

Developing SAS® Studio Repositories

Marie Dexter, Swapnil Ghan, Michael Monaco, and Amy Peters, SAS Institute Inc., Cary, NC

ABSTRACT

This paper covers developing SAS® Studio repositories. SAS Studio introduced a new way of developing custom tasks using an XML markup specification and the Apache Velocity templating language. SAS Studio repositories build on this framework and provide a flexible way to package custom tasks and snippets. After tasks and snippets are packaged into a repository, they can be shared with users inside your organization or outside your organization. This paper uses several examples to help you create your first repository.

INTRODUCTION

Do you have custom tasks and snippets written by SAS Studio users at your site? Are you looking for an easy way to share these tasks and snippets?

SAS Studio repositories are an easy way to share tasks and snippets with users inside and outside your organization. Currently, you might be sharing tasks by sending the CTM files through email. This method quickly results in out-of-date content. You might also be sharing tasks and snippets by using a network location. However, the SAS Studio user still has to upload these files into their SAS Studio session before they can access them.

Repositories provide a solution to both issues. Maintenance is easier because all the source files are in one place. To view the contents of a repository, the SAS Studio user simply sets a preference in their SAS Studio session. After this preference is set, any tasks and snippets in the repository are visible from the **Task and Utilities** and **Snippets** sections in the SAS Studio navigation pane. If you (the developer of the repository) add content to the repository, the SAS Studio user simply needs to refresh the content of the **Task and Utilities** and **Snippets** sections to view your changes.

This paper focuses on developing a repository. The paper shows several examples that are available when you download the Repository Toolkit Appliance. The Repository Toolkit Appliance is a tool that you can use to develop your repositories. For more information about how to download and use the Repository Toolkit Appliance, see *SAS Studio Developer's Guide: Creating Repositories for Tasks and Snippets*.



The specifics of how you deploy a repository are not covered in this paper. You can deploy repositories using web hosting services, source management services such as GitHub, or your own web content system. The process for deploying repositories depends on the setup at your site.

VIEWING REPOSITORIES IN SAS STUDIO

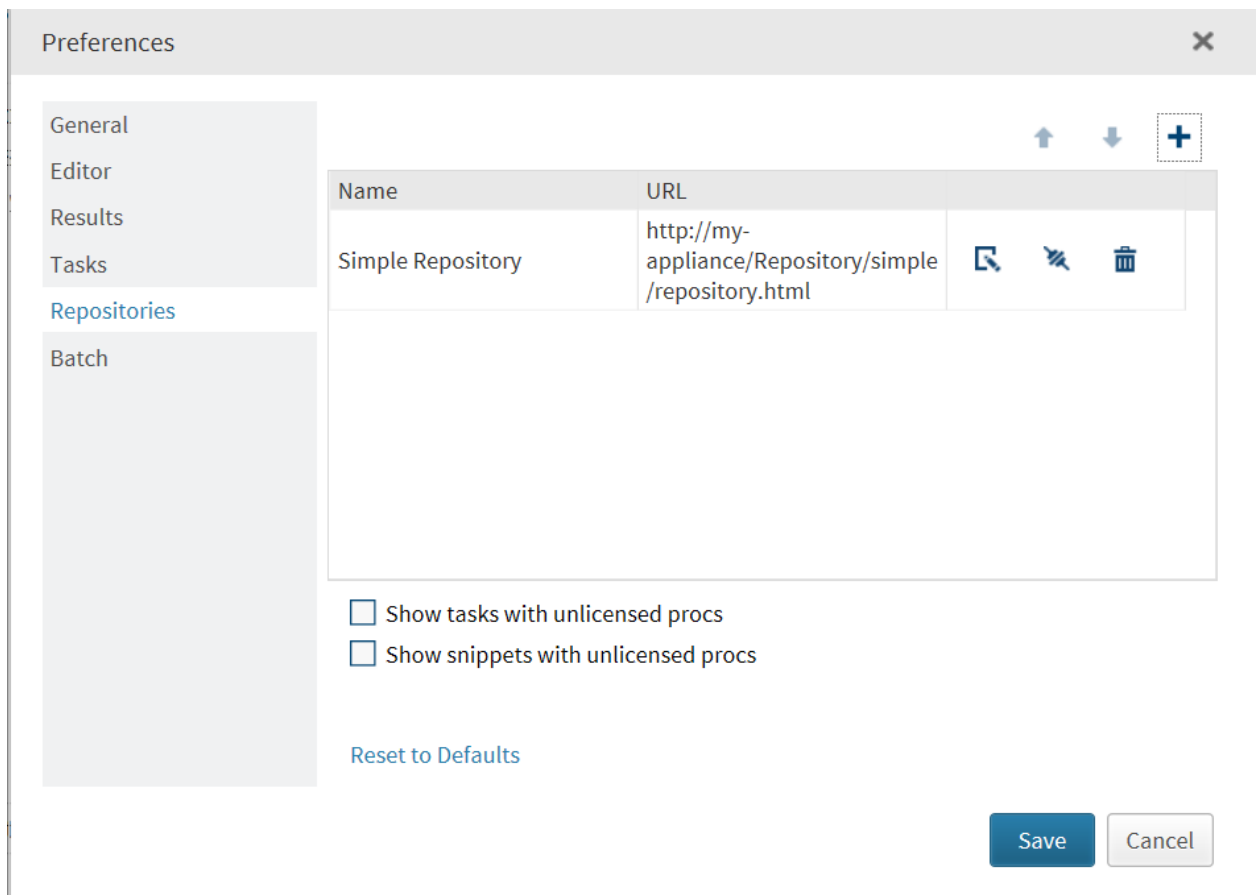
After a SAS Studio user sets a preference to connect to a repository, the experience of accessing the tasks and snippets is like accessing any other SAS Studio tasks and snippets.

To set a preference to connect to a repository:



1. In SAS Studio, click  and select **Preferences**. The Preferences window appears.
2. In the navigation pane, select **Repositories**.
3. In the Repositories pane, click . The Add Repository window appears.
4. In the **Name** box, enter the name of the repository. For this example, the name is *Simple Repository*. (The Simple Repository is one of the samples that is included in the Repository Toolkit Appliance.)

5. In the **URL** box, enter the URL for the repository. For this example, the URL is <http://my-appliance/Repository/simple/repository.html>.
6. Click **Add** to return to the Preferences window.

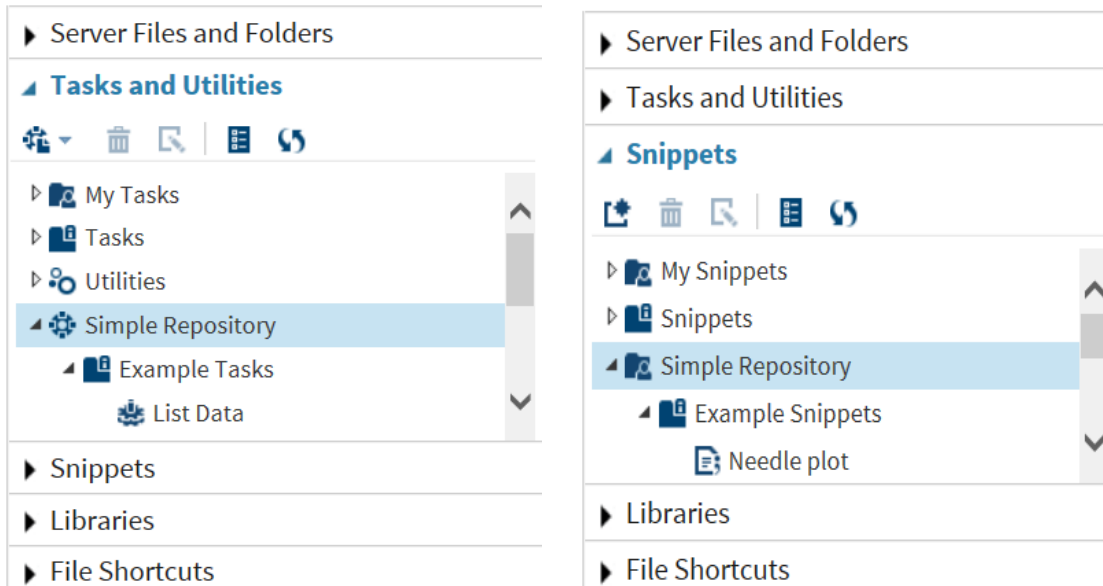


Display 1: Adding the Simple Repository to Your Preferences


7. To close the Preferences window, click **Save**.

Now open the **Tasks and Utilities** and **Snippets** sections in the navigation pane.

In the **Tasks and Utilities** section, the Simple Repository contains one task called List Data. In the **Snippets** section, the Simple Repository contains one snippet called Needle plot.



Display 2: Simple Repository in Tasks and Utilities and Snippets Sections

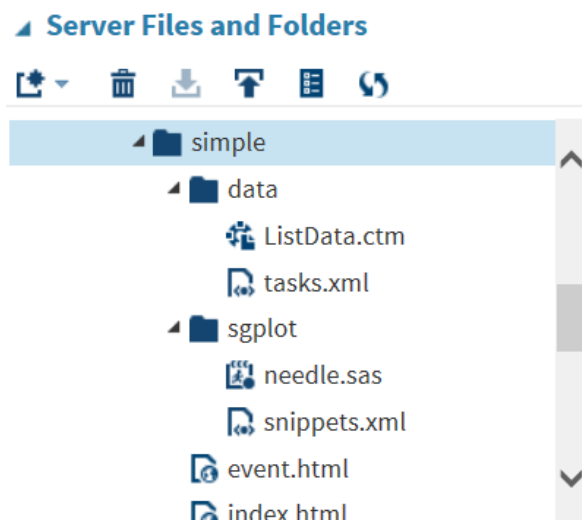
That's it! Now the SAS Studio user can access this task and snippet from their SAS Studio session. If you update the contents of the repository, the content in SAS Studio is updated when the user clicks . After refreshing the contents of the **Tasks and Utilities** or **Snippets** section, the SAS Studio user will have your latest and greatest content.

TIP: If you are an administrator, you can create global repositories. The tasks and snippets from global repositories are automatically available from the **Tasks and Utilities** and **Snippets** sections when a user opens SAS Studio. For more information, see the "Global Repositories" section of this paper.

REPOSITORY BASICS

Repositories require a minimum of four files: a repository.xml file, a tasks.xml file, a snippets.xml file, and a repository.html file. Let's examine each file more closely.

Here are the files in the Simple Repository in SAS Studio:



Display 3: Contents of the Simple Repository

REPOSITORY.XML FILE

The repository.xml file defines the repository. For the Simple Repository, here is the code in the repository.xml file:

```
<?xml version="1.0" encoding="UTF-8"?>

<Repository label="Example SAS Studio Repository" version="1.0">
  <Tasks>
    <Category label="Example Tasks">
      <Location uri="./data/tasks.xml"/>
    </Category>
  </Tasks>
  <Snippets>
    <Category label="Example Snippets">
      <Location uri="./sgplot/snippets.xml"/>
    </Category>
  </Snippets>
</Repository>
```

Here is a brief explanation of this code:

1. In the `Repository` element, the `label` and `version` parameters are required.
2. Use the `Tasks` element to group categories of tasks. You must have at least one category. The category appears under the name of the repository in the **Tasks and Utilities** section of the navigation pane. (In Display 2, this is the Example Tasks folder.) Specify the name of the category by using the `label` parameter.

Each `Category` element must contain one `Location` element. You specify the location of the tasks.xml file using the `uri` parameter. This path can be relative or absolute. A unique tasks.xml file is required for each task category.

3. The `Snippets` element works the same as the `Tasks` element. The category appears under the name of the repository in the **Snippets** section of the navigation pane. (In Display 2, this is the Example Snippets folder.)

TASKS.XML FILE

The tasks.xml file lists the tasks in a category. You need a tasks.xml file for each task category that you create. As you saw in the repository.xml file, the Simple Repository contains only one task category called Example Tasks. As a result, you need only one tasks.xml file.

Here is the code for the tasks.xml file for the Simple Repository:

```
<?xml version="1.0" encoding="UTF-8"?>

<Tasks>
  <Task>
    <Name label="List Data"/>
    <Template uri="./ListData.ctm"/>
  </Task>
</Tasks>
```

1. Use the `Tasks` element to group all tasks into the category (which you defined in the `repository.xml` file). This element must contain one or more `Task` elements. In this example, the Example Tasks category contains only one task (the List Data task), so only one `Task` element is needed in the `tasks.xml` file.
2. Use the `Task` element to define the task. Both the `Name` and `Template` elements are required.
 - o The `Name` element specifies the name that appears in SAS Studio.
 - o The `Template` element uses the required `uri` parameter to specify the location of the task file (either a CTM or CTK file). The location of the CTM or CTK file can be relative or absolute. In the Simple Repository, the `ListData.ctm` file is saved with the `tasks.xml` file in the `data` folder. (See Display 3.)

SNIPPETS.XML FILE

The `snippets.xml` file lists the snippets in a category. You need a `snippets.xml` file for each snippet category that you create. As you saw in the `repository.xml` file, the Simple Repository contains only one snippet category called Example Snippets. As a result, you need only one `snippets.xml` file.

Here is the code for the `snippets.xml` file for the Simple Repository:

```
<?xml version="1.0" encoding="UTF-8"?>

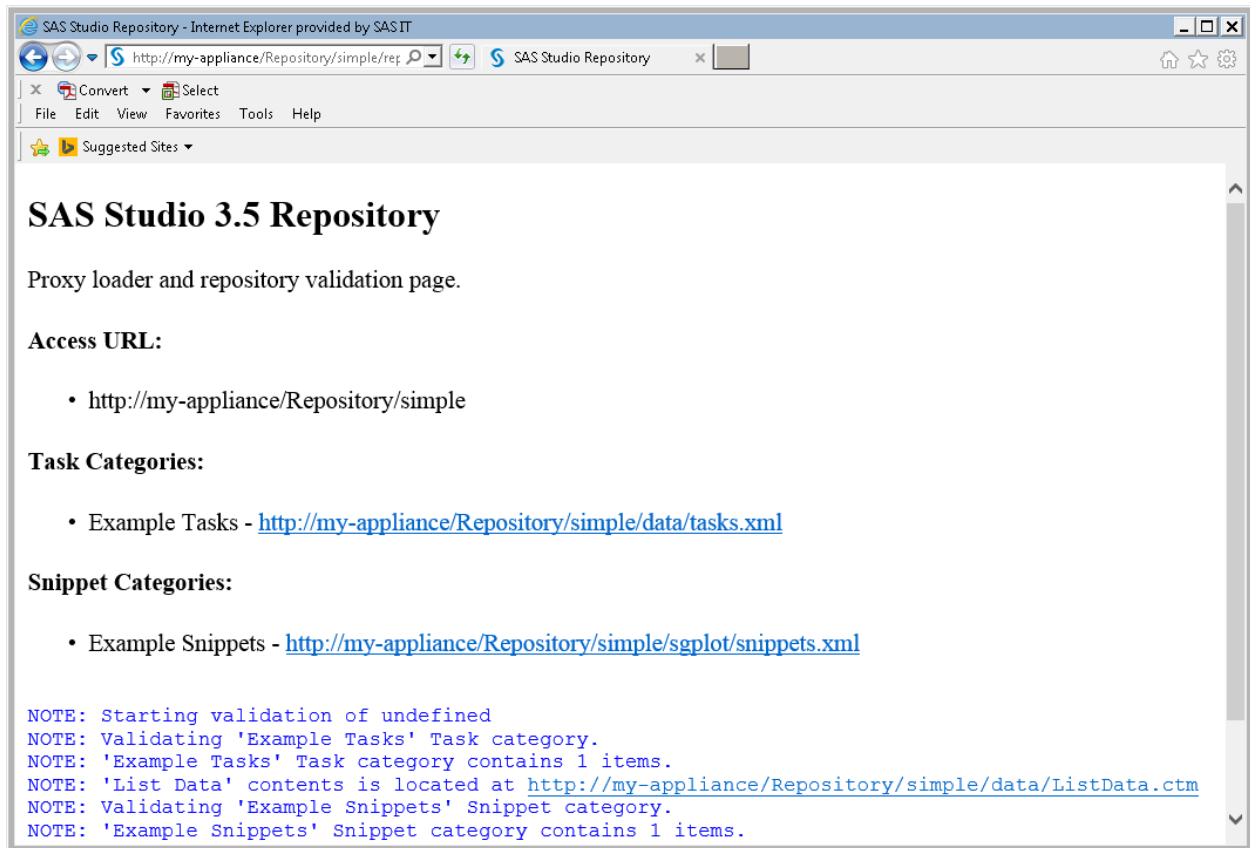
<Snippets>
  <Snippet>
    <Name label="Needle plot"/>
    <Template uri="./needle.sas"/>
  </Snippet>
</Snippets>
```

1. Use the `Snippets` element to group all snippets into a single category (which you defined in the `repository.xml` file). This element must contain one or more `Snippet` elements. In this example, the Example Snippets category contains only one snippet (the Needle Plot snippet), so only one `Snippet` element is needed in the `snippets.xml` file.
2. Use the `Snippet` element to define each snippet in the category. Both the `Name` and `Template` elements are required.
 - o The `Name` element specifies the name of the snippets by using the required `label` attribute.
 - o The `Template` element specifies the location of the SAS file by using the required `uri` attribute. The location of the SAS file can be relative or absolute. In the Simple Repository, the `needle.sas` file is saved with the `snippets.xml` file in the `sgplot` directory. (See Display 3.)

REPOSITORY.HTML FILE

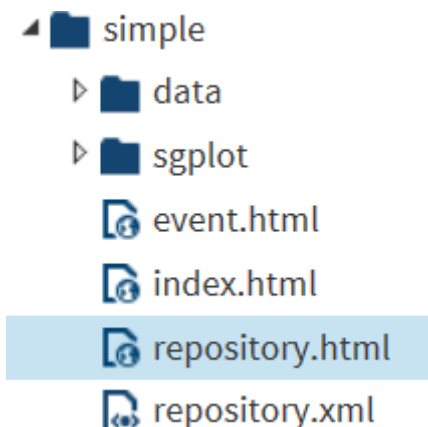
The `repository.html` file enables you to test and validate that everything is working correctly. Any warnings or errors generated when trying to load the repository appear on the `repository.html` page.

To view the `repository.html` file for the Simple Repository, open a web browser and enter the URL for the `repository.html` file. For the Simple Repository that is part of the Repository Toolkit appliance, this URL is <http://my-appliance/Repository/simple/repository.html>, which is shown in Display 4.



Display 4: Repository.html File for the Simple Repository

TIP: When creating your own repository, simply copy the repository.html file from one of the sample repositories in the Repository Toolkit Appliance. You do not need to edit this file for it to work with your repository. Save the repository.html file in the same location as the repository.xml file, as shown in Display 5.



Display 5: Location of Repository.html File

That's it! You now have all you need to create your very first repository.

ADDING THE BELLS AND WHISTLES

The Simple Repository showed you how to create a repository with one task and one snippet. Obviously, you can create repositories with multiple tasks and snippets. You can create a repository that just contains tasks, a repository that just contains snippets, or a repository that contains both tasks and snippets.

The `Repository`, `Task`, and `Snippet` elements also support additional features such as icons, documentation, and data files. For these features, let's look at some additional examples. All of the following repositories are available as part of the Repository Toolkit Appliance.

SNIPPETS REPOSITORY

The Snippets Repository contains several snippets organized into three categories. This repository is a good place to start if you need to provide SAS samples or fragments of SAS code to users in your organization.

This repository demonstrates these features:

- defining multiple snippets and snippet categories
- specifying an icon for a repository and an individual snippet
- referring to documentation that is saved with the repository and documentation that is external to the repository
- referencing data files in the repository

Specifying Multiple Snippets and Categories

From the repository.xml file, you can see that this repository contains three categories of snippets: SGPlot Samples, Import Snippets, and DBMS Snippets.

```
<?xml version="1.0" encoding="UTF-8"?>
<Repository label="Example SAS Studio Repository" version="1.0">
  <Icon uri="http://icons.iconarchive.com/icons/3xhumed/cryonic
    folder/128/Folder-Blank-sample-icon.png"/>
  <Documentation uri="./sample.pdf" label="Embedded Repository
    Documentation (via PDF)"/>
  <Documentation uri="http://support.sas.com/documentation/cdl/en/
    proc/67916/HTML/default/viewer.htm#p19bcwkxesgd19n1ciwhc8195s6s.htm"
    label="Procedures Guide"/>

  <Snippets>
    <Category label="SGPlot Samples">
      <Documentation uri="http://support.sas.com/documentation/cdl/en/
        grstatproc/67909/HTML/default/
        viewer.htm#p073bl97jzadkmn15lhq58yiy2uh.htm"
        label="SAS(R) 9.4 ODS Graphics: Procedures Guide,
        Fifth Edition"/>
      <Documentation uri="http://support.sas.com/documentation/cdl/en/
        proc/67916/HTML/default/
        viewer.htm#p19bcwkxesgd19n1ciwhc8195s6s.htm"
        label="Procedures Guide"/>
      <Location uri="./sgplot/snippets.xml"/>
    </Category>
  </Snippets>
</Repository>
```

```

<Category label="Import Snippets">
  <Documentation uri="https://support.sas.com/documentation/cdl/en/
    proc/67916/HTML/default/
    viewer.htm#n18jyszn33umngn14czw2qfw7thc.htm"
    label="PROC IMPORT Statement"/>
  <Location uri="./import/snippets.xml"/>
</Category>

<Category label="DBMS Snippets">
  <Documentation uri="https://support.sas.com/documentation/cdl/en/
    acreldb/68028/HTML/default/
    viewer.htm#titlepage.htm"
    label="SAS/ACCESS® 9.4 for Relational Databases"/>
  <Location uri="./access/snippets.xml"/>
</Category>
</Snippets>

</Repository>

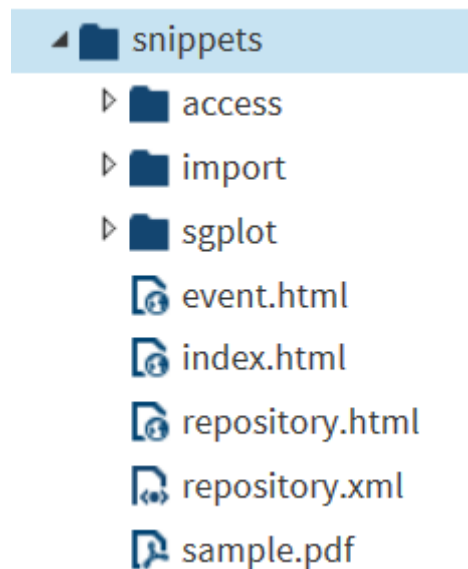
```

Note: This repository does not contain any tasks. Repositories can contain tasks, snippets, or both. You need only one task or one snippet to create a repository.

The files for each category are stored in a separate folder:

- access for the DBMS Snippets category
- import for the Import Snippets category
- sgplot for the SGPLOT Samples category

Here is the file structure for this repository:



Display 6: File Structure of the Snippets Repository

As you can see, using a separate folder for each category keeps your files organized and easy to manage.


Specifying an Icon

In SAS Studio, icons are used to help quickly identify a task, snippet, or repository. You can specify custom icons to identify each task, snippet, and repository.

Here is an example of the `Icon` element in the `repository.xml` for the Snippets repository:


```
<?xml version="1.0" encoding="UTF-8"?>

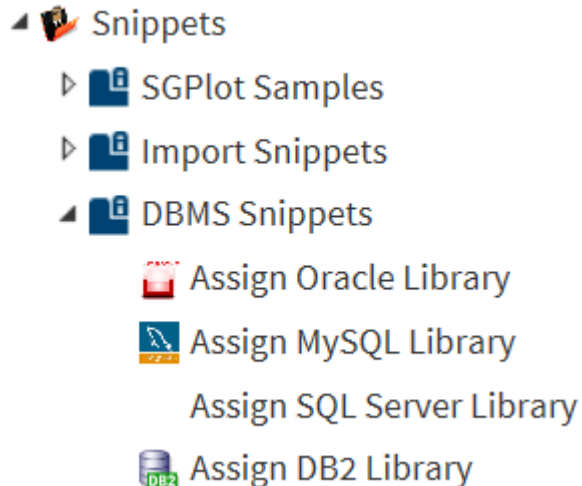
<Repository label="Example SAS Studio Repositories" version="1.0">
  <Icon class="" uri="./repository.png"/>
  ...
</Repository>
```

In the `Icon` element, the path to the icon can be absolute or relative. In this example, the `repository.png` file is saved in the same directory as the `repository.xml` file, so the path is relative. When you view the repository in SAS Studio, you see  next to Snippet Repository in the **Snippets** sections of the navigation pane. (See Display 7.)

You can also use the `Icon` element in the `tasks.xml` and `snippets.xml` files to specify icons for individual tasks and snippets. For example, here is part the `snippets.xml` file for the DBMS Snippets category. The highlighted code shows the `Icon` element for the Assign Oracle Library snippet.

```
<Snippets>
  <Snippet>
    <Name label="Assign Oracle Library"/>
    <Documentation uri="https://support.sas.com/documentation/cdl/en/
      acreldb/68028/HTML/default/viewer.htm#
      plujrhdoelp743n12awcf7mwyg81.htm"
      label="SAS/ACCESS Interface to Oracle"/>
    <Template uri="./oracle.sas"/>
    <Icon uri="http://rocketdock.com/images/screenshots/oracle.png"/>
  </Snippet>
  ...
</Snippets>
```

In this case, the icon is available from an external site, so when SAS Studio loads the repository, it pulls the image from this external site. As a result, the  icon appears next to Assign Oracle Library. (See Display 7.)



Display 7: Icons for the Snippets Repository

Including Documentation in Your Repository

To help users work with the contents of a repository, you might want to provide custom documentation or links to external documentation (such as SAS documentation on support.sas.com). You can provide documentation for a repository, task, or snippet by using the `Documentation` element.

Here is an example of `Documentation` elements in the `repository.xml` file for the Snippets Repository:

```
<?xml version="1.0" encoding="UTF-8"?>

<Repository label="Example SAS Studio Repository" version="1.0">

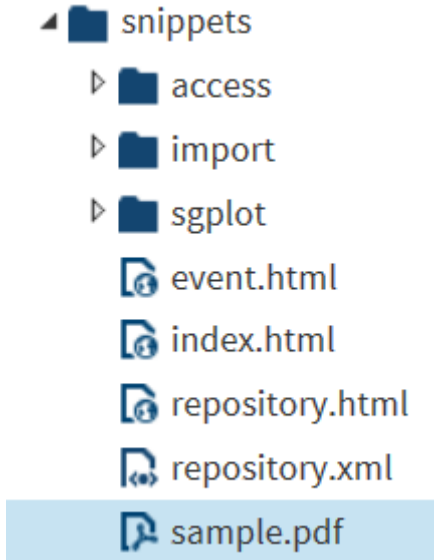
  <Icon url="http://icons.iconarchive.com/icons/3xhumed/cryonic
    folder/128/Folder-Blank-sample-icon.png"/>
  <Documentation uri="./sample.pdf" label="Embedded Repository Documentation
    (via PDF)"/>
  <Documentation uri="http://support.sas.com/documentation/cdl/
    en/proc/67916/HTML/default/viewer.htm#p19bcwkxesgd19nlciwhc8195s6s.htm"
    label="Procedures Guide"/>

  ...

</Repository>
```


You can specify multiple `Documentation` elements for a repository, task, or snippet. In this code, there are two `Documentation` elements.

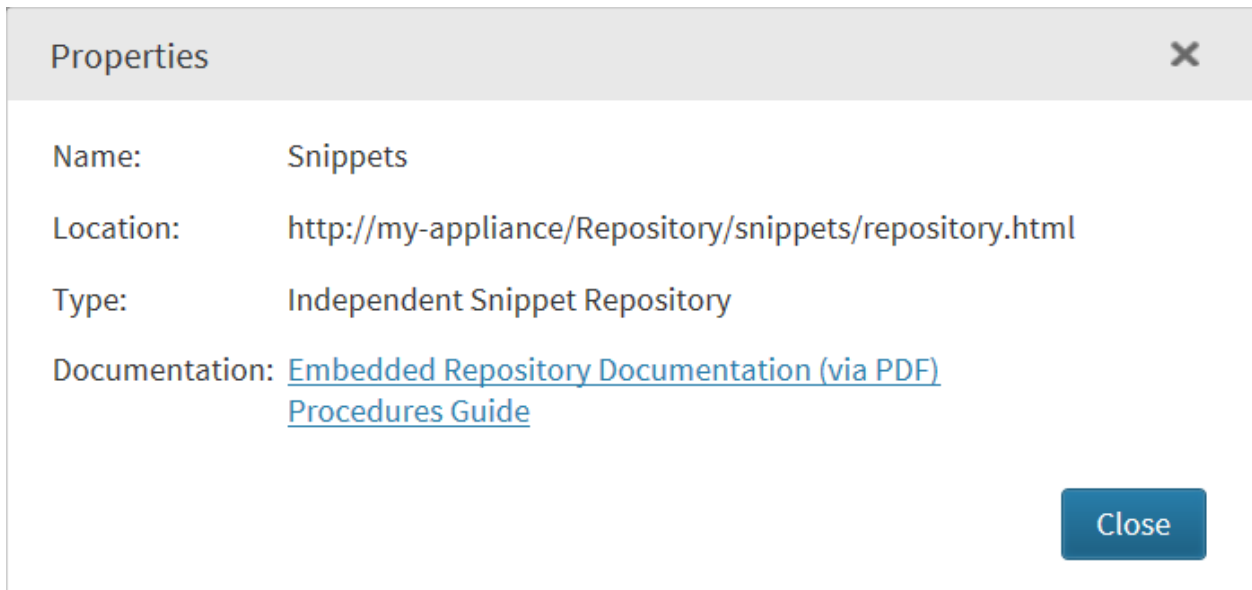
1. The first `Documentation` element uses a relative path because the `sample.pdf` file is saved with the other files in the repository. (See Display 8.)



Display 8: Example of Documentation Saved in a Repository

- The second `Documentation` element uses an absolute path because it is referring to the *SAS Procedures Guide*, which is available at <http://support.sas.com/documentation/cdl/en/proc/68954/HTML/default/viewer.htm#p19bcwkxesgd19n1ciwhc8l95s6s.htm>.

In SAS Studio, the user accesses this documentation by selecting the name of the repository (Snippets) in the **Snippets** section of the navigation pane. (See Display 7.) Then clicking . The Properties window displays the documentation links.



Display 9: Location of Documentation Links for Snippets Repository


When you click **Embedded Repository Documentation (via PDF)**, the PDF version of the documentation opens from this URL: <http://my-appliance/Repository/snippets/sample.pdf>.

When you click **Procedures Guide**, a new tab opens in your web browser and shows the HTML version of the *SAS Procedures Guide* at support.sas.com.

You can also use the `Documentation` element to add documentation to individual tasks and snippets in your repository.

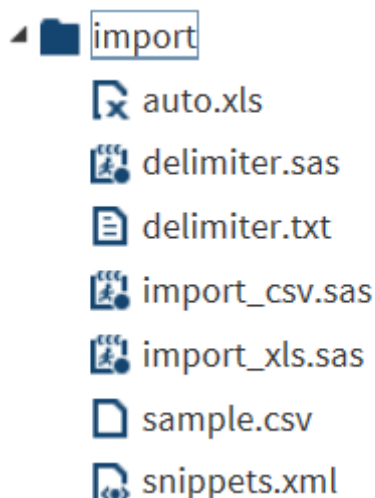
Here is an example of a `Documentation` element in the `access/snippets.xml` file:

```
<Snippet>
  <Name label="Assign Oracle Library"/>
  <Documentation uri="https://support.sas.com/documentation/cdl/
    en/acreldb/68028/HTML/default/
    viewer.htm#plujrhdoelp743n12awcf7mwyg81.htm"
    label="SAS/ACCESS Interface to Oracle"/>
  <Template uri="./oracle.sas"/>
  <Icon uri="http://rocketdock.com/images/screenshots/oracle.png"/>
</Snippet>
```

To access this documentation, select the **Assign Oracle Library** snippet in the **Snippets** section of the navigation pane. (See Display 7). Click . In the Properties window, you see the link to the SAS/ACCESS Interface to Oracle documentation.

Referencing Data Files in the Repository

You can also include data files (such as XLS, TXT, and CSV file) in a repository. The `import` folder (for the Import Snippets category) includes three data files: `auto.xls`, `delimiter.txt`, and `sample.csv`.



Display 10: Examples of Data Files in the Repository

Now, here is the content of the `import_xls.sas` file.

```
filename url url "{baseURL}/auto.xls";
filename xls temp;

data _null_;
  infile url recfm=f lrecl=1;
  file xls recfm=f lrecl=1;
  input x $char1.; put x $char1.;
```

```
run;

proc import out = WORK.autol datafile=xls
  dbms=xls replace;
  sheet="autol";
  getnames=yes;
run;

proc print;
run;
```

This code assumes that the SAS server can access URLs where the repository resides by leveraging the power of the FILENAME Statement, URL Access Method. The SAS code is saved in the same location as the external file. You can reference this data from your SAS code by using `{baseURL}`, which resolves the repository URL when accessed.

For more information about the URL Access Method, see *SAS 9.4 Statements*.

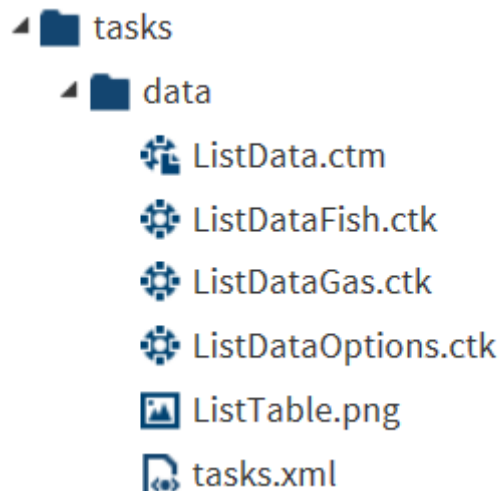
SAVED TASKS REPOSITORY

Task files can be saved as either a CTM or CTK. A CTM file is the XML and Apache Velocity code for the task. A CTK file is a task that is saved with one or more options already set. Using a CTK file makes it easier to run the same task with the same settings over and over again.

The Saved Tasks Repository demonstrates these features:

- defining multiple task and task categories
- referencing CTM and CTK task files

Task files must be saved in the repository. In this example, the repository.xml file defines only one task category called List Data. As a result, there is only one task folder (`data`) in the repository. The `data` folder contains all of the CTM and CTK files that are used in this repository.



Display 11: Contents of the Saved Tasks Repository

Here is a portion of the `data/tasks.xml` file:

```
<?xml version="1.0" encoding="UTF-8"?>

<Tasks>

  <Task>
    <Name label="List Data Model" />
    <Template uri="./ListData.ctm"/>
  </Task>

  ...

  <Task>
    <Name label="List Data - SASHELP.VOOPTIONS" />
    <Template uri="./ListDataOptions.ctk"/>
    <Documentation uri=http://www.atm.ucla.edu.stat/
      Library/samug/bt3004.pdf />
    <Icon uri="./ListTable.png">
  </Task>
</Tasks>
```

As you can see, the `Template` element accepts either CTM or CTK files.

GLOBAL REPOSITORIES

You might have a repository (or multiple repositories) that you want to make available to everyone at your site. Instead of requiring each SAS Studio user to add these repositories through the Preferences window (See the "Viewing Repositories in SAS Studio" section of this paper), you can create global repositories.

To create a global repository:

1. Create a `repositories.xml` file that lists all the repositories you want to include.

The following `repositories.xml` file combines the Simple Repository, Snippets Repository, and Saved Tasks Repository:

```
<?xml version="1.0" encoding="UTF-8"?>
<Repositories>
  <Repository uri=http://my-appliance/simple name="Simple
    Repository" />
  <Repository uri=http://my-appliance/snippets name="Snippets
    Repository" />
  <Repository uri=http://my-appliance/tasks name="Saved
    Task Repository" />
</Repositories>
```

2. Save the `repositories.xml` file to the global preferences directory for SAS Studio. (You specified this directory by using the `webdms.globalSettings` property in the `config.properties` file.) If you don't know the location of this directory, contact your SAS Studio administrator.

Now, the tasks and snippets in the Simple Repository, Snippets Repository, and Saved Task Repository are available to all SAS Studio users at your site. SAS Studio users can access these tasks and snippets from the **Tasks and Utilities** section and **Snippets** section in the navigation pane. The SAS Studio user doesn't need to set any preferences to access these repositories.

Note: Global repositories are not listed in the **Repositories** section of the Preferences window. SAS Studio users cannot change the contents of a global repository within SAS Studio. To change a global

repository, an administrator or developer with Write permission must edit the repositories.xml file in the global preferences directory.

With these settings, only global repositories are available. To allow users the ability to add their own repositories, set the `webdms.allowRepositories` property to `true` in the `config.properties` file. To access these user-defined repositories in SAS Studio, the SAS Studio user needs to use the Preferences window. (See the “Viewing Repositories in SAS Studio” section of this paper.)

For more information about these configuration properties, see *SAS Studio: Administrator’s Guide*.

CONCLUSION

Now that you understand the different elements available for repositories, you can use these examples as the basis for creating your own repositories.

You can choose the tools that you use to develop and deploy your repositories. If you are a SAS Studio user, you can edit your files directly in SAS Studio by creating an FTP folder shortcut to your repository. If you aren’t a SAS Studio user, you might prefer to use an editing tool outside of SAS Studio. You can edit the repository files using any text editor.

To deploy your repository, you can use your company’s internal web content system or a source management system, such as GitHub. The method that you use depends on the requirements and needs of your organization.

RECOMMENDED READING

- *SAS Studio: Developer’s Guide to Repositories*
- *SAS Studio: Administrator’s Guide*

All SAS Studio documentation is available at <http://support.sas.com/documentation/onlinedoc/sasstudio>.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author:

Marie Dexter
SAS Institute Inc.
919-677-8000
marie.dexter@sas.com
www.sas.com

Swapnil Ghan
SAS Institute Inc.
919-677-8000
swapnil.ghan@sas.com
www.sas.com

Michael Monaco
SAS Institute Inc.
919-677-8000
michael.monaco@sas.com
www.sas.com

Amy Peters
SAS Institute Inc.
919-677-8000
amy.peters@sas.com
www.sas.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.