BLASTING OFF with WORKFLOWS

Welcome

Workflows are all around Rock. Do you want to know what workflows do? They're used for check-in, requests, even to authorize changes to data. You have a choice: embrace workflows or deny the truth. The truth that without them you are a slave to repetition. Stuck in a virtual prison of repetitive time-wasting activity.

.... dramatic pause... sigh...

Unfortunately, you can't just hear about what workflows can do, you must see them for yourself.

This is your last chance. After this there's no turning back. You can take the blue pill—the story ends and it's back to a life of manually clicking through screen after screen. Or you can take the red pill—you'll enter a wonderland and discover the power that automation can bring to your life.

The choice is yours. You must decide.

Confused about all this pill talk? This might help.

What's The Use

Workflows. That word can be confusing. So let's simplify it. Workflows are a series of steps that can be automated. We all know computers are better at repetitive tasks than humans. Rock workflows provides a framework for getting computers to do what they're good at so we can focus on what we humans do best - relationships.

So what can Rock do? We’re glad you asked!

- **Request Systems:** One common use for Rock workflows is to create request systems that can take information from your users and provide automated flow based on their input. An HR Position Request or IT Request are good examples of these functions.
- **Data Changes:** Workflows can be launched in response to data changes in Rock. For instance you could configure a workflow to be launched whenever a group is added to the system. This workflow could email an administrator, or even prevent that creation if certain information about the group is not provided.
- **Background Tasks:** By using a Rock Job, you can enable a workflow to run on a specified schedule.

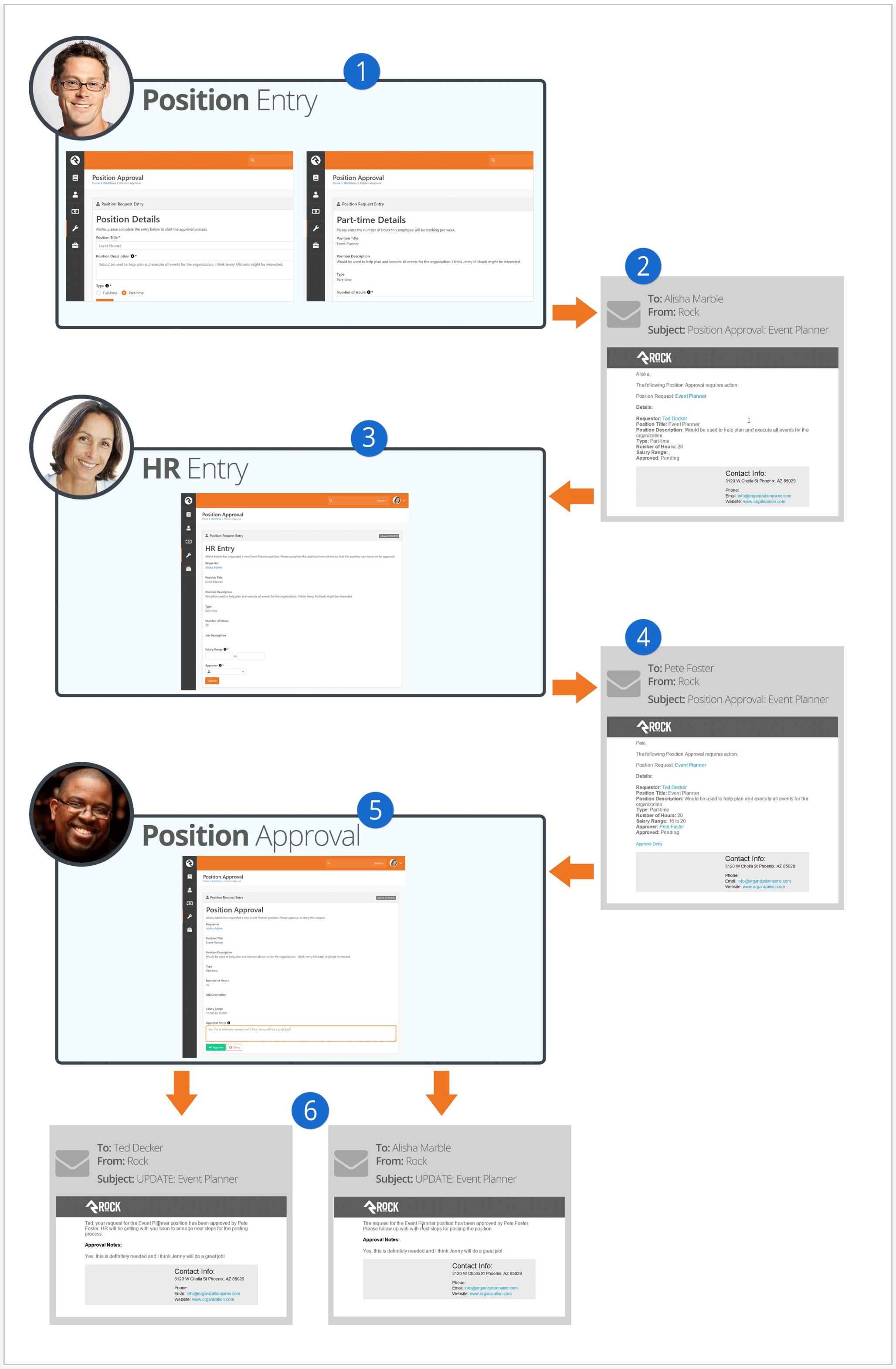
When you're done with this manual, we think you'll see how workflows empower you to create powerful application logic without needing to become a programmer. Once you understand the basics, your mind will start racing with all of the ways you can put them to use.

These are just the tip of the iceberg of how workflows can be used within an organization. Our fear is the list above will pigeonhole your thinking of when and how to use workflows. When you're looking to solve an organizational need, be sure to think out-of-the-box when it comes to using workflows.

A Sample Workflow

Let's take a look at a sample workflow to get an idea of what's possible. In our sample, the fictitious "Rock Solid Church" has implemented a human resources process to help manage new position approvals. With this process in place let's say that Ted Decker wants to get approval to hire a part-time event planner. Let's walk through the workflow that has been defined.

Sample Workflow



To: Alisha Marble
From: Rock
Subject: Position Approval: Event Planner

Alisha,

The following Position Approval requires action:

Position Request: Event Planner

Details:

Requestor: Ted Decker
Position Title: Event Planner
Position Description: Would be used to help plan and execute all events for the organization.
Type: Part-time
Number of Hours: 20
Salary Range: ,
Approved: Pending

Contact Info:
3120 W Cholla St Phoenix, AZ 85029
Phone:
Email: info@organizationname.com
Website: www.organization.com

To: Pete Foster
From: Rock
Subject: Position Approval: Event Planner

Pete,

The following Position Approval requires action:

Position Request: Event Planner

Details:

Requestor: Ted Decker
Position Title: Event Planner
Position Description: Would be used to help plan and execute all events for the organization.
Type: Part-time
Number of Hours: 20
Salary Range: 16 to 20
Approver: Pete Foster
Approved: Pending

Approve Deny

Contact Info:
3120 W Cholla St Phoenix, AZ 85029
Phone:
Email: info@organizationname.com
Website: www.organization.com

To: Ted Decker
From: Rock
Subject: UPDATE: Event Planner

Ted, your request for the Event Planner position has been approved by Pete Foster. HR will be getting with you soon to arrange next steps for the posting process.

Approval Notes:

Yes, this is definitely needed and I think Jenny will do a great job!

Contact Info:
3120 W Cholla St Phoenix, AZ 85029
Phone:
Email: info@organizationname.com
Website: www.organization.com

To: Alisha Marble
From: Rock
Subject: UPDATE: Event Planner

The request for the Event Planner position has been approved by Pete Foster. Please follow up with with next steps for posting the position.

Approval Notes:

Yes, this is definitely needed and I think Jenny will do a great job!

Contact Info:
3120 W Cholla St Phoenix, AZ 85029
Phone:
Email: info@organizationname.com
Website: www.organization.com

1 **Request**

Ted starts the process by entering his request into the system. Many of these requests would be started by going to `Tools > Workflows` , but you can add new workflow pages anywhere you'd like in the navigation. On the first entry screen, Ted selected the option of needing a *Part-Time* position. The workflow took that into consideration and immediately asked him to complete an additional entry screen specifically for part-time submissions that asks for the number of hours needed for the position.

2 **Sent**
After all of the entry screens have been completed, the workflow sends an email to the designated human resources worker, in this case Alisha Marble.

3 **Additional Info**
After clicking a link from the email, Alisha completes an entry form to add additional human resources fields like salary range and who will need to approve the position. Keep in mind this is a simple example. You could automate the selection of the approver if you'd like. In our example Alisha has selected the church's senior pastor Pete Foster.

4 **Approve/Deny**
The workflow now sends Pete an email regarding the position. From the email, Pete can click to approve or deny the position.

5 **Notes**
In our example, Pete has chosen to click the link to approve the request using the Rock website. This allows him to add further notes.

6 **Emails**
With the request approved, emails are sent to both Ted to let him know the good news and Alisha so that she can start the needed human resources paperwork.

This is just a quick example of one workflow. We'll look behind the scenes of this specific workflow later in the chapter Building a Simple Workflow.

Components Of A Workflow

Think of workflows like a big box of Legos. Each piece has a specific shape that determines how it can be used. To become a *Lego Master Builder* you must understand the possibilities that lie inside each type of brick. You also need to know how pieces work together. Before we get started with building an actual workflow let's find out a bit more about the blocks we'll be building with.

Point of Reference:

Throughout this document we'll be using the pre-configured *External Inquiry* workflow as a point of reference. This workflow is used as a *Contact Us* type tool where a guest on your website can enter a question and it gets routed to the appropriate group.

Workflow Types vs. Workflows

Let's start with a little vocabulary. *Workflow Types* are the configuration patterns that a specific workflow will use to execute. As an example, you might configure an *HR Position Approval* workflow type that an employee uses to initiate an *IT Director Position Request* workflow.

Attributes

Attributes are the data elements your workflow needs to be able to process. For our *Inquiry* example, we'll need information about the requester (name, email address, phone) as well as the topic, message and campus. Once we have the input from the guest, we'll also need attributes that store the person being assigned to the inquiry as well as any notes that they enter.

Attributes can represent many types of data including: text, numbers, images, locations, a person, date and more.

Activities

Activities are groupings of actions that function together to complete a unit of work. How many activities you use in a particular workflow is part science and part art, but in most cases there is not a right answer. In general though, think of activities as phases of your workflow. In our inquiry example there are activities for the initial entry of the request that process steps for each category of the inquiry (pastoral inquiry, website inquiry, finance inquiry, etc.)

When you configure your workflow, you'll set certain activities to *Activate* on the start of the workflow. These activities can then activate other activities depending on the nature of the input and workflow logic.

While most attributes will be defined for the entire workflow, it is also possible to define attributes that are specific to an activity. This allows for activities to have their own set of data items.

Actions

Activities are made up of *Actions*. Actions are the smallest unit of work in a workflow. Don't let their size fool you though. Like ants, they may be small but they can move large objects by working together. Some examples of what actions can do are:

- Send an email.
- Set the value of an attribute.
- Present the user with a form to enter data.
- Run a SQL query.
- Activate a new activity.

Status

Every instance of a workflow has a *Status*. There's nothing magical about the status. In fact, it's just a text field that's updated by the actions as they process through the workflow. A well-crafted workflow uses the status field to help communicate the stage the workflow is in. For instance, a work request workflow might use statuses like *Pending*, *Open,* or *Closed*.

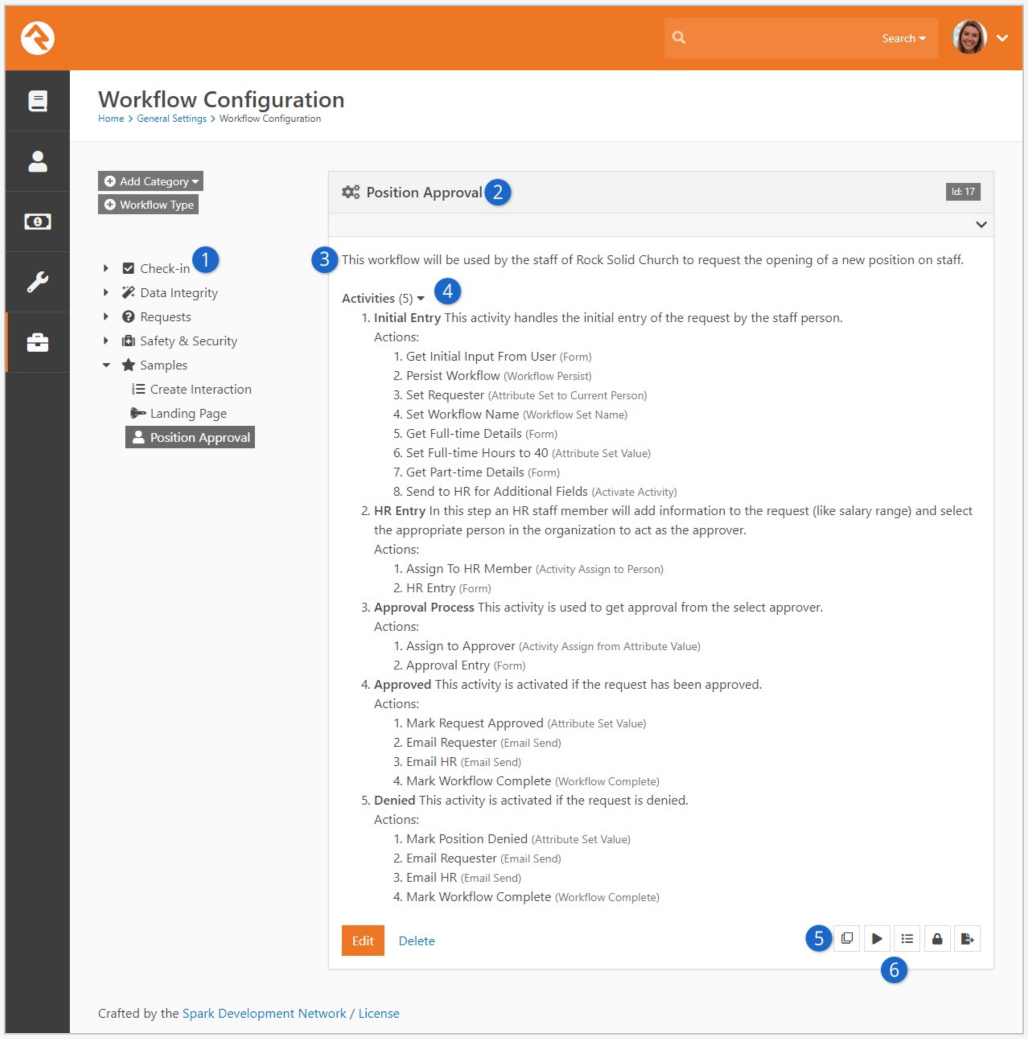
Configuring A Workflow Type

Let's take a tour of the workflow configurator located under `Admin Tools > General Settings > Workflow Configuration`. We'll break the screen down into parts to help simplify our discussion.

Viewing A Workflow Type

On the *Workflow Configuration* detail screen you can view important information about your workflow.

Workflow Configuration



Workflow Configuration
Home > General Settings > Workflow Configuration

Position Approval

This workflow will be used by the staff of Rock Solid Church to request the opening of a new position on staff.

Activities (5)

1. **Initial Entry** This activity handles the initial entry of the request by the staff person.
   Actions:
   1. Get Initial Input From User (Form)
   2. Persist Workflow (Workflow Persist)
   3. Set Requester (Attribute Set to Current Person)
   4. Set Workflow Name (Workflow Set Name)
   5. Get Full-time Details (Form)
   6. Set Full-time Hours to 40 (Attribute Set Value)
   7. Get Part-time Details (Form)
   8. Send to HR for Additional Fields (Activate Activity)
2. **HR Entry** In this step an HR staff member will add information to the request (like salary range) and select the appropriate person in the organization to act as the approver.
   Actions:
   1. Assign To HR Member (Activity Assign to Person)
   2. HR Entry (Form)
3. **Approval Process** This activity is used to get approval from the select approver.
   Actions:
   1. Assign to Approver (Activity Assign from Attribute Value)
   2. Approval Entry (Form)
4. **Approved** This activity is activated if the request has been approved.
   Actions:
   1. Mark Request Approved (Attribute Set Value)
   2. Email Requester (Email Send)
   3. Email HR (Email Send)
   4. Mark Workflow Complete (Workflow Complete)
5. **Denied** This activity is activated if the request is denied.
   Actions:
   1. Mark Position Denied (Attribute Set Value)
   2. Email Requester (Email Send)
   3. Email HR (Email Send)
   4. Mark Workflow Complete (Workflow Complete)

Crafted by the Spark Development Network / License

1. **Workflow Type Navigation**
   Shows workflow categories and the workflows in each category. While you can nest categories, it's best to keep your structure relatively flat.
2. **Name**
   The name of the workflow type.
3. **Description**
   The description of the workflow. You should make this as detailed as possible since it acts as documentation for your workflow.
4. **Activities**
   This lists each of the activities configured in the workflow. For each activity it also lists their actions. This section is a form of auto-documentation if you provide clear names and descriptions.
5. **Copy**
   Many workflows are similar. To help you create new workflows that are similar to ones you already use, Rock allows you to make a copy of an existing workflow to use as a starting point for creating a new one.
6. **Action Buttons**
   Here you can launch the workflow, view a list of workflow instances and edit security. Security will determine who is able to view a workflow as well as manage (via the *Edit* permission) a workflow.

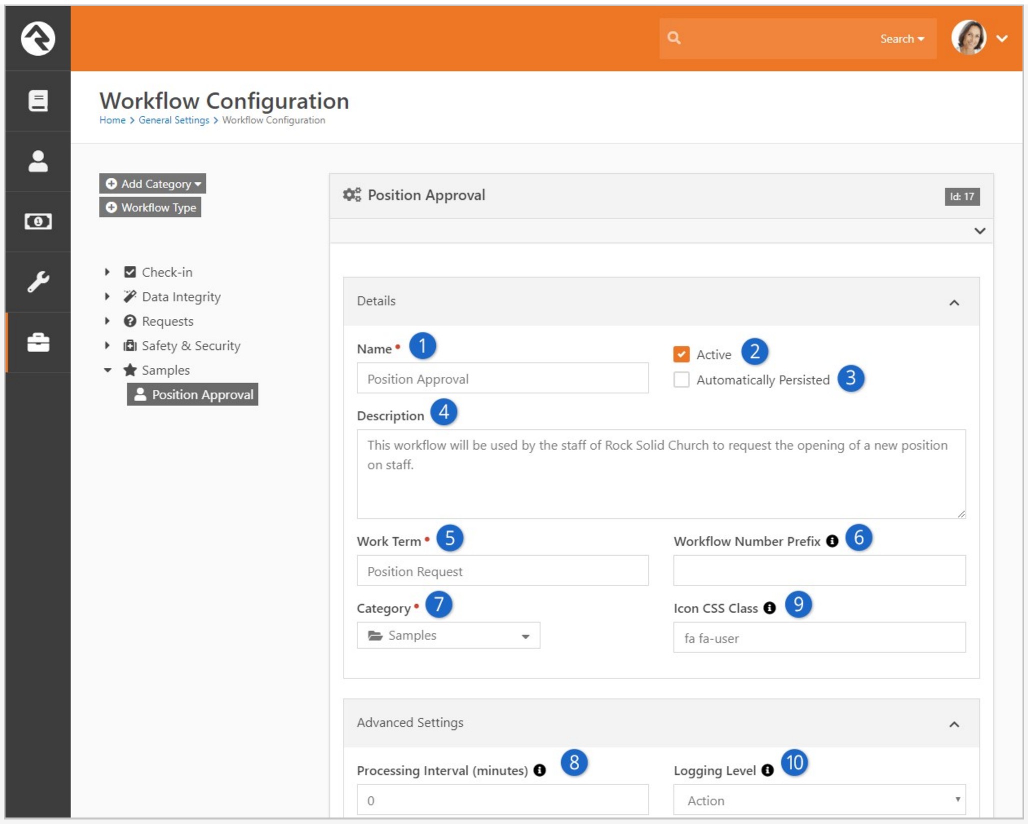
Editing A Workflow

Clicking the `Edit` button takes you to the edit screen for that workflow. This screen is made up of several sections, which are explained below.

Details

The first section is the *Details* section. Let's take a look at what it includes.

Workflow Type Details



Name: Position Approval

[✓] Active

[ ] Automatically Persisted

Description: This workflow will be used by the staff of Rock Solid Church to request the opening of a new position on staff.

Work Term: Position Request

Workflow Number Prefix:

Category: Samples

Icon CSS Class: fa fa-user

Processing Interval (minutes): 0

Logging Level: Action

1 **Name**
The name associated with the workflow type.

2 **Active**
Determines if the workflow type is active. This is helpful if you'd like to prevent new workflows from being created but would like to keep the workflow type around to view previous workflow instances.

3 **Automatically Persisted**
We'll discuss persisted vs. non-persisted workflows in detail below. Just know this is where you set the persistence type.

4 **Description**
While you may be tempted to skip over the description, we highly recommend that you enter a detailed description of your workflow covering when it should be used and its basic functionality.

5 **Work Term**
The term you will use to describe an instance of the workflow. For example, an IT work request may use the term *Request* while our inquiry example would use *Inquiry*.

6 **Workflow Number Prefix**
While every workflow will have a unique system ID, Rock also generates a workflow instance ID for the type. This makes for more logical and consistent IDs. The *Workflow Number prefix* allows you to optionally add an alpha-number prefix to the workflow instance ID. So instead of having an ID of, say, 00001 you can have something like POS0001.

7 **Category**
Workflows are grouped into categories for organization.

8 **Processing Interval**
Persisted workflows that are still active are run on a routine basis to see if there are any actions that can be completed. How often your workflow is run depends on this setting. To reduce the overhead on your server, you'll want to set this interval wisely.

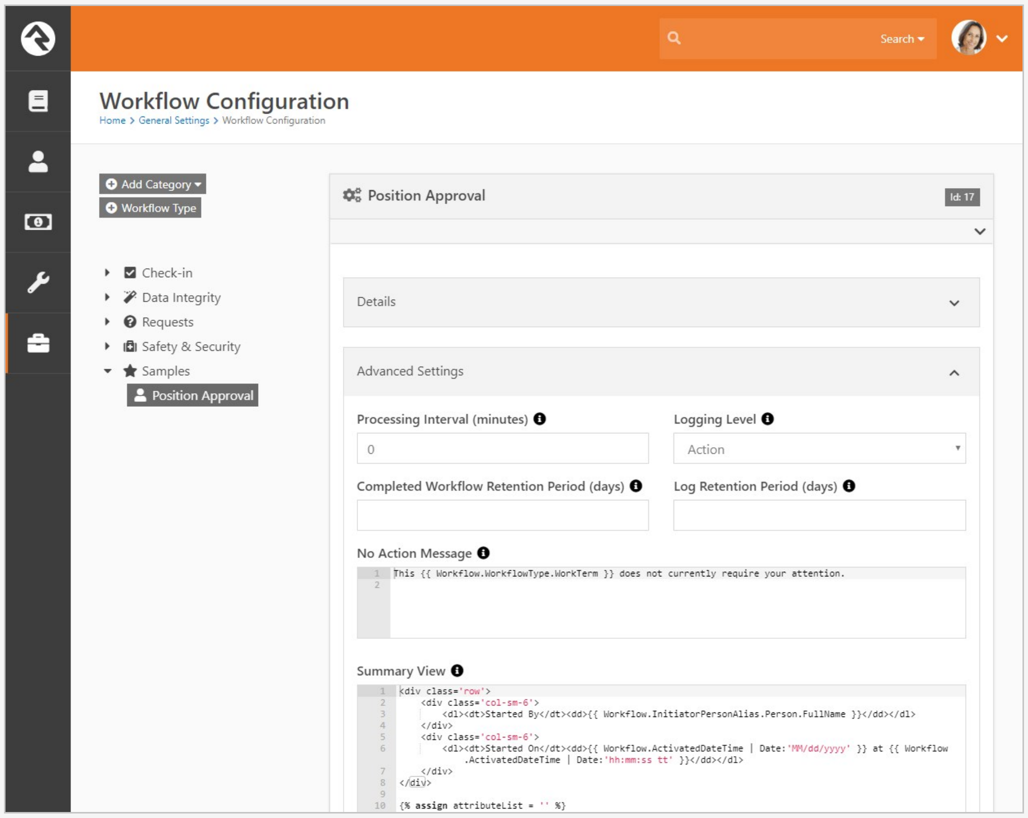
9 **Icon CSS Class**
You can provide a CSS icon to help distinguish your workflow type from others. By default, Rock supports the Font Awesome icon set . These icons should be in the form *fa fa-users*.

10 **Logging Level**
Logging is used to help debug (find logic errors) in your workflows. You can set the logging level to match the verboseness you need (none is nothing while action is pretty much everything).

Advanced Settings

The next section is the *Advanced Settings* section.

Workflow Advanced Settings



Processing Interval (minutes): 0
Logging Level: Action
Completed Workflow Retention Period (days):
Log Retention Period (days):
No Action Message

```
This {{ Workflow.WorkflowType.WorkTerm }} does not currently require your attention.
```

Summary View

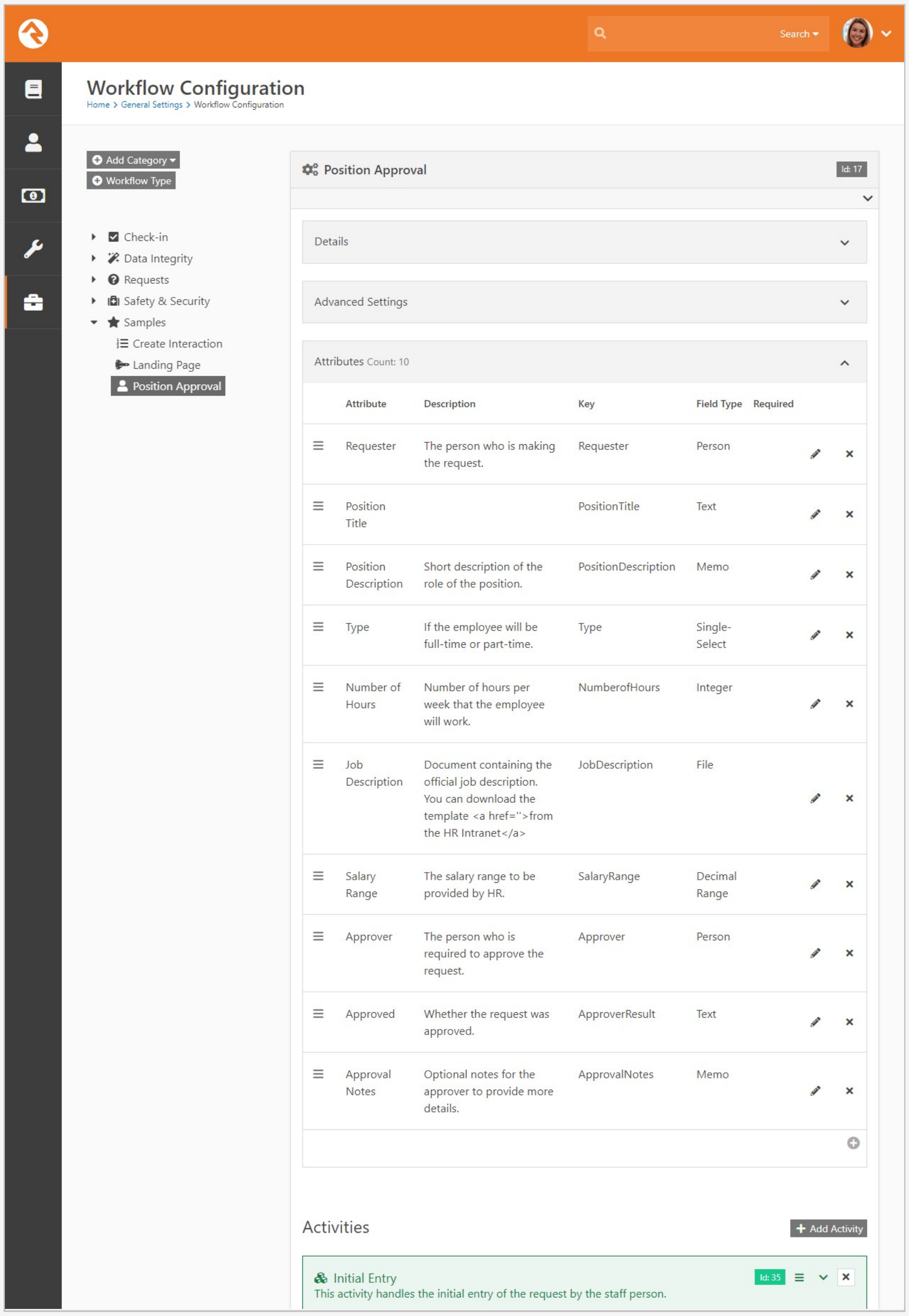
```
<div class='row'>
    <div class='col-sm-6'>
        <dl><dt>Started By</dt><dd>{{ Workflow.InitiatorPersonAlias.Person.FullName }}</dd></dl>
    </div>
    <div class='col-sm-6'>
        <dl><dt>Started On</dt><dd>{{ Workflow.ActivatedDateTime | Date:'MM/dd/yyyy' }} at {{ Workflow
            .ActivatedDateTime | Date:'hh:mm:ss tt' }}</dd></dl>
    </div>
</div>

{% assign attributeList = '' %}
```

The *Advanced Settings* section of the *Workflow Configuration* has some useful options, including the *Completed Workflow Retention Period* field. This is where you can specify how many days you want to keep a workflow before it's deleted. Over time your list of workflows will grow, including some workflows you'll only need for a limited amount of time. By default workflows are never deleted, so you might find your list getting cluttered. Using the *Completed Workflow Retention Period* option allows you to routinely and automatically clean up your workflows. If you never want the workflow to be removed, leave the field blank.

Attributes

Next is the *Attributes* section.

Workflow Type Attributes



Workflow Configuration

Home > General Settings > Workflow Configuration

Position Approval Id: 17

Attributes Count: 10

Attribute	Description	Key	Field Type	Required
Requester	The person who is making the request.	Requester	Person	
Position Title		PositionTitle	Text	
Position Description	Short description of the role of the position.	PositionDescription	Memo	
Type	If the employee will be full-time or part-time.	Type	Single-Select	
Number of Hours	Number of hours per week that the employee will work.	NumberofHours	Integer	
Job Description	Document containing the official job description. You can download the template <a href=''>from the HR Intranet</a>	JobDescription	File	
Salary Range	The salary range to be provided by HR.	SalaryRange	Decimal Range	
Approver	The person who is required to approve the request.	Approver	Person	
Approved	Whether the request was approved.	ApproverResult	Text	
Approval Notes	Optional notes for the approver to provide more details.	ApprovalNotes	Memo	

Activities

Initial Entry Id: 35

This activity handles the initial entry of the request by the staff person.

Attributes are the data elements your workflow needs to be able to process. In this section you configure each of these elements and define the types of data they will store (i.e., text, numbers, dates, people, groups, etc.)

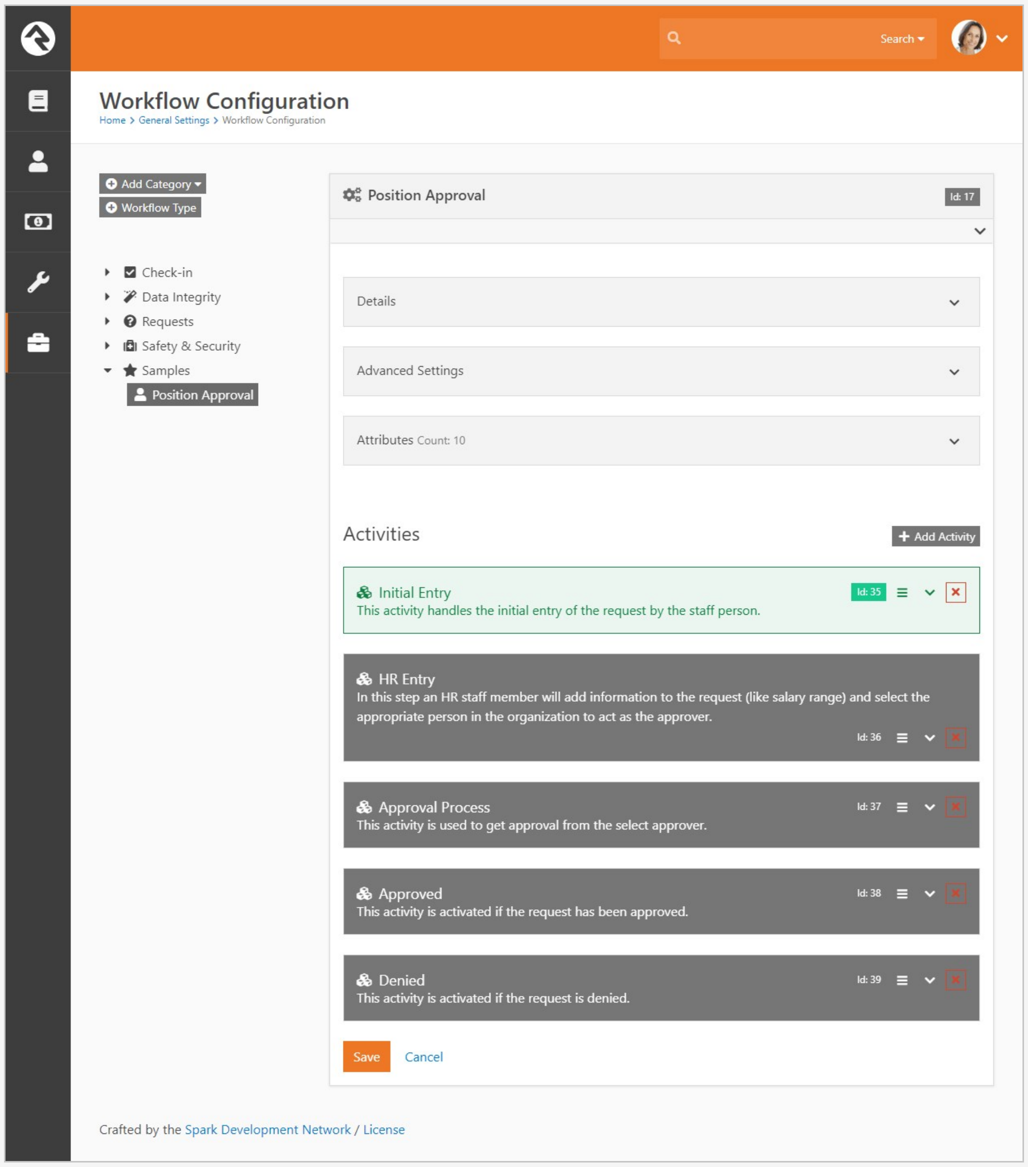
While it might be tempting to rush and define your attributes quickly by providing only a name and field type, it's wise to slow down and provide a good description of how the attribute will be used in the workflow. Trust us, you'll thank yourself later. Also, consider if a default value would make sense in your workflow.

> **Save Time:**
> Sometimes adding a good default value for your attribute can save steps in your workflow as you will only need to set the value of an attribute if a change is needed.

Activities/Actions

Finally, there is the *Activities* section.

Workflow Type Activities


Workflow Configuration
Home > General Settings > Workflow Configuration

Position Approval Id: 17

Activities

Initial Entry Id: 35
This activity handles the initial entry of the request by the staff person.

HR Entry Id: 36
In this step an HR staff member will add information to the request (like salary range) and select the appropriate person in the organization to act as the approver.

Approval Process Id: 37
This activity is used to get approval from the select approver.

Approved Id: 38
This activity is activated if the request has been approved.

Denied Id: 39
This activity is activated if the request is denied.

Crafted by the Spark Development Network / License

Activities and actions are the meat and potatoes of workflows. They control the flow logic that your workflow will use when it's processed. While we'll be talking about activities and actions in detail later, know that this is where you'll configure them for your workflow types.

> Tip:
> Be sure to check out the Workflow Control section in the Workflow Actions Documentation for more details.

As you build more complex workflows you might start to get confused about which box is an activity and which is an action. Just remember activities have a group of cubes () next to their titles while actions have a single cube ().

Workflow Import/Export

Sometimes a workflow you create is worth sharing with those outside of your organization to other community members using Rock! You can do this by using the workflow export option.

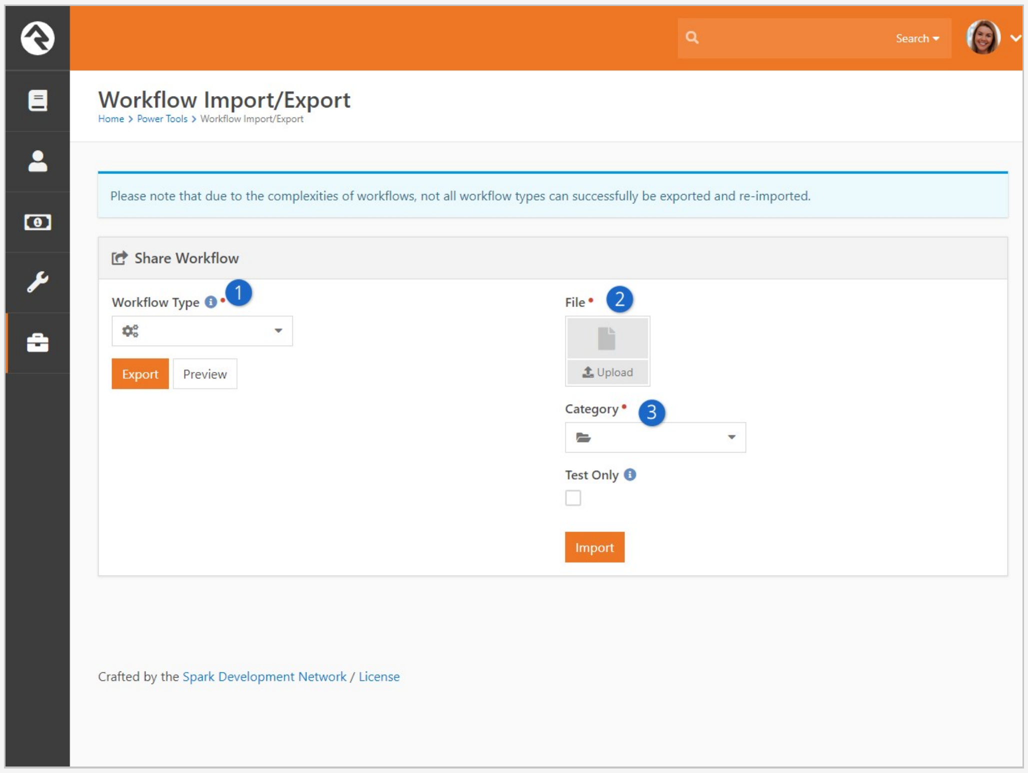
Navigate to `Admin Tools > Power Tools > Workflow Import/Export`.

Import/Export Details

Now we all know that workflows can be complex and very specific to your organization. In the event that you have a workflow that isn't so custom, this ability can be a game changer for those in the Rock community. Below is a general overview on the Export/Import page.

> **Some workflows may not export right:**
> Complex workflows may have issues being exported! Be sure to always double check your export before sharing it. Feel free to use the Rock Demo site to test it out if you'd like!

Workflow Import/Export



Please note that due to the complexities of workflows, not all workflow types can successfully be exported and re-imported.

Crafted by the Spark Development Network / License

1 **Workflow Type**
This dropdown will populate with all of the current available workflows. Navigate through the options to find yours to export.

2 **File**
If you are the one receiving a workflow, click the upload button to find the .json file to Import. OR drag and drop the file directly to the file window.

3 **Category**
Choose the desired category the imported workflow will load to.

Activities

Now that we have taken a tour of the workflow configuration screen, let's start talking turkey. Activities are groupings of actions that work together to complete a unit of work. If you think of your workflow as a flow chart on paper, activities would be the boxes (generally speaking) while the actions would be the logical steps needed to execute the task.

There really is no right answer regarding how many activities a workflow should have. Like a box of Legos, you can use different pieces in different ways and still end up with the same output. The best way to get a feel for activities is experience. Before we walk through building a sample workflow though, let’s look at some of the basic configuration options for activities.

Activation

Activities won't run until they are activated by the workflow engine. There are two ways that an activity is activated:

1. **Start-up:** You can configure certain activities to be activated when a workflow starts.
2. **Action Activation:** If an activity is not activated at start-up then it must be activated by an action on an activity that was.

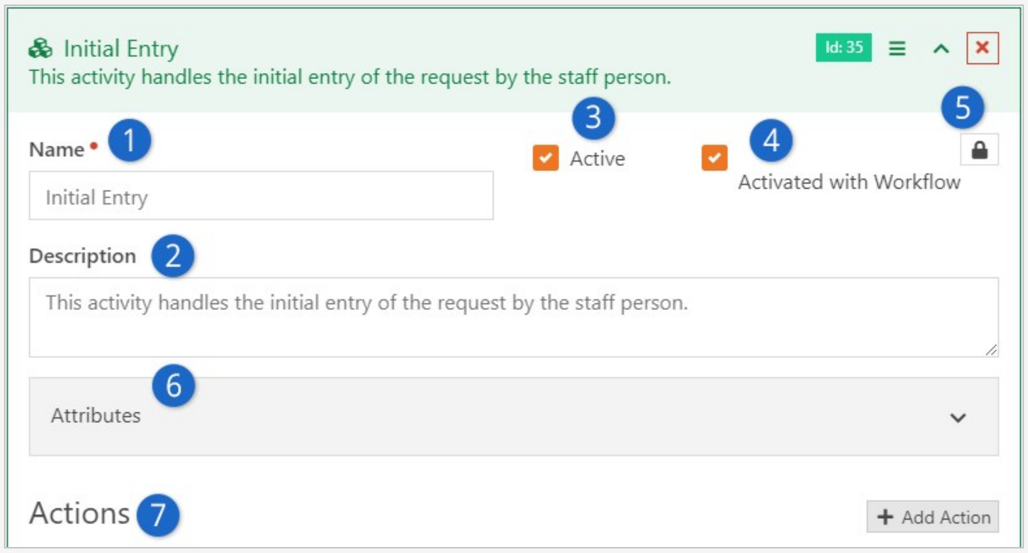
Simply defining an activity doesn't guarantee that it will ever be executed. If it is not activated with the start of a workflow and no action ever activates it, it will never run.

> Activities Don't always Run:
>
> It's not uncommon for an activity to never run. In many workflows the flow control logic you define might only run certain activities based on the input provided.

Configuring Activities

When you add a new activity to a workflow type, you'll see a new blank activity panel. The configuration options are shown below:

Activity Overview


Initial Entry
This activity handles the initial entry of the request by the staff person.
Id: 35
Name
Initial Entry
Active
Activated with Workflow
Description
This activity handles the initial entry of the request by the staff person.
Attributes
Actions
Add Action
1
2
3
4
5
6
7

1 **Name**
Be sure to give your activity a descriptive name. If this was a flow chart on paper, the name would be the text in the box.

2 **Description**
Don't cheat yourself by providing a short description. It's often helpful to outline both the purpose of the activity and the flow logic that will be needed to execute.

3 **Active**
This tells the workflow if this activity is active in the configuration. While this isn't used very often, it can be helpful if you need to temporally disable an activity from running.

4 **Activated With Workflow**
This defines whether the activity will be activated at the beginning of a workflow. If this is set to 'True', the activities header will be shown in green to help you quickly identify startup activities.

5 **Security**
Security on a workflow activity helps with activities that must interact with a user (mainly through entry forms). More on entry forms can be found below. Note: The security icon for an activity will not appear until after the workflow has been saved.

6 **Attributes**
Activities can have their own attributes. When they're needed, they're configured here. More on when to use activity attributes is discussed in the next section.

7 **Actions**
This is where you'll define the actions that make up your activity. The order of the actions is important because they will execute in the order you provide.

Activity Attributes

Like workflow attributes, activity attributes allow you to store the data needed to execute your workflow. Many workflows can get by with using just workflow attributes. But there will be times when a specific activity is run more than once. If you'd like to keep track of data for each execution, you'll need to define activity attributes. The data in these attributes is only available within the specific activity instance.

As an example, say you had an activity that seeks approval for a purchase order. As a part of the approval, you might want to allow the approver to enter notes about their decision. You'd also like your workflow to allow the approval step to be re-run until an approval is received (for instance the approver may deny it at first, it goes back to the requester who edits it and then re-submits it for approval). If the approval note was stored as a workflow attribute, it would be overwritten each time the approval activity is run. When defined as an activity attribute, each instance of the activity would have its own instance of the note attribute.

Assignment

Activities can be assigned to a specific person or group. While security determines who's allowed to view or edit an activity, the assignment describes who is responsible to complete it. Assignment only comes into play for activities that must interact with a user (mainly through entry forms). Assignments help workflows prompt the right people to enter the data that is needed. We'll touch more on assignments in the Working With Entry Forms chapter.

Actions

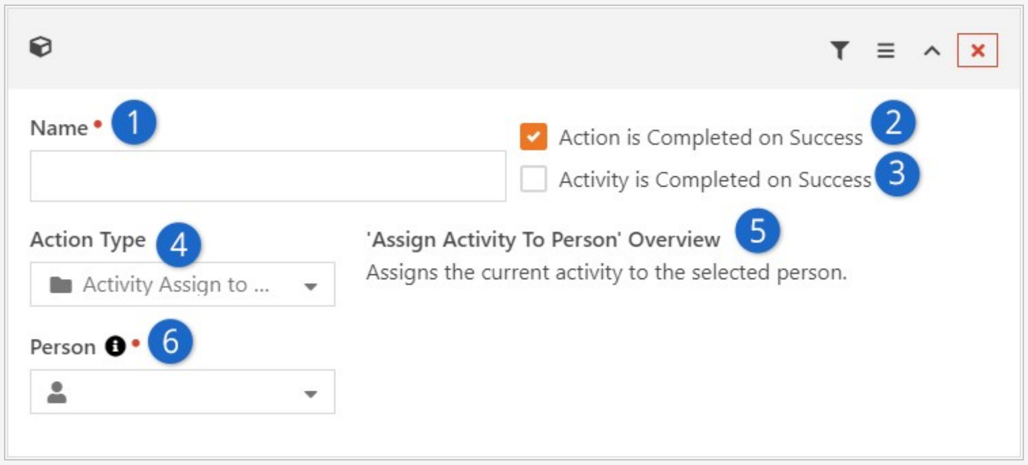
As previously mentioned, actions are the worker bees of workflows. They are broken down into categories to make them easier to find. Let's take a look at the basic configuration settings of actions and then look at the actions that come out of the box.

> Action Order Is Important:
> Be careful to define your actions in the right order because that's the order in which they'll execute.

Action Configuration

While each action type will have configuration settings unique to its purpose, all actions do share some similar configuration settings. Let's look at these common settings. They can be found by clicking on an activity in a workflow configuration.

Action Overview

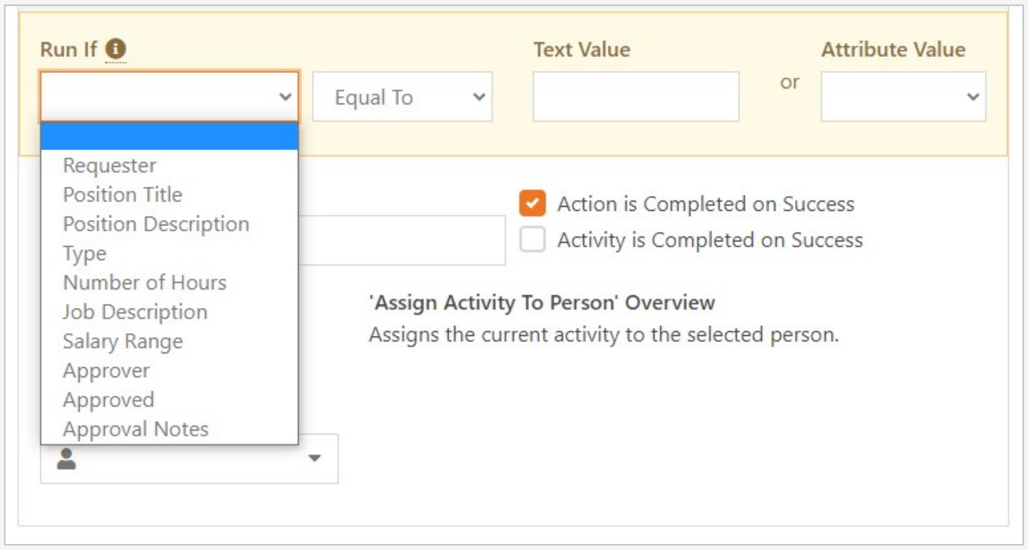

Name
1
Action is Completed on Success
2
Activity is Completed on Success
3
Action Type
4
Activity Assign to ...
'Assign Activity To Person' Overview
5
Assigns the current activity to the selected person.
Person
6

1. **Name**
   The name is used to help describe the task the action is performing within the activity.
2. **Action Is Completed On Success**
   This tells the workflow engine if the action should be marked as completed when it runs successfully. For most actions you'll want to be sure this is set to 'True'. But there may be times when you want an action to be performed every time the workflow is processed.
3. **Activity Is Completed On Success**
   This setting tells the workflow engine to mark the entire activity as complete if the action successfully runs. This has the effect of keeping all following actions from running. If you are familiar with programming logic, this is similar to a *break* statement.
4. **Action Type**
   This is where you configure what type of action you want to perform. Each of these action types are discussed below.
5. **Assign Message**
   The message that is displayed based on the Action Type selected.
6. **Person Picker**
   The person this workflow is assigned to.

Action Filters

We learned about ways we can control the flow of actions inside an activity in the previous section. *Action Filters* provide us with another powerful way of controlling the flow logic of a workflow. They allow you to only run an action if the value of an attribute meets a criteria you define.

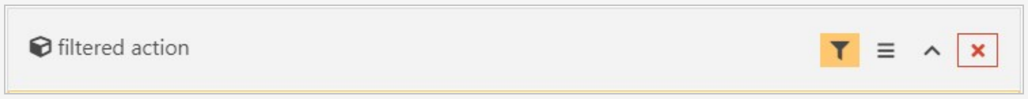
Action Filter Configuration



'Assign Activity To Person' Overview
Assigns the current activity to the selected person.

When an action has a filter configured, the filter icon will display in yellow to help you know that a filter is present.

Action Filter Notation



Although most of the filter match criteria are self-explanatory, the *Regular Expression* is possibly unfamilar to you. Simply put, a regular expression is a sequence of characters that define a search pattern and using them you can achieve powerful *text* matching. For example, if you wanted to match a prayer request text for the phrases "suicide", "SUICIDE", "kill himself", "kill herself", or "kill myself" you could use a regular expression value of `(?i)(suicide|kill (h|m)(\S+)(\s*)self)`. You can find a Microsoft Regular Expression Quick Reference online and use a tool like https://www.debuggex.com/ for testing your new creations.

> **Important**
> Knowing the text value of an attribute is key when setting up filters. For text attributes this is pretty straight forward. For other types of attributes you need to know more about their internals. For instance, a 'Boolean' attribute's text value would be 'True' or 'False' while a person attribute would be the guid of the person alias. The full list of different attribute field types can be found here.

Default Action Types

For a listing of Rock's workflow actions, see the Workflow Actions Documentation. Here we'll outline a number of actions that come with Rock, providing tips on when and how to use them. Screenshots of the settings are also provided.

A Note About Check-in Actions:

Many of Rock's workflow actions are specifically used for check-in workflows. We won't be covering them in this manual since very few people will be using them.

Launching Workflows

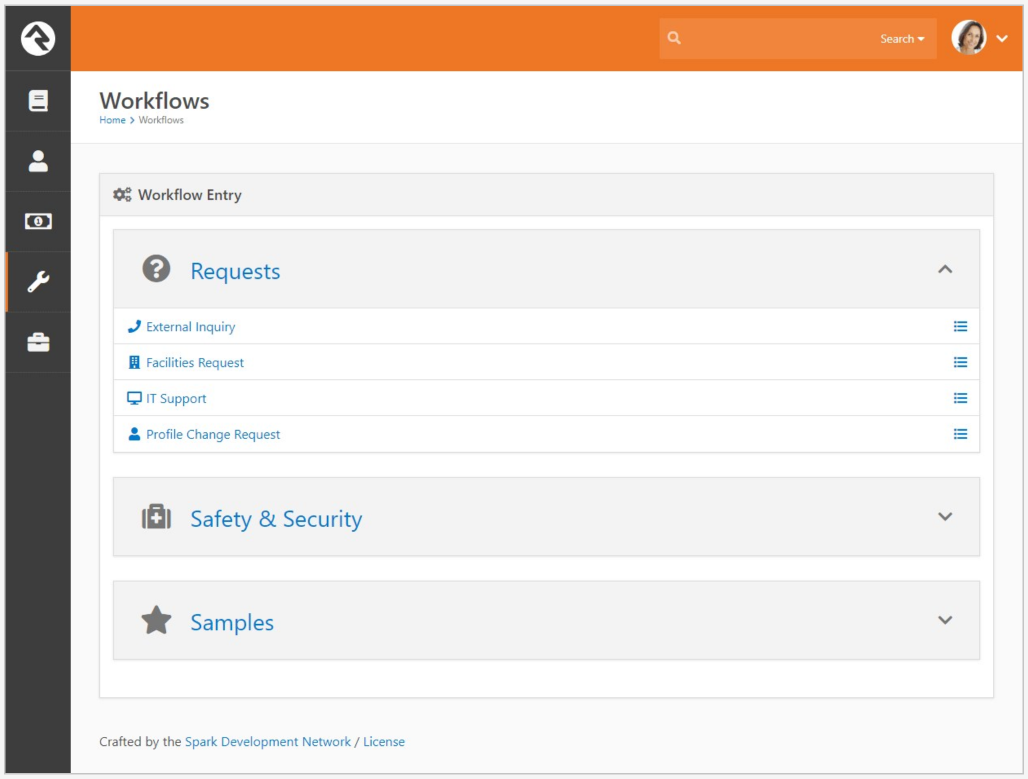
Once you have your workflow types configured, you're ready to start using them. There are several ways you can launch a workflow. Each method is discussed in detail below.

Workflow Entry Block

Many workflows will start with a user filling out a form. This is certainly the case with workflows like IT requests, facility operations requests, HR approvals, etc. There are a couple of ways you can launch these types of workflows.

Workflow Category Browser

Workflow Category Browser

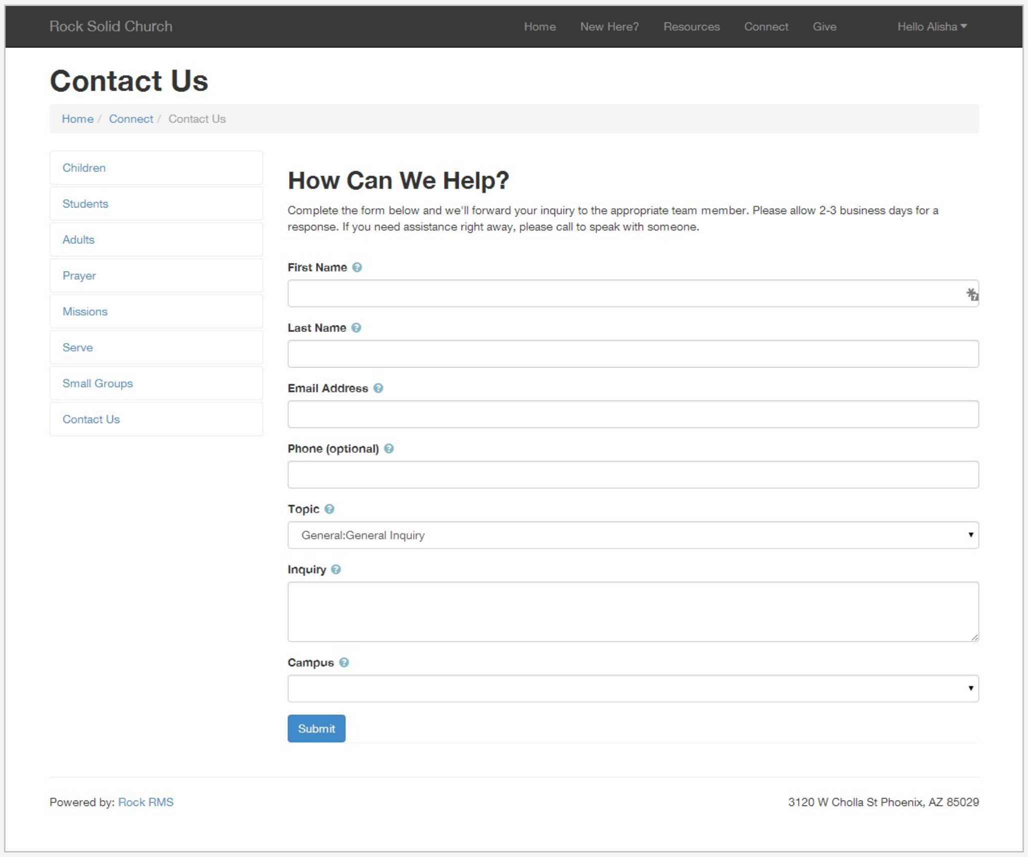


Rock ships with a workflow entry page under `Tools > Workflows`. This page displays a list of workflow categories with the ability to expand the category to view its workflows. Clicking on a workflow will launch it and show its first entry screen. You can configure which categories are displayed by modifying the block's settings. Category and workflow security will also be used to personalize the display to the specific rights of a user.

Direct Workflow Entry

There may be times when you'd like to place a specific page in the navigation that takes you directly to a workflow entry screen. An example of this is the *Contact Us* page on the external website. This page has been configured with the *Workflow Entry* block. One of the block settings of this block type allows you to define a specific workflow type to launch when the page loads. This is a great way to include links to important workflows into your internal and external sites. The magic of this technique is that the user doesn't even know that they are interacting with a workflow.

Workflow Entry


Rock Solid Church | Home | New Here? | Resources | Connect | Give | Hello Alisha

Contact Us

Home / Connect / Contact Us

Children
Students
Adults
Prayer
Missions
Serve
Small Groups
Contact Us

How Can We Help?

Complete the form below and we'll forward your inquiry to the appropriate team member. Please allow 2-3 business days for a response. If you need assistance right away, please call to speak with someone.

First Name

Last Name

Email Address

Phone (optional)

Topic: General:General Inquiry

Inquiry

Campus

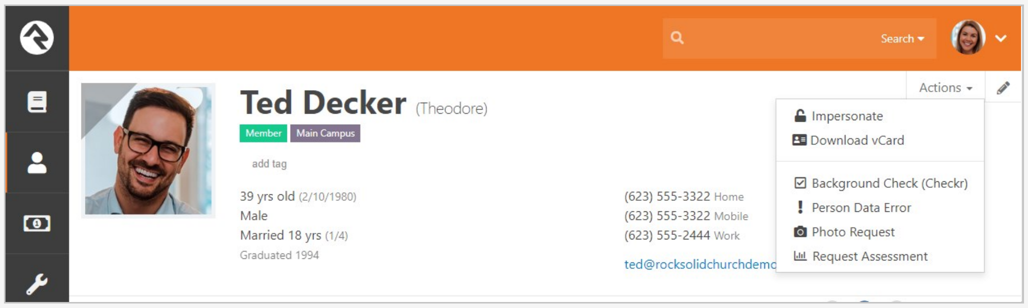
Submit

Powered by: Rock RMS

3120 W Cholla St Phoenix, AZ 85029

Person Profile Actions

Person Profile Actions


Search

Ted Decker (Theodore)

Member | Main Campus

add tag

39 yrs old (2/10/1980)
Male
Married 18 yrs (1/4)
Graduated 1994

(623) 555-3322 Home
(623) 555-3322 Mobile
(623) 555-2444 Work
ted@rocksolidchurchdemo

Actions
Impersonate
Download vCard
Background Check (Checkr)
Person Data Error
Photo Request
Request Assessment

You may have noticed that there is a menu on the *Person Profile* page labeled *Actions*. On this menu there is an item entitled *Person Data Error*. Clicking on this menu item launches a workflow that allows the user to report any data integrity issues to the proper team. You can easily add your own workflows to this list.

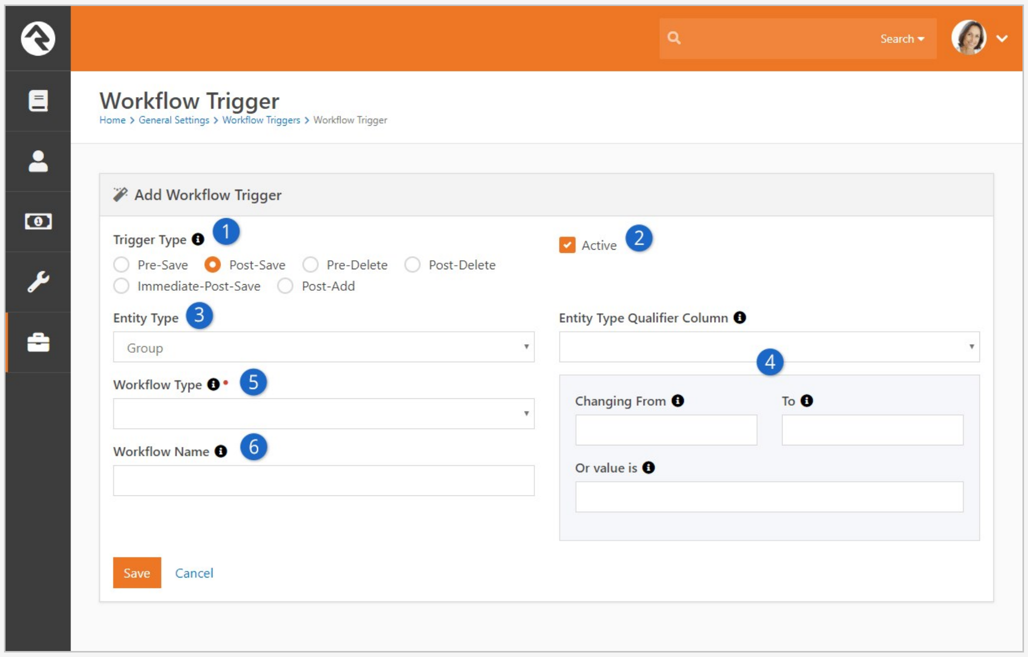
The first step is to create your new workflow. Be sure one of your first actions uses the *Set Attribute From Entity*. This takes the person whose record is being viewed and places them in a workflow attribute (this attribute should be of type *Person*). Your workflow can then add any processing logic from there.

Once your workflow is defined, you can add the workflow to the action menu. This is done by editing the block settings back on the *Person Profile* page. There you'll see a setting for selecting workflows to add. Workflow security will be considered when building the list so you can make sure that only certain people are able to launch the workflow.

Entity Triggers

Have you ever thought: "Gee, I wish I could do something every time someone saves a person in the database." Well, with Rock you can! Entity triggers can be configured under `Admin Tools > General Settings > Workflow Triggers`.

Workflow Triggers



1 **Trigger Type**
You must select when the trigger will be fired. See notes below on the difference between pre and post events.

2 **Active**
Determines if the workflow trigger is currently active. This is helpful if you want to temporarily suspend the trigger but don't want to delete it.

3 **Entity Type**
Select the entity type you'd like to add the trigger to (e.g., Person, Group, etc.)

4 **Entity Type Qualifier Column / Value (Optional)**
There are times when you will only want to run your trigger in certain situations. For instance, you may want to run a post-save trigger when groups of group type *Serving Team* are saved. In this case you would set the *Qualifier Column* to *GroupTypeId* and the *Qualifier Value* to *23* (the GroupTypeId for Serving Teams). These settings allow you to simplify your workflows for specific use cases.

5 **Workflow Type**
Select the workflow type you want to launch. Be sure that this workflow saves the passed entity to an attribute so that it has access to the value being saved or deleted.

6 **Workflow Name**
This setting provides a workflow name for the workflows that are created.

Caution: Saving While Saving

Be careful not to set up a triggered workflow that updates the entity that is *actively* being saved (for example, a pre-save or immediate-post-save trigger on a person that fires a workflow to update a property on that person). This can cause a loop that creates an out-of-memory condition which will make your server administrator pretty upset.

There are other related combinations that are also unsupported. Just because you can doesn't mean you should.

Pre vs. Post Trigger Events

You might be wondering what the difference is between a pre and post event trigger. There is a difference and it's pretty important that you select the right type.

Pre triggers launch the workflow before the save or delete occurs. The benefit of a pre trigger is that you can keep the save from occurring through the logic of your workflow. If your workflow returns with an error message, the save or delete will be aborted. Except in a few places, there is no means for these error messages to bubble up for someone to see, so keep that in mind when using pre triggers.

One downfall of pre triggers is that if they are initiated by someone working in Rock, that person must wait for the workflow to launch and complete before the save is completed. Because of this, you'll want to be sure that your workflows are simple and quick.

Post triggers, on the other hand, can't keep a save or delete from occurring. By the time they launch, the save or delete has already been done. These triggers are launched but Rock does not wait to hear back from them before moving on. This keeps workflow performance quick.

You Can Have More Than One:

You can have more than one trigger for each entity type. This saves you from having to lodge all your logic in one workflow.

Use Post Triggers Whenever Possible.

Because of their speed, try to use post triggers whenever possible.

Warning: Saved or Not Yet Saved

Even with an immediate-post-save trigger, if you try to fetch the triggered entity from the database in your workflow, there is a possibility that its data has not *yet* been written to the database. If it's critical that you know the exact values of the entity at the moment the workflow runs, you should capture the entity property in question with the *Attribute Set From Entity* action using `{{ Entity.PROPERTYNAME }}` or capture the whole object into a text attribute using `{{ Entity | ToJSON }}`. You can then safely refer to the correct value in subsequent actions.

Save the Entity First

When using the *Attribute Set From Entity* action, you may be tempted to try to do more than it was designed to do. For example, if the entity you are working with is a *Group Member*, you might try to get the group's name using `{{ Entity.Group.Name }}`. Except the ".Group" property is not guaranteed to be there -- only the .GroupId will be there. In fact, even the `{{ Entity }}` will be gone after the initial activity.

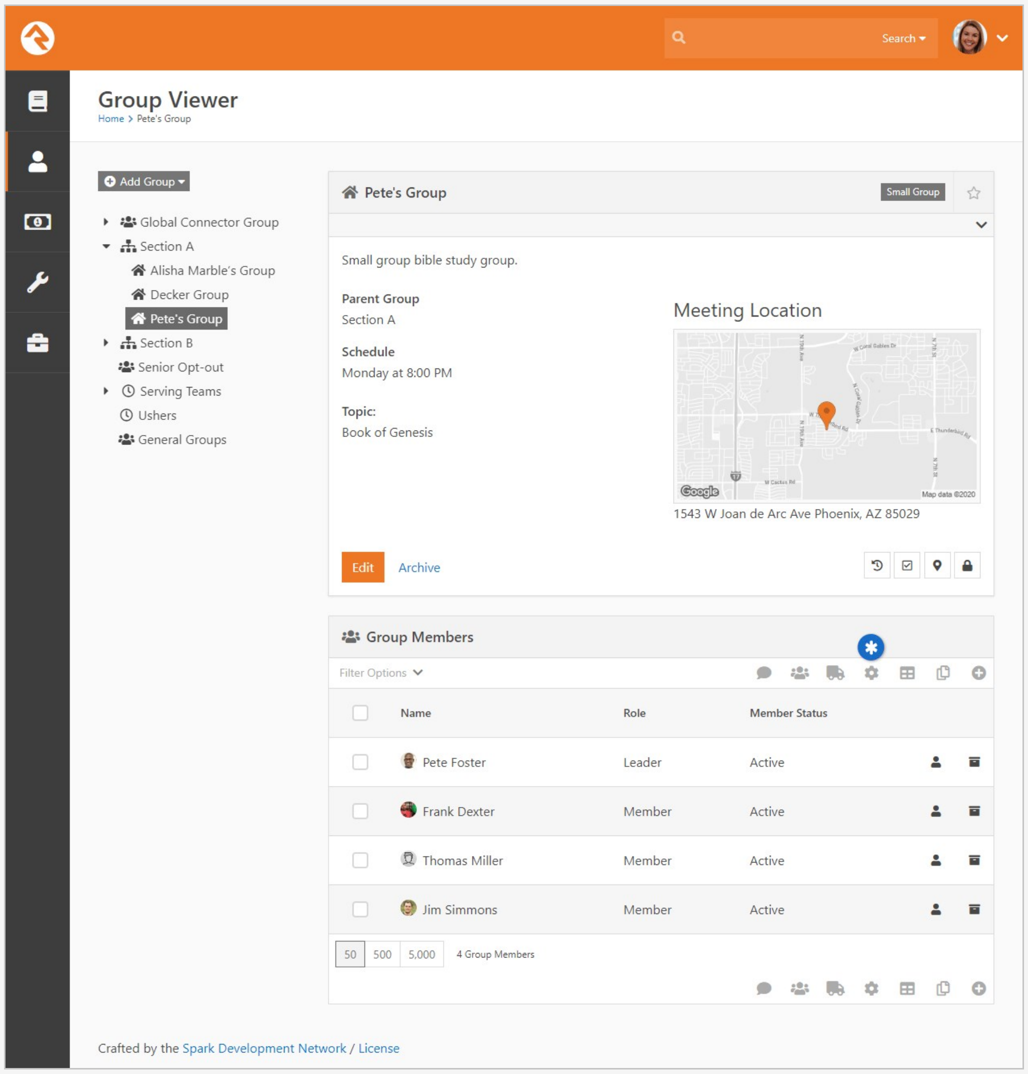
Therefore we recommend you always collect your Entity properties in your few first actions, before doing anything else. And second, if you need other related items you should load them explicitly. So, to get that group name you can use `{{ Entity.GroupId | GroupById | Property: 'Name' }}`

Launch Workflow from Grid

Have you ever wanted to run a workflow for each item on a grid? Perhaps you need to send every person in a data view into an on-boarding process. Or, maybe you want to send them an email asking for them to complete a form. If you can write a workflow for it, you can now launch it right from a grid.

When you’re looking at a grid, all you have to do is click the ⚙ button to launch a workflow for each item listed. Let’s look at an example using the *Group Viewer* page to launch workflows for the members of a group.

Launch Workflow From Grid - Group Viewer



Small group bible study group.

Parent Group
Section A

Schedule
Monday at 8:00 PM

Topic:
Book of Genesis

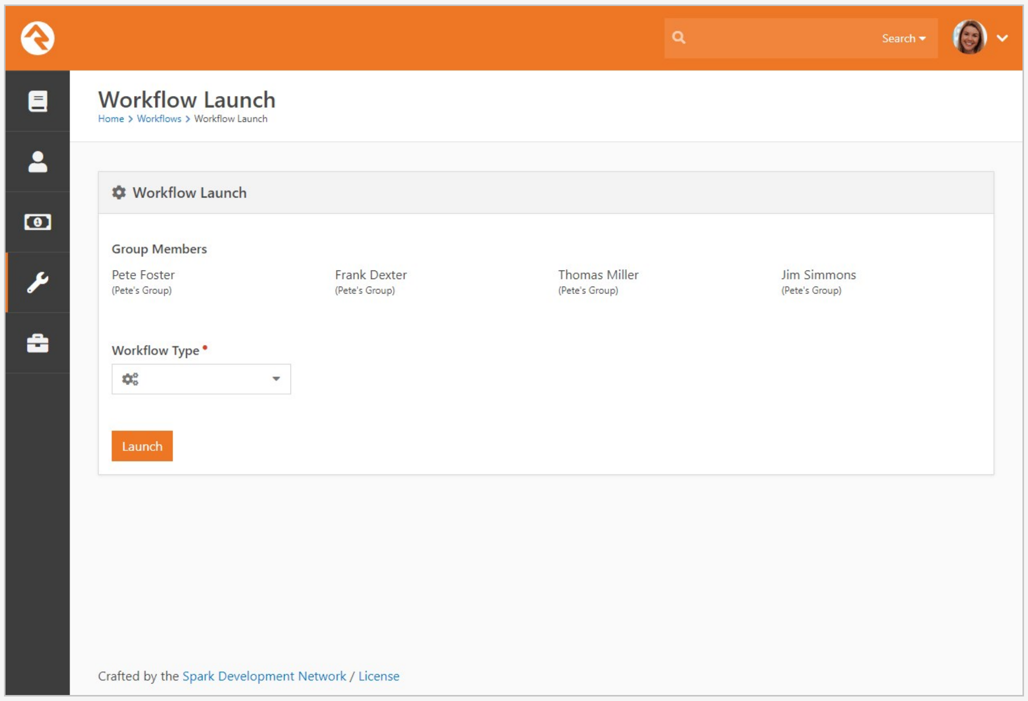
Meeting Location

1543 W Joan de Arc Ave Phoenix, AZ 85029

Name	Role	Member Status
Pete Foster	Leader	Active
Frank Dexter	Member	Active
Thomas Miller	Member	Active
Jim Simmons	Member	Active

Clicking the ⚙ icon will take you to the Workflow Launch page pictured below.

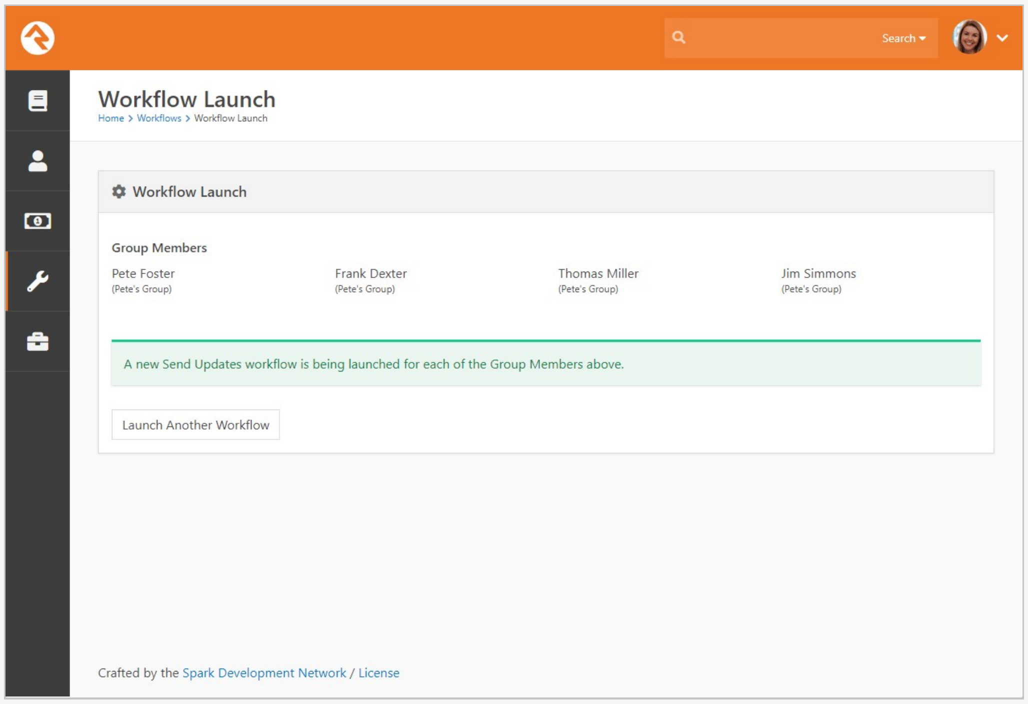
Launch From Grid - Launch Page



When you arrive at the Workflow Launch page you will see that each of the items (in this case, group members) are listed at the top. This lets you check to make sure you have the correct information and the right items.

Beneath the list of items you'll notice the *Workflow Type* picker. Use this to select the type of workflow to launch, then click the `Launch` button. Keep in mind that a new workflow will be launched for **each** item listed above the picker. In our group member example, four new workflows will be launched.

Launch From Grid - Workflow Launch Success



Workflow Launch

Workflow Launch

Group Members

Pete Foster (Pete's Group)
Frank Dexter (Pete's Group)
Thomas Miller (Pete's Group)
Jim Simmons (Pete's Group)

A new Send Updates workflow is being launched for each of the Group Members above.

Crafted by the Spark Development Network / License

You'll notice above that after clicking the `Launch` button, you can choose to *Launch Another Workflow* without leaving the page. This makes it easy to quickly launch multiple workflows for the same set of items.

Extending Workflow Launches

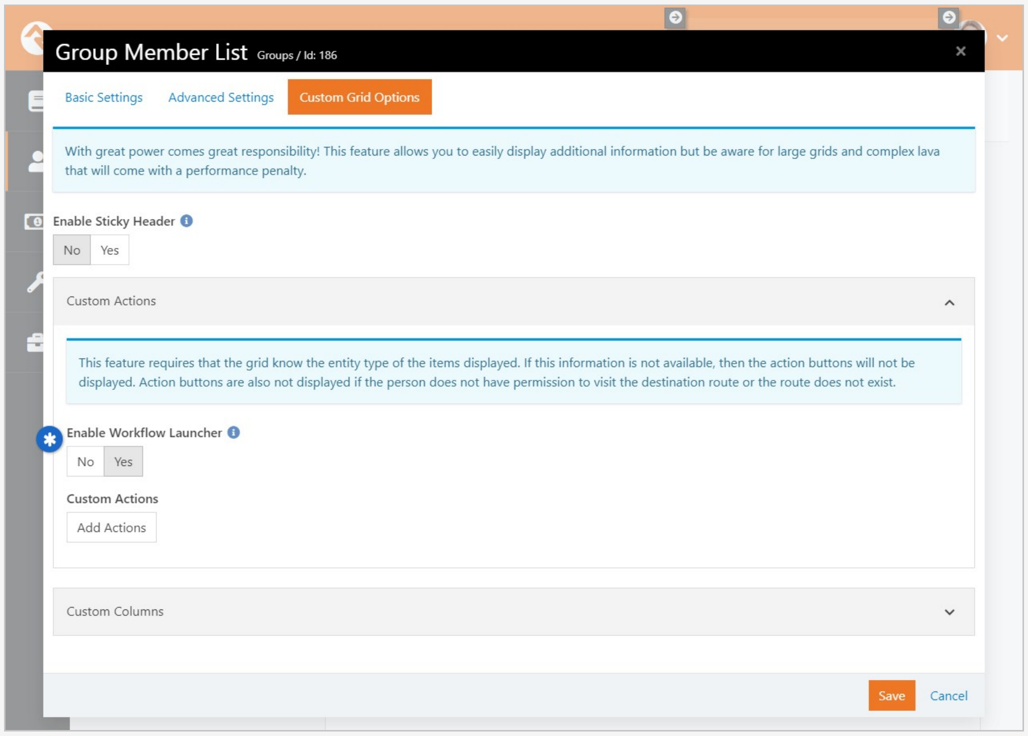
Hopefully you're already seeing tremendous power of launching workflows from grids. Now, let's look at how you can extend this feature even further.

In the above *Group Viewer* example we launched workflows for group members by clicking the ⚙ icon in the grid. This feature is enabled by default for grids in Rock, so you'll see the ⚙ in other places and it will work the same way. But, you might not want all of your grids to allow workflow launches. Or you might want to force a specific workflow to be launched instead of having a person choose from the picker. You can configure all this, and more, using the options described below.

Disable Workflow Launcher

We'll start with the most basic setting. If you want to turn off the launcher for a grid, all you have to do is access the *Custom Grid Options* in the grid's block settings. Expand the *Custom Actions* area and set *Enable Workflow Launcher* to "No".

Enable Workflow Launcher


Group Member List Groups / Id: 186

With great power comes great responsibility! This feature allows you to easily display additional information but be aware for large grids and complex lava that will come with a performance penalty.

Enable Sticky Header: [✓ No] Yes

Custom Actions

This feature requires that the grid know the entity type of the items displayed. If this information is not available, then the action buttons will not be displayed. Action buttons are also not displayed if the person does not have permission to visit the destination route or the route does not exist.

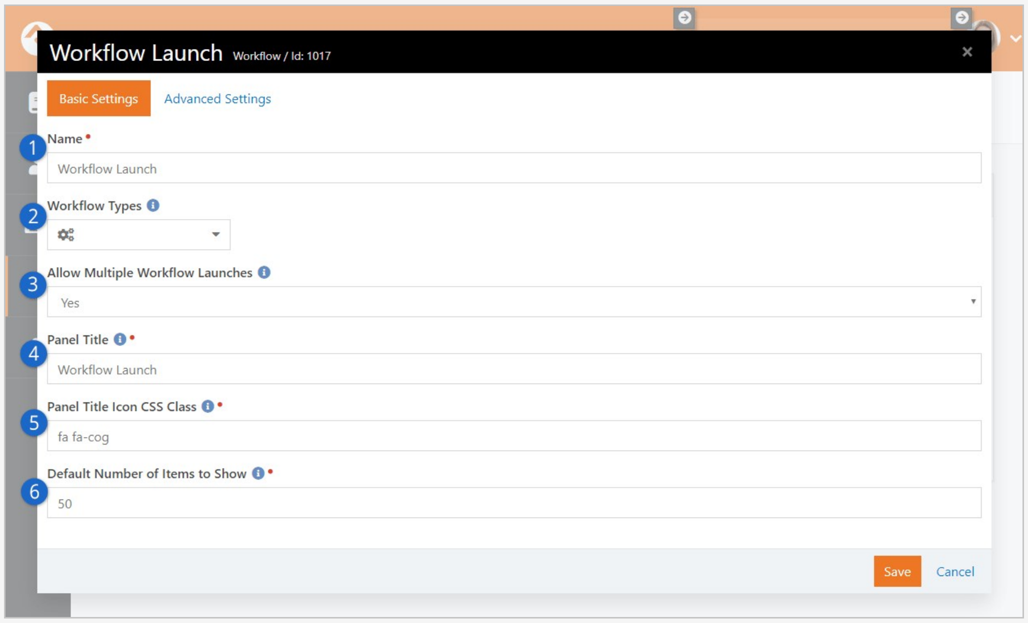
Enable Workflow Launcher: No [✓ Yes]

This setting only applies to the workflow launcher that ships with Rock, as described in the above section. Custom actions (see below) are not impacted by this setting, even if those actions will launch a workflow.

Workflow Launch Block

The workflow launcher takes you to a page with a workflow launch block. This block is the bridge that connects the items in the grid to the workflows you want to launch for those items. Administrators can change the block settings to customize different aspects of the process.

Launch From Grid - Block Settings



Name *
Workflow Launch

Workflow Types

Allow Multiple Workflow Launches
Yes

Panel Title *
Workflow Launch

Panel Title Icon CSS Class *
fa fa-cog

Default Number of Items to Show *
50

1 **Name**
The name of the block can be changed here.

2 **Workflow Types**
You can specify which workflow types are able to be launched by adding them to this list. The Custom Actions section below describes a different way to restrict the workflow types on this block, so be sure to check that out before using this field.

3 **Allow Multiple Workflow Launches**
If set to yes, this allows launching multiple different types of workflows. After one is launched, the block will allow the individual to select another type to be launched.

4 **Panel Title**
The title that's displayed in the block panel can be customized here.

5 **Panel Title Icon CSS Class**
By default the ⚙ icon is displayed, but you can change it here.

6 **Default Number of Items to Show**
This setting controls the number of items that will be shown on the launch screen. If the number of items in the grid is higher than this number, a count will be displayed showing how many aren't visible.

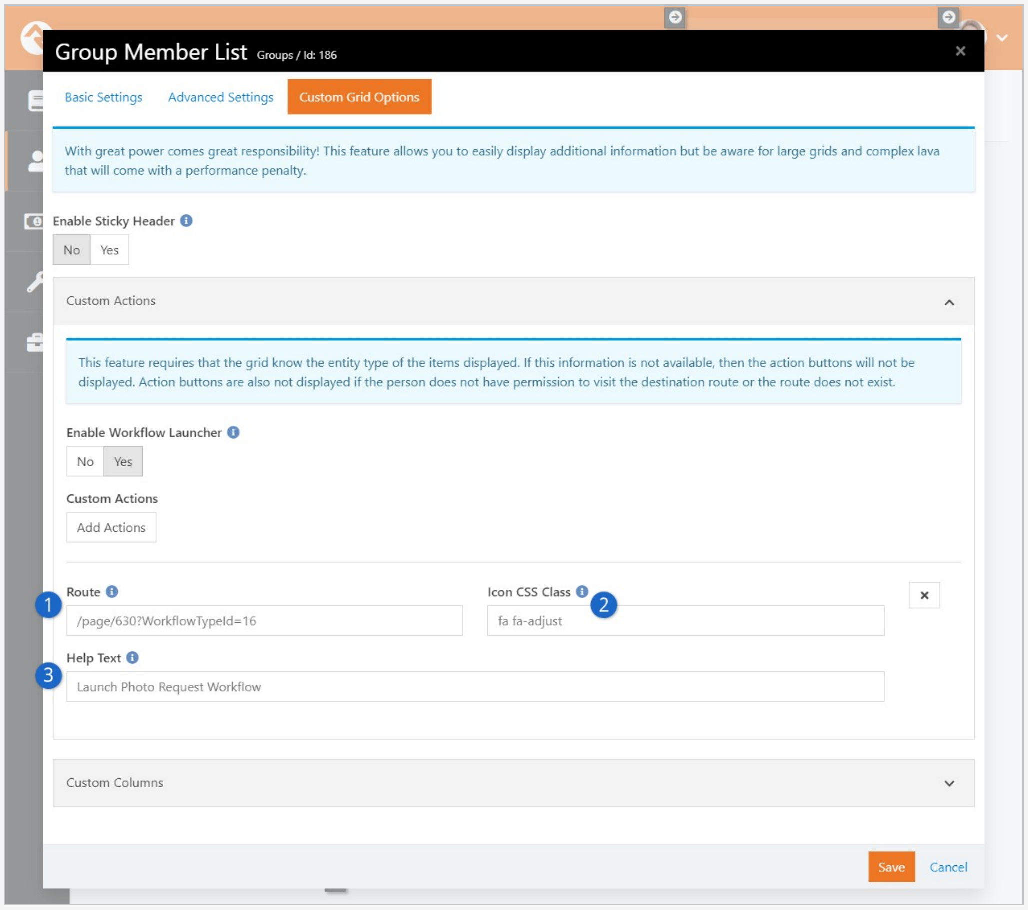
The workflow launch block is programmed to look for a Workflow Type in the URL. If it finds one, then the block will automatically lock the workflow picker to that workflow type, and it can't be changed by the person. This is like specifying a workflow type in the block's settings, except the block will dynamically change according to the query string in the URL. We'll show you what this looks like in the Custom Action Routes section below.

Custom Actions

Now let's talk about the doors that open with *Custom Actions*. Custom actions work like the workflow launcher example described above. An icon gets added to the grid (like the ⚙ icon) and then a person can click it to start your custom action. The key is that this isn't limited to group members or workflows. Custom actions can take items from any grid and use them wherever you need.

Custom actions are added in the grid's block settings, in the same *Custom Grid Options* area as the *Enable Workflow Launcher* setting described above. Click the `Add Actions` button to create a new custom action.

Grid Block Settings - Add Custom Action


Group Member List Groups / Id: 186

Basic Settings | Advanced Settings | Custom Grid Options

With great power comes great responsibility! This feature allows you to easily display additional information but be aware for large grids and complex lava that will come with a performance penalty.

Enable Sticky Header: [✓ No] Yes

Custom Actions

This feature requires that the grid know the entity type of the items displayed. If this information is not available, then the action buttons will not be displayed. Action buttons are also not displayed if the person does not have permission to visit the destination route or the route does not exist.

Enable Workflow Launcher: No [✓ Yes]

Custom Actions: Add Actions

Route: /page/630?WorkflowTypeId=16

Icon CSS Class: fa fa-adjust

Help Text: Launch Photo Request Workflow

Custom Columns

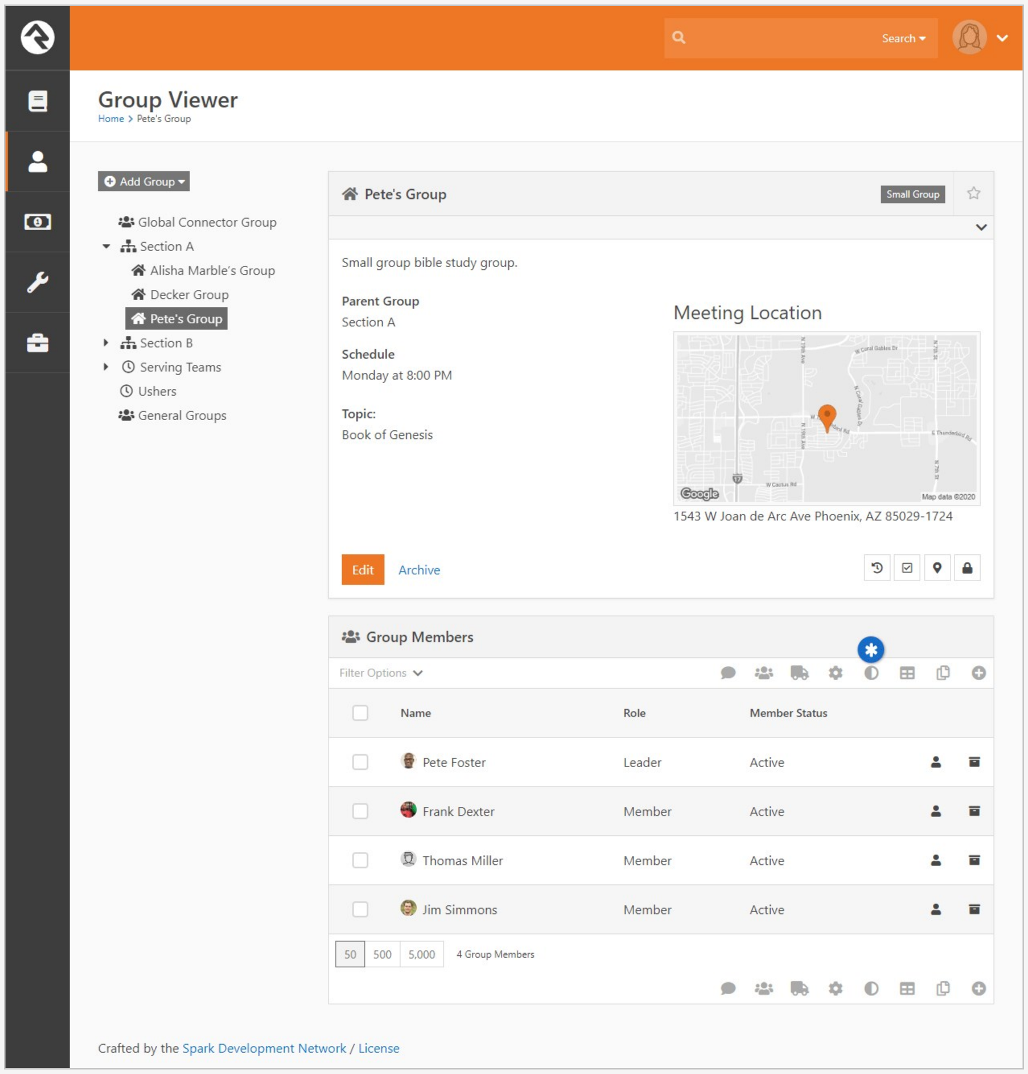
1 **Route**
The route you provide drives your custom action. People will be taken to the page provided when they initiate your custom action from the grid. If you want to launch a workflow for each grid item, the route should point to a page with a workflow launch block. We'll talk more about this field below.

2 **Icon CSS Class**
This is where you choose the icon that appears in the grid. People will click this icon to initiate your custom action.

3 **Help Text**
When a person hovers over the icon in the grid, this text will appear to indicate what action will be taken.

Using the configuration pictured above, the grid containing group members now has a new icon for the new action.

Group Viewer Grid With Custom Action Icon



Group Viewer

Home > Pete's Group

Add Group

- Global Connector Group
- Section A
  - Alisha Marble's Group
  - Decker Group
  - Pete's Group
- Section B
- Serving Teams
- Ushers
- General Groups

Pete's Group — Small Group

Small group bible study group.

Parent Group
Section A

Schedule
Monday at 8:00 PM

Topic:
Book of Genesis

Meeting Location

1543 W Joan de Arc Ave Phoenix, AZ 85029-1724

Edit Archive

Group Members

Filter Options

	Name	Role	Member Status
	Pete Foster	Leader	Active
	Frank Dexter	Member	Active
	Thomas Miller	Member	Active
	Jim Simmons	Member	Active

50 500 5,000 4 Group Members

Crafted by the Spark Development Network / License

Custom Action Routes

As noted above, there's more to the *Route* configuration than just taking the person to a different page. In the example configuration above, the route is `/page/630?WorkflowTypeId=16`. Let's break that down.

The first part of this action's route takes the person to a new page, which is `/page/630` in this example. We've added a workflow launch block to that page, so that's all we need for this route to work for launching workflows. In this case, a route of `/page/630` would result in a custom action that's very similar, maybe identical, to the workflow launch process described in the prior section.

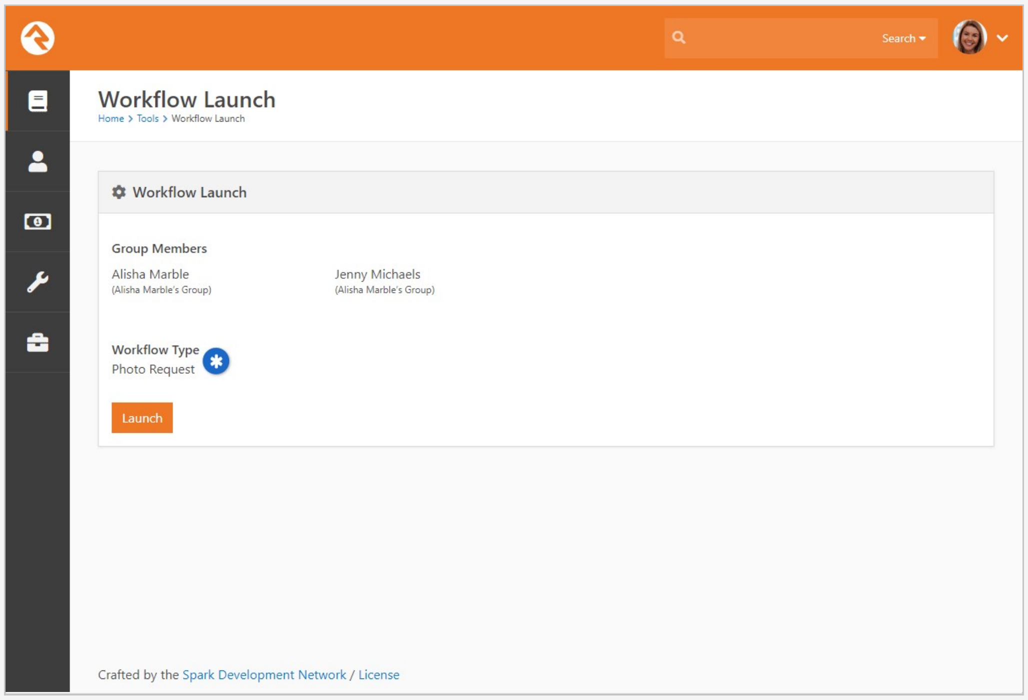
To take it a step further, don't forget that the workflow launch block checks for a workflow type in the URL. If it finds a workflow type, then it will lock the workflow picker to that type. All you need to do is add a query string parameter in the format of `?WorkflowTypeId=xx` to your route, and the block will automatically pick it up.

Populating Workflow Attribute Values

Starting in Rock v11.1, the workflow launch block will take any of the parameters that are in the query string and pass them into the workflow's matching attributes. So, if you have a group attribute in your workflow, you can pass the group's Id by including `?GroupId=12` in the route.

The example route we used above has an added parameter of `?WorkflowTypeId=16`. This tells the block to allow only the “Photo Request” workflow type.

Workflow Launch Block - Locked Workflow Type


Search
Workflow Launch
Home > Tools > Workflow Launch
Workflow Launch
Launch
Crafted by the Spark Development Network / License

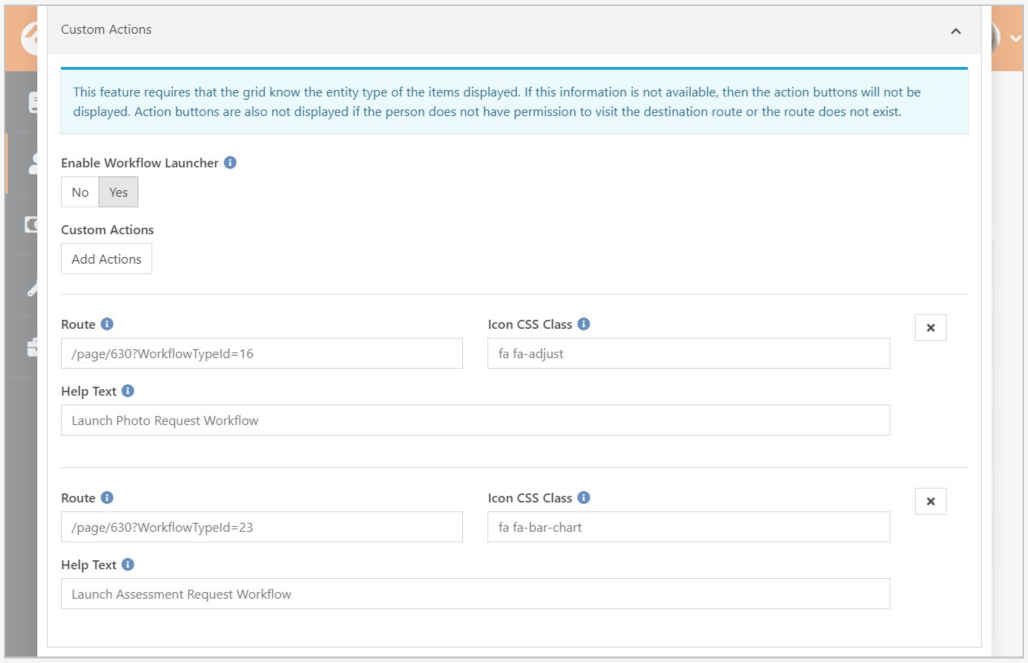
Group Members
Alisha Marble (Alisha Marble's Group)
Jenny Michaels (Alisha Marble's Group)

Workflow Type
Photo Request

This custom action now provides a dedicated icon people can use to launch only “Photo Request” workflows. Remember, the block itself hasn’t changed. It’s only locked to photo requests in this case because of the route in our custom action.

Keeping that in mind, you can add new icons to launch different types of workflows using the same target page and the same block. In addition to photo requests, you might add a new custom action for assessment requests, giving you an icon for each on the grid as pictured below.

Multiple Custom Actions



Custom Actions

This feature requires that the grid know the entity type of the items displayed. If this information is not available, then the action buttons will not be displayed. Action buttons are also not displayed if the person does not have permission to visit the destination route or the route does not exist.

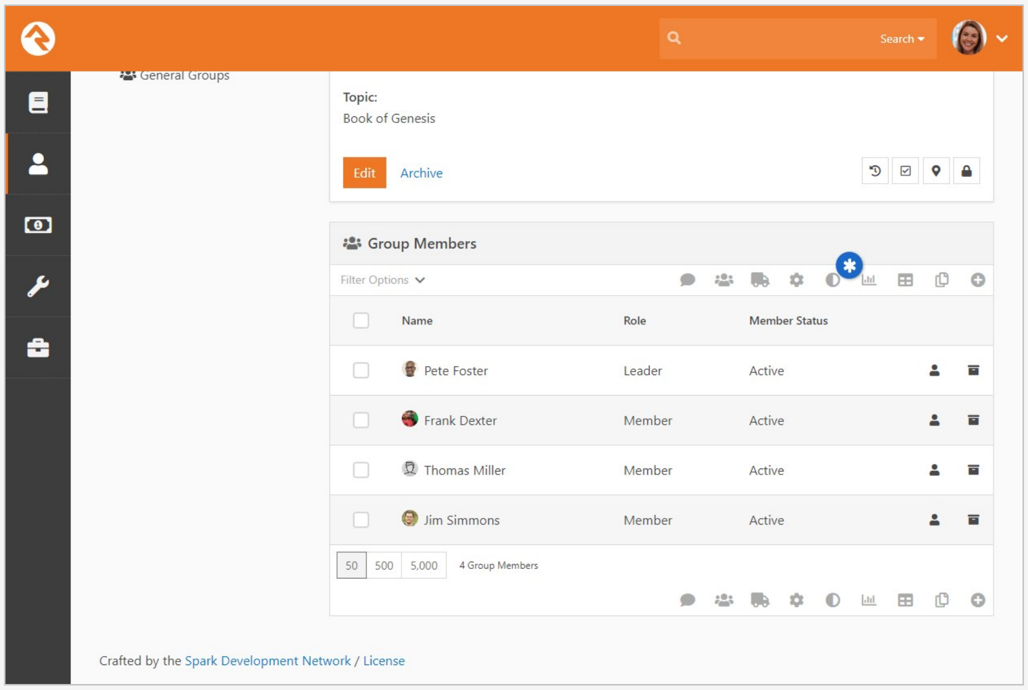
Enable Workflow Launcher: No [✓ Yes]

Custom Actions: Add Actions

Route	Icon CSS Class	Help Text
/page/630?WorkflowTypeId=16	fa fa-adjust	Launch Photo Request Workflow
/page/630?WorkflowTypeId=23	fa fa-bar-chart	Launch Assessment Request Workflow

Note in the screenshot above that both routes point to the same page, and the same workflow launch block. Adding the workflow type to the *Route* means that the ◑ icon will launch only photo requests, while the icon will launch only assessment requests.

Multiple Custom Actions - Launch From Grid Icons



Topic:
Book of Genesis

Name	Role	Member Status
Pete Foster	Leader	Active
Frank Dexter	Member	Active
Thomas Miller	Member	Active
Jim Simmons	Member	Active

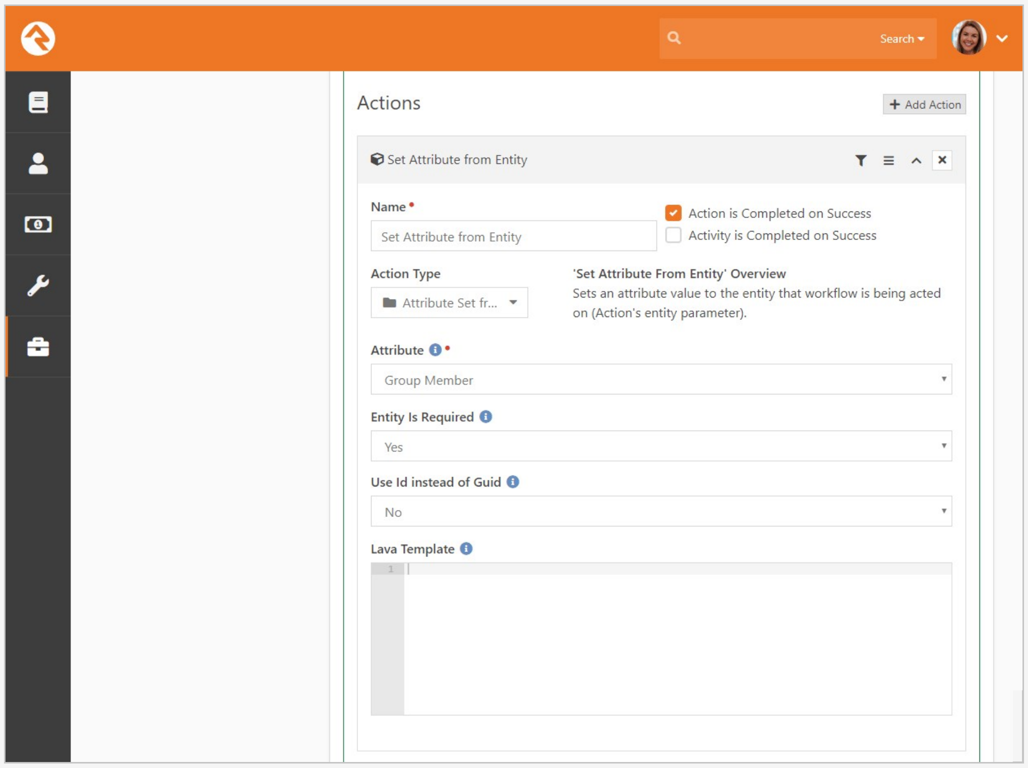
Working with Launched Workflows

In the above sections we've only been using workflow types that ship with Rock, launched from the *Group Members* grid. But you can use these features with other workflows, and with other grids.

When you’re building a workflow to launch from a grid, it’s important to know what types of items, or entities, the grid contains. The examples in prior sections were all working with *Group Member* entities because that’s the type of entity contained in the grid. Other grids may have different entities, like groups or people.

The key is knowing that the entity is passed from the grid to the workflow. That means the grid needs to know the type of entity it’s working with, but it also means your workflow needs to be designed to work with that type of entity. The best way to do that is to use the *Attribute Set from Entity* action, to capture each entity from the grid and assign it to a workflow attribute.

Attribute Set From Entity - Group Member



Name *
Set Attribute from Entity

[✓] Action is Completed on Success
[ ] Activity is Completed on Success

Action Type
Attribute Set fr...

'Set Attribute From Entity' Overview
Sets an attribute value to the entity that workflow is being acted on (Action's entity parameter).

Attribute *
Group Member

Entity Is Required
Yes

Use Id instead of Guid
No

Lava Template

You can see above that the group member from the grid will be assigned to the workflow's "Group Member" attribute. Now the entity can be referenced directly in the workflow, and you can use it in other actions. Remember, a workflow is launched for each entity in the grid. We're only processing a single group member in this workflow, not the set of group members.

Lifecycle Of A Workflow

Now that you're up to speed on how a workflow is launched you might be wondering how a workflow executes from there. Understanding the life cycle of a workflow is a key to building activities and actions that flow the way you expect them to.

When a workflow is launched, the workflow engine completes the following steps:

1. Activates each of the activities that are configured to be *Activated with Workflow*.
2. Proceeds to each newly activated activity in the order it was defined and runs its actions. If one of the actions does not complete successfully, it stops running the activity's actions. If they all complete, or if one of the actions is configured with *Activity is Completed on Success*, the activity is marked completed.

Now that a workflow is active, and not yet complete, the engine will attempt to re-run it on the interval defined by the workflow type using the following steps:

1. All uncompleted actions on an activity will be run in the order they are defined.
2. If the running of these actions has completed, the activity will be marked complete.

The active workflow will continue to be executed on the polling frequency until the workflow is marked complete.

> **Looking Under The Hood:**
> You might be wondering what keeps the workflow engine running on the interval schedule. Rock has a system job that is set to run every 10 minutes. This job can be viewed (but not edited) under `Admin Tools > System Settings > Jobs Administration`. Because this job only runs every 10 minutes, setting your workflow processing interval to less than 10 minutes will have no effect.

> **Be Aware of System Performance:**
> While it might seem nice to have your workflows execute on a short processing interval, it could impact system performance if you have a lot of active workflows to run. Consider the process interval that best fits the need of your workflow type.

Auto Closing Workflows

Have you ever wanted to give up on something? Well, this could be the case with certain types of workflows. In situations where it's no longer sensible to continue running a workflow, you now have an option to clear the slate. The `Rock.Job.CompleteWorkflows` job can be added in the *Jobs Administration* screen. You can configure it to run against one or more workflow types that are older than a certain number of minutes. Alternatively, if you leave the *Expiration Age* setting empty, it will complete all workflows when the job is run. This can be useful if you schedule the job to run at certain times of the year. You can also provide a custom *Close Status* (such as 'Expired') to know at a glance why the workflow was completed when you are reviewing them at a later date.

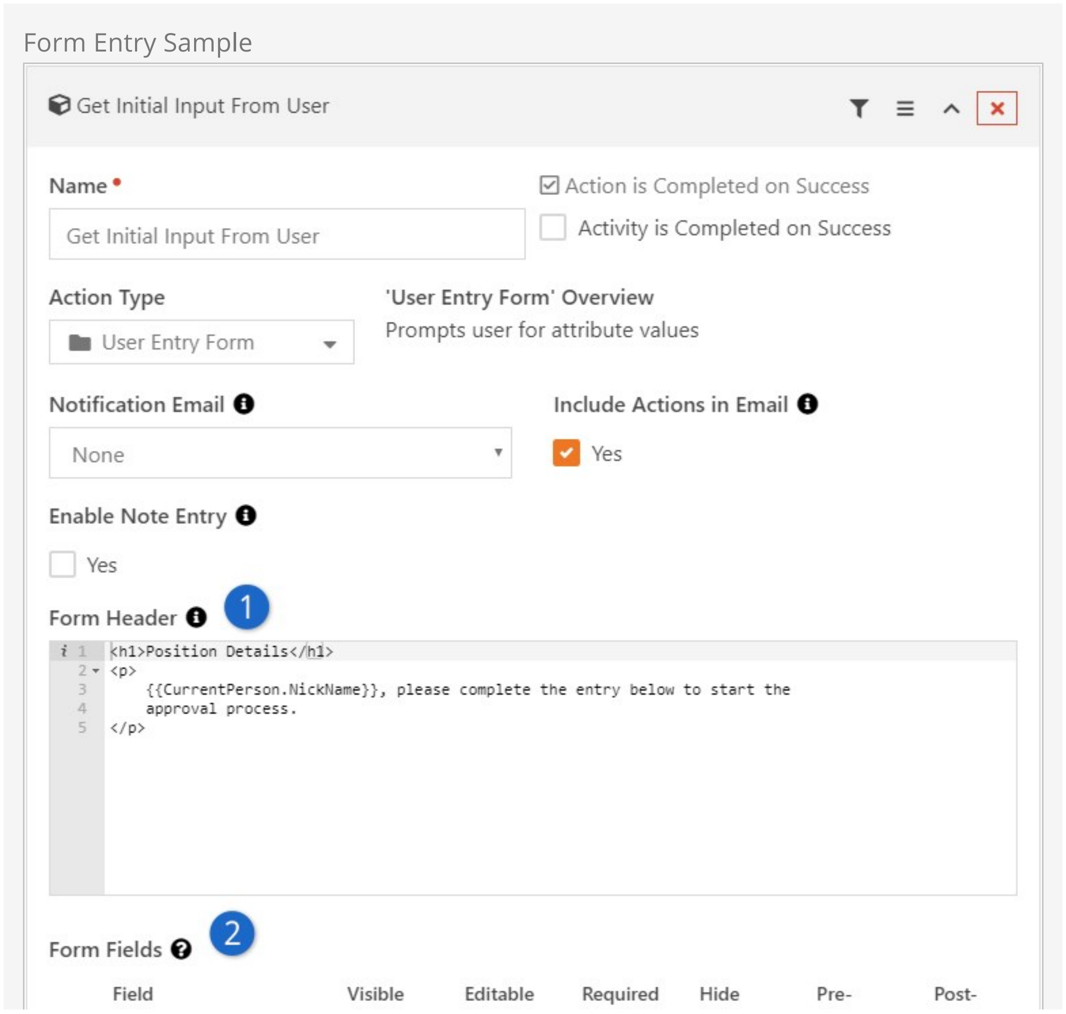
Working With Entry Forms

Workflow entry forms are one of the most exciting features of Rock's workflow engine. They allow workflows to interact with users in some powerful ways. With them, you can create mini-applications that once required a dedicated developer to produce.

The backbone of form entry is the *Form* action. This is what presents the form to the user. Let's unpack its usage a little more and see what its capabilities are.

Form Action

To help us understand this action better, let’s go back to the simple *HR Position Request* example from earlier, specifically the first entry form that Ted used to start the request. Below is a screenshot of the entry form action used in that workflow.



Name: Get Initial Input From User

Action is Completed on Success [✓]
Activity is Completed on Success [ ]

Action Type: User Entry Form

'User Entry Form' Overview
Prompts user for attribute values

Notification Email: None

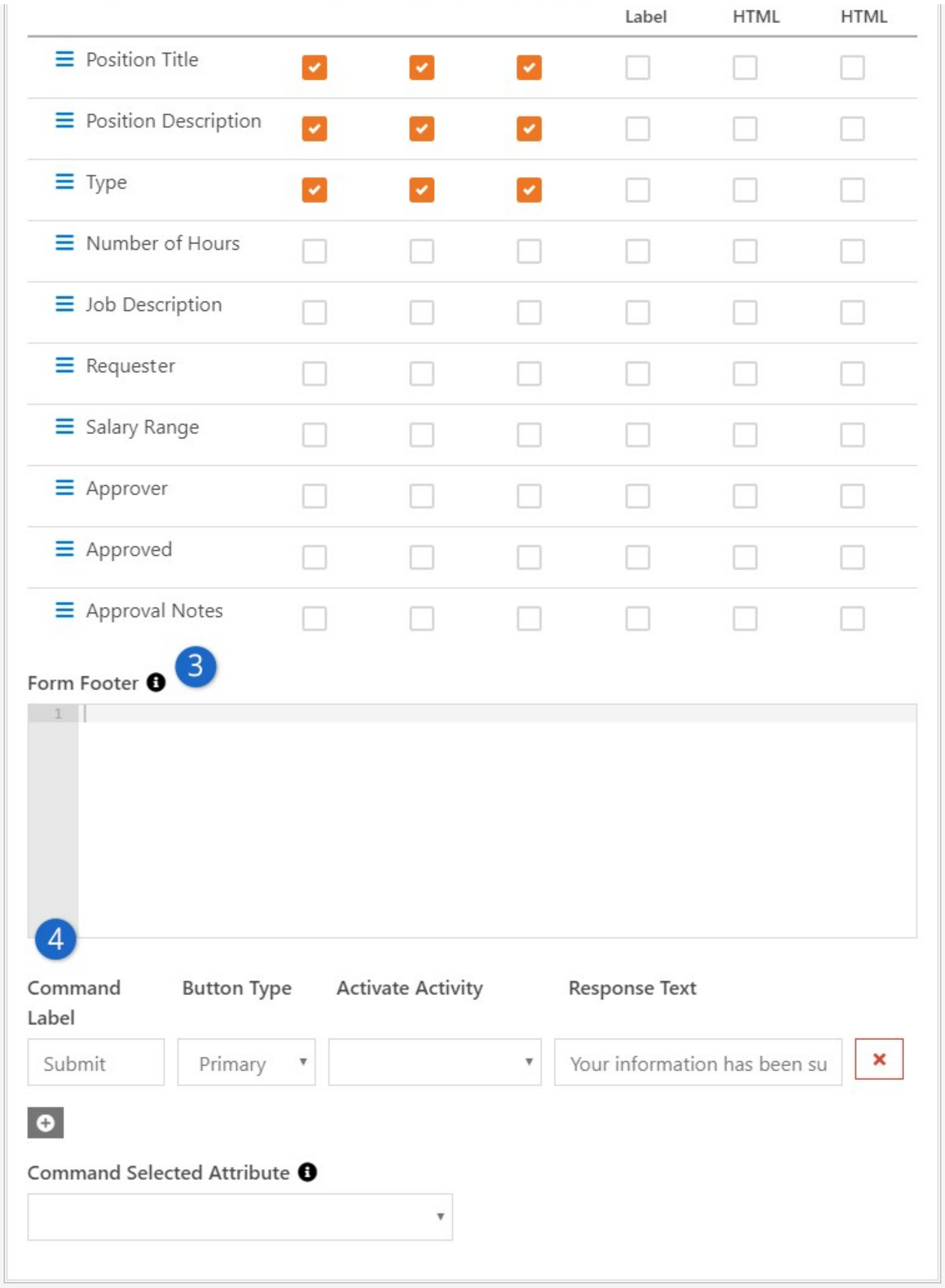
Include Actions in Email: Yes [✓]

Enable Note Entry: Yes [ ]

Form Header

```
<h1>Position Details</h1>
<p>
    {{CurrentPerson.NickName}}, please complete the entry below to start the
    approval process.
</p>
```

Form Fields



				Label	HTML	HTML
Position Title	✓	✓	✓	☐	☐	☐
Position Description	✓	✓	✓	☐	☐	☐
Type	✓	✓	✓	☐	☐	☐
Number of Hours	☐	☐	☐	☐	☐	☐
Job Description	☐	☐	☐	☐	☐	☐
Requester	☐	☐	☐	☐	☐	☐
Salary Range	☐	☐	☐	☐	☐	☐
Approver	☐	☐	☐	☐	☐	☐
Approved	☐	☐	☐	☐	☐	☐
Approval Notes	☐	☐	☐	☐	☐	☐

Form Footer 3

4

Command Label	Button Type	Activate Activity	Response Text
Submit	Primary		Your information has been su

Command Selected Attribute

1 **Purpose**
The *Form Header* is a great place to introduce the purpose of your form and any background knowledge that might be needed. As a Rock user, we know you won’t make the mistake of writing a dry and impersonalized message. Instead, we're sure you'll remember to use Lava merge fields like {{CurrentPerson.NickName}} to make your form feel personalized. That's just how you roll...

2 **Attributes**
Next, you'll pick which Workflow / Activity attributes you want your user to complete. For each, you can select whether the fields are: Visible, Editable and/or Required.

3 **Next**
The footer text is a great place to add information about what will happen next in the process.

4 **Commands**
Finally, you can add different *Commands* to the form. These commands cause the entry form to be submitted and different parts of the workflow to be activated. We'll talk more about commands next.

Entry Form Commands

Commands allow your users to execute different logic based on their selections. For instance, on an approval entry you might add two commands of *Approve* or *Deny*. Depending on which command is selected, different activities and/or actions can be run. There are two different ways commands can trigger logic. Let's consider both in detail.

Commands That Launch Activities

You can have your commands activate new activities when they are selected. You do this by selecting the activity using the *Activate Activity* property of the command. When selected, the command will activate the selected activity.

Commands That Set Attributes

Sometimes you may not want to launch a new activity based on a command. Instead, you can use actions within the same activity to process any next steps. In these cases simply leave the *Activate Activities* field empty. When empty, the next action in the current activity will be executed when the entry form is completed. You can even have the command that was executed entered into a workflow attribute using the *Command Selected Attribute*. This is helpful when multiple commands are available and you'd like to know which one was selected.

When To Launch New Activities

You might be wondering when you should launch new activities and when you should not. The choice is really up to you. But here are a couple of thoughts to help you drive your decision:

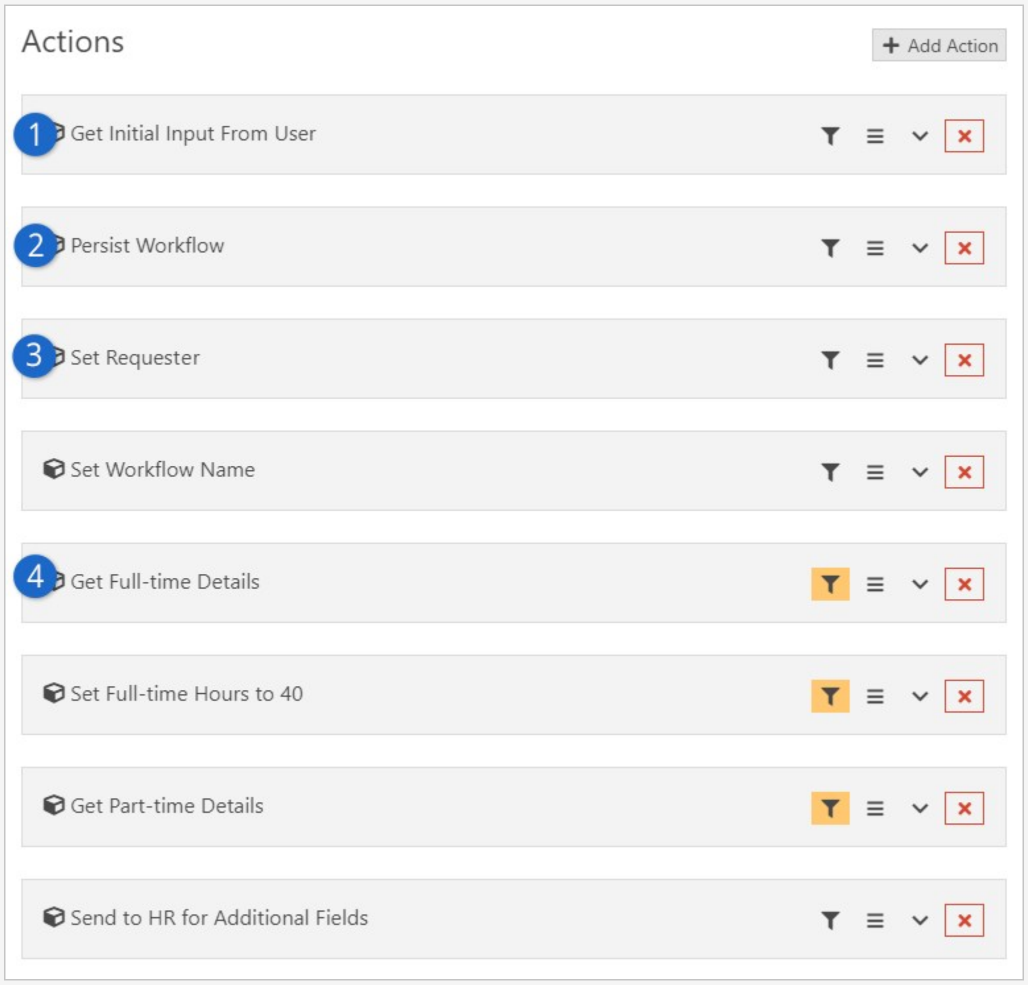
- In approval forms it's common for each option to launch a new activity. This allows the decision-making logic to be clearly separated into different activities.
- If you're using form chaining (more info below) based on user input, you may elect not to use new activities.

Entry Form Chaining

In our sample HR workflow, you'll remember that the initial entry form asked if the position was full-time or part-time. Depending on the user's selection, they were taken to a new entry form based on their input. This is a feature called entry form chaining. When the command on the first form is executed, the workflow is activated and processed. If any action in the workflow assigns a new entry form action for the current user, its form will be shown. This is a very powerful feature because it allows you to build complex interactions with the user.

Let's look at how the sample position request form was configured for chaining.

Entry Form Chaining



Get Initial Input From User

Persist Workflow

Set Requester

Set Workflow Name

Get Full-time Details

Set Full-time Hours to 40

Get Part-time Details

Send to HR for Additional Fields

1 **Entry Form**
Our initial entry form, which included the attribute of *Full-time or Part-time*.

2 **Non-presisted**
The workflow was initially configured to be non-persisted. This was to keep the workflow from being saved in the database in the case of a user who clicked on the form but then left. Now that the user has entered information and executed a command, we will persist it.

3 **Requester**
Next, we set the requester for the workflow and also provide a workflow name.

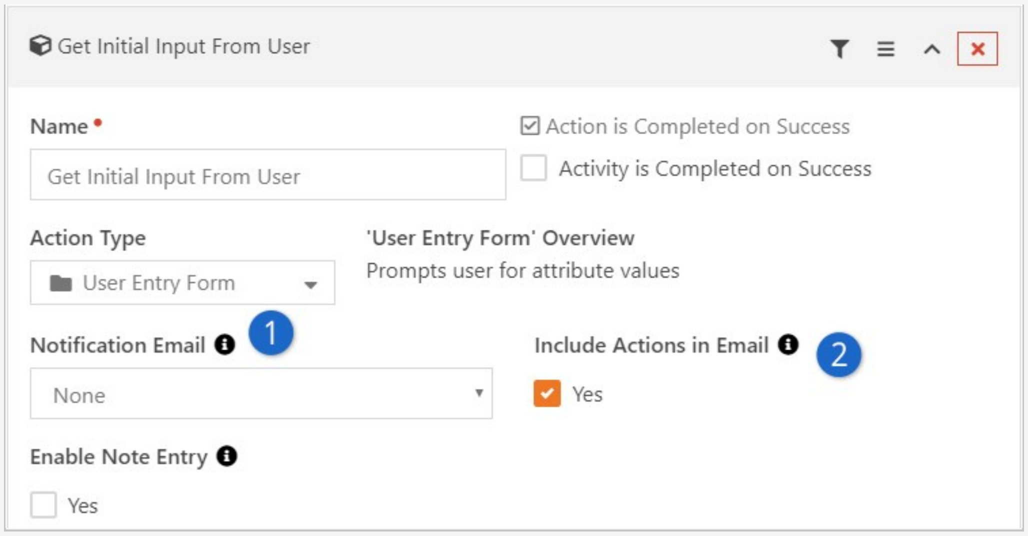
4 **Second Entry Form**
Now we're ready to show the second entry form, depending on their input. Each of these entry forms is configured with action filters that limit them to only be run if the position type is either *full-time* or *part-time*. As you can see here, using action filters with entry forms can be very powerful.

Instead of using action filters, we could have launched separate activities to get the details for each position type, one for full-time and one for part-time. In this case though, the action filters seemed a better option.

Emailing From the Entry Form

In many cases you'll want to email an individual to let them know that a workflow needs their entry before continuing. While you're welcome to configure the *Email Send* action to do this, there is a short-cut built into the *Entry Form* action.

Entry Form Email



Name • Get Initial Input From User

☑ Action is Completed on Success

☐ Activity is Completed on Success

Action Type: User Entry Form

'User Entry Form' Overview
Prompts user for attribute values

Notification Email: None

Include Actions in Email: ✓ Yes

Enable Note Entry: ☐ Yes

1 **Email**
You can select a pre-configured email template to use for the email.

2 **Commands**
You can also select whether the commands you've configured should be included in the email. This allows the recipient to execute various commands right from their email client.

Workflow Email Templates

The default communication template that ships with Rock will list the values of all of the workflow attributes selected on the entry form in the email. If this is not what you'd like, you can either create your own email using the *Email Send* action, or you can create a new template. This new template can be created under `Admin Tools > Communications > System Communications`. In order for the template to be displayed on this list, it must be added to the *Workflow* category.

> **Commands In The Default Email Template**
> As we mentioned above the default email template will list all of the attributes selected for the entry form. Also, if there are no required fields for the form, it will add buttons for each command. If an entry field is required the commands won't be shown.

See our `Communicating With Rock` guide for more information.

Buttons

Buttons on entry forms have several capabilities. Learning to extend them can help you build even more power into your workflows.

Button Types

You'll notice that Rock ships with several different button types. These provide the basic styles for the buttons. You can define new types under `Admin Tools > General Settings > Defined Types > Button HTML`. When you define a button you must provide mark-up for both a normal webpage and a HTML email.

Cancel Buttons

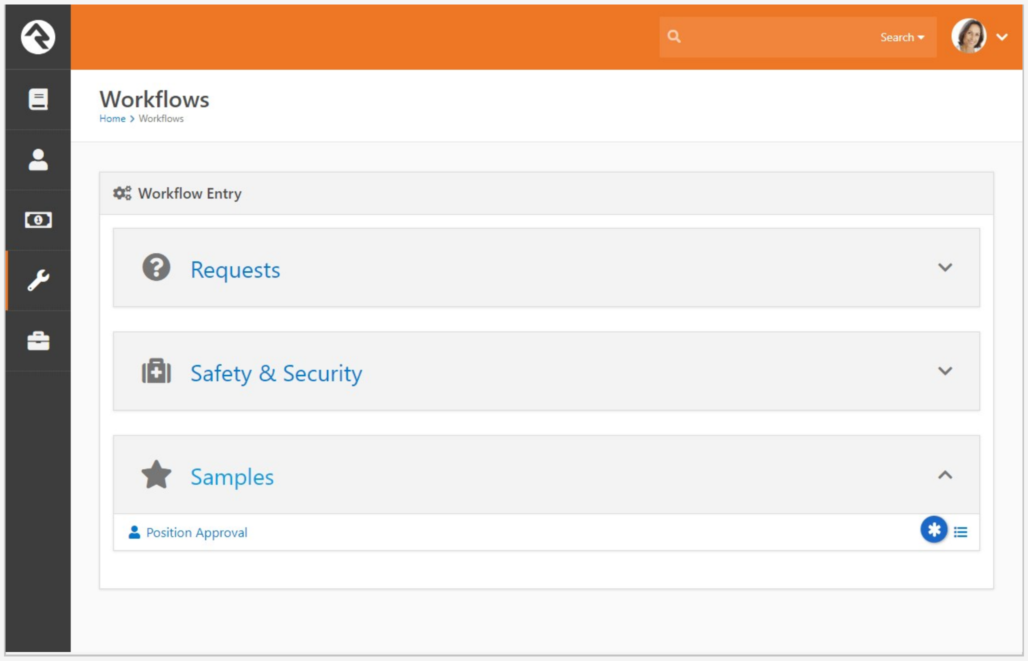
All buttons on a form will cause 'validation' to occur. This is a fancy term for checking that all the required fields are provided for. Sometimes though you want to provide a cancel button. Having to fill out all of the required fields just to cancel can be annoying. To keep the validation from occuring simply use the *Cancel* button type or change the {{ ButtonClick }} merge field to be "return true;".

Managing Workflow Instances

Not all workflows run and complete immediately. In fact, most take several hours or days to complete as they wait for input from users or other external events to occur. There are several blocks to help you manage workflows and see them to completion.

Managing Workflows

Manage Workflow Link



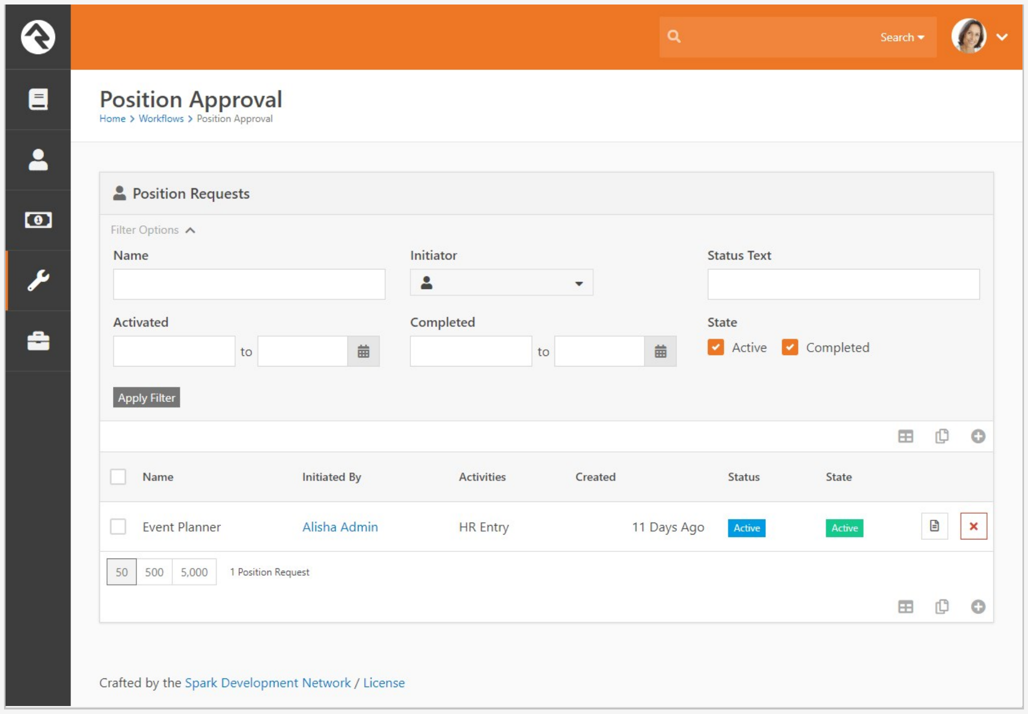
On the workflow entry screens you may have noticed a *Manage* link next to the right of the workflow name. You'll only see this link if you have *Edit* access to the workflow type.

Clicking this link will take you to a grid of all the workflows for this workflow type. This grid allows you to filter the workflows by several different criteria including:

- Workflow Name
- Initiator
- Status Text
- Activation Date (the date it was launched)
- Completion Date
- State (is it currently active or completed)

Clicking on a workflow will show you its details.

Manage Workflow


Position Approval

Home > Workflows > Position Approval

Position Requests

Filter Options

Name	Initiator	Status Text
Activated ____ to ____	Completed ____ to ____	State: ✓ Active ✓ Completed

Apply Filter

	Name	Initiated By	Activities	Created	Status	State
	Event Planner	Alisha Admin	HR Entry	11 Days Ago	Active	Active

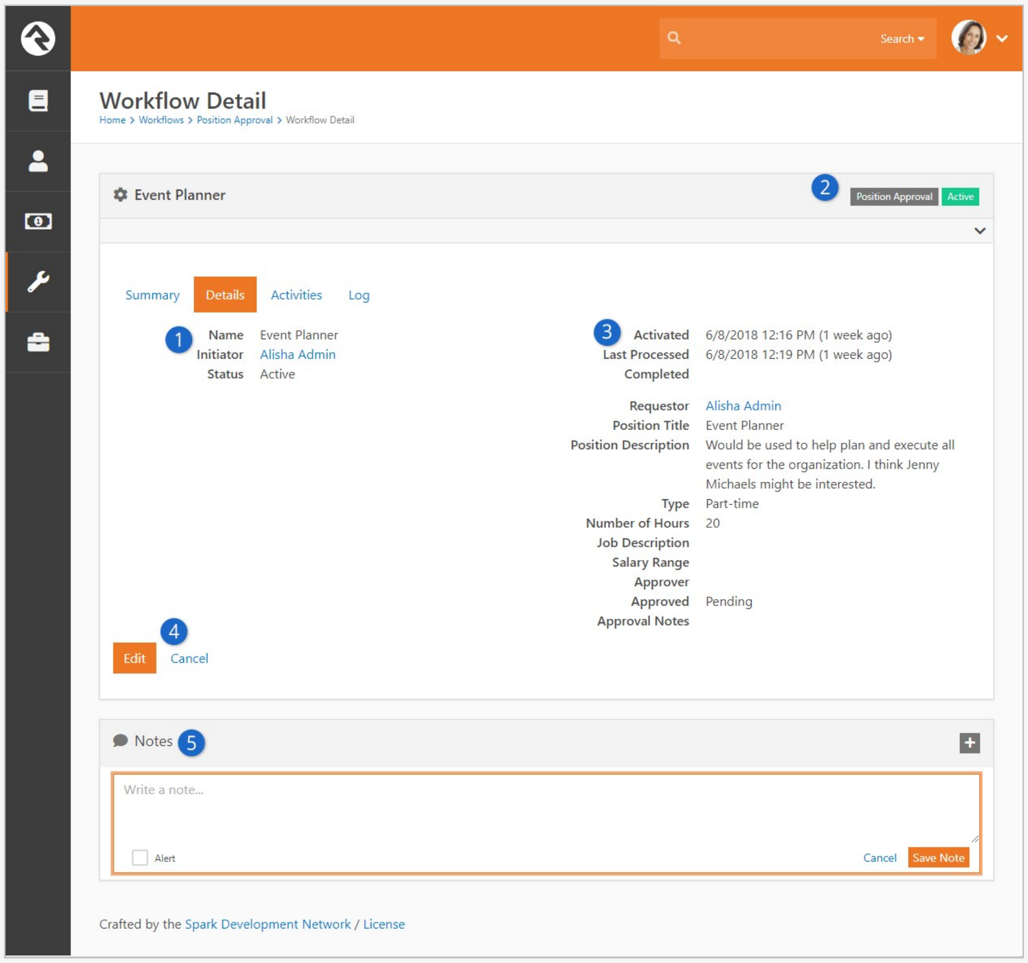
50 500 5,000 1 Position Request

Crafted by the Spark Development Network / License

Viewing Workflow Details

After clicking a workflow from the grid above you'll see the details of the workflow instance. From here you can see the state of all of the workflow attributes as well as all the activities that are in process or have completed.

Workflow Details View



Name	Event Planner
Initiator	Alisha Admin
Status	Active

Activated	6/8/2018 12:16 PM (1 week ago)
Last Processed	6/8/2018 12:19 PM (1 week ago)
Completed	
Requestor	Alisha Admin
Position Title	Event Planner
Position Description	Would be used to help plan and execute all events for the organization. I think Jenny Michaels might be interested.
Type	Part-time
Number of Hours	20
Job Description	
Salary Range	
Approver	
Approved	Pending
Approval Notes	

1 **Workflow Summary**
This area shows the name, initiator and the activation date and time. These values can be updated in the edit mode.

2 **Workflow Status**
The header also includes the workflow type and status of the current workflow.

3 **Workflow Attributes**
Next you will see a listing of all of the workflow attributes. In edit mode you can change any of these attributes.

4 **Edit Button**
The edit button allows you to change any of the properties, attributes or activities of the workflow.

5 **Notes**
See notes specific to this workflow.

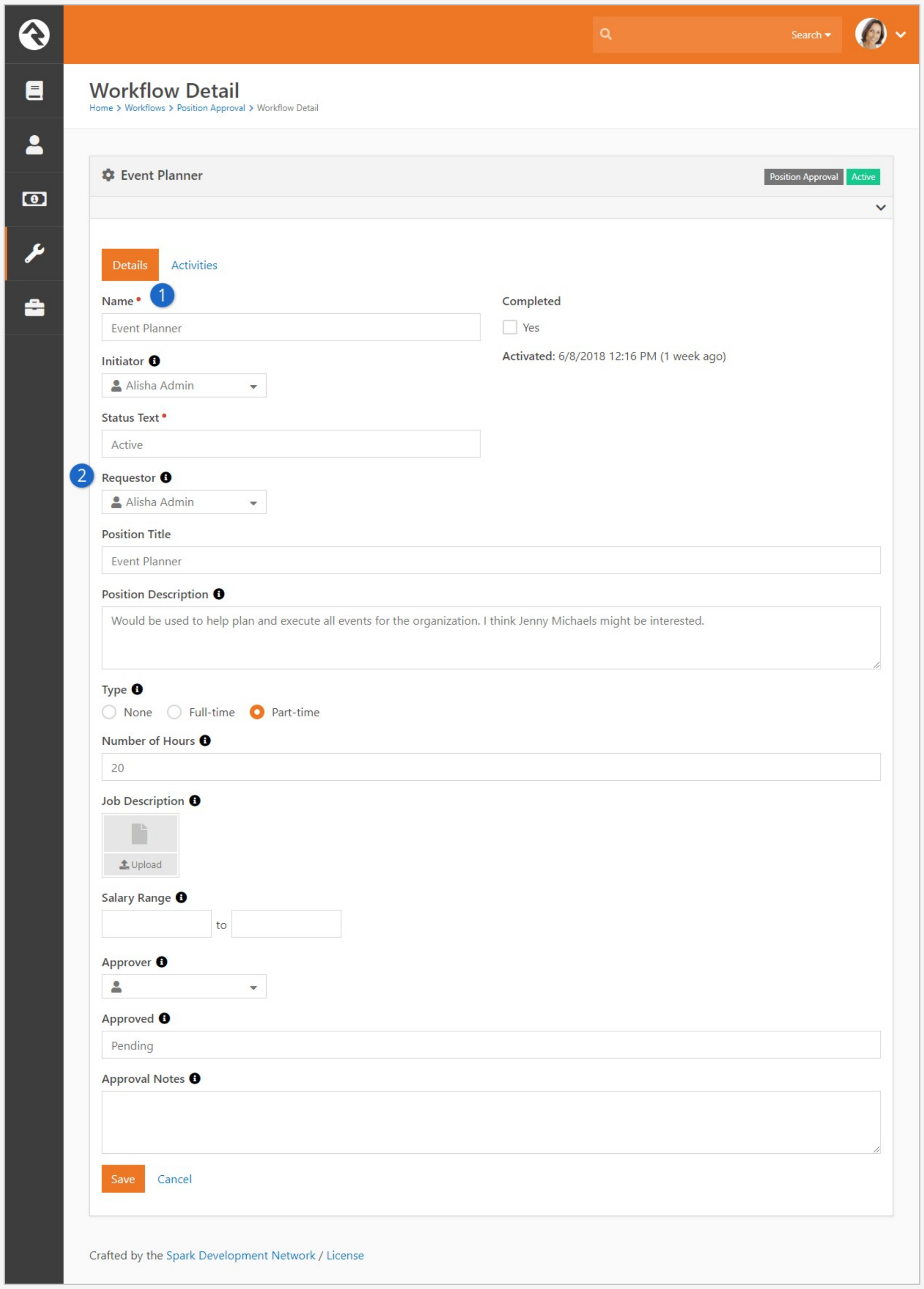
Editing Workflow Details

Viewing the workflow details page gives you a clear view about the status of a specific workflow with the ability to edit any of it's settings. Let's take a tour around this page showing each feature.

Details Tab

The details tab gives you on overview on the state of the workflow.

Workflow Details Edit


Workflow Detail
Home > Workflows > Position Approval > Workflow Detail

Event Planner
Position Approval Active

Details Activities

Name * 1
Event Planner

Completed
Yes

Activated: 6/8/2018 12:16 PM (1 week ago)

Initiator
Alisha Admin

Status Text *
Active

2 Requestor
Alisha Admin

Position Title
Event Planner

Position Description
Would be used to help plan and execute all events for the organization. I think Jenny Michaels might be interested.

Type
None Full-time [✓ Part-time]

Number of Hours
20

Job Description
Upload

Salary Range
to

Approver

Approved
Pending

Approval Notes

Save Cancel

Crafted by the Spark Development Network / License

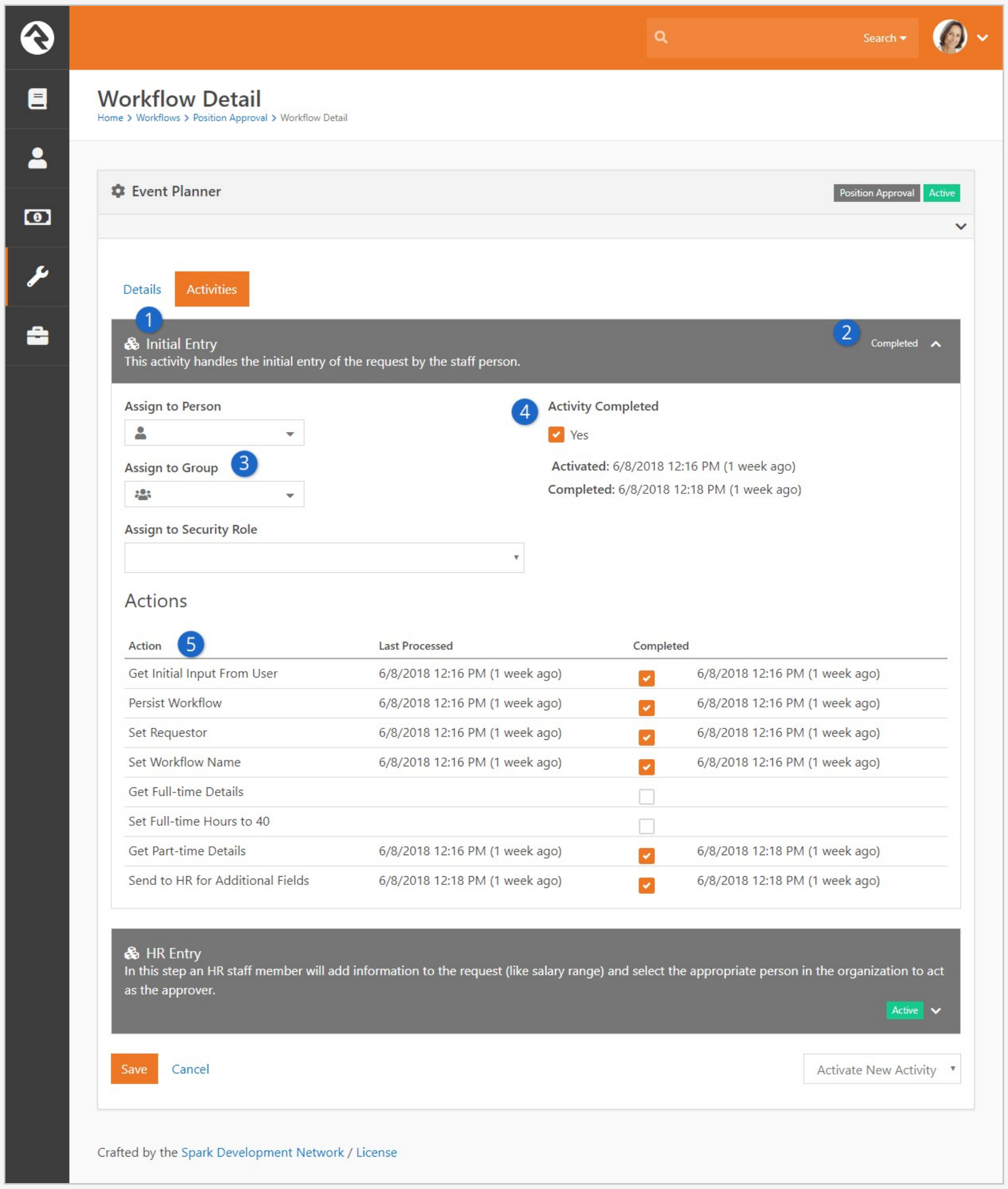
1 **Workflow Summary**
This area shows the name, initiator and the activation date and time. These values can now be updated in the edit mode.

2 **Workflow Attributes**
Next you will see a listing of all of the workflow attributes with the ability to edit them if needed.

Activities Tab

The activities tab shows a listing of each activity that was activated through the life cycle of the workflow.

Workflow Details Edit



Initial Entry
This activity handles the initial entry of the request by the staff person.

Assign to Person

Assign to Group

Assign to Security Role

Activity Completed
✓ Yes

Activated: 6/8/2018 12:16 PM (1 week ago)
Completed: 6/8/2018 12:18 PM (1 week ago)

Actions

Action	Last Processed	Completed	
Get Initial Input From User	6/8/2018 12:16 PM (1 week ago)	✓	6/8/2018 12:16 PM (1 week ago)
Persist Workflow	6/8/2018 12:16 PM (1 week ago)	✓	6/8/2018 12:16 PM (1 week ago)
Set Requestor	6/8/2018 12:16 PM (1 week ago)	✓	6/8/2018 12:16 PM (1 week ago)
Set Workflow Name	6/8/2018 12:16 PM (1 week ago)	✓	6/8/2018 12:16 PM (1 week ago)
Get Full-time Details		☐	
Set Full-time Hours to 40		☐	
Get Part-time Details	6/8/2018 12:16 PM (1 week ago)	✓	6/8/2018 12:18 PM (1 week ago)
Send to HR for Additional Fields	6/8/2018 12:18 PM (1 week ago)	✓	6/8/2018 12:18 PM (1 week ago)

HR Entry
In this step an HR staff member will add information to the request (like salary range) and select the appropriate person in the organization to act as the approver.

1. **Activity Name / Description**
2. **Completion Status**
3. **Assigned to Person / Group**
   This will show the currently assigned person or group. From this screen you can modify these values if you need to.
4. **Completion Flag**
   You can manually mark the activity as complete if needed. While you could also mark an activity back to *Uncomplete* it will most likely be marked complete again by the workflow engine the next time it runs unless you

alter the completion status of some of the activity's actions also.

5 **Action List**
Each action defined by the activity type is listed on this grid. You can see the last time the action was processed as well as its completion status and date. If you would like to re-run an action, you can choose to uncheck its completion checkbox. As long as the activity is still active, this action will be re-run. You can also mark an action as complete. This is helpful if your workflow accidently gets stuck on an action due to improper configuration.

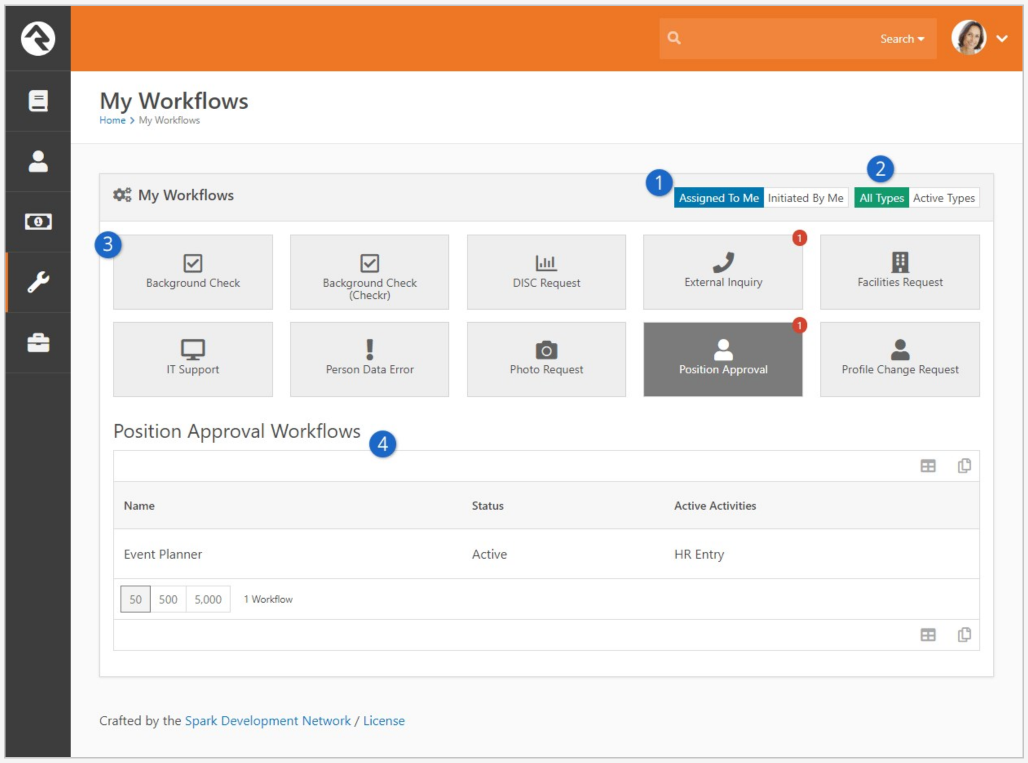
Log Tab

The log tab displays the logging messages from the workflow. The detail of the logging will be dependent on the logging level you configured for the workflow type. You can also define custom logging actions in your activities to display custom log entries. Logging is a great tool for watching your workflow to make sure that the logic you have configured is working as expected.

My Workflows

The tools described above are great for working on workflows of a specific workflow type. However, there are times you just want to see the workflows that are assigned to you or that you have initiated. You can track active workflows that are related to you under `Tools > My Workflows`.

My Workflows



Name	Status	Active Activities
Event Planner	Active	HR Entry

1 **Initiated By Me / Assigned To Me**
This toggle allows you to filter the workflows by:

- **Initiated By Me:** Those that you started. For example if you opened an IT Request it would be listed here.
- **Assigned To Me:** These are the active workflows that are assigned to you. This includes workflows that are assigned to a group that you are a member of. Using this toggle is a great way for you to track your work and what needs your attention.

2 **Active Types / All Types**
This toggle determines which workflow types will be displayed.

- **Active Types:** This Will only show workflow types that have active items related to you. Using this option helps to reduce the clutter and noise by showing only workflow types that need your attention.
- **All Types:** This Will display all workflow types you have security to view.

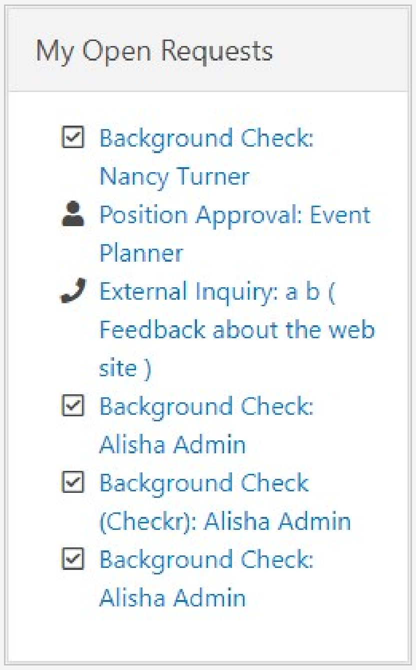
3 **Workflow Types**
Below the toggles you'll see a listing of all the workflows you have security to view. If a workflow has active requests that you're related to, a small number will appear at the top of the workflow showing a count of the related workflows.

4 **Workflow Grid**
When you click on a workflow type it will list the workflows that are related to you. Selecting one of them will take you to the workflow details screen.

Mini My Workflows

You'll notice that there is a miniature version of *My Workflows* on the homepage. This acts as a reminder to the individual of their task list as well as providing a quick link to the workflow.

My Workflow (Mini)


My Open Requests

- Background Check: Nancy Turner
- Position Approval: Event Planner
- External Inquiry: a b (Feedback about the web site)
- Background Check: Alisha Admin
- Background Check (Checkr): Alisha Admin
- Background Check: Alisha Admin

This block has several settings that allow you to customize it. These settings include:

- The ability to filter the results to a specific category(s)
- To include child categories
- To show only workflows you are assigned to
- To show only workflows you initiated
- The markup that is displayed can also be fully customized using HTML and Lava.

Using these settings, along with the power of HTML/Lava, you can reformat this block several different ways. Some organizations might even move it into the main zone on the page to give it more space.

Persisted Vs. Non-Persisted Workflows

Persisted. That might seem like a strange term if you don't have a technology background. It simply means to write something down so that it can be remembered in the future. Think of it this way - some thoughts you have are relevant only to the task you're currently working on. Sometimes, though, you have a thought that you know you'd better write down because you'll need it for a task you'll complete in the future. Writing the thought down persists the idea for use in the future.

Non-Persisted Workflows

Non-persisted workflows are executed but the attribute values are never written down (or in tech terms never written to the database). They exist only in the computer's temporary storage and go away when done. These types of workflows are great when the entire workflow will be processed immediately and will then be completed.

The check-in workflow is a non-persisted workflow. After the check-in process is complete there is no need to store all of the workflow attributes that helped the system pick the right room for the individual. Many of the entity trigger workflows may turn out to be non-persisted. These workflows will be used to make quick decisions about the nature of an update to the data. Keeping the workflow around won't provide benefits in most cases.

> Design Using Persisted Workflows:
> Even if you're certain that you want a non-persisted workflow it's wise to design the workflow as a persisted workflow. This allows you to check the logging and look at what happened under the hood as it ran. Once you're certain everything is correct you can then check the setting back to non-persisted.

Persisted Workflows

Persisted workflows write their state and status down. This allows them to run over long periods of time without forgetting their status. Most workflows that individuals interact with will be persisted (e.g. IT Requests, Facility Requests, HR processes, etc.). This allows the workflow to live for hours, days or even years.

Persisted workflows should also be used when a history of what occurred is important. These types of workflows allow you to go back and see what happened when.

Morphing Persistence

Just because a workflow starts out as a non-persisted workflow doesn't mean it has to stay that way. There is a workflow action that will change the current workflow to a persisted workflow. This is commonly used with workflow types that start with an entry form.

Picture this - a user comes to the first entry screen of a workflow and then changes their mind and closes the page. What happens? In a persisted workflow the loading of the entry screen started a new workflow that is now stored in the database. In a non-persisted workflow nothing happens. Because of this, it's common for entry workflows to start as non-persisted and then change to persisted after the first entry screen is completed.

> **We Know What You’re Thinking:**
> Ok, so you’re probably thinking: Can I turn a persisted workflow into a non-persisted workflow? The simple answer is no. Once a workflow has been set to persisted it will remain persisted. You can however, delete a workflow instance from an action inside the workflow by using the 'Delete Workflow' action.

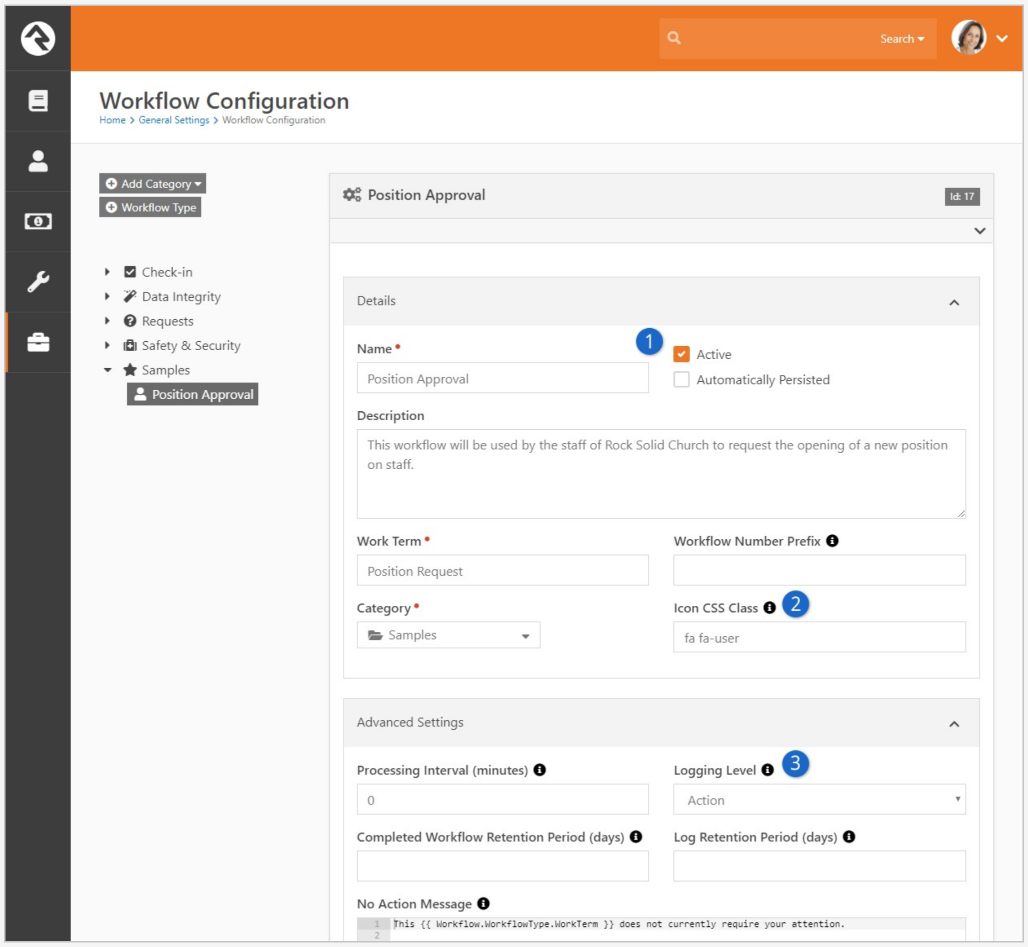
Building A Simple Workflow

Now let’s build a sample workflow from scratch. We’ll use the sample *HR Request Process* as our guide. This process has been pre-configured in your Rock install under the *Samples* category. You might review the overview of the sample workflow in the chapter What's The Use so you have a good understanding of the purpose and steps of this workflow.

Workflow Details

Workflow types are configured under `Admin Tools > General Settings > Workflow Configuration`. After adding the workflow, we'll complete the detail section.

Workflow Details


Workflow Configuration

Home > General Settings > Workflow Configuration

Position Approval

Details

1

Name *
Position Approval

Active [✓]
Automatically Persisted [ ]

Description
This workflow will be used by the staff of Rock Solid Church to request the opening of a new position on staff.

Work Term *
Position Request

Workflow Number Prefix

Category *
Samples

Icon CSS Class 2
fa fa-user

Advanced Settings

Processing Interval (minutes)
0

Logging Level 3
Action

Completed Workflow Retention Period (days)

Log Retention Period (days)

No Action Message
```
This {{ Workflow.WorkflowType.WorkTerm }} does not currently require your attention.
```

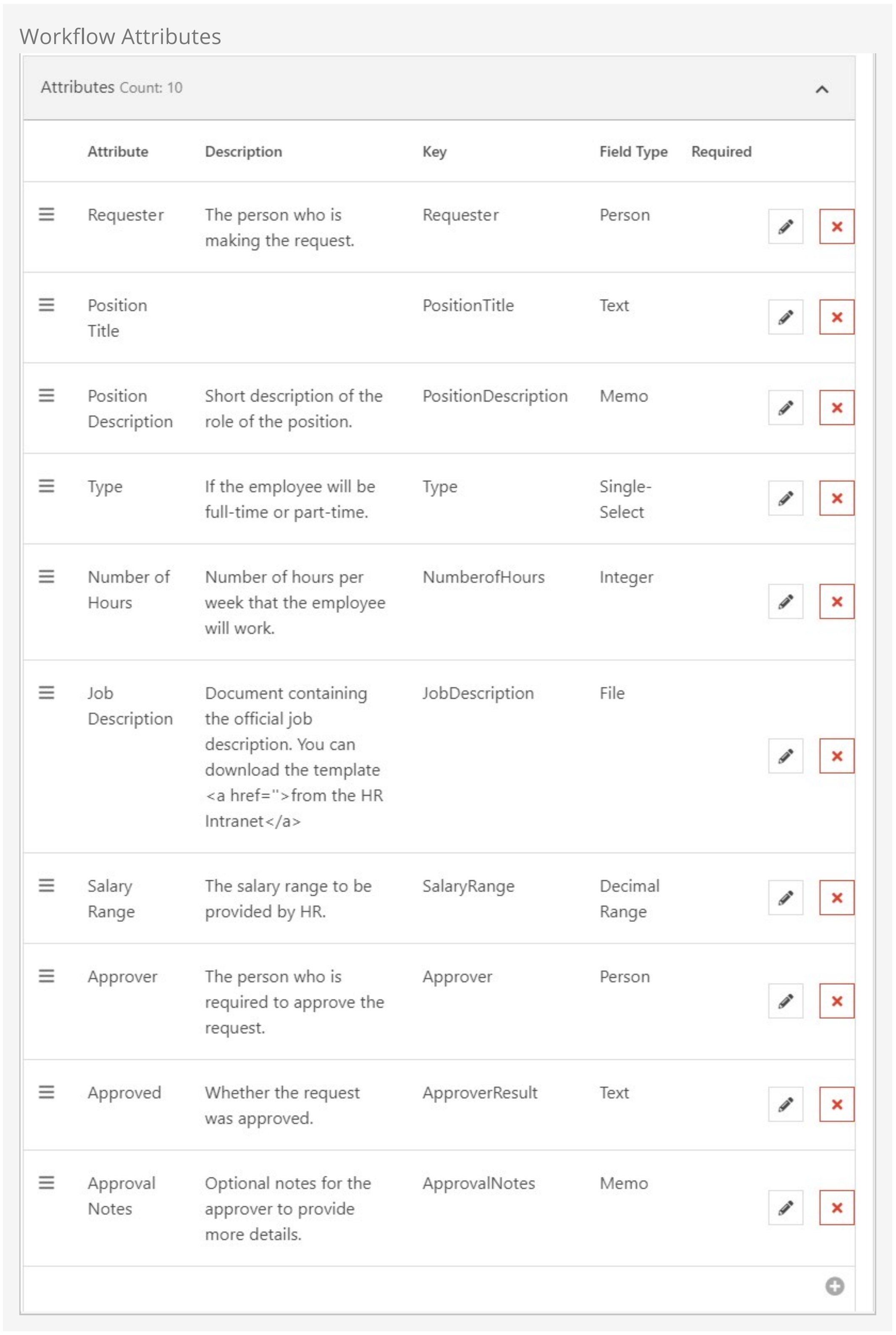
1 **Non-Presisted**
Note that this workflow is not automatically persisted. Workflows that start with an entry form are usually configured this way to keep workflows from being added to the database when the user clicks on the form but never enters anything.

2 **Icon**
Note that we have also given our workflow a custom icon so that it stands out a bit. These are simply FontAwesome CSS classes. See the FontAwesome site for a listing of all the icons.

3 **Logging**
We've set our logging level to *Action* so we can see everything that is going on under the hood.

Workflow Attributes

Next we will define all the attributes for our workflow. The attributes for the position request workflow are below.


Workflow Attributes

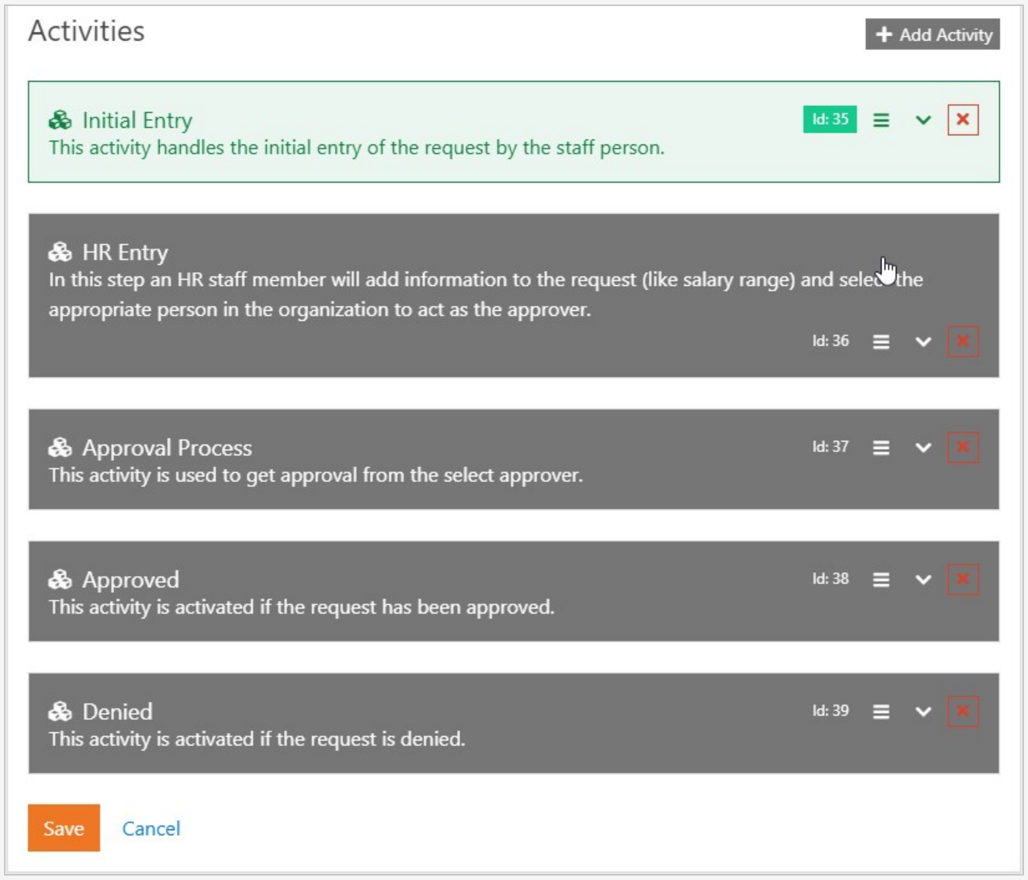
Attributes Count: 10

Attribute	Description	Key	Field Type	Required
Requester	The person who is making the request.	Requester	Person	
Position Title		PositionTitle	Text	
Position Description	Short description of the role of the position.	PositionDescription	Memo	
Type	If the employee will be full-time or part-time.	Type	Single-Select	
Number of Hours	Number of hours per week that the employee will work.	NumberofHours	Integer	
Job Description	Document containing the official job description. You can download the template <a href=">from the HR Intranet</a>	JobDescription	File	
Salary Range	The salary range to be provided by HR.	SalaryRange	Decimal Range	
Approver	The person who is required to approve the request.	Approver	Person	
Approved	Whether the request was approved.	ApproverResult	Text	
Approval Notes	Optional notes for the approver to provide more details.	ApprovalNotes	Memo	

Workflow Activities

Finally, we'll define the activities and actions for our workflow. The activities for this workflow are shown below. Note how the *Initial Entry* activity is in green. This means that it's configured to activate when the workflow is initially activated.

Workflow Activities



Activities

Initial Entry
This activity handles the initial entry of the request by the staff person.
Id: 35

HR Entry
In this step an HR staff member will add information to the request (like salary range) and select the appropriate person in the organization to act as the approver.
Id: 36

Approval Process
This activity is used to get approval from the select approver.
Id: 37

Approved
This activity is activated if the request has been approved.
Id: 38

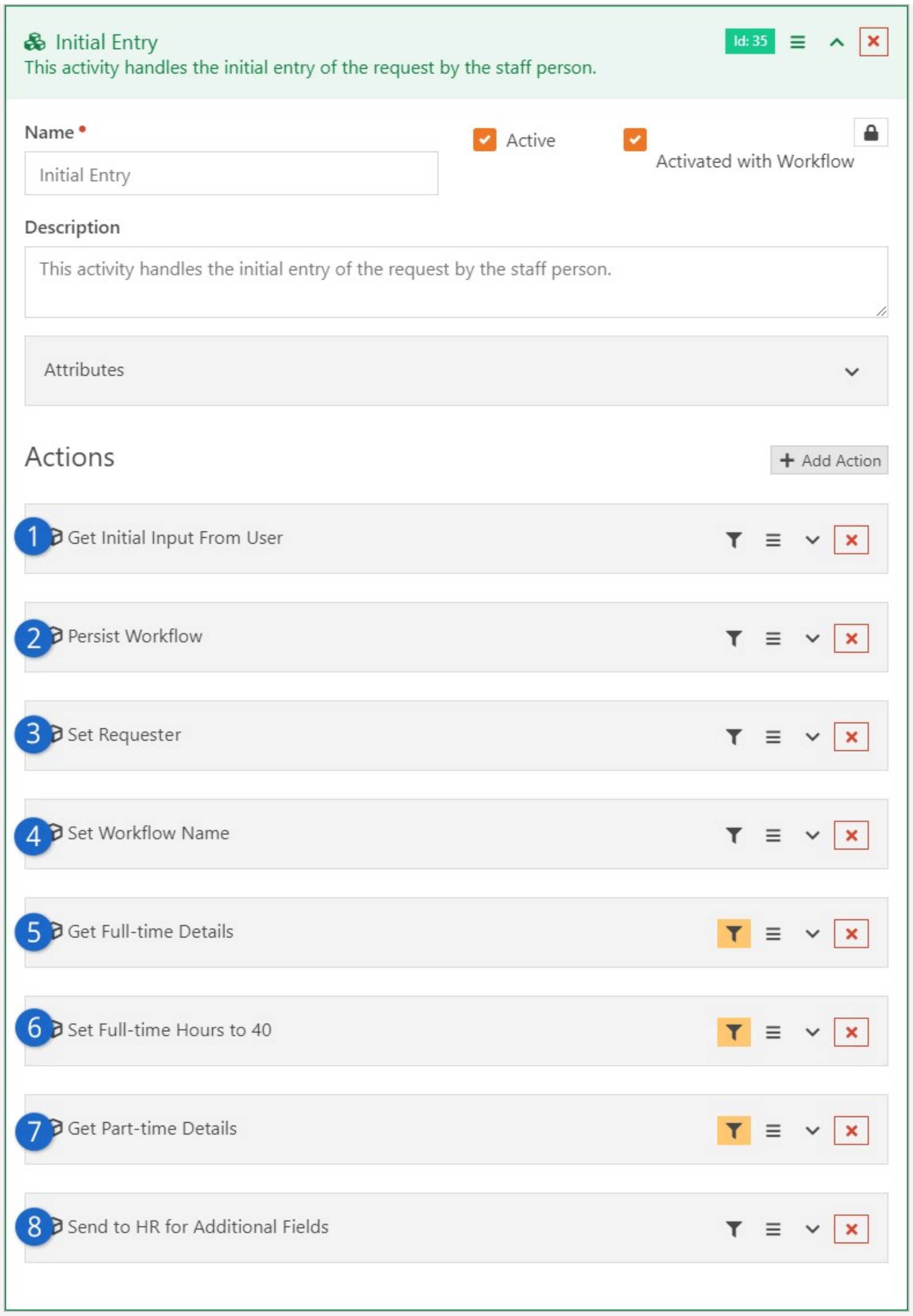
Denied
This activity is activated if the request is denied.
Id: 39

Let's walk through each activity now to look at its actions. We won't look at each setting of each action, but we'll give you an idea of the purpose of each.

Initial Entry

The purpose of this activity is to get the user's position request details. Some of these details will depend on whether they are asking for a full-time or part-time request.

Initial Entry Activity


Initial Entry

This activity handles the initial entry of the request by the staff person.

Name: Initial Entry

Active ✓

Activated with Workflow ✓

Description: This activity handles the initial entry of the request by the staff person.

1. Get Initial Input From User
2. Persist Workflow
3. Set Requester
4. Set Workflow Name
5. Get Full-time Details
6. Set Full-time Hours to 40
7. Get Part-time Details
8. Send to HR for Additional Fields

1 **Entry Form**
Initial entry form that asks for details about the position. One of these fields, *Type*, will allow the user to note if the position is full-time or part-time.

2 **Persist**
Next we persist the workflow, now that the user has entered information, so that we can maintain the values over time.

3 **Requester**
We can now set the requester for the workflow with the current person

who just entered the form.

4 **Name**
Another setup step is to provide the workflow with a name. This name will be used on the various grid displays.

5 **More Information**
Now we'll start asking for more information. This next action will get further details if the user requested a full-time position. Note that an action filter limits this entry form to only be displayed when the user selected *full-time* as the position type.

6 **Full Time**
The next step sets the position hours to 40 if the use selected full-time, using action filters.

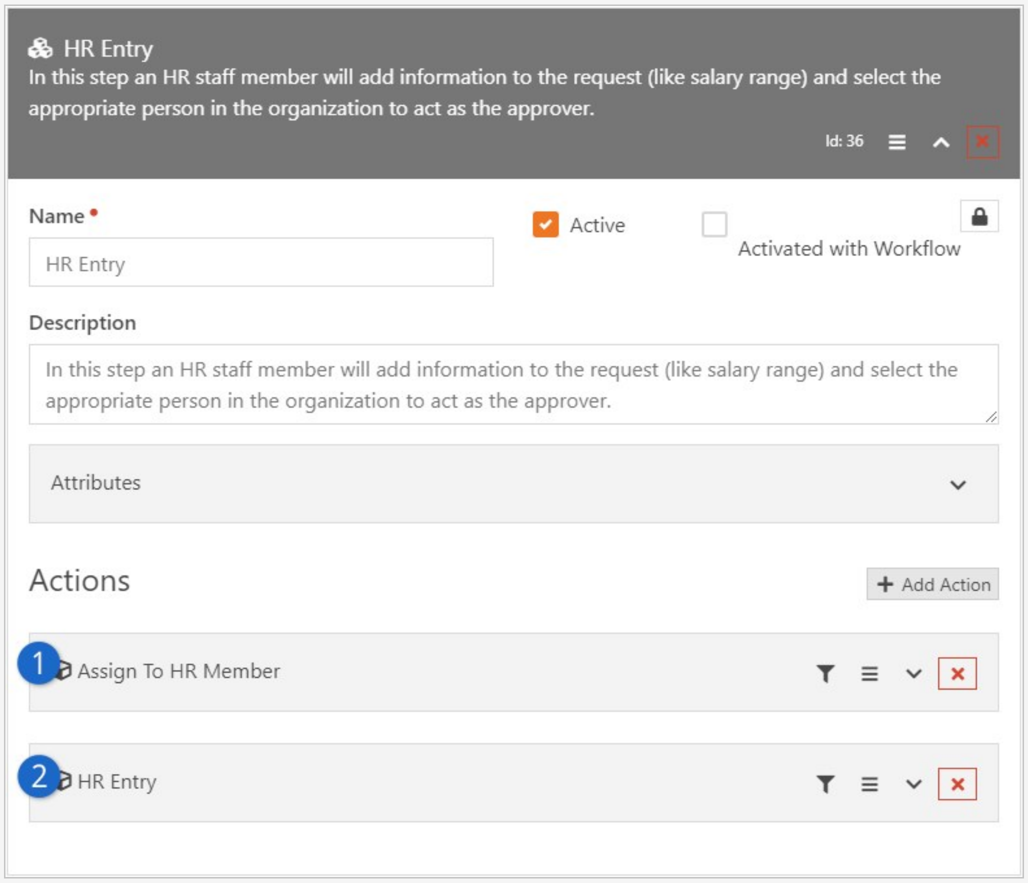
7 **Part Time**
This action collects information for part-time positions. It also uses filter actions to only display when appropriate.

8 **HR Entry**
Our final action activates the *HR Entry* action to handle the next step. This action is set to mark the activity as complete when it runs.

HR Entry

This activity is responsible for getting additional details from the human resources department. It has the following actions.

HR Entry Activity


HR Entry
In this step an HR staff member will add information to the request (like salary range) and select the appropriate person in the organization to act as the approver.
Id: 36

Name: HR Entry
Active [✓]
Activated with Workflow [ ]

Description: In this step an HR staff member will add information to the request (like salary range) and select the appropriate person in the organization to act as the approver.

Attributes

Actions

1. Assign To HR Member
2. HR Entry

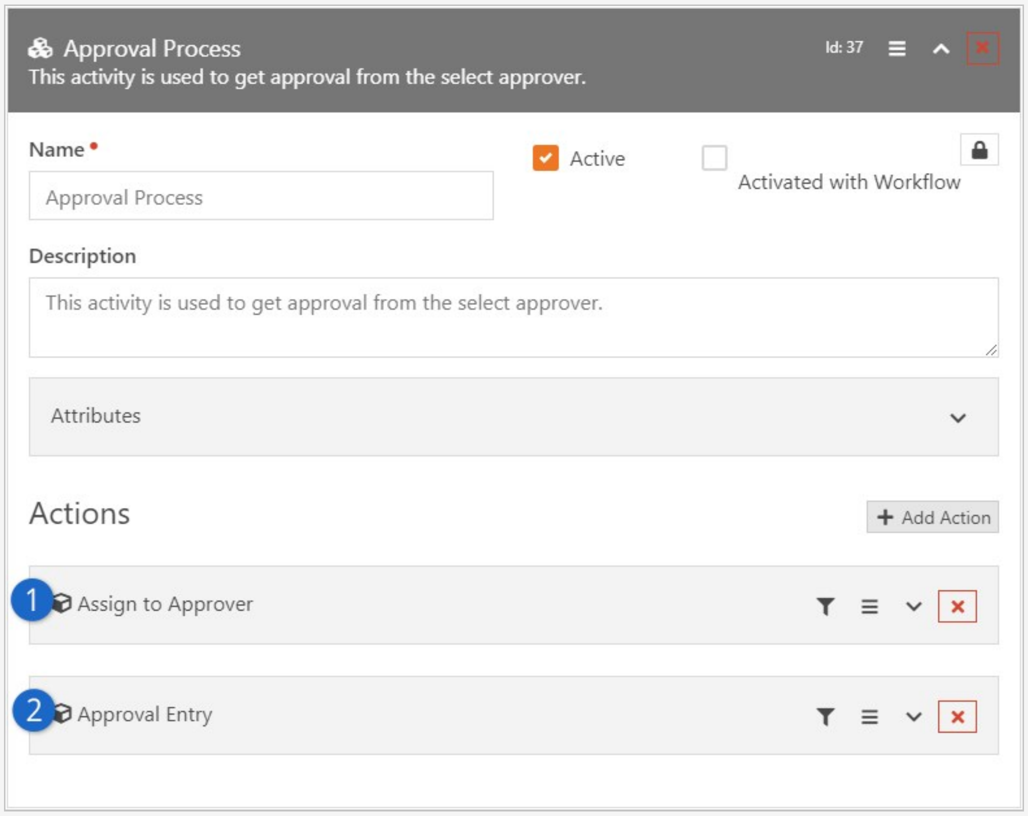
1. **Assign**
The first action assigns the activity to Alisha Marble, the human resources representative for the organization. This tells the workflow who is responsible for entering the human resources information.

2. **HR Entry**
The next and final action is the human resources entry screen. This entry form is set to email the assigned person when the action is active. This saves us from having to add our own email action. The command on this entry form is set to activate the next step, which is the approval process.

Approval Process

This activity is responsible for getting the approval for the position. Achieving this might be easier than you thought. Consider these actions.

Approval Process Activity


Approval Process
This activity is used to get approval from the select approver.

Name: Approval Process
Active [✓]
Activated with Workflow [ ]

Description: This activity is used to get approval from the select approver.

Attributes

Actions

1 Assign to Approver

2 Approval Entry

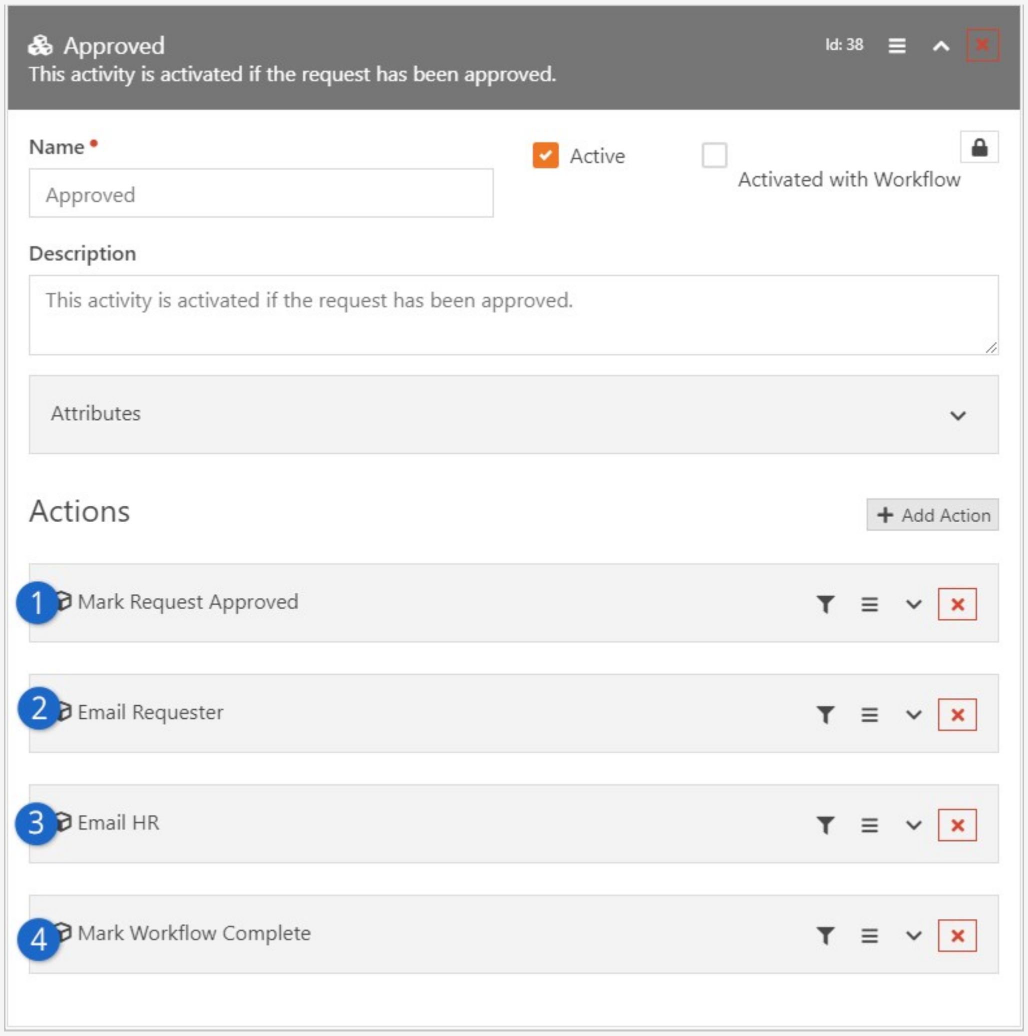
1 **Approver**
Just like the HR Entry, we assign this activity to the approver who was entered by human resources.

2 **Approval Information**
Then we add an entry to get the approval information. This form has two commands: one for Approve and the other for Deny. Each of these commands launches an activity to process actions for each response. This entry form also has to be configured to send an email. This email has the *Include Actions In Email* enabled, which allows the individual to approve or deny the request right from the email.

Approved / Denied Activities

These activities are configured pretty much the same way so we'll look at them together.

Approved / Denied Activities


Approved
This activity is activated if the request has been approved.

Name: Approved
Active: ✓
Activated with Workflow: ☐
Description: This activity is activated if the request has been approved.

1 **Approve or Deny**
The first action marks the request as either approved or denied.

2 **Response to Requester**
Next, we email the requester with the approval response.

3 **Response to HR**
Then, we send an email to the human resources representative so that they are aware of the approval state.

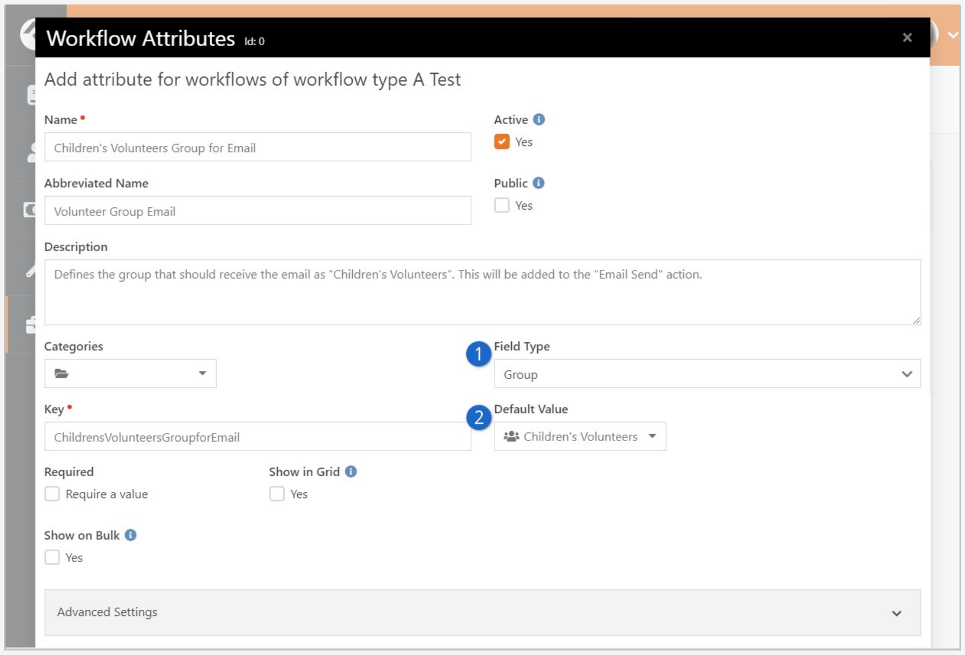
4 **Complete**
Finally, we mark the workflow as complete.

Workflow Tips

Below are some tips to keep in mind when creating new workflows.

- There are different ways to use workflows to send emails to groups. One approach is to set up an attribute that defines the group, then add an Email Send action type pointed to the attribute. Below is an example of what that would look like.

Add Workflow Attribute



Workflow Attributes Id: 0

Add attribute for workflows of workflow type A Test

Name: Children's Volunteers Group for Email

Active: [✓ Yes]

Abbreviated Name: Volunteer Group Email

Public: Yes

Description: Defines the group that should receive the email as "Children's Volunteers". This will be added to the "Email Send" action.

Categories

1 Field Type: Group

Key: ChildrensVolunteersGroupforEmail

2 Default Value: Children's Volunteers

Required: Require a value

Show in Grid: Yes

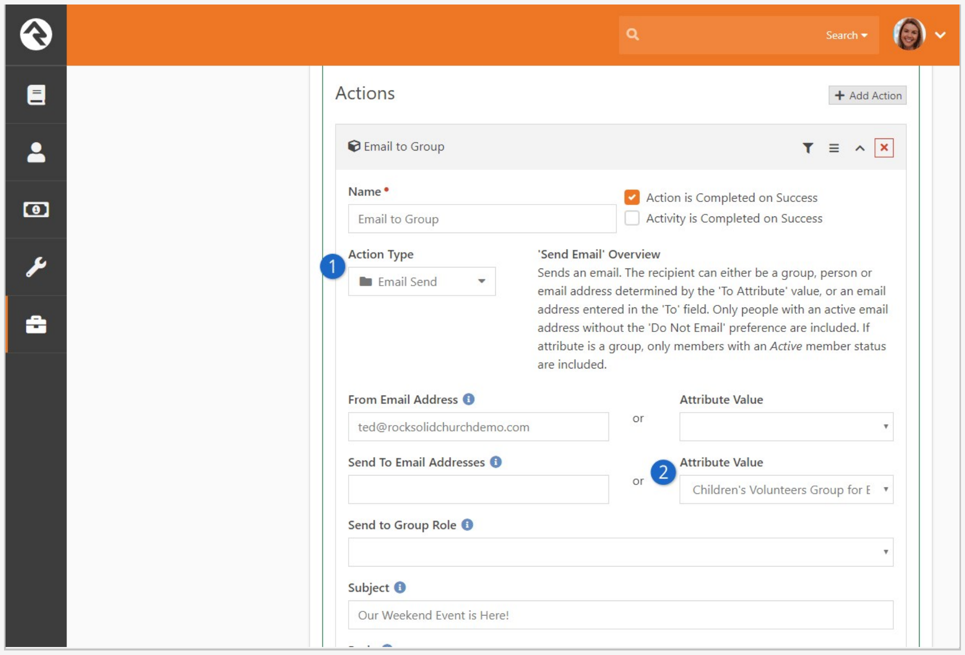
Show on Bulk: Yes

Advanced Settings

1 **Field Type**
Be sure the Field Type is set to "Group" so you can select the group you need.

2 **Default Value**
Use the drop-down to select the group that should receive the email.

Configure Email Send Action


Actions

Email to Group

Name * Email to Group

☑ Action is Completed on Success
☐ Activity is Completed on Success

Action Type: Email Send

'Send Email' Overview
Sends an email. The recipient can either be a group, person or email address determined by the 'To Attribute' value, or an email address entered in the 'To' field. Only people with an active email address without the 'Do Not Email' preference are included. If attribute is a group, only members with an *Active* member status are included.

From Email Address: ted@rocksolidchurchdemo.com

Send To Email Addresses — or — Attribute Value: Children's Volunteers Group for E

Send to Group Role

Subject: Our Weekend Event is Here!

1 **Action Type**
Select "Email Send" to have the workflow send an email.

2 **Attribute Value (Send To Email Address)**
The attribute that was configured above can now be selected from the drop-down menu. This will send an email to the "Children's Volunteers" group, according to the workflow attribute setup.

- While not required in all cases, it's wise to have your last action set the activity as complete when it's done. If you don't set this, all of the actions will need to be complete before the activity is marked complete. And if you're using action filters, this might not always occur.
- Don't forget that you can activate a workflow from a workflow. If there's something small that isn't working quite right in your big workflow, consider creating a small helper workflow to get the job done.
- There are many resources to help you build the workflow you need. Aside from the `Workflow Actions Documentation`, other guides may give insight into building the right workflow the right way. For example, our `Engagement` manual has an entire chapter on `Connection Workflows`.

Built-In Workflows

Rock ships with a several workflows to help you get started. Hopefully you'll be able to use them in your organization, but if not, they'll act as patterns when you start to write your own. We cover the details of each workflow below.

> Remember You Can Copy:
> Don't forget that you can copy a current workflow to create a clone of it. This clone is a great place to start if your workflow is similar.

Check-in

You'll notice the check-in workflow on the *Workflow Configuration* screen. This is the workflow that runs Rock's check-in system. Unless you know exactly what you're doing, we recommend that you do not alter this workflow.

Person Data Error

The *Person Data Error* workflow is an example of a workflow that is launched from the action menu on the *Person Details* screen. It allows your organization's staff to note any issues with the data that they might not know how to fix. This is a great pattern to use when you go to create your own person profile actions.

External Inquiry

This workflow is used on the *Contact Us* page of the external website. It's meant to help direct your guest's questions to the correct staff. It also helps to provide accountability that your guest's questions will be answered in a prompt timeframe.

Facilities /IT Request

These two workflows act as a request ticketing system for your facilities and information technology teams. Using them allows your staff to report issues or request new services.

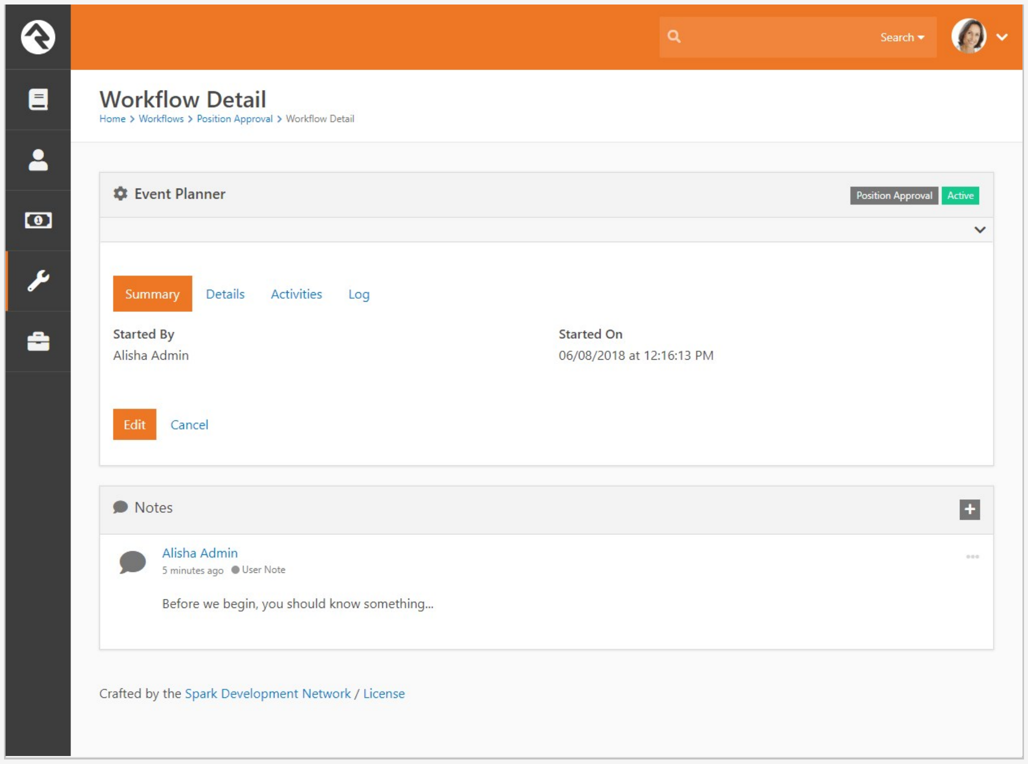
Workflow Notes

Workflows are a powerful tool to automate your organization's processes. When you're building them you'll want to pay close attention to how individuals interact with them. You especially want people to be able to interpret the current state of the workflow and a history of events that have occurred. To help with that we've created workflow notes. Let's look at a few ways that notes can be integrated into your workflows.

Adding Notes From the Workflow Details

When looking at the details of a workflow you'll notice that you have the option to add notes. This is a great way to be able to provide context or additional information at any time (even after a workflow has been completed).

Adding Notes From the Workflow Details


Search
Workflow Detail
Home > Workflows > Position Approval > Workflow Detail
Event Planner
Position Approval
Active
Summary
Details
Activities
Log
Edit
Cancel
Notes
Crafted by the Spark Development Network / License

Started By
Alisha Admin

Started On
06/08/2018 at 12:16:13 PM

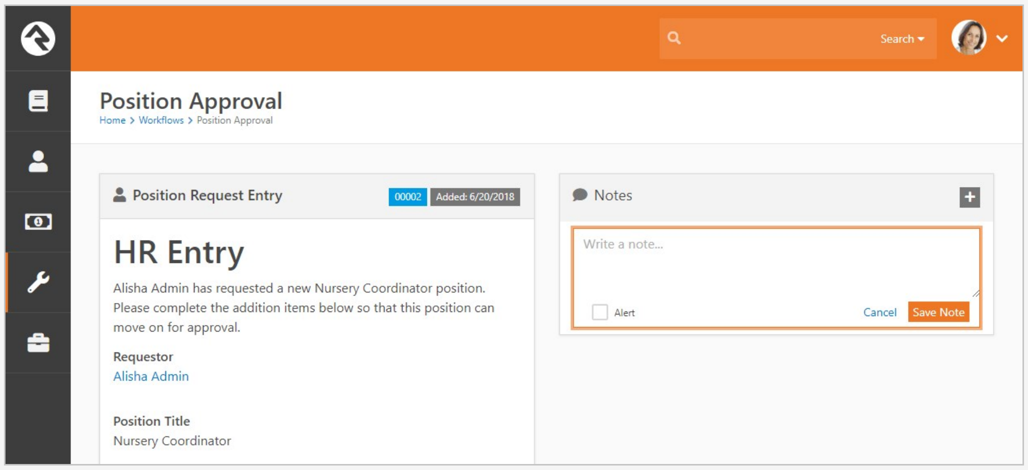
Alisha Admin
5 minutes ago ● User Note

Before we begin, you should know something...

Adding Notes From Entry Forms

Another way to add notes is from entry forms. You'll notice on the *Form* action there is a setting to *Enable Note Entry*. When it's set you'll see the note entry form next to the selected form fields.

Adding Notes From Entry Forms



HR Entry

Alisha Admin has requested a new Nursery Coordinator position. Please complete the addition items below so that this position can move on for approval.

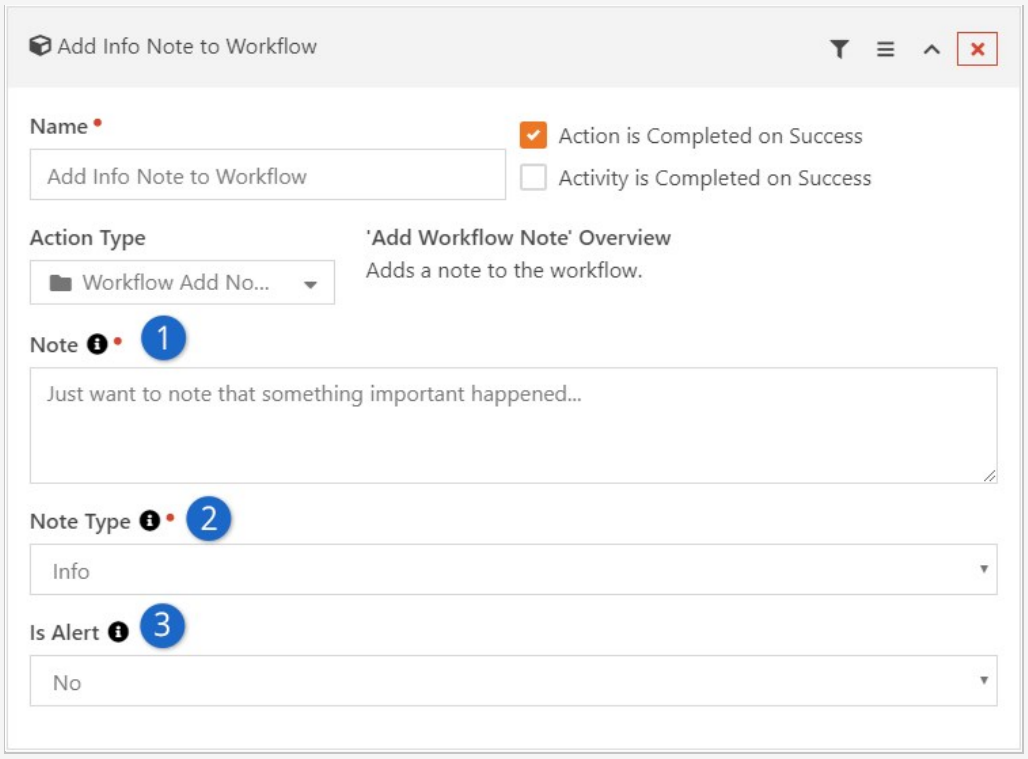
Requestor
Alisha Admin

Position Title
Nursery Coordinator

Adding Notes From Workflow Actions

The final way to add workflow notes is using workflow actions. This allows you, the workflow author, to automatically create notes about the workflow processes.

Adding Notes From Workflow Actions


Add Info Note to Workflow

Name •
Add Info Note to Workflow

✓ Action is Completed on Success
Activity is Completed on Success

Action Type
Workflow Add No...

'Add Workflow Note' Overview
Adds a note to the workflow.

Note • 1
Just want to note that something important happened...

Note Type • 2
Info

Is Alert 3
No

1 **Note**
This is the text of the note. Be sure to use Lava for extra power.

2 **Note Type**
When you add notes using actions, you can style the note in several different ways with *Note Types*. These note types are defined under `Admin Tools > General Settings > Defined Types > Workflow Note Types`.

3 **Is Alert**
Setting this property will cause the note to always be at the top and highlighted with the alert styling.

Adding New Note Types

Feel free to add new note types by adding more *Defined Values*. When you add new types, you can customize the note's icon. The name of the note type will also be placed as a CSS class on the note markup to allow you to style it. For example, the note type *System Note* would have the CSS class *system-note* applied.

Lava Tips For Workflows

Lava is based on Liquid, a simple templating language developed by the folks at Shopify. Rock use Lava everywhere so hopefully you're already familiar with it. If not, you can find more information about it at Lava.

Lava lets you supercharge your workflows, creating powerful and personalized experiences. The goal of this chapter is to open up the toolbox and show you what you can create. You'll definitely want to keep this cheat sheet handy as your make your workflows.

Action fields that support Lava will have a {{ Lava }} notation in their help section. When you see this, you'll have access to any of the data below.

Attributes

Attributes are the most commonly accessed merge fields for workflows. Let's dig in to see how we can add strength to our workflows using the power of Lava and workflow attributes.

Any attribute value can be referenced using the notation `{{ Workflow | Attribute:'AttributeKey' }}`. By default, the attribute key is the name with all of the spaces removed, although you can provide your attribute with a different key if you wish. So, if you want to display the value stored in the Attribute *Position Title*, you'd use `{{ Workflow | Attribute:'PositionTitle' }}`.

Referencing an attribute will always return the formatted value of the attribute. For example, if the *Requester* attribute was a Person type attribute, the `{{ Workflow | Attribute:'Requester' }}` merge field would return the full name of the person.

Workflow Attributes: Accessing *Their* Properties and Attributes

One thing to keep in mind when you use attributes is that they will always attempt to display the formatted value of the object you've identified. Think of this formatted value as a "friendly value". That works great when you've got a person attribute in your workflow and want to show their name. But if you want to show something other than the person's name, you need to give Lava a bit more information about what you're looking for.

For instance, since *Email* is a person property, you can use a second parameter on the `| Attribute` filter to tell Lava you want to show their email address. Your Lava would look like this: `{{ Workflow | Attribute:'Requester','Email' }}`.

If you need an *attribute* of the person instead (for instance, their Baptism Date), start by loading the full person object from your Requester attribute, and then use a second Attribute filter in your Lava. For example: `{{ Workflow | Attribute:'Requester','Object' | Attribute:'BaptismDate' }}`

Unformatted Attribute Values

There may be a time when you'd like to retrieve the raw value for an attribute. This would be helpful in creating links to pages that would need to know which person, group, etc. you are interested in. You can retrieve the unformatted attribute value by appending a *RawValue* as a second parameter in the attribute filter. For example, using Lava of `{{ Workflow | Attribute:'Requester','RawValue' }}` would return a GUID, since that is how the Person type attribute stores its value under the hood.

> **Keep In Mind**
> The Person attribute stores the GUID of the PersonAlias record, not the GUID of the Person record.

Link Values

Some attribute types are also considered *linkable* by Rock (currently this only includes the *Person* and *File* attribute types). For these attribute types, you can also append a *Url* to the attribute key to get a url to the detail page in Rock used to view that type. For example a mergefield of `{{ Workflow | Attribute:'Requester','Url' }}` would return the url value to the person profile page for the selected person.

Attribute Security on Triggered Workflows

When Workflow actions access attribute values (for a Person, Group, etc.) via Lava, Rock performs authorization checking. When they are executed by a trigger or Job, Rock requires *All Users - Allow View* to read the values because there is no user (or CurrentPerson) to base security access checking against.

However, you can override the CurrentPerson to a person who has authorization (such as a Rock Administrator) inside your Lava by assigning the CurrentPerson like this:

`{% assign CurrentPerson = Workflow | Attribute:'[AttributeOfAdminPerson]','Object' %}`

Here's a detailed example of a *Set Attribute Value* action setting a Workflow's Boolean attribute based on the age of a person's (Requester) BackgroundCheckDate. In this example an attribute called 'Admin' of type *Person* was added to the workflow with the default person set to the Rock Administrator:

```
{% assign CurrentPerson = Workflow | Attribute:'Admin','Object' %}
{% assign years = Workflow | Attribute:'Requester','Object' | Attribute:'BackgroundCheckDat
e','Object' | DateDiff:'Now', 'Y' %}
{% if years and years < 2 %}True{% else %}False{% endif %}
```

Warning

Make sure you don't have any extra spaces in your Lava output otherwise you will get unexpected results.

Checking Boolean Attributes

When checking boolean attributes you'll need to compare the value with the *True Text* or *False Text* used when the attribute was defined. Unless you've changed it, by default a boolean attribute uses the text "Yes" and "No".

Here's an example of a *Set Attribute Value* action checking several Activity attributes in order to set a Workflow boolean attribute:

```
{% assign isStep1Ready = Activity | Attribute:'Step1Ready' %}
{% assign isStep2Ready = Activity | Attribute:'Step2Ready' %}
{% if isStep1Ready == 'Yes' and isStep2Ready == 'Yes' %}True{% else %}False{% endif %}
```

Workflow

The workflow item represents the current workflow instance. The following merge fields are available:

```
Workflow = {
    "Id":,
    "Guid":,
    "Name":,
    ""Description":
    "Status":
    "WorkflowType = {
        "Id":,
        "Guid":
        "Name":,
        "Description":
        "Order":
        "WorkTerm":
        "Category": = {
            "Name":
            "Description":
        }
    }
    "Activities":[]
    "ActivityTypes": []
}
```

Examples:

- `{{ Workflow.Name }}`
- `{{ Workflow.WorkflowType.Name }}`

Current Activity

The activity merge object has the following fields available.

```
Activity = {
    "Id":,
    "Guid":,
    "AssignedPersonAliasId":,
    "AssignedGroupId":,
    "ActivatedDateTime":
    "WorkflowId":,
    "Workflow":,
    "ActivityType = {
        "Id":,
        "Guid":
        "Name":,
        "Description":
        "Order":
        "ActionTypes": []
    }
}
```

Examples:

- `{{ Activity.Id }}`
- `{{ Activity.ActivityType.Name }}`

Current Action

The action has the following merge fields available.

```
Action = {
    "Id":,
    "Guid":,
    "ActivityTypeId":,
    "ActivityType":,
    "ActionType = {
        "Id":,
        "Guid":
        "Name":,
        "Order":
    }
}
```

Examples:

- `{{Action.Id}}`
- `{{Action.ActionType.Name}}`

Global Attributes

Rock's global attributes are also available for actions that support Lava. You can get a full list of these attributes under `Admin Tools > General Settings > Global Settings`. These attributes can be accessed through Lava using:

`{{ 'Global' | Attribute:'[AttributeKey]' }}`

Examples:

- `{{ 'Global' | Attribute:'OrganizationName' }}`
- `{{ 'Global' | Attribute:'OrganizationAddress' }}`

Additional Merge Fields

Merge Fields for Sending Emails

When using the *Email Send*, or *Email Send Template* actions, and using a person type attribute as the Send To Attribute Value, there will be an additional *Person* merge field that contains the current recipient. This is a full person object and can be used to reference properties of the person. For example, a mergefield of {{ Person.NickName }} would display the recipient's nick name.

Merge Fields for Entry Forms

In addition to the merge fields described above, the entry form also has access to the current person who is filling out the form. Merge values for the *CurrentPerson* object are listed below.

```
{
    "FirstName": "Alisha",
    "NickName": "Alisha",
    "MiddleName": "Mary",
    "LastName": "Marble",
    "IsDeceased": false,
    "PhotoId": 56,
    "BirthDay": 10,
    "BirthMonth": 10,
    "BirthYear": 1961,
    "Gender": 2,
    "AnniversaryDate": "1980-04-09T00:00:00",
    "GraduationDate": null,
    "Email": "jonedmiston@ccvonline.com",
    "IsEmailActive": true,
    "EmailNote": null,
    "EmailPreference": 0,
    "ReviewReasonNote": null,
    "InactiveReasonNote": null,
    "SystemNote": null,
    "PrimaryAliasId": 1,
    "FullName": "Alisha Marble",
    "BirthdayDayOfWeek": "Friday",
    "BirthdayDayOfWeekShort": "Fri",
    "PhotoUrl": "/GetImage.ashx?id=56",
    "BirthDate": "1961-10-10T00:00:00",
    "Age": 52,
    "DaysToBirthday": 45,
    "Grade": null,
    "GradeFormatted": "",
    "CreatedDateTime": null,
    "ModifiedDateTime": "2014-08-26T21:28:56.99",
    "CreatedByPersonAliasId": null,
    "ModifiedByPersonAliasId": 1,
    "Id": 1,
    "Guid": "ad28da19-4af1-408f-9090-2672f8376f27",
    "UrlEncodedKey": "EAAAACE88i304vQJcwoNWW6P7fZ!2bSxiGgjyCooBFy4H332mMnPJoySCsXjQXVMJF87M
ynUFCAKPz4gIs1RshCy3dL7M!3d",
    "ConnectionStatusValue": {
        "Value": "Member",
    },
    "MaritalStatusValue": {
        "Value": "Married",
    },
    "PhoneNumbers": [
    {
        "NumberTypeValue": {
            "Value": "Mobile",
        },
        "CountryCode": "1",
        "Number": "5555555555",
        "NumberFormatted": "(555) 555-5555",
        "IsMessagingEnabled": true,
        "IsUnlisted": false,
        "NumberFormattedWithCountryCode": "+1 (555) 555-5555",
    }],
    "RecordStatusReasonValue": null,
    "RecordStatusValue": {
        "Value": "Active",
    },
    "SuffixValue": {
        "Value": "Ph.D.",
    },
    "TitleValue": {
        "Value": "Mrs."
    }
}
```

Securing Workflows

While we've already covered workflow security in other chapters, we thought we'd summarize workflow security in one place. This should give you a good understanding of what's possible.

Editing A Workflow Type

To be able to add or edit a workflow type, you’ll need *Edit* access to the workflow configuration page

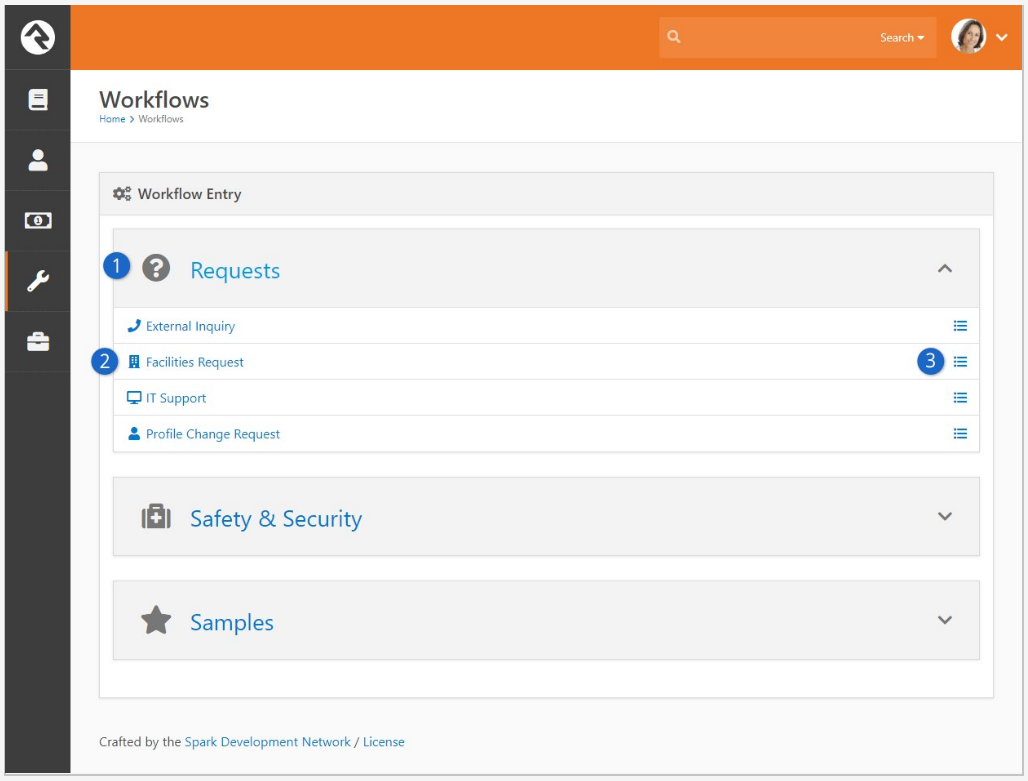
`Admin Tools > General Settings > Workflow Configuration`

and the *Workflow Type Detail* block on it. Currently, only the *Rock Administrator's* security role has access to edit workflow types.

Workflow Navigation Page

The workflow navigation page is a great place for your staff to start and manage workflows. Below is a summary of the security needed to interact with the various components of this page.

Securing Workflow Navigation



1 **View Permission**
One must have *View* permission to the workflow category in order for it to display. The category also must be selected on the block setting to be shown.

2 **View Access**
Once the category is displayed, the current user must also have view access to the workflow type in order for it to be displayed on the list.

3 **List View**
If the logged in individual has *List View* access to the workflow type, they'll also see the ☰ link to be able to view the active/completed workflows of this type.

> Workflow Type Not A Link?
>
> You may notice that when some workflow types are listed they are not linked. This means the workflow type does not have an entry form configured for the current person.

Workflow Entry and URL Links

The following security is required for the Workflow Entry block:

- The user must be authorized to *View* the workflow type in order to enter information.

- The first active action form that user is assigned to will be displayed.
- The user must also be authorized to either *Edit* the block, or *View* the activity type.

Workflow List

The Workflow List block requires that a user must be authorized to *Edit* workflow type in order to view a list or add/edit/delete a workflow.

Workflow Detail

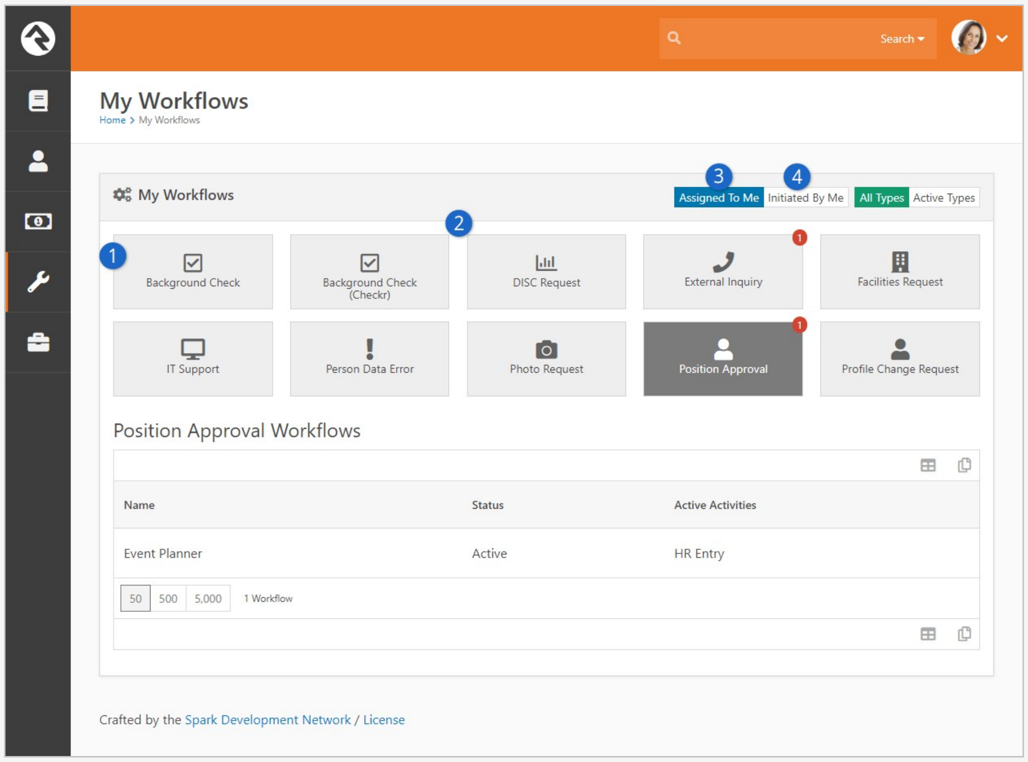
The following security is required on the Workflow Detail block:

- The user must be authorized to *View* the workflow in order to view read-only details.
- The user must be authorized to *Edit* the block, or *Edit* the workflow, in order to edit the workflow.

My Workflows

The My Workflow block has the following security settings.

Securing My Workflows



Position Approval Workflows

Name	Status	Active Activities
Event Planner	Active	HR Entry

50 500 5,000 1 Workflow

Crafted by the Spark Development Network / License

1 **Authorized**
The user must be authorized to *View* the workflow type.

2 **Criteria**
Displays workflows for any activity that meets the following criteria :

- Workflow type is active
- Activity is active and has an active form
- The user must by authorized to *View* the activity

3 **Assigned**
If viewing *Assigned to*, the user must be assigned to the activity

4 **Initiated**
If viewing *Initiated by*, the workflow must have been initiated by user.

Linking To Workflows

All of the workflow entry and management tools in Rock should be able to meet all of your needs. If you want to extend some of the functionality, say linking the *Dynamic Data* block with workflows, it's helpful to know some tricks about interacting with workflows using URL links.

Workflow Entry

The *Workflow Entry* block checks for a *WorkflowId* or *WorkflowGuid* parameter. If found, it will load the existing workflow. If parameters are not included, or workflow is not found, a new workflow is created (In this case, a *WorkflowTypeId* query string parameter is required). Either way, the workflow is processed and the block will look for the first active form action on the first active activity.

If a *command* query string parameter is included, the block will immediately process the form, assuming the user selected the included action.

The out-of-the-box workflow entry page is configured with the route of http:///WorkflowEntry/17. Note that in this case, 17 is the *WorkflowTypeId*.

A Few Technical Details

Attribute Values

When working with workflows and attributes, it's helpful (actually it's pretty much essential) to know how those attributes store their values. Below is a list of a few commonly used attribute field types, with a description of how the value is stored internally.

Field Type	Stored Value
Boolean	'True' or 'False'
Campus	A campus's GUID
Defined Value	A comma-delimited list of defined value GUIDs (if attribute is not configured for multiple values, there should only be one GUID)
Group	A group's GUID
Group Member	A GroupMember Record's GUID
Multi-Select	A comma-delimited list of the values (e.g. 1,2,3) of the selected items
Person	A person alias GUID
Single-Select	The value (e.g. "1") of the selected item
Text	The value of the textbox

The full list of field types can be found in our Workflows & Lava documentation.

When creating workflow actions that update an attribute value, make sure to reference this list so that your attribute values get set correctly.

Webhook to Workflow

Now that you're a workflow guru, let's turn up the volume to 11. As you've seen there's several ways that you can launch workflows built into Rock. But what if we told you you could launch workflows from Twitter, Wufoo, Formstack, Dropbox, Evernote, Email, or... basically any modern web service? That's the power of Webhooks to Workflows.

Most modern Web Services use the concept of webhooks. This allows the service to make calls out to other systems when certain events occur. For instance Formstack can call out to Rock, using a webhook, everytime a form is completed. There are also services like Zapier or Flow that help to manage the flow of webhooks from service to service.

Configuring a Webhook to a Workflow

You configure the linkage from a webhook to a workflow using a *Defined Value*. These are configured under
`Admin Tools > General Settings > Defined Types > Workflow Webhook`.
Each webhook that comes in will go through this list to determine which workflow types to launch. That determination is made by evaluating the Lava in the *Process Request* attribute of the *Defined Value*. If the Lava returns 'True' a workflow of that type will be started.

All of the *Defined Values* will be considered for each webhook that is called. All of the defined values whose *Process Request* Lava returns 'True' will run. It's important that this template is very discerning in making sure that only the correct workflow types are launched. One way to simplify this matching is to use query string parameters in the URL you define in your webhook. The default URL you provide for the webhook is:
http://rock.yourchurch.com/Webhooks/LaunchWorkflow.ashx
To help in your matching you could tack on a query string paramater such as:
http://rock.yourchurch.com/Webhooks/LaunchWorkflow.ashx?WorkflowTypeId=12
You could then provide the following template for your *Process Request*.

```
{% assign workflowTypeId = QueryString['WorkflowTypeId'] %} {% if workflowTypeId == '12' %}
 True {% else %} False {% endif %}
```

This makes the selection of the workflow type very easy.

When No Workflow Is Found

For security reasons if no workflow is matched a 404 error will be returned. This can be confusing at first when you are attempting to debug, but it's a best practice to help ward off evildoers.

Ok, we've now discussed how to select the workflow type to launch. Now let's look at ways we can pass information to those workflows. You'll first notice that you can configure a Lava template for setting the Workflow Name. This allows you to customize the name on each run based on information from the webhook call.

You can also list attribute keys from the workflow and provide Lava templates for setting their values. The challenge here is knowing what data is available to you. The *Defined Type* screen has a feature that shows/hides these available fields. The basic items are:

- Url
- RawUrl
- Method
- QueryString
- RemoteAddress
- RemoteName
- ServerName
- RawBody
- Headers
- Cookies

Each of these items can have a lot of child properties. The documentation on the *Defined Type* screen shows some of these details, but the actual values will change greatly based on the calling service. We recommend that you start by setting various test attributes with items like 'RawBody' to see what is available.

While you can put Lava into the Workflow Atttribute configuration on the *Defined Value* it's recommended that a majority of that logic be done inside of the workflow using the RawBody. The Lava on the *Defined Value* can't contain special characters like a | that are require in Lava filters.

Example

For instance if you passed the RawBody into the workflow using the attribute key of 'Body', then the Lava below could be used to access properties of the body.

```
{% assign body = Workflow | Attribute:'Body' | FromJSON -%} {{ body.propertyname }}
```

SMS Pipeline Workflows

In this chapter we'll describe how to configure workflows that are launched from the `SMS Pipeline`.

> **Looking for Text to Workflow?**
> Before the SMS Pipeline was introduced, Text to Workflow was the go-to solution for automating SMS processes in Rock. If you're already using Text to Workflow your configuration will still work, but we recommend using the `SMS Pipeline` feature going forward. You'll find it's even more powerful and flexible than Text to Workflow.

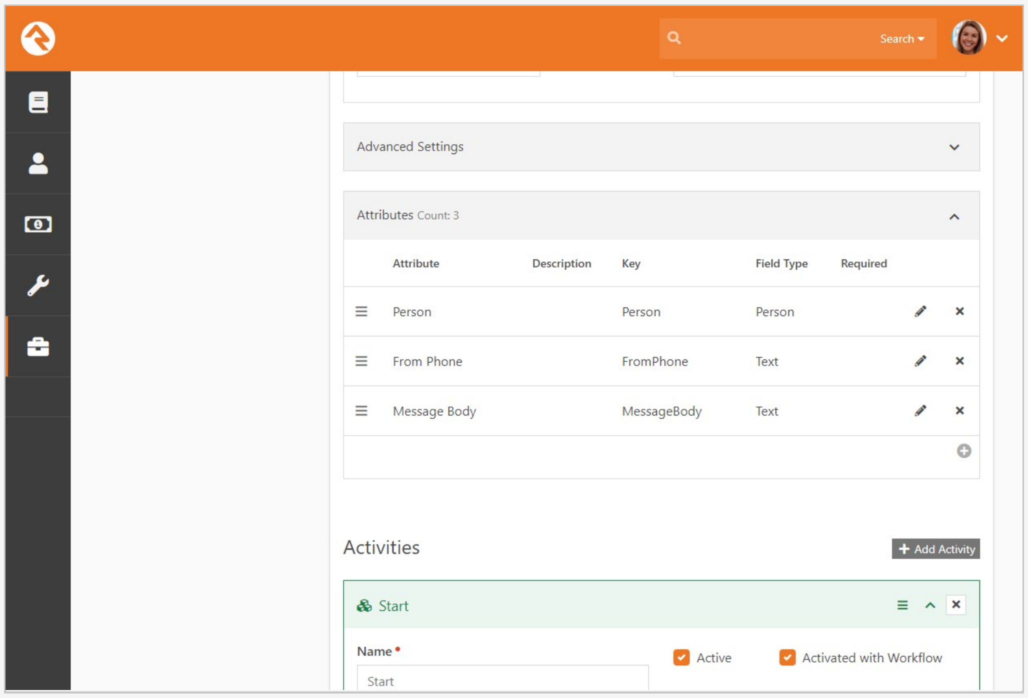
Workflows initiated from the *SMS Pipeline* page are configured like other types of workflows described in this guide. The only difference is that these workflows will need to be configured to receive information about the SMS message from the pipeline. We'll walk you through that process below.

Workflow Configuration for SMS Pipeline

As you've learned throughout this guide, there are lots of different ways to configure your workflow types. The example below is not intended to be a set of instructions that you must follow exactly for your workflow to function. We'll go over other options a little later. For now, we'll walk through an example just to explain the features.

First, keep in mind that the pipeline can send the information it has to your workflow automatically. For instance, your workflow configuration doesn't require an action to identify the person. You'll still need workflow attributes to store that information, so that's where we'll start. In this example we've set up only a few attributes that the pipeline can populate. See below for the full list.

Configure Workflow Attributes


Advanced Settings

Attributes Count: 3

Attribute	Description	Key	Field Type	Required
Person		Person	Person	
From Phone		FromPhone	Text	
Message Body		MessageBody	Text	

Activities

Start

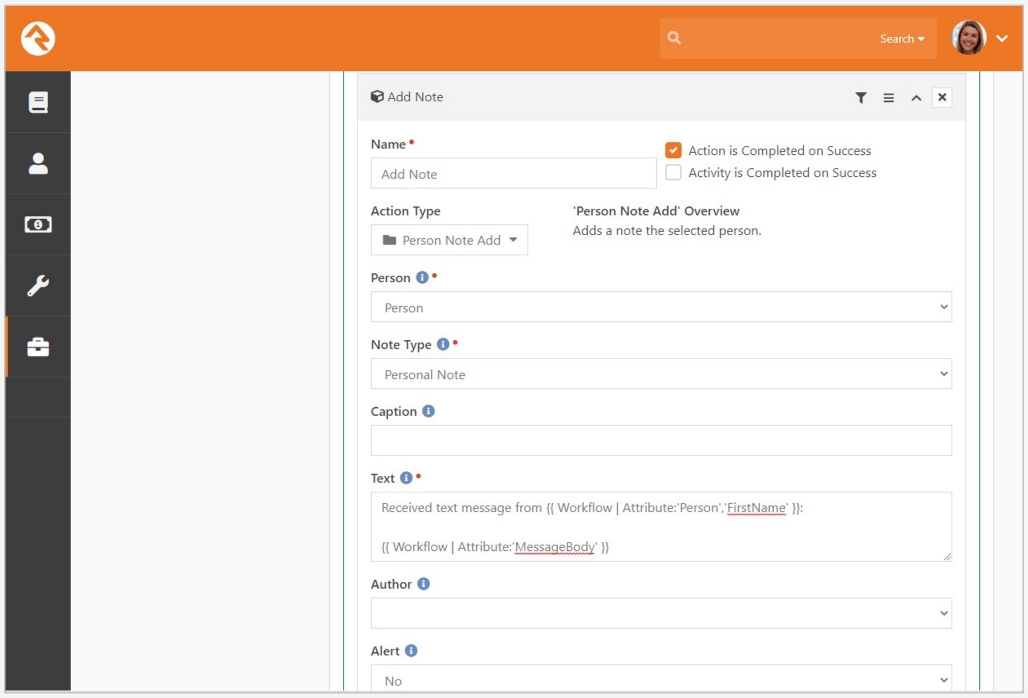
Name *: Start

✓ Active ✓ Activated with Workflow

These attributes allow the person, their phone number and the content of their SMS text message to be sent directly to the workflow. From there, you can perform any number of Actions or Activities using those attributes.

In the example pictured below, we'll add a Person Note that contains the person's name and the text of the SMS message they sent.

Example Workflow Action


Add Note

Name *: Add Note

✓ Action is Completed on Success

☐ Activity is Completed on Success

Action Type: Person Note Add

'Person Note Add' Overview

Adds a note the selected person.

Person *: Person

Note Type *: Personal Note

Caption:

Text *: Received text message from {{ Workflow | Attribute:'Person','FirstName' }}:

{{ Workflow | Attribute:'MessageBody' }}

Author:

Alert: No

As you can see in the above example, your workflow configuration for pipelines won’t be very different from other kinds of workflows. Actions and Attributes are set up and used in the same ways. However, there is some configuration you’ll need within the SMS Pipeline itself to make it all work. We’ll cover that in the next section.

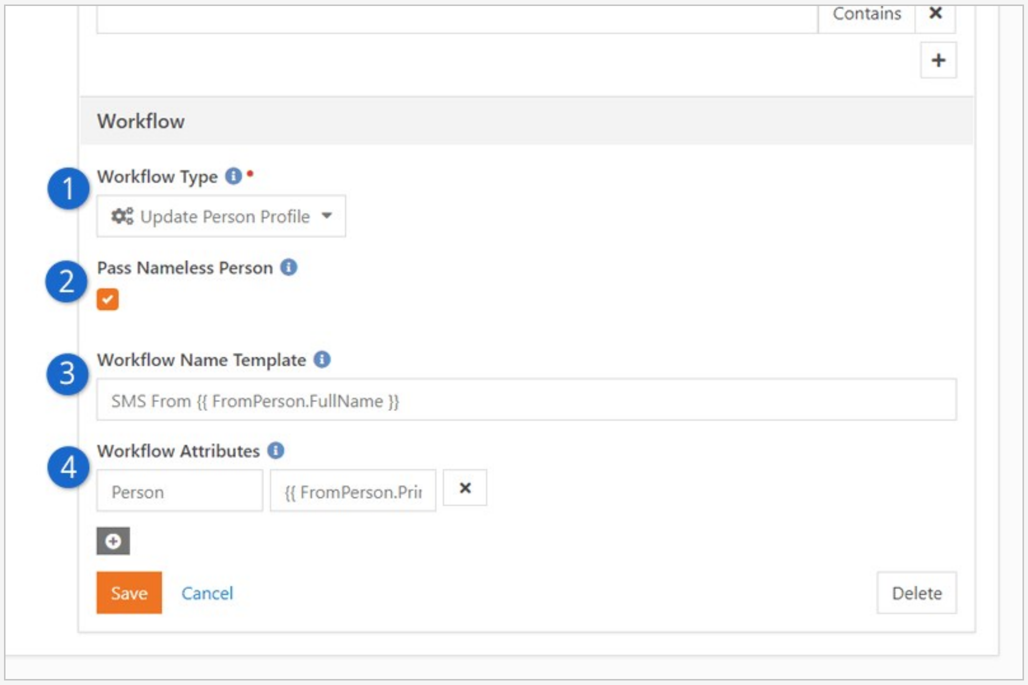
Working with the SMS Pipeline

After your workflow has been configured, you're ready to add it to the *SMS Pipeline*. In this section we'll highlight the pipeline configuration related to workflows.

For information on *SMS Pipelines* in general, see the SMS Pipeline chapter of our `Communicating with Rock` guide.

Pictured below is a basic example of a workflow configuration in the pipeline. Keep in mind the workflow configuration that was described in the prior section, because it will be referenced here.

SMS Pipeline - Workflow Configuration



SMS From {{ FromPerson.FullName }}

Person | {{ FromPerson.Pri

1. **Workflow Type**
   This is the workflow type that we want to launch for an SMS message.
2. **Pass Nameless Person**
   You can disable this setting to prevent *Nameless People* records from being sent to your workflow. We'll talk more about working with *Nameless People* below.
3. **Workflow Name Template**
   This field lets you dynamically assign the name of each launched workflow, for each SMS message received. In this example, the workflow name would be "SMS From Ted Decker" if Ted Decker sent the message.
4. **Workflow Attributes**
   This is where you can feed information from the SMS message directly into your workflow. You'll use the *Key* value of the workflow's attribute in the first field. In the second field you'll add the Lava that identifies how you want to fill in that attribute's value.

As described above, your pipeline configuration can send attribute values directly to your workflow's attributes. In the next section, we'll look at the details of how this works and the different attributes you can set.

SMS Pipeline - Workflow Merge Fields

The SMS messages that you receive have more information associated with them than you might think. You can use Lava to send that information directly from the pipeline to the workflow.

In the example pipeline above we used `{{ FromPerson.PrimaryAlias.Guid }}` in the *Workflow Attributes* Merge Template field. Using `{{ FromPerson.PrimaryAlias.Guid }}` identifies the person who sent the SMS message in a way that the workflow can receive it. This will automatically be assigned to the 'Person' workflow attribute when the workflow is launched, using the attribute's *Key* as a reference point between the pipeline and the workflow. Similarly, we used `{{ FromPerson.FullName }}` to set the name of the launched workflow. Adding `.FullName` gives you the name of the person, as text.

> **Warning: FromPhone is automatic**
> Don't add an attribute called `FromPhone` to the Workflow action in the pipeline unless you are intentionally trying to overwrite it. The one that is automatically placed there is the 'raw' value which has the '1' (country code) prepended onto it. If you manually add `FromPhone`, it will overwrite the value and use the one without the country code. That could lead to unexpected behavior in your workflow.

The table below lists other merge fields you can use to pass different types of information from the pipeline to your workflow's attributes.

SMS Pipeline Lava Merge Fields

Merge Field	Description	Field Type
{{ FromPerson.PrimaryAlias.Guid }}	The pipeline uses the person's phone number to look up the first person with that phone number. If it finds a match, it assigns that individual's record to FromPerson. If the phone number is used in more than one profile, the pipeline defaults to the first record of an adult with children.	Person
{{ FromPhone }}	The person's phone number, pulled from the inbound message, from the SMS gateway. This will automatically get added to the workflow as FromPhone and will include the country code (i.e., the raw phone number 18645555555)	Phone Number
{{ ToPhone }}	The SMS gateway number where the message was sent	Phone Number
{{ ReceivedDate }}	The date the message was received	Date
{{ ReceivedTime }}	The time the message was received	Time

SMS Pipeline Lava Merge Fields

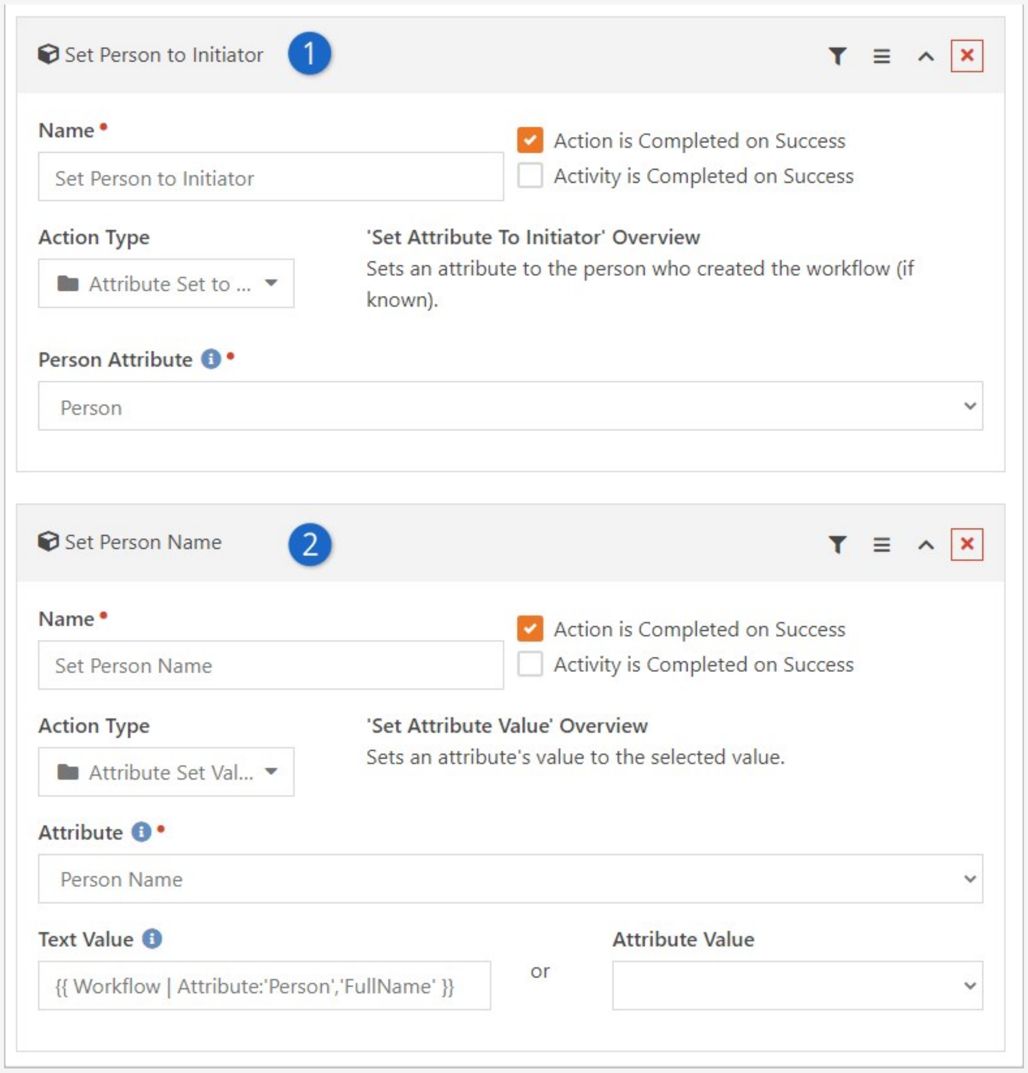
Merge Field	Description	Field Type
{{ ReceivedDateTime }}	The date and time the message was received	Date Time
{{ MessageBody }}	The content of the SMS message that was received	Text or Memo
{{ MatchedGroups }}	If the RegEx expression provided contains match groups, they are loaded into an array here. This is an advanced feature, so if you're not sure what this means, don't worry. You probably don't need it.	Typically, you fill in a text field with a merge expression of a single result from the MatchedGroups array.

Using Attribute Set to Initiator

There are some reasons why you might want to have the workflow find values for its attributes instead of sending them from the pipeline. The first consideration is maintenance. The workflow attributes and the pipeline configuration must be kept in sync to work properly. Using the *Attribute Set to Initiator* approach instead, you don't have to worry about changes to the pipeline impacting your workflow's functionality as long as your workflow accounts for different scenarios.

The above point leads to the second consideration, which is flexibility. Using a method like *Attribute Set to Initiator* means the workflow can be launched from a variety of different sources. It will be more difficult to use the workflow in other contexts if it's designed to rely on specific data from the pipeline.

Get Person and Name



Set Person to Initiator
1
Name
Set Person to Initiator
Action is Completed on Success
Activity is Completed on Success
Action Type
Attribute Set to ...
'Set Attribute To Initiator' Overview
Sets an attribute to the person who created the workflow (if known).
Person Attribute
Person
Set Person Name
2
Name
Set Person Name
Action is Completed on Success
Activity is Completed on Success
Action Type
Attribute Set Val...
'Set Attribute Value' Overview
Sets an attribute's value to the selected value.
Attribute
Person Name
Text Value
{{ Workflow | Attribute:'Person','FullName' }}
or
Attribute Value

1 **Attribute Set to Initiator**
The person who sent the SMS message is the initiator. By assigning the initiator to the "Person" attribute, we now have the person's record in the workflow.

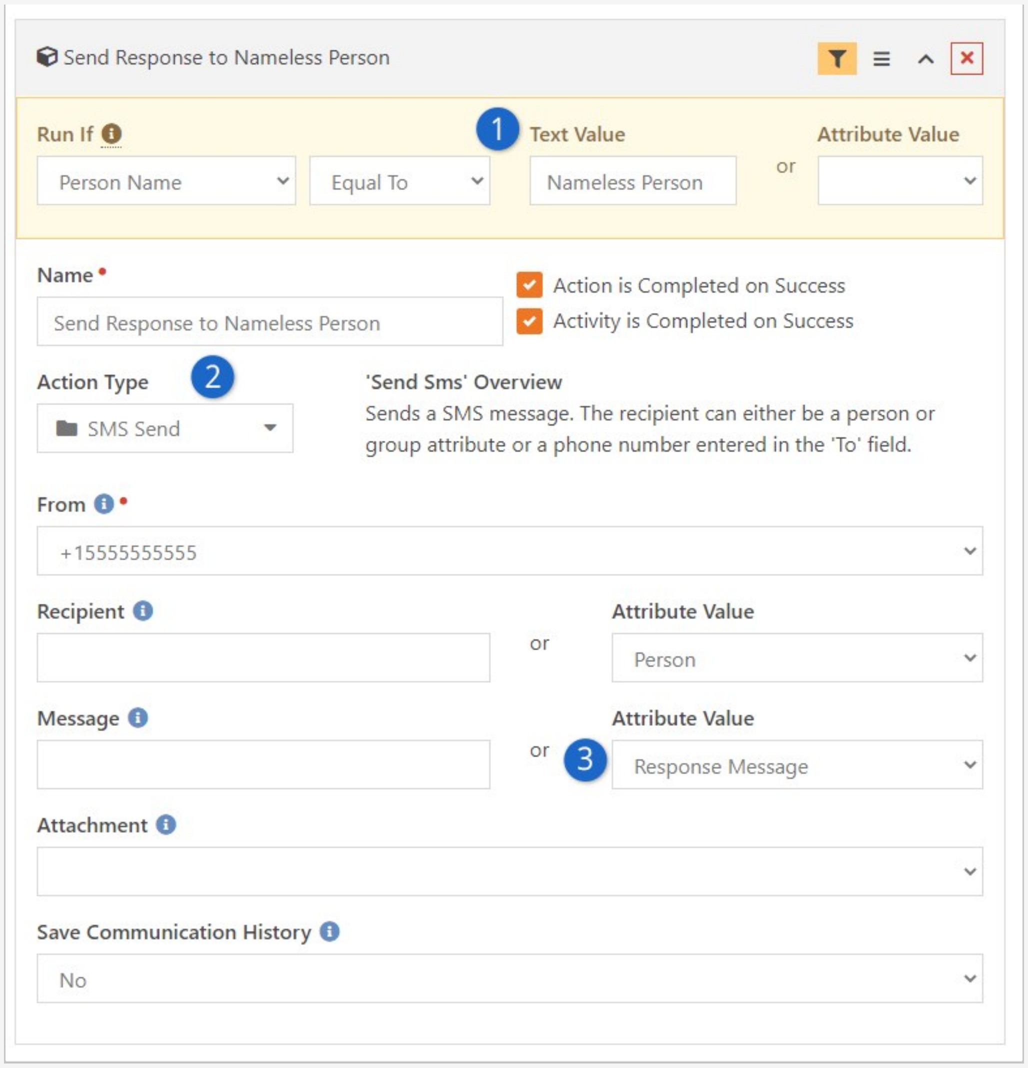
2 **Attribute Set Value**
Here we'll use some Lava to get the name of the person we identified above. We'll use the name in the next workflow action (see below) to see if the person has a record in Rock.

Let's pause here for a moment to mention *Nameless People*. When an SMS message is received, Rock will automatically try to identify the person. If the person isn't in Rock, a *Nameless Person* record is created. This special type of record gives Rock a way to keep track of the person even though all we have is their phone number. For more details, see the Nameless People section of our `Communicating with Rock` guide.

The next workflow action, pictured below, is where we'll check for a *Nameless Person*. Your workflow options are limited if you don't have a person's record to work with.

However, that doesn’t mean your process needs to exclude *Nameless People*. For instance, you can send the person a link to create a profile as pictured in the example below.

Nameless Person - Send SMS


Send Response to Nameless Person

Run If: Person Name | Equal To | Text Value: Nameless Person | or | Attribute Value:

Name: Send Response to Nameless Person

✓ Action is Completed on Success

✓ Activity is Completed on Success

Action Type: SMS Send

'Send Sms' Overview

Sends a SMS message. The recipient can either be a person or group attribute or a phone number entered in the 'To' field.

From: +15555555555

Recipient: | or | Attribute Value: Person

Message: | or | Attribute Value: Response Message

Attachment:

Save Communication History: No

1 **Action Filter**
This workflow action filter checks the name of the initiator, which we obtained in the prior action. If the person's name is "Nameless Person" then we know the sender of the SMS message doesn't have a record in Rock.

2 **Send SMS Message**
In this case, we'll send an SMS message to the person. We can do that because the *Nameless Person* record contains the phone number the person used to send their original message.

3 **Message Attribute**
The content of the SMS message the person receives is maintained as a workflow attribute in this example. The message encourages the person to visit the website to create a profile, so we'll know who they are going forward.

If it's not a *Nameless Person* record then the workflow action described above wouldn't run. Instead, we now know the workflow has a complete person record in-hand. You can use that information to perform any other actions or activities you need. At this point in the workflow configuration we've left the pipeline behind. From here, the same tools and features described earlier in this guide can be used to build the rest of your workflow according to your needs.