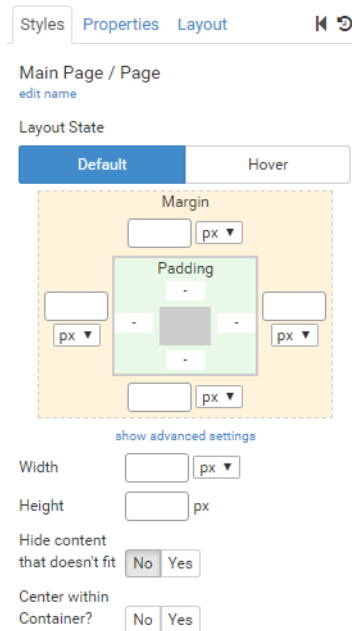
In this area you can chose page elements to edit.

Right Side Panel

On the right of the workspace there is a side panel containing three tabs. These tabs correlate with the page object that is selected on the layout.

- **Styles** contains the tools to alter the chosen page object's aesthetic design.

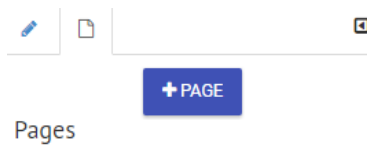
- **Properties** contains the tools to add functionality to the page object.
- **Layout** contains a tree formation to change the page layout.



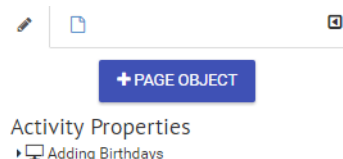
Left Side Panel (Top) -

On the left side panel, there are two tabs containing headings:

- **Pages**



- **Activity Properties**

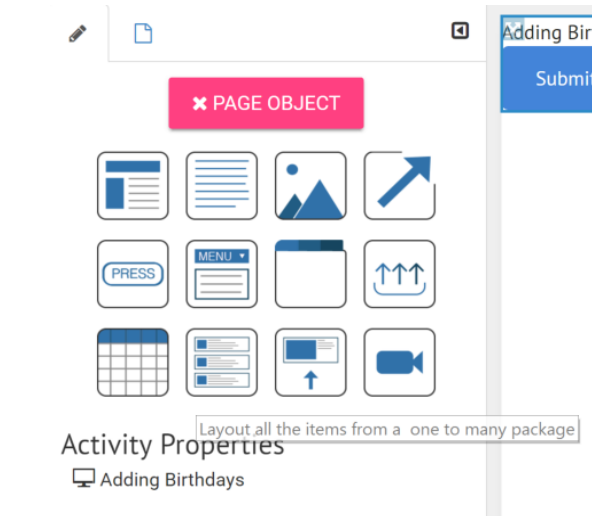


You can toggle between these options by clicking on the tabs.

Note: The **Pages** tab helps you to locate which page you want to design.

The **Activity Properties** tab is where you modify the design for the page you selected using the Pages tab. Within the Activity Properties tab, the **+ Page Object** button is located at the top. Below this button you will find the Activity content hierarchy. Activity Properties includes all the Packages and Attributes in the Activity, independent of whether they are included in the page layout.

- Click **+ Page Object** to see a panel of page objects that you can insert into the layout.



Left Side Panel - Bottom

At the bottom of the left side panel, you will see **Layout Mode**.



Click on a **Layout Mode** button to preview the page design for each screen size:

- Desktop
- Tablet
- Phone Horizontal
- Phone Vertical

Steps

1. Choose the Pages tab from the upper-left panel.
2. Select the Main Page by choosing **Main Page**, directly below Pages.
3. Toggle over to the **Activity Properties** tab (pencil icon), still within the left side panel.

We're going to delete the Table and replace it with a Repeater.

Using Repeaters

First, we are going to remove the Birthday Data Package Table design that was produced in the default layout generation and replace it with a Repeater.

A Repeater allows you to repeat groups of related items on a page. Repeaters are flexible for configuration and are often used in place of a Table. This will give us the flexibility to modify the design by deleting and adding new page objects so that we can achieve a required design. The design will resemble a row of cards or Post-Its on a board. Only Activity Packages that are set to multiple rows can be used for Repeaters.

► **NOTE:** To create a Repeater, the **Multiple rows?** check box must be set for the Package, in the Activity Definition main page.

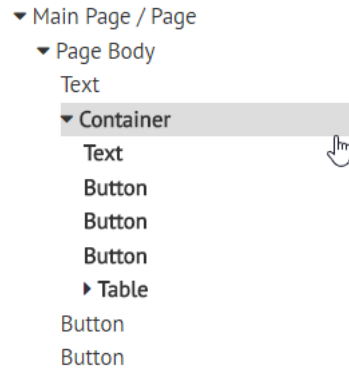
Create a Repeater

In the following steps, we will transform the Birthday Data Package Table into a Repeater on the page layout.

Steps

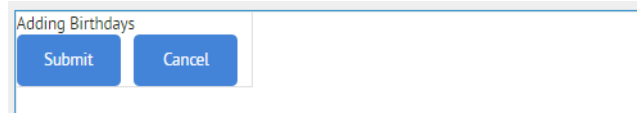
1. In the right side panel, click on the third tab named **Layout**.
2. Click the **Main Page/Page** to display its objects. Continue to click on the sub-objects to expand the entire object tree.
3. In the page layout structure, click the container holding the Table and other page objects.

4. Ensure the entire container is selected (Text, etc) - not just the Table. A precise way of selecting page objects is to click the **Layout** tab and expand the object tree - then click **Container**.

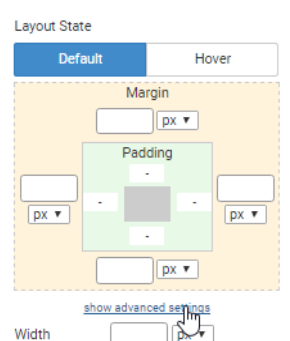


5. In the upper-right of the **Styles** tab, click the grey **trash can**.
A confirmation message appears.
6. Click **OK**.

The object is removed. Only the **Activity title**, **Submit** and **Cancel** buttons remain. The page body is still selected with a blue line.



7. Click **show advanced settings** below **Padding** to display advanced settings.



8. With the page body selected, in the **Styles** tab, check that the **Max Width** setting is blank.

hide advanced settings

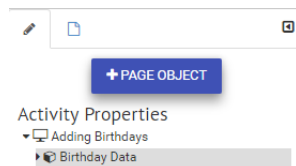
Width % ▾

Max Width px ▾

9. Then set the Width to be 100% and click **Save** (located in the lower-right of screen).

This ensures that the width restriction is removed for the selected object so that the page body is now the size as the main page.

10. In the left side panel, click the panel tab to toggle to the Activity Properties.



11. To add a Repeater, click + **Page Object**.
12. In the panel, click the **Repeater** (3rd last button). When you hover over this button a tool tip displays: **Layout all the items from a one-to-many package.**



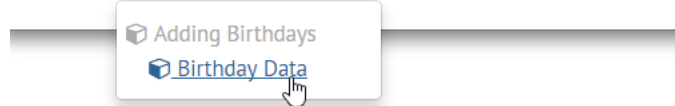
A form displays to choose which Package to base the Repeater on.

13. From the **Activity content**, click the **Birthday Data** Package (if dimmed, see note below).

Which Package?

Which package's data would you like to display. (Must be One to Many)

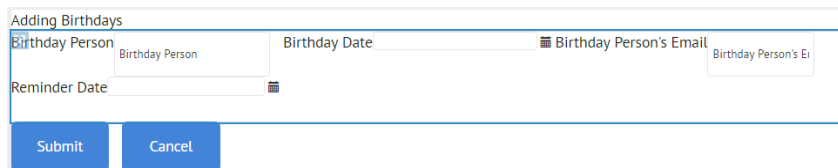
Package: Package ▾



The Repeater appears on the page layout and is populated with the Birthday Data Package Attributes.

- **NOTE:** Repeaters can only be formed from Packages that are set to multiple rows (or from a Linked Package in a one-to-many relationship). If you cannot select the Birthday Data Package in step 9, check that Multiple rows is set for the Birthday Data Package in the Activity Definition main page.

14. Move the Repeater so that it sits beneath the page header and above the **Submit** and **Cancel** button as shown below.



You can move the position of the Repeater on the page layout by dragging and dropping it into a different location. Alternatively, you can move its position from within in the **Layout** tab by using the layout tree.

- The layout tree is a precise way of identifying and moving page objects.

Try both Functions now so that you are familiar with them.

The Three Repeater Components

It is important to note that there are three main components to the Repeater:

- **Repeater** is a container that holds the repeated items.

- **Repeated items** represents a row of a Package.
- **Repeater Package** is the selected Package for the Repeater (and its Attributes)

When you edit a Repeater, and any of its contents, it's important to keep these separate components in mind.

Selecting the Repeater Container - Not the Repeated Item

When a Repeater is first added to a page, there is no padding between the Repeater container and the repeated item. Therefore, you may think you've selected the Repeater, when in fact you have really selected the repeated item.

As described above, the Layout tab tree view provides a precise method for selecting objects. When an object is selected and highlighted with a blue border, the tree view provides a description of what is selected. You can drag the object in the layout tree to move its position on the page. Use this method to identify page elements until you are more familiar with Page Designer.

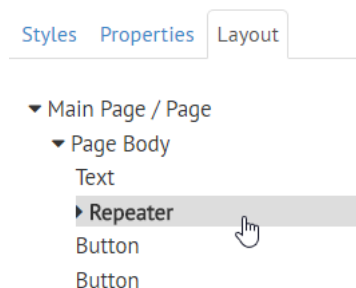
Name the Objects Using the Properties Tab

To avoid confusion, we are first going to name the Repeater, and then the repeated item.

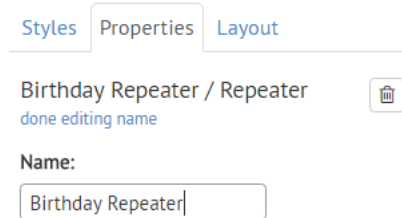
Steps

To name the Repeater:

1. In the **Layout** tab tree, click **Repeater**.



2. Click the **Properties** tab and then click **edit name**.
The **Name field** displays.
3. Type **Birthday Repeater**.



4. Click **done editing name**.
5. Click on the **Layout** tab and perform these four steps for the **Repeated Item** and name it **Birthday Repeated Item**.
6. In the lower-right corner, click **Save**.
The new names display in the Layout tab.

Sizing the Repeater and Repeater Items

We want the repeated items to sit alongside each other like cards or Post-its on a board. The Repeater container will need to be set to the total width of the main page and the repeated items can be set to a smaller card size.

We will now use the Styles tab to size the Repeater and Repeater items.

Steps

1. On the page, click the **Birthday Repeater**.
2. In the **Styles** tab, set the **Width** to **100%**.
3. Click the **Birthday Repeated item**.
4. In the **Styles** tab, set the **Width** to **25%**.
This will fit four cards across within the Repeater container.
5. Click **Save**.
Your page should now look like the following layout.

Adding Birthdays

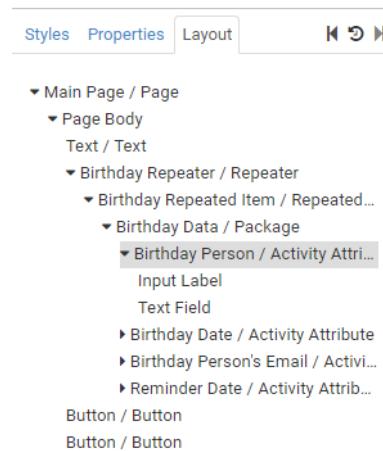
Birthday Person

Birthday Date

Birthday Person's Email

Reminder Date

- In the **Layout** tree, expand the tree by clicking the black caret. Then, click **Birthday Person**.



- In **Styles**, locate **Padding**. Add vertical padding of **10** above and below the Attribute.
- Repeat steps 6 and 7 for each Attribute: **Birthday Date**, **Birthday Person's Email** and **Reminder Date**.

The padding creates space between each Attribute.

Birthday Person

Birthday Date

Birthday Person's Email

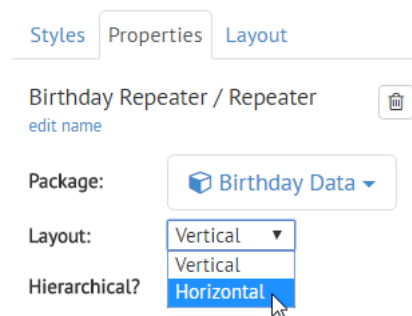
Reminder Date

Repeat Items Horizontally in the Birthday Repeater

By default, all Repeaters are set to have their internal repeated items to repeat vertically. In this page design we want the cards to repeat *horizontally* across the page.

Steps

1. Click the **Birthday Repeater**.
2. In the **Properties** tab under **Layout**, ensure **Horizontal** is selected



This sets the items to repeat horizontally in the Birthday Repeater.

3. Click **Save**.

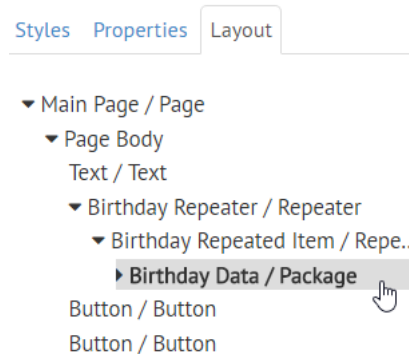
Create a Border

We also need to give the repeated Packages a border.

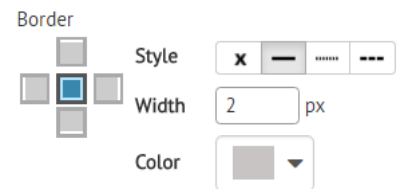
Steps

1. Click the **Birthday Data** Package page object that sits within the **Birthday Repeated Item**.

If you can't select this by clicking it on the page, click on it in the **Layout** tab tree structure.



2. In the **Styles** tab, scroll down to the **Border** setting.
3. Set the border **Style** to solid, **Width** to 2 pixels and **Color** grey.



We have chosen a simple grey color. Later in the tutorial we will be adding more colors to the cards.

4. Click **Save**.

Don't stop exploring here! You can set up different background colors for the Repeater or page, change the text, or Attribute data input sizes.

However, be aware that most professional user interface designers use subtle color palettes when designing forms and controls. You should avoid using strong primary colors for backgrounds and generally use black as your text color.

Before you created the Repeater, you added rows to your Table by clicking the **Plus** button. Now that you have removed the Table there is no **Plus** button. Also, the **Add** and **Delete** buttons are not automatically generated for Repeaters. In the next tutorial you're going to learn another new skill - how to create User Input Actions.

5. In the lower-right corner, click **Close** to exit Page Designer.

Tutorial Three - Creating User Input Actions

Learning Objectives



- ▶ User Input Actions
- ▶ Modifying Buttons

About User Input Actions

User Input (UI) actions are unique to the *User Input* Activity. An example of a UI action is a cancel button on a page. When the cancel button is clicked, the UI action attached to this button is run, and the Activity is canceled. UI actions are generally attached to a page object, so when that page object is clicked, the UI action will run.

You will need to create two UI actions here:

- Adding the Birthday Data Package Repeater.
- Deleting the Birthday Data Package Repeater.

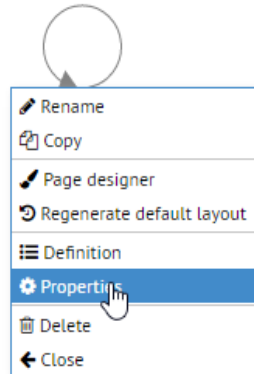
Create a UI Action for Adding the Birthday Package Repeater

In the last tutorial, you saved your work and exited from the Page Designer. You were redirected back to the All Birthdays Pattern flow. We will now create a UI action for adding the Birthday Package Repeater.

Steps

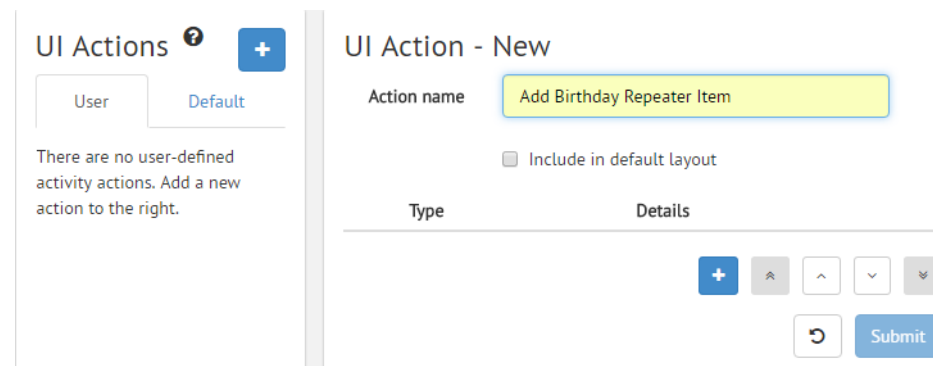
1. Ensure the All Birthdays Pattern is highlighted in the Organizational Explorer. If not, click **Patterns** and navigate to the All Birthdays Pattern.

- On the stage, hover over the Adding Birthdays Activity, click the **List** icon and then click **Properties**.



The Adding Birthdays **Activity Properties** page displays.

- In the **Activity Properties** page, scroll down to the **UI Actions** section.
This is where you add and edit UI actions you have made or those that have been created by default.
- Locate **UI Action - New**.
- In **Action name**, type **Add Birthday Repeater Item**.



- Click the blue **Plus** button below the field.
A new component is created for this action.
For every UI Action component you must define the component type and details.
- In **Type**, click **Other system action**.
- In **Details**, click **Add Repeater item**.

A small blue **Box** icon appears below **Details**. This indicates that you need to choose which Activity Package the Repeater item action is attached to.

9. Click the blue **Box** icon.
The Package selection form displays.
10. Click the **Birthday Data** Package.

Select

11. Click **Submit**.
12. In **UI Action New**, click **Submit** to save.
The **Add Birthday Repeater Item** displays in **UI Actions**.

You have now successfully made a UI Action. You will need to create another one for deleting the Repeater items.

Have a go at creating a second UI Action using the method show in the steps 4 to 12 above, except:

To begin, in **UI Actions**, click the blue **Plus** button.

-> The **UIs Action - New** field displays.

At step 5, type **Delete Birthday Repeater Item**.

At step 8, click **Delete Repeater item**.

Adding and Modifying Buttons

Once both UI Actions are created and saved, we will need to put them on the page layout. This requires adding buttons to the layout and attaching the UI Actions as click actions to the buttons.

Add a Button in Page Designer

Steps

1. From the top of the workspace, click **Page Designer**.

The Page Designer workspace displays.

2. Click **+ Page Object**.

The object panel displays.



3. In the second row, click **Button**.

Button is added to the page.

The screenshot shows a form titled "Adding Birthdays" with the following elements:

- A text input field labeled "Birthday Person" with a placeholder "Birthday Person".
- A text input field labeled "Birthday Date".
- A text input field labeled "Birthday Person's Email" with a placeholder "Birthday Person's".
- A text input field labeled "Reminder Date".
- A horizontal scrollbar below the input fields.
- Three blue buttons at the bottom: "Submit", "Cancel", and "Button".

We want the **Add Birthday** button to be positioned *outside* of the Repeater container.

- **NOTE:** If you select a page object and then choose to add a new object, the new object will appear directly after the selected object. Give it a try, click on the **Adding Birthdays** text object so that its border is highlighted blue. Now add in a button, it will be placed on the page directly after the text. Delete the button by clicking the Trash can in the Styles tab.

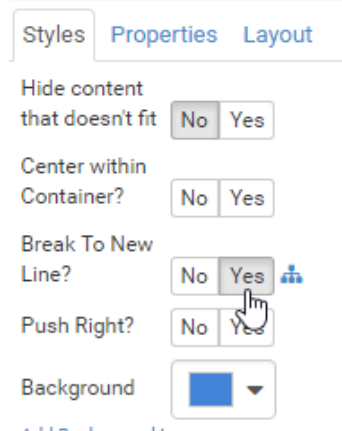
Modify a Button Using the Styles Tab

The button should be positioned as described in step 3 above. Now we will make a few adjustments to it to improve the page design.

Steps

1. Click on the button.
Its border is highlighted blue.

2. In the **Styles** tab, scroll down to **Break to New Line** and click **Yes**.



The button moves down to the next line.

3. In **Padding**, type **5** for each border edge.
The white space around the button increases by 5px.

4. Click **Save**.

You can explore the styles tab a little more and modify the border color, the background color, or the text.

Next we will add a button name and attach a UI action.

Add a Button Name and Attach a UI Action

Now we will use the properties tab to add a button name **Add a Birthday** and attach a UI Action.

Steps

1. Ensure the button is selected and click the **Properties** tab.
2. Click **edit name** and in **Name**, type **Add a Birthday**.
3. In **Click Action**, click **Add Birthday Repeater Item**.

This action is now attached to this button.

► **NOTE:** You will see that the Click Action list contains all the default UI actions that were created for this action, as well as the two actions that you created.

4. In **Button Text**, type **Add a Birthday**.

Add a Birthday / Button

done editing name

Name:

Click Action:

Button Text:

Icon:

- In the **Layout** tab, move the button to the top of the page. Position it directly below the Text.
- Click **Save**.

▼ Main Page / Page

- ▼ Page Body
 - Text / Text
 - Add a Birthday / Button**

- In the **Styles** tab, use the **Width** field to increase the width to **120px**.
- In the **Styles** tab, set the **Center within Container?** option to **Yes**.
- In the **Styles** tab, use the **Background** color drop-down to change the color to light blue.

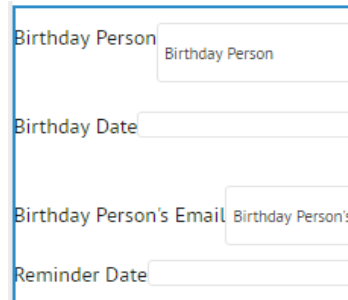
10. Click **Save**.

Create a button for the UI action Delete a Birthday

Now we will create a button for the UI action **Delete a Birthday Repeater Item**. We will name it **Delete**. When you are adding the button for this action, you must add it within the Birthday Repeated item. This ensures that every Birthday Data row has its own individual delete button.

When the button is clicked, it will only delete the Birthday Data row that it is contained within the Birthday Repeated Item.

1. In the **Layout** tab, click **Birthday Repeated Item**.
The Birthday Repeated Item is highlighted in blue.



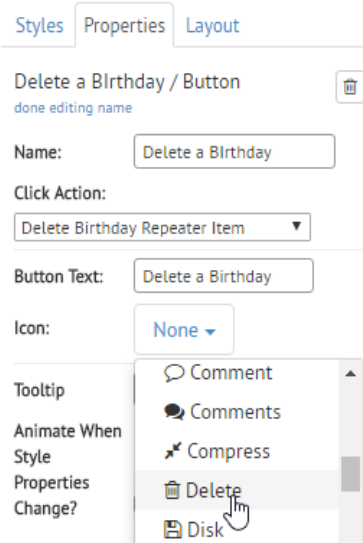
2. Click **Add Page Object** and then click **Button**.



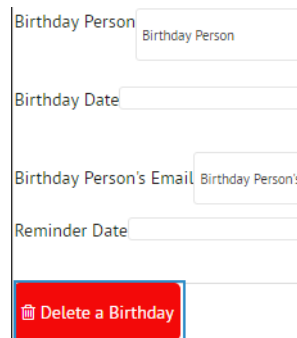
A button is added to the Birthday Repeated Item.

3. In the **Styles** tab, set the use the **Background** color drop-down to change the color to red.
4. Ensure the button is still selected and click the **Properties** tab.
5. Click **edit name** and in **Name**, type **Delete a Birthday**.

6. In **Click Action**, click **Delete Birthday Repeater Item**.
This action is now attached to this button.
7. In **Button Text**, type **Delete a Birthday**.
8. In **Icon**, click **Add an Icon** and click the **Delete** trash can icon.



9. In **Styles**, set a width of 130px. The changes are applied to the button.



10. Click **Save**.

Modify the Submit Button

We will now modify the **Submit** button that is under the Repeater.

This button submits the Activity and saves all the data you have entered on the form. However, right now it's a little hidden so we're going to make some adjustments to it.

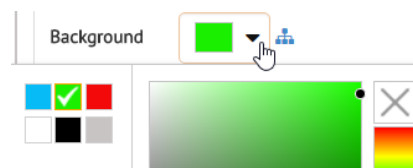
Steps

1. Click on the **Submit** button.
2. In the **Properties** tab, click **Edit Name** and in **Name**, type **Submit**.
3. In the lower-right corner, click **Save**.
4. In the **Layout** tab, move the **Submit** button below the **Add a Birthday** button.

Styles Properties Layout ↩ ↻

▼ Main Page / Page
 ▼ Page Body
 Text / Text
 Add a birthday / Button
 Submit button / Button
 ▶ Birthday Repeater / Repeater

5. In the **Properties** tab, use the **Button Text** field to change the text to **Save Birthdays**.
6. In the **Styles** tab, use the **Width** field to increase the width to 130px.
7. In the **Styles** tab, set the **Center within Container?** option to **Yes**.
8. In the **Styles** tab, use the **Background** color drop-down to change the color to green.



9. Click **Save**.
 We can remove the Cancel button from the form as it is not needed for this Activity.
10. Click the **Cancel** button, then in the **Properties** tab, click the trash can icon.

A confirmation message displays.

11. Click **OK**.
12. On the Stage, click on the Adding Birthdays text and edit it to read **Birthday Management**.
13. In the **Styles** tab, set **Width** to **30%**, scroll down and set the **Font Size** to **20pt**, and set **Center within Container?** to **Yes**.
14. In the lower-right corner of the Page Designer workspace, click **Save**.
Your page elements should now be looking better.

The screenshot shows a mobile app interface titled "Birthday Management". At the top right, there are two buttons: "Add a Birthday" (cyan) and "Save Birthdays" (green). Below these is a form with four input fields, each with a calendar icon to its right: "Birthday Person", "Birthday Date", "Birthday Person's Email", and "Reminder Date". Each field has a placeholder text that matches the field name. At the bottom left of the form area is a red button with a trash icon and the text "Delete a Birthday".

Now that your page design has progressed, click on the URL in Pattern Properties to see how the changes you've made look and to make sure it all works.

The Birthdays that you added in Tutorial One should appear now as Repeaters. You can add new Birthdays by clicking on the **Add a Birthday** button to show a new Repeater.

Tutorial Four – Defining Application Rules

Learning Objectives



- ▶ Defining Rules
- ▶ The Rule Builder Workspace
- ▶ The Complex Expression Builder
- ▶ Building a Rule

Defining Rules

Now that you've got the page elements looking and working better, let's improve the application functionality. We are going to add in a rule. Rules are defined pieces of logic that work to perform an action in the Activity. Rules can be added to any Activity.

Before we actually get into the process of building out rules, let us find out what this rule is going to do first. In the Birthday Data Package there is currently a Reminder date Attribute. Up until now we've had to manually set this date. Let's automate this so the reminder date is always set to two days before the person's birthday.

When entering a person's birthday, you may not know the year, you may only know the date and the month. In other instances, you may know the date, month and year the person was born, and you can enter this data in. Either way, the rule that we will create configures the Reminder date to be the correct date for the current year you are in. Therefore, you'll never lose track of a person's upcoming birthday.

Structure of This Rule

The structure of this rule will be as follows:

Expression Part 1

- Set the Reminder date equal to be **two days before** the birthday date.
- Set the Reminder date to the current year.

Therefore, the Reminder date will be two days before the birthday of this year.

Expression Part 2

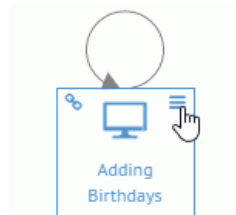
But what about if you enter in someone's birthday from earlier this year? You'll want a reminder date, but you will want it to be set for the following year, when the person's birthday occurs again. We'll set up a special condition in the rule that will set the Reminder date year to the next year, if the birthday day is before the current day it is entered in.

Access the Rule Builder

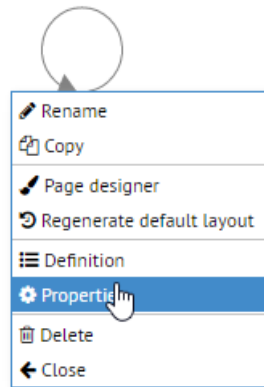
First we will select the Activity Properties page as shown in steps 1 and 2 below.

Steps

1. In the Organizational Explorer, click the **All Birthdays** Pattern.
The **Adding Birthdays** Activity displays on the Stage.
2. Click the **List** icon.



3. Click **Properties**.



4. Scroll down to **Rules** on the Properties page.

The Rules pane lists all the rules created for a single Activity. Right now, it is empty.

5. Click **Create rule** to launch the Rule Builder.



The Rule Builder displays.

Understanding the Rule Builder Workspace

Because this is your first time in the Rule Builder, let's examine the workspace.

The Rule Builder workspace comprises three sections, as shown below.

Top Panel (1)

The top panel contains all the general information about the rule you are creating. This information includes the **Rule title**, **Rule Description** and the timing of when this rule should run in the Activity. It's always good practice to give your rule a title. A title makes the rule easily identifiable in the Rules section of the Properties page. If a rule is not given a title, the rule statement will simply be substituted for the title of the rule. This can make it hard to differentiate between rules.

Left Panel (2)

The left panel contains the rule statements. Rule statements are used to compose a rule. When a statement is selected, this panel is replaced with either of the following parameters:

- Another set of statements
- or
- An input or selection option (dependent on the logic path that has been taken).

Right Panel (3)

The right panel is where your rule statement will form and be displayed. As you add in statements and other values, the rule is composed in this section.

Define a Rule

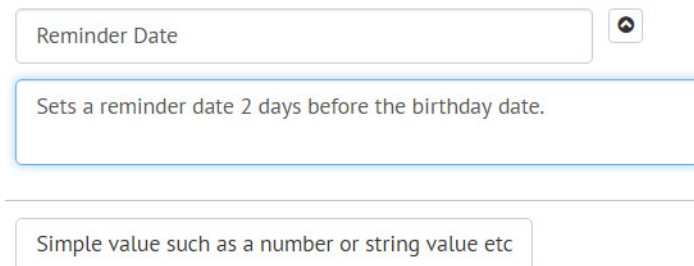
Now that you are in the Rule Builder workspace, let's start to define our first rule and introduce you to the Complex Expression Editor.

Add the Reminder Date

We will add the Reminder Date in the following steps.

Steps

1. In **Rule Title**, type **Reminder Date**.
2. In Rule Description, type **Sets a reminder date 2 days before the birthday date**.



The screenshot shows the Rule Builder interface. The 'Rule Title' field contains the text 'Reminder Date'. The 'Rule Description' field contains the text 'Sets a reminder date 2 days before the birthday date.' Below these fields, there is a section for 'Simple value such as a number or string value etc' which is currently empty.

3. In the left panel, click **Set the value of one or more Attributes**.
The statements options change in the left panel.
4. Click **Assign a value or expression to an Attribute**.
Activity content displays in the left panel and **assign** displays in on the right.
5. Click **Reminder Date Attribute** and click **Add**.

The screenshot shows the Rule Builder interface. At the top, there is a text box labeled 'Reminder Date' with a circular icon to its right. Below it, a larger text box contains the description: 'Sets a reminder date 2 days before the birthday date.' To the right of these boxes is a checkbox labeled 'Data' which is checked. Below the description box is a section titled 'Activity' with a blue 'Add' button. Under 'Activity', there is a dropdown menu currently set to 'Auto'. Below the dropdown is a list of attributes under the 'Birthday Data' category: 'Birthday Person', 'Birthday Date', 'Birthday Person's Email', and 'Reminder Date'. The 'Reminder Date' attribute is highlighted with a blue background. To the right of this list is a grey area labeled 'assign'.

The Reminder Date Attribute is added to the rule expression.

The screenshot shows the rule expression editor. It displays the text 'assign' followed by a blue box containing the attribute path 'Birthday Data.Reminder Date', followed by an equals sign '=' and a small blue box.

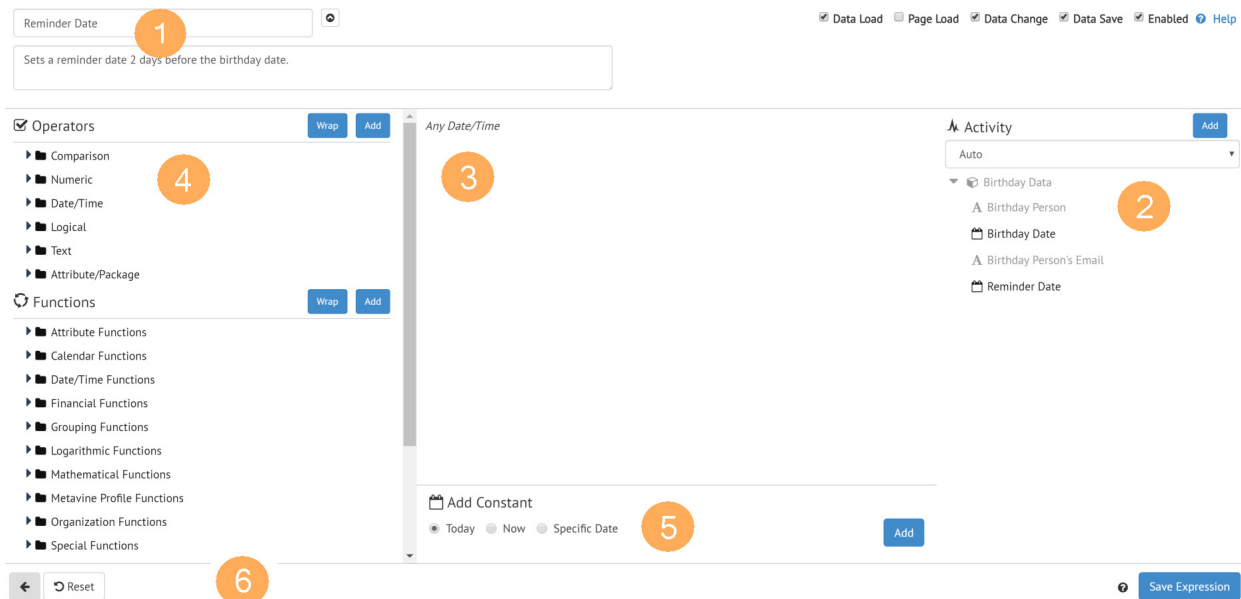
6. Click **Expressions involving functions, formulas or more than two Attributes**.

Selecting this statement opens the Complex Expression Editor. Let's take a break from constructing the rule here and introduce you to the Complex Expression Editor.

The Complex Expression Builder Workspace

This Rule Builder tool allows you to construct complex expressions to insert into your rules using a variety of operators and Functions.

The Complex Expression Editor is divided into six panels, as shown below.



1. The top panel displays the general rule information including the **Rule title**, **Rule Description** and **run settings**. You can hide this panel by clicking the **Up arrow** next to the Rule Title.
2. The right-hand panel is the Activity content hierarchy. All the Packages and Attributes that are contained in the Activity are displayed here. You can insert an Activity Package or Attribute into an expression from here.
3. The center panel is where the expression statement is built. The Operators and Functions that you choose will form the expression statement here.
4. The Operators and Functions that are available to create the expression are listed and categorized in the left panel.

An **Operator** provides a construct for inserting arithmetic and logical operations into an expression. An Operator is the most basic form of a function string, in that it generally takes two inputs of the same type to perform an operation. For example, the **Addition** operator adds two values together.

A **Function** is used to create more complex constructs in an expression. A Function takes a series of inputs to return an output. You will see a folder structure that contains a range of specialized Functions such as Financial Functions, Mathematical Functions and Timespan Functions.

5. The **Add Constant** panel is used to insert a constant such as "1 Day" into an expression parameter.
6. The bottom panel has three buttons:

- **Arrow** returns to the rule without saving.
- **Reset** resets the expression.
- **Save** performs a save and inserts the expression into the rule.

Now that you know the basics of the Complex Expression Editor, we can get back to building the Rule.

Build a Rule - Part 1

In part 1, we will complete the following:

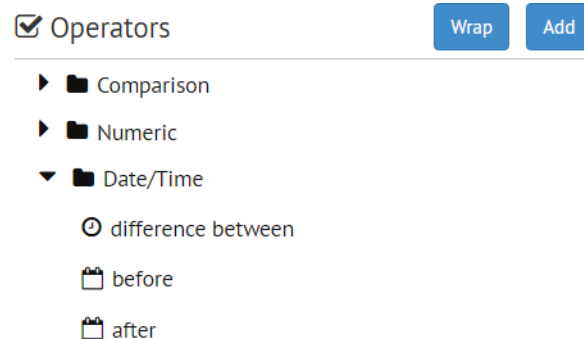
- Set the Reminder Date to be two days before the Birthday Date and set the year as the current year.

First we will add an operator called **before** and set it to be 2 days before the birthday date.

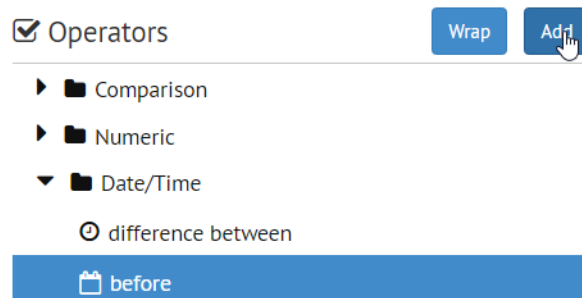
Steps

1. In the **Operators** panel, expand the **Date/Time** folder by clicking the black caret.

A list of Date/Time functions displays.



2. Click **before** and then **Add**.



The **time-span operand 1** is highlighted.

3. In the **Add Constant** panel, set a value of **2**, a period type of **day**, and click **Add**.

(**time-span operand 1** before *date/time operand 2*)

⌚ Add Constant

2 Period type ▾ Add

day
hour

The time-span operand 1 is replaced with a value of **2 days** and now the **date/time operand 2** is highlighted.

(2 days before **date/time operand 2**)

4. In **Functions**, expand **Calendar Functions**, click **Date** and then click **Add**.

🔄 Functions Wrap Add

▶ Attribute Functions

▼ Calendar Functions

📅 Date

The expression now reads **2 days before Date (Year, Month, Day))** with the Year highlighted.

5. In **Functions**, under **Calendar Functions**, click **Year** and then click **Add**.

In the function, **Date from which to extract year** is highlighted.

```
( 2 days before Date( Year Date from which to extract year , Month , Day ) )
```

6. In **Add Constant**, ensure **Today** is selected and click **Add**.

In the function, **today** is added as the year.

```
( 2 days before Date( Year today , Month , Day ) )
```

 Add Constant

☒ Today
 ☐ Now
 ☐ Specific Date

Add

7. In the function, click **Month**.
8. Under **Functions**, in the **Calendar Functions** list, click **Month** and then **Add**.

In the function, **Date from which to extract month** is added.

```
( 2 days before Date( Year today , Month Date from which to extract month , Day ) )
```

9. In the **Activity** content panel, click **Birthday Date**, and then **Add**.

```
( 2 days before
Date( Year today ,
Month Date from which to extract month , Day )
)
```

 Add Constant

☒ Today
 ☐ Now
 ☐ Specific Date

Add

 Activity

Add

Auto ▾

 Birthday Data

 Birthday Person

 **Birthday Date**

 Birthday Person's Email

 Reminder Date

☐ Invite ?

☐ Email Sent?

10. In the function, the Birthday Date is inserted as the date to extract the month, click **Day** to highlight it.

(2 days before Date(Year today , Month *Date from which to extract month* , **Day**))

11. Under **Functions**, in the **Calendar Functions** list, click **Day** and then **Add**. In the function, **Date from which to extract day** is added.

(2 days before
Date(Year today , Month  Birthday Data.Birthday Date , Day **Date from which to extract day**))

12. Once again, in the **Activity** content panel, click **Birthday Date**, and then **Add**.

In the function, the Birthday Date is inserted as the date to extract the day.

(2 days before
Date(Year today , Month  Birthday Data.Birthday Date , Day  Birthday Data.Birthday Date))

13. Click **Save Expression**.

The first part of the rule is now complete and should read as follows:

assign  Birthday Data.Reminder Date =
2 days before Date(Year today , Month Birthday Data.Birthday Date , Day Birthday Data.Birthday Date)

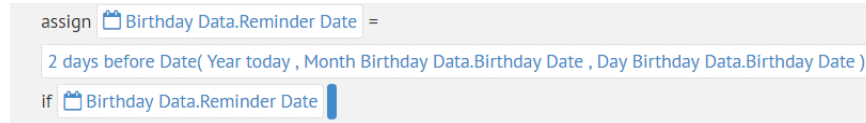
Build a Rule - Part 2

In part 2, we will accommodate the special case of if the Birthday date is before today. If the birthday date is before today, the reminder date will need to be set to two days before the birthday date *in the following year*.

Steps

1. In the left panel of the Rule Builder, click **Specify a condition to this rule (e.g. if Price is greater than \$0.00 then...)**
The statements options change in the left panel.
2. Click **If condition is met perform one or more actions otherwise perform one or more other actions**.

3. Click **Simple condition involving only one or two Attributes.**
4. From the Activity content, click the **Reminder Date** Attribute and then click **Add.**



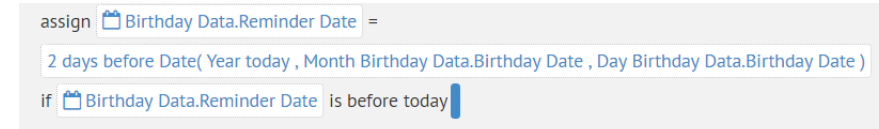
assign  Birthday Data.Reminder Date =
 2 days before Date(Year today , Month Birthday Data.Birthday Date , Day Birthday Data.Birthday Date)
 if  Birthday Data.Reminder Date

5. Click **is before.**
 The statement is added to the expression.

6. Click **A date and/or time value.**

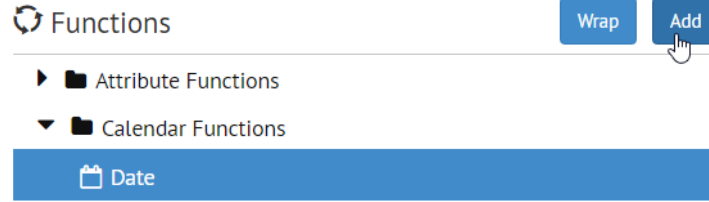
7. Click **The current date or time.**

8. Click **today.**
 Today is added to the expression.



assign  Birthday Data.Reminder Date =
 2 days before Date(Year today , Month Birthday Data.Birthday Date , Day Birthday Data.Birthday Date)
 if  Birthday Data.Reminder Date is before today

9. Click **End of Condition.**
 Now that the condition is spelled out we need to define what should be done if this condition is met. If the condition is met we'll be changing the year of the reminder date to be one year after today.
10. Click **Set the value of one or more Attributes.**
11. Click **Assign a value or expression to an Attribute.**
12. From the **Activity** content, click **Reminder Date** Attribute and then click **Add.**
 The Attribute displays in the expression.
13. Click **Expressions involving functions, formulas or more than two Attributes.**
 The Complex Expression Editor displays.
14. In **Functions**, expand the **Calendar Functions** folder, click **Date** and then click **Add.**

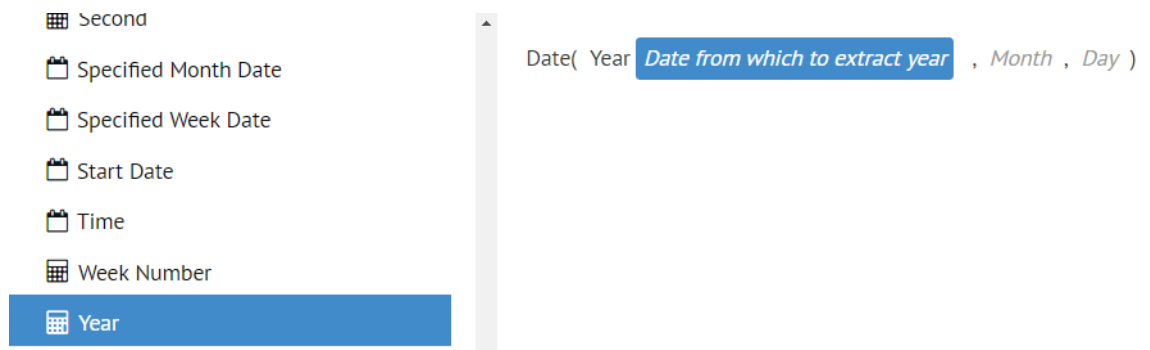


This time, we will set the year to equal one year after today.

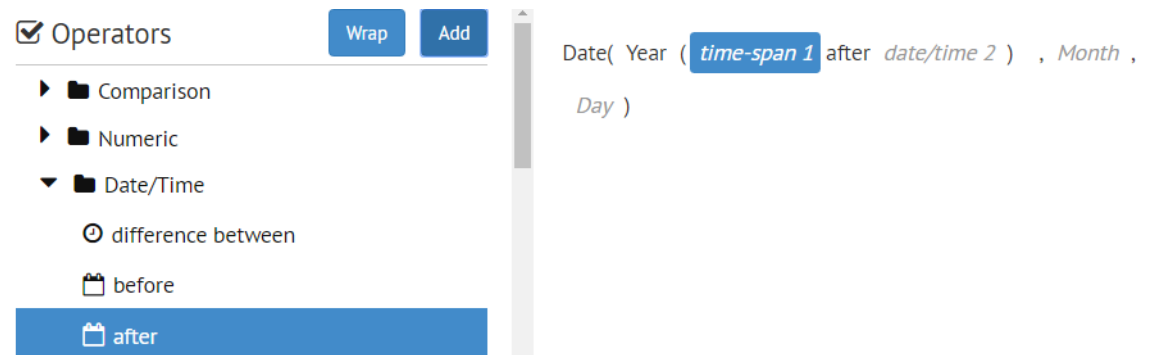
Doing this will set the Reminder date to be two days before the birthday in the following year, if the birthday is before today in the current year.

15. With **Year** highlighted, scroll through the **Calendar Functions** list, click **Year** and **Add**.

The Year function is inserted into the expression.



16. In the **Operators** panel, expand the **Date/Time** folder, click **After** and then click **Add**.



17. In **Add Constant**, set to **1**, **Period type** as **Year**, and click **Add**.

Date(Year (**time-span 1** after *date/time 2*) ,
Month , *Day*)

🕒 Add Constant

18. In **Add Constant**, set to **Today** and then click **Add**.

Date(Year (1 year after **today**) , *Month* , *Day*)

📅 Add Constant

☒ Today ☐ Now ☐ Specific Date

19. In the string, click **Month**.
20. Under **Functions**, in the **Calendar Functions** list, click **Month** and then **Add**.

It is inserted into the rule.

21. In the Activity content panel, click **Reminder Date**, and then **Add**.

The reminder date is inserted as the date to extract the month.

Date(Year (1 year after today) , Month **📅 Birthday Data.Reminder Date** , *Day*)

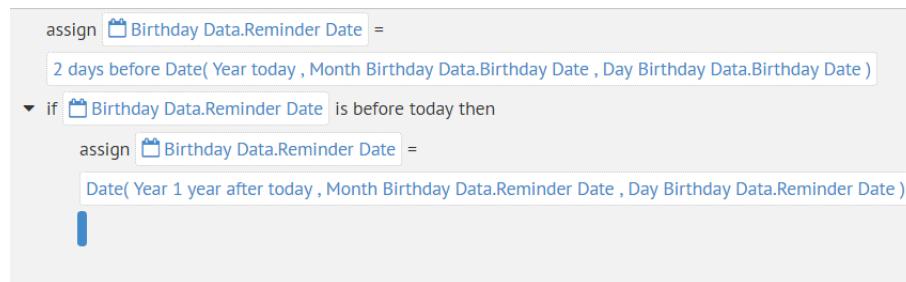
22. In the function, click **Day**.
23. Under **Functions**, in the **Calendar Functions** list, click **Day** and then click **Add**.

24. Once again, click the **Reminder date** and **Add** to set the date from which to extract the day for.



25. Click **Save Expression**.

Your rule should now read as below.



26. Click **End of statement list**.

27. Click **End of if Otherwise statement**.

28. Click **End of statement list**.

Rule Complete now appears in the left-hand panel.

29. In the lower-right corner, click **Save Rule**.

The Save Rule message confirms your save and the rule appears in the rules list.

You've now completed your first rule in Design Studio. Do you want to see it work in action? We'll test that out now!

Test the Reminder Date Rule

You can test the Reminder Date rule as follows:

1. In the Organizational Explorer, select the **All Birthdays** Pattern.
2. Click **Pattern Properties**.

3. In **Details**, click on the Pattern **Development URL**.
The Birthday Management screen displays.
4. First, try adding a birthday that is *after* the current day.
As per the rule you should see the reminder date set to be *two days before* that date this year.
5. Now try selecting a birthday date that was earlier this year that is *before* the current date.
This time, you should see the reminder date set to two days before the birthday, but in the following year.

Tutorial Five - Adding an Email Reminder Pattern

Learning Objectives



- ▶ Creating an Automated Activity
- ▶ Creating an Email Activity
- ▶ Setting up a Splitting Connector
- ▶ Setting up an Event in Calendar

About the Email Reminder Pattern

In this tutorial we will add a Pattern to the application that sends a reminder email containing a list of upcoming birthdays to recipients. The email body will contain the text literal 'People whose birthdays are coming up:' followed by a list of upcoming birthday names.

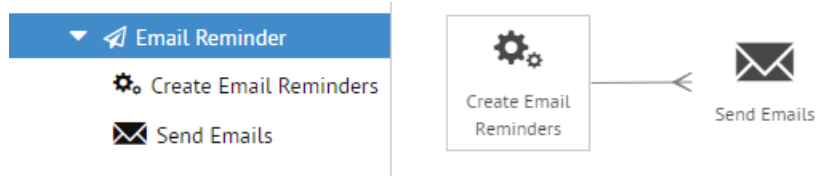
The reminder email recipients will be all those who have been added to the Birthday App. We will exclude recipients who have an upcoming birthday - we don't need to send a reminder email to those people who are on the upcoming birthday list.

To achieve this outcome, we will need to perform the following tasks:

- Create an **Email Contents** Package to hold the contents of the email.
- Create a **Calendar** event to schedule the daily email reminders
- Create an **Email Reminder** Pattern to manage the email reminder process.

The **Email Reminder** Pattern will contain the components described below.

- An Automated Activity called **Create Email Reminders** that will generate the email content.
- An Email Activity called **Send Emails** with a **Splitting Connector** that will retrieve email content and send out emails.
- A new Package called **All Birthday People** that will determine the upcoming birthdays list.



This combination of Package logic, Activities and scheduling will make a daily check against reminder dates and send out reminder emails accordingly.

Create the Email Contents Package

First we need to create a Package to hold the email contents. We'll call this Package Email Contents and make two Attributes within it called **Email Body** and **To Address**.

Steps

In the following steps we will create a new Package called Email Contents.

1. In the Organizational Explorer, click **Packages**.
2. In the lower-right corner of the Organizational Explorer, hover over the **Plus** button.
3. Click New **Package**.

The new Package **Details** pane displays on the Stage.

4. In **Name**, type **Email Contents**.

Details

Name	Email Contents
Name in UI	Name in UI
Package type	☒ Data ☐ Reporting
Parent package	None

Now we will add two Attributes in Attributes.

5. In **Attributes**, click in **Attribute name** and type **Email Body**.
6. In **Data type**, click **Memo**.

Attributes

Attribute name	Data type
Email Body	Text ▼ Text Number Date/Time True/False Memo Document Time-span
Add attribute	

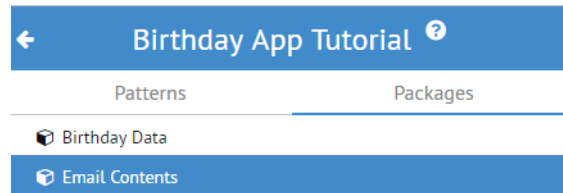
7. Click the **Add Attribute** button.
- A new Attribute name field displays.
8. In **Name**, type **To Address**.
9. Leave the **Data type** set as **Text**.

Attributes

Attribute name	Data type
Email Body	Memo
To Address	Text
Add attribute	

10. Click **Submit**.

The Email Contents Package displays



11. Click on the **Patterns** tab.

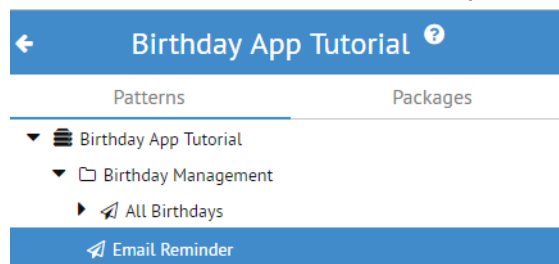
Create the Email Reminder Pattern

Now we will create an Email Reminder Pattern that will manage emails for upcoming birthdays.

Steps

1. In **Patterns**, click the **Birthday Management** folder, so that it is highlighted.
2. Hover over the **Plus** button in the lower-right corner of the Organizational Explorer and click **New Pattern**.
3. In **New Pattern - Details**, name the Pattern **Email Reminder** and in the upper-right corner, select **Submit**.

The Email Reminder Pattern displays in the Organizational Explorer.



Creating an Automated Activity

In Tutorial One, we created and worked on a single User Input Activity. This allowed us to create a functioning Birthday Management application that requires user interaction. Now we will create an Automated Activity.

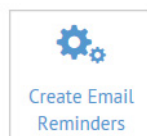
An **Automated Activity** runs in the background of your application. It executes any rules and processes associated with it without the need for user interaction. We'll use the combination of the Automated Activities, Connectors and the Email Activity to send out reminder emails.

Create an Automated Activity called Create Email Reminders

Now we will create the Create Email Reminders Automated Activity in the Email Reminder Pattern flow.

Steps

1. In **Patterns**, click **Email Reminder**.
The Email Reminder Pattern flow displays on the Stage.
2. From the top-right of the Stage, drag the **Automated Activity** icon (gears) onto the Email Reminder Pattern flow.
A form displays the basic details for the new Automated Activity
3. In **Name**, type **Create Email Reminders** and click **Submit**.
You will now see your Activity in the Pattern flow represented by the Automated Activity icon and labeled with the Activity title beneath it.
4. Hover over the Activity icon and click on the **List** icon (in the top-right corner).
5. From this drop-down list, click **1st Mark first**.
A grey square surrounds the Activity icon to indicate that it is the first Activity to run when the Pattern initially loads.

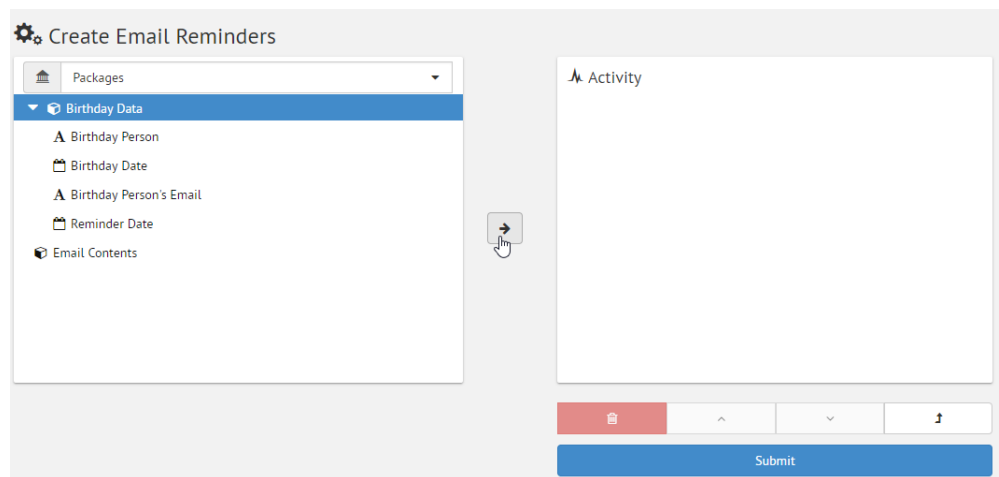


Create the All Birthday People Package

Now we will create the All Birthday People Package in the Create Email Reminders Automated Activity.

Steps

1. On the Stage, hover over the **Create Email Reminders** Activity and in the upper-right corner, click the **List** icon.
2. Click **Definition**.
The Create Email Reminders Activity definition transfer page displays.
3. In Packages, click **Birthday Data**, and then click the **Right-arrow** to transfer the Package.



The Source Package Details form displays.

4. In **Package name**, change the name from Birthday Data to **All Birthday People** and click **Submit**.

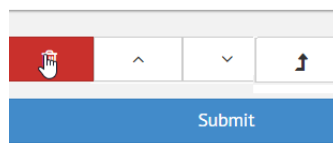
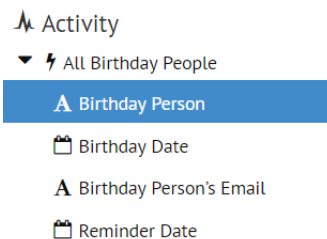
Source Package Details

Package name	All Birthday People
Name in UI	Name in UI
	☐ No name in UI ☐ Suppress display of attributes in UI
Packages to include	☒ Linked packages ☒ Sub-packages
Attributes to include	☐ Only key attributes ☐ Only search attributes ☒ All attributes
Transfer at top-level	☒

← Submit

The Package transfers.

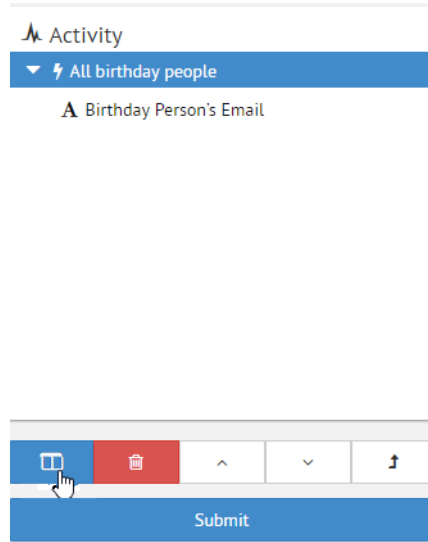
5. Within the Activity column, highlight the **Birthday Person** Attribute. To delete this unnecessary Attribute, below the Activity column, click the **trash can** button.



6. Repeat step 5 for **Birthday Date** and **Reminder Date** so that only **Birthday Person's Email** remains.
7. Below the **Activity** column, click **Submit**.

We will now define how the All Birthday People Package behaves in this Activity. This Package is not creating or updating data rows - it is retrieving existing birthday people rows.

8. Below the **Activity** column, click the **Open Book** button.



9. Click **Create NEW ROW(S)** and **UPDATE ROW(S)** to clear the selections. Create NEW ROW(S) and UPDATE ROW(S) change from blue to clear.
10. Leave **DELETE ROW(S)**, **SELECT ROW(S)** and **REQUIRE ROW(S)** clear.
11. Click the **Multiple rows?** check box so that it is set. There are multiple birthday people in the application.
12. Ensure the **Retrieve existing rows** check box is set. All existing birthday people in the application will be retrieved.

13. Check that the **Restrict to** option check boxes are unchecked. We do not want any restrictions.

14. Under the **Activity** column, click **Submit**.
15. Then, under the **Activity** column, click **Plus**.
16. The Activity Definition Transfer page displays.

Next we will transfer the Email Contents Package into All Birthday People.

Transfer the Email Contents Package

The Email Contents Package holds the email contents. Now we will transfer the Email Contents Package into All Birthday People and set up a relationship named **Daily Email Reminder** with a cardinality of one-to-many. Cardinality is one-to-many because every day we will create a new email Package for the *many* users that we are sending a reminder to.

Steps

- **NOTE:** Before you transfer an Attribute, you must **FIRST** highlight the parent Package as shown in step 2 below. This ensures that the highlighted Package remains the Controlling Package in this Activity.
1. The Create Email Reminders Activity definition transfer page should be still displayed. Under **Packages**, click **Email Contents**.

2. In **Activity**, click **All Birthday People** to highlight it.
3. Click the **Right-arrow**.
4. The Source Package Details form displays with a Package name **Email Contents**.
5. In **New relationship name** (not in the Package name field), type **Daily Email Reminder**.
6. Set **Cardinality** as **One to many**.

Source Package Details

Package name	Email Contents
Name in UI	Name in UI
	☐ No name in UI ☐ Suppress display of attributes in UI
Packages to include	☒ Linked packages ☒ Sub-packages
Attributes to include	☐ Only key attributes ☐ Only search attributes ☒ All attributes
Transfer at top-level	☐
New relationship name	Daily Email Reminder
Cardinality	☐ One to one ☒ One to many

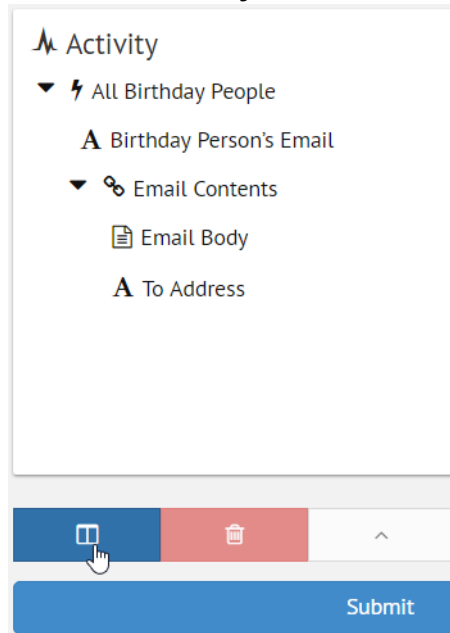
 

7. Click **Submit**.
8. Under the Activity column, click **Submit**.

A new email Package is created for the *many* users that we are sending a reminder to.

We will now define how the Email Contents Package behaves in this Activity.

9. Below the **Activity** column, click the **Open Book** button.



The Activity definition main page displays.

10. In the **Activity** column, click **Email Contents**.
11. In the right panel, click **Create NEW ROW(S)** to set this option.
This is the new email to create.
12. Click the **Multiple rows?** check box to clear this option.
Multiple is not necessary as only one email is created per person.
13. Check that the **Retrieve** option is set to **None**. If not, click **None** to set this option.

As this is a new email, there is nothing to retrieve.

Create Email Reminders

Activity

- ⚡ All Birthday People
 - A Birthday Person's Email
 - 📧 Email Contents**
 - 📄 Email Body
 - A To Address
 - 📅 Upcoming Birthday Data

Email Contents [edit name](#)

How can this package be altered? Packages related to the pagination-enabled package are non-persistent.

CREATE NEW ROW(S) **UPDATE ROW(S)** **DELETE ROW(S)** **SELECT ROW(S)** **REQUIRE ROW(S)**

Multiple rows? ☐

Relationship **Daily Email Reminders (Linked)**

Allow existing related rows to be removed ☐

Retrieve ☐ All existing rows ☐ Related rows ☒ None

Only submit ☐ Modified by user

Submit

- Under the **Activity** column, click **Submit**.
- Then, under the **Activity** column, click **Plus**.
The Activity definition transfer page displays.

Create the Upcoming Birthday Data Package

The reminder email contains a list of upcoming birthdays. This Package will determine which birthdays are included in the reminder email birthday list.

- The Create Email Reminders Activity definition transfer page should be still displayed. Under **Packages**, click **Birthday Data**.
- In Activity, click **Email Contents** to highlight it.

3. Click the **Right-arrow**.

The screenshot shows the 'Create Email Reminders' interface. On the left, under 'Global Packages', the 'orgname' folder is expanded, and 'Birthday Data' is selected. The right pane shows 'Activity' with 'All Birthday People' expanded, and 'Email Contents' selected. A right arrow button is visible between the panes. At the bottom are icons for a table, a trash can, and up/down arrows, followed by a 'Submit' button.

4. The Source Package Details form displays.
5. In **Package name**, change Birthday Data to **Upcoming Birthday Data**.
6. Click the **Create new relationship** check box to set it.
The New relationship field displays.
7. In **New relationship name**, type **Birthdays to Include**.
8. Set **Cardinality** as **One to many**.
There can be *many* birthdays to list in the reminder email.

Source Package Details

Package name	Upcoming Birthday Data
Name in UI	Name in UI
	☐ No name in UI ☐ Suppress display of attributes in UI
Packages to include	☒ Linked packages ☒ Sub-packages
Attributes to include	☐ Only key attributes ☐ Only search attributes ☒ All attributes
Transfer at top-level	☐
Create new relationship	☒
New relationship name	Birthdays to include
Cardinality	☐ One to one ☒ One to many

9. Click **Submit**.

10. Under the Activity column, click **Submit**.

We will now define how the Upcoming Birthday Data Package behaves in this Activity.

11. Below the Activity column, click the **Open Book** button.

Activity

- ▼ ⚡ All Birthday People
 - 📄 Birthday Person's Email
 - ▼ 🔗 Email Contents
 - 📄 Email Body
 - 📄 To Address
 - ▼ 🔗 Upcoming Birthday Data (Birthday Data)
 - 📄 Birthday Person
 - 📅 Birthday Date
 - 📄 Birthday Person's Email
 - 📅 Reminder Date

Submit

12. In the Activity column, click **Upcoming Birthday Data (Birthday Data)**.
13. Ensure that the **Multiple rows?** check box is set.
There can be multiple upcoming birthdays every day.
14. Ensure that **Retrieve** is set to **None**.

15. Leave all other options deselected.

Create Email Reminders

Activity

- ⚡ All Birthday People
 - 📧 Birthday Person's Email
 - 📧 Email Contents
 - 📧 Email Body
 - 📧 To Address
 - 📧 Upcoming Birthday Data (Birthday Data)**
 - 📧 Birthday Person
 - 📧 Birthday Date
 - 📧 Birthday Person's Email
 - 📧 Reminder Date

Upcoming Birthday Data [edit name](#)

How can this package be altered? Packages related to the pagination-enabled package are non-persistent.

+

CREATE NEW ROW(S)

✎

UPDATE ROW(S)

🗑️

DELETE ROW(S)

🔍

SELECT ROW(S)

✳️

REQUIRE ROW(S)

Multiple rows? ☒

Minimum rows required

Maximum rows required

Relationship Birthdays to Include (Linked)

Allow existing related rows to be removed ☐

Retrieve ☐ All existing rows ☐ Related rows ☒ None

Only submit ☐ Modified by user

Row(s) that fulfill the condition

R

Submit

16. Under the Activity column, click **Submit**.

Create a Rule to Generate Email Reminders

In the following steps, we will create a rule to generate the email reminders.

- **Steps 1 to 3** Create a rule.
- **Step 4** Name the rule **Generate Email Reminders**.
- **Steps 6 to 12** Create an email for each birthday person.
- **Steps 13 to 18** Set a filter condition where the birthday reminder date is equal to today.
- **Steps 19 to 25** Set a filter where upcoming persons email is not equal to birthday persons email.
- **Steps 26 to 35** For each birthday person add the email contents.
- **Steps 36 to 40** Add the birthday person's email to the **To address** field.
- **Steps 41 to 45** Add the text literal **People whose birthdays are coming up:** to the email body.

- **Steps 46 to 65** Add each upcoming birthday person's name to the email body.
- **Step 66** Save the rule.

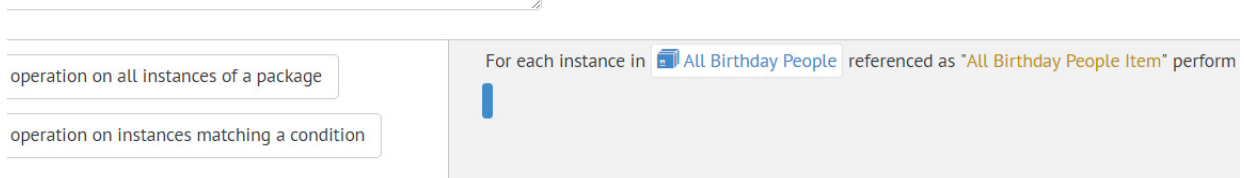
Steps

1. In the Organizational Explorer, click **Create Email Reminders**.
2. From above the stage, click **Activity Properties**.
3. Scroll to **Rules** and click **Create rule**.
4. In the **Rule Title** field, type **Generate Email Reminders**.
5. In the **Rule Description**, type **Generates reminders for upcoming birthdays**.

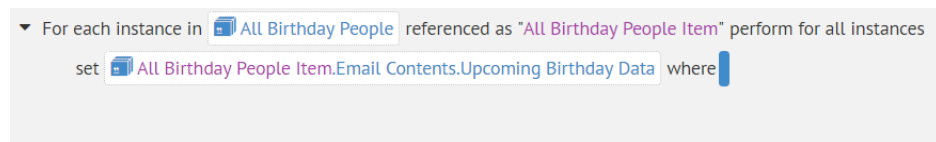
Now we will create an email for each birthday person.

6. Click **Select a Package instance**.
7. Click **Perform an operation on one or more Package instance(s)**.
8. In **Activity Packages**, click **All Birthday People** and then click **Add**.

The rule should now read:



9. Click **Perform an operation on all instances of a Package**.
10. Click **Select a Package instance**.
11. Click **Search for an instance of a Package based on a filter**.
12. In **Activity Packages**, click **Upcoming Birthday Data (Birthday Data)** and then click **Add**.



Now we will set a filter condition where the birthday reminder date is equal to today.

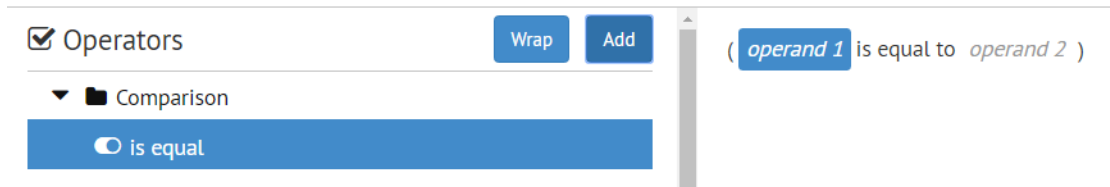
13. Click **Add a filter where all filter conditions must be met.**

14. Click **Add a filter to the current Package.**

The Complex Expression Editor displays.

15. In **Operators**, under **Comparison**, click **is equal** and then click **Add**.

Now **operand 1** is highlighted.

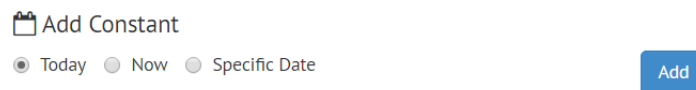


16. In **Activity**, expand **Upcoming Birthday Data (Birthday Data)**, click **Reminder Date**, and then click **Add**.

Now **operand 2** is highlighted.

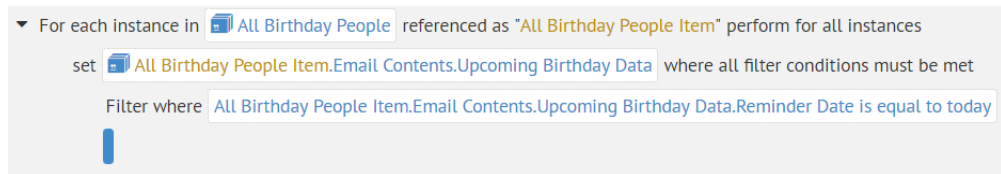
17. In **Add Constant**, **Today** is selected, click **Add**.

(
 All Birthday People Item.Email Contents.Upcoming Birthday Data.Reminder Date
 is equal to **today**)



18. Click **Save Expression**.

The rule should now read as follows:



Now we will set a filter where the upcoming persons email is not equal to birthday persons email.

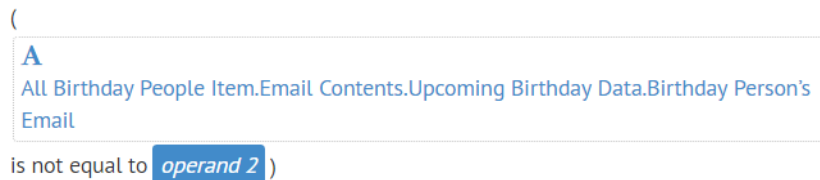
19. Click **Add a filter to the current Package.**

The Complex Expression Builder displays.

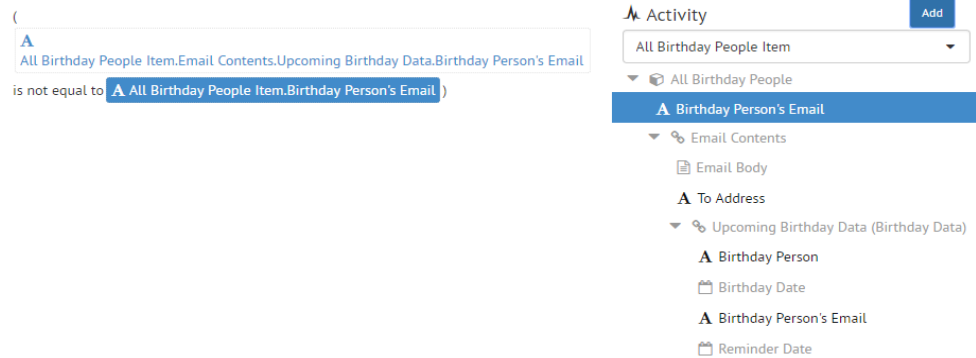
20. In **Operators**, under **Comparison**, click **is not equal** and then click **Add**.
Now **operand 1** is highlighted.

21. In **Activity**, expand **Upcoming Birthday Data (Birthday Data)**, click **Birthday Person's Email** and click **Add**.

Now **operand 2** is highlighted.



22. In **Activity**, under **All Birthday People**, locate **Birthday Person's Email** - where it appears directly under All Birthday People and click **Add**.



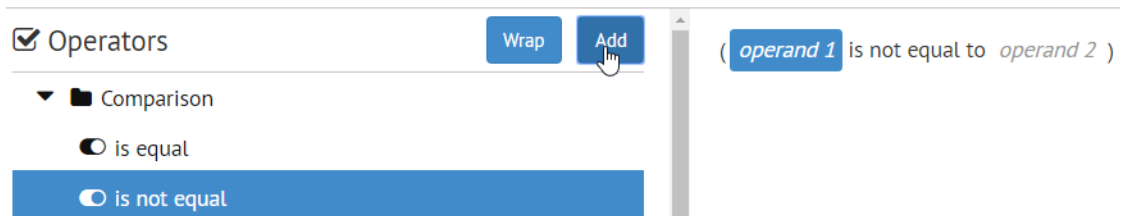
23. Click **Save Expression**.
24. Click **End of Filter Statement List**.
25. Click **End set statement**.

Now we will add the email contents for each birthday person.

26. Click **Specify a condition to this rule(e.g if Price is greater than \$0.00 then...)**.
27. Click **If condition is met perform one or more actions otherwise perform...**
28. Click **Conditions involving functions, formulas or more than two Attributes**.

The Complex Expression Builder displays.

29. In **Operators**, expand **Comparison**, click **is not equal** and then click **Add**.



30. In **Functions**, expand **Grouping Functions**, click **Count** and then click **Add**.

31. In **Activity**, expand **Upcoming Birthday Data**, click **Birthday Person** and **Add**.

32. Click **operand 2**.

33. In **Add Constant**, select **0** and click **Add**.

```
(
  Count
  All Birthday People Item.Email Contents.Upcoming Birthday Data.Birthday
  Person
  is not equal to 0 )
```

34. Click **Save Expression**.

35. Click **End of Condition**.

```
▼ For each instance in All Birthday People referenced as "All Birthday People Item" perform for all instances
  set All Birthday People Item.Email Contents.Upcoming Birthday Data where all filter conditions must be met
  Filter where All Birthday People Item.Email Contents.Upcoming Birthday Data.Reminder Date is equal to today
  Filter where
    All Birthday People Item.Email Contents.Upcoming Birthday Data.Birthday Person's Email is not equal to All
    Birthday People Item.Birthday Person's Email
  ▼ if Count All Birthday People Item.Email Contents.Upcoming Birthday Data.Birthday Person is not equal to 0 then
```

Now we will add the birthday person's email to the **To address** field.

36. Click **Set the value of one or more Attributes**.

37. Click **Assign a value or expression to an Attribute**.

38. In **Activity**, click **To Address** and click **Add**.

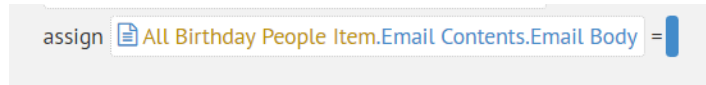
39. Click **An Attribute**.

40. In **Activity**, click **Birthday Person's Email** (where it first appears in the tree) and then click **Add**.

```
▼ For each instance in All Birthday People referenced as "All Birthday People Item" perform for all instances
  set All Birthday People Item.Email Contents.Upcoming Birthday Data where all filter conditions must be met
  Filter where All Birthday People Item.Email Contents.Upcoming Birthday Data.Reminder Date is equal to today
  Filter where
    All Birthday People Item.Email Contents.Upcoming Birthday Data.Birthday Person's Email is not equal to All
    Birthday People Item.Birthday Person's Email
  ▼ if Count All Birthday People Item.Email Contents.Upcoming Birthday Data.Birthday Person is not equal to 0 then
    assign All Birthday People Item.Email Contents.To Address =
      All Birthday People Item.Birthday Person's Email
```

Now we will add the text literal **People whose birthdays are coming up:** to the email body.

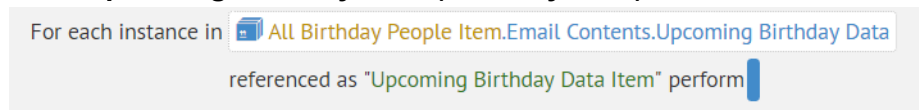
41. Click **Set the value of one or more Attributes.**
42. Click **Assign a value or expression to an Attribute.**
43. In Activity, click **Email Body** and then click **Add.**



44. Click **Simple value such as a number or string value etc.**
45. In the text box, type **People whose birthdays are coming up:** and click **Submit.**

Now we will add each upcoming birthday person's name to the email body

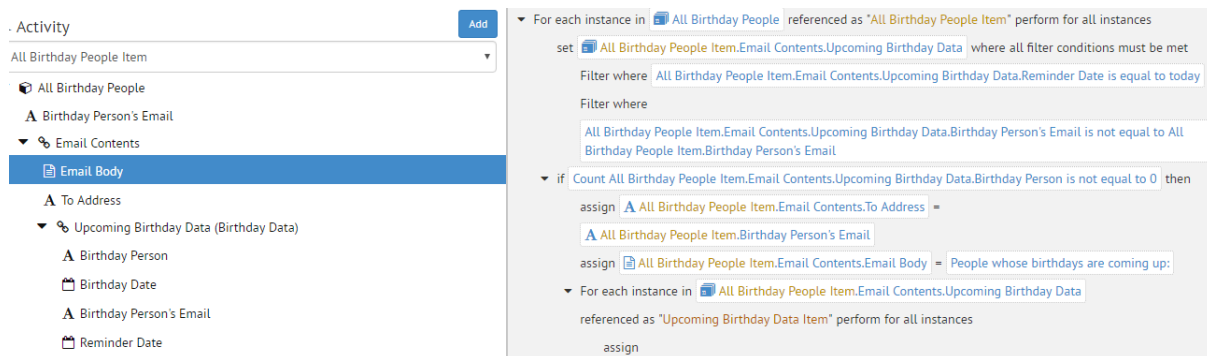
46. Click **Select a Package instance.**
47. Click **Perform an operation on one or more Package instance(s)**
48. Click **Upcoming Birthday Data (Birthday Data)** and then click **Add.**



49. Click **Perform an operation on all instances of a Package.**
50. Click **Set the value of one or more Attributes.**
51. Click **Assign a value or expression to an Attribute.**
52. In **Activity**, click on the drop-down and select the **All Birthday People Item.**



53. Expand **All Birthday People**, then expand **Email Contents**, then click **Email Body** and click **Add**.



54. Click **Expressions involving functions, formulas or more than two Attributes**.

The Complex Expression Editor displays.

55. In **Functions**, expand **Text Functions**, click **Concatenate** and click **Add**.

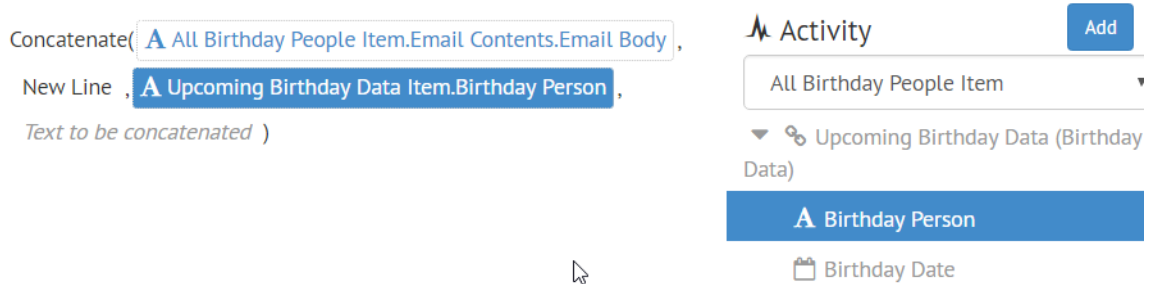
56. In **Activity**, click on the drop-down and select the **All Birthday People Item**.

57. Expand **All Birthday People** and then expand **Email Contents**. Click **Email Body** and then click **Add**.

58. In **Functions**, under **Text Functions**, click **New Line** and then click **Add**.

Concatenate(**All Birthday People Item.Email Contents.Email Body** ,
New Line , Text to be concatenated)

59. Click on **Text to be concatenated**
60. In **Activity**, check that the **Upcoming Birthday Data Item** is selected from the drop-down and expand **Upcoming Birthday Data (Birthday Data)**.
Ignore the symbol.
61. Click **Birthday Person** and then click **Add**.



62. Click **Save Expression**.
63. Click **End of Statement List** (twice).
64. Click **End of If Otherwise Statement**.
65. Click **End of Statement List** (twice).
Rule Complete displays in the left-hand panel.

Rule Complete

▼ For each instance in **All Birthday People** referenced as "All Birthday People Item" perform for all instances

set **All Birthday People Item.Email Contents.Upcoming Birthday Data** where all filter conditions must be met

Filter where **All Birthday People Item.Email Contents.Upcoming Birthday Data.Reminder Date is equal to today**

Filter where

All Birthday People Item.Email Contents.Upcoming Birthday Data.Birthday Person's Email is not equal to All Birthday People Item.Birthday Person's Email

▼ if **Count All Birthday People Item.Email Contents.Upcoming Birthday Data.Birthday Person is not equal to 0** then

assign **All Birthday People Item.Email Contents.To Address** =

All Birthday People Item.Birthday Person's Email

assign **All Birthday People Item.Email Contents.Email Body** = **People whose birthdays are coming up**

▼ For each instance in **All Birthday People Item.Email Contents.Upcoming Birthday Data** referenced as "Upcoming Birthday Data Item" perform for all instances

assign **All Birthday People Item.Email Contents.Email Body** =

Concatenate(All Birthday People Item.Email Contents.Email Body , New Line , Upcoming Birthday Data Item.Birthday Person ,)

← Reset Save Rule

66. Click **Save Rule**.

The rule is listed in the Rules section of the Properties page.

Rules

Rule	Enabled
 Generate Email Reminders	☒

Create rule

Create the Send Emails Activity

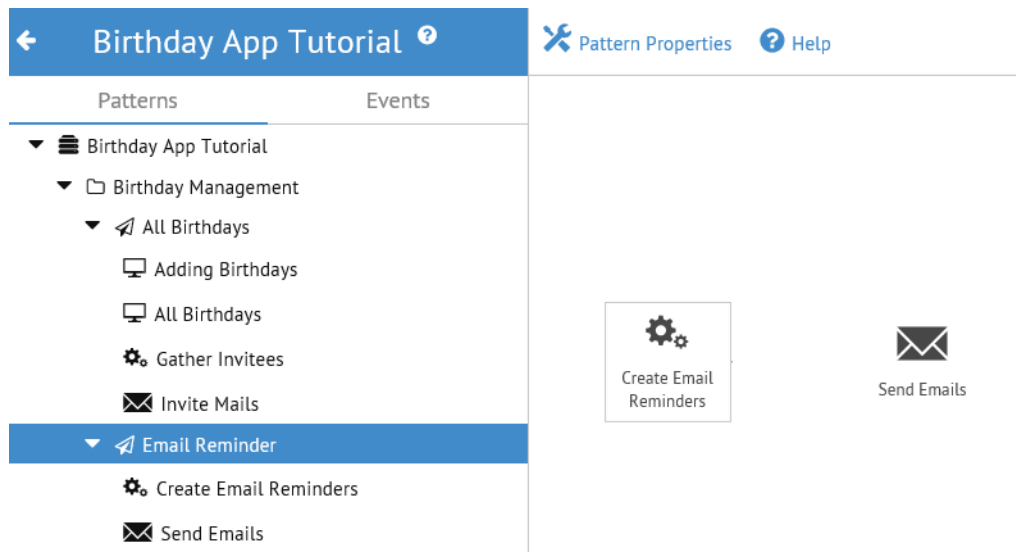
Now we will create the Send Emails Activity in the Email Reminder Pattern. The Send Emails Activity will run after the Email Reminders Activity.

- **Send Emails** is an *Email Activity* that retrieves the email content and sends out emails.

We will transfer the Email Contents Package into the Send Emails Activity and define its Package rows.

Steps

1. In the Organizational Explorer, click the **Email Reminder** Pattern flow.
2. From the upper-right corner drag and drop an **Email Activity** icon (envelope) onto the Pattern flow.
The **New Activity** form displays.
3. In **Activity name**, type **Send Emails** and click **Submit**. The Activity icon displays on the Stage.



4. Hover over **Send Emails**, click the **List** icon, and click **Definition**.

- In the left column, under **Packages**, click **Email Contents** and click the **Right-arrow** button. The **Source Package Details** form displays.

Source Package Details

Package name	Email Contents
Name in UI	Name in UI
	☐ No name in UI ☐ Suppress display of attributes in UI
Packages to include	☒ Linked packages ☒ Sub-packages
Attributes to include	☐ Only key attributes ☐ Only search attributes ☒ All attributes

←
Submit

- Click **Submit**.
- In the lower-right corner, click **Submit**.
Next we will define the data Package for this Activity.
- Below the **Activity** column, click the **Open Book** button.
- Click **Create NEW ROW(S)** and **UPDATE ROW(S)** to clear the selections.
Create NEW ROW(S) and UPDATE ROW(S) change from blue to clear.
- Leave **DELETE ROW(S)**, **SELECT ROW(S)** and **REQUIRE ROW(S)** clear.
- Ensure the **Multiple rows?** check box is clear.
There.
- Ensure the **Retrieve existing rows** check box is set.

13. Ensure the **Restrict to Row(s) most recently referenced in Pattern** check box is set.

Send Emails

Activity

- ▼ Email Contents
 - Email Body
 - ▲ To Address

Email Contents [edit name](#)

How can this package be altered? Packages related to the pagination-enabled package are non-persistent.

+ CREATE NEW ROW(S)
 UPDATE ROW(S)
 DELETE ROW(S)
 SELECT ROW(S)
 * REQUIRE ROW(S)

Multiple rows? ☐

Retrieve existing rows ☒

Restrict to ☒ Row(s) most recently referenced in pattern
☐ Row(s) created by pattern
 Row(s) that fulfill the condition:

UI details ☐ No name in UI
☐ Suppress display of attributes in UI

Submit

14. Under the Activity column, click **Submit**.

Defining the Send Email Properties

In the following steps, we will define the email properties for the Send Emails Activity.

Steps

1. Above the **Send Emails** Activity definition page, click **Activity Properties**.
2. Scroll down to **Email**.
3. In **To**, click the **Plus** button.
4. Click **Directory** and from the drop-down click **Activity**.
The **Email Contents** Package displays.
5. Expand **Email Contents**, click **To Address** and then click **Right-arrow**.

To Address transfers.

Select Email Users/Groups

Enter an e-mail address

Activity

▼ Email Contents

Email Body

To Address

>

To Address

Select Users/Groups through rule: R

← Submit

6. Click **Submit**.
7. In **From**, type **noreply@metavine.com**
8. In **Subject**, type **Upcoming Birthday Reminder**.
9. In **Body**, click the blue **Magnifying Glass** button.
The Select Attribute displays.
10. Expand the **Email Contents** Package.
11. Click the **Email Body** Attribute and click **Submit**.

Email

Enable Email Open Verification ☐

To + ×

From Q

Cc +

Bcc +

Subject Q

Body Q

Attachment Q

Enable HTML ☐

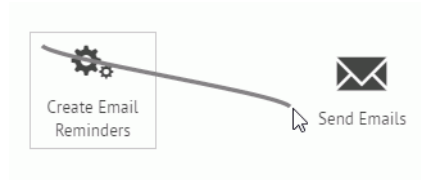
12. In the upper-right of **Email**, click **Submit** to save your changes.

Set up a Splitting Connector

Let's finish off this Pattern by setting up the Connector between the Create Email Reminders Activity and the Send Emails Activity. This Connector will be a Splitting Connector to allow multiple emails to be sent out.

1. In the **Organizational Explorer**, click the **Email Reminder Pattern**.
The Email Reminder Pattern displays on the Stage.
2. Hover over the **Create Email Reminders** Activity to display the Connector Link icon in the top left corner.

Drag and drop the **Connector Link** icon from this Activity to the **Send Emails** Activity.



3. The **New Connector** form is displayed. Leave the first three settings at their default values.
4. In **Subsequent action**, click **Create multiple instances of the next Activity**.

New Connector ?

If the activity completes ☒ Successfully
☐ With an exception

Under the following circumstances ☒ Always
☐ By default
☐ Under the following condition R

Link to activity

⚙️ Create Email Reminders
✉️ Send Emails

Subsequent action ☐ No change
☒ Create multiple instances of the next activity
 Top-level package from the current activity:

None ▾

 Matching top-level package from the next activity:

Connector Label

←
↺
Submit

5. Click the **None** drop-down.
6. Expand the **All Birthday People** data Package.

7. Click **Email Contents** and then click **Select**.

New Connector

If the activity completes: ☒ Successfully ☐ With an exception

Under the following circumstances: ☒ Always ☐ By default ☐ Under the following condition

Link to activity:

Subsequent action: ☒ No change ☒ Create multiple instances of the next activity

Top-level package from the current activity:

Matching top-level package from the next activity:

Connector Label:

8. Under **Matching top-level package from the next activity**, choose the drop-down and click **Email Contents**.

top-level package from the current activity:

Matching top-level package from the next activity:

Connector Label:

9. Click **Submit**.

The Splitting Connector is created.



This Splitting Connector enables the creation of multiple emails.

- **NOTE:** A Connector that has a three-pronged arrowhead represents a Splitting Connector.

Setting an Event in Calendar

We want this entire Pattern to run every day at 9:00 am. That means every day at 9:00 am, reminder emails will be distributed for any birthdays where the Reminder Date is equal to the current day.

We're going to begin this schedule co-ordination by first setting up an event. This event is to be created in the Calendar section of the Design Studio.

Clicking the Calendar icon in the Main Navigation Panel launches the Calendar.

In the Calendar there are five different sections. We won't describe each section in this tutorial, but you can find out more about the Calendar on the Metavine Knowledge base.

Set Up a Daily Email Reminders Event

We will access the Calendar and set up a Daily Email Reminders event.

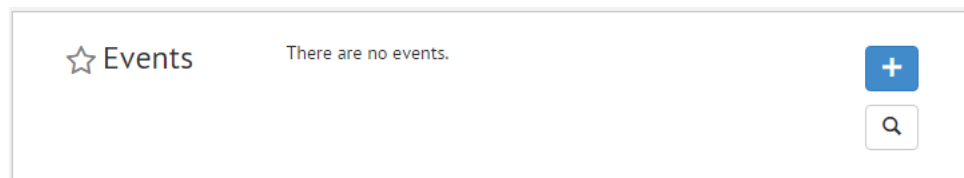
Steps

1. In the Main Navigation menu, click **Calendar**.



2. Scroll to **Events**.

The panel is empty as there are no events defined for this organization.



3. To create a new event, click the blue **Plus** button.

The New Event form displays.

4. In **Event name**, type **Daily Email Reminders**.

Calendar > Event - New

New Event

Event name

If the event falls on a non-working day

Relative to another event? ☐

5. Leave the next three default settings as they are.

6. In **Event starts on**, click the **Ongoing event** check box to set it.
- The fields options change to allow you to set an event as continually occurring and not a one-off event.

New Event

Event name

If the event falls on a non-working day

Relative to another event? ☐

How many times does the event occur? ☒ Multiple times ☐ Only once

Event starts on  OR ☒ Ongoing event

Repeats every

Repeats ☒ Indefinitely

☐ For times (excluding initial occurrence)

☐ Until specified date & time (inclusive)

7. In **Repeats every**, set a value of **1**. Set the period to **days**.
8. The **Occurs at** setting determines the timing of the event within the day. Set the time to 9:00AM using the drop-down settings and AM/PM toggle button:
- hour: 9
- minute: 0
- second: 0

AM/PM: AM

Repeats every

Occurs at

hour minute second

9. Leave the repeat schedule set to **Repeats Indefinitely**.

Repeats ☒ Indefinitely

☐ For times (excluding initial occurrence)

☐ Until specified date & time (inclusive)

10. Click **Submit**.

The event is saved.

Set a Schedule on the Email Reminder Pattern

Now we have an event upon which to base the schedule of the email reminders. Next we will set up the schedule on the Email Reminder Pattern itself.

Steps

1. In the Main Navigation menu, click **Applications**.
2. Click **Birthday App Tutorial**.
3. In the Organizational Explorer, expand the tree structure to reveal the **Email Reminder Pattern**.
4. Click **Email Reminder**.
The Pattern flow displays on the Stage.
5. In the upper-left corner of the Stage, click **Pattern Properties**.
The **Pattern - Email Reminder** displays.
6. In **Details**, under **Schedule**, click **Relative to a specific event**.
The field options change.

7. To the right of **Period type**, click in the drop-down field and click **on**.

Schedule ☐ None ☐ Specific date & time ☒ Relative to a specific event

Period type

Event

Notes

before
on
after

The Period type and number fields are hidden so that only the **on** field displays.

8. Click **Daily Email Reminders**.

Details

Name

Parent folder

Schedule ☐ None ☐ Specific date & time ☒ Relative to a specific event

on

Event

Daily Email Reminders

Notes

9. Click **Submit**.

The Pattern is now scheduled to run on the Daily Email Reminders event.

The Pattern is now complete! You now have an Automated Activity that gathers the correct Package rows for which to send out reminder emails and an Email Activity to distribute these emails. You won't have to worry about running this Pattern every day yourself as the Schedule based on the ongoing event will do this for you.

Test the Email Reminder Pattern

Do you want to see this Pattern in action straight away? Well because the Pattern runs on the schedule you have a few options for testing this. You can either wait until the next occurrence of 9:00 am, depending on what time you are completing

this tutorial, and make sure you have a birthday or two set up with reminder dates set for the current date.

Alternatively you can set up the schedule to run in the next ten minutes, do this by simply editing the **event time** in the Calendar.

Once again you'll need to have Birthday Data rows with Reminder Dates that are equal to the current date. When the Pattern runs you will receive emails for those birthdays. Once you are content that the Pattern and its included Activities are running correctly you can go back and change the event to its original settings.

Tutorial Six - Adding an Email Invite Feature

Learning Objectives



- ▶ Adding and Transferring Attributes
- ▶ Setting a Restriction Rule
- ▶ Creating a One-to-one Connector Between Activities
- ▶ Setting up an Event in Calendar

About the Email Invite Feature

In this tutorial we will add a feature to the application that provides the ability to invite others to access your Birthday Application. When these invited users access the application, they too, will have the ability to add birthdays to the application.

To make this work, we will add in an Email Activity that sends out the invites to others in the All Birthdays Pattern.

Adding Two New Attributes

Before we set up the Email Activity, we will need to add two new Attributes to the Birthday Data Package. Both are True/False Attributes. The first Attribute will determine which people you've already added to the application that you want to invite to participate in using it. The second Attribute will identify which people have been sent an invitation email.

Create Two New Attributes

In the following steps we will create and add two new Attributes.

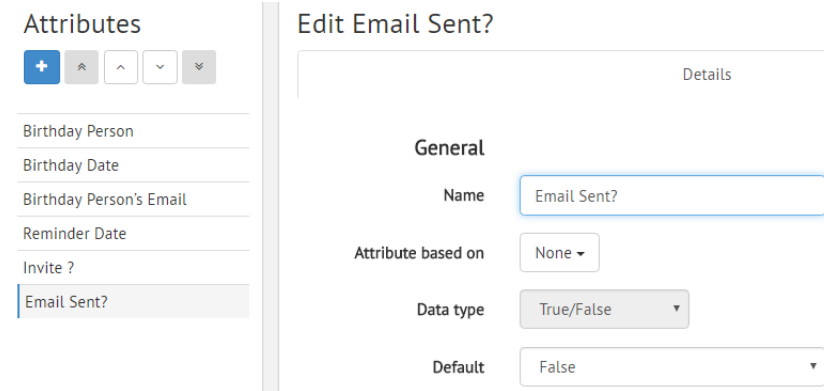
Steps

1. In the Main Navigation menu, click **Applications** and select the Birthday App Tutorial.
2. In the Organizational Explorer, click the **Packages** tab and then click the Birthday Data Package.
3. The Package details display on the Stage.
4. In **Attributes**, click the blue **Plus** button.
5. In **Name**, type **Invite?**.
6. In **Data Type**, click **True/False**.
7. Ensure **Default** is set as **False**.

The screenshot shows the 'Attributes' panel on the left and the 'Edit Invite ?' form on the right. The 'Attributes' panel lists several attributes: Birthday Person, Birthday Date, Birthday Person's Email, Reminder Date, and Invite ?. The 'Invite ?' attribute is selected. The 'Edit Invite ?' form shows the 'General' tab with the following fields: Name (Invite ?), Attribute based on (None), Data type (True/False), and Default (False). There is also an 'Encrypt data' checkbox which is unchecked.

8. Click **Submit**.
To add a second Attribute, repeat steps 2 to 6 and name the second Attribute **Email Sent?**
9. In the top-right corner of the page, click **Submit**.

The new Attributes are saved in the Package.



Attributes

+ ✖ ^ v

- Birthday Person
- Birthday Date
- Birthday Person's Email
- Reminder Date
- Invite ?
- Email Sent?

Edit Email Sent?

Details

General

Name: Email Sent?

Attribute based on: None

Data type: True/False

Default: False

Transferring New Attributes

Now we'll need to add the new Attributes into the Birthday Data Package in the Adding Birthdays Activity.

When you add a new Attribute to an existing Package that has been used in Activities, the Attribute is not automatically added to that Package in every Activity it is used in.

To add the new Attribute to an Activity, you must access the Activity definition transfer page and select the parent Package. Then you transfer the new Attribute to the correct Package in the Activity.

Transfer the New Attributes to Birthday Data in Adding Birthdays

In the following steps we will transfer the two new Attributes.

► **NOTE:** Before you select an Attribute to transfer, you must FIRST highlight the parent Package Birthday Data in Activities, as shown in step 4.

Steps

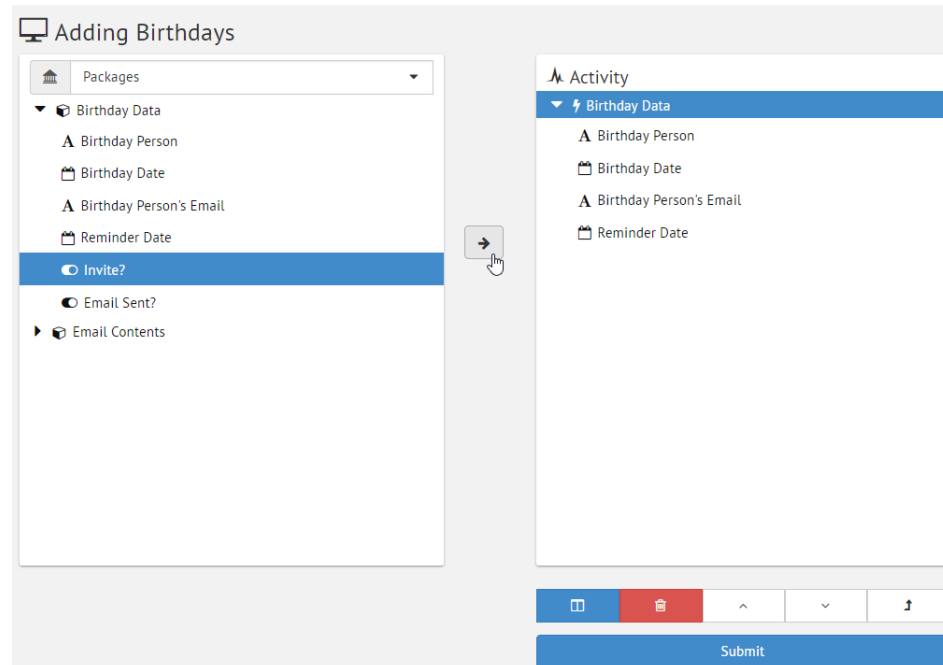
1. In the Organizational Explorer, click **Patterns**, expand **All Birthdays** and then click the **Adding Birthdays** Activity.

The Activity definition page displays on the Stage.

2. In the lower-left corner, click the blue **Plus** button.

The Activity definition transfer page displays.

3. In **Packages**, on the left, expand the tree by clicking the black caret beside the **Birthday Data** Package to expand it.
4. In **Activities**, on the right, click **Birthday Data** to highlight it.
This tells the Activity that Birthday Data is the parent Package of the Attribute being transferred.
5. In **Packages**, click **Invite?** and then click the **Right-arrow** button.



The **Invite?** Attribute transfers.

6. Repeat step 5 for **Email Sent?**.
7. Click **Submit**.

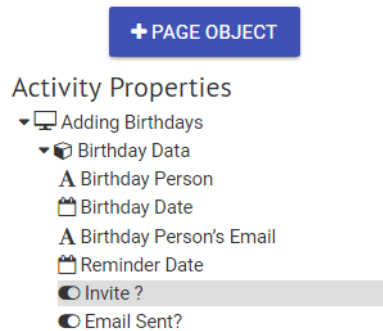
The new Attributes are saved in the Package.

Add the Invite Attribute to the Page Design

Now we will use the Page Designer to add the **Invite?** Attribute to the page design for this Activity.

Steps

1. Above the Stage, click **Page Designer**.
2. In the upper-left corner of Page Designer, expand the **Activity Properties** of **Adding Birthdays**.
3. Expand **Birthday Data** and scroll to reveal the **Invite?** Attribute.



4. Drag and drop the **Invite?** Attribute into the **Birthday Data** Package Repeater.

You can use the **Layout** tab tree structure to position the Attribute if necessary.

Birthday Management

The screenshot shows the 'Birthday Management' form. At the top right are two buttons: 'Add a Birthday' (cyan) and 'Save Birthdays' (green). Below these is a form with the following fields: 'Birthday Person' (text input), 'Birthday Date' (date picker), 'Birthday Person's Email' (text input), 'Reminder Date' (date picker), and 'Invite ?' (checkbox). At the bottom left is a red button labeled 'Delete a Birthday'.

5. Click **Save**, then click **Close**.

Creating the Gather Invitees Automated Activity

We need to create and add an Automated Activity called Gather Invitees. This Activity will retrieve all the Birthday Data Package rows that have Attributes set as follows:

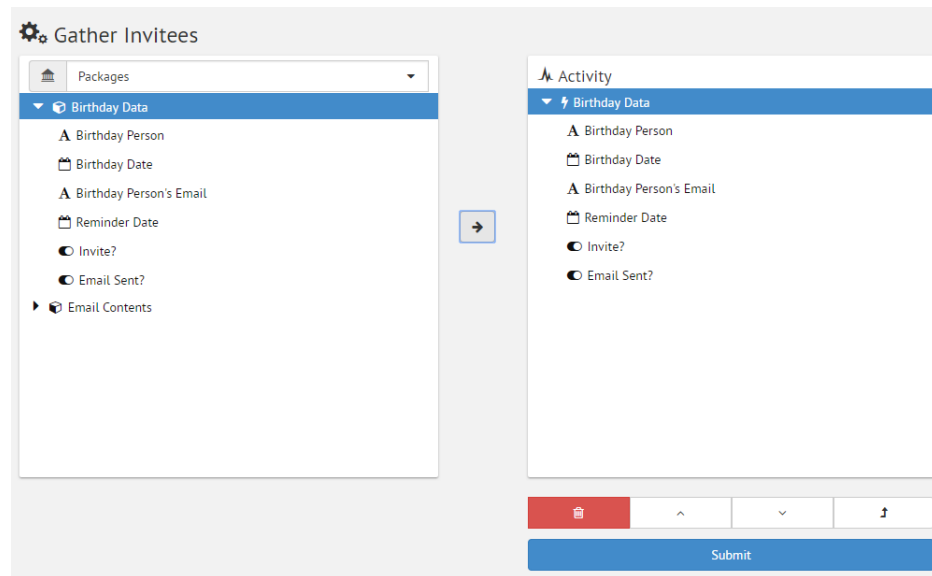
- **Invite?** set to **true**
and
- **Email Sent?** set to **false**.

Create the Gather Invitees Activity

In the following steps we will create a new Automated Activity to gather invitees.

Steps

1. In the Organizational Explorer, click the **All Birthdays** Pattern flow.
2. From the upper-right corner drag and drop an **Automated Activity** icon (gears) onto the Pattern flow.
The **New Activity** form displays.
3. In **Activity name**, type **Gather Invitees** and click **Submit**.
The Activity icon displays on the stage.
Next, we need to transfer the Birthday Data Package into the new Activity.
4. Hover over **Gather Invitees**, click the **List** icon, and click **Definition**.
5. In **Packages**, click **Birthday Data** and click the **Right-arrow** button.
The **Source Details** form displays.

6. Click **Submit**.7. In the lower-right corner, click **Submit**.

Setting a Restriction Rule

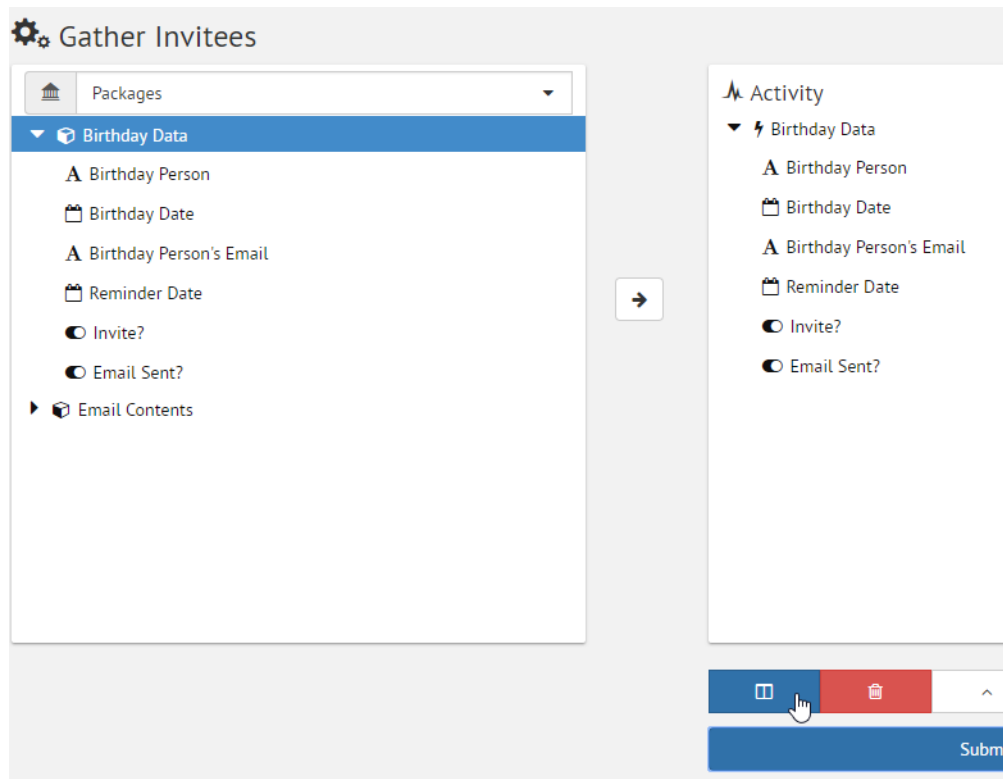
We need to set up a restriction rule on the Birthday Data Package in the Gather Invitees Activity. This rule will ensure that only the Package rows with the **Invite?** Attribute set to true and the **Email Sent?** Attribute set to false are retrieved.

Set Up a Restriction Rule in the Birthday Data Package

Complete the following steps to define the retrieval and Package definition settings. In step 9, we will use the Complex Expression Editor to build a rule to retrieve the rows where **Invite?** Attribute is equal to **True** and **Email Sent?** Attribute is equal to **False**.

Steps

1. From the **Gather Invitees** Activity transfer page, click the blue **Open Book** button.



2. Click **Create NEW ROW(S)** to clear the selection.
Create NEW ROW(S) changes from blue to clear.
3. Leave **UPDATE ROW(S)** selected.
UPDATE ROW(S) is blue.
4. Leave **DELETE ROW(S)**, **SELECT ROW(S)** and **REQUIRE ROW(S)** clear.
5. Click the **Multiple rows?** check box so that it is set.
6. Ensure the **Retrieve existing rows** check box is set.
7. Click **Rows most recently referenced in the Pattern** to clear the selection.

8. Click the orange **R**.

Birthday Data [edit name](#)

How can this package be altered? Packages related to the pagination-enabled package are non-persistent.

+ CREATE NEW ROW(S)	U UPDATE ROW(S)	D DELETE ROW(S)	Q SELECT ROW(S)	* REQUIRE ROW(S)
-------------------------------	---------------------------	---------------------------	---------------------------	----------------------------

Multiple rows? ☒

Minimum rows required

Maximum rows required

Retrieve existing rows ☒

Restrict to ☐ Row(s) most recently referenced in pattern
☐ Row(s) created by pattern
 Row(s) that fulfill the condition

The Complex Expression Editor displays.

9. In the left panel, in **Operators**, expand the **Logical** folder.
10. Click **and**, then click **Add. boolean operand 1** is highlighted.
11. In the **Comparison** folder, click **is equal** and then click **Add**. In the expression, **operand 1** is highlighted.
12. Click **boolean operand 2** and in the **Logical** folder select **or** and then click **Add**. In the expression, **boolean operand 1** is highlighted.

```
( ( operand 1 is equal to operand 2 ) and ( boolean operand 1 or boolean operand 2 ) )
```

13. In the **Comparison** folder, select **is equal** and then click **Add**.
14. Click **boolean operand 2** and under **Operators** expand **Attribute/Package** and select **does not have a value** and click **Add**.

```
( ( operand 1 is equal to operand 2 ) and  
( ( operand 1 is equal to operand 2 ) or ( attribute operand does not have a value ) )  
)
```

15. Click **operand 1** in the first statement of the expression.

operand 1 is highlighted in blue.

16. In **Activity**, click **Invite?** and then click **Add**.

Birthday Data.Invite is inserted into the expression and operand 2 is highlighted.

```
( ( Birthday Data.Invite ? is equal to operand 2 ) and
  ( ( operand 1 is equal to operand 2 ) or ( attribute operand does not have a value ) )
)
```

17. In **Add Constant**, click **True** and then click **Add**.

18. In the second statement of the expression, click **operand 1**.

19. In **Activity**, click **Email Sent?** and then click **Add**. **operand 2** is highlighted.

```
( ( Birthday Data.Invite ? is equal to true ) and
  ( ( Birthday Data.Email Sent? is equal to operand 2 ) or ( attribute operand does not have a value ) ) )
```

20. In **Add Constant**, click **False** and then click **Add**.

```
( ( Birthday Data.Invite ? is equal to true ) and
  ( ( Birthday Data.Email Sent? is equal to false ) or ( attribute operand does not have a value ) ) )
```

21. In the third statement of the expression, click on **attribute operand**. In Activity, click **Email Sent?** and then click **Add**.

22. The rule should read **((Birthday Data.Invite? is equal to true) and (Birthday Data.Email Sent? is equal to false) or (Birthday Data.Email Sent? does not have a value))**.

23. Click **Save Expression**.

The rule appears under retrieval settings in the Package **Details**.

This rule states that the Gather Invitees Activity will only retrieve the rows where those Attributes, **Invite?** and **Email Sent?** are set to true and false, respectively.

24. Click **Submit**.

► **NOTE:** A restrictive rule on a Package does not require a **Rule Title**.

Setting the Invite? Attribute Manually

Earlier in the tutorial, we added a check box to set the **Invite?** Attribute in the Adding Birthdays Activity. This check box is selected manually by you to

determine each person that you want to invite to participate in application usage. This is fairly straightforward as it a manually setting. However, the **Email Sent?** Attribute is not that straightforward, and therefore, we will need to create a rule.

Creating a Rule to Set the Email Sent Attribute

We need to have the **Email Sent?** Attribute set to true to identify which people have been sent an email. Therefore, we will need to create rule in the Automated Activity. This rule determines that any rows that have been retrieved in this Activity, will have their **Email Sent?** Attribute set to true.

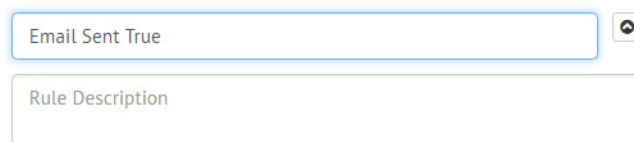
When a new birthday is added, and that person is invited to the application, all rows where the **Invite?** Attribute is set to true (email sent) will not be retrieved again. This combination of having both the **Invite?** Attribute and **Email sent?** Attribute set to true, means those rows will fail the retrieval condition set on Birthday Data Packages in this Activity and will not be retrieved.

Create a Rule for Email Sent

In the following steps, we will create a rule to set the **Email Sent?** Attribute to true.

Steps

1. In the Organizational Explorer, click **Gather Invitees**.
2. From above the stage, click **Activity Properties**.
3. Scroll to **Rules** and click **Create rule**.
4. In the **Rule Title** field, type **Email Sent True**.



The screenshot shows a form with two fields. The first field, labeled 'Rule Title', contains the text 'Email Sent True'. To the right of this field is a small circular icon with a question mark. Below the first field is a second field labeled 'Rule Description', which is currently empty.

5. Click **Select a Package instance**.
6. Click **Perform an operation on one or more Package instances**.

7. From **Activity Packages**, click **Birthday Data** and then click **Add**.

The rule should now read '**For each instance in Birthday Data perform...**'

▼ For each instance in **Birthday Data** referenced as "Birthday Data Item" perform

8. Click **Perform an operation on all instances of a Package**.

9. Click **Set the value of one or more Attributes**.

10. Click **Assign a value or expression to an Attribute**.

11. From **Activity**, click **Email Sent?** and then click **Add**.

▼ For each instance in **Birthday Data** referenced as "Birthday Data Item" perform for all instances
assign **Birthday Data Item.Email Sent?** =

12. Click **Simple value such as a number or string value etc**.

13. Click **true**.

14. Click **End of statement list** (twice).

▼ For each instance in **Birthday Data** referenced as "Birthday Data Item" perform for all instances
assign **Birthday Data Item.Email Sent?** = true

Rule Complete displays.

15. Click **Save Rule**.

The rule is listed in the Rules section of the Properties page.

16. Check that the **Email Sent True** rule is **Enabled**.

Your Automated Activity is now set up to:

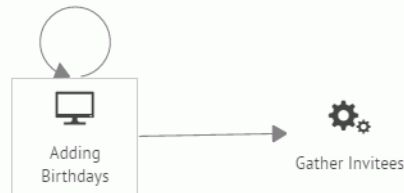
- Retrieve and sort the correct data for the next Email Activity,
and
- Set the **Email Sent?** Attribute to true to ensure these rows that are emailed
will not be sent an invite email again.

Create a Connector Between Activities

In the following steps we will create a Connector between the Adding Birthdays and Gather Invitees Activities.

Steps

1. In the Organizational Explorer, click **All Birthdays**.
The All Birthdays Pattern displays on the stage.
2. Hover over the **Adding Birthdays** Activity, click the **Link Connector** icon, then drag and drop a Connector to **Gather Invitees**.
The **New Connector** form displays.
3. Leave the **Under the following circumstances** option set to **Always**.
4. Click **Submit**.



Create the Invite Emails Activity

Now we will create the Invite Emails Activity in the All Birthday's Pattern. The Invite Emails Activity will run after the Adding Birthdays Activity is submitted and has gathered the correct Birthday Data Package rows to send emails for.

We will transfer the Birthday Data Package and Email Contents Package into this Activity and create a relationship between them.

Steps

1. In the Organizational Explorer, click the **All Birthdays** Pattern flow.
2. From the upper-right corner drag and drop an **Email Activity** icon (envelope) onto the Pattern flow.
The **New Activity** form displays.

3. In **Activity name**, type **Invite Emails**.

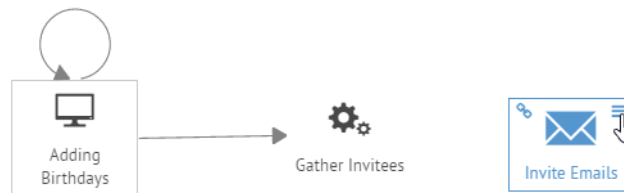
✉ New Activity - Email

Activity name

Activity type

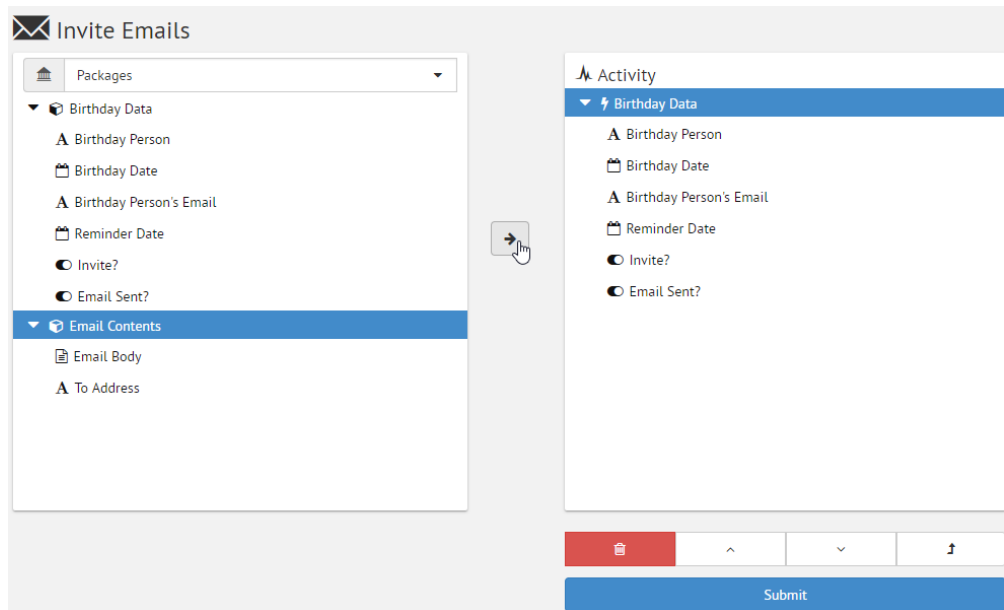
Description

4. Click **Submit**.
The Activity icon displays on the Stage.
5. Hover over **Invite Emails**, click the **List** icon.



6. Click **Definition**.
7. In **Packages**, click **Birthday Data** and click the **Right-arrow** button. The **Source Details** form displays.
8. Click **Submit**.
9. In **Activities** in the right panel, **Birthday Data** is highlighted.
10. Under **Packages** in the left panel, click **Email Contents**.

11. Click the **Right-arrow**.



The **Source Package Details** form displays.

12. Click the **Create new relationship** check box to set it.
13. In **New relationship name**, type **Invite Contents**.
14. Leave the relationship set as **One to one**.

Source Package Details

Package name	Email Contents
Name in UI	Name in UI
	☐ No name in UI ☐ Suppress display of attributes in UI
Packages to include	☒ Linked packages ☒ Sub-packages
Attributes to include	☐ Only key attributes ☐ Only search attributes ☒ All attributes
Transfer at top-level	☐
Create new relationship	☒
New relationship name	Invite Contents
Cardinality	☒ One to one ☐ One to many

15. Click **Submit**.

16. In the lower-right corner, click **Submit**.

Next we will define both data Packages for this Activity.

Define the Data Packages for Invite Emails

In the following steps we will use the Activity Definition main page to define the data Packages for the Invite Emails Activity.

Steps

1. From the Invite Emails transfer page, click the blue **Open Book** button.
The Birthday Data Package details display on the stage.

2. Click **Create NEW ROW(S)** and **UPDATE ROW(S)** to clear these selections.

Create NEW ROW(S) and UPDATE ROW(S) change from blue to clear.

(We will not be creating or editing any of the Birthday Data Package rows in this Activity.)

3. Click the **Multiple rows?** check box so that it is set.
4. Ensure the **Retrieve existing rows** and **Row(s) most recently referenced in the Pattern** check boxes are set.

We want to retrieve the rows most recently referenced in the Pattern. These are the rows from the Automated Activity which have been sorted as the correct rows for which to distribute emails.

Invite Emails

Activity

- Birthday Data**
 - Birthday Person
 - Birthday Date
 - Birthday Person's Email
 - Reminder Date
 - Invite ?
 - Email Sent?
 - Email Contents

Birthday Data [edit name](#)

How can this package be altered? Packages related to the pagination-enabled package are non-persistent.

Multiple rows? ☒
 Minimum rows required
 Maximum rows required

Retrieve existing rows ☒

Restrict to ☒ Row(s) most recently referenced in pattern
☐ Row(s) created by pattern

Submit

5. In **Activity**, click **Email Contents**.
The Email Contents Package details display on the stage.
6. No settings should be selected, except **Retrieve** should be set to **None**.

Invite Emails

Activity

- Birthday Data
 - Birthday Person
 - Birthday Date
 - Birthday Person's Email
 - Reminder Date
 - Invite?
 - Email Sent?
 - Email Contents**
 - Email Body
 - To Address

Email Contents [edit name](#)

How can this package be altered? Packages related to the pagination-enabled package are non-persistent.

+ CREATE NEW ROW(S)
 ✎ UPDATE ROW(S)
 🗑 DELETE ROW(S)
 🔍 SELECT ROW(S)
 * REQUIRE ROW(S)

Multiple rows? ☐

Relationship: Invite Contents (Linked)

Allow existing related rows to be removed ☐

Retrieve: ☐ All existing rows ☐ Related rows ☒ None

Only submit: ☐ Modified by user

Row(s) that fulfill the condition:

Submit

7. Click **Submit**.

Now that the definition is correct, we're going to work on the email properties.

Create a Rule for Email Contents Body

You'll need to create a rule for the email contents body. Our message is going to tell the invitee that they have been added to your Birthday App and then grant them access. We'll need to send them the Pattern URL to give them this access.

Steps

1. In the Organizational Explorer, click **All Birthdays**.
2. In the upper-left of the Stage, click **Pattern Properties**.
3. Right-click and copy the URL.
Got your URL copied? Okay, let's make the **Email body** rule.

Create the Email Body Rule.

Steps

1. Go to the Invite Emails **Activity Properties** page, click **Create rule**.
The Rule Builder displays.
2. In the **Rule Title**, type **Email Body**.
3. Click **Set the value of one or more Attributes**.
4. Click **Assign a value or expression to an Attribute**.
5. Click the **Email Body** Attribute from the Activity content list and then click **Add**.
6. Click **Expressions involving functions, formulas or more than two Attributes**.
We're going to create an email body message again here just like we did in the previous Email Activity.
7. In **Operators**, expand the **Text** folder, click **joined with** and **Add**.
The expression contains **text operand 1 joined with text operand 2**.
8. In the expression, click **text operand 2**.
9. From the **Text** folder, click **joined with** and **Add**.

(text operand 1 joined with (text operand 1 joined with text operand 2))

10. Repeat steps 8 and 9 again, so that your expression reads **(text operand 1 joined with (text operand 1 joined with (text operand 1 joined with text operand 2)))**.
11. To enter in a message to the invitee, click the *first* **text operand 1**.
12. In **Add Constant**, type **You've been added to [insert your name]'s Birthday App!** and click **Add**.

(You've been added to Peter's Birthday App! joined with
(text operand 1 joined with (text operand 1 joined with text operand 2)))

13. Click the *second* **operand 1**.
14. Under **Functions**, expand **Text Functions** and click **New Line** and then click **Add**.
15. Click the *third* **text operand 1**.

16. In **Add Constant**, type the text **Go to** (insert Pattern URL link) **to check it out** and click **Add**.

```
( You've been added to Peter's Birthday App! joined with
  ( New Line joined with
    (
      Go to https://genesis.metavine.com/development/runtime/runprocess?
      process=3B7494A2B3B311E7894002E6476F0C2B
      joined with text operand 2 )
    )
  )
)
```

17. In the *final* **text operand 2**, in Add Constants, type **Happy Exploring!** and click **Add**.

18. Click **Save Expression**.

assign  Email Contents.Email Body =

```
You've been added to Peter's Birthday App! joined with New Line joined with Go to
https://genesis.metavine.com/development/runtime/runprocess?process=3B7494A2B3B311E7894002E6476F0C2B joined with Happy
Exploring!
```

19. Click **End of Statement List**.
20. Click **Save Rule**.

Defining the Email Properties

Great! Now that the rule is done, we can go ahead and finish off setting up this Email Activity by defining the Email Properties. You will be sending this email to the birthday people who have been added in the Birthday Data Packages. We'll set up the email distribution to be Bcc, so that you won't be distributing other people's emails to someone they might not know.

In the following steps, we will define the email properties for the Invite Emails Activity.

Steps

1. You should still be in the Invite Emails Activity Properties page. If not, click the **Invite Emails** Activity and then click **Activity Properties**.
2. Scroll down to **Email**.

- In **From**, type **noreply@metavine.com**.

Email

To		+
From	noreply@metavine.com	Q
Cc		+
Bcc		+

- In **Bcc**, click the blue **Plus** button.
The **Select Email Users/Groups** form displays.
- In **Directory**, click the **arrow** and then click **Activity**.
The Activity content displays the Birthday Data Package.
- Expand **Birthday Data** and click **Birthday Person's Email**.
- Click **Right-arrow**.
The Attribute now transfers across.

Select Email Users/Groups

 Activity
▼

▼
Birthday Data

▲ Birthday Person
📅 Birthday Date
▲ Birthday Person's Email
📅 Reminder Date
🕒 Invite ?
🕒 Email Sent?
▶ 🔗 Email Contents

Birthday Person's Email

Select Users/Groups through rule: R

- Click **Submit**.
- In **Subject**, type **Check out my Birthday App!**

10. In **Body**, click the blue **Magnifying Glass** button.
The Select Attribute displays.
11. Expand both the **Birthday Data** and **Email Contents** Packages.
12. Click the **Email Body** Attribute and click **Submit**.
Select attribute

Email Body

- Activity
 - Birthday Data
 - Birthday Person
 - Birthday Date
 - Birthday Person's Email
 - Reminder Date
 - Invite ?
 - Email Sent?
 - Email Contents
 - Email Body

← Submit

13. In the upper-right of **Email**, click **Submit** to save your changes.

Email

To		+
From	noreply@metavine.com	Q
Cc		+
Bcc	Birthday Person's Email	+ x
Subject	Check out my Birthday App!	Q
Hide message header	☒	
Body	Email Body	Q
Attachment		Q
Enable HTML	☐	

Your Invite Emails Activity is now configured and ready to distribute emails to the people you choose to invite. Now, when you add a Birthday and provide the birthday person's email address you will be able to choose to invite them to collaborate on your application.

Set up a Splitting Connector

Let's finish off this Pattern by setting up the Connector between the Gather Invitees Activity and the Invite Emails Activity. This Connector will be a Splitting Connector to allow multiple emails to be sent out.

1. In the Organizational Explorer, select the All Birthdays Pattern flow and hover over the **Gather Invitees** Activity.
Drag and drop the **Connector Link** icon from this Activity to the **Invite Emails** Activity.
2. The **New Connector** form is displayed. Leave the first three settings at their default values.
3. In **Subsequent action**, click **Create multiple instances of the next Activity**.

 New Connector 

If the activity completes ☒ Successfully
☐ With an exception

Under the following circumstances ☒ Always
☐ By default
☐ Under the following condition 

Link to activity

 Adding Birthdays

 Gather Invitees

 Invite Emails

Subsequent action ☐ No change
☒ Create multiple instances of the next activity

Top-level package from the current activity:

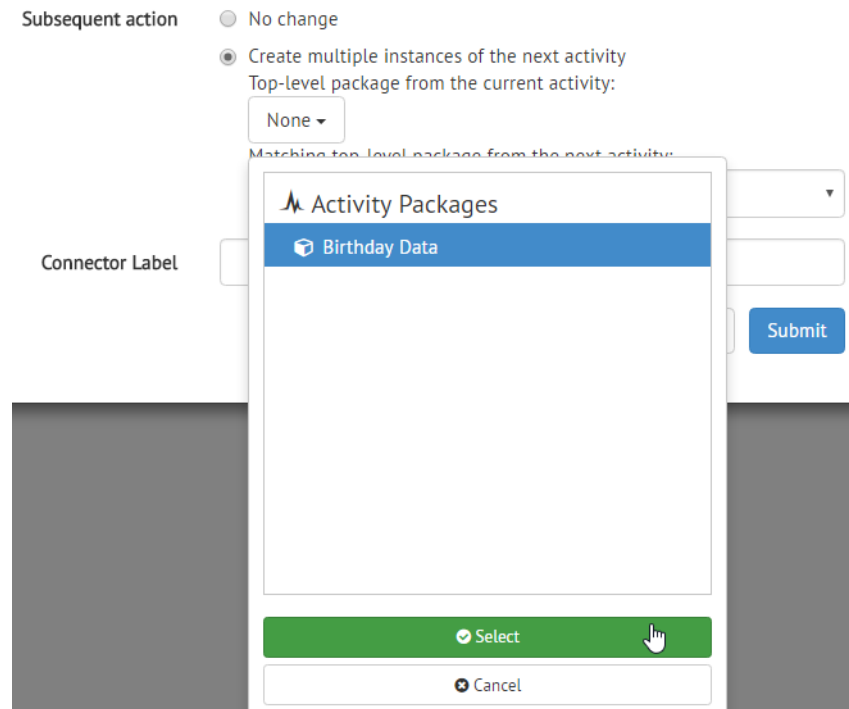
None ▾

Matching top-level package from the next activity:

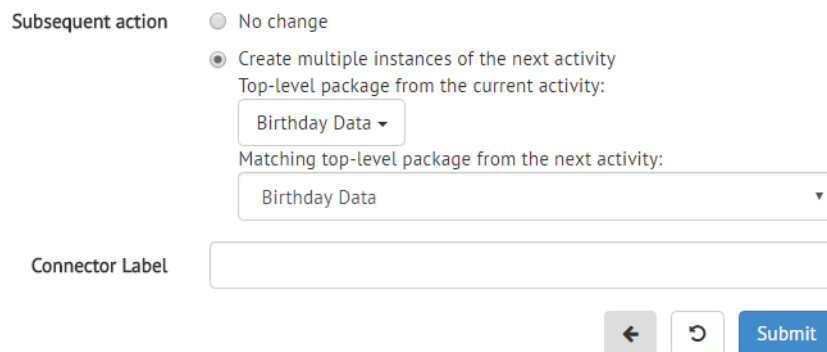
Connector Label

4. Click the **None** drop-down.
5. Click the **Birthday Data Package** and then click **Select**.

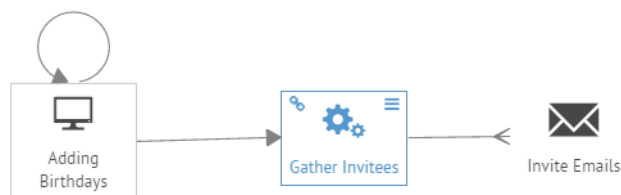


6. Under **Matching top-level package from the next activity** click the drop-down and then click **Birthday Data**.



7. Click **Submit**.

The Splitting Connector is created.



Now when you submit the Adding Birthdays Activity and you've chosen to invite a person to look at your application, the Automated Activity and Email Activity will run to send out the emails to the correct people.

Test the Email Invite Feature

Now you can test the Email Invite feature.

1. In the Organizational Explorer, select the **All Birthdays** Pattern.
2. Click **Pattern Properties**.
3. In Details, click on the Pattern URL.
The Birthday Management screen displays.
4. In **Birthday Person**, add your name.
5. In **Birthday Date**, add your birthday.
6. In **Birthday Person's Email**, add your email address.
7. Click the **Invite?** check box so that it is set.
8. Click **Save Birthdays** to submit this Activity.
9. Check your email address to see the Invite email that has been distributed to you.

You'll be able to access your application via the link in the email and should see the email message composed in the Rule Builder as the content of the email.

Now, when you add your friends and family to your Birthday App and choose to invite them, they will receive this email invite to collaborate and check out the application you have built.

Congratulations! You've just built your own application.

With the Metavine Design Studio you have created a Birthday Management application all on your own, without writing code. There is endless potential to expand on the functionality and design of this application.

