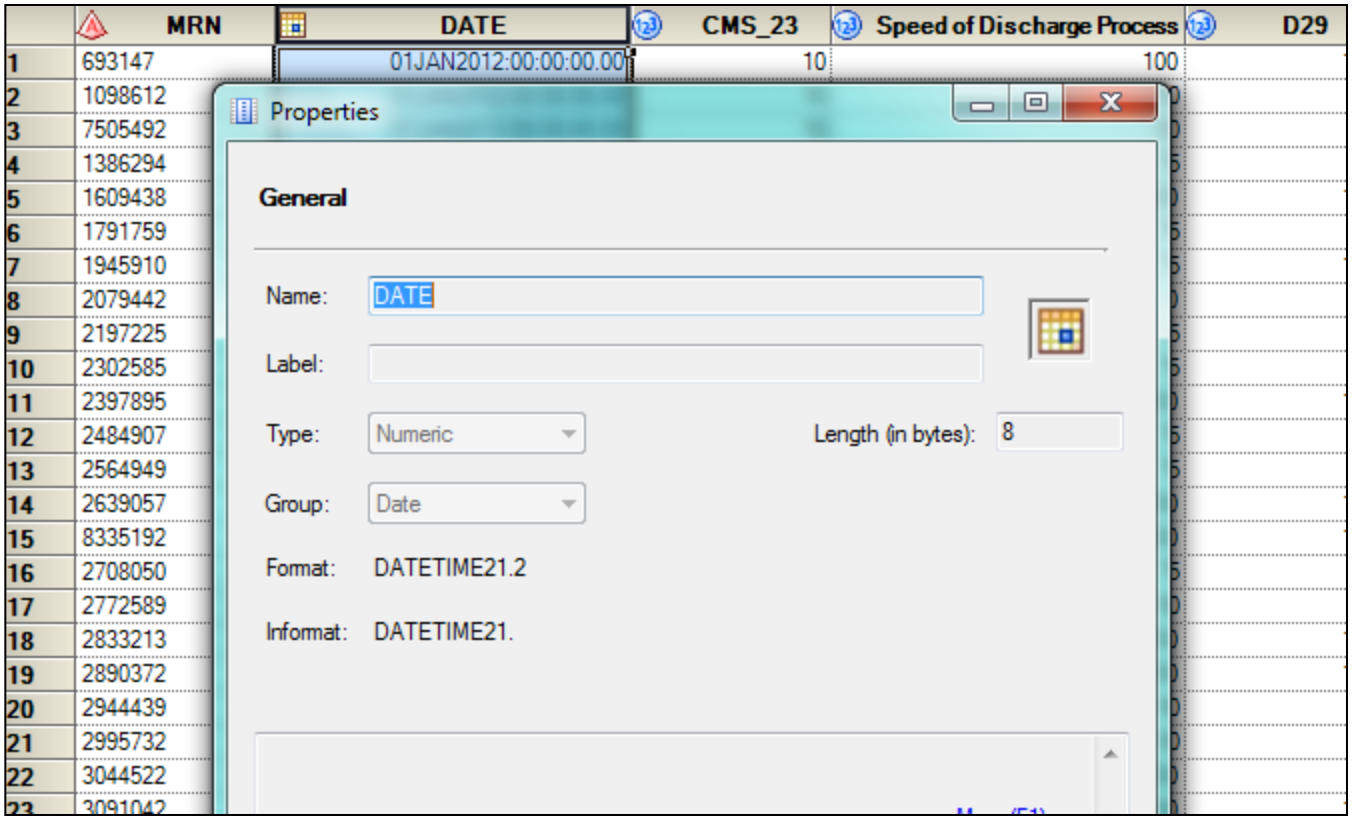
Dates and Times in SAS



SYSTEMS SEMINAR CONSULTANTS, INC.

Questions, Comments

- Technical Difficulties: Call 1-800-263-6317
- We have several hundred attendees, so we won't be able to answer questions live.
- Please email questions or comments to train@sys-seminar.com.
- A copy of the presentation and a recording of the webinar will be emailed out to attendees. This can be shared with people who did not attend.

Systems Seminar Consultants, Inc.

Consulting, Training, and Support

- SAS
- SQL
- ETL
- Reporting Systems
- Business Analytics
- Data modeling
- Automating Processes
- “Big Data”

Apply good IT and systems practices to real life business applications.

Work with Marketing, Finance, Retail, Insurance, Utilities, Manufacturing, Healthcare, Government organizations.

Steven J. First, President

- Over 30 years of SAS experience, including hundreds of manufacturing, retail, government, marketing, and financial applications.
- Over 25 years as President and Founder of SSC
- Founder of WISAS and WISUG
- Invited speaker at local, regional, and international user groups

Dates and Times in SAS

SAS provides powerful date and time handling tools.

Problems with dates and times:

- Date and time are cyclic number systems: 1 to 12 to 1, 1 to 7 to 1, etc.
- The number of days in a month and year vary.
- There are many date and time formats:

  01JUL2013 07/01/13 2013-07-31
  1:37PM 13:37

- Dates and times may be mixed together and you only want one alone.
- Many applications need to do date math

SAS Date and Time Tools

How does SAS solve these display and arithmetic problems?

SAS Dates

- Dates are represented as the number of days from January 1, 1960.
- Times are represented as the number of seconds since midnight.
- Date-times are the number of seconds since January 1, 1960.
- Numeric dates allow date arithmetic between years 1582-20000.
- Dates before Jan 1, 1960 are negative numbers.

Notes:

- Dates, Date-times, and times read from databases (oracle, Sql Server Excel etc), are automatically converted to respective SAS values.

Reading and Displaying Dates From Text Files

Read, store, and display a date:

1. Match a date INFORMAT with the format of the date you are reading.
2. The INFORMAT converts the date text to a SAS numeric value.
3. Use a date FORMAT to display the SAS numeric value as familiar text.

fileref RAWIN

```
19MAY12   05/19/12   01:36:02   05MAY12:01:36:02
```

```
data timedemo;
    infile rawin;
    input @1  d1 date7.  @10 d2 mmddyy8.
          @20 t1 time8.  @30 dt datetime16.;
    datealone=datepart(dt);
    timealone=timepart(dt);
    putlog 'Unformatted ' d1= d2= t1= dt= datealone= timealone=;
    putlog 'Formatted   ' d1= mmddyy10. d2= mmddyy10.  t1= time.
           dt= datetime22.2  datealone= mmddyy10.  timealone= time.;
run;
```

Reading and Displaying Dates (continued)

Partial log:

```
Unformatted d1=19132 d2=19132 t1=5762 dt=1651800962
            datealone=19118 timealone=5762
Formatted   d1=05/19/2012 d2=05/19/2012 t1=1:36:02 dt=05MAY2012:01:36:02.00
            datealone=05/05/2012 timealone=1:36:02
```

Reading and Displaying Dates From Excel Files

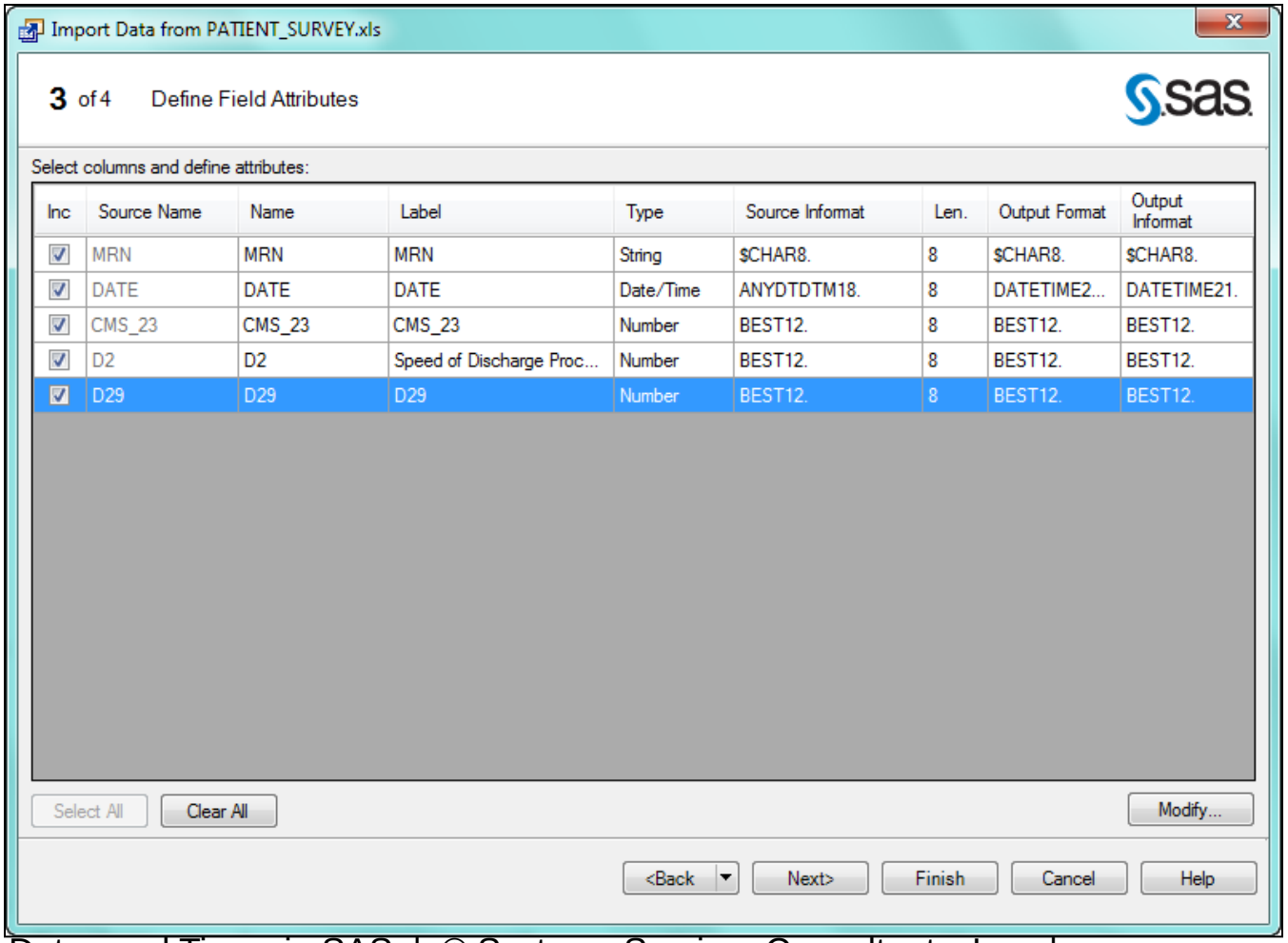
Using the import wizard, specify an appropriate informat and format.

Example:

	A	B	C	D	E
1	MRN	DATE	CMS_23	D2	D29
2	693147	January 1, 2012	10	100	100
3	1098612	January 1, 2012	10	50	50
4	7505492	January 1, 2012	10	50	50
5	1386294	January 1, 2012	11	75	
6	1609438	January 1, 2012	11	100	100
7	1791759	January 1, 2012	8	75	75
8	1945910	January 1, 2012	10	75	100
9	2079442	January 1, 2012	8	0	0
10	2197225	January 1, 2012	9	75	50
11	2302585	January 1, 2012	10	75	75
12	2397895	January 1, 2012	11	100	100
13	2484907	January 1, 2012	10	75	75
14	2564949	January 1, 2012	10	75	75
15	2639057	January 2, 2012	11	100	100

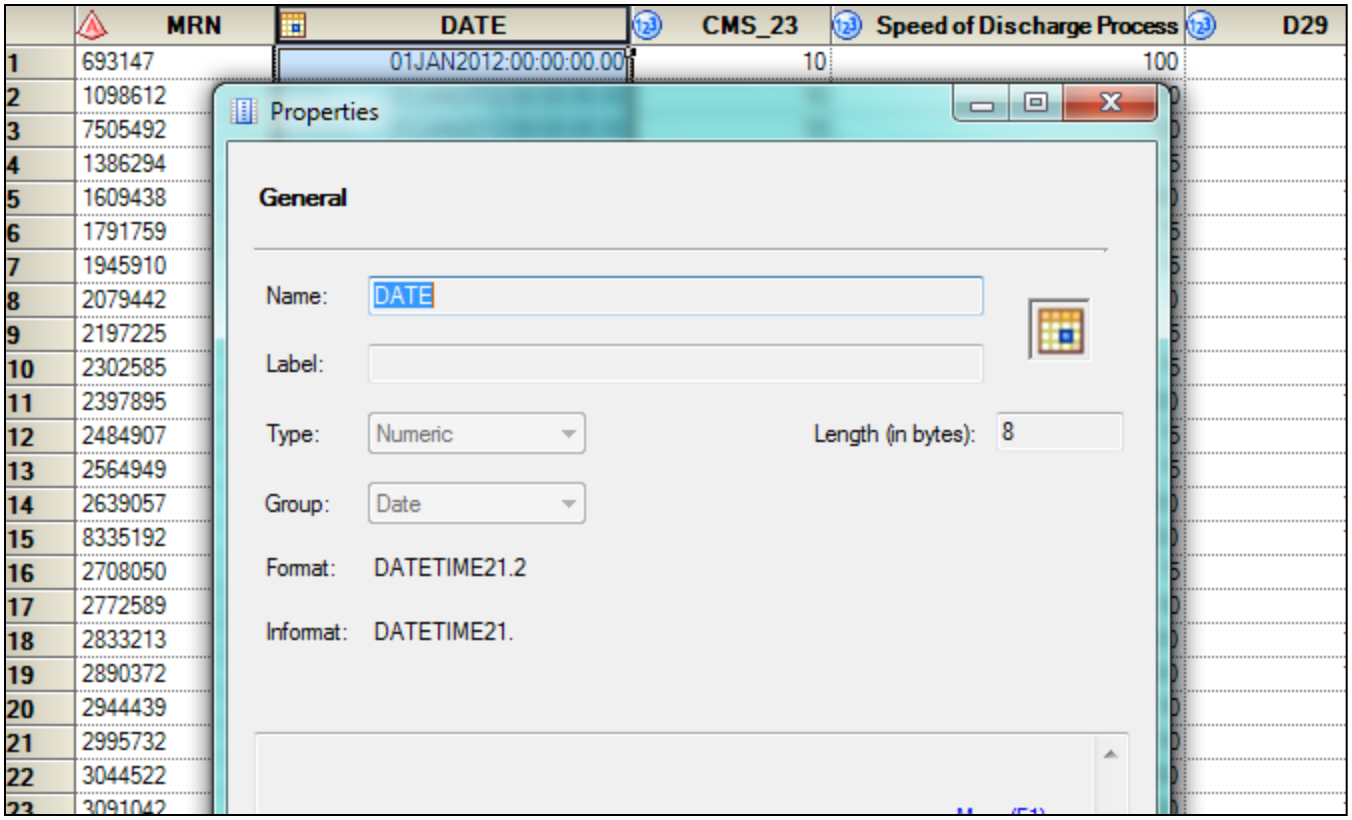
EG Excel Import - Dates

ANYDTDTM reads most types of dates and datetimes and returns a SAS datetime.



The Resulting SAS File

DATE is imported and can be used for date math.



SAS Date INFORMATS and FORMATS

SAS provides INFORMATs and FORMATs for most styles of date and time.

FORMAT	DESCRIPTION	EXAMPLE
DATE*w*.	dates of form ddMMMyy	31JAN2013
DATETIME*w.d*	date-time values	31JAN2013:10:02:01
DDMMYY*xw*.	date values	31/01/2013
HHMM*w.d*	hours and minutes	12:23
HOUR*w.d*	hour	11.5
JULIAN*w.*	Julian dates	2013031
MMDDYY*xw*.	date values	01/31/2013
MMSS*w*.	minutes and seconds	59:30
MONYY*w*.	month and year	JAN2013
MMYY*xw*.	date values	01M13
NENGOW*w*.	Japanese dates	
PDTIME*w*.	packed decimal time from SMF/RMF	
RMFDUR*w*.	RMF time interval measurements	
RMFSTAMP*w*.	time-date from RMF	

SAS Date INFORMATS and FORMATS (continued)

SMFSTAMP*w*.	time-date from SMF	
TIME*w.d*	time values	14:21:23
TOD*w*.	time of day from date-time value	3:25:3
TODSTAMP*w*.	date-time from store clock instr	
TU*w*.	timer units	
WEEKDATE*w*.	date values	WEDNESDAY
WORDDATE*w*.	date values	JANUARY 31, 2013
YYMM*xw*.	date values	13M01
YYMMDD*xw*.	date values	20130131
YYQ*w*.	year and quarter	13Q1

Others

SAS Date and Time Functions

Date and time functions are powerful tools for manipulating date and time.

FUNCTION	DESCRIPTION
DATE	returns today's date
DATEJUL	converts a Julian date to a SAS date
DATEPART	extracts a date from a SAS date-time value
DATETIME	returns the current date and time
DAY	returns the day of the month from a SAS date
DHMS	returns a SAS date-time from a date, hour, min, and sec
HMS	returns the SAS time from an hour, min, and sec
HOUR	returns the hour from a SAS time or datetime
INTCK	returns the number of intervals between dates
INTNX	advances a date or time by an interval
JULDATE	returns a JULIAN date from a SAS date
MDY	returns a SAS date from month, day, and year
MINUTE	returns a minute from a SAS time and date-time
MONTH	returns the month from a SAS date

SAS Date and Time Functions (continued)

QTR	returns the quarter from a SAS date
SECOND	returns the second from a SAS time or date-time
TIME	returns the current time of day
TIMEPART	returns the time part of the date-time
TODAY	gets the current date as a SAS date
WEEKDAY	returns the day of the week from a SAS date
YEAR	returns the year from a SAS date
YYQ	returns a SAS date from a year and quarter

Note: SAS documentation has a full description of date functions.

Date FORMAT Separators and Day of the Week

Some formats allow the specification of a separator value.

N = no separator **B** = blank **C** = colon **D** = dash **P** = period **S**=slash

DDMMYY***s****w*.	DDMMYY**D**10.	24-06-2014
MMDDYY***s****w*.	MMDDYY**C**10.	06:24:2014
YYMMDD***s****w*.	YYMMDD**N**8.	20140624

Changing the width (*w*) of the WEEKDATE format allows you to list the spelled out day of the week only, or with date and year with various levels of abbreviation:

WEEKDATE9.	- Wednesday
WEEKDATE20.	- Wed, Sep, 12, 2012
WEEKDATE29.	- Wednesday, September 12, 2012

Example Date Functions

Date and time functions can be used in data steps and query builder.

```
data function;
 todate=today();            /* Today's date as SAS numeric */
 sasdate=todate;            /* Save it in another variable */
 yearout=year(todate);      /* Today's year as a number    */
 qtrout=qtr(todate);        /* Today's quarter (1-4)       */
 monthout=month(todate);    /* Today's month (1-12)        */
 weekday=weekday(todate);   /* Today's weekday (1-7)       */
 day=day(todate);           /* Day of the month (1-31)     */
run;

proc print;
   title 'Dates';
   format  sasdate date9.;
run;
```

Dates

Obs	todate	sasdate	yearout	qtrout	monthout	weekday	day
1	19318	21NOV2012	2012	4	11	4	21

Date, Time, and Date-Time Constants

You can specify a date, time, or datetime as a constant.

'03MAY13'D	SAS date constant (two-digit year)
'03MAY2013'D	SAS date constant (four-digit year)
'8:27'T	SAS time constant
'03MAY2013:8:27'DT	SAS datetime constant

Note: There is no space between the closing quote and the constant indicator (D, T, DT).

Either single or double quotes can be used.

Date Constant Example

Example: Print a bonus list for employees hired before 01JAN2012.

```
                         123456789012345678901234567890
  fileref               ┌──────────────────────────────┐
   RAWIN                │SAM           06/18/11        │
                        │MARY          02/14/12        │
                        │BARB          05/10/09        │
                        └──────────────────────────────┘
```

```
data bonuslst;
   infile rawin;
   input @1 name $10. @15 startdte mmddyy8.;
   if startdte < '01jan2012'd;
run;

proc print;
   title 'EMPLOYEES EMPLOYED BEFORE 01JAN2012';
   format startdte mmddyy10.;
run;
```

```
              EMPLOYEES EMPLOYED BEFORE 01JAN2012

                  OBS    name    startdte

                   1     SAM     06/18/2011
                   2     BARB    05/10/2009
```

Example: Passing a Date to a SAS Program

You can pass a date constant to a batch SAS job with the SYSPARM option.

Example: Batch job to print a bonus list for employees hired before 01JAN2012.

```
                           1234567890123456789012
             fileref      SAM           06/18/11
             RAWIN        MARY          02/14/12
                          BARB          05/10/09
```

```
MVS           //  EXEC SAS,OPTIONS='SYSPARM="01JAN2012" '
TSO           SAS OPTIONS('SYSPARM="01JAN2012" ')
Windows       C:\SAS\SAS.EXE -SYSPARM "01JAN2012"
```

```
data bonuslst;
 infile rawin;
 input @1 name $10. @15 startdte mmddyy8.;
 if startdte < "&sysparm"d ;
run;
```

Resolves to "01JAN2012"D

SYSPARM Date Example Output

```
proc print;
 title "EMPLOYEES EMPLOYED BEFORE &SYSPARM ";
 format startdte mmddyy8.;
run;
```

EMPLOYEES EMPLOYED BEFORE 01JAN2012

OBS	name	startdte
1	SAM	06/18/11
2	BARB	05/10/09

Date and Time Arithmetic

To determine the number of days between two SAS dates simply subtract one from the other:

```
startToEnd = end - start;
```

INTNX advances a date, time, or datetime value by an interval.

Syntax: **INTNX**(*interval, start-from*, *increment*<, *alignment*>)

INTCK counts the number of intervals between two dates.

Syntax: **INTCK**(*interval, from, to*)

INTNX and INTCK Intervals

The INTNX and INTCK interval argument may be one of the following:

Date Intervals	Datetime Intervals	Time Intervals
DAY	DTDAY	HOUR
WEEKDAY	DTWEEKDAY	MINUTE
WEEK	DTWEEK	SECOND
TENDAY	DTTENDAY	
SEMIMONTH	DTSEMIMONTH	
MONTH	DTMONTH	
QTR	DTQTR	
SEMIYEAR	DTSEMIYEAR	
YEAR	DTYEAR	

Note:

- The type of interval must match the type of period (date, time, datetime) you are working with.
- See SAS documentation for the INTERVALDS= system option for information on creating custom intervals.

INTNX Example

```
data test;
  data test;
  start_date = '10MAR2000'd;
   nextmonth1 = intnx( 'month', start_date, 1 );
  thismonth1 = intnx( 'month', start_date, 0);
  month12back = intnx( 'month', start_date, -12);
  format start_date nextmonth1 thismonth1 month12back  date9.;
run;
proc print data=test;
run;
```

The SAS System

Obs	start_ date	nextmonth1	thismonth1	month12back
1	10MAR2000	01APR2000	01MAR2000	01MAR1999

Note: INTNX increments by "firsts of" the interval. E.g., first of the month if no alignment is specified.

INTCK Example

Count intervals between dates or times.

```
data test;
  start_date = '10MAR2000'd;
  end_date = '31OCT2005'd;

  num_year = intck('year', start_date, end_date);
  num_month = intck('month',start_date, end_date);
  num_days = intck('day',start_date, end_date);
  num_weeks = intck('week', start_date, end_date);

  format start_date end_date date9.;
run;

proc print data=test;
run;
```

INTCK Example (continued)

The SAS System

Obs	start_ date	end_date	num_year	num_ month	num_days	num_ weeks
1	10MAR2000	31OCT2005	5	67	2061	295

Note: By default INTCK counts "firsts of" the interval.

Examples Using Alignment Argument

Sas Statement	Result
date1=intnx('month','01jan95'd,5,'beginning'); put date1 / date1 date7.;	12935 01JUN95
date2=intnx('month','01jan95'd,5,'middle'); put date2 / date2 date7.;	12949 15JUN95
date3=intnx('month','01jan95'd,5,'end'); put date3 / date3 date7.;	12964 30JUN95
date4=intnx('month','01jan95'd,5,'sameday'); put date4 / date4 date7.;	12935 01JUN95
date5=intnx('month','15mar2000'd,5,'same'); put date5 / date5 date9.;	14837 15AUG2000
Interval='month'; date='1sep2001'd; align='m'; date4=intnx(interval,date,2,align); put date4 / date4 date7.;	15294 15NOV01
x1='month '; x2=trim(x1); date='1sep2001'd + 90; date5=intnx(x2,date,2,'m'); put date5 / date5 date7.;	15356 16JAN02

INTNX Additional Options

Optional arguments allow you to use more complex interval specifications.

INTNX(*interval*<*multiple*><*.shift-index*>, *start-from*, *increment*<,*alignment*>)

multiple	sets the interval equal to a multiple of the interval type.
shift-index	specifies the starting point of the interval.

Notes:

- Both the *multiple* and the *shift-index* arguments are optional.
- Valid units of *shift-index* vary with *interval*. For example, years are shifted by months.
- The value of the *shift-index* cannot be greater than the number of sub-periods within the *multiple*. Example: YEAR2.25 is not valid because the shift for year is month and a 2 year period only contains 24 months.
- See SAS documentation for a complete list of possible shift intervals.

Example INTNX with Shift and Multiple

```
data test;
  start_date = '15FEB2006'd;

  day3 = intnx('day3',start_date,1);          /* 3 day interval beg on Sun */

  week2 = intnx('week2',start_date,1);        /* Biweekly beg on 1st Sun */

  week_mon = intnx('week.2',start_date,1);    /* Begin on Mon, notf Sun */

  fiscal_year = intnx('year.3',start_date,1); /* Fiscal yr begins in Mar */

  year2_even = intnx('year2.4',start_date,1); /* Biennial beg Apr even yrs */

  year2_odd = intnx('year2.16',start_date,1); /* Biennial beg Apr odd yrs */
run;

proc print data=test;
  format start_date day3 week2 week_mon fiscal_year year2_even
         year2_odd  date9.;
run;
```

Example INTNX with Shift and Multiple (continued)

The SAS System

Obs	start_ date	day3	week2	week_mon	fiscal_ year	year2_ even	year2_odd
1	15FEB2006	16FEB2006	19FEB2006	20FEB2006	01MAR2006	01APR2006	01APR2007

Note:

- SAS assigns shifted periods relative to Jan. 1, 1960. The date you pass into INTNX is incremented within a series of periods shifted from Friday, Jan 1, 1960.
- It's not always obvious how this will work out!
- Check SAS documentation and expect to run test programs to verify your shifted intervals.

INTCK Additional Options

```
INTCK(interval<multiple> <.shift-index>,
      start-date, end-date, <'method'> )
```

multiple	set the interval equal to a multiple of the interval type.
shift-index	specify a shifted starting point for the interval.
method	DISCRETE or CONTINUOUS (D or C). Measure continuous time (e.g., calendar years) or count discrete time intervals (e.g., firsts of the year).

Note: The *shift-index* works as with the INTNX function using Friday, Jan 1, 1960 as the start point. Results may not be obvious.

Be prepared to test your program.

INTCK with CONTINUOUS

Use the CONTINUOUS method to determine age.

- DISCRETE counts firsts of the year (the default).
- CONTINUOUS counts elapsed time.

```
data _null_;
    birthday = '15mar1979'd;
    howOldThisDate = '14mar2019'd;

    discreteAge = INTCK('years', birthday,
       howOldThisDate, 'discrete' ) ;
    continuousAge = INTCK('years', birthday,
       howOldThisDate, 'continuous' ) ;

   putlog discreteAge=;
   putlog continuousAge=;
run;
```

INTCK with CONTINUOUS to Calculate Age

Results of the previous program.

```
discreteAge=40
continuousAge=39
```

39 is the correct age.

Why the different ages?

On March 14 a person born on March 15 has not yet seen their birthday but they have passed the first of the year. First of the year is the DISCRETE counting point.

Note: INTCK with CONTINOUS always handles leap years correctly so it is a reliable method for calculating age.

YEARCUTOFF

When SAS encounters a two-digit year, the YEARCUTOFF= option controls which century to use.

Use the following PROC OPTIONS step to display YEARCUTOFF at your site:

```
proc options
      option= yearcutoff;
run;
```

How YEARCUTOFF= Works

Two Digit Year	Date	Two Digit Year	Date
00	2000	26	2026
01	2001	27	2027
02	2002	28	2028
03	2003	29	2029
04	2004	**30**	**1930**
05	2005	31	1931
06	2006	32	1932
07	2007	33	1933
08	2008	34	1934
09	2009	35	1935
10	2010	36	1936
11	2011	37	1937
12	2012	38	1938
13	2013	39	1939
14	2014	40	1940
15	2015	41	1941
16	2016	42	1942
17	2017	43	1943
18	2018	44	1944
19	2019	45	1945
20	2020	46	1946
21	2021	47	1947
22	2022	48	1948
23	2023	49	1949
24	2024	50	1950
25	2025	… 99	…1999

OPTIONS YEARCUTOFF=1930;

Up to the two digit year before YEARCUTOFF the next century is used.

After and including YEARCUTOFF the YEARCUTOFF century is used.

Date and Time Summary

One approach:

- Determine if you need a date (days), a time(seconds) , or a datetime(seconds).
- Get to SAS dates, times, datetimes as soon as possible.
- If you are reading from a text file, use the appropriate INFORMAT to read values correctly.
- If you are reading from a database, dates will convert automatically.
- If you have a character value from a database, convert it to a SAS value using the input function with appropriate informat.
- If you need to separate a date and time from a datetime, use datepart and timepart functions.
- Use output formats that you like for dates and times.
- Dates will sort correctly and you can do date math.

Questions, Comments

- Please email questions or comments to train@sys-seminar.com or contact us at 1-800-997-7081.

- A copy of the presentation and a recording of the webinar will be emailed out to attendees. This can be shared with people who did not attend.

Upcoming Presentations

SAS Global Forum

The Query Builder: The Swiss Army Knife of SAS® Enterprise Guide®

Best Practices in SAS® Enterprise Guide®

Next Free Lunch and Learn

The Query Builder: The Swiss Army Knife of SAS® Enterprise Guide®

April 16, Noon Central, Register at www.sys-seminar.com

Upcoming Presentations and Training

SAS Training with SSC – Live Web or in Madison, WI

Register at www.sys-seminar.com or call 1-800-997-7081

SAS Report Writing	March 3-4
Introduction to PROC Report	March 5
Exploiting SAS ODS	March 6-7
The SAS SQL Procedure	April 7
SAS Efficiencies	April 8
SAS Macros	April 9-10
Advanced SAS Macros	April 11
SAS Enterprise Guide	May 5-6
Introduction to SAS	May 7-9

Contact Us

SYSTEMS SEMINAR CONSULTANTS, INC.

2997 Yarmouth Greenway Drive • Madison, WI 53711

(608) 278-9964 | 1-800-997-7081

www.sys-seminar.com

sas | SAS Alliance *Silver Member*

Steven First

President

sfirst@sys-seminar.com