stsFormattedDate

Usage

```
stsFormattedDate([date[,formatCode[,useBrackets]]])
put stsFormattedDate(the short date && long time,"MM/DD/YY HHHH:NNPPP")
    >> 03/29/06 22:54 pm
```

Description

Formats any incoming date (***date***) in a variety of formats based on a format string (***formatCode***) that is passed to the function. If a date is not supplied, today's date is assumed; if a format string is not supplied, "MM/DD/YYYY' is assumed. **stsFormattedDate** uses the following format codes:

Format Code	Description
M	month number, no leading zeroes (1-12)
MM	month number, with leading zeroes (01-12)
MMM	month name, abbreviated (Jan - Dec)
MMMM	month name, long (January - December)
D	day number, no leading zeroes (1-31)
DD	day number, with leading zeroes (01-31)
W	weekday name, single letter (S/M/T/W/t/F/s)
WW	weekday name, shortest (Su/M/Tu/W/Th/F/Sa)
WWW	weekday name, abbreviated (Sun/Mon/Tue/Wed/Thu/Fri/Sat)
WWWW	weekday name, long (Sunday/Monday/Tuesday/Wednesday/Thursday/Friday/Saturday)
Y	two-digit year (00-99)
YY	four digit year (2006)
H	hours, no leading zeroes, 12 hour format (1-12)
HH	hours, leading zeroes, 12 hour format (01-12)
HHH	hours, no leading zeroes, 24 hour format (1-23)
HHHH	hours, leading zeroes, 24 hour format (01-23)
HHHHH	military time format (0000 - 2359)
P	AM/PM, single character, lower case (a/p)
PP	AM/PM, single character, upper case (A/P)
PPP	AM/PM, two characters, lower case (am/pm)
PPPP	AM/PM, two characters, upper case (AM/PM)
N	minutes, no leading zeroes (0-59)
NN	minutes, leading zeroes (00-59)
S	seconds, no leading zeroes (0-59)
SS	seconds, leading zeroes (00-59)
G	Calculation based on GMT (-1100 to +1100)

Any other characters in ***formatCode*** are kept and implemented to return a formatted date string.

If for some reason you want to use M, D, W, Y, H, P, N, S or G in your date string but not have them formatted (for example if you want to put "GMT" as a string after the actual date returned), enclose all of the patterns in square brackets (like "[MMMM]") and pass "true" for ***useBrackets***.

Here are some examples for formats:

```
MM/DD/YYYY -> 04/07/2004
WWW, MMM D YYYY -> Mon, Aug 2 2004
WWW, MMM D YYYY G --> Mon, Aug 2 2004 -0500
[WWW],[MMM] [D] [YYYY] [G] GMT --> Mon, Aug 2 2004 -0500 GMT
```

Code

Select the code below and copy it to the clipboard, or retrieve it from Scripter's Scrapbook Online.

```
function stsFormattedDate pDate,pFormat,pUseBrackets
  if (pDate = "") or (pDate = "Now") then put the date && the long time into pDate
  if (pFormat = "") then put "MM/DD/YYYY" into pFormat
  if isNumber(word -1 of pDate) then
    if (word -1 of pDate <=2359) and (length(word -1 of pDate)=4) then
      --date and military time sent in, just needs a colon between hour and minute for it to be converted
      put ":" before char -2 of pDate
    else
      -- simple number, coerce to AM
      if word -1 of pDate <= 12 then
        put ((word -1 of pDate) & ":00 AM") into word -1 of pDate
      else
        -- a number larger than 12 but not military? Can't do anything with that
        return "invalid time"
      end if
    end if
  end if
  -- Check for am/pm without preceding space
  put offset("a",pDate) into tLoc
```

```
  if (tLoc <> 0) and (char (tLoc-1) of pDate <> " ") then
    if char (tLoc+1) of pDate <> "m" then put "m" after char tLoc of pDate
    put " " before char tLoc of pDate
    put word -2 of pDate into tTime
    if ":"is not in tTime then
      -- probably something like "2am"
      put (tTime & ":00") into word -2 of pDate
    end if
  end if
  put offset("p",pDate) into tLoc
  if (tLoc <> 0) and (char (tLoc-1) of pDate <> " ") then
    if char (tLoc+1) of pDate <> "m" then put "m" after char tLoc of pDate
    put " " before char tLoc of pDate
    put word -2 of pDate into tTime
    if ":"is not in tTime then
      -- probably something like "2pm"
      put (tTime & ":00") into word -2 of pDate
    end if
  end if

  put ((pUseBrackets <> "") and (pUseBrackets <> "false")) into pUseBrackets
  put pDate into tOrigDate

  -- Check to see if it's mySQL-formatted
  if matchText(word 1 of pDate,"(?s)(.*)-(.*)-(.*)",tY,tM,tD) and length(tY)=4 then
    if word 2 of pDate <> "" then
      put tM & "/" & tD & "/" & tY && (word 2 of pDate) into pDate
    else
      put tM & "/" & tD & "/" & tY into pDate
    end if
    convert pDate to dateItems
    if (pDate is "invalid date") or (("69" is not in tOrigDate) and (item 1 of pDate="1969")) then
      return "invalid date"
    end if
  else
    convert pDate to dateItems
    if (pDate is "invalid date") or (("69" is not in tOrigDate) and (item 1 of pDate="1969")) then
      return "invalid date"
    end if
  end if
  put item 1 of pDate into tYear
  put item 2 of pDate into tMonthNum
  put item 3 of pDate into tDayNum
  put item 4 of pDate into tHour
  put item 5 of pDate into tMinute
  put item 6 of pDate into tSecond
  put item 7 of pDate into tWeekdayNum
  put word -1 of the internet date into G
  set the numberFormat to "00"

  if pUseBrackets then
    put "[MMMM],[MMM],[MM],[M],[WWWW],[WWW],[WW],[W],[DD],[D],[YYYY],[YY],[HHHHH],[HHHH],[HHH],[HH],[H],[PPPP],[PPP],[PP],[P],[NN],[N],[SS],[S],[G]" into tFormatWords
  else
    put "MMMM,MMM,MM,M,WWWW,WWW,WW,W,DD,D,YYYY,YY,HHHHH,HHHH,HHH,HH,H,PPPP,PPP,PP,P,NN,N,SS,S,G" into tFormatWords
  end if
  put "!@#$%^&*()_+{}|:'<>?~`-=[]" into tReplaceChars
  repeat with x = 1 to the number of items of tFormatWords
    replace (item x of tFormatWords) with "[[[" & char x of tReplaceChars & "]]]]" in pFormat
  end repeat

  put tYear into YYYY
  put char -2 to -1 of tYear into YY

  put tMonthNum into M
  put (tMonthNum+0) into MM
  put line tMonthNum of the abbreviated monthNames into MMM
  put line tMonthNum of the long monthNames into MMMM

  put tDayNum into D
  put (tDayNum+0) into DD

  put char tWeekDayNum of "SMTWtFs" into W
  put item tWeekDayNum of "S,M,Tu,W,Th,F,Sa" into WW
  put line tWeekDayNum of the abbreviated weekdayNames into WWW
  put line tWeekDayNum of the long weekdayNames into WWWW

  put tHour into HHH
  put (tHour+0) into HHHH
  if tHour < 12 then
    if tHour = 0 then put 12 into H
    else put tHour into H
    put "a" into P
    put "A" into PP
    put "am" into PPP
    put "AM" into PPPP
  else
    put tHour-12 into H
    if H = 0 then put 12 into H
    if H < 10 then
      delete char 1 of H  -- remove leading 0
    end if
    put "p" into P
    put "P" into PP
    put "pm" into PPP
    put "PM" into PPPP
  end if
  put (H+0) into HH

  put tMinute into N
  put (tMinute+0) into NN

  put HHHH & NN into HHHHH
```

```
    put tSecond into S
    put (tSecond+0) into SS

    repeat with x = 1 to the number of items of tFormatWords
      if pUseBrackets then
        local tTemp
        get matchText(item x of tFormatWords,"\[(.*?)\]",tTemp)
        do "put" && tTemp && "into tVal"
      else
        do "put" && (item x of tFormatWords) && "into tVal"
      end if
      replace "[[[" & char x of tReplaceChars & "]]]]"  with tVal in pFormat
    end repeat

    return pFormat
  end stsFormattedDate
```

Change History

Date	Description
4/26/07	• Added support for simple time values like "2a", "2p", "2am", "2pm" translating properly (ex. "2a" = "2:00 AM"), and simple numbers like "4" becoming "4:00 AM".
2/10/07	• Fixed bug where 12:00 times were coming back as 0's, even if military time was not requested.
9/30/06	• Initial posting

Print this code entry

News and Rumors Products Services Developer Resources Contact STS About STS