

## **Dynamic Prompts Make Data Cascading Easy: Introducing New Features in SAS® 9.2 Prompt Framework**

LanChien Hsueh and Diane Hatcher, SAS Institute Inc., Cary, NC

### **ABSTRACT**

The SAS 9.2 prompt framework provides a consistent and easy user interface for creating prompts that can be used across applications and platforms. The prompt framework is available across all desktop and Web clients in the platform for SAS® Business Analytics. A new feature in SAS 9.2, dynamic cascading prompts, enables users to create custom prompts that easily extract necessary information based on previously entered prompted values. This paper will highlight dynamic prompts and give an overview of new or enhanced features such as cascading prompts, relative date and time prompts, and range prompts.

### **INTRODUCTION**

The flexibility of the platform for SAS Business Analytics can be surfaced to end users through the use of prompts. Prompts can be used to drive query generation, guide a user through what-if analyses, or determine the best approach for displaying content. The more robust the prompting framework, the more you can help users find the answers they need. In SAS 9.2, a new prompt framework is introduced to provide numerous options for report creators and analytical model developers to present relevant information to users within a single report or application.

The SAS 9.2 prompt framework has many new and powerful features such as dynamic prompts, cascading prompts, relative date and time prompts, range prompts, and shared prompts. This paper will use examples from three different client applications: SAS® Web Report Studio, SAS® Management Console, and SAS® Information Map Studio. It will explain how the prompts work, advantages of using them, and how the prompts can be used across applications and platforms.

### **What Is a Prompt?**

A prompt is a mechanism to capture the answer for a specific question. The answer provided to a prompt is stored as a parameter that can be used by another question or as a filter value for a data query.

### **What Are Prompt Groups?**

Prompts can be grouped to encourage reuse, to improve the user interface for a prompted object, or to provide specific group-related functionality. In general, groups can contain any number of prompts and any number of subgroups.

### **What Is a Dynamic Prompt?**

A dynamic prompt is used to surface a list of possible values that result from a query performed at the time the prompt is presented to an end user. A dynamic prompt allows the lookup of possible prompt values from a data source such as a SAS® data set or a SAS® Information Map.

In SAS 9.2, dynamic prompts support values from the following types of data:

- relational table
- relational information map

## What Are Cascading Prompts?

Cascading prompts are dynamic prompts that subset the possible prompt values based on the value of a previously answered prompt or set of prompts.

Prompts can cascade from any number of other prompts and to any number of subsequent prompts. For example, you might want users to enter the type of computer that they want and the type of user that they are before giving them a set of computer systems to choose from. The prompts that you create might include the following:

- Enter the type of computer: PDA, Laptop or Desktop
- Enter the type of user: Business or Home
- Select a system: <subset by above selections>

### Example 1

This example is a sample Java application using dynamic and cascading prompts to enable end users to cascade from one prompt to another. We use the sample application to create three dynamic prompts for student information: AgeLimit, Height, and name. We then define dependencies between these prompts so that the three prompts are interrelated and have cascading ability.

At run time, a report is opened to show a single student or a group of students who meets the specified criteria. The report user might enter **14** for the AgeLimit prompt. This would subset the data to include the heights of only the students who are of age 14 or under. Then, the report user might pick **59.0** for the height of a student, which would result in the list of students who are 14 or under and have a height of 59.0 inches or less.

### About the CLASS Table

The CLASS table that is used in this example is a SAS data set that can be found in the SASHELP library. The table has five columns that contain student information: Name, Sex, Age, Height, and Weight.

A prompt has two phases: design time and run time.

- The *design-time* phase is for creating the prompts and configuring the questions that the user will answer.
- The *run-time* phase is for presenting the questions and enabling the end user to answer the questions.

### Design Time

Below is the design-time user interface that shows how we created the three dynamic prompts. The CLASS table used by this prompt design is on the left in Figure 1.

**Figure 1** CLASS Table and Cascading Dynamic Prompts

	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69	112.5
2	Alice	F	13	56.5	84
3	Barbara	F	13	65.3	98
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	112.5
9	Jeffrey	M	13	62.5	84
10	John	M	12	59	99.5
11	Joyce	F	11	51.3	50.5
12	Judy	F	14	64.3	90
13	Louise	F	12	56.3	77
14	Mary	F	15	66.5	112
15	Philip	M	16	72	150
16	Robert	M	12	64.8	128
17	Ronald	M	15	67	133
18	Thomas	M	11	57.5	85
19	William	M	15	66.5	112

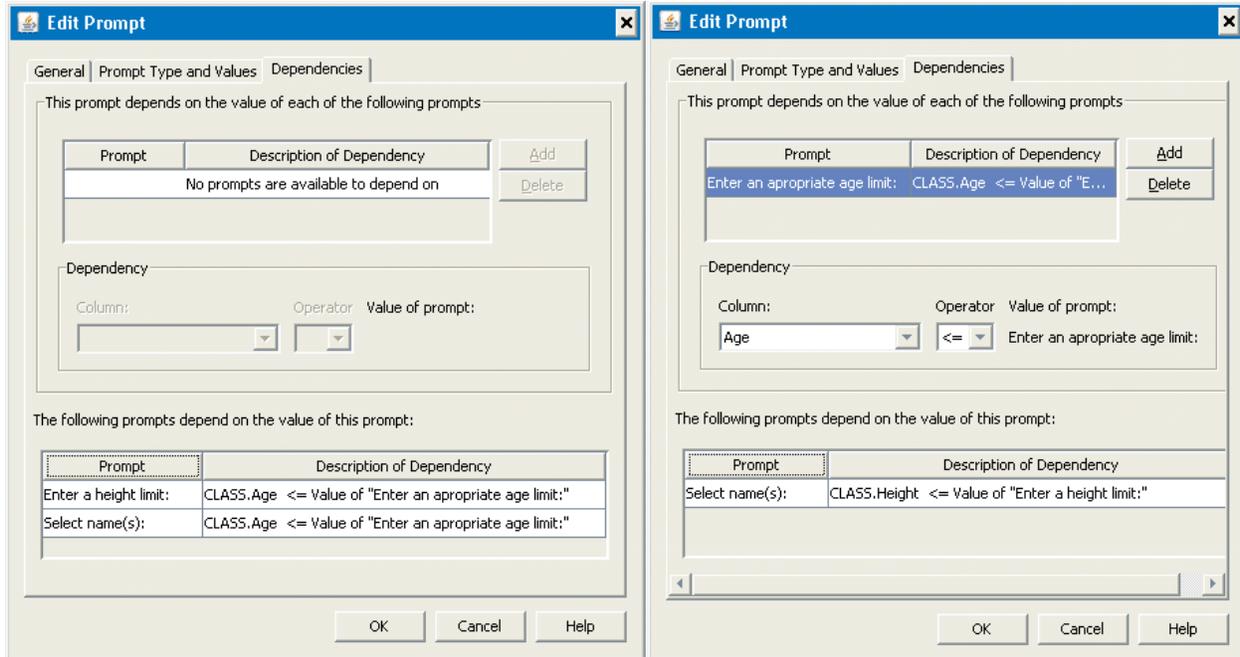
  

Displayed Text	Name	Type
General		Standard group
Enter an appropriate age limit:	AgeLimit	Numeric (integer)
Enter a height limit:	Height	Numeric (double)
Select name(s):	name	Text

## Dependencies

Figure 2 shows the user interface for defining dependencies between prompts. In the left image, the Height prompt and name prompt are dependent on the value of the Agelimit prompt. In the right image, the name prompt depends on the value of the Height prompt.

**Figure 2** Prompt Dependencies



## Run Time

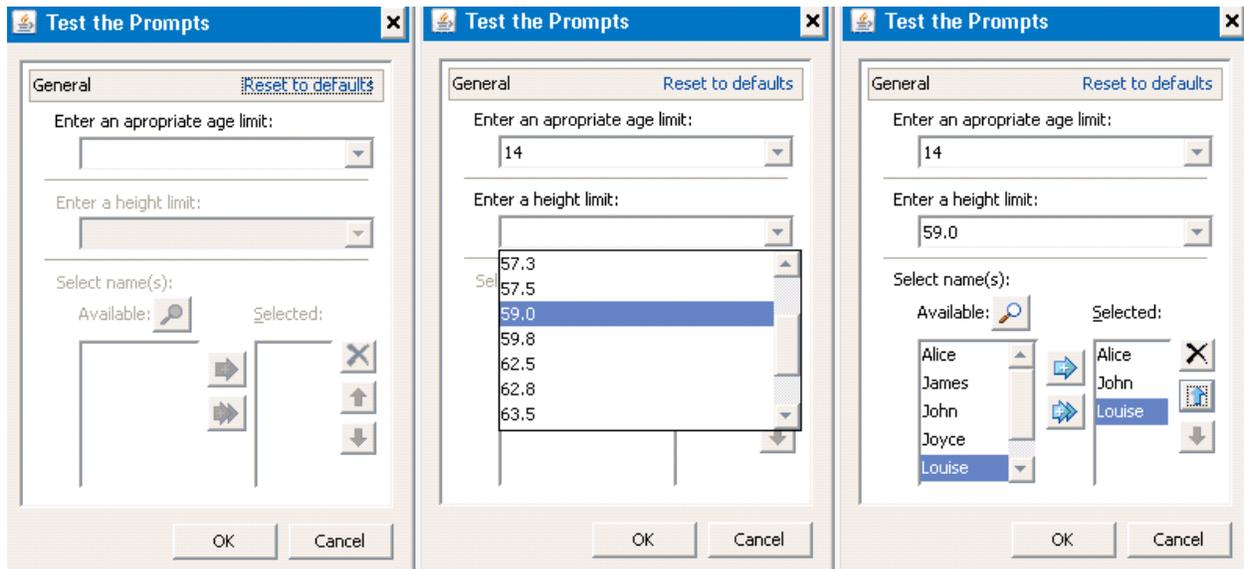
Figure 3 shows the run-time user interface of the sample Java application where a report can be run to find a single student or a group of students who meets the specified criteria.

**Step 1** When the prompts are first displayed, only the **Enter an appropriate age limit** field (for the Agelimit prompt) is shown and can be interacted with. See the first image in Figure 3.

**Step 2** Once an age limit of **14** is selected, the **Enter a height limit** field (for the Height prompt) becomes active and a list of heights for students whose age is 14 or under is available. See the middle image in Figure 3.

**Step 3** Once a height limit of **59.0** is selected, the **Select name(s)** field (for the name prompt) becomes active. A list of student names is available to be selected, representing those who are 14 or under *and* 59.0 inches or shorter. See the image on the right in Figure 3.

**Figure 3** Dynamic Cascading Prompts at Run Time

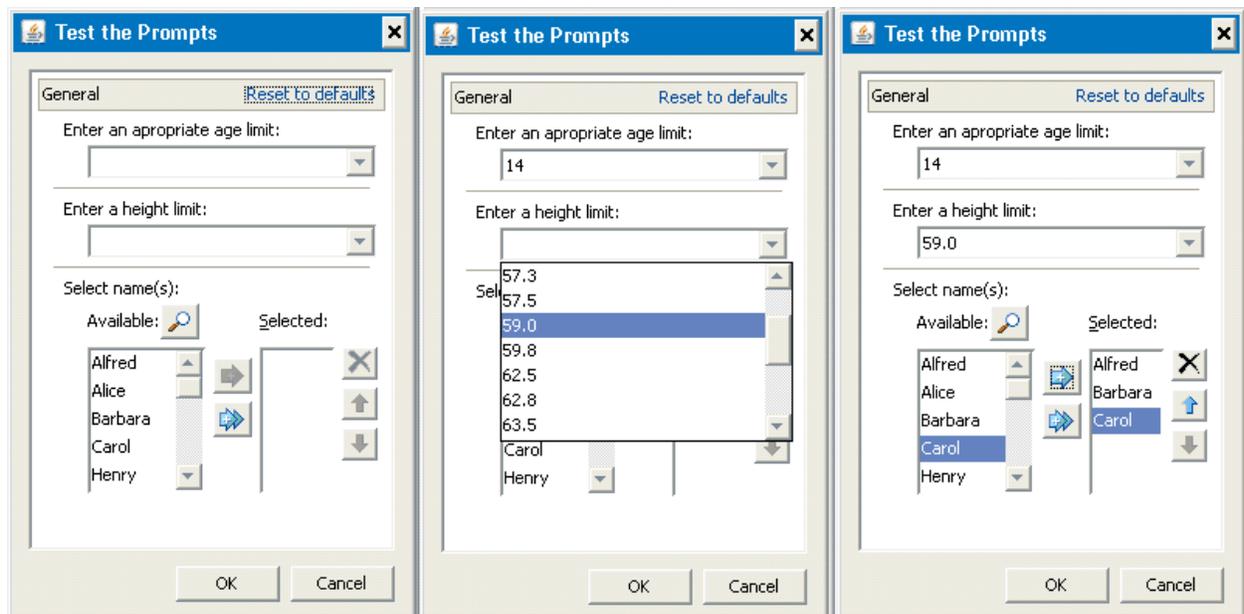


**Comparison with Static Prompts**

Static prompts cannot have cascading relationships applied to them. Before SAS 9.2, it was not possible to have anything but static prompts. Figure 4 shows how the same set of prompts would operate without the power to narrow down the selection list.

- Step 1** When the static prompts first come up, notice that all three prompts are active. There is no interrelationship between these three prompts. See the first image in Figure 4.
- Step 2** If an age limit is selected, the **Enter a height limit** prompt does not have the ability to narrow down to a list of heights of students whose ages are under 14. See the second image in Figure 4.
- Step 3** The **Select name(s)** prompt is also not able to subset its values. The list contains the names of all students. See the third image in Figure 4.

**Figure 4** Static Prompts at Run Time



## Example 2

This example defines a SAS® Stored Process that uses cascading prompts in SAS Management Console.

Using SAS Management Console, input parameters for stored processes are defined as prompts. You can add prompts or prompt groups when you are using the New Stored Process wizard to register a new stored process or when you are viewing properties for a currently registered stored process.

In this example, we create a stored process named cascading1. The stored process presents end users with three prompts that use a CARS data table: origin, make, and invoice. The make prompt cascades from origin, and invoice cascades from make. Once all three prompts have been selected, the end users can execute the stored process and obtain results based on the input parameters. The code for this example stored process includes a simple PRINT procedure call which creates an HTML table that is output back to the client.

### About the CARS Table

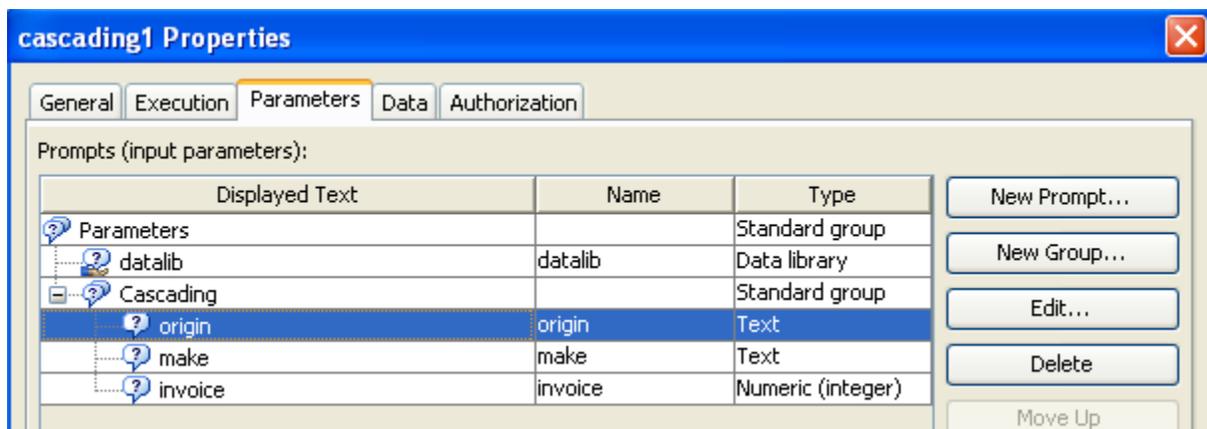
The CARS table is a sample table that contains three columns:

- the Origin column represents the place of manufacture
- the Make column represents the make of a car
- the Invoice column represents the invoice price of a car

### Design Time

Figure 5 shows the user interface in SAS Management Console for designing prompts in stored processes.

**Figure 5** SAS Management Console User Interface for Designing Prompts in a Stored Process



### The Associated Stored Process Source Code

Below is the SAS code that is used for the stored process. Note that the datalib prompt is a shared hidden prompt that is assigned to reference the user-defined library in the metadata server.

**Note:** A hidden prompt is a prompt that is hidden from the end user. The functionality of a hidden prompt is transparent to an end user. For the definition and usage of a *shared* prompt, refer to Example 6.

- \* The testing code uses the PRINT procedure with WHERE clause and ODS
- \* to generate an output based on the client request;

```
LIBNAME &DATALIB meta library="&DATALIB_PATH";
```

```
* Declare reserved server environment macro variables;  
%GLOBAL ORIGIN MAKE INVOICE;
```

```
* Declare macro variables used by this program;  
%LET DATASET = &DATALIB..CARS;
```

```
*ProcessBody;
```

```

%STPBEGIN;
  data a;
    format titledt dollar8.;
    titledt = &invoice;
    call symput('title',put(titledt,comma8.));

  run;

  proc print data = &DATASET noobs label n;
    where (origin = "&origin" AND make = "&MAKE" AND invoice <= &INVOICE);
    format invoice DOLLAR8.;
    title bold bcolor='#c0c0c0'"CAR Info - Origin: &origin/Manufacture:
&make(Invoice <= $&title)";

  run;

%STPEND;

```

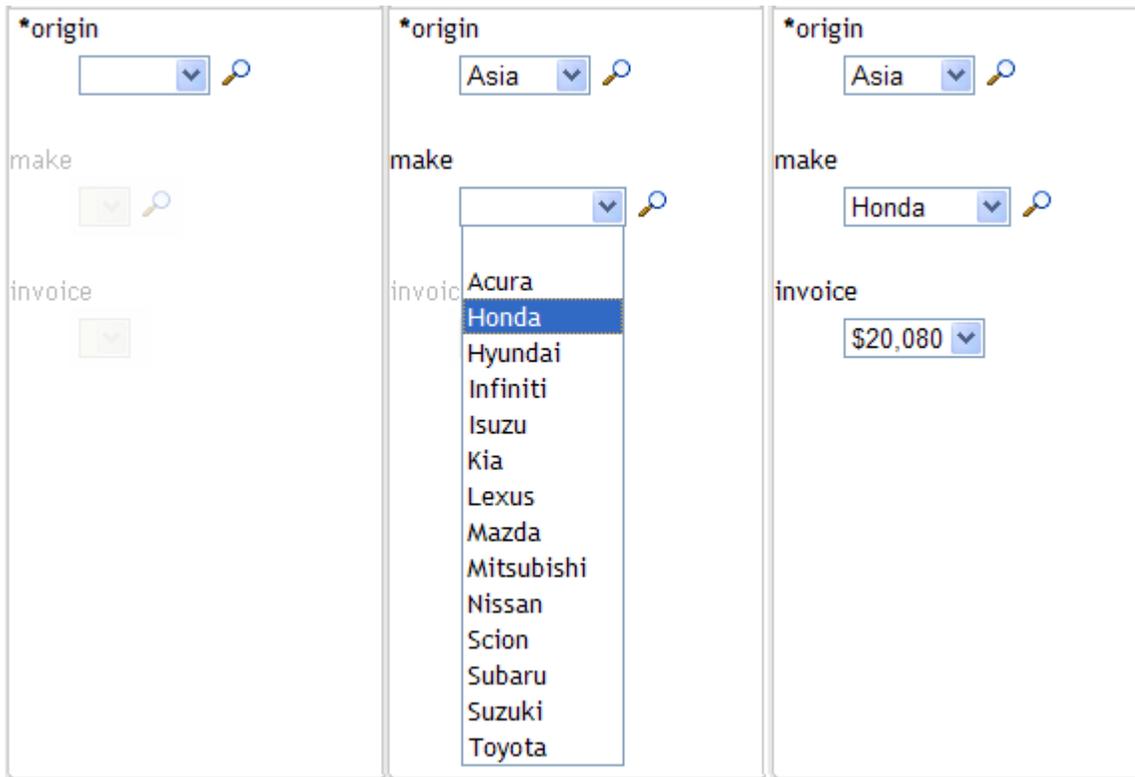
### Run Time

For this example, we use the SAS® Stored Process Web Application to run the cascading1 stored process we created above. We have set up the execution options to stream the results back to the client (which is the SAS Stored Process Web Application in this case). The SAS Stored Process Web Application is included as part of SAS® Integration Technologies.

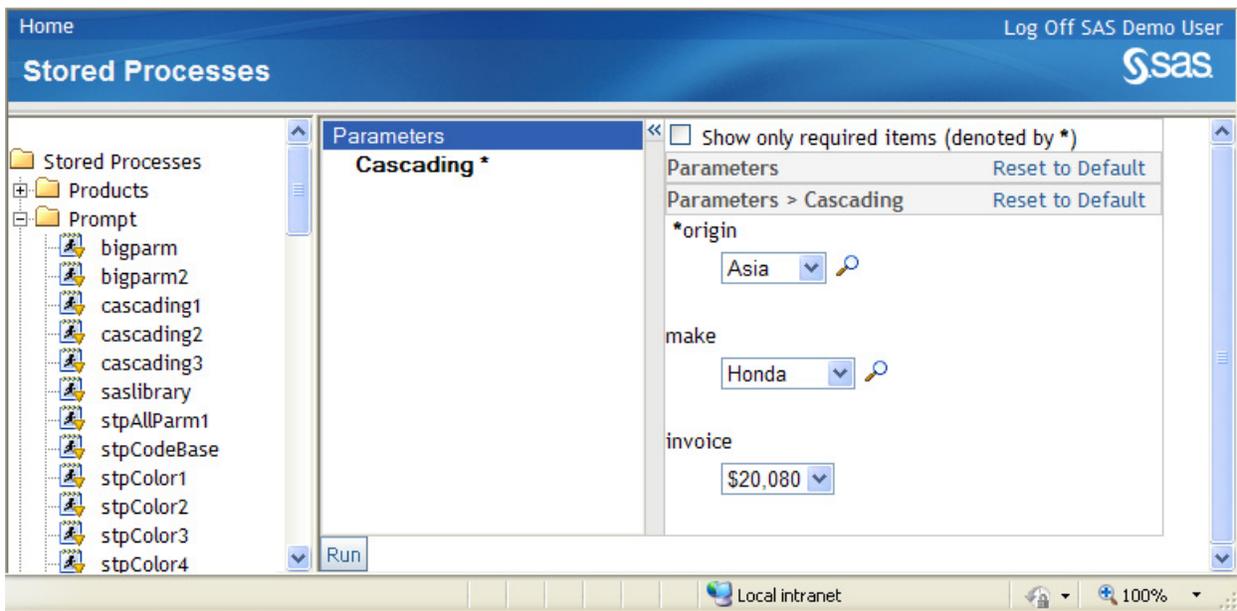
When the cascading prompts for the stored process are displayed, only the **origin** prompt is active. Once a user selects a value (for example, **Asia**), then the **make** prompt becomes available, allowing the user to choose from the car makes that are available in Asia. Once a make is selected (for example, **Honda**), the user can select from the list of invoice values for Hondas in Asia. This is shown in Figures 6 and 7.

Once an invoice value is selected (for example, **\$20,080**), then the user can click **Run** at the bottom of the SAS Stored Process Web Application to display the results. A new window will display the stored process results based on the user's input. The results are shown in Figure 8.

**Figure 6** Step-by-Step Execution of Cascading Prompts in the SAS Stored Process Web Application



**Figure 7** Selected Values for Cascading Prompts in the SAS Stored Process Web Application



**Figure 8** Stored Process Results Based on Invoice Values Equal To or Less Than \$20,080 for Hondas in Asia

**CAR Info - Origin: Asia / Manufacture: Honda (Invoice <= \$ 20,080)**

Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	Engine Size (L)	Cylinders	Horsepower
Honda	Civic Hybrid 4dr manual (gas/electric)	Hybrid	Asia	Front	\$20,140	\$18,451	1.4	4	
Honda	Insight 2dr (gas/electric)	Hybrid	Asia	Front	\$19,110	\$17,911	2.0	3	
Honda	CR-V LX	SUV	Asia	All	\$19,860	\$18,419	2.4	4	
Honda	Element LX	SUV	Asia	All	\$18,690	\$17,334	2.4	4	
Honda	Civic DX 2dr	Sedan	Asia	Front	\$13,270	\$12,175	1.7	4	
Honda	Civic HX 2dr	Sedan	Asia	Front	\$14,170	\$12,996	1.7	4	
Honda	Civic LX 4dr	Sedan	Asia	Front	\$15,850	\$14,531	1.7	4	
Honda	Accord LX 2dr	Sedan	Asia	Front	\$19,860	\$17,924	2.4	4	
Honda	Accord EX 2dr	Sedan	Asia	Front	\$22,260	\$20,080	2.4	4	

**Example 3**

This example uses SAS Information Map Studio and the prompting framework to create a compound filter with cascading rules in its expression. The prompts in this example are designed to allow multiple values to be selected each time, and the last prompt is set to allow the user to reorder the final selections.

SAS Information Map Studio is a desktop client application that is used to create and manage information maps. Information maps are essentially a semantic layer that describes underlying physical data structures in terms that non-technical end users can understand. Information maps contain data items (referencing specific elements from a data source or computational expressions) and filters. When a user accesses an information map, they can select data items and filters, which resolve to an SQL expression. The SQL is generated on behalf of the user without requiring him or her to understand data structures or SQL syntax. This allows you to provide consistent and expected results, regardless of who is running the query.

In SAS Information Map Studio, a user can use the Manage Prompts dialog box to create, add, edit, or delete prompts for an information map. This example uses an information map called MailOrder, which is based on a single data source.

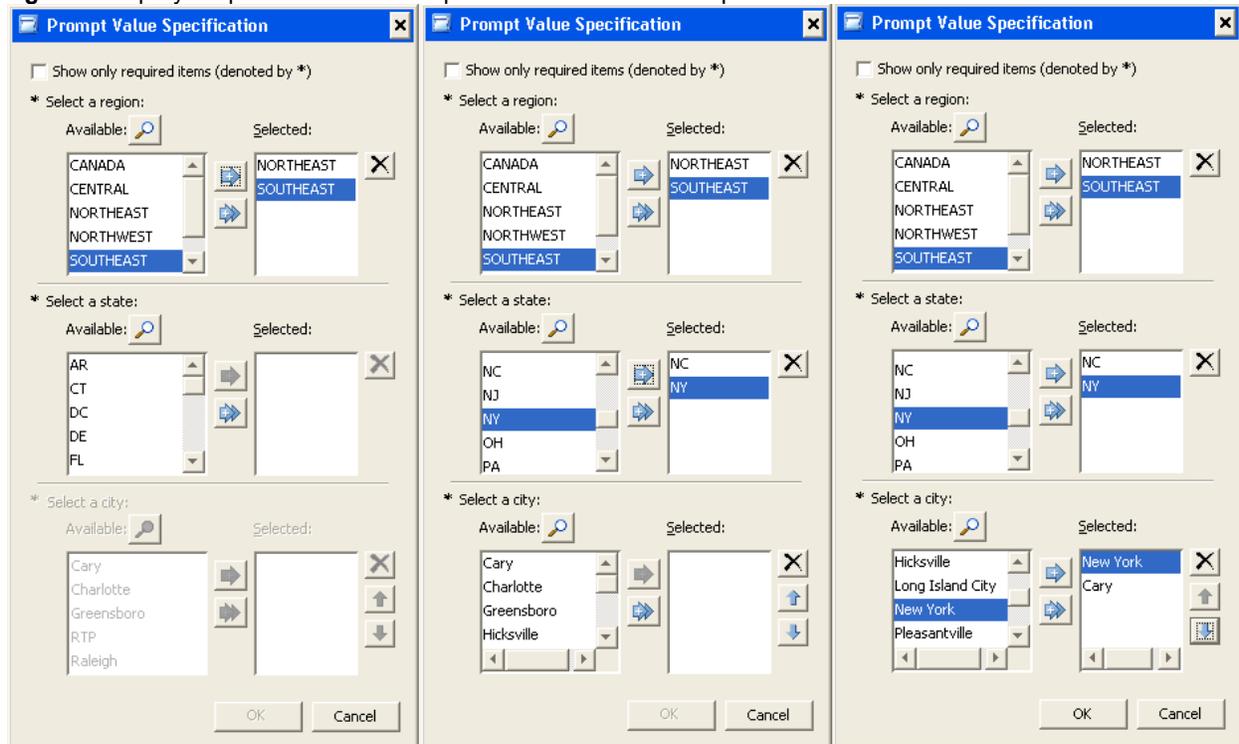
**About the MailOrder Information Map**

The MailOrder information map is based on a data set that contains a subset of four columns from a customer database in the United States and Canada:

- the Region column represents the region where a customer resides
- the State column represents the state where a customer resides
- the City column represents the city where a customer resides
- the Name column represents the name of a customer

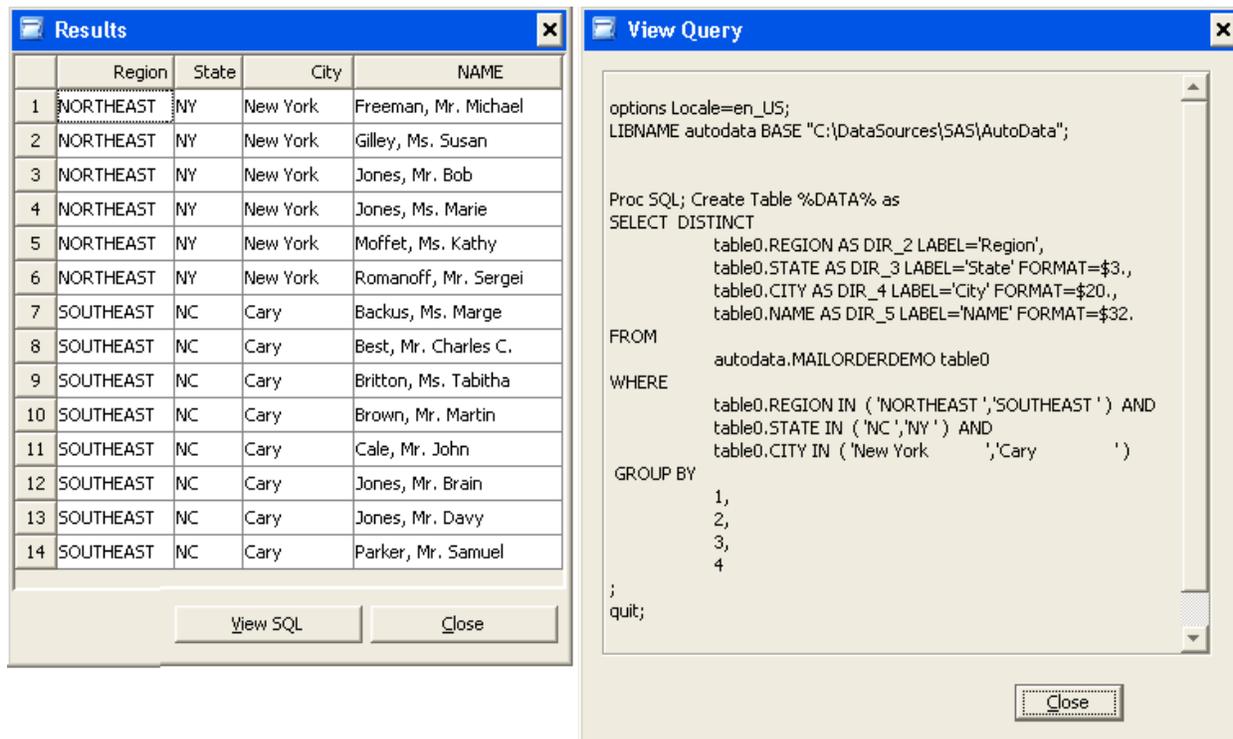
Figure 10 shows the flow of the prompts that are cascaded via the rules in the filter expression. An end user can select from one or more regions. Based on the selected regions, the user can select from one or more states. Based on the selected states, the user can select from multiple cities. Then the user can click **OK** to see the results based on the user's selections.

**Figure 9** Step-by-Step Execution of Prompts in SAS Information Map Studio



The Results dialog box in Figure 10 shows the final subsetted result set after the filter has been applied. If the user clicks the **View SQL** button, then the View Query dialog box is displayed (see right side of Figure 10). As you can see in the View Query dialog box, the query has already resolved the values of the prompts in the WHERE clause.

Figure 10 Results Dialog Box and View Query Dialog Box



#### Example 4

This example highlights using SAS Web Report Studio 4.2 to create a report that uses cascading prompts in filters.

#### Defining and Applying Prompted Filters

In SAS Web Report Studio, a prompted filter can be included in a cascade if it meets the following requirements:

- The prompted filter is based on a relational data source.
- The data item associated with the prompted filter allows the user to see its values. (The data source creator sets this in the information map using SAS Information Map Studio.) To verify, in SAS Information Map Studio select **Properties** when you right-click a data item. On the **Value-Generation Method** tab, the **User selects values from a dynamic list** option should be selected.
- The prompt type of the prompted filter allows the values of the data item that is associated with the prompted filter to be loaded dynamically. In SAS Web Report Studio, a user can set the prompt type by selecting **Dynamically generated values** in the **Prompt type** field on the Custom Filter dialog box (see Figure 11).

A non-prompted filter can be included in a cascade if it begins the cascade; however, it will not appear in the prompts page.

Prompted filters defined in either SAS Web Report Studio or SAS Information Map Studio can be used in a cascade. You have to select them in the Section Filters dialog box to participate in the cascade (see Figure 12).

Figure 11 Prompted Filter in the Custom Filter Dialog Box

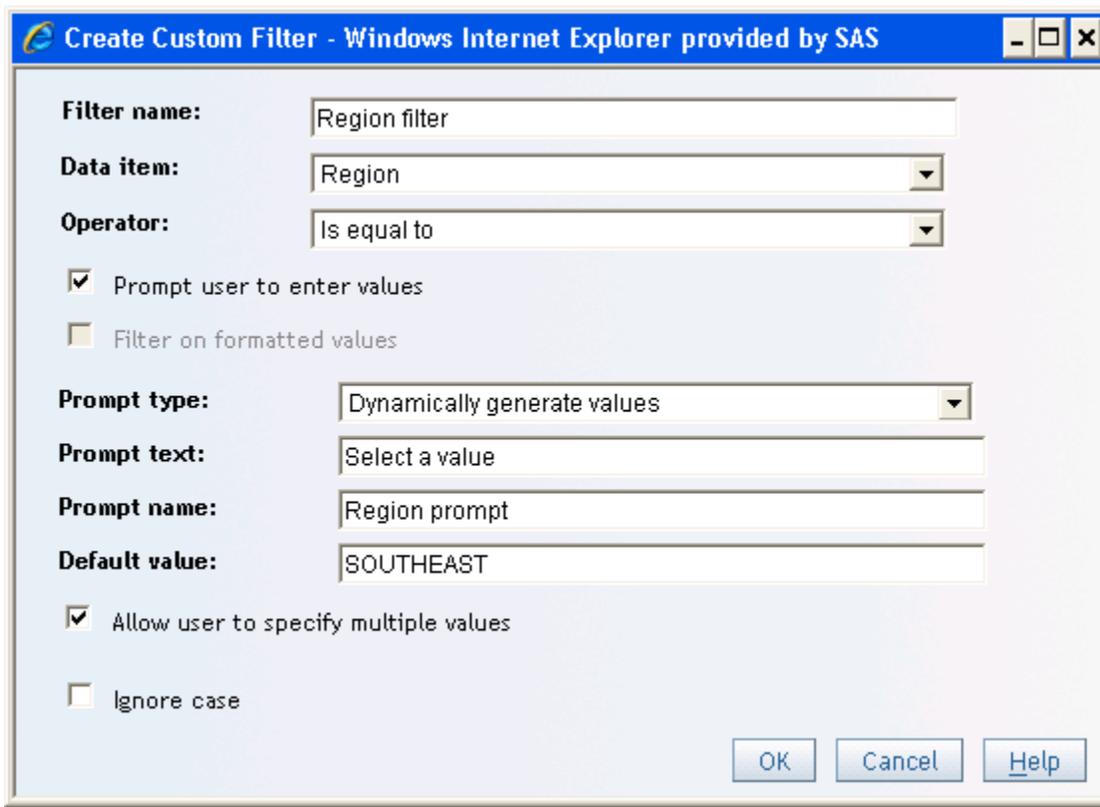


Figure 12 Section Filters Dialog Box with Prompted Filters for Region, State, and City

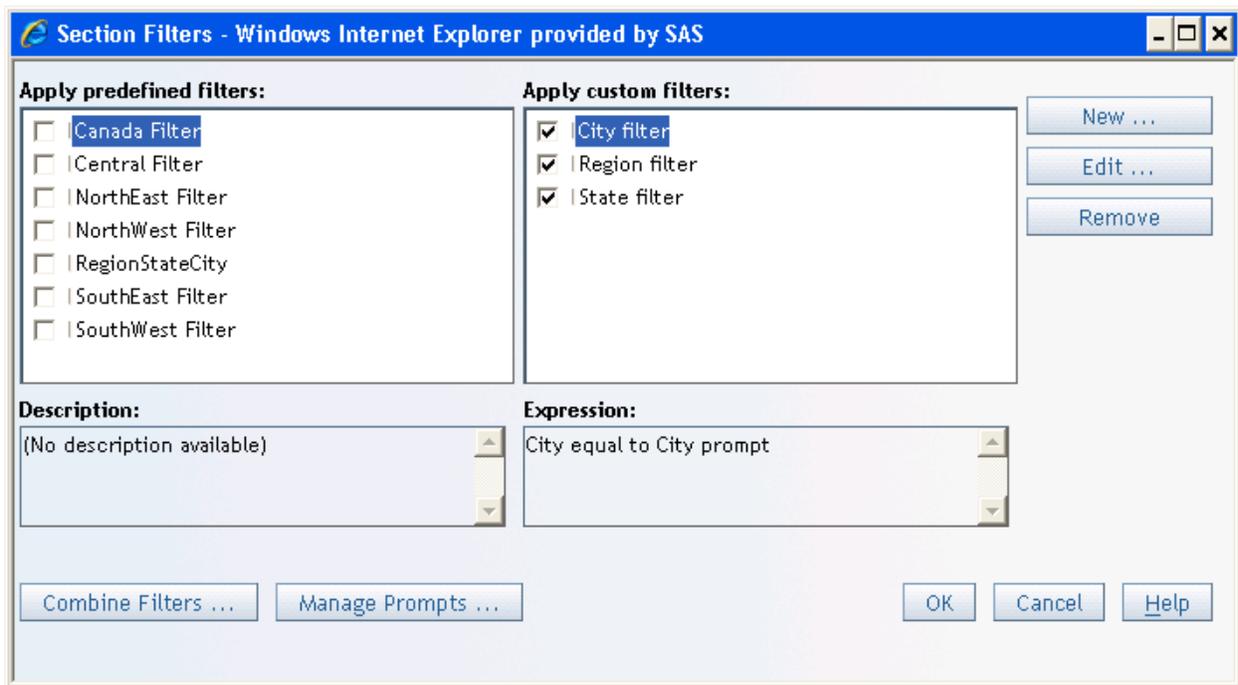
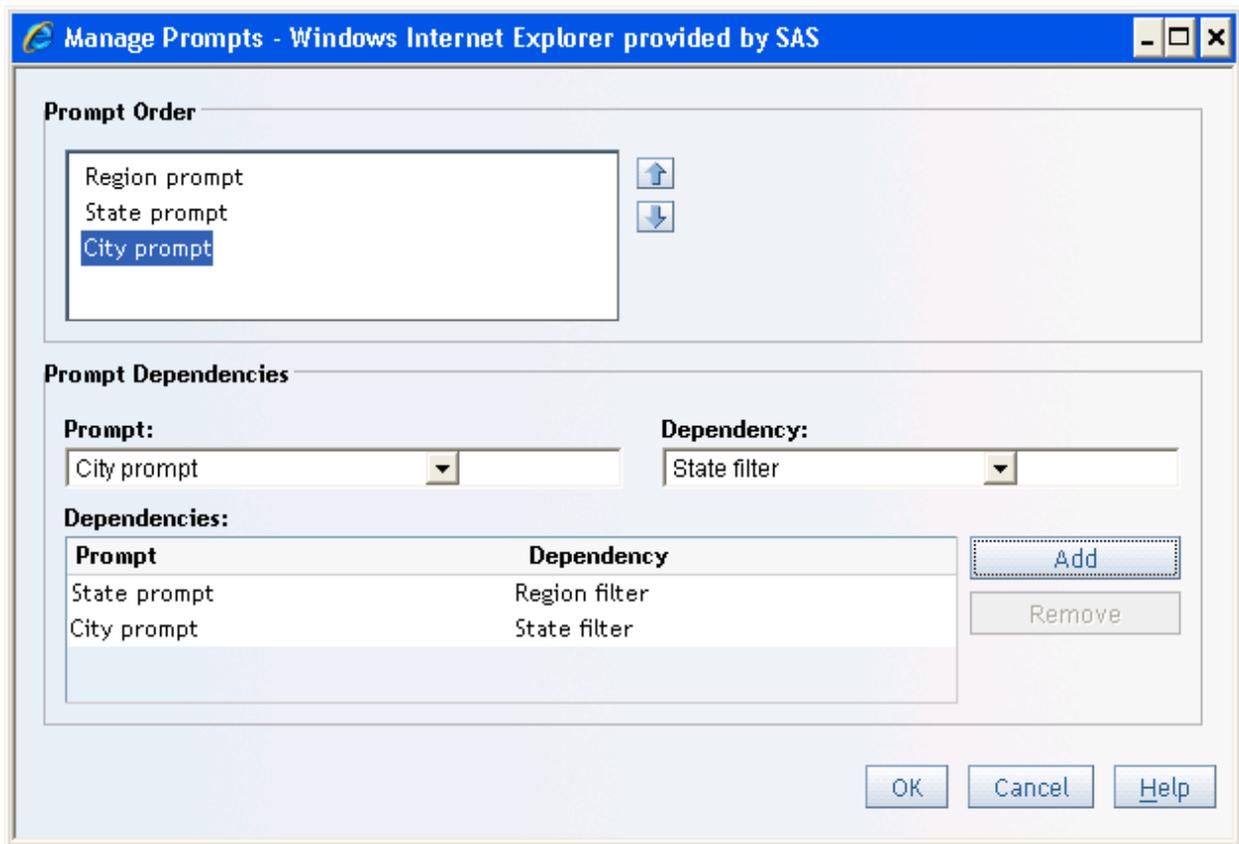


Figure 13 Manage Prompts Dialog Box with Prompts for Region, State, and City



The selection of values for Region, State, and City are similar to the selections made in SAS Information Map Studio in Example 3. After all selections are made, a user can click **View Report** to see the report that is based on the results of the cascaded prompts.

### What Are Relative Date and Time Prompts?

Relative date and time prompts allow you to set prompt values based on relative dates and times such as the following:

- today
- current day of last year
- three hours ago

### What Are Range Prompts?

One of the new features possible with the SAS 9.2 prompting framework is the range prompt. Range prompts allow a user to enter both a minimum and a maximum value for a prompt. So a user can use one prompt to implement a range of dates or other variables without having to create two prompts. For example, a report designer can enable a user to select and report on a range of dates such as February 18, 2008 to February 28, 2008 (see Figure 16 in Example 5).

The following types of ranges are supported:

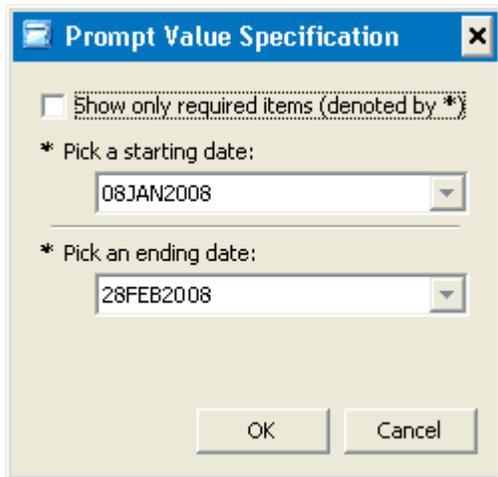
- numeric integer
- numeric double
- date
- time
- timestamp
- text string

**Note:** The values for a range prompt are not determined by the minimum and maximum functions for a variable but by specifically defined settings.

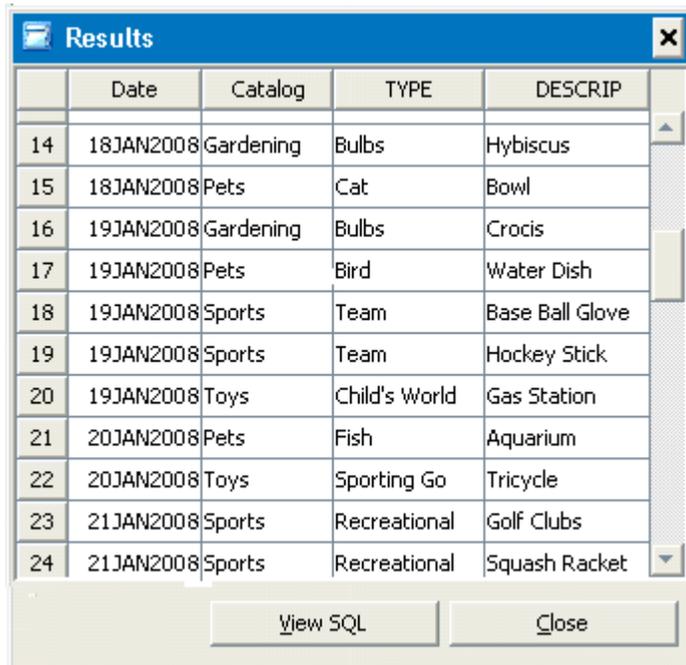
### Example 5

This example shows two ways to prompt for date ranges in SAS Information Map Studio. One way is to create two date prompts to implement a range of date values. In Figure 14, the end user can pick a starting date in the first date prompt, and then pick an ending date in the second prompt. Figure 15 shows the resulting output.

**Figure 14** Date Range Specified Using Two Date Prompts



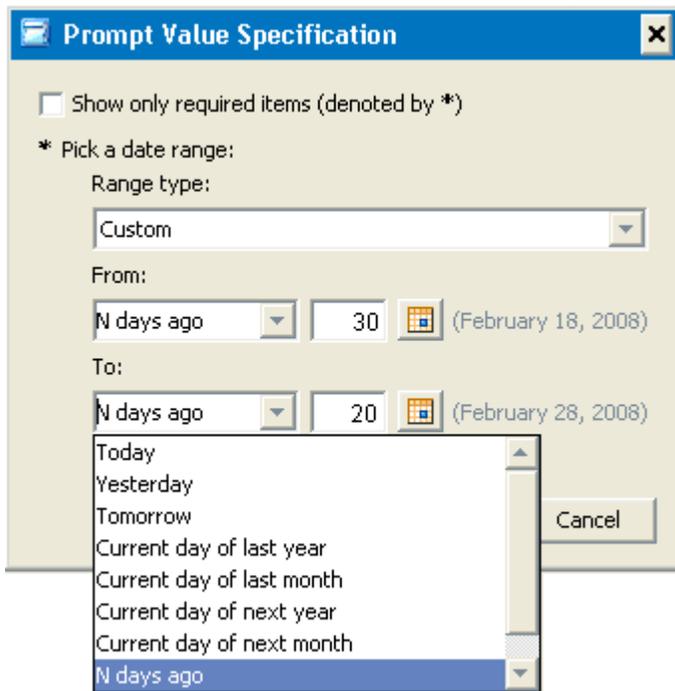
**Figure 15** Results from Using Two Date Prompts



	Date	Catalog	TYPE	DESCRIP
14	18JAN2008	Gardening	Bulbs	Hybiscus
15	18JAN2008	Pets	Cat	Bowl
16	19JAN2008	Gardening	Bulbs	Crocis
17	19JAN2008	Pets	Bird	Water Dish
18	19JAN2008	Sports	Team	Base Ball Glove
19	19JAN2008	Sports	Team	Hockey Stick
20	19JAN2008	Toys	Child's World	Gas Station
21	20JAN2008	Pets	Fish	Aquarium
22	20JAN2008	Toys	Sporting Go	Tricycle
23	21JAN2008	Sports	Recreational	Golf Clubs
24	21JAN2008	Sports	Recreational	Squash Racket

In Figure 16, a date *range* prompt enables a user to select a range type and easily define the range in a single prompt. Notice that the user can choose to select relative dates such as **N days ago** or **Today**.

**Figure 16** Date Range Specified Using a Single Date Range Prompt



**Figure 17** Results from Using a Single Date Range Prompt

	Date	Catalog	TYPE	DESCRIP
1	18FEB2008	Gardening	Seeds	Radishes
2	18FEB2008	Pets	Cat	Bowl
3	19FEB2008	Sports	Recreational	Roller Blades
4	19FEB2008	Sports	Recreational	Water Skis
5	21FEB2008	Pets	Cat	Toy
6	22FEB2008	Pets	Bird	Water Dish
7	22FEB2008	Pets	Cat	Litter Box
8	23FEB2008	Pets	Dog	Toy
9	24FEB2008	Gardening	Hand Tools	Ax
10	24FEB2008	Gardening	Hand Tools	Edger

## What Is a Shared Prompt?

A shared prompt is a prompt that is stored in a shared location and can be accessed by multiple users, applications, and software features. A shared prompt is created by first creating a prompt and then saving that prompt as a shared prompt. The shared prompt can then be added one or more times to the list of prompts in your application or software feature.

After you add a shared prompt, you can modify the values of the following prompt properties only:

- name (used by a stored process as a parameter name)
- displayed text (prompt label)
- description
- parent group

**Note:** Only certain applications, such as SAS Management Console, and certain software features can save shared prompts, but any application or software feature that supports shared prompts can add and use them. For example, SAS<sup>®</sup> Enterprise Guide cannot save shared prompts, but it can add and use shared prompts. A shared prompt is stored in metadata in a SAS folder. When you add the shared prompt to the list of prompts in your application or software feature, you can provide your own name, displayed text, and description for the prompt. These changes do not affect the information that is stored for the prompt in the SAS folder; these changes exist only for your particular view of that shared prompt. All other prompt properties can be modified only when you open a shared prompt from its SAS folder. Changes that are made when you open a shared prompt from its SAS folder will be seen by all consumers of the shared prompt.

## What Are Shared Groups?

Groups of prompts can also be shared. Shared prompt groups are useful for the following cases:

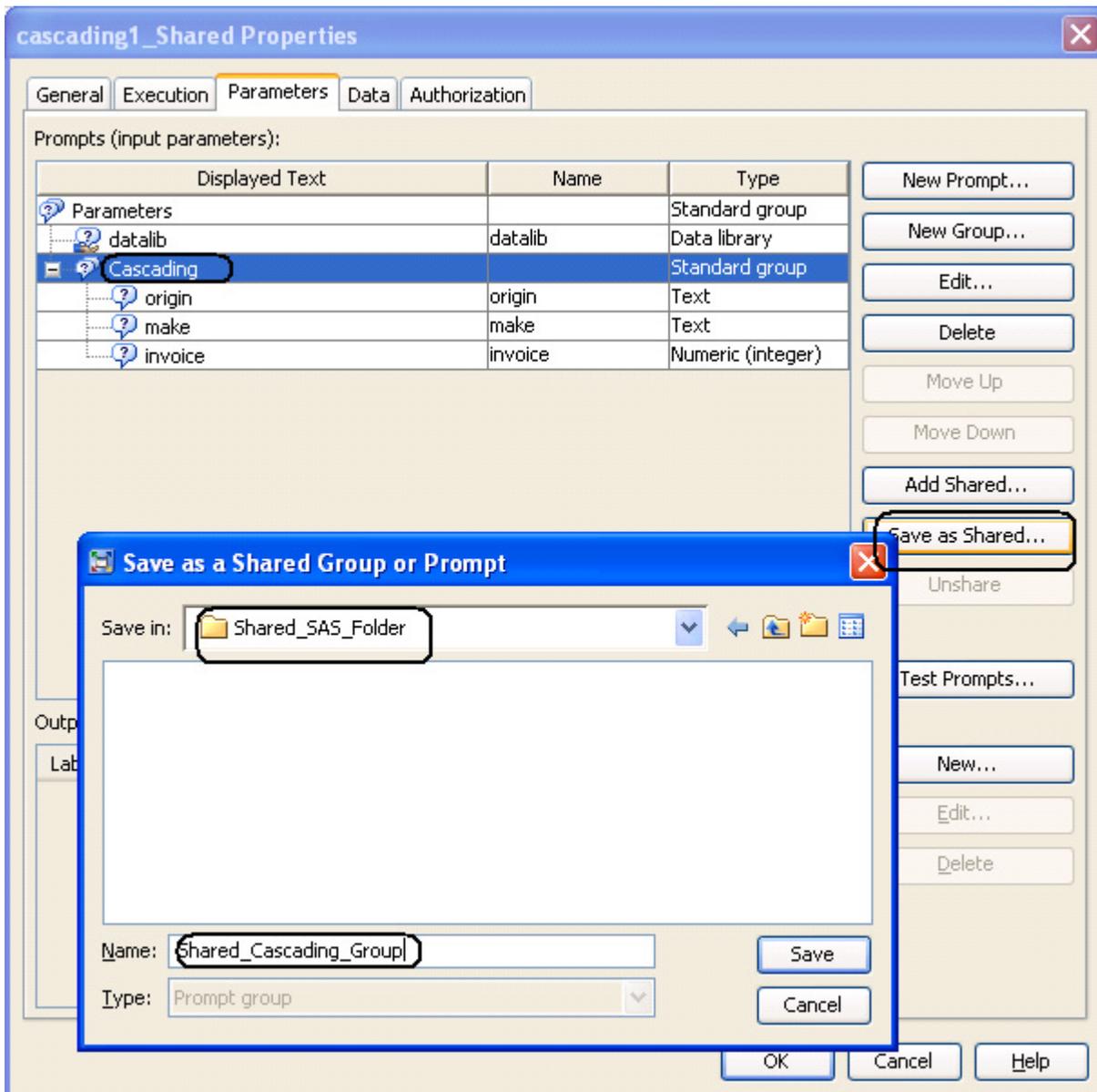
- dynamic prompts with complex configurations
- sets of cascading prompts
- groups of prompts that are often reused (for example, chart options)

## Example 6

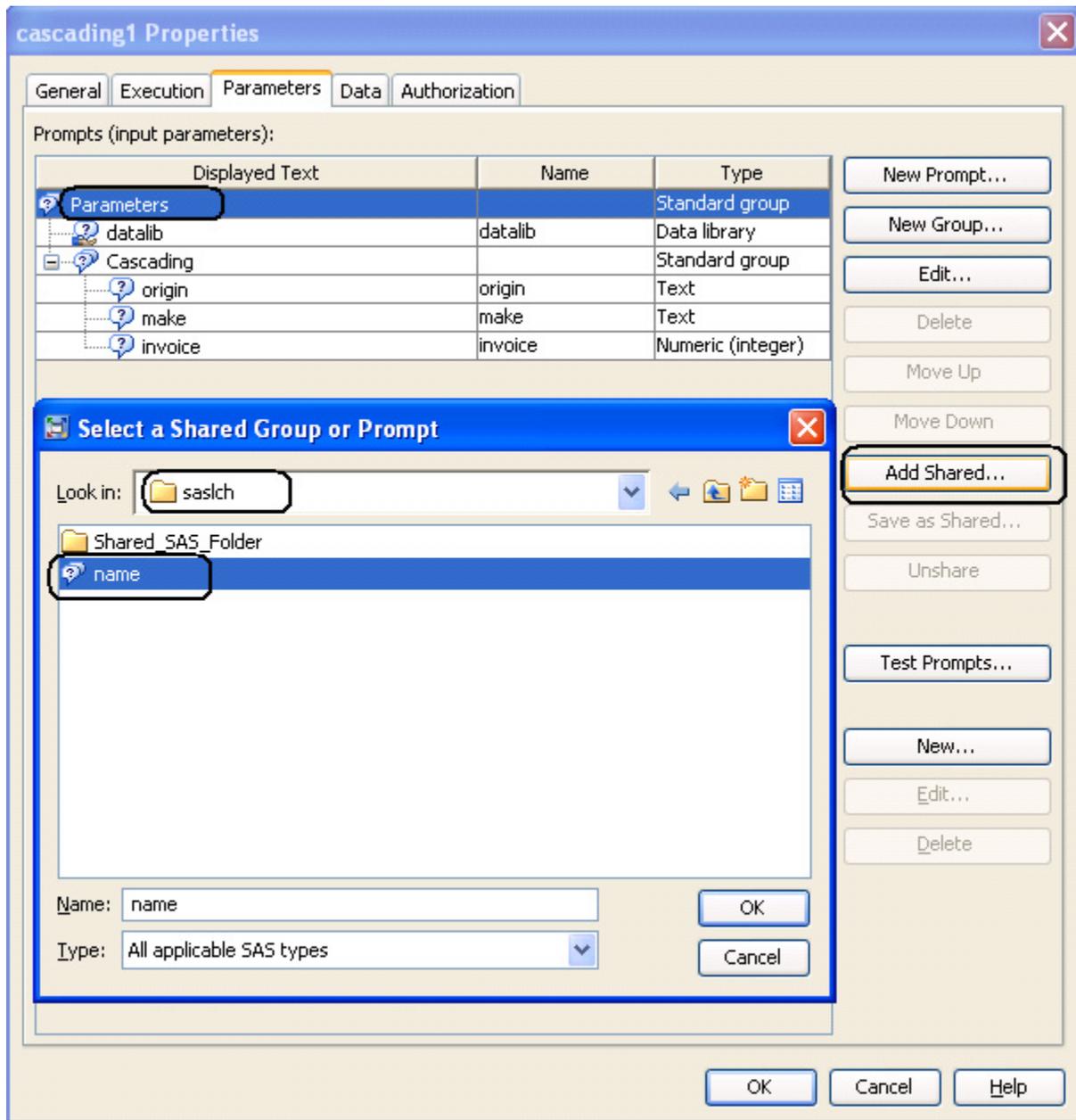
In this example, we continue from Example 2. A user can copy the cascading1 stored process that is created in SAS Management Console to a new stored process named cascading1\_Shared. In the **Parameters** tab for **cascading1\_Shared**, the user can click **Save as Shared** to save a prompt group named **Cascading** to a shared SAS folder named **Shared\_SAS\_Folder**. The user names the shared group **Shared\_cascading\_group** (see Figure 18).

A user can also use the **Add Shared** button to add and use prompts and groups from a shared location (see Figure 19).

Figure 18 Saving a Prompt Group as a Shared Group



**Figure 19** Adding a Shared Prompt Group to a Stored Process



## CONCLUSION

The prompt framework is available across all desktop and Web clients in the platform for SAS Business Analytics. In addition to the examples covered in the paper for SAS Web Report Studio, SAS Information Map Studio, and SAS Management Console, the prompt framework also applies to SAS® Information Delivery Portal, SAS Enterprise Guide, SAS® Add-in Microsoft Office, and SAS® Data Integration Studio. The prompt framework will also be integrated into new releases of other SAS solutions as they are delivered for SAS 9.2.

## **ACKNOWLEDGMENTS**

Special thanks to the following people for their contributions to this paper: Scott Leslie, Pat Berryman, Tracy Ellis, Kenneth Parker, Dan Hils, Tammy Gagliano, Drew Anderson, Ken McAdams, Todd Dalness, Trevor Butler, Zhenrong Liu, Vincent DelGobbo, Leon (Yinliang) Wu, Lihua Liu, Meenu Jaiswal, Venkata Upparapati, Vishal Bhavsar, Anita Pundrotu, Richa Sharma, Anuja Yadwad, Zhirong Liu, Miao Li, Houda Moudouni, Weston McManus, Joe Sumpter, Helena Pearce

## **CONTACT INFORMATION**

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

LanChien Hsueh  
SAS Campus Drive  
SAS Institute Inc.  
E-mail: [LanChien.Hsueh@sas.com](mailto:LanChien.Hsueh@sas.com)

Diane Hatcher  
SAS Campus Drive  
SAS Institute Inc.  
E-mail: [Diane.Hatcher@sas.com](mailto:Diane.Hatcher@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.