

**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 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.

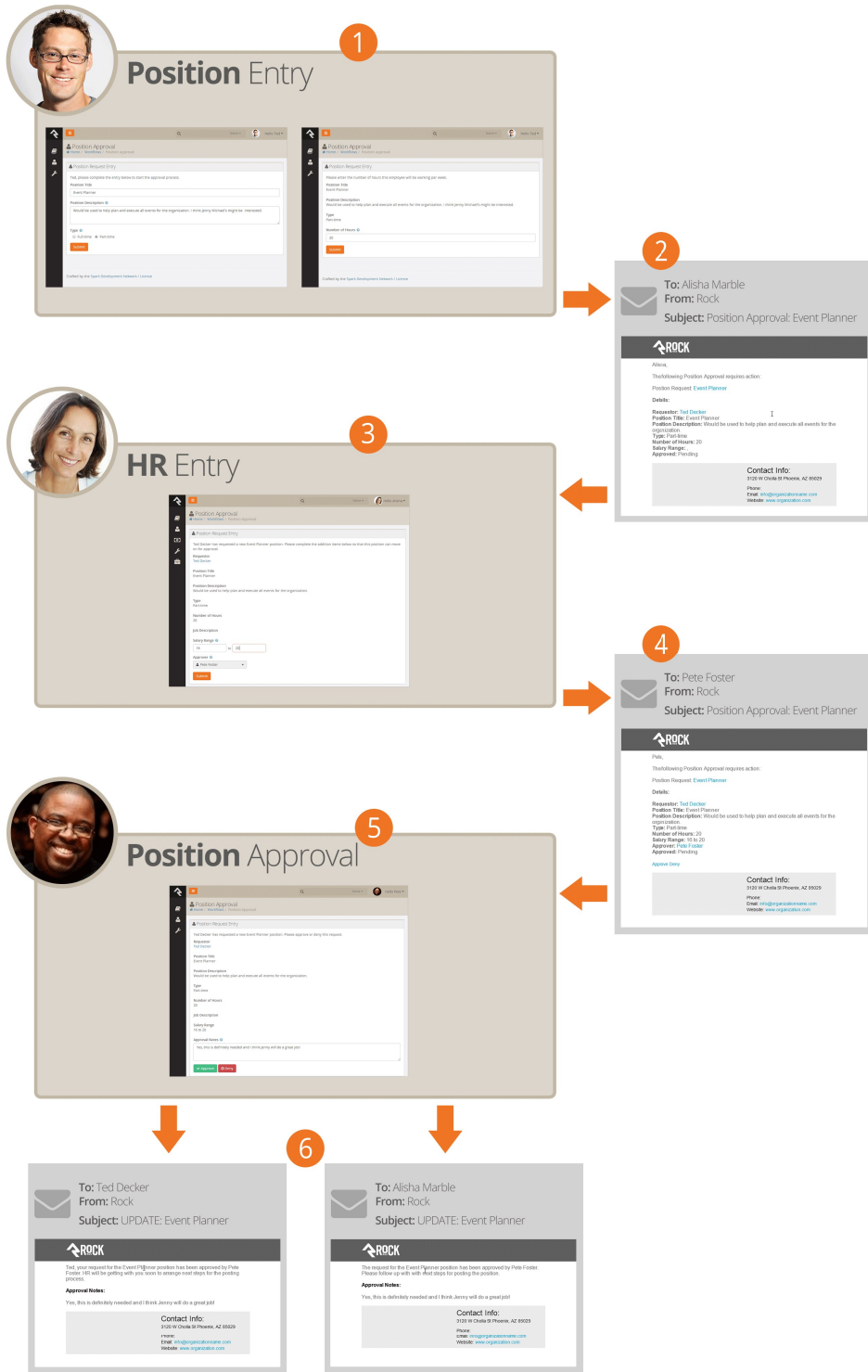
When you're done with this manual, we think you'll see how workflows empower you to create powerful application logic without needing to become a programmer. Once you understand the basics, your mind will start racing with all of the ways you can put them to use.

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



**1** 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 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 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 The workflow now sends Pete an email regarding the position. From the email, Pete can click to approve or deny the position.
- 5 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 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 detail screen you can view important information about your workflow.



## Workflow Type View

The screenshot displays the 'Workflow Configuration' page for a 'Position Approval' workflow. The interface is organized into several sections:

- 1. Workflow Type Navigation:** A sidebar on the left contains a list of categories: Check-in, Data Integrity, Requests, Safety & Security, and Samples. The 'Position Approval' workflow is highlighted under the 'Samples' category.
- 2. Name:** The workflow is named 'Position Approval'.
- 3. Description:** The description states: 'This workflow will be used by the staff of Rock Solid Church to request the opening of a new position on staff.'
- 4. Activities:** A list of five activities is shown, each with a description and a list of actions:
  - 1. Initial Entry:** This activity handles the initial entry of the request by the staff person. Actions: 1. Get Initial Input From User, 2. Persist Workflow, 3. Set Requestor, 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.
  - 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, 2. HR Entry.
  - 3. Approval Process:** This activity is used to get approval from the select approver. Actions: 1. Assign to Approver, 2. Approval Entry.
  - 4. Approved:** This activity is activated if the request has been approved. Actions: 1. Mark Request Approved, 2. Email Requester, 3. Email HR, 4. Mark Workflow Complete.
  - 5. Denied:** This activity is activated if the request is denied. Actions: 1. Mark Position Denied, 2. Email Requester, 3. Email HR, 4. Mark Workflow Complete.
- 5. Copy:** An 'Edit' button and a 'Delete' button are located at the bottom left of the workflow details.
- 6. Action Buttons:** A 'Copy' button and a 'Lock' icon are located at the bottom right of the workflow details.

- 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 & 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

## Details

The screenshot shows the 'Workflow Type Details' page for a workflow named 'Position Approval'. The page is divided into a sidebar and a main content area. The sidebar contains navigation options: 'Add Category', 'Workflow Type', 'Check-in', 'Data Integrity', 'Requests', 'Safety & Security', and 'Samples'. The main content area is titled 'Position Approval' and contains a 'Details' section with the following fields:

- 1 Name:** A text input field containing 'Position Approval'.
- 2 Active:** A checked checkbox.
- 3 Automatically Persisted:** An unchecked checkbox.
- 4 Description:** A text area containing 'This workflow will be used by the staff of Rock Solid Church to request the opening of a new position on staff.'
- 5 Work Term:** A text input field containing 'Position Request'.
- 6 Workflow Number Prefix:** An empty text input field.
- 7 Category:** A dropdown menu with 'Samples' selected.
- 9 Icon CSS Class:** A text input field containing 'fa fa-user'.
- 8 Processing Interval (minutes):** A text input field containing '0'.
- 10 Logging Level:** A dropdown menu with 'Action' selected.

- 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 is dependent upon 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 verbosity you need (none is nothing while action is pretty much everything).

## Attributes

# Workflow Type Attributes

**Workflow Configuration**  
Home / General Settings / Workflow Configuration

**Position Approval** (Id: 17)

Attributes Count: 10

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

**Activities** (+ Add Activity)

- 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.

Save Cancel

Crated by the Spark Development Network / License

Attributes are the data elements your workflow needs to be able to process. In this

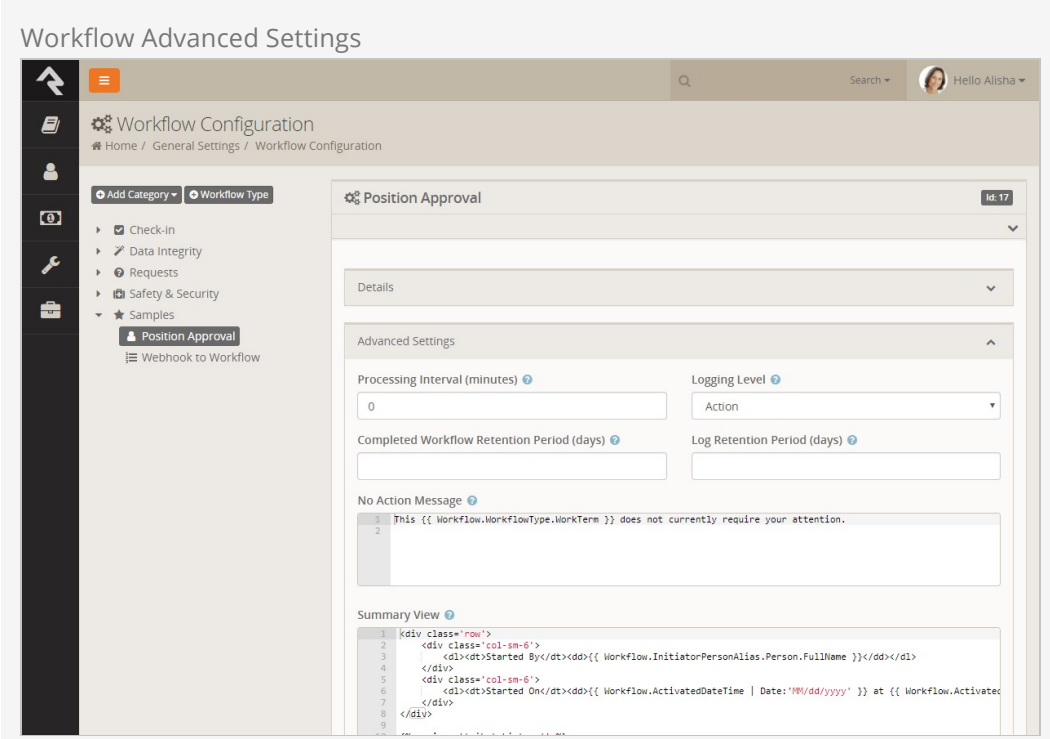
section you configure each of these elements and define the types of data they will store (text, numbers, dates, people, groups, etc.)

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.

## Advanced Settings

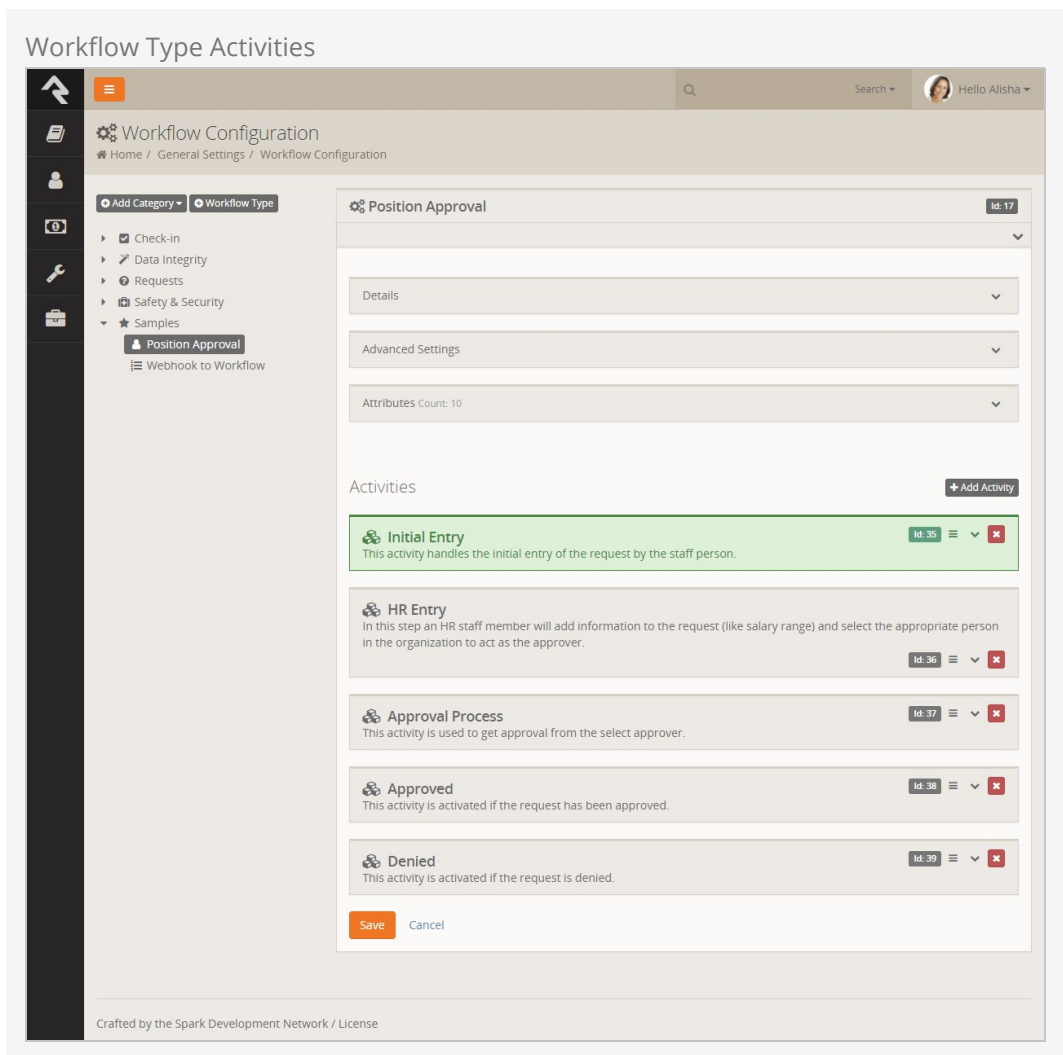


The screenshot shows the 'Workflow Advanced Settings' page for a 'Position Approval' workflow. The interface includes a navigation sidebar on the left with options like 'Check-in', 'Data Integrity', 'Requests', 'Safety & Security', 'Samples', 'Position Approval', and 'Webhook to Workflow'. The main content area is titled 'Position Approval' and contains several configuration fields:

- Processing Interval (minutes):** A text input field with the value '0'.
- Logging Level:** A dropdown menu set to 'Action'.
- Completed Workflow Retention Period (days):** A text input field that is currently blank.
- Log Retention Period (days):** A text input field that is currently blank.
- No Action Message:** A text area containing the message: 'This {{ Workflow.WorkflowType.WorkTerm }} does not currently require your attention.'
- Summary View:** A code editor showing HTML snippets for workflow details, including fields for 'Started By' and 'Started On'.

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.

## Activities/Actions



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 just know that this is where you'll configure them for your workflow types.

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 ( ).

# 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

- 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 temporarily 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 over-written 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

- 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 always be performed every time the workflow is processed.
- 3 Activity Is Completed On Success:**

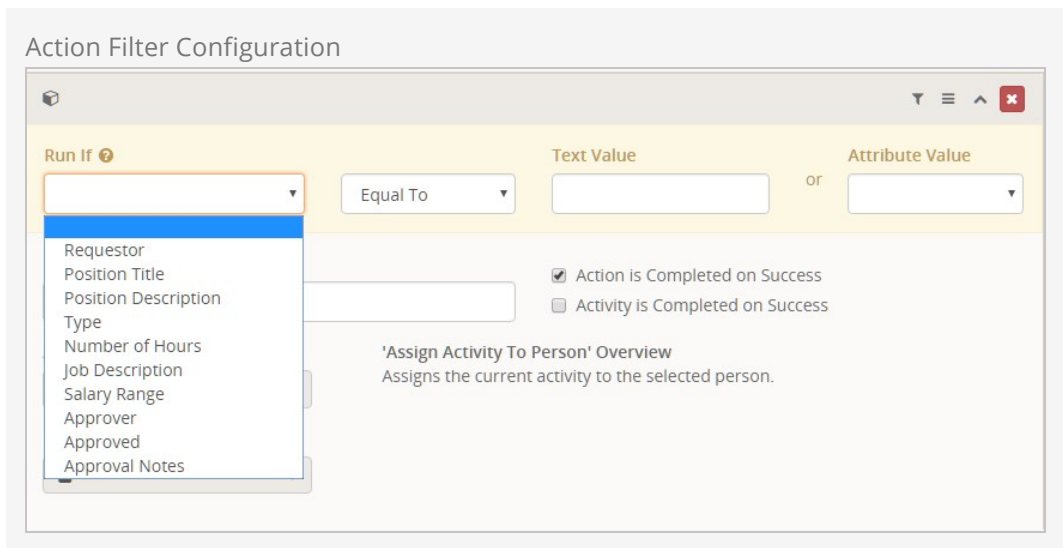
This setting tells the workflow engine to mark the entire activity as complete if the action successfully runs. This has the effect of keeping all following actions from running. If you are familiar with programming logic, this is similar to a *break* statement.
- 4 Action Type:**

This is where you configure what type of action you want to perform. Each of these action types are discussed below.
- 5 Action Description**
- 6 Action Configuration:**

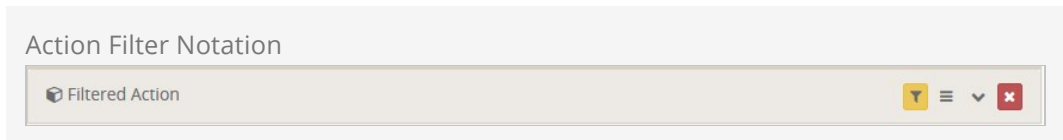
This part of the action configuration will be unique to each action type.

## 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.



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



Although most of the filter match criteria are self-explanatory, the *Regular Expression* is possibly unfamiliar 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 her-self", 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 an 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

Below we outline each action that comes with Rock, providing tips on when and how to use them. Screenshots of the settings for an action type are provided for those that make sense.

### 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 document since very few people will be using them.

For a complete listing of Rock's workflow actions please see the [On-line Workflow Actions](#) documentation.

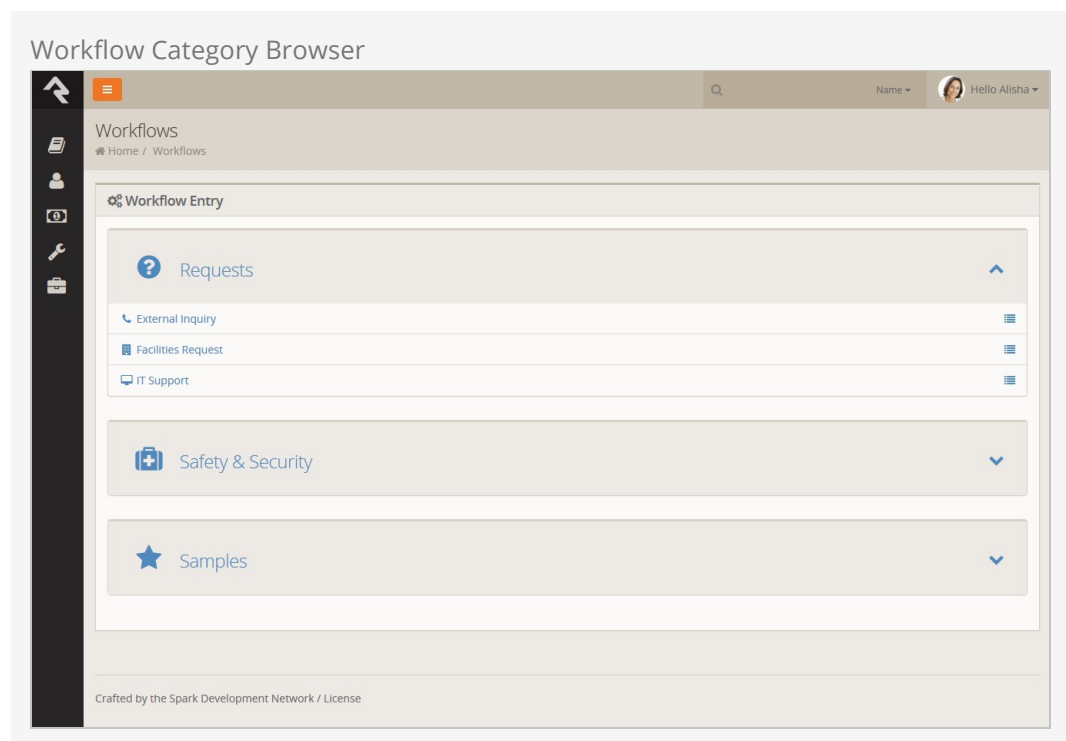
# Launching Workflows

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



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.

## Person Profile Actions

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 to 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

Workflow Trigger

Home / General Settings / Workflow Triggers / Workflow Trigger

Add Workflow Trigger

Trigger Type 1  
 Pre-Save  Post-Save  Pre-Delete  Post-Delete  Immediate-Post-Save

Entity Type 3  
Group

Workflow Type 5

Workflow Name 6

Active 2

Entity Type Qualifier Column 4

Value is 4

Save Cancel

Crafted by the Spark Development Network / License

- 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 (example: a pre-save or immediate-post-save trigger on a person that fires a workflow to updates 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.

### 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.

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 to the user so keep that in mind when using pre triggers.

One downfall of pre triggers is that if they are initiated by a user working in Rock, the user must wait for the workflow to launch and complete before their 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 the user experience quick.

### 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.

# 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. The engine then goes 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 it will be marked complete.

The active workflow will continue to be executed on the polling frequency until the workflow gets 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 you know 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 under Jobs Administration. It can be configured 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 *User Entry 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.

## User Entry 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.

The screenshot shows the configuration interface for the 'Get Initial Input From User' action. The window title is 'Form Entry Sample'. The configuration includes the following fields and options:

- Name:** A text input field containing 'Get Initial Input From User'.
- Action Type:** A dropdown menu set to 'User Entry Form'.
- Notification Email:** A dropdown menu set to 'None'.
- Enable Note Entry:** A checkbox labeled 'Yes' which is unchecked.
- Form Header:** A text area containing HTML code: 

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

 A red circle with the number '1' is placed above the first line of code.
- Form Fields:** A table with columns for 'Field', 'Visible', 'Editable', 'Required', 'Hide Label', 'Pre-HTML', and 'Post-HTML'. A red circle with the number '2' is placed above the 'Field' column header.

Field	Visible	Editable	Required	Hide Label	Pre-HTML	Post-HTML
Position Title	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

☰	Position Description	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
☰	Type	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
☰	Number of Hours	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
☰	Job Description	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
☰	Requestor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
☰	Salary Range	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
☰	Approver	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
☰	Approved	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
☰	Approval Notes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Form Footer ⓘ 3

4

Command Label	Button Type	Activate Activity	Response Text
<input type="text" value="Submit"/>	<input type="text" value="Primary"/>	<input type="text"/>	<input type="text" value="Your information has been submit"/>

Command Selected Attribute ⓘ

- 1 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 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 The footer text is a great place to add information about what will happen

next in the process.

- 4 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

The screenshot shows a workflow configuration interface titled "Entry Form Chaining". At the top right, there is a "+ Add Action" button. Below it, a list of actions is displayed, each in a light gray box with a red 'x' delete button on the right. The actions are numbered 1 through 4 in orange circles:

- 1. Get Initial Input From User
- 2. Persist Workflow
- 3. Set Requestor
- 4. Get Full-time Details

Below the numbered actions, there are several more actions, some of which are indented:

- Set Workflow Name
- Set Full-time Hours to 40
- Get Part-time Details
- Send to HR for Additional Fields

Each action has a dropdown menu icon (three horizontal lines) and a red 'x' delete button.

- 1 Our initial entry form, which included the attribute of *Full-time or Part-time*.
- 2 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 Next, we set the requester for the workflow and also provide a workflow name.
- 4 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 *Send Email* action to do this, there is a short-cut built into the *Entry Form* action.

## Entry Form Email

Actions + Add Action

Get Initial Input From User T ☰ ^ ✕

Name \*   Action is Completed on Success  
 Activity is Completed on Success

Action Type

Notification Email 1   Include Actions in Email 2  
 Yes

Enable Note Entry  Yes

Form Header ?

```

1 <h1>Position Details</h1>
2 <p>
3   {{CurrentPerson.NickName}}, please complete the entry below to start the
4   approval process.
5 </p>

```

- 1 You can select a pre-configured email template to use for the email.
- 2 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 email 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 *Send Email* action, or you can create a new template. This new template can be created under [Admin Tools > Communications > System Emails](#). 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.

## 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.

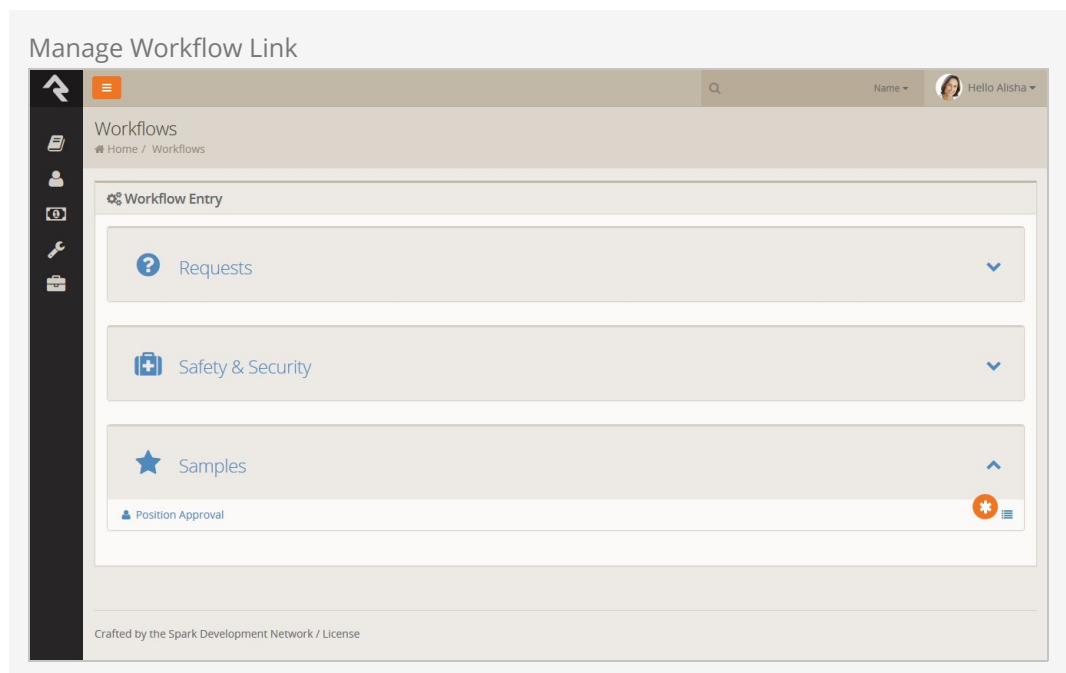
### Cancel Buttons

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 occurring 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



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

<input type="checkbox"/>	Name	Initiated By	Activities	Created	Status	State		
<input type="checkbox"/>	Operations Team Lead	Alisha Marble	HR Entry	1 Minute Ago	Active	Active	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	Event Planner	Alisha Marble	HR Entry	2 Minutes Ago	Active	Active	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>	Receptionist	Alisha Marble	HR Entry	4 Minutes Ago	Active	Active	<input type="text"/>	<input type="text"/>

50 500 5,000 3 Position Requests

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

Workflow Detail  
Home / Workflows / Position Approval / Workflow Detail

### Event Planner

Position Approval Active

Details Activities Log

1	Name	Event Planner	3	Activated	10/11/2015 11:12 PM (11 hours ago)
	Initiator	Alisha Marble		Last Processed	10/11/2015 11:26 PM (11 hours ago)
	Status	Active		Completed	
	Requestor	Alisha Marble		Position Title	Event Planner
	Position Description	test		Type	Full-time
	Number of Hours	40		Job Description	<input type="button" value="View"/>
	Salary Range	.		Approver	Pending
	Approved			Approval Notes	

4 Edit Cancel

5 Notes

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 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 an 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 <sup>1</sup>  Completed  Yes

Initiator  Activated: 10/11/2015 11:12 PM (11 hours ago)

Status Text

<sup>2</sup> Requestor

Position Title

Position Description

Type  None  Full-time  Part-time

Number of Hours

Job Description

Salary Range  to

Approver

Approved

Approval Notes

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

The screenshot displays the 'Workflow Detail' page for an 'Initial Entry' activity. The interface includes a sidebar with navigation icons, a top navigation bar with the user's name 'Hello Alisha', and a main content area. The 'Initial Entry' activity is marked as 'Completed' (2). It has assignment fields for 'Person' (3), 'Group', and 'Security Role'. A 'Completion Flag' (4) is set to 'Yes'. Below these are 'Actions' (5) with columns for 'Action', 'Last Processed', and 'Completed'. The actions list includes 'Get Initial Input From User', 'Persist Workflow', 'Set Requestor', 'Set Workflow Name', 'Get Full-time Details', 'Set Full-time Hours to 40', 'Get Part-time Details', and 'Send to HR for Additional Fields'. At the bottom, there is an 'HR Entry' section and 'Save' and 'Cancel' buttons.

**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 accidentally 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

Home / My Workflows

My Workflows

Initiated By Me Assigned To Me Active Types All Types

Background Check DISC Request External Inquiry Facilities Request IT Support

Person Data Error Photo Request Position Approval

Position Approval Workflows

Name	Status	Active Activities
Receptionist	Active	HR Entry
Event Planner	Active	HR Entry
Operations Team Lead	Active	HR Entry

50 500 5,000 3 Workflows

Crafted by the Spark Development Network / License

### 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.

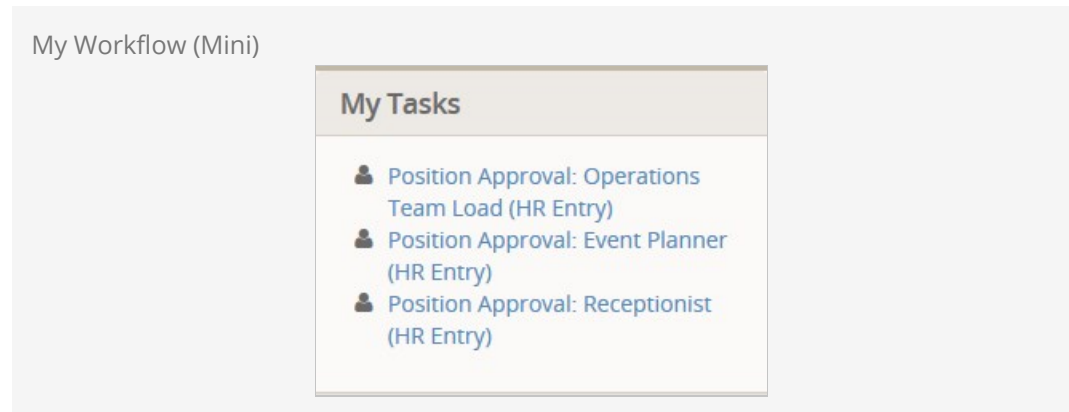
### 4 Workflow Grid:

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.



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






















The screenshot shows the 'Position Approval' workflow configuration page. The left sidebar contains a navigation menu with categories like 'Check-in', 'Data Integrity', 'Requests', 'Safety & Security', and 'Samples'. The main content area is titled 'Position Approval' and includes a 'Details' section and an 'Advanced Settings' section. The 'Details' section has fields for 'Name' (Position Approval), '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), 'Category' (Samples), 'Workflow Number Prefix', and 'Icon CSS Class' (fa fa-user). The 'Advanced Settings' section includes 'Processing Interval (minutes)' (0), 'Logging Level' (Action), 'Completed Workflow Retention Period (days)', and 'Log Retention Period (days)'. A 'No Action Message' field contains a message: 'This {{ Workflow.WorkflowType.WorkTerm }} does not currently require your attention.' Three orange callout boxes with numbers 1, 2, and 3 point to the 'Automatically Persisted' checkbox, the 'Icon CSS Class' field, and the 'Logging Level' dropdown, respectively.

- 1 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 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 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	
Requestor	The person who is making the request.	Requestor	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 show 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 Attributes

Activities + Add Activity

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

**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.

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

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

**Denied**  
This activity is activated if the request is denied.

Save Cancel

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 \***   Active  Activated with Workflow 🔒

**Description**

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

Attributes ▼

Actions + Add Action

1
➔ Get Initial Input From User
⌵ ≡ ▼ ✖

2
➔ Persist Workflow
⌵ ≡ ▼ ✖

3
➔ Set Requestor
⌵ ≡ ▼ ✖

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 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 Next we persist the workflow, now that the user has entered information, so that we can maintain the values over time.
- 3 We can now set the requester for the workflow with the current person who just entered the form.
- 4 Another setup step is to provide the workflow with a name. This name will be used on the various grid displays.
- 5 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 The next step sets the position hours to 40 if the use selected full-time, using action filters.
- 7 This action collects information for part-time positions. It also uses filter actions to only display when appropriate.

- 8 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.

Name \*

  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 + Add Action

- 1 🔗 Assign To HR Member ⌵ ☰ ⌵ ✖
- 2 🔗 HR Entry ⌵ ☰ ⌵ ✖

- 1 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 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 \*   Active  Activated with Workflow

Description

Attributes

Actions + Add Action

- 1 Assign to Approver
- 2 Approval Entry

- 1 Just like the HR Entry, we assign this activity to the approver who was entered by human resources.
- 2 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 <sup>\*</sup>   Active  Activated with Workflow

Description

Attributes

Actions + Add Action

- 1 Mark Request Approved
- 2 Email Requester
- 3 Email HR
- 4 Mark Workflow Complete

- 1 The first action marks the request as either approved or denied.
- 2 Next, we email the requester with the approval response.
- 3 Then, we send an email to the human resources representative so that they are aware of the approval state.
- 4 Finally, we mark the workflow as complete.

## Workflow Tips

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

- 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.

# 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

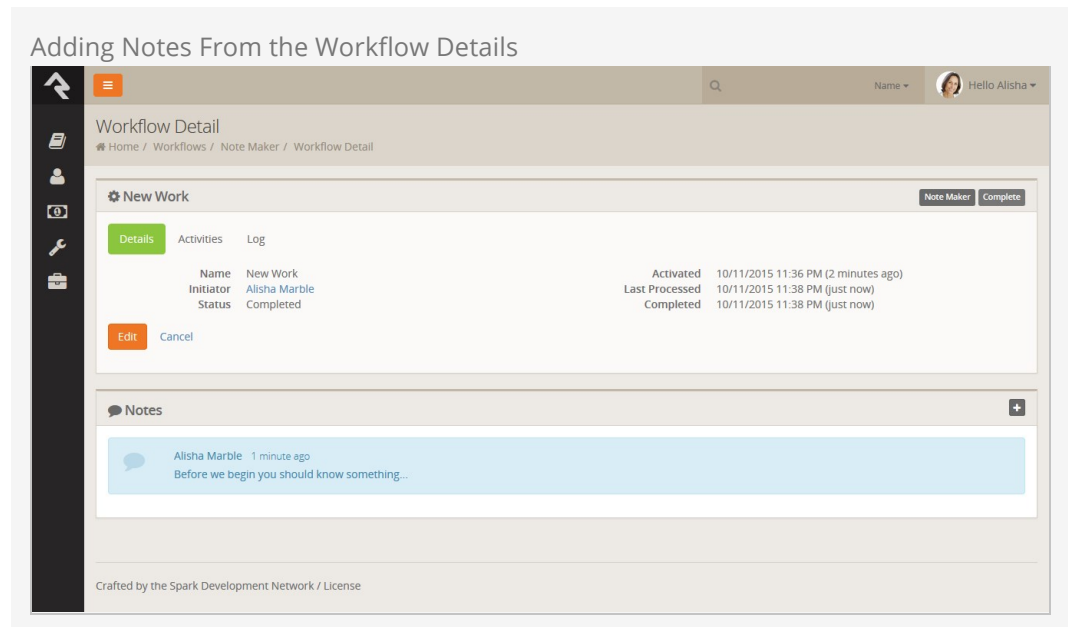
These two workflows act as a request ticketing system for your facilities and information technology teams. Using them allows your staff to report issues or request new services.

# 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.

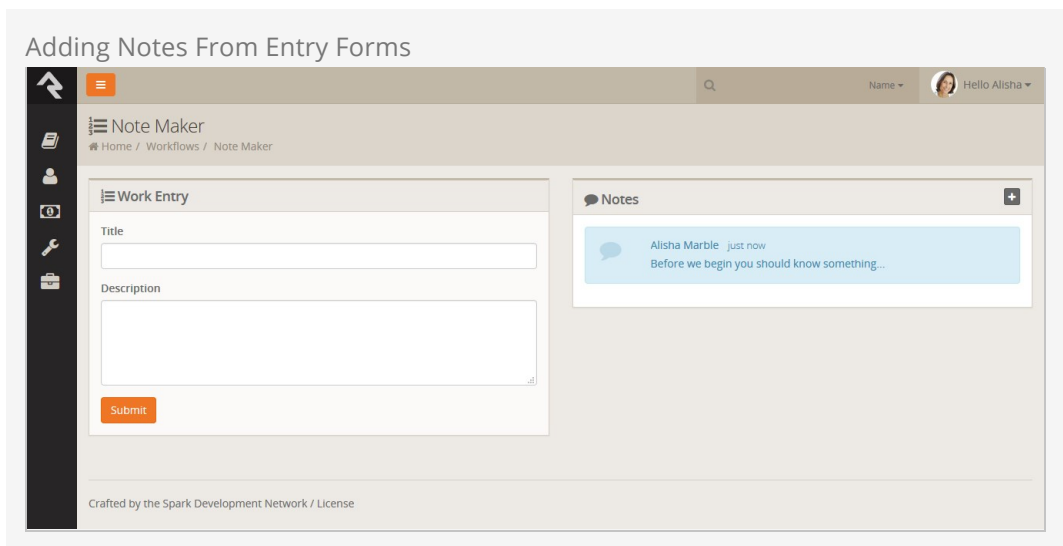
## Adding Notes From the Workflow Details

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 Entry Forms

Another way to add notes is from entry forms. You'll notice on the *User Entry 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 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 Entry Forms

**Add Info note to Workflow**

Name <sup>\*</sup>  
Add Info note to Workflow

Action is Completed on Success  
 Activity is Completed on Success

Action Type  
Add Workflow Note

Note <sup>?</sup> <sup>\*</sup> **1**  
Just want to note that something important happened.

Note Type <sup>?</sup> <sup>\*</sup> **2**  
Info

Is Alert <sup>?</sup> <sup>\*</sup> **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 <http://dotliquidmarkup.org/> and <http://rockrms.com/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 you 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 any key 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 merge field will always return the formatted value of the attribute. For example, if the *Requester* attribute was a Person type, the merge field would return the full name of the person.

### Mergefields Aren't Objects

One thing to keep in mind with attribute mergefields is that they are just text values, not objects. For example, you can't use a merge field like `{{ Workflow.Requester.NickName }}` since *Requester* only represents an attribute value and not an actual person object.

### Unformatted Attribute Values

There may be a time when you'd like to retrieve the data identifier 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* to the attribute syntax. For example, using a merge field of {{ Workflow | Attribute:'Requester','RawValue' }} would return a GUID since that is how the Person type attribute stores its value.

### 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 complex 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:'BackgroundCheckDate', '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 *Send Email*, or *Send Email Template* actions, and using an attribute for the recipient 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
ynUFCAPz4gIs1RshCy3dL7M!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

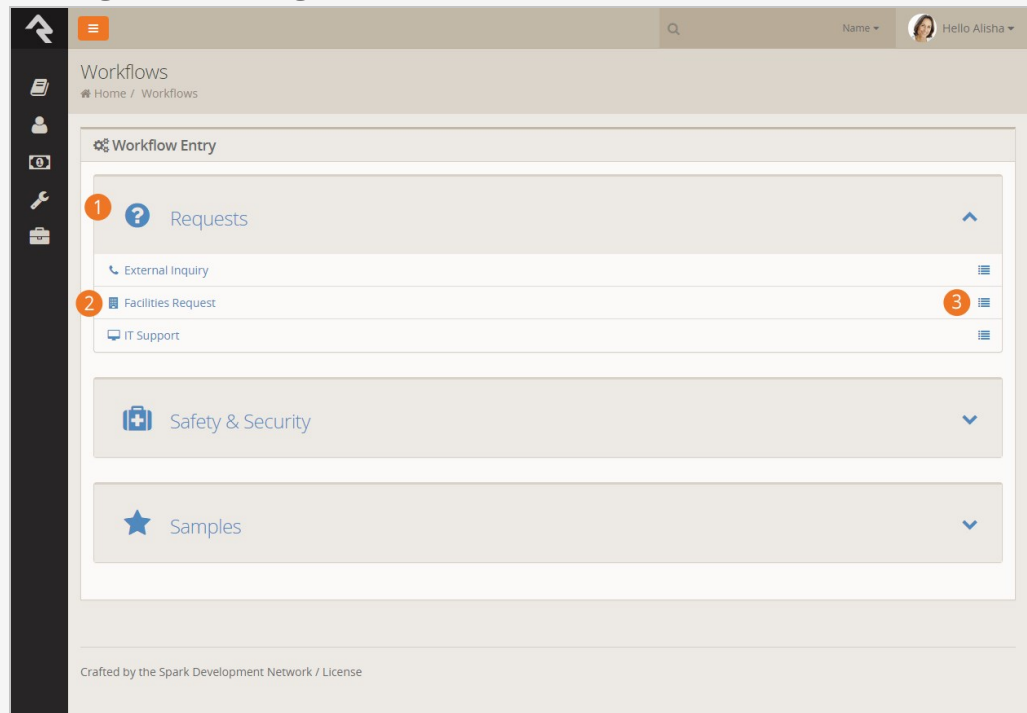
To be able to add or edit a workflow type, you'll need *Edit* access to the workflow configuration page ( [Admin Tools > General Settings > Workflow Configuration](#) ) and the *Workflow Type Detail* block on it. Currently, only the *Rock Administrator's* security role has access to edit workflow types.


## 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 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 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 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

My Workflows

Initiated by Me Assigned To Me Active Types All Types

1 Background Check 2 DISC Request External Inquiry 3 Position Approval

Person Data Error Photo Request

Position Approval Workflows

Name	Status	Active Activities
Receptionist	Active	HR Entry
Event Planner	Active	HR Entry
Operations Team Load	Active	HR Entry

50 500 5,000 3 Workflows

Crafted by the Spark Development Network / License

- 1 The user must be authorized to *View* the workflow type.
- 2 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 be authorized to *View* the activity
- 3 If viewing *Assigned to*, the user must be assigned to the activity
- 4 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 each of the different field types for attributes and a description of how its value is stored. When creating workflow actions that update an attribute value, make sure to reference this list so that your attribute values get set correctly.

Field Type	Stored Value
Account	A financial account's GUID
Accounts	A comma-delimited list of financial account GUIDs
Attribute	An attribute's GUID
Attribute Category	A category's GUID (the category should be an attribute category)
Binary File	A binary file's GUID
Binary File Type	A binary file type's GUID
Binary File Types	A comma-delimited list of binary file type GUIDs
Boolean	'True' or 'False'
Campus	A campus's GUID
Campuses	A comma-delimited list of campus GUIDs
Category	A category's GUID
Code Editor	The text of the code editor
Communication Template	A communication template's GUID
Comparison	'1' for Equal To, '2' for Not Equal To, '4' for Starts With, '8' for Contains, '16' for Does Not Contain, '32' for Is Blank, '64' for Is Not Blank, '128' for Greater Than, '256' for Greather Than or Equal To, '512' for Less Than, '1024' for Less Than Or Equal To, '2048' for Ends With, '4096' for Between, or '8192' for Regular

Field Type	Stored Value
	Expression
Component	The GUID of the Entity Type for the selected component
Components	A pipe-delimited list of Entity Type guids for the selected components
Connection Activity Type	A connection activity type's GUID
Connection Opportunity	A connection opportunity's GUID
Connection Request	A connection request's GUID
Connection State	'0' for Active, '1' for Inactive, '2' for Future Follow Up, or '3' for Connected
Connection Status	A connection status's GUID
Connection Type	A connection type's GUID
Connection Types	A comma-delimited list of connection type GUIDs
Content Channel	A content channel's GUID
Currency	A decimal value
Custom Checkbox List	A comma-delimited list of one or more of the values that are configured for the attribute
Custom Dropdown List	A comma-delimited list of one or more of the values that are configured for the attribute
Custom Radio List	One of the values that are configured for the attribute
Data View	A data view's GUID
Date	The selected date formatted as 'yyyy-MM-ddThh:mm:ss' or 'CURRENT:#' where # represents a day offset from the current day
Date Range	Two comma-delimited dates where first date is lower value, and second date is upper value formatted as 'yyyy-MM-ddThh:mm:ss,yyyy-MM-ddThh:mm:ss'
Date Time	The selected date formatted as 'yyyy-MM-ddThh:mm:ss' or 'CURRENT:#' where # represents a minute offset from the current time

Field Type	Stored Value
Day of Week	'0' for Sunday, '1' for Monday, '2' for Tuesday, '3' for Wednesday, '4' for Thursday, '5' for Friday, or '6' for Saturday
Days of Week	A comma-delimited list of Day of Week numbers (see previous)
Decimal	A decimal value
Decimal Range	Two comma-delimited decimal values where first number is lower value, and second number is upper value
Defined Type	A defined type'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)
Defined Value Range	Two comma-delimited GUID values where first GUID is the lower defined value GUID, and second GUID is the upper defined value GUID
Email	An email address
Encrypted Text	The text value encrypted using Rock's Encryption.EncryptString() static helper method
Entity	A pipe-delimited GUID and integer, where the GUID is an entity type's GUID, and the integer is the Id of the selected entity
Entity Type	An entity type's GUID
Enum	The value of the enumeration
Enums	A comma-delimited list of the selected enumeration integer values
Event Calendar	An event calendar's GUID
Event Item	An event item's GUID
Financial Gateway	A financial gateway's GUID
Group And Role	Three pipe-delimited GUID values where first GUID is a group type's guid, second GUID is a group's GUID, and third GUID is a group type role's GUID
Group	A group's GUID
Group Location Type	One of the configured group type's location type defined value's GUID
Group Role	A group type role's GUID
Group Type	A group type's GUID
Group Type Group	Two pipe-delimited GUID values where first GUID is a group type's guid, and second GUID is a group's GUID

Field Type	Stored Value
Group Types	A comma-delimited list of group type GUIDs
Integer	An integer value
Key Value List	A pipe-delimited list of two caret-delimited values where first is the selected key, and second is the selected value (ex: 'key1^value1   key2^value2   key3^value3'). If attribute is configured to use a defined type, the values should be the Ids of the selected defined values
Linked Page	Either a page's GUID, or two comma-delimited GUIDs where the first is the page's GUID, and second is the selected route's GUID
Location	A location's GUID
Markdown	The markdown text
Memo	The value of the textbox
Merge Template	A merge template's GUID
Metric Categories	A comma-delimited list of two pipe-delimited GUIDs where the first guid is a metric's GUID, and the second is a category's GUID (ex: MetricGUID1   CategoryGUID1, MetricGUID2   CategoryGUID2)
Metric Entity	Five pipe-delimited values where first is the metric's GUID, second is the entity's Id, third is a 'True' or 'False' indicating if metric should be gotten from page context, fourth is a 'True' or 'False' indicating if multiple values should be combined, and final value is a metric categories' guid (ex: 'MetricGuid   EntityId   False   False   CategoryGuid')
Note Type	A note type's GUID
Person Badges	A comma-delimited list of person badge GUIDs
Person	A person alias GUID
Phone Number	A formatted phone number
Remote Auths	A pipe-delimited list of entity type GUIDs (entity types should only be active authentication components that require remote authentication)
Schedules	A comma-delimited list of schedule GUIDS
Security Role	A security role (group) GUID
Site	A site's Id
Sliding Date Range	Five pipe-delimited values where first value is the mode ('All', 'Last', 'Current', 'DateRange', 'Previous', 'Next', or 'Upcoming'), second value is number of time units (may be blank depending on mode), third value is the time units ('Hour', 'Day', 'Week',

Field Type	Stored Value
	'Month', or 'Year'), fourth value is start date if mode is Date Range, fifth value is end data if mode is Date Range (ex: 'DateRange   5/22/2016 12:00:00 AM 5/24/2016 12:00:00 AM' or 'Previous 3 Week  ' )
System Email	A system email's GUID
Text	The value of the textbox
Time	A timespan value formatted as 'd.hh:mm:ss.fff'
Url Link	The text of the Url
Value List	A pipe-delimited list of values (ex: 'value1 value2 value3'). If attribute is configured to use a defined type, the values should be the ID of the selected defined values
Workflow Activity Type	An activity type's GUID
Workflow Attribute	The key of selected attribute
Workflow Text Or Attribute	The contents of Text field or the GUID of selected attribute
Workflow Type	A workflow type's GUID



# 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 you 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 Attribute 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 }}
```

# Text to Workflow

Text to Workflow is a creative and convenient way to interact with the members of your organization. Using Rock's powerful workflow capability, Text to Workflow allows you to launch a workflow in response to an incoming SMS text message. It's super easy to configure, and because of Rock's flexibility, you can do just about anything with your response. Think of Text to Workflow as a set of building blocks. What you decide to make depends on how creative you want to be. The possibilities here are endless.

There are two main components to Text to Workflow—the Text to Workflow type and the workflow itself—but additional components, such as content channels, can be utilized as well. The Text to Workflow type is where you specify your defined values: which number to text, which keyword to use, which workflow to trigger, etc. The workflow is where you define which actions will be performed in response to the text, and loop in any other Rock components you want to include in the process. This may sound a bit technical, but it's actually really easy to use, and the end result is super effective. Let's look at how you set it up.

## Text to Workflow Configuration

## Text to Workflow

Defined Type Detail

Home / General Settings / Defined Types / Defined Type Detail

**Text To Workflow**

Note Because this defined type is used by Rock, editing is not enabled.

Matches SMS phones and keywords to launch workflows of various types

**Help Text**  
The following merge fields are available for both the 'Name Template' and 'Workflow Attributes' attributes.

[show/hide fields](#)

Category: Workflow

**Attributes for Defined Type**

- Keyword Expression
- Workflow Type
- Workflow Name Template
- Workflow Attributes

**Text To Workflow**

	Value	Description	Keyword Expression	Workflow Type	
	+1 (444) 555-6789	Baptism signup request	bapt*		
	+1 (444) 555-6789	Summer Camp signup	cam*		
	+1 (444) 555-6789	help	*		

50 | 500 | 5,000 | 3 Defined Values

Text to Workflow is a defined type, so it's located at [Admin Tools > General Settings > Defined Types](#). Click the [+](#) button to create a new Text to Workflow defined value.

## Text to Workflow Defined Value

Defined Value id: 0

Add defined value for Text To Workflow

Value **1**

+1 (444) 555-6789

Description

Baptism signup request

Keyword Expression **2**

bapt\*

Workflow Type **3**

Samples

Workflow Name Template **4**

Workflow Attributes **5**

FromPerson	{{ FromPerson }}	
FromPhone	{{ FromPhone }}	
Message	{{ MessageBody }}	

Save Cancel

- 1 Value**

This is the phone number people will text the keyword to in order to launch the workflow. This should be in the form of +16235524427 for Twilio.
- 2 Keyword Expression**

This is the keyword people will text and that will launch the workflow. For instance, to match the keyword 'neighborhood' you could use the expression 'nei.\*' (without the quotes). If you leave this field blank, Rock will match any request. Note that the first matching keyword will be processed and the remaining defined values will not be considered. It's a good idea to create a general catchall expression at the bottom of your defined types list to process any items that didn't match up with previous keywords.
- 3 Workflow Type**

This is where you choose which workflow will be launched when the keyword expression is received.
- 4 Workflow Name Template**

This Lava-enabled field is where you can create a string to use as the workflow name. For example, Request From {{FromPerson.FullName}} ({{FromPhone}}) See the listing below of available Lava merge fields.
- 5 Workflow Attributes**

You can set Lava-enabled attributes and values for use in your Text to Workflow defined type, such as the name of the person who texted the keyword. See the listing below of available Lava merge fields.

The Workflow Attributes are Lava-enabled and give you a lot of useful options for how

to execute your text to workflow.

Available Lava Merge Fields

Merge Field	Description	Field Type
{{ FromPhone }}	The user's phone number, pulled from the inbound message from the SMS gateway	Text
{{ ToPhone }}	The SMS Gateway number where the message was sent	Text
{{ ReceivedDate }}	The date the message was received	Text
{{ ReceivedTime }}	The time the message was received	Text
{{ FromPerson }}	Text to Workflow uses the FromPhone attribute to lookup the first person with that phone number and assigns the value to FromPerson. If the number is used in more than one profile, Text to Workflow defaults to the first record of an adult with children. While the FromPerson attribute isn't required, it does save you from having to do the lookup manually or using another method.	Person
{{ MessageBody }}	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 express of a single result from the MatchedGroups array.

## Workflow Configuration

The next step is to set up your workflow. Again, the possibilities here are endless. It all depends on what you're trying to accomplish and how creative you want to be. For more information, see [Configuring a Workflow Type](#) above. When configuring your workflow, there are a few things to keep in mind.

### Responding

If you're able to send an immediate response—say, a simple text message—the easiest way is to use the workflow attribute "SMSResponse" when configuring your Text to Workflow defined values in the Text to Workflow Type Detail screen. If your response requires human action—say, a follow-up phone call from a small group leader—consider setting up a "SMS Send" action in your workflow to send the response.

### Getting the Person Attribute into Your Workflow

There are a couple of ways to identify the owner of the mobile number that sent the

incoming keyword prompt. One way is to query using the 'Run SQL' action. Or you could use {{ FromPerson }} in a Lava merge field if all you want is the owner's name as a text attribute. If you want a full Person Attribute, though, you can load the workflow initiator into your workflow attribute using the "Set Attribute to Initiator" action.

#### Sample Workflow Configuration

Workflow Configuration
Search
Hello Alisha

Home / General Settings / Workflow Configuration
Add Category

### Add Workflow Type

Details

Advanced Settings

Attributes Count: 8

Attribute	Description	Key	Field Type	Required
FromPhone		FromPhone	Text	<input type="checkbox"/>
ToPhone		ToPhone	Text	<input type="checkbox"/>
ReceivedDate		ReceivedDate	Text	<input type="checkbox"/>
ReceivedTime		ReceivedTime	Text	<input type="checkbox"/>
ReceivedDateTime		ReceivedDateTime	Text	<input type="checkbox"/>
FromPerson		FromPerson	Text	<input type="checkbox"/>
MessageBody		MessageBody	Text	<input type="checkbox"/>
SMSResponse		SMSResponse	Text	<input type="checkbox"/>

### Activities

Process Message

Name: Process Message  Active  Activated with Workflow

Description:

Attributes:

Actions

Set From PersonAttribute

Name: Set From PersonAttribute  Action is Completed on Success  Activity is Completed on Success

Action Type: Attribute Set to Initiator Overview  
Sets an attribute to the person who created the workflow (if known).

Person Attribute: FromPerson

Set Response

Name: Set Response  Action is Completed on Success  Activity is Completed on Success

Action Type: Attribute Set Value Overview  
Sets an attribute's value to the selected value.

Attribute: SMSResponse

Text Value: Thank you! or Attribute Value:

Save
Cancel

Crafted by the Spark Development Network / License



## Configuring Twilio

Text to Workflow uses Twilio for its SMS messaging capabilities. In order to receive messages from Twilio, you'll need to set up phone numbers using the Messaging Request URL:

```
http://{{YourExternalRockAddress}}/Webhooks/TextToWorkflowTwilio.ashx
```

To read more about Twilio and Rock, see the SMS in Detail section of the [Communicating with Rock](#) manual.

## Troubleshooting

Sometimes things don't go as planned. Mistakes and errors happen. Here are some common configuration issues that can happen, as well as some things to keep in mind as you set up and use Text to Workflow.

### Error Messages

If an error occurs, Text to Workflow will send one of the following messages to the sender:

- **No response could be provided for this keyword** – the texted keyword matched a defined value, but no workflow was configured.
- **The keyword you provided was not valid** – No defined value was found the keyword.
- **This keyword is no longer valid** – A defined value was matched to the keyword and a workflow was defined, but the workflow type no longer exists.

### SMSResponse

If you send a text and don't receive a response, the most likely cause is a nonexistent or empty SMSResponse workflow attribute. Keep in mind that even if no keyword is matched, Text to Workflow will send back an error message. If you get no response it means the workflow was called but no reply was generated. If this happens, check that your workflow has the SMSResponse attribute, and the attribute has a value.

### Industry Keywords

Some keywords are reserved for industry use, including: STOP, STOPALL, UNSUBSCRIBE, CANCEL, END, QUIT, HELP and INFO. For more information on these keywords and their approved uses, please see <https://www.twilio.com/help/faq/sms/>.

### Truncating Keywords

One way to avoid errors caused by keyword typos is to shorten the amount of the keyword required to trigger the workflow. For example, for the keyword "baptism", you could shorten the keyword defined value to "bapt\*". The asterisk (\*) acts as a catchall for the rest of the word. As long as that much of the keyword is sent, the Text to Workflow process will be launched.

### Keyword Order

Rock processes keywords from top to bottom, as listed in the Text to Workflow Defined Types Detail screen. So the order of the keywords is important. If you have a catchall

keyword (such as a simple “\*”) listed above more specific keywords, all incoming Text to Workflow calls will be processed according to the “\*” keyword. To make sure this doesn’t happen, move any catchall keywords to the bottom of the list. Then if the incoming keyword doesn’t match any previous keywords, it will be processed by the catchall keyword.