Changes and Enhancements

Introduction

This section describes the features of SAS Component Language that have been implemented or enhanced since Release 6.12.

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 are reference types that store identifiers for SCL lists and components, respectively. The object types can be specific (CLASS or INTERFACE) or generic (OBJECT).

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

```plaintext
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:

```plaintext
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 debugger includes 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
Changes and Enhancements

IMGCTRL, IMGOP and PICFILL Functions,” on page 741. 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.

New Language Elements

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.

CREATESCL function

writes a class definition from a CLASS or INTERFACE catalog entry to a CLASS or INTERFACE statement in an SCL entry.

COMPARELIST function

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

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.)

DIALOG function

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

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.

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.

_NEEO_ 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.

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.

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**

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 15, “SAS Component Language Dictionary,” on page 243 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.