Custom Tasks for SAS® Enterprise Guide® Using Microsoft .NET

Chris Hemedinger

SAS® Press

Contents

Chapter 1: Why Custom Tasks?

For more than 35 years, people have been writing SAS programs to solve business problems, conduct research, and report on data. For almost as long, the people who use those SAS programs have strived to invent ways to make those programs reusable, adaptable to changing business processes, and approachable by non-programmers.

A custom task provides one way to leverage your SAS programs and make them usable by a wider audience. Custom tasks are a hook for extending SAS Enterprise Guide and the SAS Add-In for Microsoft Office—two popular desktop applications that bring the power of SAS to a wide range of users. If you work with SAS users who use these applications as their primary interfaces to SAS, then custom tasks can bring your SAS solutions right to them, without asking them to leave the environment that they know and love.

A custom task is the easiest and most natural way for a user to access a proprietary process. However, a custom task can require a significant investment in time and expertise to create. In most cases, the return on investment—realized in consistency, control, and ease of use—far outweighs the cost of developing the task.

Why Isn't Everything Built In for Me?

If every conceivable SAS process was already built into SAS and accessible at the touch of a button, there wouldn't be any room for customers to add their own intellectual property to the system. (And, SAS consultants couldn't earn a living.)

SAS offers hundreds of modules, called *procedures*, which perform specific analyses, produce reports, and manipulate data. SAS Enterprise Guide provides almost 90 tasks that use the most popular procedures to accomplish much of what customers need to do.

In addition, SAS provides rich programming languages that you can use to do almost anything in the world of data manipulation, reporting, and logical flow. These languages include the DATA step and the SAS macro language. It would be impossible to create a set of tasks that represents everything that you can do with these languages. Why, SAS experts are still inventing new uses for these languages every day! The application of SAS languages varies from company to company, industry to industry, and department to department.

Options for Custom Processes in SAS Enterprise Guide

If you need to customize a process that isn't covered by a built-in task in SAS Enterprise Guide, you have several options.

Write a SAS program.
For a SAS programmer, it's the ultimate in flexibility. Anything that you can do in a DATA step, macro language, or SAS procedure is available to you in a SAS program. What's the problem with this approach? The only person who understands how to write the ultimate SAS program is a SAS programmer. If you have an audience that includes non-programmers, you probably need something more controlled.

Write a SAS program and add prompts.
With SAS Enterprise Guide, you can add placeholder values, called *prompts*, to your project. When you reference one of these values in a SAS program or in any task, SAS Enterprise Guide presents a user-friendly prompt dialog box that collects values from the user. When you attach prompts to a SAS program, you can make the program more usable. There is a wide range of prompt values that address a variety of situations, and the user doesn't need to

understand the content of the program to provide values. However, the user does need to know how to *run* the SAS program. The prompt dialog box, while somewhat flexible, does not offer much interactivity. You can define prompts that include a predefined list of values and simple range checking, but you have little control over how the prompts are presented.

Create a stored process.

A stored process is a SAS program that you store in a central repository. You can add prompts to the program and define permissions to control who can run it. And, you can run the program from various application environments, including SAS Enterprise Guide, the SAS Add-In for Microsoft Office, SAS Web Report Studio, and custom web applications. A stored process is a great way to reuse a SAS program and use it consistently across the enterprise.

With authoring tools like SAS Enterprise Guide, it is relatively simple to create a stored process. However, just like writing a SAS program with prompts, your control of the user experience is limited. You have little control over how the prompts are presented to the user, although you can include a predefined list of values and simple range checking. Because stored processes can leverage the SAS metadata environment, you can build prompts that are dynamic. Dynamic prompts pull values from live data and permit cascading prompt selections.

Create a custom task.

Anything that you can do with a SAS program, you can do with a custom task. And, anything that you can do with a Windows application (which means *the best* in terms of an interactive and responsive user experience), you can do with a custom task. If you need access to data or resources that you cannot reach with a SAS program, a custom task can act as the bridge to bring this information into SAS.

Unlike a stored process, custom tasks work in only two SAS applications: SAS Enterprise Guide and the SAS Add-In for Microsoft Office. If you have to provide the feature to a user of a SAS web application or a user in SAS Web Report Studio, a stored process might be how to go.

Table 1.1 summarizes your programming options and how they rank in usability and difficulty. I don’t want to mislead you: creating a custom task is more involved and requires more technical skill than some of the other options. But, the payoff can be well worth the investment. And, the tools and techniques presented in this book can make creating a custom task much easier.

Table 1.1: Comparing Programming Options in SAS Applications

Option	User Skills	Designer Skills	Usability
SAS program	Run or modify SAS program	Program in SAS	None; works only in code view
SAS program with prompts	Run SAS program; point and click to answer prompts	Program in SAS; design prompts	Constrained by prompting technology; only static prompts are allowed
SAS stored process	Point and click to run process and answer prompts	Program in SAS; design prompts; know about SAS metadata	Constrained by prompting technology; dynamic prompts are possible
SAS custom task	Point and click to run task	Program in SAS; program in Microsoft .NET design UI	Almost anything is possible; UI is intuitive; dynamic and responsive experience

What Can I Do with Custom Tasks?

Custom tasks straddle the gateway between two very powerful worlds: the world of your SAS session and the world of your Windows desktop.

Your SAS session provides access to the most advanced analytics and reporting capabilities available in business intelligence applications today. Your Windows desktop, using Microsoft .NET as a framework, provides a truly innovative and rich user experience while accessing data from your desktop, network, or the Internet.

Here are some example uses for custom tasks:

- Create a user interface as a front end for a SAS procedure or SAS technique that you use all of the time, but that doesn't have a built-in task in SAS Enterprise Guide. With the more than 250 procedures that SAS offers plus the extensive programming language, it's no surprise that there are gaps between what the user interface offers and what SAS can actually do for you. (For an example, see Chapter 9, "The Top N Report.")
- Connect to a web service to gather data from an external source and import the data into SAS. (For an example, see Chapter 15, "Running PROCs on Your Facebook Friends.")

- Provide simple access to SAS macro programs used within your organization and make them more approachable to non-programmers. (For an example, see Chapter 13, "Putting the Squeeze on Your SAS Data Sets.")
- Provide a sign-in prompt to a proprietary database or server resource without revealing the SAS code required for this action.

There are a few things that customers often want to do with custom tasks, but currently cannot. These include:

- Automate the SAS Enterprise Guide interface to add to and run parts of your project. You can accomplish some of this with the SAS Enterprise Guide automation model, which is a separate mechanism from the custom task APIs (application programming interfaces).
- In SAS Enterprise Guide 4.1, you cannot use custom tasks to examine project contents and extract information such as SAS programs. However, there are newer APIs in SAS Enterprise Guide 4.2 and later that allow this. (For an example, see Chapter 10, "For the Workbench: A SAS Task Property Viewer.")

Who Uses Custom Tasks Today and What Do They Use Them For?

Who uses custom tasks? What do they use them for? The short answer is that lots of people use custom tasks for lots of things. Even SAS uses custom tasks to prototype new features in SAS Enterprise Guide and to deliver those new features between major releases.

Before Custom Tasks: SAS/AF

Custom tasks are not a new concept in SAS. For years, SAS customers have created customized SAS applications using the SAS application framework (the product SAS/AF). SAS/AF provides an environment to host a full-screen application based in SAS with a custom user interface and full access to the power of SAS.

The objectives of SAS/AF are the same as the objectives of custom tasks: provide a simplified, focused user interface to enable users to perform a task that is specific to their industry, company, or department. Even now, there are companies that have a large investment in SAS/AF applications and their users continue to interact with SAS in this way.

SAS/AF applications were just the ticket when SAS users used to "live" in the SAS Display Manager, which acted as the host for SAS/AF. Today, most SAS users access SAS using other products, such as SAS Enterprise Guide. New SAS users might never see the full-screen environment of the SAS windowing environment. In fact, because many enterprises centralize access to the SAS session on powerful servers (where there is no windowing environment) instead of on user desktops, it's not even possible for users to access the SAS windowing environment. Users must get to SAS using a SAS product, such as SAS Enterprise Guide.

Bringing Custom Tasks to the Desktop

In 2001, SAS Enterprise Guide 2.0 introduced the concept of custom tasks. At that time, SAS documented a simple set of APIs that customers used with Microsoft Visual Basic 6.0. Since then, desktop technology has advanced, but the concept of custom tasks remains the same. Today, the APIs support Microsoft .NET, so customers can select their language of preference. C# and Visual Basic .NET are the most popular. These are modern programming languages with many uses outside of SAS. Learning Microsoft .NET skills can be a career-enhancing move, helping you grow your marketable professional skills.

In the Field: Custom Tasks at Work

SAS includes SAS Enterprise Guide with many of its packaged solutions. In this context, SAS Enterprise Guide is usually intended for advanced or power users and helps them with ad hoc reporting on data within the solution's domain. To provide this help, solutions have custom tasks that perform special functions. Examples of SAS solutions that include custom tasks are SAS Activity-Based Management, SAS Warranty Analysis, SAS Enterprise Miner, and SAS Forecast Server.

SAS users can create their own custom tasks to provide a wide range of features to their internal audiences. Examples of real-world custom tasks include a wizard to aid with clinical trial analysis, a dialog box to facilitate user sign-on with SAS/CONNECT, a task that provides easy access to data by assigning the SAS libraries needed by a user or a group, and tasks that create a set of reports and charts for use within Microsoft Excel.

The pluggable nature of custom tasks makes them a perfect mechanism for SAS to enhance SAS Enterprise Guide between releases. On the *Downloads & Hot Fixes* page of the support.sas.com site, you can find custom tasks to enhance your SAS Enterprise Guide installation. These tasks can help you upload and download SAS data sets to and from your SAS server, register SAS tables within metadata libraries, and create graphs using an interactive interface.

Deploying Custom Tasks

A custom task is shipped in an executable .NET file called an *assembly*. (An assembly usually takes the form of a DLL file.) You can have multiple custom tasks in a single .NET assembly, or you can have one custom task per assembly (or DLL file), depending on your preferences. Keep in mind that it is more convenient to package several related custom tasks together into a single .NET assembly because it makes it easier to share code and implementations.

Once the .NET assembly has been built, deploying and registering the custom tasks to target client machines is simple, and two methods are available. Once the tasks are added, they are available from the **Add-In** option in the **Tools** menu and from the **Task List** option in the **View** menu in SAS Enterprise Guide. In the SAS Add-In for Microsoft Office, you can find custom tasks by selecting **Manage SAS Favorites**.

Method 1: Drop-In Deployment

You do not need to perform a separate registration step for SAS Enterprise Guide or the SAS Add-In for Microsoft Office to recognize your custom task. You simply copy the DLL file to a special folder and SAS Enterprise Guide or the SAS Add-In for Microsoft Office recognizes it automatically the next time you start either product.

The special folder locations are different, depending on which version of the product you have installed.

SAS Enterprise Guide 4.1 and the SAS Add-In for Microsoft Office 2.1

In SAS Enterprise Guide 4.1 and the SAS Add-In for Microsoft Office 2.1, the special folder location is:

```
C:\Program Files\SAS\Shared Files\BIClientTasks\Custom
```

You need to create the **`Custom`** subdirectory if it doesn't exist. The first part of the path (`C:\Program Files\SAS\Shared Files`) might vary, depending on your installation.

SAS Enterprise Guide and the SAS Add-In for Microsoft Office 4.2, 4.3, 5.1, and later

Beginning with SAS Enterprise Guide 4.2 and the SAS Add-In for Microsoft Office 4.2, there are multiple special folders for storing custom tasks. These folders are specific to each release, which means that the folder names are slightly different.

Each folder location supports different deployment scenarios. For example, if you do not have administrative privileges on your PC, you might not be able to copy a DLL file to the `Program Files` location. As a result, one of the folder locations is in the user profile area, which is an area specific to each user account on the PC. Most configurations permit a local user to place files in his or her user profile area.

For SAS Enterprise Guide, the folder locations are:

- `%appdata%\SAS\EnterpriseGuide\<`*`version`*`>\Custom`
- `C:\Program Files\SAS\EnterpriseGuide\<`*`version`*`>\Custom  (SAS 9.2 installation)`
- `C:\Program Files\SASHome\x86\SASEnterpriseGuide\<`*`version`*`>\Custom (SAS 9.3 installation)`

For the SAS Add-In for Microsoft Office, the folder locations are:

- `%appdata%\SAS\AddInForMicrosoftOffice\<version>\Custom`
- `C:\Program Files\SAS\AddInForMicrosoftOffice\<version>\Custom (SAS 9.2 installation)`
- `C:\Program Files\SASHome\x86\SASAddInForMicrosoftOffice\<version>\Custom  (SAS 9.3 installation)`

If you want to deploy a single copy of a custom task in both applications, you can copy it to this folder location:

```
%appdata%\SAS\SharedSettings\<version>\Custom
```

Notes: The %appdata% value is a Microsoft Windows environment variable that maps to your personal profile area, which is an area specific to your user account on the PC. You can add custom tasks to your installation without affecting other users who might share your machine.

If your custom task depends on other .NET assemblies that are shipped with SAS Enterprise Guide or the SAS Add-In for Microsoft Office, you do not need to copy those .NET assembly files to the **`Custom`** directory. However, if your custom task depends on .NET assemblies that are not provided by SAS (for example, they are from third-party vendors or they are .NET assemblies that you developed), you *do* need to copy those .NET assemblies to the **`Custom`** directory when you copy the custom task.

With drop-in deployment, the SAS client application searches the designated `Custom` folders and any applicable subfolders on your system. If your custom task includes several DLL files, you can group them into a single subfolder to make the deployment run more efficiently.

Example Deployment Scenarios

Suppose you have a custom task that uses forecasting techniques to predict product inventory needs. You have built this task into a DLL file named SupplyTasks.dll. How should you deploy the task for use?

Scenario 1: A Few Users on Local Machines

The task is used by just a few people in your organization. They use SAS Enterprise Guide 4.3 on their desktop machines. SAS Enterprise Guide was installed using the default settings with the SAS 9.3 Software Depot.

Scenario 1: Approach

Each user can copy the SupplyTasks.dll file to his machine for use. The user opens Windows Explorer and navigates to `%appdata%\SAS\EnterpriseGuide\4.3`. The user creates a new folder named `Custom` and copies the DLL file into it. The next time the user opens SAS Enterprise Guide 4.3, the task appears in the **Tools→Add-In** menu.

Scenario 2: Many Users on Many Machines

The task is very popular, and more people need to use it. The self-service method of allowing users to install it themselves might not scale well with a larger audience. In some cases, the task might need to be installed on a machine that is shared by multiple users.

Scenario 2: Approach

On each machine, copy the SupplyTasks.dll file to the `Custom` folder in the SAS Enterprise Guide installation directory. Open a Windows Explorer and navigate to `C:\Program Files\SASHome\x86\SASEnterpriseGuide\4.3`. Create a new folder named `Custom` (if it does not already exist) and copy the DLL file into it. The next time any user on the shared machine opens SAS Enterprise Guide 4.3, the task appears in the **Tools→Add-In** menu.

Method 2: Add-In Manager

The Add-In Manager dialog box lets you select a custom task from any location and register it for use in your SAS applications. To get started:

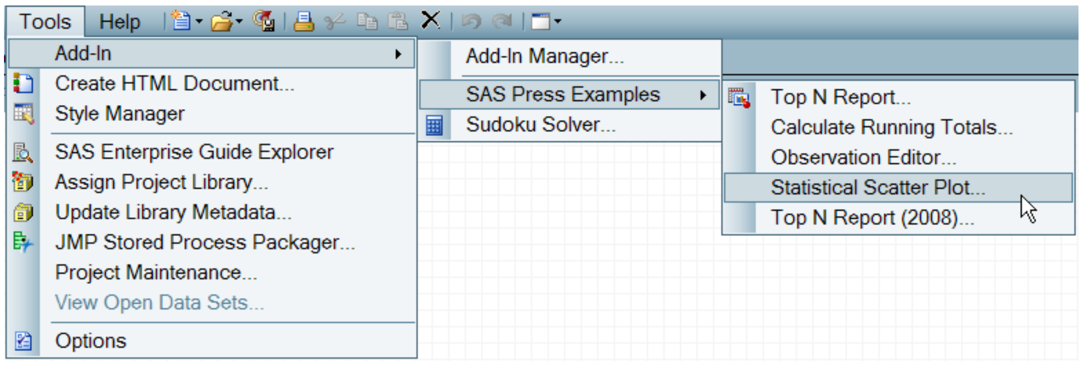
1. Copy the .NET assembly (and any dependent .NET assemblies excluding those that are shipped with SAS Enterprise Guide or the SAS Add-In for Microsoft Office) to a location on the target machine. If you plan to deploy more than one .NET assembly with add-in tasks, you can group them into a single directory.
2. To start the Add-In Manager:
   - In SAS Enterprise Guide 4.1, select **Add-In→Add-In Manager**.
   - In SAS Enterprise Guide 4.2, 4.3, or 5.1, select **Tools→Add-In→Add-In Manager**. In the SAS Add-In for Microsoft Office 2.1, run `C:\Program Files\SAS\Shared Files\BIClientTasks\4\RegAddin.exe`. (The actual location of this program might vary depending on your configuration.)
   - In the SAS Add-In for Microsoft Office 4.2 or 4.3, run `C:\Program Files\SAS\AddInForMicrosoftOffice\4.2\RegAddin.exe` or `C:\Program Files\SAS\AddInForMicrosoftOffice\4.3\RegAddin.exe`. (The actual location of this program might vary depending on your configuration.)
3. Click **Browse** to find the .NET assembly, and then click **Open**. The Add-In Manager dialog box shows the available add-in tasks in the .NET assembly that you selected.
4. Click **OK** to accept the add-in tasks.

Accessing Custom Tasks from the Menu

After your custom task is registered (by using drop-in deployment or the Add-In Manager dialog box), you can launch the task from either the **Add-In** menu or the **Task List** in the **View** menu in SAS Enterprise Guide.

In SAS Enterprise Guide 4.1, the top-level **Add-In** menu includes all of the custom tasks that you currently have registered. In SAS Enterprise Guide 4.2 and later, the **Add-In** menu is located under the **Tools** menu. Figure 1.2 shows the **Add-In** menu in SAS Enterprise Guide 4.2.

Figure 1.1: The Add-In Menu in SAS Enterprise Guide 4.2



You can also find the custom tasks in the Task List (**View→Task List**). In the Task List, the tasks are organized by category and the custom tasks are intermixed with the built-in tasks. In the **Add-In** menu, the custom tasks are separated into their own Add-In menu structure.

Note: Custom task developers often ask how they can get their custom tasks to appear in the main menus of SAS Enterprise Guide. They want them to be under a top-level menu or in the main categories of Data, Describe, Graph, and so on. Currently, you cannot add a custom task to a built-in menu. Only SAS tasks can appear in these built-in menus. All custom tasks appear under the **Add-In** menu.

Common Questions about Task Deployment

Developers who create custom tasks and the SAS administrators who support SAS Enterprise Guide users often have questions about the best methods to deploy and monitor custom tasks. Here are some common questions with answers:

Can I place the custom task DLL file in a shared network location so that all users can access it without having to copy it to their individual machines?

The answer is "it depends." In most environments, the Microsoft .NET run time is configured to allow .NET assemblies (DLL files) to load only from trusted locations. Usually, a network location (such as a mapped drive, UNC path, and especially the Internet) is not granted full trust. Without the full trust level, the Microsoft .NET run time refuses to load the .NET assembly.

An administrator can modify the Microsoft .NET security settings to grant levels of trust to a location or to a set of .NET assemblies. The specific instructions are not included in this book, but you find them by searching for “Microsoft .NET code access security” in Microsoft documentation.

Can I control which users are allowed to use my task?

If your SAS environment includes a SAS Metadata Server, then you can control which users have access to your tasks. The SAS metadata environment supports the concept of role-based capabilities.

Each built-in feature in SAS Enterprise Guide is registered as a capability that you can control. You can control who accesses your custom tasks by following these steps:

1. Disable the capability for users to access unregistered custom tasks.
2. For each custom task that you want to add, register the task in the SAS metadata environment. SAS provides the tools and documentation to accomplish this. Each task will appear as a new capability in the SAS metadata.
3. Add the new capability to the role definition of each group or user that you want to allow to access your custom tasks.

The tricky part about relying on metadata roles is that when the user is not connected to a SAS metadata environment, the user’s access roles are not checked. If you want your task to be available only when a user has been checked out in metadata, you can code your task as requiring metadata before it is enabled. (For more information, see the SasMetadataRequired attribute in Chapter 5, “Meet the Task Toolkit.”)

Can I see which users are actually using my task?

There is no built-in auditing for any of the SAS Enterprise Guide features, including custom tasks. However, there are a few ways to get this information:

- Use the Microsoft .NET logging mechanism (called log4net) to add information about the custom task activity to the SAS Enterprise Guide system log. The logging mechanism is described in Chapter 8, “Debugging Techniques: Yes, You Will Need Them.”
- Use Microsoft .NET classes to add information to the Windows event log. The Windows event log is easy to search using built-in administrative tools on Windows. You can even use log4net to add entries to the Windows event log.
- Create your own custom logging mechanism that records when a user uses the custom task. For example, you can use Microsoft .NET classes to add a record to a database or add an entry to a central log that is on the network.

Accessing Ready-to-Use Example Tasks and Source Code

All of the examples that are included in this book are available electronically on the companion web site at http://support.sas.com/publishing/authors/hemedinger.html. The examples are provided as prebuilt .NET assemblies (DLL files), and they are ready to drop in and use with your SAS software. Each example includes one or more Microsoft Visual Studio projects that have all of the source code that you need to build the examples and modify them for your use. Some examples are provided in the C# language, some in Visual Basic .NET, and some in both.

You do not need to understand how these tasks are built before you can use them. You should deploy them in your existing SAS Enterprise Guide installation and explore some of the capabilities that you can add with custom tasks.

Exercise: Download, Deploy, and Access Custom Tasks from SAS

This book includes a collection of custom tasks that illustrate many of the possible features that you can implement. Most of these custom tasks are described in greater detail later in this book. For now, you can deploy a ready-to-use DLL file that contains the tasks and you can see them in action.

To download the collection of custom tasks, follow these steps:

1. Using your web browser, navigate to http://support.sas.com/hemedinger.
2. In the section for this book, click **Ready-to-use custom tasks examples**. A Zip archive file appears. Use your web browser to download the Zip file to your PC.
3. Navigate to the location where the Zip file is located. Extract the contents of the Zip file to a folder on your PC. The Zip file contains a DLL file named SAS.Tasks.Examples.DLL.
4. Copy the DLL file to the appropriate `Custom` folder on your PC based on the version of SAS Enterprise Guide that you are using. For example, if you are using SAS Enterprise Guide 4.3, the location is `%appdata%\SAS\EnterpriseGuide\4.3\Custom`. (For the list of `Custom` folders for each of the different versions of SAS Enterprise Guide, see "Method 1: Drop-In Deployment" earlier in this chapter.)

   Tip: The *%appdata%* location is specific to your local Windows profile area, and the folder structure varies for different versions of Windows. An easy way to navigate to the correct folder is to open a new Windows Explorer and type `%appdata%` in the address bar.
5. Start SAS Enterprise Guide. Select **Tools→Add-In**. You should see several new menu items that provide access to the example custom tasks.
6. Select a menu item to launch your first custom task!

Chapter Summary

Custom tasks provide a powerful way to extend SAS business intelligence applications. They aren't the only way to present custom features to your users, but they are often the most flexible way. When you need a custom task, nothing else will do.

Many organizations use custom tasks to add valuable features to SAS Enterprise Guide and the SAS Add-In for Microsoft Office. Even SAS uses custom tasks to enhance its packaged solutions and to provide new features in its products that are in the field.

The mechanism for deploying and using custom tasks is simple. Once it is in place, your users can access custom tasks the same way they access built-in tasks from SAS.

About The Author

Chris Hemedinger has been with SAS since 1993. In that time he has been a writer, a software developer, and an R&D manager. He was on the founding team for SAS Enterprise Guide, leading the development team for 10 years. Today, Hemedinger is a principal technical architect in SAS Professional Services (also known as "consulting"). He is also coauthor of *SAS For Dummies* and writes The SAS Dummy blog at http://blogs.sas.com/sasdummy.

Learn more about this author by visiting his author page at http://support.sas.com/hemedinger. There you can download free chapters, access example code and data, read the latest reviews, get updates, and more.