

Look Me Up Baby:
The whys & wherefores of table lookup

Robert Rosofsky
Health Information Systems Consulting



March 21, 2007

Table Lookup Programming Goals

- Get your data out of the code
- Make the project easy to maintain
- Enable changes to values by non-programmers or other systems within your organization

The beginner way

```
Data Whatever;  
  Set WhichEver;  
  If sex="F" then SexName="Female";  
  Else SexName="Male";
```

```
Run;
```

```
Data Whatever;  
  Set WhichEver;  
  If sex="F" then SexName="Female";  
  Else if sex="M" then SexName="Male";  
  Else SexName="*Invalid*";
```

```
Run;
```

The beginner way (continued)

```
Data Whatever;  
  Set WhichEver;  
  If St="VT" then State="Vermont";  
  Else If St="NH" then State="New Hampshire";  
  Else If St="ME" then State="Maine";  
  Else If St="MA" then State="Massachusetts";  
  Else If St="RI" then State="Rhode Island";  
  Else If St="CT" then State="Connecticut";  
Run;
```

What about all 50 states??

...and the territories

...and the Canadian Provinces

...and all countries?

Many Key Values

```
Data Whatever;  
  Set WhichEver;  
  if ICD9="001" then Description="CHOLERA";  
  else if ICD9="0019" then Description="CHOLERA NOS";  
  else if ICD9="0020" then Description="FEVER TYPHOID";  
  else if ICD9="0059" then Description="POISONING FOOD";  
Run;
```

... and thousands of additional codes

Multiple Key Values = 2

```
Data Whatever;  
  Set WhichEver;  
  If Proc="98765" then do;  
    If Provider= "A123456" then Pay=85.00;  
    Else If Provider= "B789544" then Pay=86.40;  
    Else If Provider= "D001234" then Pay=87.80;  
    ... and hundreds of other providers  
  End;  
  Else If Proc="96666" then do;  
    If Provider= "A123456" then Pay=95.20;  
    Else If Provider= "B789544" then Pay=96.60;  
    Else If Provider= "D001234" then Pay=97.95;  
    ... and hundreds of other providers  
  End;  
  Else If ...thousands of additional codes  
Run;
```

Multiple Key Values = 3

```
Data Whatever;  
  Set WhichEver;  
  If Proc="98765" then do;  
    If Provider= "A123456" then do;  
      If "01JAN2004"D <= DOS <= "30JUN2004"D then Pay=85.00;  
      Else If "01JUL2004"D <= DOS <= "30SEP2004"D then Pay=85.50;  
      Else If "01OCT2004"D <= DOS <= "31DEC2004"D then Pay=87.25;  
      ... 1 year of rates (3 separate rate periods)  
    End;  
    If Provider= "B789544" then do;  
      If "01JAN2004"D <= DOS <= "30JUN2004"D then Pay=85.30;  
      Else If "01JUL2004"D <= DOS <= "30SEP2004"D then Pay=85.55;  
      Else If "01OCT2004"D <= DOS <= "31DEC2004"D then Pay=87.80;  
    End;  
  End;  
  ... and multiple providers  
  ... and thousands of additional codes  
Run;
```

No More Hard Coding: Merging

PatientID	State	Health_Plan
123456	TN	Medicaid
678910	MA	Neighborhood Health Plan
987654	AZ	Aetna
321654	CT	Kaiser
789789	ME	Dirigo
998877	ID	Consumer Choice
665544	OR	Go Anywhere Health Care
332211	OR	Consumer Directed
012567	MN	Veterans Administration
789545	MI	Medicare

Patients

Abbreviation	State_Name
AL	ALABAMA
AK	ALASKA
AZ	ARIZONA
...	...
WY	WYOMING

States

No More Hard Coding: Merging

```
proc sort data=Patients;  
    by State;  
run;
```

```
proc sort data=States;  
    by Abbreviation;  
run;
```

```
data Patients_States;  
    merge Patients (in=p)  
           States (in=s rename=(Abbreviation=State));  
    By State;  
    If p;  
run;
```

No More Hard Coding: Merging With SQL

```
proc sql;  
  create table Patients_States as  
  select p.*, s.State_Name  
  from Patients as p  
    left join  
      States as s  
    on p.State = s.Abbreviation  
  order by State;  
quit;
```

No More Hard Coding: Merging (2 Keys)

Proc	Provider	PayRate
98765	A123456	85.00
98765	B789544	86.40
98765	D001234	87.80
96666	A123456	95.20
96666	B789544	96.60
96666	D001234	97.95
94100	A123456	95.00
94100	B789544	96.40
94100	D001234	97.80

PayRates

PatientID	State	Health_Plan	Proc	Provider
123456	TN	Medicaid	98765	A123456
578910	MA	Neighborhood Health Plan	94222	B789544
987654	AZ	Aetna	95123	D001234
321654	CT	Kaiser	94100	Z998877
789789	ME	Dirigo	98765	T665588
998877	ID	Consumer Choice	94100	M884512
365544	OR	Go Anywhere Health Care	85621	Q789865
332211	OR	Consumer Directed	45895	K789854
012567	MN	Veterans Administration	87635	K985421
789545	MI	Medicare	87635	A987654

Patients

No More Hard Coding: Merging (2 Keys)

```
proc sort data=Patients;  
  by Proc Provider;  
run;
```

```
proc sort data=Patients;  
  by PatientID;  
run;
```

```
proc sort data=PayRates;  
  by Proc Provider;  
run;
```

```
data Patients_States;  
  merge Patients (in=p)  
        PayRates (in=r);  
  by Proc Provider;  
  If p;  
run;
```

No More Hard Coding: Merging (2 Keys) With SQL

```
proc sql;  
  create table Patients_Rates as  
  select p.*, s.PayRate  
  from Patients as p  
    left join  
      PayRates as r  
    on p.Proc = r.Proc  
      and p.Provider = r.Provider  
  order by PatientID;  
quit;
```

User-Defined FORMATS

Abbreviation	State Name
AL	ALABAMA
AK	ALASKA
AZ	ARIZONA
...	...
WY	WYOMING

This table...

Format Name: \$STATE LENGTH: 30 NUMBER OF VALUES: 50 MIN LENGTH: 1 MAX LENGTH: 40 DEFAULT LENGTH 30 FUZZ: 0		
Start	End	Label
AL	AL	ALABAMA
AK	AK	ALASKA
AZ	AZ	ARIZONA
...	...	...
WY	WY	WYOMING

Can become this FORMAT

User-Defined FORMATS

...and used as follows with the **put()** function

```
Data Whatever;  
  Set WhichEver;  
  State = put(St,$State.);  
Run;
```

User-Defined FORMATs

How do we create them?

Hard coded method

```
Proc Format;  
  value $State  
    "AL" = "ALABAMA"  
    "AK" = "ALASKA"  
    "AZ" = "ARIZONA"  
    ...  
    "WY" = "WYOMING"  
;  
run;
```

User-Defined FORMATs

How do we create them?

Dataset Method

```
Data StateFMTS;  
    Retain  FMTname "STATE"      Type "C" ;  
    Set StateList  
    (Rename=(Abbreviation=Start  
              Name=Label));  
  
Run;  
  
Proc Format CNTLIN=StateFMTS;  
Run;
```

User-Defined FORMATs

How do we create them? (2 Key Values)

Procedure Code	Provider	Payment
98765	A123456	85.00
98765	B789544	86.40
98765	D001234	87.80
96666	A123456	95.20
96666	B789544	96.60
96666	D001234	97.95
94100	A123456	95.00
94100	B789544	96.40
94100	D001234	97.80

This table...

Format Name: \$PAY			LENGTH: 30	NUMBER OF VALUES: 50
MIN LENGTH: 1			MAX LENGTH: 40	DEFAULT LENGTH 30 FUZZ: 0
Start	End	Label		
98765^A123456	98765^A123456	85.00		
98765^B789544	98765^B789544	86.40		
98765^D001234	98765^D001234	87.80		
96666^A123456	96666^A123456	95.20		
96666^B789544	96666^B789544	96.60		
96666^D001234	96666^D001234	97.95		
94100^A123456	94100^A123456	95.00		
94100^B789544	94100^B789544	96.40		
94100^D001234	94100^D001234	97.80		

Can become this FORMAT

User-Defined FORMATS

How do we USE them? (2 Key Values)

...and used as follows, again with the **put()** function

```
Data Whatever;  
Set WhichEver;  
Key= Proc_Code || "^" || ProviderID;  
Payment_Char= put(key,$pay.);  
Payment= input(Payment_Char,6.2);  
Run;
```

User-Defined FORMATs

How do we create them? (2 Key Values)

Hard coded method

```
Proc Format;
```

```
  value $Pay
```

```
    "98765^A123456" = "85.00"
```

```
    "98765^B789544" = "86.40"
```

```
    "98765^D001234" = "87.80"
```

...and thousands of lines

```
;
```

```
run;
```

User-Defined FORMATs

How do we create them? (2 Key Values)

Dataset Method

```
Data PayPMTS;  
  Retain FMTname 'Pay' Type 'C' ;  
  Set Payment_Rates;  
  Start = Proc_Code || '^' || ProviderID;  
  Label = put(Rate,6.2);  
Run;  
  
Proc Format CNTLIN= PayFMTS;  
Run;
```

User-Defined FORMATs

How do we create them? (3 Key Values)

Procedure Code	Provider	Effective_Date	Payment
98765	A123456	10/1/05	85.00
98765	A123456	1/1/06	85.50
98765	A123456	7/1/06	86.10
98765	A123456	1/1/07	86.65
98765	B789544	10/1/05	86.00
98765	B789544	1/1/06	86.10
98765	B789544	7/1/06	86.75
98765	B789544	1/1/07	86.95

This table...

Format Name: \$PAYDT LENGTH: 30		
MIN LENGTH: 1 MAX LENGTH: 40		
Start	End	Label
98765^A123456^16710	98765^A123456^16801	85.00
98765^A123456^16802	98765^A123456^16982	85.50
98765^A123456^16983	98765^A123456^17166	86.10
98765^A123456^17167	98765^A123456^17531	86.65
98765^B789544^16710	98765^B789544^16801	86.00
98765^B789544^16802	98765^B789544^16982	86.10
98765^B789544^16983	98765^B789544^17166	86.75
98765^B789544^17167	98765^B789544^17531	86.95

Can become this FORMAT

User-Defined FORMATs

How do we USE them? (3 Key Values)

...and used as follows, again, with the **put()** function

```
Data Whatever;  
Set Whichever;  
Key= Proc_Code || '^' || ProviderID ||  
      '^' || put(DOS,5.);  
Payment_Char=put(key,$paydt.);  
Payment= input(Payment_Char,6.2);  
Run;
```

User-Defined FORMATs

Retrieving Multiple Values

PatientID	Name	Address	City_St_Zip
012567	Hutchins, Sam	871 47 Street	Brooklyn, NY 11220
123456	Kablanka, Mira	22 Webster Avenue	Brooklyn, NY 11230
321654	Bloviator, Richard	39 Orchard Lane	Urbana, IL 60654
332211	Cipisin, Stella	4756 Allenhurst Road	Buffalo, NY 14215
665544	Mercyme, Bruce	512 Minnesota Ave	Buffalo, NY 14214
678910	Buckley, Gerda	345 Waldo Road	Milton, MA 02186
789545	Wilkinson, Louis	59 Beacon Street Circle	Rock Hills, ID 84511
789789	Brooks, Sandra	223 Hinckley Road	Plano, TX 75447
987654	Costa, John	9887 Brooks Avenue	Little Rock, AK 35448
998877	Sacco, Leslie	83 Flowering Lane	Utica, NY 15447

```
Data All_Patient_Data;  
  Merge PatientID(in = p)  
    PatientData (in = d);  
  By PatientID;  
  If p;  
Run;
```

User-Defined FORMATs

Retrieving Multiple Values

```
Data Patient_Info;  
    Retain fmtname "patinfo" type "C";  
    Set PatientData;  
    Start = PatientID;  
    Label =  
        Name || "^" || Address || "^" || City_State_Zip;  
    Drop Name Address City_State_Zip;  
Run;  
  
Proc format cntlin=Patient_Info fmtlib;  
Run;
```

User-Defined FORMATs

Retrieving Multiple Values

Format Name: \$patinfo LENGTH: 30 NUMBER OF VALUES: 5658		
MIN LENGTH: 1 MAX LENGTH: 40 DEFAULT LENGTH 30 FUZZ: 0		
Start	End	Label
012567	012567	Hutchins, Sam^871 47 Street^Brooklyn, NY 11220
123456	123456	Kablanka, Mira^22 Webster Avenue^Brooklyn, NY 11230
321654	321654	Bloviator, Richard^39 Orchard Lane^Urbana, IL 60654
332211	332211	Cipisin, Stella^4756 Allenhurst Road^Buffalo, NY 14215
665544	665544	Mercyme, Bruce^512 Minnesota Ave^Buffalo, NY 14214
678910	678910	Buckley, Gerda^345 Waldo Road^Milton, MA 02186
789545	789545	Wilkinson, Louis^59 Beacon Street Circle^Rock Hills, ID 84511
789789	789789	Brooks, Sandra^223 Hinckley Road^Plano, TX 75447
987654	987654	Costa, John^9887 Brooks Avenue^Little Rock, AK 35448
998877	998877	Sacco, Leslie^83 Flowering Lane^Utica, NY 15447

User-Defined FORMATS

Retrieving Multiple Values

```
Data Whatever;  
Set WhichEver;  
Values = put(pat_id,$patinfo.);
```

* Use **SCAN** function to parse values;

```
Name = scan(Values,1,"^");  
Street_Address = scan(Values,2,"^");  
City_State_Zip= scan(Values,3,"^");  
Run;
```

Performance

Dataset $\approx$ 1.1 million observations

Lookup Table $\approx$ 16,000 values

Method	Time
Hard-coding	27min 18 sec
Merge (includes sort steps)	6.8 sec
SQL Merge	13.8 sec
User-defined Format	27.9 sec (+ 0.3 sec to create format)

Finale

Questions?...

Robert Rosofsky
Health Information Systems Consulting
www.HealthInfoSys.net