AcmeStudio Tutorial 1:
Getting to know AcmeStudio

Introduction

AcmeStudio is an architectural design environment that has been developed at Carnegie Mellon University. It provides a graphical interface that allows software engineers to create architectural designs in various styles, and to manipulate and analyze those designs. The terminology used in AcmeStudio is the same as the *Acme architecture description language*, which is an extensible textual notation for capturing architectures.

This lab serves as an introduction to AcmeStudio. After completing this assignment, you should be able to create and manipulate designs in AcmeStudio.

Download AcmeStudio and Tutorial

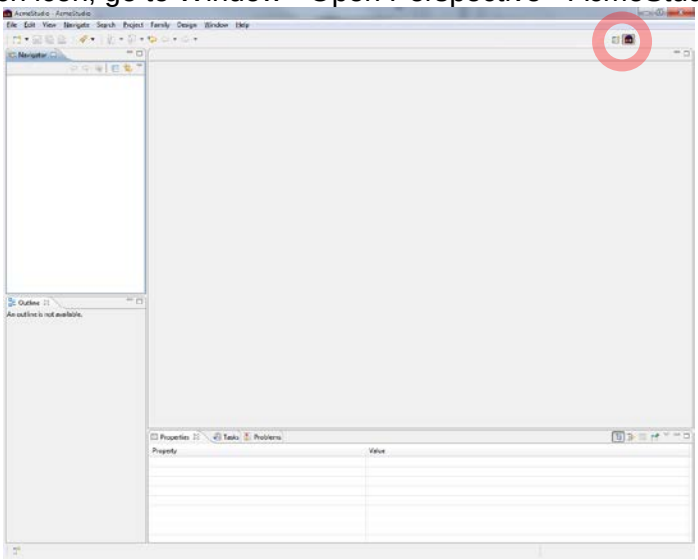
1. Download AcmeStudio. To get AcmeStudio as a standalone application, go to acme.able.cs.cmu.edu, click on Downloads, and sign in (create an account if you don't have one). Download the latest standalone application. The necessary components of Eclipse are included into the package, so that AcmeStudio can run separately from Eclipse. Extract the downloaded archive. To start the application, run *AcmeStudio.exe*.

2. Download and extract *AcmeLab1.zip* to a separate location. The archive contains this file and a lab project, *AcmeLab1.zip*.

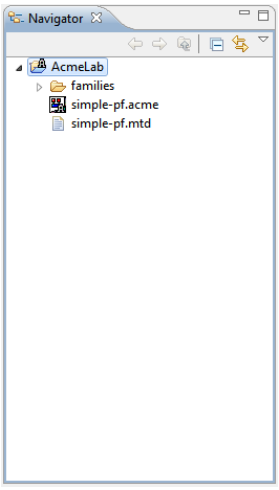
Navigate a pipe and filter design

1. Run AcmeStudio.

   You will see an empty workspace of AcmeStudio, as shown below. If the  icon in the top right is not selected, click it to bring the AcmeStudio perspective up. If there is no such icon, go to Window->Open Perspective->AcmeStudio.



2. Import the lab project.



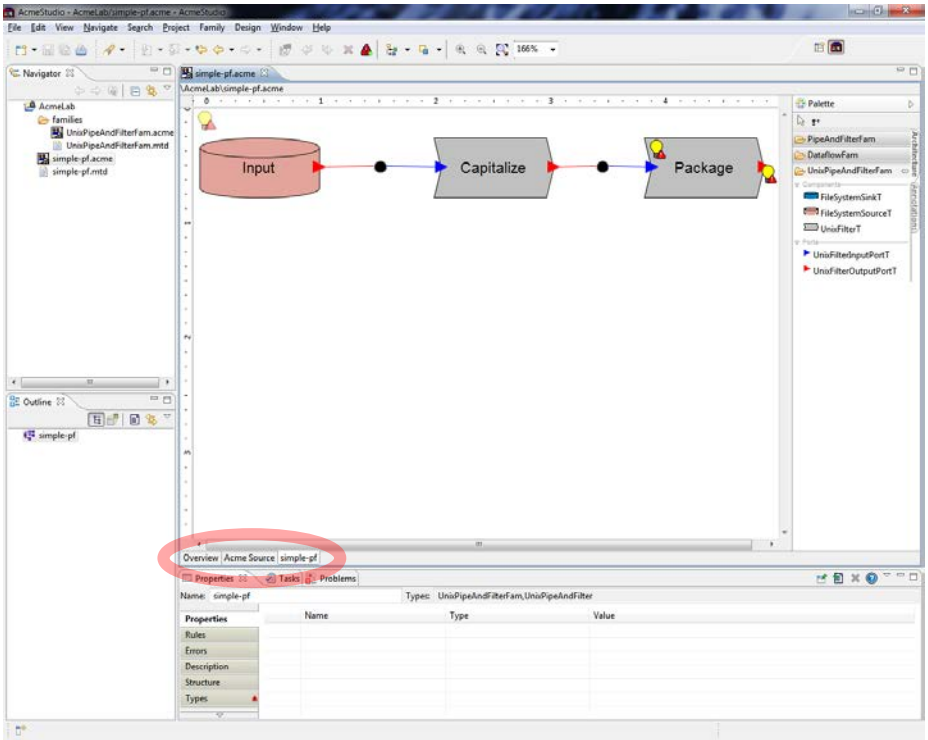
Open the *File->Import* dialog, select *Existing Project into Workspace* (under the General section), and click Next. Select the *Archive File* option, browse to *AcmeLab1.zip*, which was included in this tutorial, and click Finish. Now the current lab is placed in *<your user directory>/AcmeWorkspace*.

You should see the new project in the Navigator view of AcmeStudio. This project contains the file *simple-pf.acme* – the system you will be working on – and the *families* directory. Only *simple-pf.acme* should be changed in this lab.

3. Open the pipe and filter system.

Double-click *simple-pf.acme*. It contains an Acme architecture that uses the *UnixPipeAndFilterFam* Acme family[1], which captures the principles behind *Unix pipes*. If a dialog appears warning you that a resource provider is not configured, click OK.

This example architecture consists of an input file, a filter *Capitalize*, and a filter *Package*. Two pipe connectors link these three components into a pipeline.



[1] Acme families are formal descriptions of architectural styles. The main families are pipe and filter, event-based, call-return, and data-based families.

Note that … symbols may appear momentarily on the elements. This indicates that the architecture is being typechecked. A quick fix icon appears when one of the family rules was violated by the architecture, but AcmeStudio has a quick fix to correct the error. We will go into this later.

4. Explore the available editors.

   Opening a system brought up an editor consisting of three sub-editors. These editors are selected using the tabs circled in red in the picture above.

The *Acme Overview* editor contains an overview of the contents of the Acme file, including the systems and families defined in the file, in addition to the files that are imported.
The *Acme Source* editor contains the textual source of the architecture in Acme. Although it is sometimes convenient to edit the Acme source, you should not do so it in this lab.
The *Acme System* editor, opened by default, contains the architectural diagram. We will use this editor for this lab. A light bulb and a red icon appear at the top left because there is currently an error in this architecture. We will explain this in detail later.

5. Explore the views.

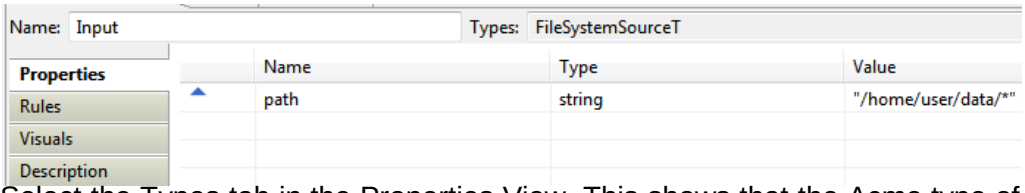
   The AcmeStudio design environment consists of four views[2]:
   - *System* is a graphical editor for architectural designs, placed in the center of the screen.
   - *Properties* show the attributes of the selected entity. This view is heavily used in AcmeStudio. If you remember to have seen something and cannot find it, it is probably in the Properties view.
   - *Navigator* shows projects, families, and systems in your workspace.
   - *Outline* shows entities in the current system or family.

   For more information on those windows, access the AcmeStudio user guide through Help->Help Contents->AcmeStudio User Guide or through buttons, scattered across the interface.

6. Explore the properties of components.

   Click on the Input component in the design. In the Properties view, you can use several tabs to inspect various aspects of the selected component. For example, the Rules tab displays the rules constraining this component. The Properties tab shows that the input file has a string property *path* equal to */home/user/data/**. The icon indicates that the property name and type are defined by the type.[3] Properties and rules are typically defined in component and connector types in Acme families.

[2] Note these "views" are not the same notion of view that we have used when talking about architectural "viewtypes" in class. Here the term "view" refers to a panel in the AcmeStudio editor.
[3] A icon near the value the value is also defined in the type and cannot be changed.



Select the Types tab in the Properties View. This shows that the Acme type of Input is *FileSystemSourceT*. It also shows the type hierarchy going back to the basic component type *DFlowCompT* of the basic data flow family.

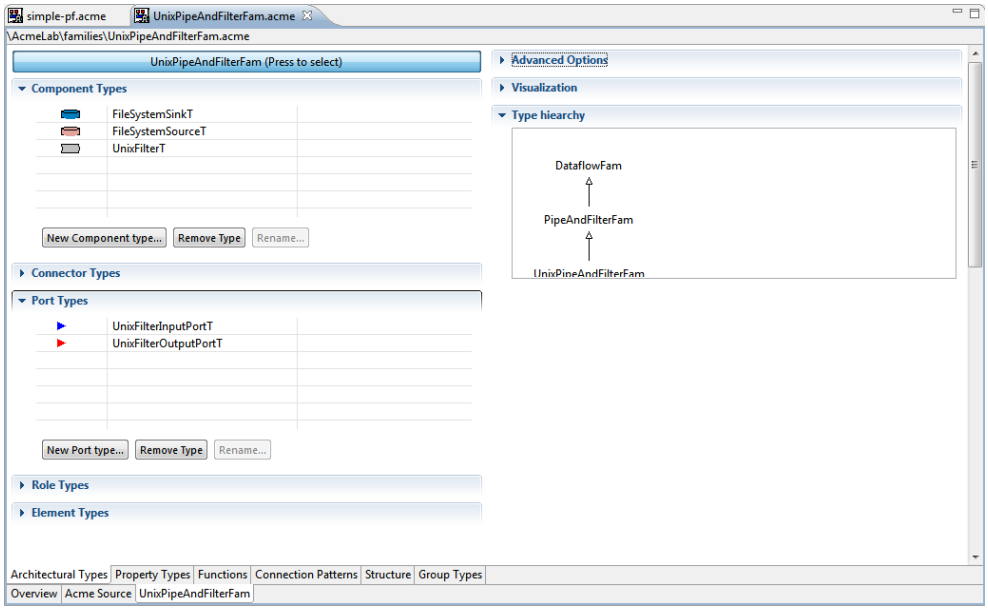
Try inspecting other elements displayed in the System editor. For example, select filters, pipes, or ports to examine their types and properties.

7. Explore the Unix pipe and filter family.

   The system you have been inspecting is an instance of a family that is provided in the lab package. You can find it by expanding the *families* folder in Navigator. Double-click *UnixPipeAndFilterFam.acme.*

   Acme families define types of components, connectors, ports, and roles. You can treat families as vocabularies of building blocks for actual architectures. This Unix pipe and filter family defines three types of components – sources, sinks, and filters – and two types of ports – Unix pipe input and output. This family inherits the pipe type and roles from the generic pipe and filter family. It is not possible to open this style in an editor. To examine this style, you may open Family->Inspect Type and select one of the types to see its properties, rules, etc.

Now you should now be familiar with loading existing Acme designs and examining them.



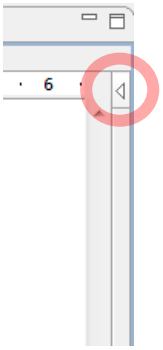
Modifying the design

In this part of the lab you will make changes to the simple pipe and filter design that we provided. These changes include adding components and pipes, as well as expanding the level of detail of a component by adding a representation.

Components and Connectors

Let us assume that in our system, all characters flowing through the pipeline should be capitalized, packaged, and compressed. So far the design accounts only for input, capitalization, and packaging. We will link a compression component and a sink to the design.

1. Add a new filter called Compress to the design.

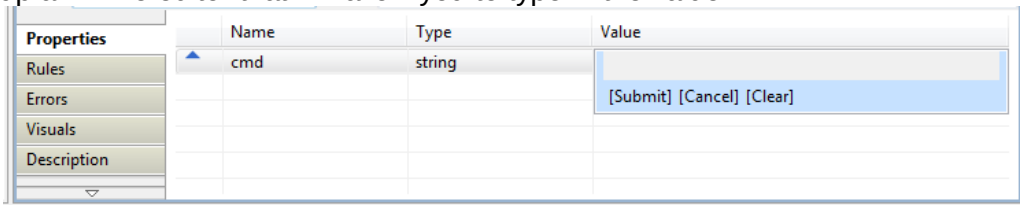
   First, select the *UnixFilter* type from the palette. By default, the palette appears on the right of the system editor and may be closed. To open the palette, click the arrow icon.

   Single-click to select the *UnixFilter* type in the palette. Then click on an empty space in the System view to place a filter. To rename it to Compress, click on the filter a second time (or press F2). Pressing return commits this change.

   Move the Compress filter to the right of the Package filter, making sure to leave enough space to add a pipe. You can zoom in or out of the diagram using the zoom tools in the toolbar.

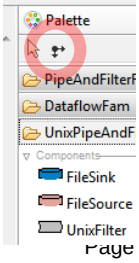
   Notice that two ports are automatically created for you when you construct a new Filter. These ports are defined in the Filter type. The ports are assigned a default position, but can be moved by dragging them around the component.

2. Look at the Properties View for the Compress component. Examine the properties. You will notice that the cmd property has <<No Value>>. Change it to be “zip”. This can be done in one of two ways: double-clicking on the property will bring up a dialog box, which will allow you to type the value. Alternatively, clicking in the value column of the Properties Tab. This will bring up an inline editor that will allow you to type in the value.

   To commit the value, press [Submit]. Canceling will return the value to the original value; Clear will set the value to <<No Value>>. Note that pressing return *will not* submit the value.

3. Add a pipe between Package and Compress.

   You can do in two ways. First is using the connect tool from the palette. Select it, click on the Package component (or its output port) and then click on Compress (or its input port). This

automatically created a corresponding pipe with appropriate roles. The connection tool is modal; you will need to press the ESC key to be able to select elements on the diagram again. This tool is enabled by connection patterns, stored in families.

The second way to connect components is more elaborate. Select a pipe in the palette, create it, and then manually connect the dangling roles to the appropriate component ports. Select the blue-colored input role near the arrowhead[4] and connect it to Compress. Analogously connect the red output role to the Package component's output port. If it attaches correctly, you should no longer be able to see the end of the role.

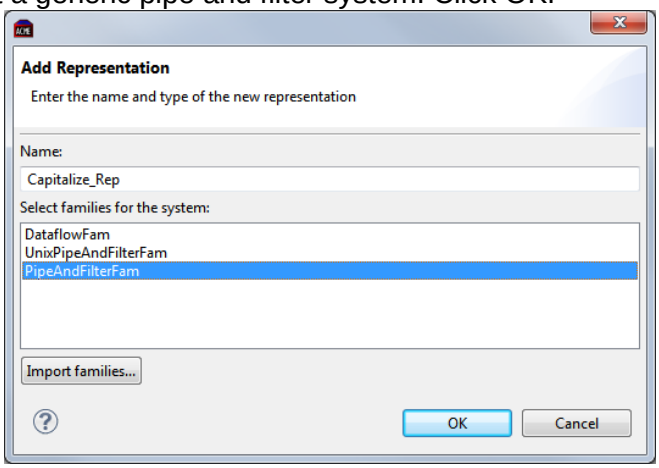
4. Similarly create a Sink component of type FileSink, place it to the right of Package, and connect them with a pipe. Change its *path* property to be "/home/user/processed.tar.zip".

Representations

In order to manage complexity in a design it is desirable to use hierarchy so that details can be hidden at higher levels of abstraction. Hierarchy is achieved in Acme with *representations* – nested systems that represent the internal structure of components and connectors. So far, we have created the architecture for a Unix style pipe and filter system. The Capitalize component represents a complex operation, which we can imagine being implemented by a Java program. To represent this, you will add a representation to this component *in the PipeAndFilterFam style*.[5]

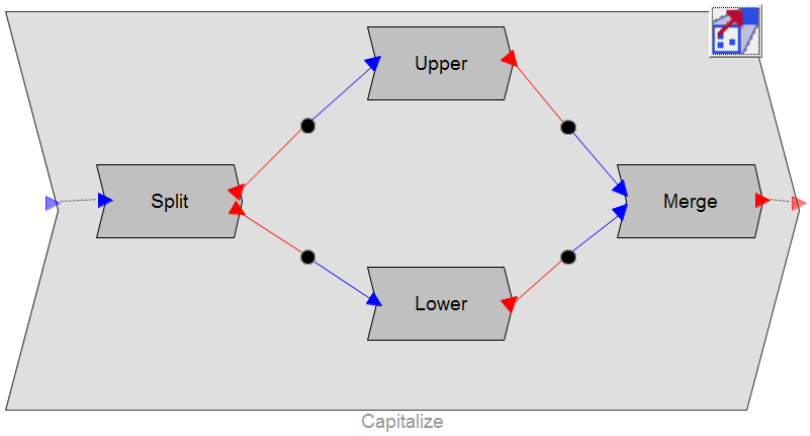
1. Right-click the Capitalize component and select Add Representation.

2. In the resulting dialog box, leave the name as Capitalize_Rep and assign it the family PipeAndFilterFam. Note that by default, the family selected will be the same as the family of the parent component. It is a important to select PipeAndFilterFam because the internals of Capitalize are no longer Unix pipes and filters, but a generic pipe and filter system. Click OK.



[4] This is because when the role is attached to a port, the port will obscure the role. Clicking and dragging on the line of the role allows you to manipulate (move, select, resize, rename) the role.
[5] The sub-architecture will be in this style because it is not a Unix pipe and filter system like its parent.

3. Add filters and pipes to the representation to make it look like the following screenshot. You will need to add extra ports to Split and Merge. Notice that the border of the representation moves as you place elements on the canvas.[6]



4. To connect the representation properly to the design, it is necessary to indicate which ports in the lower level correspond to ports in the upper level. In this design, the input port of Capitalize should correspond to the input port of split; the output port of Capitalize should correspond to the output port of merge.

   To do this, we must create *bindings*. To create a binding, select the Bind tool from the palette. Click on the outer port and then drag and click on the inner port. Once the binding has been created, a dotted line will appear between the outer port and inner port. Do this with the outward-facing ports of Split and Merge.

5. To return to the upper level, you can do any of the following:
   - Double-click on the parent overview to the left of the system (under the type palette).
   - Click the icon from the tool bar.
   - Click the icon in the upper right of the representation.
   - Right-click over the system and select *Navigate Up Representation*.

6. To return to the representation, you can either:
   - Double-click on the Capitalize component.
   - Select Capitalize, then go to the Representations tab in the Properties View and double click on the desired representation.

Working with rules

The Acme language provides a constraint language based on first-order predicate logic that allows rules to be defined. It is similar in spirit to UML's OCL. Acme rules are used to check whether an architectural model is well-formed[7]. AcmeStudio

[6] If you are unsatisfied by the automatic sizing of the component border, you can manually set its size by selecting the figure near the border and then resizing using the selection handles on one of the corners of the selection box.

[7] This may mean whatever an architect decides to check: presence of at least one database, correct load server balance, and so on.

checks these rules continuously. Typically rules are defined by the style designer, although you can also define your own rules. For this lab, we will just inspect some rules.

Acme allows the definition of two types of rules: *invariants*, violations of which are errors, and *heuristics*, violations of which lead to warnings.

1. Select one of filter components in your high-level system, e.g. Capitalize.

2. Go to the the Rules tab in the Properties View.

3. This component has a rule stating the legal types of ports that the component can have. This rule is defined in the UnixPipesAndFiltersFam. The label used in the Rule View is a short textual description of the rule.

   The ✓ icon next to the rule indicates that the rule is satisfied by the current design. To inspect the definition of this rule, double-click on the rule[8]. You will see a dialog box appear that has the first-order predicate logic definition of the rule.

4. Select the architectural design as a whole by clicking the mouse in an empty space on the diagram. By inspecting the Rules tab you will see that there are a number of system-wide constraints that must be satisfied by the design. If an *invariant* fails, the icon is placed next to the rule; if a *heuristic* fails, the icon appears.[9]

5. Detach a role from its port. Select the connector. Notice that the rule indicating that all roles should be attached now fails.

6. Reconnect the role and notice that the rule is rechecked, and ✓ should appear by the rule.

7. Remove any pipe by selecting and pressing *Delete*.[10] A quick fix icon will appear on both ports of the pipe-deprived components, because there is a rule stating that there must be at least one input and output port for Unix filters. Click on it (aim for the right bottom quadrant) and select Create a new connection... This invokes the connect tool, with which you recreate a pipe.

Further information

For further information about AcmeStudio, you can refer to the user manual, accessed through the *Help->Help Contents* menu or contextual help buttons. This gives information about AcmeStudio, as well as the Acme language. Context-sensitive help is available in some areas. Pressing F1 in the property view or editor will open this help, and pressing the icon where it appears.

Submitting your lab

Once you have completed all the instructions above, submit your lab:

---

[8] Alternatively, you may hover over the rule and a balloon pop-up will show the rule.

[9] … indicates that the rule is being evaluated, indicates that the rule has type errors in it, indicates that constraint evaluation has been turned off, indicates that the rule could not evaluated.

[10] Note that in Mac OS X, you need to press fn-Delete.

1. Save your work.
2. Rename the project to *AcmeLab1-<andrew-id>*, where <*andrew-id*> is your CMU Andrew ID. This can be done by selecting the project in the Navigator view and clicking File->Rename.
3. Export the project by selecting *File->Export*. Select zip file and then click Next. Make sure that all files in *AcmeLab1-<andrew-id>* are selected in the next page, and give the zip file the name *AcmeLab1-<andrew-id>.zip*. Click Finish.
4. Submit the zip file via the Digital Dropbox on blackboard. In the comment line, put “AcmeLab1”.

Appendix: Advanced topics

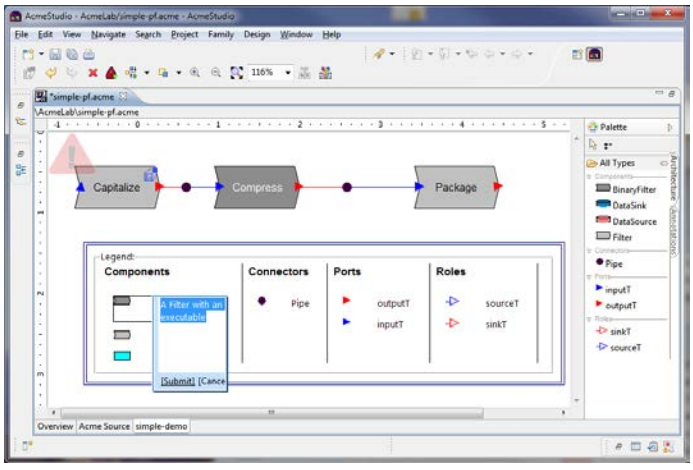
Exporting to Image

To include your diagrams as part of architecture documentation, you will need to paste images of the architecture into, for example, Microsoft Word. You can do this in AcmeStudio by exporting the diagram to either JPEG, PNG, BMP, SVG, or PDF image formats. To export, choose *File -> Export to Image* from the menu, and follow the directions. On Windows, you can also cut and paste the diagram into Microsoft Office as vector graphics. To do this, use the *Edit -> Copy System Image* action.

Legends

When developing architectural diagrams, it is always a good idea to have a legend describing what elements of the diagram mean. A legend can be automatically generated by AcmeStudio with the *View->Generate Legend* action. It generates a legend containing an entry for each of the types that are used in the diagram, and allows you to edit or remove the text describing each entry. To do this, you click twice on the text and enter the new text, or press delete to remove it. Clicking outside the editor causes the entry to be changed.

Legend has a number of useful visual properties. Select it and explore the Properties view. You may also move it around the diagram.



Annotations

When drawing an architectural diagram it is often needed to add annotations to the diagram that do not have any architectural significance, like notes. *Annotations* implement this function in AcmeStudio. The types of annotations that can be added are Rectangles, Ellipses, Rounded Rectangles, Labels, and Notes.

The tools for creating annotations are accessible from the Annotations section of the palette. Annotations are created in the same manner as other elements in the diagram. You change the colors, fonts, and styles of annotations using the Properties view.

Annotations can be added to the legend by dragging them from the diagram to the legend. The Annotation will remain in place in the diagram and an entry will be added to the legend.