Wanna Date?
Steve Rhoades, IMS Health, Plymouth Meeting, Pa

ABSTRACT
Ever need to read a date into SAS or write it out? Need to know the number of days between two days, or the weeks, months, years? What about incrementing (or decrementing) a date to the first of next month, or the fifteenth or; well you get the idea. SAS is very, very flexible and offers quite a few ways to handle dates. However it is not always apparent what is available. The intent of this paper is to provide a concise presentation of what is available and includes a file of self contained code that allows you to run a number of the informats, formats and date functions and see the results quickly. International dates are not covered thoroughly in this paper, but there are separate sections for both informats and formats that do provide some details.

INTRODUCTION
Originally I was going to cover SAS Dates, Times and Date-Times, but as the number of pages kept growing I was forced to cut back dramatically. Actually I just moved them to the accompanying files along with directories containing SAS programs. Check the zip file to see this. In addition, International and Special Date-Time Informats and Formats have been moved to the zip file. It is very important to know that SAS Dates, Times and Date-Times are different and mixing them will create incorrect results. There are a number of other papers that discuss how to handle time and date-time values and how to switch SAS variables between them. If you look at the SAS values of the variables, pure date values will never have a decimal portion; date-time values can have decimals, although they do not always have them. So what can happen? I mixed and matched a date-time SAS variable with a date format and ended up with a date for 1960 even though the actual year was 2006. In talking to SAS Support there is no definitive way to tell them apart, you need to know your data to avoid this problem. Fortunately many systems are written using only dates rather than SAS date-times and SAS times so this problem can be avoided. Beware that it is a potential problem and be sure to review any dates to make sure this is not a problem. Charley Mullin of SAS has written a technical document TS-668 that provides more details on this topic.

As I want to keep this fairly concise I concentrate on limited explanations of the date informats, formats and functions, and provide examples of the data fields being read in and written out. If you are interested in more detailed explanations the SAS web site should fill this need.

YEARCUTOFF WHEN THERE IS NO CENTURY
If the century is not included in a date, the YEARCUTOFF option is used to determine the century. In version 9.1.3, the default YEARCUTOFF is currently set to 1920. Two digit years below 20 will use ‘20’ as the century, two digit years from 20 to 99 will be assigned a century of ‘19’. Notice that if the date should be ‘1812’, the date would end up as ‘2012’. Either a System option or the Options statement within your SAS session can overwrite the YEARCUTOFF. To verify the value of YEARCUTOFF run the following code:

    Proc options option=yearcutoff; run;

INFORMAT WIDTH ISSUES
SAS is pretty flexible when reading in dates; however, if the informat width is less than the width of the date field, you will end up with missing values, or worse, the wrong date. Here’s what I mean.

Suppose you are reading at an input date field that contains 01$JAN$2006 (I told you SAS is flexible) and you use a DATEw. informat of less than 11 (the above date field is 11 characters). In reading in the above date field during a test ‘DATE.’, ‘DATE7.’, ‘DATE10.’ Informats resulted in missing values, ‘DATE8.’ and ‘DATE9.’ Informats resulted in dates of ‘01JAN2002’ and ‘01JAN1920’.
SELF-CONTAINED PROGRAMS
There seventy-eight self-contained SAS programs included in the zipped file. By running these programs you can see first hand how a number of the informats, formats and functions work and can tailor these for specific needs. A sample program follows:

```sas
/*------------------------------------------------------------------------*/
/* ANYDTDTEW.sas (extracts DATE values, NOT TIME) */
/* INFORMAT */
/* Default width is 9 */
/* Valid widths are 5-32 */
/* Valid informat values must be in any of these forms: */
/* date, datetime, ddmmyy, julian, mmdyy, monyy, time, yymmdd, yyq */
/* If it is a time value will become Jan 1, 1960, SAS Value 0 */
/* */
/* Very flexible, use if there are various date formats in incoming data. */
/* See individual informat SAS programs for additional details of the */
/* various forms. */
/* Note: Dates will be incorrect if the width of informat less than WIDTH */
/* of value read in. Time informat results in date of Jan 1, 1960 */
/* Ambiguous month/day/year combinations are read using the DATESTYLE */
/* system option to decide the order. */
/* See log for actual SAS Date values */
/* S. Rhoades 5/19/2006 */
/*------------------------------------------------------------------------*/

Data Dates;
INFILE Datalines;
Input @1 AnyDtDte      AnyDtDte. /* Width here is 9*/
     @1 AnyDtDte5     AnyDtDte5. /* If value > 5, will not convert to valid SAS date*/
     @1 AnyDtDte15    AnyDtDte15. /* If value > 15, will not convert to valid SAS date*/
     @1 AnyDtDte25    AnyDtDte25.
     @1 AnyDtDte30    AnyDtDte30.
     @1 AnyDtDte32    AnyDtDte32.
;
Put 'SAS Dates ' _ALL_;
Datalines;
01jan06
28FEB2006/12:12:12.3 PM
31/03/2006
2006095
05312006
JUN2006
02:30:32
20060731
2006Q3
;
Proc Print Data=Dates;
    anydtdte25 anydtdte30 anydtdte32 ddmmyy10.;
Title 'print of dates, informat is ANYDTDTEW., various formats for print using both
data/datetime';
run;
```
DATE INFORMAT DESCRIPTIONS

- **ANYTDTEw.** DATE Informat. Default width is 9, valid widths are 5-32
  - Valid informat values may be in any of these forms:
    - `date`, `datetime`, `ddmmyy`, `julian`, `mmddyy`, `monyy`, `time`, `yymmdd`, `yyq`
  - Sample input values:
    - `01jan06`, `28FEB2006/12:12:12.3 PM`, `1/03/2006`, `2006095`, `05312006`, `JUN2006`, `02:30:32`, `20060731`, `2006Q3`
  - Where order of month, day, year are ambiguous DATESTYLE System option indicates order
  - If reading in only a TIME value, the date is Jan 1, 1960

- **DATEw.** default width is 7, Valid widths are 7-32
  - SAMPLE INPUT DATE VALUES
    - `01$JAN$2006`, `2FEB06`, `03mar06`, `04-APR06`, `5may2006`, `06jun 2006`, `7/jul/2006`, `08/Aug/06`

- **DATETIMEw.** Default width is 18, Valid widths are 13-40,
  - Valid datetime values must be in this form: `ddmmmyy` (or `yyyy`) blank or special character `hh:mm:ss:ss`
  - Ex: `31dec2006$23:59:59:12`  (NOTE: `ss:ss` portion is optional)
  - SAS can read in AM or PM within the datetime values.
  - Sample input values:
    - `31jan2006$23:59:59.12`, `28Feb2006#13:1:2.3`, `31mAR06@01:59 PM`, `31mAR06 01:59 AM`, `31mAR06*01:59 PM`, `1Apr06)13:1:2`, `7May06#13:1:2.3`, `28Jun2006!13:1:2.3`

- **DDMMYYw.** Default width is 6, valid widths are 6-32.
  - Valid forms are `ddmmmyy` or `ddmmyyyy` and blanks or delimiters between the components are valid.
  - Sample input values:
    - `310106`, `28022006`, `31/03/2006`, `04 30 2006`, `05 1 2005`, `6$3$04`

- **JULIANw.** Default width is 5, valid widths are 5-32.
  - Valid forms are `yydd`, `yyydd`, `yydd`, `yyyydd`, blanks or delimiters between the components are not valid.
  - Julian dates prior to 1582 are invalid for conversion
  - Sample input values:
    - `0001`, `200001`, `1401001`, `23365`, `02005`, `2006180`, `2002200`

- **MMDDYYw.** Default width is 6, valid widths are 6-32.
  - Valid forms are `mmddyy` or `mmddyyyy` and blanks or delimiters between the components are valid.
  - Sample input values:
    - `013006`, `1022006`, `03/01/2006`, `04 30 2006`, `05 1 2005`, `6$3$04`

- **MONYYw.** Default width is 5, valid widths are 5-32.
  - Valid forms are `mmmyy` or `mmmyyyy` and blanks or delimiters between the components are valid.
  - Sample input values:
    - `JAN 06`, `feb 2006`, `Mar$06`, `Apr/2006`, `May%2006`

- **PDw.d** – Is an IBM packed decimal format that may be on multiple platforms. It is listed here as sometimes dates are written in packed formats.
  - Within a packed field dates can be written a variety of ways, once the data is read into SAS it needs to be converted by using put/input functions and SAS date informat
  - See NOTE2 for an example. Also the SAS 9.1.3 Language Reference: Dictionary has a good example of creating a SAS Date on page 1003.
- **PDJULGw** – Is an IBM packed decimal format that may be on multiple platforms. It reads packed Julian date values. Both the default and the range are four bytes.
  - Packed field data can be positive, negative or absolute, SAS ignores the sign when using this informat.
  - Example: The packed values 2006365C or 2006365D or 2006365F all create a SAS date equivalent to Dec 31, 2006. Packed values contain two values per byte when looking at them in hex mode, if not in hex mode you will not be able to read the values.

- **PDJULIw** – Is an IBM packed decimal format that may be on multiple platforms. It reads packed Julian date values. Both the default and the range are four bytes. NO DIFFERENCE WAS FOUND BETWEEN THIS INFORMAT AND PDJULGw.
  - Packed field data can be positive, negative or absolute, SAS ignores the sign when using this informat.
  - Example: The packed values 2006365C or 2006365D or 2006365F all create a SAS date equivalent to Dec 31, 2006. Packed values contain two values per byte when looking at them in hex mode, if not in hex mode you will not be able to read the values.

- **WEEKUw** – Default width is 11, valid widths are 3-200
  - Forms are Www, yyWww, yyWwwdd, yyyyWwwdd, yyyy-Www-dd
  - Examples: W01, 07W01, 06W0101, 2006W0102, 2006-W01-03
  - Sunday, day of week = 1

- **WEEKVw** – Default width is 11, valid widths are 3-200
  - Forms are Www, yyWww, yyWwwdd, yyyyWwwdd, yyyy-Www-dd
  - Examples: W01, 07W01, 06W0101, 2006W0102, 2006-W01-03
  - Monday, day of week = 2 (Must include Jan 4th and first Thursday of year)

- **WEEKWw** – Default width is 11, valid widths are 3-200
  - Forms are Www, yyWww, yyWwwdd, yyyyWwwdd, yyyy-Www-dd
  - Examples: W01, 07W01, 06W0101, 2006W0102, 2006-W01-03
  - Monday, day of week = 2

- **YYMMDww** – Default width is 6, valid widths 6-32
  - Forms are yymmdd or yyyymmdd
  - Examples: 060130, 20061102, 2006 12 31, 2006/07/04, 06#05#29

- **YYMMNww** – Default width is 4, valid widths 4-6
  - Forms are yymm or yyyymm
  - Examples: 0512, 20063, 200604
  - Monday, day of week = 2

- **YYQww** – Default width is 6, valid widths 4-32
  - Forms are yyQq, yyQqq, yyyyQq, yyyyQqq
  - Examples: 06Q01, 06Q01, 2006Q1, 2006Q01

### TABLE OF SAS DATE INFORMATS

Read in external data, convert to SAS dates, write out using a variety of SAS formats.

**NOTE:** Once a value is converted to a SAS date, any date format can be used for output. Again, beware that date, datetimes, and times are different. If you use a date format on a SAS date that used a datetime informat, expect incorrect results.

<table>
<thead>
<tr>
<th>SAS informat</th>
<th>Data format</th>
<th>Input value</th>
<th>SAS format (out)</th>
<th>Output value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDDMMYYYY10.</td>
<td>Dddmmmyy</td>
<td>30#04#2006</td>
<td>DDMYY10.</td>
<td>30/04/2006</td>
</tr>
<tr>
<td>DATE07.</td>
<td>ddmmmyy</td>
<td>01jan98</td>
<td>DATE07.</td>
<td>01jan98</td>
</tr>
<tr>
<td>DATE07.</td>
<td>ddmmmyy</td>
<td>01jan98</td>
<td>DATE09.</td>
<td>01jan1998</td>
</tr>
<tr>
<td>DATE09.</td>
<td>ddmmmyyccyy</td>
<td>01jan1998</td>
<td>DATE09.</td>
<td>01jan1998</td>
</tr>
<tr>
<td>DDDMMYYYY8.</td>
<td>Dddmmccyy</td>
<td>31121998</td>
<td>DDMYY6.</td>
<td>311298</td>
</tr>
<tr>
<td>DDDMMYYYY8.</td>
<td>Dddmmccyy</td>
<td>31121998</td>
<td>DDMYY7.</td>
<td>311298</td>
</tr>
<tr>
<td>DDDMMYYYY8.</td>
<td>Dddmmccyy</td>
<td>31121998</td>
<td>DDMYY8.</td>
<td>31/12/98</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ddmmyyc10.</td>
<td></td>
</tr>
<tr>
<td>JULIAN5.</td>
<td>julian - yyddd</td>
<td>01365</td>
<td>JULIAN5.</td>
<td>01365</td>
</tr>
<tr>
<td>JULIAN5.</td>
<td>julian - yyddd</td>
<td>01365</td>
<td>JULIAN7.</td>
<td>2001365</td>
</tr>
<tr>
<td>JULIAN7.</td>
<td>julian - ccyyddd</td>
<td>2001365</td>
<td>JULIAN7.</td>
<td>2001365</td>
</tr>
<tr>
<td>MMDDYY6.</td>
<td>mm-dd-yy</td>
<td>02-03-95</td>
<td>MMDDYY6.</td>
<td>020395</td>
</tr>
<tr>
<td>MMDDYY6.</td>
<td>mm-dd-yy</td>
<td>02-03-95</td>
<td>MMDDYY8.</td>
<td>020395</td>
</tr>
<tr>
<td>MMDDYY6.</td>
<td>mm-dd-yy</td>
<td>02-03-95</td>
<td>MMDDYY8.</td>
<td>02/03/95</td>
</tr>
<tr>
<td>MMDDYY8.</td>
<td>mm dd yy</td>
<td>01 01 90</td>
<td>MMDDYY6.</td>
<td>010190</td>
</tr>
<tr>
<td>MMDDYY8.</td>
<td>mm dd yy</td>
<td>01-01-00</td>
<td>MMDDYY6.</td>
<td>010100</td>
</tr>
<tr>
<td>MMDDYY8.</td>
<td>mm/dd/yy</td>
<td>01/01/01</td>
<td>MMDDYY6.</td>
<td>010101</td>
</tr>
<tr>
<td>MMDDYY8.</td>
<td>mm/dd/yy</td>
<td>110190</td>
<td>MMDDYY10.</td>
<td>122301</td>
</tr>
<tr>
<td>MMDDYY8.</td>
<td>mm/dd/yyyy</td>
<td>2001365</td>
<td>MMDDYY8.</td>
<td>12.23.2001</td>
</tr>
<tr>
<td>MMDDYY8.</td>
<td>mm dd yy</td>
<td>01 01 90</td>
<td>MMDDYY7.</td>
<td>010190</td>
</tr>
<tr>
<td>MMDDYY8.</td>
<td>mm/dd/yyyy</td>
<td>01-01-00</td>
<td>MMDDYY7.</td>
<td>010100</td>
</tr>
<tr>
<td>MMDDYY8.</td>
<td>mm/dd/yyyy</td>
<td>01/01/01</td>
<td>MMDDYY7.</td>
<td>110190</td>
</tr>
<tr>
<td>MMDDYY8.</td>
<td>mm/dd/yyyy</td>
<td>12230201</td>
<td>MMDDYY7.</td>
<td>122301</td>
</tr>
<tr>
<td>MMDDYY8.</td>
<td>mm/dd/yyyy</td>
<td>01 01 90</td>
<td>MMDDYY8.</td>
<td>01/01/90</td>
</tr>
<tr>
<td>MMDDYY8.</td>
<td>mm/dd/yyyy</td>
<td>01-01-00</td>
<td>MMDDYY8.</td>
<td>01/01/00</td>
</tr>
<tr>
<td>MMDDYY8.</td>
<td>mm/dd/yyyy</td>
<td>01/01/01</td>
<td>MMDDYY8.</td>
<td>01/01/01</td>
</tr>
<tr>
<td>MMDDYY8.</td>
<td>mm/dd/yyyy</td>
<td>110190</td>
<td>MMDDYY8.</td>
<td>11/01/90</td>
</tr>
<tr>
<td>MMDDYY8.</td>
<td>mm/dd/yyyy</td>
<td>12230201</td>
<td>MMDDYY8.</td>
<td>12/23/01</td>
</tr>
<tr>
<td>MMDDYY8.</td>
<td>mm/dd-ccoy</td>
<td>01-01-1990</td>
<td>MMDDYY10.</td>
<td>010190</td>
</tr>
<tr>
<td>MMDDYY8.</td>
<td>mm/dd-ccoy</td>
<td>01/01/1990</td>
<td>MMDDYY8.</td>
<td>01/01/90</td>
</tr>
<tr>
<td>MMDDYY10.</td>
<td>mm/dd-ccoy</td>
<td>01011990</td>
<td>MMDDYY10.</td>
<td>0101/1990</td>
</tr>
<tr>
<td>MONYY5.</td>
<td>mony</td>
<td>Jan39</td>
<td>MONYY5.</td>
<td>JAN39</td>
</tr>
<tr>
<td>MONYY5.</td>
<td>mony</td>
<td>Jan39</td>
<td>MONYY7.</td>
<td>JAN2039</td>
</tr>
<tr>
<td>MONY7.</td>
<td>moncyy</td>
<td>Jan2250</td>
<td>MONYY5.</td>
<td>JAN50</td>
</tr>
<tr>
<td>MONY7.</td>
<td>moncyy</td>
<td>Jan2250</td>
<td>MONYY7.</td>
<td>JAN2250</td>
</tr>
<tr>
<td>PD4. *Note1</td>
<td>PD4 - ymmmd6.</td>
<td>0391201C</td>
<td>PD4. *Note2</td>
<td>0391201C</td>
</tr>
<tr>
<td>PD4. *Note1</td>
<td>PD4 - ymmmd6.</td>
<td>0391201F</td>
<td>PD4. *Note2</td>
<td>0391201C</td>
</tr>
<tr>
<td>PD5. *Note1</td>
<td>PD5 - ccymmddd8.</td>
<td>019391201C</td>
<td>PD5. *Note2</td>
<td>019391201C</td>
</tr>
<tr>
<td>WEEKUw.</td>
<td>weeku9.</td>
<td>2006W0102 Wk start-</td>
<td>weeku9.</td>
<td>2006W0102</td>
</tr>
<tr>
<td>WEEKVw.</td>
<td>weekv9.</td>
<td>2006W0102 Wk start-</td>
<td>weekv9.</td>
<td>2006W0102</td>
</tr>
<tr>
<td>WEEKWw.</td>
<td>weekw9.</td>
<td>2006W0102 Wk start-</td>
<td>weekw9.</td>
<td>2006W0102</td>
</tr>
<tr>
<td>YYMMD6.</td>
<td>yymmdd</td>
<td>961231</td>
<td>YYMMD2.</td>
<td>96</td>
</tr>
<tr>
<td>YYMMD6.</td>
<td>yymmdd</td>
<td>961231</td>
<td>YYMMD4.</td>
<td>9612</td>
</tr>
<tr>
<td>YYMMD6.</td>
<td>yymmdd</td>
<td>961231</td>
<td>YYMMD5.</td>
<td>96-12</td>
</tr>
<tr>
<td>YYMMD6.</td>
<td>yymmdd</td>
<td>961231</td>
<td>YYMMD8.</td>
<td>96-12-31</td>
</tr>
<tr>
<td>YYMMD6.</td>
<td>yymmdd</td>
<td>961231</td>
<td>YYMMD10.</td>
<td>1996-12-31</td>
</tr>
<tr>
<td>YYMMD8.</td>
<td>yymmdd</td>
<td>19961231</td>
<td>YYMMD8.</td>
<td>96-12-31</td>
</tr>
<tr>
<td>YYMMD8.</td>
<td>yymmdd</td>
<td>36 12 31</td>
<td>YYMMD10.</td>
<td>2036-12-31</td>
</tr>
<tr>
<td>YYMMD8.</td>
<td>yymmdd</td>
<td>36-12-31</td>
<td>YYMMD10.</td>
<td>2036-12-31</td>
</tr>
<tr>
<td>YYMMD8.</td>
<td>yymmdd</td>
<td>36/12/31</td>
<td>YYMMD10.</td>
<td>2036-12-31</td>
</tr>
<tr>
<td>YYMMD10.</td>
<td>yymmdd</td>
<td>21571231</td>
<td>YYMMD6.</td>
<td>571231</td>
</tr>
<tr>
<td>YYMMD10.</td>
<td>yymmdd</td>
<td>2157-12-31</td>
<td>YYMMD8.</td>
<td>57-12-31</td>
</tr>
<tr>
<td>YYMMD10.</td>
<td>yymmdd</td>
<td>2157 12 31</td>
<td>YYMMD10.</td>
<td>2157-12-31</td>
</tr>
<tr>
<td>YYMMDD10.</td>
<td>yymmdd</td>
<td>2157 12 31</td>
<td>YYMMDDxw.*</td>
<td>Yymmdds10.</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>---------</td>
<td>------------</td>
<td>-----------</td>
</tr>
<tr>
<td>YYMMDD10.</td>
<td>yymmdd</td>
<td>36/12/31</td>
<td>Day2.</td>
<td>31</td>
</tr>
<tr>
<td>YYMMDD10.</td>
<td>yymmdd</td>
<td>36/12/31</td>
<td>MONNAME3.</td>
<td>Dec</td>
</tr>
<tr>
<td>YYMMDD10.</td>
<td>yymmdd</td>
<td>36/12/31</td>
<td>MONNAME9.</td>
<td>December</td>
</tr>
<tr>
<td>YYMMDD10.</td>
<td>yymmdd</td>
<td>36/12/31</td>
<td>MONTH.</td>
<td>12</td>
</tr>
<tr>
<td>YYMMDD10.</td>
<td>yymmdd</td>
<td>36/01/31</td>
<td>MONTH.</td>
<td>1</td>
</tr>
<tr>
<td>YYMMDD10.</td>
<td>yymmdd</td>
<td>36/12/31</td>
<td>YEAR2.</td>
<td>36</td>
</tr>
<tr>
<td>YYMMDD10.</td>
<td>yymmdd</td>
<td>2136/12/31</td>
<td>YEAR4.</td>
<td>2136</td>
</tr>
<tr>
<td>yymmns4.</td>
<td>yymm</td>
<td>9912</td>
<td>Yymmdd10.</td>
<td>1999-01-12</td>
</tr>
<tr>
<td>yymmns4.</td>
<td>yymm</td>
<td>9912</td>
<td>Yymmns6.</td>
<td>199912</td>
</tr>
<tr>
<td>yymmns4.</td>
<td>yymm</td>
<td>9912</td>
<td>Yymmns4.</td>
<td>9912</td>
</tr>
<tr>
<td>yymmns4.</td>
<td>yymm</td>
<td>0001</td>
<td>Yymmns4.</td>
<td>1</td>
</tr>
<tr>
<td>Yyy4.</td>
<td>YyQq</td>
<td>06Q1</td>
<td>Yyy4.</td>
<td>06Q1</td>
</tr>
<tr>
<td>yyq6.</td>
<td>YyyYq</td>
<td>2006Q1</td>
<td>Yyy6.</td>
<td>2006Q1</td>
</tr>
<tr>
<td>Yyy7.</td>
<td>YyyYqq</td>
<td>2006Q01</td>
<td>YyyyYqq.</td>
<td>2006Q01</td>
</tr>
<tr>
<td>Today()</td>
<td>ddmmyy2.</td>
<td></td>
<td>07 &lt;= day 7 of mth</td>
<td></td>
</tr>
</tbody>
</table>

In the xw notation above, the w refers to the width of the field, x specifies the type of separator or no separator. Replace the x with one of the following letters to indicate the type of separator: B for blank, C for colon, D for dash, N for no separator, P for period, or S for slash.

**Note1** - Packed fields, typically on IBM systems, have two numbers in a single byte of data, a sign signified by C, D or F is on the right side and sometimes there is a leading 0. Packed fields can not be read in directly as date fields, instead they must be read in using a PD informat and converted to SAS dates. Example: The date 19981231 stored as PD5. Would look like:

```
09821
1913C
```

- Packed fields need to be converted to SAS dates (need to know input data informat - this example is yymmdd8.).
- The leading zero and the 'C' are not used in the conversion
- For PD5 logic: sasdate5= INPUT(PUT(in_pd_dt),8.),YYMMDD8.);

**Note2** - Packed fields, which have been converted to SAS dates, can be converted back to the original data format before being written to an external file as a packed field.
- Be aware that an ‘F’ appearing on input will be changed to ‘C’ on output.
  - For PD5 logic, century has to be concatenated to front.
    - Length Century $ 2;
    - Century=SUBSTR(LEFT(YEAR(sasdate5)),1,2); /*Getting century*/
    - Cnv_dk5=Century || PUT(sasdate5,YYMMDD6.);
- Code to write the data to an external file needs a PUT statement like: PUT @01 Cnv_dk5 PD5.;

**Note3** - To ensure leading zeroes are output for yymmns4. formats use this code
- PANLDATE is a SAS Date, for example the date is Jan, 2000
- CHARDT1=PUT(&PANLDATE,YYMMNS4.); /* CONV TO CHAR DATE OF YYMM */
- CHARDT=PUT(INPUT(CHARDT1,4.),Z4.); /* CONVERT TO NBR, THEN ADD 0’S*/
- OUTPUT IS 0001 Leading zeroes are put in the record

**FORMAT DESCRIPTIONS**

- **DATEw.** (DATE) Default width is 7, Valid widths are 5-9
  - Sample Output Date Values DATE5. and DATE6. =31DEC DATE7. and DATE8. =31DEC06, DATE9. =31DEC2006

- **DAYw.** (DATE) Default width is 2, Valid widths are 2-32
  - Writes the SAS Date as the day of the month.
  - Examples 1, 25, 31
- **DDMMYYw.** (DATE) Default width is 8, Valid widths are 2-10
  - Will write as little as day only or if there is room will add slashes between date components.
  - Will add leading zeroes if needed
  - Examples for Jan 2, 2007: 02, 0201, 02/01, 020107, 02/01/07, 02/01/2007

- **DOWNAMEw.** (DATE) Default width is 9, Valid widths are 1-32
  - Writes out mixed case day of week, if width not wide enough will write what it can
  - Examples for Dec 31, 2006: S, SU, SUN, SUND, SUNDA, SUNDAY

- **JULDAYS.** (DATE) Default width is 3, valid widths 3-32.
  - Shows day portion of the date with no leading zeroes.
  - Examples: 1, 12, 365

- **JULIANw.** (DATE) Default width is 5, valid widths 5-7 and it is left aligned.
  - Shows day portion of the date with leading zeroes.
  - Examples: 07001, 2007001

- **MMDDYYw.** (DATE) Default width is 8, Valid widths are 2-10
  - Will write as little as MONTH or if room will add slashes between date components.
  - Will add leading zeroes if needed
  - Examples for Jan 2, 2007: 01, 0102, 01/02, 010207, 01/02/07, 01/02/2007

- **MMDDYYxw.** Default width is 8, Valid widths are 2-10
  - Will write as little as MONTH only.
  - X is a specified separator B-blank, C-colon, D-dash, N-none, P-period, S-Slash
  - Ex: for Jan 2, 2007: 01, 0102, 010207, 01 02 07, 01.02.07, 01-02-07, 01/02/07, 01:01:2007

- **MMYYw.** (DATE) Default width is 7, Valid widths are 5-32
  - Will write in form of mmMyy(yy).
  - Will add leading zeroes if needed
  - Examples for Jan 2, 2007: 01M07, 01M2007

- **MMYYxw.** Default width is 7, Valid widths are 5-32
  - X is a specified separator B-blank, C-colon, D-dash, N-none, P-period, S-Slash

- **MONNAMEw.** (DATE) Default width is 9, Valid widths are 1-32
  - Examples for Dec 31, 2007: D, De, Dec, Decembe, December

- **MONTHw.** (DATE) Default width is 2, Valid widths are 1-21
  - MONTH1. is suppose to give hex values
  - Examples for Dec 31, 2007: C, 31

- **MONYYw.** (DATE) Default width is 5, Valid widths are 1-7
  - Examples for Dec 31, 2007: JAN07, JAN2007

- **PDJULGw.** (DATE) Default width is 4, Valid widths are 3-16
  - For IBM writes date values in hexadecimal format yyyydddF
  - Examples for Jan 31 and Dec 31, 2007: 2007031F, 2007365F

- **PDJULIw.** (DATE) Default width is 4, Valid widths are 3-16
  - For IBM writes date values in hexadecimal format yyyydddF
  - Subtracts 1900 from the four digit year to get the ccyy representation
  - Example for Dec 31, 2007: 0107031F

- **QTRw.** (DATE) Default width is 1, Valid widths are 1-32
  - Right justifies, outputs a single digit
  - Example for Dec 31, 2007: 4

- **QTRRw.** (DATE) Default width is 3, Valid widths are 3-32
  - Right justifies, outputs a Roman numerals for quarter
  - Example for Dec 31, 2007: IV

- **WEEKDATEw.** (DATE) Default width is 29, Valid widths are 3-37
  - Right justifies, outputs day-of-wk, mth name dd, yy(yy)
  - Leading Zeroes Dropped

- **WEEKDATXw.** (DATE) Default width is 29, Valid widths are 3-37
  - Right justifies, outputs day-of-wk, mth name dd, yy(yy)
Leading Zeroes Dropped

Ex: for Jan 1, 2007: Mon, Monday, Mon, 1 Jan, 07, Mon, 1 Jan, 2007, Monday, 1 January, 2007

WEEKUw. (DATE) Default is 11, widths are 3-200
o Writes a week number in decimal format using the U algorithm.
o Formats are Www, yyWww, (yy)yyWwwdd, yyyy-Www-dd dd=day of week
o Default day is first day of week - Sunday
o Note: zeroes for week below indicate it is not a full week
o Ex: for Jan 1, 2007: W00, 07W00, 07W0002, 2007-W00-02, 2007W0002, 2007-W00-02

WEEKVw. (DATE) Default is 11, widths are 3-200
o Writes a week number in decimal format using the V algorithm.
o Formats are Www, yyWww, (yy)yyWwwdd, yyyy-Www-dd dd=day of week
o Default day is first day of week – Monday, default year is current year.
o First week of year must contain Jan 4th and the first Thursday of year
o Examples for Jan 1, 2007: W01, 07W01, 07W01, 07W0101, 2007-W01-01, 2007W0101

WEEKWw. (DATE) Default is 11, widths are 3-200
o Writes a week number in decimal format using the W algorithm.
o Formats are Www, yyWww, (yy)yyWwwdd, yyyy-Www-dd dd=day of week
o Default year is current year, default day is first day of week – Monday, default year is current year.
o Examples for Jan 1, 2007: W01, 07W01, 07W01, 07W0101, 2007-W01-01, 2007W0101

WORDDATEw. (DATE) Default width is 18, Valid widths are 3-32
o Right justifies, outputs mth name dd, yy(yy)
o Leading Zeroes Dropped
o Examples for Jan 1, 2007: Jan, January, Jan 1, 2007, January 1, 2007

WORDDATXw. (DATE) Default width is 18, Valid widths are 3-32
o Right justifies, outputs dd mth name yy(yy)
o Leading Zeroes Dropped
o Examples for Jan 1, 2007: Jan, January, 1 Jan 2007, 1 January 2007

YEARw. (DATE) Default width is 4, Valid widths are 2-32
o Right justifies, outputs yy(yy) portion of date
o Examples for Jan 1, 2007: 07, 2007

YYMMw. (DATE) Default width is 7, Valid widths are 5-32
o Right justifies, outputs (yy)yyMmm
o Examples for Jan 1, 2007: 07M01, 2007M01

YYMmxw. (DATE) Default width is 7, Valid widths are 5-32
o Right justifies, outputs (yy)yyXmm X is B(lank) C(olon), D(ash), N(one), P(eriod), S(lash)

YYMMDDw. (DATE) Default width is 8, Valid widths are 2-10
o Right justifies, outputs (yy)ymmmdd or (yy)yy-mm-dd
o Examples for Jan 1, 2007: 07, 0701, 07-01, 070101, 07-01-01, 2007 01 01

YYMMDXXw. (DATE) Default width is 8, Valid widths are 2-10
o Right justifies, outputs (yy)ymmmmd or (yy)yyXmmXdd
o X is B(lank) C(olon), D(ash), N(one), P(eriod), S(lash)
o Examples for Jan 1, 2007: 07, 0701, 07:01, 0701, 07/01/01, 2007 01 01

YYMONw. (DATE) Default width is 7, Valid widths are 5-32
o Right justifies, outputs (yy)ymmm
o Examples for Jan 1, 2007: 07JAN, 2007JAN

YYQw. (DATE) Default width is 6, Valid widths are 4-32
o Right justifies, outputs (yy)yyQq
o Examples for Jan 1, 2007: 07Q1, 2007Q1

YYQxw. (DATE) Default width is 6, Valid widths are 4-32
o Right justifies, outputs (yy)yyQq or (yy)yyXq
o X is C(olon), D(ash), N(one), P(eriod), S(lash)
o Examples for Jan 1, 2007: 07JAN, 2007JAN
- **YYQRW. (DATE)** Default width is 8, Valid widths are 6-32
  - Right justifies, outputs (yy)yyQR
  - Quarters are Roman numerals
  - Examples for Jan 1, 2007: **2007Q1**

- **YYQRXW. (DATE)** Default width is 8, Valid widths are 6-32
  - Right justifies, outputs (yy)yyQr or (yy)yyXqR
  - X is C(olon), D(ash), N(one), P(eriod), S(lash)

### TABLE OF SAS DATE FORMATS

SAS Date formats are used to write out or display SAS dates. Once a variable becomes a SAS Date, any SAS Date format may be applied to it.

<table>
<thead>
<tr>
<th>DATE FORMATS</th>
<th>Default Width</th>
<th>Range</th>
<th>Value of the SAS Date</th>
<th>Format Written</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATEw.</td>
<td>7</td>
<td>5-9</td>
<td>December 31, 2006</td>
<td>Date5.</td>
<td>31DEC</td>
</tr>
<tr>
<td>DATEw.</td>
<td>7</td>
<td>5-9</td>
<td>December 31, 2006</td>
<td>Date6.</td>
<td>31DEC</td>
</tr>
<tr>
<td>DATEw.</td>
<td>7</td>
<td>5-9</td>
<td>December 31, 2006</td>
<td>Date. / Date7.</td>
<td>31DEC06</td>
</tr>
<tr>
<td>DATEw.</td>
<td>7</td>
<td>5-9</td>
<td>December 31, 2006</td>
<td>Date8.</td>
<td>31DEC06</td>
</tr>
<tr>
<td>DATEw.</td>
<td>7</td>
<td>5-9</td>
<td>December 31, 2006</td>
<td>Date9.</td>
<td>31DEC2006</td>
</tr>
<tr>
<td>DAYw.</td>
<td>2</td>
<td>2-32</td>
<td>January 01, 2006</td>
<td>Day2.</td>
<td>1</td>
</tr>
<tr>
<td>DAYw.</td>
<td>2</td>
<td>2-32</td>
<td>January 31, 2006</td>
<td>Day32.</td>
<td></td>
</tr>
<tr>
<td>DDMYYw.</td>
<td>8</td>
<td>2-8</td>
<td>January 2, 2007</td>
<td>DDMYY2.</td>
<td>02</td>
</tr>
<tr>
<td>DDMYYw.</td>
<td>8</td>
<td>2-8</td>
<td>January 2, 2007</td>
<td>DDMYY3.</td>
<td>02</td>
</tr>
<tr>
<td>DDMYYw.</td>
<td>8</td>
<td>2-8</td>
<td>January 2, 2007</td>
<td>DDMYY4.</td>
<td>0201</td>
</tr>
<tr>
<td>DDMYYw.</td>
<td>8</td>
<td>2-8</td>
<td>January 2, 2007</td>
<td>DDMYY5.</td>
<td>02/01</td>
</tr>
<tr>
<td>DDMYYw.</td>
<td>8</td>
<td>2-8</td>
<td>January 2, 2007</td>
<td>DDMYY6.</td>
<td>020107</td>
</tr>
<tr>
<td>DDMYYw.</td>
<td>8</td>
<td>2-8</td>
<td>January 2, 2007</td>
<td>DDMYY7.</td>
<td>020107</td>
</tr>
<tr>
<td>DDMYYw.</td>
<td>8</td>
<td>2-8</td>
<td>January 2, 2007</td>
<td>DDMYY8.</td>
<td>02/01/07</td>
</tr>
<tr>
<td>DDMYYw.</td>
<td>8</td>
<td>2-8</td>
<td>January 2, 2007</td>
<td>DDMYY9.</td>
<td>02/01/07</td>
</tr>
<tr>
<td>DDMYYw.</td>
<td>8</td>
<td>2-8</td>
<td>January 2, 2007</td>
<td>DDMYY10.</td>
<td>02/01/2007</td>
</tr>
<tr>
<td>DOWNAMEw.</td>
<td>9</td>
<td>1-32</td>
<td>December 31, 2006</td>
<td>DOWNAME.</td>
<td>Sunday</td>
</tr>
<tr>
<td>DOWNAMEw.</td>
<td>9</td>
<td>1-32</td>
<td>December 31, 2006</td>
<td>DOWNAME1.</td>
<td>S</td>
</tr>
<tr>
<td>DOWNAMEw.</td>
<td>9</td>
<td>1-32</td>
<td>December 31, 2006</td>
<td>DOWNAME2.</td>
<td>Su</td>
</tr>
<tr>
<td>DOWNAMEw.</td>
<td>9</td>
<td>1-32</td>
<td>December 31, 2006</td>
<td>DOWNAME3.</td>
<td>Sun</td>
</tr>
<tr>
<td>DOWNAMEw.</td>
<td>9</td>
<td>1-32</td>
<td>December 31, 2006</td>
<td>DOWNAME4.</td>
<td>Sund</td>
</tr>
<tr>
<td>DOWNAMEw.</td>
<td>9</td>
<td>1-32</td>
<td>December 31, 2006</td>
<td>DOWNAME5.</td>
<td>Suna</td>
</tr>
<tr>
<td>DOWNAMEw.</td>
<td>9</td>
<td>1-32</td>
<td>December 31, 2006</td>
<td>DOWNAME6.</td>
<td>Sunday</td>
</tr>
<tr>
<td>DOWNAMEw.</td>
<td>9</td>
<td>1-32</td>
<td>December 31, 2006</td>
<td>DOWNAME7.</td>
<td>Sunday</td>
</tr>
<tr>
<td>DOWNAMEw.</td>
<td>9</td>
<td>1-32</td>
<td>December 31, 2006</td>
<td>DOWNAME8.</td>
<td>Sunday</td>
</tr>
<tr>
<td>DOWNAMEw.</td>
<td>9</td>
<td>1-32</td>
<td>December 31, 2006</td>
<td>DOWNAME9.</td>
<td>Sunday</td>
</tr>
<tr>
<td>DOWNAMEw.</td>
<td>9</td>
<td>1-32</td>
<td>December 31, 2006</td>
<td>DOWNAME20.</td>
<td>Sunday</td>
</tr>
<tr>
<td>DOWNAMEw.</td>
<td>9</td>
<td>1-32</td>
<td>December 31, 2006</td>
<td>DOWNAME32.</td>
<td>Sunday</td>
</tr>
<tr>
<td>JULDAYw.</td>
<td>3</td>
<td>3-32</td>
<td>December 31, 2006</td>
<td>JULDAY3.</td>
<td>365</td>
</tr>
<tr>
<td>JULDAYw.</td>
<td>3</td>
<td>3-32</td>
<td>January 1, 2007</td>
<td>JULDAY4.</td>
<td>1</td>
</tr>
<tr>
<td>JULIANw.</td>
<td>3</td>
<td>5-7</td>
<td>January 1, 2007</td>
<td>JULAY5.</td>
<td>07001</td>
</tr>
<tr>
<td>JULIANw.</td>
<td>3</td>
<td>5-7</td>
<td>January 1, 2007</td>
<td>JULAY7.</td>
<td>2007001</td>
</tr>
<tr>
<td>MMDDYYw.</td>
<td>8</td>
<td>2-10</td>
<td>January 2, 2007</td>
<td>MMDDYY2.</td>
<td>01</td>
</tr>
<tr>
<td>MMDDYYw.</td>
<td>8</td>
<td>2-10</td>
<td>January 2, 2007</td>
<td>MMDDYY3.</td>
<td>01</td>
</tr>
<tr>
<td>MMDDYYw.</td>
<td>8</td>
<td>2-10</td>
<td>January 2, 2007</td>
<td>MMDDYY4.</td>
<td>0102</td>
</tr>
<tr>
<td>MMDDYYw.</td>
<td>8</td>
<td>2-10</td>
<td>January 2, 2007</td>
<td>MMDDYY5.</td>
<td>0102</td>
</tr>
<tr>
<td>MMDDYYw.</td>
<td>8</td>
<td>2-10</td>
<td>January 2, 2007</td>
<td>MMDDYY6.</td>
<td>010207</td>
</tr>
<tr>
<td>MMDDYYw.</td>
<td>8</td>
<td>2-10</td>
<td>January 2, 2007</td>
<td>MMDDYY7.</td>
<td>010207</td>
</tr>
<tr>
<td>MMDDYYw.</td>
<td>8</td>
<td>2-10</td>
<td>January 2, 2007</td>
<td>MMDDYY8.</td>
<td>01/02/07</td>
</tr>
<tr>
<td>MMDDYYw.</td>
<td>8</td>
<td>2-10</td>
<td>January 2, 2007</td>
<td>MMDDYY9.</td>
<td>01/02/07</td>
</tr>
<tr>
<td>MMDDYYw.</td>
<td>8</td>
<td>2-10</td>
<td>January 2, 2007</td>
<td>MMDDYY10.</td>
<td>01/02/2007</td>
</tr>
<tr>
<td>MMDDYYw.</td>
<td>8</td>
<td>2-10</td>
<td>January 2, 2007</td>
<td>MMDDYYC3.</td>
<td>01</td>
</tr>
<tr>
<td>MMDDYYw.</td>
<td>8</td>
<td>2-10</td>
<td>January 2, 2007</td>
<td>MMDDYYD4.</td>
<td>0102</td>
</tr>
<tr>
<td>MMDDYYw.</td>
<td>8</td>
<td>2-10</td>
<td>January 2, 2007</td>
<td>MMDDYYN5.</td>
<td>0102</td>
</tr>
</tbody>
</table>
**DATE FUNCTIONS**

- **DATDIF(StartSASDate,EndSASDate,CalculationBasis)** Calcs the nbr of days between two dates.
  - Calculation basis is either '30/360' (assumes yr is 360 days and each month is 30 days) or 'act/act' (actual number between dates)
  - Example: Start Date=01JAN2007 End Date=31DEC2007
    - Days30_306 = DATDIF(StartSASDate,EndSASDate,'30/360'); Days30_306=360
    - DaysACT_ACT = DATDIF(StartSASDate,EndSASDate,'ACT/ACT'); DaysACT_ACT=364
    - JustSubtract = sum(EndSASDate,-StartSASDate); JustSubtract=364
    - JustSubtract = EndSASDate-StartSASDate; JustSubtract=364 **NO FUNCTION!!!**

- **DATE() or TODAY()** Returns the current system date as a SAS date.
  - Example: SASDATE=DATE() If the date were 03JUN2006 the SASDate would be SASDATE=16955.
  - See dateFunctions.sas to run yourself.

- **DATEJUL(JulianDate)** Converts a Julian Date to a SAS date. Typical use would be when a SAS variable contains a Julian date.
  - Appears that both numeric and character SAS variables can be used
  - Examples: Input Juldate5 $Char5. Value is 06001. Code is SASJULDate5 = DateJul(Juldate5); The SASDate value is 16802.
  - Input Juldate7 7. Value is 20006001. Code is SASJULDate7 = DateJul(Juldate7); The SASDate value is 16802.
  - See DATEJUL.sas for more examples.

- **DAY(SASDate) DATE** function to select the day of the month from a SAS Date.
  - The resulting day is a numeric value with no leading zero.
  - Example: Actual date= 2006/31/01, SAS Date value=17160
  - SAS Code to get the day of the month `DayEq = DAY(SASDate);` DayEq value=31
  - See DAY.sas code for more examples.

- **INPUT(Source, <? Or ??>informat.)** Transform the value of a SAS variable into a value specified by the informat.
  - Example: SAS character variable is (mysas='20071231'), to make a SAS date

**DATE FUNCTIONS**

<table>
<thead>
<tr>
<th>YYMDDXw</th>
<th>8</th>
<th>2-10</th>
<th>January 1, 2007</th>
<th>YYMDDN3.</th>
<th>07</th>
</tr>
</thead>
<tbody>
<tr>
<td>YYMDDXw</td>
<td>8</td>
<td>2-10</td>
<td>January 1, 2007</td>
<td>YYMDDB4.</td>
<td>0701</td>
</tr>
<tr>
<td>YYMDDXw</td>
<td>8</td>
<td>2-10</td>
<td>January 1, 2007</td>
<td>YYMDD6.</td>
<td>070101</td>
</tr>
<tr>
<td>YYMDDXw</td>
<td>8</td>
<td>2-10</td>
<td>January 1, 2007</td>
<td>YYMDDN7.</td>
<td>070101</td>
</tr>
<tr>
<td>YYMDDXw</td>
<td>8</td>
<td>2-10</td>
<td>January 1, 2007</td>
<td>YYMDDP8.</td>
<td>07:01:01</td>
</tr>
<tr>
<td>YYMDDXw</td>
<td>8</td>
<td>2-10</td>
<td>January 1, 2007</td>
<td>YYMDDS9.</td>
<td>07/01/01</td>
</tr>
<tr>
<td>YYMDDXw</td>
<td>8</td>
<td>2-10</td>
<td>January 1, 2007</td>
<td>YYMDDB10.</td>
<td>2007 01 01</td>
</tr>
<tr>
<td>YYMONw</td>
<td>7</td>
<td>5-32</td>
<td>January 1, 2007</td>
<td>YYMON5.</td>
<td>07JAN</td>
</tr>
<tr>
<td>YYMONw</td>
<td>7</td>
<td>5-32</td>
<td>January 1, 2007</td>
<td>YYMON7.</td>
<td>2007JAN</td>
</tr>
<tr>
<td>YYQw</td>
<td>6</td>
<td>4-32</td>
<td>January 1, 2007</td>
<td>YYQ4.</td>
<td>07Q1</td>
</tr>
<tr>
<td>YYQw</td>
<td>6</td>
<td>4-32</td>
<td>January 1, 2007</td>
<td>YYQ6.</td>
<td>2007Q1</td>
</tr>
<tr>
<td>YYQw</td>
<td>6</td>
<td>4-32</td>
<td>January 1, 2007</td>
<td>YYQN4.</td>
<td>071</td>
</tr>
<tr>
<td>YYQw</td>
<td>6</td>
<td>4-32</td>
<td>January 1, 2007</td>
<td>YYQ5.</td>
<td>20071</td>
</tr>
<tr>
<td>YYQw</td>
<td>6</td>
<td>4-32</td>
<td>January 1, 2007</td>
<td>YYQC5.</td>
<td>07:1</td>
</tr>
<tr>
<td>YYQw</td>
<td>6</td>
<td>4-32</td>
<td>January 1, 2007</td>
<td>YYQD6.</td>
<td>2007-1</td>
</tr>
<tr>
<td>YYQw</td>
<td>6</td>
<td>4-32</td>
<td>January 1, 2007</td>
<td>YYQ7.</td>
<td>2007.1</td>
</tr>
<tr>
<td>YYQw</td>
<td>6</td>
<td>4-32</td>
<td>January 1, 2007</td>
<td>YYQS8.</td>
<td>2007/11</td>
</tr>
<tr>
<td>YYQw</td>
<td>6</td>
<td>4-32</td>
<td>January 1, 2007</td>
<td>YYQ6.</td>
<td>2007QI</td>
</tr>
<tr>
<td>YYQw</td>
<td>6</td>
<td>4-32</td>
<td>January 1, 2007</td>
<td>YYQR.</td>
<td>2007QI</td>
</tr>
<tr>
<td>YYQw</td>
<td>6</td>
<td>4-32</td>
<td>January 1, 2007</td>
<td>YYQR7.</td>
<td>2007I</td>
</tr>
<tr>
<td>YYQw</td>
<td>6</td>
<td>4-32</td>
<td>January 1, 2007</td>
<td>YYQR6.</td>
<td>2007I</td>
</tr>
<tr>
<td>YYQw</td>
<td>6</td>
<td>4-32</td>
<td>January 1, 2007</td>
<td>YYQRN6.</td>
<td>2007I</td>
</tr>
<tr>
<td>YYQw</td>
<td>6</td>
<td>4-32</td>
<td>January 1, 2007</td>
<td>YYQRD6.</td>
<td>2007-I</td>
</tr>
<tr>
<td>YYQw</td>
<td>6</td>
<td>4-32</td>
<td>January 1, 2007</td>
<td>YYQRN6.</td>
<td>2007I</td>
</tr>
<tr>
<td>YYQw</td>
<td>6</td>
<td>4-32</td>
<td>January 1, 2007</td>
<td>YYQRC7.</td>
<td>2007:1</td>
</tr>
<tr>
<td>YYQw</td>
<td>6</td>
<td>4-32</td>
<td>January 1, 2007</td>
<td>YYQRD.</td>
<td>2007-I</td>
</tr>
<tr>
<td>YYQw</td>
<td>6</td>
<td>4-32</td>
<td>January 1, 2007</td>
<td>YYQRS8.</td>
<td>2007/I</td>
</tr>
</tbody>
</table>
sasdate=input(mysas,yymmdd8.);  sasdate is 17531.

- ? suppress print of invalid data messages, ?? also prevent _ERROR_ being set to 1

- **INTCK(interval,from,to)** Can be used with two date, two time or two datetime values.
  - Provides an integer count between the two values.
  - There are lots of choices for this function including days weeks, weekdays, semimonth, month, qtr, semiyear, year and hours as well as multiplier and shift index options. Since it is so extensive it will not be covered here. It is suggested that you look it up on the web.

- **INTNX(interval<multiple><.Shift-index>,start-from,increment<,alignment>)** Can be used with date, time or datetime values. It increments these values and returns a date, time or datetime value.
  - Provides the ability to increment and decrement. It handles the differences between month values and leap years.
  - There are lots of choices for this function including days weeks, weekdays, semimonth, month, qtr, semiyear, year and hours as well as multiplier and shift index options. Since it is so extensive it will not be covered here. It is suggested that you look it up on the web.

- **JULDATE(sasdate)**. Converts a SAS Date to a Julian date value.
  - If the date falls within the 100 year window the result will be a 5 digit year otherwise it will be 7 digits.
  - However since it is a numeric variable, the digits could be as little as 1 as my examples show:
    - January 1, 2000 ends up as 1, the Jan 2nd would be 2 and so forth.
    - January 1, 2001 the Julian date would be 1001
    - January 1, 2010 the Julian date is 10001
  - If leading zeroes are needed use the z format to convert to a character variable.
  - Code is in function JULDATE.sas

- **JULDATE7(sasdate)**. Converts a SAS Date to a 7-digit Julian date value.
  - Examples:
    - January 1, 2000 ends up as 2000001.
    - January 1, 2001 the Julian date would be 2001001
    - January 1, 2010 the Julian date is 2010001
  - Code is in function JULDATE7.sas

- **MDY(month,day,year)**. Creates a SAS Date from numeric month, day, year variables.
  - If only a two-digit year is used, the century comes from the YEARCUTOFF value. coto a 7-digit Julian date value.
  - Example: Month=12, Day=31, Year=2006, SASDate=(Month,day,year) becomes SASDate=17167
  - Code is in function MDY.sas

- **MONTH(SASDate)** DATE function to select the month from a SAS Date.
  - The resulting day is a numeric value with no leading zero.
  - Example: Date value=20070101, SAS Date value=17167
  - SAS Code to get the month ‘Month=Month(SASDate);’ DayEq value=1
  - Code is in function MONTH.sas code for more examples.


- **PUT(Source,informat.)** Provides the ability to transform the value of a SAS variable into a value specified by the informat.
  - Example: There is a SAS numeric variable (mysas=20071231) that you want to make a SAS date. Will need to use PUT to make it character
    - Mysaschar=put(mysas,z8.);  SWITCH FROM TO NUMERIC TO CHARACTER
    - sasdate=input(mysaschar,yymmdd8.);  sasdate is 17531.

- **QTR(SASDate)** DATE function to select the numeric quarter from the date, will be 1, 2, 3 or 4.
  - Example: Date value=20070101, SAS Date value=17167, SAS Code to get the quarter ‘Qtrr = QTR(SASDate);’ Qtr value=1

- **TODAY()** - See DAY() function
WEEK(SASDate,descriptor) DATE Function. SASDate and descriptor are optional, will default to the week of the current date.
  o If a descriptor is used, must have a SASDate.
    Examples: myweek=week(); myweekU=week(SASDATE,'U'); myweekV=week(SASDATE,'V'); myweekW=week(SASDATE,'W'); See Week.sas for more details or http://support.sas.com/91doc/getDoc/nlsref.hlp/a002626758.htm

WEEKDAY(sasDate) DATE Function returns the day of the week Sunday=1 to 7=Saturday.
  o Example Myday = WEEKDAY(SASDate);

YEAR(sasDate) DATE Function returns the a four-digit numeric year.
  o Example MyYear = YEAR(SASDate);

YRDIF(StartSASDate,EndSASDate,basis) DATE Function returns the number of years different as integer and decimal part.
  o Basis is ‘ACT/ACT’, ‘30/360’, or a combination of these. See YRDIF function on web for details.

YYQ(year,quarter) Date Function. Year can be two (uses yearcutoff) or four digits. Must be valid year and quarter values. Will return a SAS date corresponding to the first day of the quarter.

MATH USING SAS DATES
Since a SAS Date is a number, you can manipulate it using math. For example, suppose you need to create another SAS date for next week using this week’s date, the date needed is a character variable that will need to be converted first. So here it goes:

adate='JAN2006'; /* Creating a character variable containing JAN2006 */
WkDate = input(adate,monyy7.); /* Convert to a SAS Date. It's value is 16802*/
wkdate2 = wkdate+7; /* Create a second variable for next week the value is 16809 */
Fdate1 = put(Wkdate,worddate12.); /*Use a format to create a variable containing ‘Jan 1, 2006’ */
Fdate2 = put(Wkdate2,worddate12.); /*Use a format to create a var containing ‘Jan 8, 2006’ */

Conclusions
As can be seen in this paper SAS has a number of ways to help with dates, whether reading them into or out of SAS or converting SAS variable values either into SAS date variables or from them. Of course there may be times when you will need to create your own and the PUT, INPUT and SUBSTR functions make it very easy to do. See the attached file for details on DATETIME and TIME functions as well as a number of self-contained SAS programs covering a number of the INFORMATS, FORAMTS and FUNCTIONS.

References & Acknowledgements
• As usual, the staff of SAS Institute, particularly Charley Mullin, was informative, helpful and much appreciated.
• SAS is a registered trademark of SAS Institute Inc. in the USA and other countries. ® Indicates USA registration. Other brand and product names are registered trademarks or trademarks of their respective companies.

Contact Information
Steve Rhoades
IMS HEALTH Inc.
660 W. Germantown Pike
Plymouth Meeting, Pa. 19462
Work Phone (610) 834-4614
Email: SRrhoades@US.IMSHEALTH.COM