Glossary

Program Control

- An imaginary pointer to the run time DATA step statement being currently executed.
- The active statement highlighted by the DATA step debugger.
- When a statement is currently executed, it is said that it *has program control*.

SAS Variable

- A data container in memory whose value can be used and changed by a program.
- A column (field) in a SAS data file, i.e. a SAS data set or data view.
- Any SAS variable has a number of *attributes*, such as the name, data type, length, etc.

Program Data Vector (PDV)

- A logical image of the SAS variables in the DATA step memory, both created by the program and automatic, such as _N_, _IORC_, and _ERROR_.
- The PDV is commonly represented graphically as a sequence of containers named after the corresponding variables, holding their current values, and listed in the compilation order.

SAS Expression

- An elemental program component with a specific action that resolves to a value. The elemental components include literal constants, variables, functions, formats, and informats.
- A *group* of elemental components combined via SAS operators that resolves to a value.

Data Type

- A data attribute indicating the kind of operation that can be performed on the data.
- An attribute of an elemental program component indicating the kind of data it can operate on. Such components include literal constants, variables, functions, operators, formats, and informats.
- An attribute of a SAS expression indicating the data type of the value to which it resolves.

Scalar

- A singular unit of actual data, such as a numeric or character value.
- A data type designed to process a specific kind of scalar data. Base SAS has two scalar data types: Numeric and character.
- A SAS variable of a scalar data type, i.e. a numeric or character variable.
- Any other numeric or character SAS expression.

Non-Scalar

- A data structure with multiple data points, such as a list, an array, a table, an object, etc.
- A data type designed to process a specific kind of non-scalar data.
- Specific to this book, are two non-scalar data types: Type hash and type hash iterator (hiter).
- A PDV variable of non-scalar data type, whose values identify an occurrence of non-scalar data.

Big *O* Notation

- A convention used to indicate how a computer operation's run time *scales* as the data grows.
- For example, if for a data table with *N* rows with *N* distinct keys it takes time *T* to read 1 row, the mean time *ST* needed to find a key via the *linear search* is proportional to *T***N*. Using the big *O* notation, this fact is denoted as *O*(*N*) or by saying that the search "runs in *O*(*N*) time".
- If the table is *ordered* by the key and the *binary search* is used, *ST* is proportional to *T**log2(*N*). Respectively, in the big *O* notation it is denoted as *O*(log2(*N*)).
- The table can be organized to make its search time *ST* proportional to *T**1=*T* , i.e. the same regardless of *N*. In this case, it is said that the search runs in *O*(*1*) or *constant time*.

Hash Table

- A memory-resident table, i.e. a data structure with rows and columns.
- The rows are called *hash items,* and the columns are called *hash variables*.
- One or more columns are collectively designated as a *key*.
- The table is specifically organized to be searched by the key via the *hash algorithm*.

Hash Algorithm

- An algorithm designed to search a hash table by its key in *O*(*1*) , i.e. *constant*, time.
- In the SAS hash object, the algorithm is based on coupling a *hash function* with *AVL trees*.
- It is executed by first locating a key-value in one of the AVL trees and then searching it.

Hash Function

- A function whose response *H* splits *N* distinct arbitrary keys into *H* nearly equal groups (*buckets*) with approximately *N*/*H* key values in each group.
- The SAS hash object hash function splits distinct input key values evenly between *H* AVL trees.
- *H* is defined by the values of the HASHEXP:*P* argument tag as *H*=2***P*. By default, *P*=8.

AVL Tree

- A self-balancing binary search tree (named by its inventors ***A***delson-***V***elsky and ***L***andis).
- Self-balancing is a specific mechanism of inserting input keys into the tree. It ensures that the search time of a tree with *N* distinct keys stays *O*(log2(*N*)) regardless of the input key values.
- The AVL trees implemented in SAS are fast and efficient. They underlie not only the hash object, but also SAS procedures like MEANS, FREQ, TABULATE, etc.

Hash Variable

- Hash table column, whose names adheres to the SAS variable naming conventions.
- Hash variables are structurally analogous to SAS data set variables.
- They are divided in two *functionally distinct* groups: *Key portion* and *data portion*.

PDV Host Variables

- PDV variables whose names match the names of the defined hash variables.
- For each hash variable (defined at run time), there must have existed a host variable with the same name (defined at compile time).
- Hash variables inherit their attributes (such as length) from their respective host variables.
- The host variables are the PDV variables updated when data is retrieved from the hash table.
- The current PDV host variable values are the values assumed by implicit method calls adding, removing, or updating hash table items.

Parameter Type Matching

- Any technique of placing the host variables in the PDV at compile time.

- Parameter type matching ensures that by the time the hash variables are being defined at run time, the host variables with the requisite names already exist in the PDV.

Hash Table Item

- A hash table row populated with values of the key portion and data portion variables.
- A hash item in a hash table is structurally analogous to an observation in a SAS data set.

Hash Table Entry

- A list of all hash variables, both in the key and data portions, and their attributes (such as length).
- The order of the variables in the hash table entry follows the order they are defined by the DEFINEKEY and DEFINEDATA methods.
- The hash table entry is analogous to the descriptor (aka the header) of a SAS data set.

Key Portion

- The hash variables defined in the program to serve collectively as the *hash table key*.
- The values of key portion variables within an existing item cannot be updated. Conversely, their values cannot be used to update PDV variables.
- The data type of the key portion variables can be only *scalar*, i.e. either numeric or character.
- The key portion is *mandatory*, i.e. it must be defined with at least one hash variable.

Hash Table Key

- The key portion variables whose combined value identifies one or more hash items.
- The hash variables comprising the key are called *key variables* or *key components*.
- If the key consists of one key variable, it is *simple*; otherwise, it is *composite*.
- The key variables are defined using the DEFINEKEY method.

Key-Value

- The combination of the values of all key variables in the order they are defined.
- A combination of values of PDV variables (or other SAS expressions) the table is searched for to identify one or more hash items with the matching values of the key variables.

Same-Key Item Group

- A group of hash items with the same key-value.
- Items with the same key-value can exist in a hash table if the MULTIDATA:"Y" argument tag is specified at the time when its hash object instance is created.

Data Portion

- The hash variables whose values can be both updated by the program and used to update the values of PDV variables.
- The data portion hash variables are called *hash data variables*.
- The hash data variables can be both *scalar* (i.e. numeric or character) and *non-scalar*. The values of non-scalar data variables can be used to identify non-scalar data, such as objects.
- The data portion, like the key portion, is *mandatory*. It must contain at least one hash variable.
- The hash data variables are defined using the DEFINEDATA method.

Data-Value

- The value of an *individual* data portion hash variable.
- The value of a PDV variable (or other SAS expression) used to update a data portion variable.

Hash Object

- A memory-resident data structure associated with an *object reference variable*.
- It comprises a hash table and a set of *hash object tools* invoked in the DATA step to perform *hash operations* on the table and its data.

Hash Object Name

- A valid SAS name given to the hash object in the DECLARE statement.

Hash Object Reference Variable

- A non-scalar DATA step PDV variable of type hash defined in the DECLARE statement at compile time with the same name as the hash object name.

Hash Object Tools

- Functions or routines invoked via the DATA Step Component Interface (DSCI) to perform hash object operations. They include:
  - DECLARE statement
  - _NEW_ operator
  - Hash attributes
  - Hash methods
  - Hash argument tags
  - Hash arguments

Hash Object Instance

- A run time physical occurrence of a hash object uniquely identified by a value of type hash.
- The actual container of a hash table that hash tools can manipulate.

Active Hash Object Instance

- The hash object instance whose identifier value (also referred to as a hash instance pointer) is currently stored in the hash object reference PDV variable.The instance against which any hash tool called by the hash object name currently operates.

Initialized Hash Object Instance

- An operable hash object instance whose hash table variables are defined and validated.
- Only the hash table data related to an initialized instance can be manipulated.

DECLARE Statement

- A compile time statement to declare a hash object and define its reference PDV variable.

NEW Operator

- A run time tool to create a hash object instance and generate its unique identifier value.

Compound DECLARE Statement

- A statement combining the actions of the DECLARE statement at compile time and the actions of the _NEW_ operator at run time.

Hash Attributes

- Hash object tools that return descriptive information about the *active* hash object instance:
    - NUM_ITEMS. Returns the current number of hash items in a hash table.
    - ITEM_SIZE. Returns the actual length of hash table entry in bytes.

Hash Methods

- Hash object functions called to manipulate the *active* hash object instance and its hash table data.
- They belong to two groups:
    - Table-level: Acting upon the object instance and/or its hash table as a whole.
    - Item-level: Acting upon individual hash table items.

Hash Argument Tags

- Keyword parameters through which most methods accept their arguments.

Hash Arguments

- Expressions passed as values to argument tags (i.e. most methods).
- Expressions passed as values to the methods and/or statements with no argument tags (e.g., the DEFINEKEY and DEFINEDATA methods and the statements creating a hash iterator instance).

Hash Iterator Object

- A data structure designed to process hash table items sequentially.

Hash Iterator Name

- A valid SAS name given to the hash iterator object in the DECLARE statement.

Hash Iterator Reference Variable

- A non-scalar DATA step PDV variable of type hash iterator defined in the DECLARE statement at compile time with the same name as the hash iterator object name.

Hash Iterator Instance

- A run time physical occurence of the hash iterator object uniquely identified by a value of type hash iterator.
- It is uniquely linked to and always works with a specific hash object instance.

Hash Iterator Active Instance

- The hash iterator instance whose identifier value is currently stored in its hash iterator reference PDV variable.
- The iterator instance the program uses whenever it is called by its iterator reference name.

Hash Iterator Pointer

- An imaginary cursor pointing to the hash item the iterator is currently processing.
- If the iterator is not latched onto any item, the iterator pointer is thought to dwell outside the hash table.

Hash Iterator Locking

- A condition when an item cannot be removed from a hash table because it is locked by a hash iterator. It occurs when:
    - o The iterator pointer dwells on any item and an attempt is made to remove all items from the table.

- The iterator pointer dwells on one item in a same-key item group, and an attempt is made to remove any item from that same-key group.

Explicit Method Call

- A method call in which its argument tags are valued with expressions of the required data type.
- The method accepts the values to which the expessions resolve.

Implicit Method Call

- A method call in which its argument tags are left out.
- The method accepts the current values from the corresponding PDV host variables.

Assigned Method Call

- A method call whose return code is assigned to either:
  - A separate receiving variable in an assignment statement.
  - An internal buffer when the method call is used as part of a statement or an expression.

Unassigned Method Call

- A standalone method call whose return code is not assigned to either a separate variable or the internal buffer.

LINK Statement

- A labeled block of SAS DATA step statements called by it label name and coded at the bottom of the step after the RETURN or STOP statement.
- Used to avoid repetitive coding when the same block of statements needs to be executed more than once.
- An alternative to coding a similarly functional macro.

DoW Loop

- A programming construct representing a DO loop wrapped around a file-reading statement, such as SET or MERGE, and terminated when a certain break-event condition is met.
- Typically, the break-event represents either having read the last record in a BY group or the last record in the entire file.
- The DoW loop takes over control of reading the file from the implied DATA step loop.
- As such, it has a number of significant advantages compared to relying exclusively on the implied loop in terms of simplifying programming logic, eliminating the need to retain and reinitialize aggregation variables, eschewing unnecessary conditional statements, and fostering better performance.

- It can be used, in a single execution of the DATA step, to read each BY group or the entire file repeatedly or read different files one at a time, allowing for pre/post-processing before/after each separate read.
- "DoW loop" is an informal term for this construct by the SAS community since about 2000.
- See, for example: https://analytics.ncsu.edu/sesug/2011/SS01.Dorfman.pdf