Changes and Enhancements

Introduction

This section describes the features of SAS Component Language that have been implemented or enhanced since Release 6.12. Information about changes and enhancements that were implemented in Version 8 is preceded by V8. All other changes and enhancements described in this section were implemented in Version 7. In other words, if your site upgraded from Version 7 to Version 8, then only the items preceded by V8 are new to you.

Name Change to SAS Component Language

SAS Component Language is the new name of what was known in Version 6 as SAS Screen Control Language. A component is a self-contained, reusable object with specific properties that include a set of public attributes, public methods, event handlers that execute in response to various types of events, and a set of supported interfaces.

Long Names

Names (except for libref and fileref names) can contain up to 32 characters. This includes names of SCL variables, arrays, SCL lists, SAS tables, views, indexes, catalogs, catalog entries, macros, and macro variables.

Scope of Variables

SCL variables can have the local scope of a DO or SELECT block.

Shortcut for Invoking Methods

A new feature called dot notation provides a shortcut for invoking methods and for setting or querying attribute values. Using dot notation reduces typing and makes SCL programs easier to read.

Dot notation cannot be used within DATA step INPUT or PUT functions.

New Data Types

SCL provides the new data types NUM, CHAR, and LIST, as well as new object types. NUM and CHAR are new as named data types, although these data types have been available in previous versions as unnamed data types (for example, $). LIST and object types store identifiers for SCL lists and components, respectively. The object types can be specific (CLASS or INTERFACE) or generic (OBJECT).

V8 Version 8 introduces another specific object type, CLASSPKG.

You declare a variable to have a specific object type by specifying a class name or interface name as the variable type. For example:

```
dcl sashelp.fsp.object.class object1;
```

If the object type of the variable cannot be determined at compile time, you can declare a variable to have a generic object type by using the named type OBJECT. For example:

```
dcl object object2;
```

Terminology Change

To be more consistent with database management terminology, SAS Component Language now uses the terms *table*, *row*, and *column* in place of *data set*, *observation*, and *variable*. However, you may still encounter the old terms in some SAS products and documentation.

Integrity Constraints for SAS Tables

SCL provides a set of functions that enable you to work with integrity constraints (IC). Integrity constraints are a Version 7 feature of SAS software for preserving the consistency and correctness of data that is stored in SAS tables. When integrity constraints are assigned to a table, they are automatically enforced for each addition, update, and deletion action on that table.

The new IC functions are ICCREATE, ICDELETE, ICTYPE, and ICVALUE.

SCL Debugger

The SCL debuggerincludes new functionality that enables you

- to execute SCL functions through the CALC command
- to use dot notation in any debugger command that takes an expression or variable as an argument.

Image Functions

The SCL functions that control Image objects in the Image Extensions to SAS/GRAPH software are now documented in Appendix 1, "Commands Used with the IMGCTRL, IMGOP and PICFILL Functions," on page 735. New Image commands for the IMGOP function are

FILTER
: allows user-provided Convolution Filters. This support is available in the Image Editor, in the SAS/AF Frame Image Data Object, and in Image SCL functions.

GET-BARCODE
: decodes a bar code in an image and returns the value of the bar code. It can be used with the Image data object.

V8 PRINT
: prints an image. This command is available for the Frame Image Data Object and the Image SCL functions.

V8 The READ command now supports TWAIN scanners and cameras in the Windows operating environments.

New Language Elements

For details about these new language elements, see Chapter 16, "SAS Component Language Dictionary," on page 249.

V8 CATCH statement
: defines the actions that SCL takes when a specific exception is raised.

CLASS statement
: enables you to use SCL to create a SAS/AF class and to define all the properties for the class, including attributes, methods, events, and interfaces.

V8 COMPAREARRAY function
: enables you to compare two arrays for size and data equality.

COMPARELIST function
: compares two SCL lists. This comparison can include item names, values, or both.

V8 COPYARRAY function
: enables you to copy data from one array into another array.

CREATESCL function
: writes a class definition from a CLASS or INTERFACE catalog entry to a CLASS or INTERFACE statement in an SCL entry.
  V8 CREATESCL will also write a class package definition from a CLASSPKG entry to a PACKAGE statement in an SCL entry and generate proxy classes.

DCREATE function
: creates a directory on the user's host operating system.

DECLARE statement
: declares a variable of any type, including the new CHAR, NUM, LIST and OBJECT types. (By contrast, the LENGTH statement can declare only unnamed numeric and character variables.)

V8 DELARRAY function
: deletes a dynamic array.

DIALOG function
runs a FRAME entry and disables all other windows when the FRAME window opens.

V8 GETVARF function
assigns the formatted value of a SAS table column to an SCL character variable.

IMPORT statement
defines a search path for references to CLASS entries in an SCL program so that you can refer to a class by its one- or two-level name instead of having to specify the four-level name each time.

INITROW function
initializes the table data vector (TDV) to missing values. This prevents bad data from being written to a row of a SAS table when values are not explicitly assigned to columns and you use the APPEND function with the NOINIT option.

INTERFACE statement
specifies the design of an interface.

V8 ITEM statement
specifies a class on the server that can be accessed by applications on the client.

V8 MAKEARRAY function
creates an array of the given size and initializes all of the elements in the array either to missing (for numeric elements) or blank (for character elements).

MESSAGEBOX function
displays a host message box that can contain an icon, buttons, and message text. MESSAGEBOX returns the user's selection.

NAMEDIVIDE function
returns the number of parts in a two- to four-level compound name as well as the values of each part.

NAMEMERGE function
merges two to four name parts into a compound name.

NEO operator
provides a faster and more direct way to create an object. This operator combines the actions of loading a class (using LOADCLASS) and initializing the object with the _new method, which invokes the object's _init method.

NEW operator
creates an object and runs the associated class constructor.

OPENENTRYDIALOG function
displays a list of entries in SAS catalogs and returns the user's selection.

OPENSASFILEDIALOG function
displays a list of SAS files and returns the user's selection.

V8 PACKAGE statement
defines a group of classes whose metadata must be recognized by objects that are defined on the client.

V8 REDIM function
resizes a dynamic array.

SAVEENTRYDIALOG function
enables you to implement a **Save As** choice by displaying a dialog box that lists entries in SAS catalogs and returns the selected catalog name.

SAVESASFILEDIALOG function
enables you to implement a **Save As** choice by displaying a dialog box that lists SAS files and returning the selected filename.

SELECTICON function
displays a selection list of icons and returns the number that identifies the selected icon.

V8 SUBMITCLEAR function
aborts a pending submit transaction.

V8 THROW statement
raises an exception.

UNIQUENUM function
returns a number that is unique for each call to the function during a SAS session.

USECLASS statement
binds methods that are implemented within it to the specified class definition.

Enhanced SCL Elements

For details about these new language elements, see Chapter 16, "SAS Component Language Dictionary," on page 249.

ATTRN function
now provides the NLOBSF option, which returns the number of logical rows (those not marked for deletion) in a SAS table when an active WHERE clause is applied. Also, the new ICONST option provides information about integrity constraints for generation data sets, and the GENMAX and GENNEXT options provide information about generation numbers.

CONTROL statement
now provides a new option, HALTONDOTATTRIBUTE, which enables you to specify whether your application halts execution if SCL detects an error in the dot notation.

DELETE function
now provides the FILE type for deleting external files or directories.

ENTRY statement
now enables you to specify the parameter modifiers INPUT, UPDATE, and OUTPUT. In Version 6, all parameters were treated as if they were UPDATE parameters.

FILEDIALOG function
now provides a DESCRIPTION option for each filter that you specify so that you can provide a description of each filter.

FONTSEL function
can now open the portable font selector for hardware fonts, using the new 'H' font-selector option.

GETLATTR function
can now return the value of the HONORCASE|IGNORECASE attribute, which specifies whether character values are stored in the case in which they are entered or whether they are stored in uppercase.

METHOD statement
now enables you to specify an access scope for methods. You can specify PUBLIC, PRIVATE, or PROTECTED. All Version 6 methods were treated as PUBLIC

methods. Also, you can specify the parameter modifiers INPUT, UPDATE, or OUTPUT in the argument list. All Version 6 parameters were treated as UPDATE parameters.

NAMEDITEM function
now enables you to specify whether a search is case sensitive.

RENAME function
now provides the FILE option for renaming external files or directories.

SETLATTR function
now provides the HONORCASE|IGNORECASE attribute, which specifies whether character values are stored in the case in which they are entered or whether they are stored in uppercase.

In addition, most data set functions that take a data set name as a parameter have been enhanced to support generation data sets. *Generation data sets* enable you to keep multiple copies of a SAS data set. Refer to *SAS Language Reference: Concepts* for more information on generation data sets.

Compatibility Issues

ACCESS routine
The functionality of ACCESS is available through the CALL BUILD routine. Both ACCESS and BUILD now open the Explorer window. Old programs that use ACCESS will still work, although the TYPE and MODE parameters are not supported from the Explorer window. Using BUILD for new programs is recommended because it provides additional functionality.

CALC routine
The CALL CALC routine is not supported in Version 7.

CARDS statement
The DATA step statement DATALINES replaces the CARDS statement.

CATALOG function
The functionality of CATALOG is now available through the CALL BUILD routine. Like CATALOG, BUILD opens the Explorer window when a catalog is specified as the first parameter. Old programs with the CATALOG function will still run, although the SHOWTYPE and PMENU parameters are not supported from the Explorer window. Using BUILD for new programs is recommended because it provides additional functionality.

CATLIST, DIRLIST, and FILELIST, and IVARLIST functions
Version 7 tables may contain mixed-case names. If any existing application relies on one of these functions to return an uppercased name, you may need to modify the application. See Chapter 16, “SAS Component Language Dictionary,” on page 249 for more information.

CATLIST, DIRLIST, FILELIST, and LIBLIST windows
These windows have been replaced with host selector windows in which the AUTOCLOSE option will now be ignored.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS® Component Language: Reference, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS® Component Language: Reference, Version 8

ISBN 1–58025–495–0

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999