

## Example 1

This example produces a text-only HTML report that describes a Program that is specified when the report is run. Because the Program is specified as a parameter, this example report will be able to describe any Program.

---

### Dashboard: Simple Report with out Charts

---

## ISO 27001 Assessment

### Program key information

**Program Name:** *ISO 27001 Assessment*

### Program Description:

**Program Owner:** *mphelps*

**Program Status:** *Started*

**Program Workflow:** *Agiliance Assessments*

**Survey Taking:** *Control Assessments*

In this sample output, the name, description, and properties, such as owner and status, reflect the run-time Program parameter selected.

### Creating a Template in RiskVision

The first step is to create the template and open the editor to begin editing the HTML template.

1. Go to Analytics > **Dashboards and Reports**. On the landing page, click **New Dashboard**.
2. In the **New Dashboard or Report** dialog, enter a name for your sample custom report, such as My Example 1. Enter a description, header, and footer, or skip those fields, since they are optional and not the focus of this example.
3. For the layout, select **HTML Report**, and click **New**.
4. The RiskVision editor appears. Either enter or copy-and-paste the dashboard template content into the editor.

### Creating an HTML template

The report template is a straightforward HTML file, with a head section and a body section, and some references to the Velocity template language. Lines that start with '#' are Velocity directives, and a number of Velocity variables are pre-defined. Within your HTML markup, you can refer to Velocity variables by name, preceded with a '\$':

The first step is a Velocity directive to create a variable called \$program.

```
#set($program = $dashboardParameters.getProgram("program1", "Program"))
```

This line uses the predefined Velocity variable \$dashboardParameters. This variable acts like an object in that it has a number of methods that act upon it. In this case, the code invokes the method \$dashboardParameters.getProgram(). When the report is run, RiskVision will detect statements such as this one and build a set of parameters for the report. This statement requests a parameter of type Program which can be called 'program1' in the context of the report template. Because we're using the #set directive, however, the Velocity reference to the Program parameter will be the variable \$program. Notice that we did not have to #set \$dashboardParameters. That variable, and a few others, are pre-defined in the RiskVision report context.

It is convenient to establish report parameters first, but it is not required. We could embed the getProgram() call in the HTML itself, for example. Setting up variables that refer to parameters separately from the HTML code makes it easier to maintain the HTML template in the future.

Next, create a familiar HTML document that looks something like this:

```
#set($program = $dashboardParameters.getProgram("program1", "Program"))
```

>

---

**///Program Name will go here///**

///Program Description///

///Display selected Program properties///

To display the program name, invoke the `getName` method on the program variable, just as you invoked the `getProgram` method on the pre-defined `dashboardParameters` object.

## `$program.getName()`

To display the program description, invoke the `getDescription` method on the program.

```
$program.getDescription()
```

In a similar fashion, fetch various properties of the program using different methods. Label the results using ordinary HTML text. To differentiate, we have wrapped the labels in **bold** tags and wrapped the property values in *italic* markup. *The last two lines invoke methods (`getRaWorkflowTemplate` and `getSurveyTakingPreferences`) on the program object that return objects, rather than strings. We invoke the `getName()` method on these objects. The Velocity syntax allows us to run these method invocations together using the dot (`.`) operator. The 'program' object has a method called 'getSurveyTakingPreferences' that returns an object that has a method called 'getName' that returns a string.*

Program key information

Program Name: *`$program.getName()`*

Program Description: *`$program.getDescription()`*

Program Owner: *`$program.getAuthor()`*

Program Status: *`$program.getStatusDB()`*

Program Workflow: *`$program.getRaWorkflowTemplate().getName()`*

Survey Taking: *`$program.getSurveyTakingPreferences().getName()`*

### *Saving the New HTML Report*

*After the HTML template has been entered into the editor:*

- 1. Click OK to close the editor*
- 2. Click OK to save the report*

*The new report runs immediately. Because the template includes a call to `getProgram()`, RiskVision prompts for a program. The next time the report is run, it will use the program selected the first time it was run. To select a different program, click Parameters on the report display page.*

### *Example Template*

*The complete HTML template for this example is:*

```
#set($program = $dashboardParameters.getProgram("program1", "Program"))
```

>

---

**`$program.getName()`**

`$program.getDescription()`

**Program key information**

**Program Name:** `$program.getName()`

**Program Description:** `$program.getDescription()`

**Program Owner:** `$program.getAuthor()`

**Program Status:** `$program.getStatusDB()`

**Program Workflow:** `$program.getRaWorkflowTemplate().getName()`

**Survey Taking:** `$program.getSurveyTakingPreferences().getName()`

*Subsequent examples are more complicated and include charts to visualize the data, but they will build logically on this example.*

*The next example, [Example 2](#), adds a chart to this report.*