

## NAG Library Routine Document

### F01VLF (DTFTTP)

**Note:** before using this routine, please read the Users' Note for your implementation to check the interpretation of *bold italicised* terms and other implementation-dependent details.

#### 1 Purpose

F01VLF (DTFTTP) copies a real triangular matrix stored in a Rectangular Full Packed (RFP) format to packed format. The RFP storage format is described in Section 3.3.3 in the F07 Chapter Introduction and the packed storage format is described in Section 3.3.2 in the F07 Chapter Introduction.

#### 2 Specification

```
SUBROUTINE F01VLF (TRANSR, UPLO, N, ARF, AP, INFO)
  INTEGER          N, INFO
  REAL (KIND=nag_wp) ARF(N*(N+1)/2), AP(N*(N+1)/2)
  CHARACTER(1)    TRANSR, UPLO
```

The routine may be called by its LAPACK name *dtfttp*.

#### 3 Description

F01VLF (DTFTTP) packs a real  $n$  by  $n$  triangular matrix  $A$  stored in RFP format to packed format. This routine is intended for possible use in conjunction with routines from Chapters F06 and F07 where some routines that use triangular matrices store them in RFP format.

#### 4 References

None.

#### 5 Parameters

- 1: TRANSR – CHARACTER(1) *Input*  
*On entry:* specifies whether the RFP representation of  $A$  is normal or transposed.  
 TRANSR = 'N'  
 The matrix  $A$  is stored in normal RFP format.  
 TRANSR = 'T'  
 The matrix  $A$  is stored in transposed RFP format.  
*Constraint:* TRANSR = 'N' or 'T'.
- 2: UPLO – CHARACTER(1) *Input*  
*On entry:* specifies whether  $A$  is upper or lower triangular.  
 UPLO = 'U'  
 $A$  is upper triangular.  
 UPLO = 'L'  
 $A$  is lower triangular.  
*Constraint:* UPLO = 'U' or 'L'.

- 3: N – INTEGER *Input*  
*On entry:*  $n$ , the order of the matrix  $A$ .  
*Constraint:*  $N \geq 0$ .
- 4: ARF( $N \times (N + 1)/2$ ) – REAL (KIND=nag\_wp) array *Input*  
*On entry:* the triangular matrix  $A$  in RFP format, as described in Section 3.3.3 in the F07 Chapter Introduction.
- 5: AP( $N \times (N + 1)/2$ ) – REAL (KIND=nag\_wp) array *Output*  
*On exit:* the  $n$  by  $n$  triangular matrix  $A$ , packed by columns.  
 More precisely,  
   if UPLO = 'U', the upper triangle of  $A$  is stored with element  $A_{ij}$  in AP( $i + j(j - 1)/2$ ) for  $i \leq j$ ;  
   if UPLO = 'L', the lower triangle of  $A$  is stored with element  $A_{ij}$  in AP( $i + (2n - j)(j - 1)/2$ ) for  $i \geq j$ .
- 6: INFO – INTEGER *Output*  
*On exit:* INFO = 0 unless the routine detects an error (see Section 6).

## 6 Error Indicators and Warnings

Errors or warnings detected by the routine:

INFO < 0

If INFO =  $-i$ , argument  $i$  had an illegal value. An explanatory message is output, and execution of the program is terminated.

## 7 Accuracy

Not applicable.

## 8 Further Comments

None.

## 9 Example

This example reads in a triangular matrix in RFP format and copies it to packed format.

### 9.1 Program Text

```

Program f01vlfe
!      F01VLF Example Program Text
!      Mark 24 Release. NAG Copyright 2012.
!
!      .. Use Statements ..
      Use nag_library, Only: dtfttp, nag_wp, x04cbf
!      .. Implicit None Statement ..
      Implicit None
!      .. Parameters ..
      Integer, Parameter          :: incl = 1, indent = 0, ncols = 80,    &
                                   nin = 5, nout = 6
      Character (1), Parameter    :: diag = 'N', intlabel = 'I', matrix = &
                                   'G', nolabel = 'N'

```

```

      Character (4), Parameter      :: form = 'F5.2'
!   .. Local Scalars ..
      Integer                       :: ifail, info, lenap, lenarf, n
      Character (21)                :: title
      Character (1)                 :: transr, uplo
!   .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable :: ap(:), arf(:)
      Character (1)                 :: clabs(1), rlabs(1)
!   .. Executable Statements ..
      Write (nout,*) 'F01VLF Example Program Results'
!   Skip heading in data file
      Read (nin,*)
      Write (nout,*)
      Flush (nout)
      Read (nin,*) n, uplo, transr
      lenap = (n*(n+1))/2
      lenarf = lenap

      Allocate (ap(lenap),arf(lenarf))

!   Read an RFP matrix into array ARF
      Read (nin,*) arf(1:lenarf)

!   Print the Rectangular Full Packed array
      title = 'RFP Packed Array ARF:'
      ifail = 0
      Call x04cbf(matrix,diag,lenarf,incl,arf,lenarf,form,title,intlabel, &
        rlabs,nolabel,clabs,ncols,indent,ifail)

      Write (nout,*)
      Flush (nout)

!   Convert to packed vector form
      info = 0
!   The NAG name equivalent of dtfttp is f01vlf
      Call dtfttp(transr,uplo,n,arf,ap,info)

!   Print the packed vector
      title = 'Packed Array AP:      '
      ifail = 0
      Call x04cbf(matrix,diag,lenap,incl,ap,lenap,form,title,intlabel,rlabs, &
        nolabel,clabs,ncols,indent,ifail)

      End Program f01vlfe

```

## 9.2 Program Data

```

F01VLF Example Program Data
  4 'U' 'N'                : n, uplo, transr
  1.30
  2.30
  3.30
  1.10
  1.20
  1.40
  2.40
  3.40
  4.40
  2.20                    : RFP Matrix ARF

```

## 9.3 Program Results

F01VLF Example Program Results

```

RFP Packed Array ARF:
  1  1.30
  2  2.30
  3  3.30
  4  1.10
  5  1.20

```

6 1.40  
7 2.40  
8 3.40  
9 4.40  
10 2.20

Packed Array AP:

1 1.10  
2 1.20  
3 2.20  
4 1.30  
5 2.30  
6 3.30  
7 1.40  
8 2.40  
9 3.40  
10 4.40

---