

## Paper 171-27

**Tools for Sharing SAS<sup>®</sup> Macros over the Internet**

Xiao Chen and Michael N. Mitchell, UCLA, Los Angeles, CA

**ABSTRACT**

In this paper, we introduce a set of SAS<sup>®</sup> macro programs for automating the process of downloading and documenting SAS<sup>®</sup> macro programs over the Internet. We also describe a standard mechanism for organizing and documenting SAS macros on an Internet server.

**INTRODUCTION**

There are a variety of methods for delivering SAS macros over the Internet and a variety of ways of documenting the downloaded macros. This paper describes a set of SAS macros that we developed and called, **%net tools**, to ease the process of downloading macros and documenting the downloaded macros. We also introduce a standard mechanism for organizing and documenting SAS macros on an Internet server. The **%net tools** allow a SAS user, from within SAS, to view the list of macros stored at a remote web site, view information about the macros, view help files associated with the macros, and download the macro(s). When a user downloads the **%net tools**, the macros are stored in a user-chosen directory, which we call, Personal Macro Folder, where SAS will automatically look for and load these macros. This document describes this set of macros and how to download and use them. This also describes how to create a web site from which SAS macros can be distributed using the **%net tools**.

**AUTOMATING THE PROCESS OF DOWNLOADING SAS MACROS**

Currently, there are many web sites that store SAS macro programs for other users to download and use. A very common scenario for a common user to download a SAS macro program over the Internet involves following steps.

- Correctly locate the desired SAS macro program over the Internet (e.g. through searching the Internet or getting the URL from a colleague);
- Copy the SAS macro program code from the web site into a local file and use the %include statement to include the downloaded SAS macro program in a SAS program;
- Or copy and embed the SAS macro program into a SAS program before the macro is called;
- Debug the SAS program including the SAS macro program downloaded from the Internet.

There are several obvious problems with this process.

- For a novice SAS user, most of the time, this process is simply too difficult to follow;
- There is no default directory to store the downloaded macros;
- Help file associated with the macro program is usually embedded inside the macro program and is not very convenient for debugging the entire program.

In order to ease the process of downloading SAS macros over the Internet, we focus on two ideas. One is to have a SAS macro program for downloading a SAS macro program over the Internet to a default directory on a local machine and have access to it right away within the SAS Display Manager System. The other idea is to organize a user-written SAS macro program into a package. This integrated package consists of the macro program itself, the help file associated with it and possibly a SAS sample data set. This idea is borrowed from STATA. The first part is from the point of view of a user for easy downloading and the second part is from the point of view of an Internet server for easy distributing. Combining the two ideas together, we have created our **%net tools** with following features:

- On an ideal Internet server, an integrated package associated with each SAS macro programs;
- On a user's side, a default directory to store the downloaded macro packages or simply macro programs;
- Given a URL to a SAS macro program, the macro will be downloaded to the default directory from within SAS and can be called immediately;
- A database associated with the downloaded macro programs.

**IMPLEMENTATION OF %NET TOOLS**

SAS 8 has enhanced the filename statement with URL or ftp access method. Our implementation of **%net tools** takes full advantage of this feature. Therefore it is sufficient to only have the base SAS software to run our **%net tools**. We consider this as an important feature of convenience of the **%net tools**. On user's side, a small data set is created and maintained by SAS/SQL procedure to keep track of the macro programs that have been downloaded. The information on downloading status is displayed using SAS window statement. The interactive feature of the SAS window statement has been used a few times to get user's input on the location of the default folder to store the macro programs, whether to overwrite an already existing macro and etc. The content of a package file or a help file is displayed in the log window.

**INSTALLING THE %NET TOOLS**

The **%net tools** are stored at our web server and anyone who has Internet access can install the **%net tools** by visiting <http://www.ats.ucla.edu/stat/sas/macros/> and clicking on the link that points to the **%net tools installation program**. The installation program is a SAS program that a user can copy and paste into a SAS program editor and then submits the program to install the **%net tools**. The user will then be asked the folder he or she wants to use as the **Personal Macro Folder**, the folder where the user wants the downloaded SAS macros

to be stored. It will then install the **%net tools** to user's computer into the folder that has been specified. If this folder does not exist, it will be created automatically in the setup program. The autoexec.sas file will be modified (or created) adding a couple of lines that indicate where the **Personal Macro Folder** should be located and expanding the search path for **sasautos** statement so SAS will automatically look in the Personal Macro Folder for macros.

## USING THE %NET TOOLS

Once a user has run the **%net tools installation program**, he or she can then access and use the following macros listed in the following table.

%net_list	List packages stored on a web server
%net_describe	Describe a package stored on a web server
%net_help	Get help for a macro stored on a web server
%net_get	Download files in a package from a web server
%net_update	Updates the net utilities from a server
%local_list	List packages that have downloaded
%local_describe	Describe a package that has been downloaded
%local_help	Get help for a macro that has been downloaded

The **%net\_list** macro allows a user to view the list of packages available at a particular URL. For example,

```
%net_list(
http://www.ats.ucla.edu/stat/
sas/macros/ )
```

would list all of the packages available from **http://www.ats.ucla.edu/stat/sas/macros/**. A user can get more information about a particular package with the **%net\_describe** macro, for example

```
%net_describe(test,
http://www.ats.ucla.edu/stat/
sas/macros/ )
```

gives information about the package named **test** located at **http://www.ats.ucla.edu/stat/sas/macros** and the list of files contained in the package **test.**, which can be useful to help the user to decide whether to download it. Please note that these macros recall the last URL that you have visited and will default to that location. So, the above macro could have been issued as

```
%net_describe( test )
```

and it would have assumed the same URL, because we last visited **http://www.ats.ucla.edu/stat/sas/macros/**.

You can get view the help file associated with a macro called **%net\_help**. For example we can view the help file associated with the macro **test** as shown below.

```
%net_help(test,
http://www.ats.ucla.edu/stat/
sas/macros/ )
```

You can use **%net\_get** to download a package to your computer, for example

```
%net_get(test,
http://www.ats.ucla.edu/stat/
sas/macros/ )
```

will download the files in the package named **test** to your computer, storing the files in your Personal Macro Folder you specified when you ran the **%net tools** installation program. After running **%net\_get** as shown above, the macros contained within **test** will be available for you to use. If the package has supplemental files, you can indicate that you want to download the supplemental files and the location for storing them as shown below.

```
%net_get(test,
http://www.ats.ucla.edu/stat/
sas/macros/, c:\mydata)
```

This would store the supplemental files, if any, in **c:\mydata**. If **http://www.ats.ucla.edu/stat/sas/macros/** was the last url you had visited, you could instead type

```
%net_get(test,, c:\mydata)
```

and if you did not want to download the supplemental files (or if there were no supplemental files) then you could simply type:

```
%net_get( test )
```

If you wish to download a package again (for example, to get the latest version) you can use the **replace=yes** option to indicate that you wish the replace the existing version of the macro. Otherwise, if you go to download package that you have already downloaded, a window will pop up indicating that the package already exists and it will ask you if you wish to replace the existing version. Below are a couple of examples of using the **replace=yes** option.

```
%net_get(test,
http://www.ats.ucla.edu/stat/
sas/macros/,replace=yes)
```

Or you could type the following if **http://www.ats.ucla.edu/stat/sas/macros/** was the last url you had visited

```
%net_get( test , replace=yes )
```

**PLEASE NOTE:** If you download a package again, the new version of the macros contained in the package will **not** be available to you until you quit SAS and then restart SAS. Otherwise, SAS may invoke the old version of the macros contained in the package.

If you wish to get the most up-to-date version of the **%net tools**, you can use the **%net\_update** macro and this will download the most recent version of the **%net tools** to your computer. Note that you need to quit and restart SAS for the most recent versions of the macros to become available.

There are additional macros for managing files on your local machine, as described below. The **%local\_list**

macro lists the packages that you have downloaded, for example

```
%local_list .
```

The `%local_describe` macro will describe a package that you have downloaded. For example, assuming you have downloaded the package `test`, executing

```
%local_describe( test )
```

will show information about the package called `test`. Finally, the `%local_help` macro will display the help file associated with a macro. For example,

```
%local_help( test )
```

will show the help file associated with the macro `test`.

## LIST OF MACROS AND DESCRIPTION

The following is a list of the macros that are part of the `%net tools` along with the syntax for using the macros. In each case, the `url` is optional and would default to the most recently accessed `url`.

- `%net_list( url )` - List packages stored on a web server.
- `%net_describe( package_name, url )` - Describe a package stored on a web server.
- `%net_help( macro_name, url )` - Get help for a macro stored on a web server.
- `%net_get( package_name, url , local_path , replace=yes )` - Download files in a package from a web server.
- `%net_update` - Update the net utilities to the latest version from the ATS web server.
- `%local_list` - List packages you have downloaded .
- `%local_describe( package_name )` - Describe a package you have downloaded.
- `%local_help( macro_name )` - Get help for a macro you have downloaded.

## CREATING YOUR OWN SITE

If you find this to be a useful and convenient way to deliver and document SAS macros, then you may wish to create your own site for distributing macros using the `%net tools`. The steps for creating your own site are described below.

Here is a simple example based on a site we have created at <http://www.ats.ucla.edu/stat/sas/samplesite/>.

First, we have a macro called `test.sas`. Here is what `test.sas` looks like

```
%macro test;
put This is a very simple macro;
%mend;
```

Associated with this macro, we created a help file called `test.hlp`, which is a simple text file. Here is what `test.hlp` looks like.

```
This is the test help file for
test.sas.
```

We also have an associated sample data file that accompanies this macro called `hsb2.sas7bdat`. We want to distribute these files together in a `package`, so we create a file called `test.pkg` located in <http://www.ats.ucla.edu/stat/sas/samplesite/> that indicates all of the files contained in this package. The file `test.pkg` looks like this.

```
d test. Test Program
d Statistical Computing and Consulting
d UCLA Office of Academic Computing
d
f test.sas
f test.hlp
s hsb2.sas7bdat
```

The top of `test.pkg` contains a series of lines that describe the package, each starting with a "d". The contents of the description can be anything, as long as they start with a "d" and a space. Following the description is a list of files. Macros and the help files associated with them start with an "f". When a user runs

```
%net_get(test,
http://www.ats.ucla.edu/stat/sas/
samplesite/ ),
```

the files `test.sas` and `test.hlp` will be downloaded to the user's **Personal Macro Folder**. We also specify that `hsb2.sas7bdat` is a supplemental file, indicated by an "s". If the user runs

```
%net_get(test,
http://www.ats.ucla.edu/stat/sas/
samplesite/ ),
```

these sample files will not be copied; however if the user runs

```
%net_get(test,
http://www.ats.ucla.edu/stat/sas/
samplesite/ , c:\mydata ),
```

the macro will be downloaded and the file `hsb2.sas7bdat` will be copied to the user's `c:\mydata` directory.

Although the package `test.pkg` contained just one macro, your package can contain multiple macros, multiple supplemental data files, or no supplemental files.

As you create more and more packages, you would probably like users to be able to see a list of all of the packages available at your site, a kind of table of contents. You can create a file called `sas.toc` that lists all of the packages that are available at that url which users can view with the `%net_list` command. The `sas.toc` file is needed for the `%net_list` command to work for the users, but is not needed for `%net_get` (or

**%net\_help**) to work properly. Here is the **sas.toc** file we have located at <http://www.ats.ucla.edu/stat/sas/samplesite/>.

```
d Welcome to UCLA Academic
d Technology Services SAS sample
d site.
d For more information about these
d programs, see our web page at
d http://www.ats.ucla.edu/stat/sas/
d
p test This is a simple test program
```

The format of the **sas.toc** file is much like a package file. It contains one or more lines describing the site, each starting with a "d". It then has a list of the packages found at that url, each starting with a "p". As you can see, the "p" lines start with "p" followed by a space and then the name of the package, and then a space and a description of the package. You can have as many packages in a **sas.toc** file as you like. Having created this **sas.toc** file, a user can then type

```
%net_list(
http://www.ats.ucla.edu/stat/sas/
samplesite/ )
```

and then the user would see the description lines followed by the list of packages available.

### ADDING NEW PACKAGES TO YOUR SITE

Above we showed the steps for creating a site from scratch. Suppose you want to add a new package to your site, called **newmacro.sas**. These are the steps you would follow:

1. Create a help file for the macro (called **newmacro.hlp**) that is a simple text file describing how to use the macro.
2. Create a package file (called **newmacro.pkg**) that describes the package and lists the files in the package (like the package file shown above).
3. Copy **newmacro.sas**, **newmacro.hlp** and **newmacro.pkg** to your web server.
4. Update your **sas.toc** file, adding a "p" line for the **newmacro** package.

Users could then use

```
%net_list(
http://www.ats.ucla.edu/stat/sas/sa
mplesite/ )
```

and see that **newmacro** is now one of the packages located at the site and

```
%net_get(newmacro,
http://www.ats.ucla.edu/stat/sas/samplesite/ )
```

to download the package.

### CONCLUSION

The **%net tools** are designed to provide a straightforward and user-friendly way of sharing SAS macros from within

SAS. As compared to some of the current methods used for delivering SAS macros, the **%net tools** offer the following advantages:

- A standard way of obtaining macros from within SAS.
- A way of bundling a group of related macros and supplemental files into a single package.
- The macros are automatically accessible after downloading.
- The **%net\_help** and **%local\_help** provide a standard mechanism for delivering help associated with the macros.

We invite other sites to join us in using the **%net tools** for sharing macros over the Internet.

### REFERENCES

SAS Institute Inc., *SAS OnlineDoc® SAS Macro Language: Reference, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS Institute Inc., *SAS OnlineDoc® SAS Procedures Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999.

### CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Xiao Chen  
Academic Technology Services, UCLA  
University of California, Los Angeles  
Los Angeles, CA 90095  
Work Phone: (310) 206-0660  
Fax: (310) 206-7025  
Email: [jingy1@ucla.edu](mailto:jingy1@ucla.edu)  
Web: <http://www.ats.ucla.edu/stat/>

Michael N. Mitchell  
Academic Technology Services, UCLA  
University of California, Los Angeles  
Los Angeles, CA 90095  
Work phone: (310) 825-7431  
Fax: (310) 206-7025  
Email: [mnm@ucla.edu](mailto:mnm@ucla.edu)  
Web: <http://www.ats.ucla.edu/stat/>