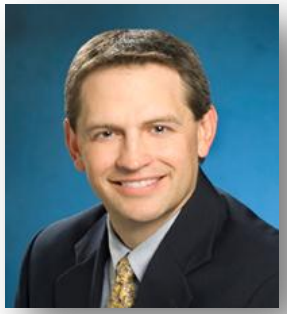
Getting Started with SAS® Hash Objects

Michele Burlew
2012 SAS Talks

Speakers



- Stacy Hobson
  - Director, Customer Loyalty and Retention
    SAS Institute

- Michele M. Burlew
  - SAS Author and Consultant

Understanding the Material in This Webinar

- Basic SAS language programming skills
- Familiarity with lookup process using SAS language statements, formats, and/or arrays
- Familiarity with combining data sets using DATA steps and/or procedures
- Little or no experience in writing DATA steps that include hash objects

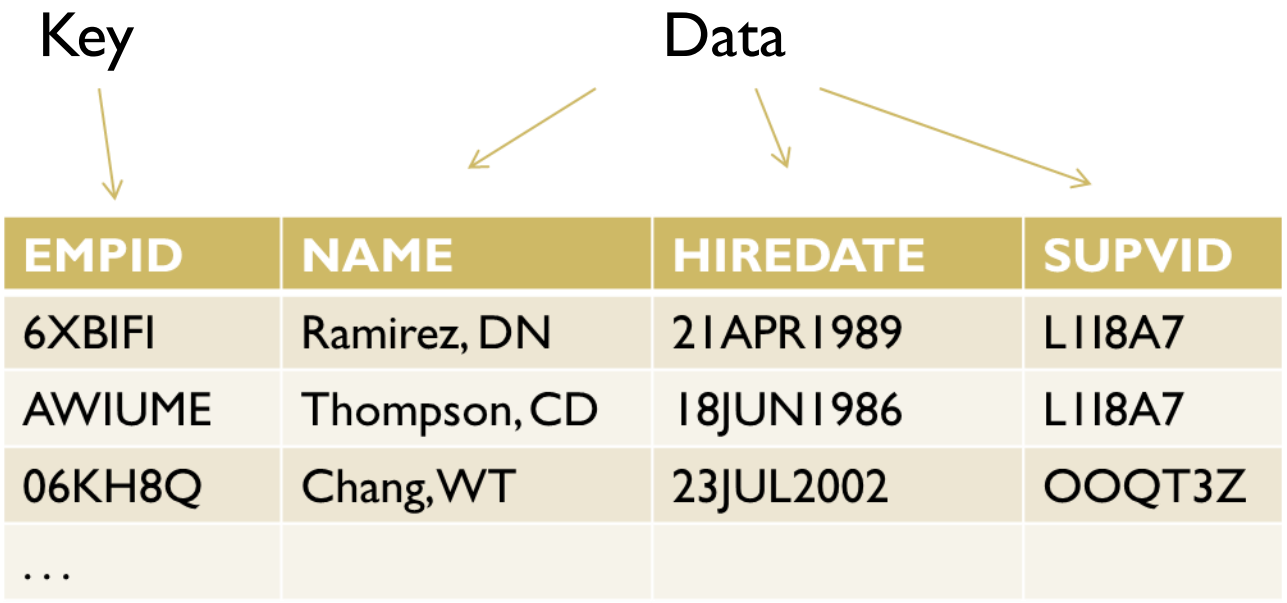
Outline

1. What is a SAS Hash Object?
2. How is a Hash Object Used?
3. Advantages and Disadvantages of SAS Hash Objects
4. Example: Lookup Table
5. Hash Object Terminology
6. Writing Hash Object Code
7. Example: Searching for a Key Value in Multiple Hash Objects
8. Example: Creating a Data Set from a Hash Object
9. Summary
10. Questions

What is a SAS Hash Object?

- Like an array that a DATA step accesses using keys
- Consists of key items and data items
- Key items can be any combination of numeric and character items
- Hash objects can return more than one data item and these can be any combination of numeric and character items
- SAS applies a hash function that maps the key values to positions in the hash table

What is a SAS Hash Object?



What is a SAS Hash Object?

- Resides in memory

What is a SAS Hash Object?

- Resides in memory
- Used only in the DATA step

What is a SAS Hash Object?

- Resides in memory
- Used only in the DATA step
- Exists only during execution of the DATA step that creates it

What is a SAS Hash Object?

- Resides in memory
- Used only in the DATA step
- Exists only during execution of the DATA step that creates it
- Filled from a data set or from data generated by the DATA step

What is a SAS Hash Object?

- Resides in memory
- Used only in the DATA step
- Exists only during execution of the DATA step that creates it
- Filled from a data set or from data generated by the DATA step
- Modifies its size as you add and remove items

How is a Hash Object Used?

- Lookup table
- Combine data
- Organize data

Advantages and Disadvantages of SAS Hash Objects

Advantages

- Fast retrieval of data from memory
- Conserve resources
- Can return more than one data item. Items can be a mix of character and numeric data items.

Disadvantages

- Size of hash object too large for available memory
- Some features limited compared to DATA step and PROC step capabilities

Outline

1. What is a SAS Hash Object?
2. How is a Hash Object Used?
3. Advantages and Disadvantages of SAS Hash Objects
4. Example: Lookup Table
5. Hash Object Terminology
6. Writing Hash Object Code
7. Example: Searching for a Key Value in Multiple Hash Objects
8. Example: Creating a Data Set from a Hash Object
9. Summary
10. Questions

Example: Lookup Table

Goal: Find data about the employees in EMPHOURS by looking up information in data set EMPLOYEES based on the values of EMPID

Data Set EMPHOURS

EMPID	HRSWORKED
6XBIFI	38.5
WA4D7N	22.0
VPA9EF	43.0
TZ6OUB	11.5
L6KKHS	29.0
8TN7WL	38.0

Example: Lookup Table

Data Set EMPLOYEES is the Lookup Table

EMPID	EMPLN	EMPFN	EMPMI	GENDER	STARTDATE	EMPPAYLEVEL
6XBIFI	Ramirez	Danielle	N	F	04/21/1989	AIb
AWIUME	Thompson	Catherine	D	F	06/18/1986	PIIIa
06KH8Q	Chang	William	T	M	07/23/2002	PIIa
WA4D7N	Garcia	Breanna	X	F	08/20/1982	AIb
VPA9EF	Baker	Cheyenne	C	F	02/24/1990	AIIa
. . .						

Example: Lookup Table

Non-Hash Object Solution

```
proc sort data=emphours;
  by empid;
run;
proc sort data=mylib.employees;
  by empid;
run;
data empnames;
  merge emphours(in=inhours)
        mylib.employees(keep=empid empln empfn empmi in=inall);
    by empid;
  length empname $ 60;
  keep empid emphours empname;
  if inhours;
  if inall then empname=catx(' ',empfn,empmi,empln);
  else empname='** Not Found';
run;
```

Example: Lookup Table

Hash Object Solution

```
data empnames;
  length empid $ 6 empln $ 30 empfn $ 25 empmi $ 1
         empname $ 60 ;
  if _n_=1 then do;
    declare hash e(dataset: 'mylib.employees');
    e.definekey('empid');
    e.definedata('empln','empfn','empmi');
    e.definedone();
    call missing(empln,empfn,empmi);
  end;
  set emphours;
  keep empid hrsworked empname;
  rc=e.find();
  if rc=0 then empname=catx(' ',empfn,empmi,empln);
  else empname='** Not Found';
run;
```

Example: Lookup Table

- Create Hash Object E

```
data empnames;
  length empid $ 6 empln $ 30 empfn $ 25 empmi $ 1
         empname $ 60 ;
  if _n_=1 then do;
    declare hash e(dataset: 'mylib.employees');
    e.definekey('empid');
    e.definedata('empln','empfn','empmi');
    e.definedone();
    call missing(empln,empfn,empmi);
  end;
  set emphours;
  keep empid hrsworked empname;
  rc=e.find();
  if rc=0 then empname=catx(' ',empfn,empmi,empln);
  else empname='** Not Found';
run;
```

Example: Lookup Table

- Load Lookup Data from EMPLOYEES into Hash Object E

```
data empnames;
  length empid $ 6 empln $ 30 empfn $ 25 empmi $ 1
         empname $ 60 ;
  if _n_=1 then do;
    declare hash e(dataset: 'mylib.employees');
    e.definekey('empid');
    e.definedata('empln','empfn','empmi');
    e.definedone();
    call missing(empln,empfn,empmi);
  end;
  set emphours;
  keep empid hrsworked empname;
  rc=e.find();
  if rc=0 then empname=catx(' ',empfn,empmi,empln);
  else empname='** Not Found';
run;
```

Example: Lookup Table

- Define the Key Item to Find Data in E

```
data empnames;
  length empid $ 6 empln $ 30 empfn $ 25 empmi $ 1
         empname $ 60 ;
  if _n_=1 then do;
    declare hash e(dataset: 'mylib.employees');
    e.definekey('empid');
    e.definedata('empln','empfn','empmi');
    e.definedone();
    call missing(empln,empfn,empmi);
  end;
  set emphours;
  keep empid hrsworked empname;
  rc=e.find();
  if rc=0 then empname=catx(' ',empfn,empmi,empln);
  else empname='** Not Found';
run;
```

Example: Lookup Table

- Define the Data Items to Extract from E

```
data empnames;
  length empid $ 6 empln $ 30 empfn $ 25 empmi $ 1
         empname $ 60 ;
  if _n_=1 then do;
    declare hash e(dataset: 'mylib.employees');
    e.definekey('empid');
    e.definedata('empln','empfn','empmi');
    e.definedone();
    call missing(empln,empfn,empmi);
  end;
  set emphours;
  keep empid hrsworked empname;
  rc=e.find();
  if rc=0 then empname=catx(' ',empfn,empmi,empln);
  else empname='** Not Found';
run;
```

Example: Lookup Table

- End the Definition of Hash Object E

```
data empnames;
  length empid $ 6 empln $ 30 empfn $ 25 empmi $ 1
         empname $ 60 ;
  if _n_=1 then do;
    declare hash e(dataset: 'mylib.employees');
    e.definekey('empid');
    e.definedata('empln','empfn','empmi');
    e.definedone();
    call missing(empln,empfn,empmi);
  end;
  set emphours;
  keep empid hrsworked empname;
  rc=e.find();
  if rc=0 then empname=catx(' ',empfn,empmi,empln);
  else empname='** Not Found';
run;
```

Example: Lookup Table

- Read Each Observation in EMPHOURS

```
data empnames;
  length empid $ 6 empln $ 30 empfn $ 25 empmi $ 1
         empname $ 60 ;
  if _n_=1 then do;
    declare hash e(dataset: 'mylib.employees');
    e.definekey('empid');
    e.definedata('empln','empfn','empmi');
    e.definedone();
    call missing(empln,empfn,empmi);
  end;
  set emphours;
  keep empid hrsworked empname;
  rc=e.find();
  if rc=0 then empname=catx(' ',empfn,empmi,empln);
  else empname='** Not Found';
run;
```

Example: Lookup Table

- Look in E for the Current Value of EMPID in EMPHOURS

```
data empnames;
  length empid $ 6 empln $ 30 empfn $ 25 empmi $ 1
         empname $ 60 ;
  if _n_=1 then do;
    declare hash e(dataset: 'mylib.employees');
    e.definekey('empid');
    e.definedata('empln','empfn','empmi');
    e.definedone();
    call missing(empln,empfn,empmi);
  end;
  set emphours;
  keep empid hrsworked empname;
  rc=e.find();
  if rc=0 then empname=catx(' ',empfn,empmi,empln);
  else empname='** Not Found';
run;
```

Example: Lookup Table

- Process Data Returned (or Not Returned) from E

```
data empnames;
  length empid $ 6 empln $ 30 empfn $ 25 empmi $ 1
         empname $ 60 ;
  if _n_=1 then do;
    declare hash e(dataset: 'mylib.employees');
    e.definekey('empid');
    e.definedata('empln','empfn','empmi');
    e.definedone();
    call missing(empln,empfn,empmi);
  end;
  set emphours;
  keep empid hrsworked empname;
  rc=e.find();
  if rc=0 then empname=catx(' ',empfn,empmi,empln);
  else empname='** Not Found';
run;
```

Example: Lookup Table

Data Set EMPNAMES

EMPID	EMPNAME	HRSWORKED
6XBIFI	Danielle N Ramirez	38.5
WA4D7N	Breanna X Garcia	22.0
VPA9EF	Cheyenne C Baker	43.0
TZ6OUB	Heather T White	11.5
L6KKHS	** Not Found	29.0
8TN7WL	Tyler J Miller	38.0

Outline

1. What is a SAS Hash Object?
2. How is a Hash Object Used?
3. Advantages and Disadvantages of SAS Hash Objects
4. Example: Lookup Table
5. Hash Object Terminology
6. Writing Hash Object Code
7. Example: Searching for a Key Value in Multiple Hash Objects
8. Example: Creating a Data Set from a Hash Object
9. Summary
10. Questions

Hash Object Terminology

- *Instantiation* is the creation of a component object such as a hash object.

Hash Object Terminology: Instantiation

```
data empnames;
  length empid $ 6 empln $ 30 empfn $ 25 empmi $ 1
         empname $ 60 ;
  if _n_=1 then do;
    declare hash e(dataset: 'mylib.employees');
    e.definekey('empid');
    e.definedata('empln','empfn','empmi');
    e.definedone();
    call missing(empln,empfn,empmi);
  end;
  set emphours;
  keep empid empname hrsworked;
  rc=e.find();
  if rc=0 then empname=catx(' ',empfn,empmi,empln);
  else empname='** Not Found';
run;
```

Hash Object Terminology

- *Instantiation* is the creation of a component object such as a hash object.
- An *attribute* is a property of the hash object.
- A *method* is an operation that an object can perform.

Hash Object Terminology: Methods

```
data empnames;
  length empid $ 6 empln $ 30 empfn $ 25 empmi $ 1
         empname $ 60 ;
  if _n_=1 then do;
    declare hash e(dataset: 'mylib.employees');
    e.definekey('empid');
    e.definedata('empln','empfn','empmi');
    e.definedone();
    call missing(empln,empfn,empmi);
  end;
  set emphours;
  keep empid empname hrsworked;
  rc=e.find();
  if rc=0 then empname=catx(' ',empfn,empmi,empln);
  else empname='** Not Found';
run;
```

Hash Object Terminology

- *Instantiation* is the creation of a component object such as a hash object.
- An *attribute* is a property of the hash object.
- A *method* is an operation that an object can perform.
- Methods and attributes are written in *dot notation*.

Hash Object Terminology: Dot Notation

```
data empnames;
  length empid $ 6 empln $ 30 empfn $ 25 empmi $ 1
         empname $ 60 ;
  if _n_=1 then do;
    declare hash e(dataset: 'mylib.employees');
    e.definekey('empid');
    e.definedata('empln','empfn','empmi');
    e.definedone();
    call missing(empln,empfn,empmi);
  end;
  set emphours;
  keep empid empname hrsworked;
  rc=e.find();
  if rc=0 then empname=catx(' ',empfn,empmi,empln);
  else empname='** Not Found';
run;
```

Hash Object Terminology

- *Instantiation* is the creation of a component object such as a hash object.
- An *attribute* is a property of the hash object.
- A *method* is an operation that an object can perform.
- Methods and attributes are written in *dot notation*.
- *Argument tags* specify options to the method.

Hash Object Terminology: Argument Tag

```
data empnames;
  length empid $ 6 empln $ 30 empfn $ 25 empmi $ 1
         empname $ 60 ;
  if _n_=1 then do;
    declare hash e(dataset: 'mylib.employees');
    e.definekey('empid');
    e.definedata('empln','empfn','empmi');
    e.definedone();
    call missing(empln,empfn,empmi);
  end;
  set emphours;
  keep empid empname hrsworked;
  rc=e.find();
  if rc=0 then empname=catx(' ',empfn,empmi,empln);
  else empname='** Not Found';
run;
```

Hash Object Terminology

- *Instantiation* is the creation of a component object such as a hash object.
- An *attribute* is a property of the hash object.
- A *method* is an operation that an object can perform.
- Methods and attributes are written in *dot notation*.
- *Argument tags* specify options to the method.
- A hash object always contains at least one *key item* and it optionally contains *data items*.

Hash Object Terminology: Key Items and Data Items

```
data empnames;
  length empid $ 6 empln $ 30 empfn $ 25 empmi $ 1
         empname $ 60 ;
  if _n_=1 then do;
    declare hash e(dataset: 'mylib.employees');
    e.definekey('empid');
    e.definedata('empln','empfn','empmi');
    e.definedone();
    call missing(empln,empfn,empmi);
  end;
  set emphours;
  keep empid empname hrsworked;
  rc=e.find();
  if rc=0 then empname=catx(' ',empfn,empmi,empln);
  else empname='** Not Found';
run;
```

Hash Object Terminology

- *Instantiation* is the creation of a component object such as a hash object.
- An *attribute* is a property of the hash object.
- A *method* is an operation that an object can perform.
- Methods and attributes are written in *dot notation*.
- *Argument tags* specify options to the method.
- A hash object always contains at least one *key item* and it optionally contains *data items*.
- SAS by default allows one set of data items per key value. Options exist that allow multiple sets of data items per key value.

Hash Object Terminology

- *Instantiation* is the creation of a component object such as a hash object.
- An *attribute* is a property of the hash object.
- A *method* is an operation that an object can perform.
- Methods and attributes are written in *dot notation*.
- *Argument tags* specify options to the method.
- A hash object always contains at least one *key item* and it optionally contains *data items*.
- SAS by default allows one set of data items per key value. Options exist to allow multiple sets of data items per key value.
- A *hash iterator object* works with hash objects. It allows you to traverse a hash object in either forward or backward key value order.

Outline

1. What is a SAS Hash Object?
2. How is a Hash Object Used?
3. Advantages and Disadvantages of SAS Hash Objects
4. Example: Lookup Table
5. Hash Object Terminology
6. Writing Hash Object Code
7. Example: Searching for a Key Value in Multiple Hash Objects
8. Example: Creating a Data Set from a Hash Object
9. Summary
10. Questions

Writing Hash Object Code

- Define the hash object – typically once
- Define key items and data items – typically once

Writing Hash Object Code: Define Hash Object, Key Items, and Data Items

```
data empnames;
  length empid $ 6 empln $ 30 empfn $ 25 empmi $ 1
         empname $ 60 ;

  if _n_=1 then do;
    declare hash e(dataset: 'mylib.employees');
    e.definekey('empid');
    e.definedata('empln','empfn','empmi');
    e.definedone();
    call missing(empln,empfn,empmi);
  end;

  set emphours;
  keep empid empname hrsworked;

  rc=e.find();
  if rc=0 then empname=catx(' ',empfn,empmi,empln);
  else empname='** Not Found';
run;
```

Writing Hash Object Code

- Define the hash object – typically once
- Define key items and data items – typically once
- Define and initialize variables that have the same names as the key items and data items – typically once

Writing Hash Object Code: Define and Initialize Variables

```
data empnames;
  length empid $ 6 empln $ 30 empfn $ 25 empmi $ 1
         empname $ 60 ;

  if _n_=1 then do;
    declare hash e(dataset: 'mylib.employees');
    e.definekey('empid');
    e.definedata('empln','empfn','empmi');
    e.definedone();
    call missing(empln,empfn,empmi);
  end;

  set emphours;
  drop rc;

  rc=e.find();
  if rc=0 then empname=catx(' ',empfn,empmi,empln);
  else empname='** Not Found';
run;
```

Writing Hash Object Code

- Define the hash object – typically once
- Define key items and data items – typically once
- Initialize variables that have the same names as the key items and data items – typically once
- Invoke methods that access the hash object – as many times as needed as the DATA step executes

Writing Hash Object Code: Invoke Methods

```
data empnames;
  length empid $ 6 empln $ 30 empfn $ 25 empmi $ 1
         empname $ 60 ;

  if _n_=1 then do;
    declare hash e(dataset: 'mylib.employees');
    e.definekey('empid');
    e.definedata('empln','empfn','empmi');
    e.definedone();
    call missing(empln,empfn,empmi);
  end;

  set emphours;
  drop rc;

  rc=e.find();
  if rc=0 then empname=catx(' ',empfn,empmi,empln);
  else empname='** Not Found';
run;
```

Writing Hash Object Code: Methods and Attributes

Action	Methods
Definition	DECLARE statement, DEFINEDATA, DEFINEKEY
Find or Verify Presence of an Entry	CHECK, FIND, FIRST, LAST, NEXT,PREV
Add, Modify, or Remove Items in a Hash Object	ADD, REF, REMOVE, REPLACE
Empty or Remove Entire Hash Object	CLEAR, DELETE
Create a SAS Data Set from a Hash Object	OUTPUT
Compare Hash Objects	EQUALS
Manage Hash Objects	ITEM_SIZE (attribute), NUM_ITEMS (attribute)

Additional methods exist that work with the special situation of multiple sets of data items per key value.

Writing Hash Object Code: DATA Step Variables and Hash Object Data Items

- Define hash object key and data items as DATA step variables
- Initialize these DATA step variables
- Data item values retrieved from hash object are deposited in same-named DATA step variables in PDV
- Values loaded into hash object data items may come from same-named DATA step variables in PDV, from other DATA step variables, or from literal values

Writing Hash Object Code: DATA Step Variables and Hash Object Data Items

Goal: Show the contents of the PDV when looking for key values in a hash object. Find conference information for rooms specified in ROOMSCHEDULE2.

Data Set ROOMSCHEDULE2

mtgdate	mtgtime	roomid
08/13/2013	8:30	C0P01
08/13/2013	11:30	B1004
08/13/2013	1:15	A0122
08/13/2013	2:00	B1004

Data Set CONFROOMS

roomid	roomno	floor	bldg	cap
A0210	10	2	Anderson	50
A0120	20	1	Anderson	75
B0B05	5	B	Baylor	100
B1004	4	10	Baylor	15
B0212	12	2	Baylor	30
C0P01	1	P	Cummings	150

Writing Hash Object Code: DATA Step Variables and Hash Object Data Items

- Includes ATTRIB statement

```
data pdvck3;
 attrib roomno length=8
        floor  length=$2
        bldg   length=$20
        cap    length=8;
  if _n_=1 then do;
    declare hash cr(dataset: 'confrooms');
    cr.definekey('roomid');
    cr.definedata('roomno','floor','bldg','cap');
    cr.definedone();
    call missing(roomno,floor,bldg,cap);
  end;
  set roomschedule2;

  rc=cr.find();
  put _all_;
run;
```

Writing Hash Object Code: DATA Step Variables and Hash Object Data Items

```
NOTE: There were 6 observations read from the data set
WORK.CONFROOMS.
roomno=1 floor=P bldg=Cummings cap=150 meetingdate=08/13/2013
meetingtime=8:30 roomid=C0P01 rc=0 _ERROR_=0 _N_=1

roomno=4 floor=10 bldg=Baylor cap=15 meetingdate=08/13/2013
meetingtime=11:30 roomid=B1004 rc=0 _ERROR_=0 _N_=2

roomno =. floor=  bldg=  cap=. meetingdate=08/13/2013
meetingtime=1:15 roomid=A0122 rc=160038 _ERROR_=0 _N_=3

roomno=4 floor=10 bldg=Baylor cap=15 meetingdate=08/13/2013
meetingtime=2:00 roomid=B1004 rc=0 _ERROR_=0 _N_=4

NOTE: There were 4 observations read from the data set
WORK.ROOMSCHEDULE2.
NOTE: The data set WORK.PDVCK3 has 4 observations and 8
variables.
```

Writing Hash Object Code: DATA Step Variables and Hash Object Data Items

```
NOTE: There were 6 observations read from the data set
WORK.CONFROOMS.
roomno=1 floor=P bldg=Cummings cap=150 meetingdate=08/13/2013
meetingtime=8:30 roomid=C0P01 rc=0 _ERROR_=0 _N_=1

roomno=4 floor=10 bldg=Baylor cap=15 meetingdate=08/13/2013
meetingtime=11:30 roomid=B1004 rc=0 _ERROR_=0 _N_=2

roomno =. floor=  bldg=  cap=. meetingdate=08/13/2013
meetingtime=1:15 roomid=A0122 rc=160038 _ERROR_=0 _N_=3

roomno=4 floor=10 bldg=Baylor cap=15 meetingdate=08/13/2013
meetingtime=2:00 roomid=B1004 rc=0 _ERROR_=0 _N_=4

NOTE: There were 4 observations read from the data set
WORK.ROOMSCHEDULE2.
NOTE: The data set WORK.PDVCK3 has 4 observations and 8
variables.
```

Writing Hash Object Code: DATA Step Variables and Hash Object Data Items

- No ATTRIB or LENGTH statement

```
data pdvck1;
  if _n_=1 then do;
    declare hash cr(dataset: 'confrooms');
    cr.definekey('roomid');
    cr.definedata('roomno','floor','bldg','cap');
    cr.definedone();

    call missing(roomno,floor,bldg,cap);
  end;
  set roomschedule2;
  rc=cr.find();

  put _all_;
run;
```

Writing Hash Object Code: DATA Step Variables and Hash Object Data Items

```
319  data pdvck1;
320    if _n_=1 then do;
321      declare hash cr(dataset: 'confrooms');
322      cr.definekey('roomid');
323      cr.definedata('roomno','floor','bldg','cap');
324      cr.definedone();
325      call missing(roomno,floor,bldg,cap);
326    end;
327    set roomschedule2;
328    rc=cr.find();
329    put _all_;
330  run;
```

ERROR: Type mismatch for data variable floor at line 324 column 5.
ERROR: Hash data set load failed at line 324 column 5.
ERROR: DATA STEP Component Object failure. Aborted during the
  EXECUTION phase.
NOTE: The SAS System stopped processing this step because of
  errors.

Outline

1. What is a SAS Hash Object?
2. How is a Hash Object Used?
3. Advantages and Disadvantages of SAS Hash Objects
4. Example: Lookup Table
5. Hash Object Terminology
6. Writing Hash Object Code
7. **Example: Searching for a Key Value in Multiple Hash Objects**
8. Example: Creating a Data Set from a Hash Object
9. Summary
10. Questions

Example: Searching for a Key Value in Multiple Hash Objects

Goal: Find whether the doctors in DOCTORLIST work at two or more of three clinic locations or at none of the locations.

Data Set DOCTORLIST

empid	empln	empfn	empmi	
2VIHPJ	Thomas	Noah	S	
3D401A	Long	Alicia	E	3
BKK94F	Robinson	Michael	M	0
HPM27C	Cooper	Brian	F	
LDT47L	Torres	Chelsea	Q	
MN2ZY6	Anderson	Chase	P	0
NBK588	Scott	Kyle	S	2
O0OBRU	Cooper	Bradley	D	
WE3HJH	Peterson	Jamie	Z	

Example: Searching for a Key Value in Multiple Hash Objects

Clinic Locations: Southside, Maplewood, and Midtown

Southside EMPIDs
3D401A
NBJ588
WE3HJH

Maplewood EMPIDs
3D401A
LDT47L
NBK588
O0OBRU

Midtown EMPIDs
2VIHPJ
3D401A
HPM27C
NBK588
NTWJLF
S7JHOM

Example: Searching for a Key Value in Multiple Hash Objects

- Create two output data sets:

1. MULTWORKLOCS: doctors who work at two or more of the three clinics
2. NOWORKLOC: doctors who do not work at any of the three clinics

```
data multworklocs noworkloc;
```

Example: Searching for a Key Value in Multiple Hash Objects

- Define a hash object for each of the three location lookup data sets.

```
if _n_=1 then do;
  declare hash ss(dataset: "doctors_southside");
  ss.definekey('empid');
  ss.definedone();

  declare hash mw(dataset: "doctors_maplewood");
  mw.definekey('empid');
  mw.definedone();

  declare hash mt(dataset: "doctors_midtown");
  mt.definekey('empid');
  mt.definedone();
end;
```

Example: Searching for a Key Value in Multiple Hash Objects

- Read each observation in DOCTORLIST, look for its EMPID value in each hash object, and sum up the number of locations where the doctors work.

```
set doctorlist;

ssrc=ss.check();
mwrc=mw.check();
mtrc=mt.check();

nlocs=(ssrc=0) + (mwrc=0) + (mtrc=0);
```

Example: Searching for a Key Value in Multiple Hash Objects

- Process the information obtained from the lookups.
- Output selected observations to MULTWORKLOCS and NOWORKLOC.

```
 attrib location_list length=$50
                          label='Employee Work Locations';

  if nlocs ge 2 then do;
    if ssrc=0 then location_list="Southside";
    if mwrc=0 then location_list=
             catx(',',location_list,"Maplewood");
    if mtrc=0 then location_list=
             catx(',',location_list,"Midtown");
    output multworklocs;
  end;
  else if nlocs=0 then do;
    location_list=
        "**Not at Southside, Maplewood, or Midtown";
    output noworkloc;
  end;
run;
```

Example: Searching for a Key Value in Multiple Hash Objects

Data Set MULTWORKLOCS

empid	empln	empfn	empmi	location_list
3D401A	Long	Alicia	E	Southside,Maplewood,Midtown
NBK588	Scott	Kyle	S	Maplewood,Midtown

Data Set NOWORKLOC

empid	empln	empfn	empmi	location_list
BKK94F	Robinson	Michael	M	**Not at Southside, Maplewood, or Midtown
MN2ZY6	Anderson	Chase	P	**Not at Southside, Maplewood, or Midtown

Outline

1. What is a SAS Hash Object?
2. How is a Hash Object Used?
3. Advantages and Disadvantages of SAS Hash Objects
4. Example: Lookup Table
5. Hash Object Terminology
6. Writing Hash Object Code
7. Example: Searching for a Key Value in Multiple Hash Objects
8. Example: Creating a Data Set from a Hash Object
9. Summary
10. Questions

Example: Creating a Data Set from a Hash Object

Goal: Generate 10 random numbers between 1 and 100. Create a data set where the observations are in order by the values of the random numbers. Save in another variable the sequence order in which the random number was generated.

Example: Creating a Data Set from a Hash Object

Non-Hash Object Solution

```
data random10;
  attrib sequence length=8 label='Selection Order'
         random_number length=8 label='Random Number';

  call streaminit(123456);

  do sequence=1 to 10;
    random_number=ceil(100*rand('uniform'));
    output;
  end;
run;
proc sort data=random10;
  by random_number;
run;
```

Example: Creating a Data Set from a Hash Object

Hash Object Solution

```
data _null_;
  attrib sequence length=8 label='Selection Order'
         random_number length=8 label='Random Number';

  declare hash r(ordered: 'yes');
  r.definekey('random_number');
  r.definedata('sequence','random_number');
  r.definedone();

  call missing(sequence,random_number);
  call streaminit(123456);

  do sequence=1 to 10;
    random_number=ceil(100*rand('uniform'));
    rc=r.add();
  end;

  rc=r.output(dataset: 'random10');
run;
```

Example: Creating a Data Set from a Hash Object

- No data set name on the DATA statement
- Data set RANDOM10 created by OUTPUT method

```
data _null_;
  attrib sequence length=8 label='Selection Order'
         random_number length=8 label='Random Number';

  declare hash r(ordered: 'yes');
  r.definekey('random_number');
  r.definedata('sequence','random_number');
  r.definedone();

  call missing(sequence,random_number);
  call streaminit(123456);

  do sequence=1 to 10;
    random_number=ceil(100*rand('uniform'));
    rc=r.add();
  end;

  rc=r.output(dataset: 'random10');
run;
```

Example: Creating a Data Set from a Hash Object

- No data set loaded into R.
- Entries in R retrieved by values of RANDOM_NUMBER.

```
data _null_;
  attrib sequence length=8 label='Selection Order'
         random_number length=8 label='Random Number';

  declare hash r(ordered: 'yes');
  r.definekey('random_number');
  r.definedata('sequence','random_number');
  r.definedone();

  call missing(sequence,random_number);
  call streaminit(123456);

  do sequence=1 to 10;
    random_number=ceil(100*rand('uniform'));
    rc=r.add();
  end;

  rc=r.output(dataset: 'random10');
run;
```

Example: Creating a Data Set from a Hash Object

- Entries added to hash object R

```
data _null_;
  attrib sequence length=8 label='Selection Order'
         random_number length=8 label='Random Number';

  declare hash r(ordered: 'yes');
  r.definekey('random_number');
  r.definedata('sequence','random_number');
  r.definedone();

  call missing(sequence,random_number);
  call streaminit(123456);

  do sequence=1 to 10;
    random_number=ceil(100*rand('uniform'));
    rc=r.add();
  end;

  rc=r.output(dataset: 'random10');
run;
```

Example: Creating a Data Set from a Hash Object

- SAS outputs contents of R *once* after R is completely filled.

```
data _null_;
  attrib sequence length=8 label='Selection Order'
         random_number length=8 label='Random Number';

  declare hash r(ordered: 'yes');
  r.definekey('random_number');
  r.definedata('sequence','random_number');
  r.definedone();

  call missing(sequence,random_number);
  call streaminit(123456);

  do sequence=1 to 10;
    random_number=ceil(100*rand('uniform'));
    rc=r.add();
  end;

  rc=r.output(dataset: 'random10');
run;
```

Example: Creating a Data Set from a Hash Object

- SAS outputs the items named in the DEFINEDATA method.

```
data _null_;
  attrib sequence length=8 label='Selection Order'
         random_number length=8 label='Random Number';

  declare hash r(ordered: 'yes');
  r.definekey('random_number');
  r.definedata('sequence','random_number');
  r.definedone();

  call missing(sequence,random_number);
  call streaminit(123456);

  do sequence=1 to 10;
    random_number=ceil(100*rand('uniform'));
    rc=r.add();
  end;

  rc=r.output(dataset: 'random10');
run;
```

Example: Creating a Data Set from a Hash Object

- SAS outputs the items named in the DEFINEDATA method.

```
data _null_;
  attrib sequence length=8 label='Selection Order'
         random_number length=8 label='Random Number';

  declare hash r(ordered: 'yes');
  r.definekey('random_number');
  r.definedata('sequence','random_number');
  r.definedone();

  call missing(sequence,random_number);
  call streaminit(123456);

  do sequence=1 to 10;
    random_number=ceil(100*rand('uniform'));
    rc=r.add();
  end;

  rc=r.output(dataset: 'random10');
run;
```

Example: Creating a Data Set from a Hash Object

Data Set RANDOM10

sequence	random_number
7	2
3	31
10	33
9	45
2	66
5	70
4	73
6	77
8	87
1	90

Example: Creating a Data Set from a Hash Object

Considerations and Modifications

- Omission of ORDERED: 'YES' Argument Tag

```
data _null_;
  attrib sequence length=8 label='Selection Order'
         random_number length=8 label='Random Number';

  declare hash r();
  r.definekey('random_number');
  r.definedata('sequence','random_number');
  r.definedone();

  call missing(sequence,random_number);
  call streaminit(123456);

  do sequence=1 to 10;
    random_number=ceil(100*rand('uniform'));
    rc=r.add();
  end;

  rc=r.output(dataset: 'random10');
run;
```

Example: Creating a Data Set from a Hash Object

Data Set RANDOM10 (no ORDERED: 'YES' argument tag)

sequence	random_number
8	87
7	2
4	73
3	31
1	90
2	66
6	77
9	45
10	33
5	70

Example: Creating a Data Set from a Hash Object

Considerations and Modifications

- Duplicate random numbers could be generated

```
data _null_;
  attrib sequence length=8 label='Selection Order'
         random_number length=8 label='Random Number';

  declare hash r(ordered: 'yes');
  r.definekey('random_number');
  r.definedata('sequence','random_number');
  r.definedone();

  call missing(sequence,random_number);
  call streaminit(12345);

  do sequence=1 to 10;
    random_number=ceil(100*rand('uniform'));
    rc=r.add();
  end;

  rc=r.output(dataset: 'random10');
run;
```

Example: Creating a Data Set from a Hash Object

Data Set RANDOM10 with duplicate random numbers generated

sequence	random_number
6	29
8	39
9	45
1	59
7	65
5	83
4	86
10	88
2	100

Example: Creating a Data Set from a Hash Object

Considerations and Modifications

- Allows multiple sets of data items per key value

```
data _null_;
  attrib sequence length=8 label='Selection Order'
         random_number length=8 label='Random Number';

  declare hash r(ordered: 'yes', multidata: 'yes');
  r.definekey('random_number');
  r.definedata('sequence','random_number');
  r.definedone();

  call missing(sequence,random_number);
  call streaminit(12345);

  do sequence=1 to 10;
    random_number=ceil(100*rand('uniform'));
    rc=r.add();
  end;

  rc=r.output(dataset: 'random10');
run;
```

Example: Creating a Data Set from a Hash Object

Data Set RANDOM10 with duplicate random numbers generated

sequence	random_number
6	29
8	39
9	45
1	59
3	59
7	65
5	83
4	86
10	88
2	100

Example: Creating a Data Set from a Hash Object

Considerations and Modifications

- Code-dependent and run-time naming of output data sets

```
data _null_;
  attrib sequence length=8 label='Selection Order'
         random_number length=8 label='Random Number';
  declare hash r(ordered: 'yes', multidata: 'yes');
  r.definekey('random_number');
  r.definedata('sequence','random_number');
  r.definedone();
  call missing(sequence,random_number);
  call streaminit(12345);

  numbers=15;
  do sequence=1 to numbers;
    random_number=ceil(100*rand('uniform'));
    rc=r.add();
  end;

  rc=r.output(dataset: cats('random',put(numbers,2.),
                            '_le50(where=(random_number le 50))'),
              dataset: cats('random',put(numbers,2.), '_',
                            put(today(),date9.)));
run;
```

Example: Creating a Data Set from a Hash Object

Considerations and Modifications

- Code-dependent naming of output data sets

```
207
208     rc=r.output(dataset:
  cats('random',put(numbers,2.),'_le50(where=(random_number
  le 50))'),
209                   dataset:
  cats('random',put(numbers,2.),'_',put(today(),date9.)));
210   run;


NOTE: The data set WORK.RANDOM15_LE50 has 5 observations and
  2 variables.
NOTE: The data set WORK.RANDOM15_11OCT2012 has 15
  observations and 2 variables.
```

Example: Creating a Data Set from a Hash Object

Considerations and Modifications

- Specifying KEY and DATA argument tags

```
data _null_;
  attrib sequence length=8 label='Selection Order'
         random_number length=8 label='Random Number';
  declare hash r(ordered: 'yes', multidata: 'yes');

  r.definekey('random_number');
  r.definedata('sequence','random_number');
  r.definedone();

  call missing(sequence,random_number);
  call streaminit(12345);

  do i=1 to 10;
    n=ceil(100*rand('uniform'));
    rc=r.add(key: n, data: i+1000, data: n);
  end;

  rc=r.output(dataset: 'random10');
 run;
```

Example: Creating a Data Set from a Hash Object

Data Set RANDOM10 after specifying the KEY and DATA argument tags in the ADD method

sequence	random_number
1006	29
1008	39
1009	45
1001	59
1003	59
1007	65
1005	83
1004	86
1010	88
1002	100

Example: Creating a Data Set from a Hash Object

Considerations and Modifications

- Specifying KEY and DATA argument tags

```
89      do i=1 to 10;
90        n=ceil(100*rand('uniform'));
91        rc=r.add(key: n, data: i+1000, data: n);
92        put _all_;
93      end;
94
95      rc=r.output(dataset: 'random10');
96    run;
sequence=. random_number=. i=1 n=59 rc=0 _ERROR_=0 _N_=1
sequence=. random_number=. i=2 n=100 rc=0 _ERROR_=0 _N_=1
sequence=. random_number=. i=3 n=59 rc=0 _ERROR_=0 _N_=1
sequence=. random_number=. i=4 n=86 rc=0 _ERROR_=0 _N_=1
sequence=. random_number=. i=5 n=83 rc=0 _ERROR_=0 _N_=1
sequence=. random_number=. i=6 n=29 rc=0 _ERROR_=0 _N_=1
sequence=. random_number=. i=7 n=65 rc=0 _ERROR_=0 _N_=1
sequence=. random_number=. i=8 n=39 rc=0 _ERROR_=0 _N_=1
sequence=. random_number=. i=9 n=45 rc=0 _ERROR_=0 _N_=1
sequence=. random_number=. i=10 n=88 rc=0 _ERROR_=0 _N_=1
```

Example: Creating a Data Set from a Hash Object

Considerations and Modifications

- Specifying KEY and DATA argument tags

```
62   data _null_;
63     declare hash r(ordered: 'yes', multidata: 'yes');
64
65     r.definekey('random_number');
66     r.definedata('sequence','random_number');
67     r.definedone();
68
69     call streaminit(12345);
70
71     do i=1 to 10;
72       n=ceil(100*rand('uniform'));
73       rc=r.add(key: n, data: i+1000, data: n);
74       put _all_;
75     end;
76
77     rc=r.output(dataset: 'random10');
78    run;
ERROR: Undeclared key symbol random_number for hash object at line
   67 column 3.
ERROR: DATA STEP Component Object failure.  Aborted during the
   EXECUTION phase.
NOTE: The SAS System stopped processing this step because of errors.
```

Summary

- Description of what a hash object is
- Overview of hash object syntax and terminology
- A few simple examples

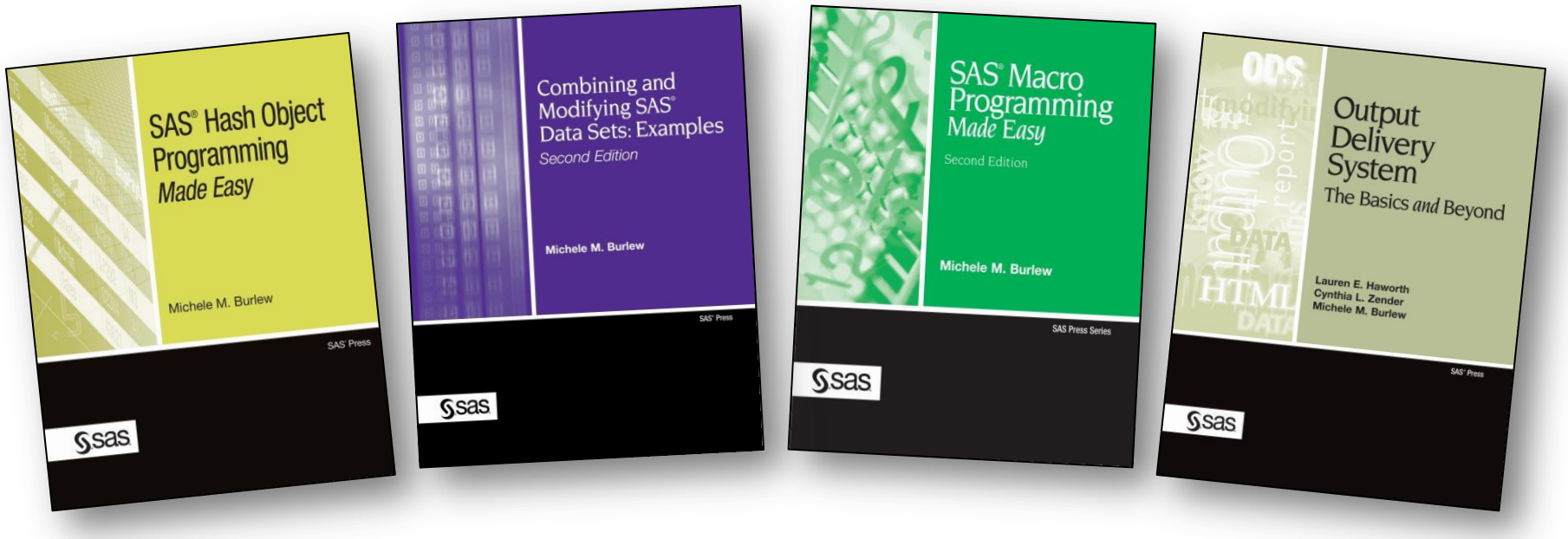
Summary: Where to Go from Here

- Start simple
- Test your processing with familiar tools
- Experiment to gain understanding of hash object syntax
- Remember processing constraints

Questions?

Additional Resources

- Michele Burlew Author page



- SAS Talks on support.sas.com

support.sas.com