

Use of SD_RESULTAT

Summary:

SD_RESULTAT [D4.06.08]: is accessible only through the routines described in this document.

The routine RSCRSD must be supplemented to create new types of SD_RESULTAT.

Contents

1 List of the Routines.....	3
2 Routines of creation, destruction, enlarging.....	5
3 Routines of access starting from the keywords of the order Aster.....	5
4 Routines of access to the structures SD_RESULTAT.....	7
5 Routine interpolation.....	10
6 Routines of existence.....	10
7 Example.....	13

1 List of the Routines

Routines of Creation, Destruction, Enlarging	
RSCRSD	To create a structure SD_RESULTAT
RSDLSD	To destroy a structure SD_RESULTAT
RSRUSD	To destroy all the fields with the sequence number IORDR of a structure SD_RESULTAT
RSAGSD	To increase a structure SD_RESULTAT

Routines of access starting from the keywords of the order Aster	
RSUTNU	Recovery starting from the keywords of the order, the sequence numbers in a structure SD_RESULTAT
RSTRAN	Recovery from the keywords of the order for a structure tran_gene Result

Routines of access to the structures SD_RESULTAT	
RSORAC	Recovery starting from a variable of access, sequence numbers of a structure SD_RESULTAT
RSEXCH	Recovery from a reference symbol and of a sequence number, name of a field in a structure SD_RESULTAT.
RSNOCH	To note starting from a reference symbol and of a sequence number, the name of a field in a structure SD_RESULTAT.
RSADPA	Recovery starting from and a list sequence number of name (S) of parameter (S) or variable (S) of access, addresses JEVEUX of a structure SD_RESULTAT
RSNOPA	Recovery of the names and amongst parameters in a structure SD_RESULTAT as well as the names and the number of variables of access. These names are stored in a vector JEVEUX of name nomjv.
RSUTNC	Recovery starting from a reference symbol, noted fields and their sequence numbers associated in a structure SD_RESULTAT .
RSUTOR	Recovery starting from a field, sequence number and possibly the reference symbol in a structure SD_RESULTAT.

Routine of interpolation	
RSINCH	Interpolation for a reference symbol and a variable of access, a field in a structure SD_RESULTAT

Routines of existence	
RSEXIS	Checking of the existence of a structure SD_RESULTAT.
RSEXPA	Checking of the existence of a name of parameter or a name of variable of access in a structure SD_RESULTAT
RSVPAR	Checking of the existence of a name of parameter and its value for a sequence number given in a structure SD_RESULTAT

2 Routines of creation, destruction, enlarging

CAL RSCRSD (nomsd, typesd, nbordr)

IN	nomsd	K8	name of the structure of data RESULT
IN	typesd	K16	type of the structure of data RESULT
IN	nbordr	I	number of sequence number

To create a structure SD_RESULTAT (if this structure already exists, it is destroyed). The modification of this routine makes it possible to create new types of SD_RESULTAT or to enrich those existing.

CAL RSDLSD (nomsd)

IN	nomsd	K8	name of the structure of data RESULT
----	-------	----	--------------------------------------

To destroy a structure SD_RESULTAT. This routine does not destroy the fields indexed by SD_RESULTAT.

CAL RSRUSD (nomsd, iordr)

IN	nomsd	K8	name of the structure of data RESULT
IN	iordr	I	sequence number

To destroy in the structure SD_RESULTAT fields starting from a given sequence number.

CAL RSAGSD (nomsd, nbordr)

IN	nomsd	K8	name of the structure of data RESULT
IN	nbordr	I	new size of RESULT (the size is doubled if NBORDR = 0)

To increase a structure SD_RESULTAT.

3 Routines of access starting from the keywords of the order Aster

The access via the process control language to the data indexed in SD_RESULTAT by the same keywords whatever the order is possible (IMPR_RESU, CALC_G, etc...).

The description of these keywords (TOUT_ORDRE, NUME_ORDRE, etc...) is for example in [U4.71.01].

For certain orders (IMPR_RESU, for example) these keywords are under a keyword factor (RESU) for others not (CALC_G). In this last case the argument keyword specified in the routines below must be “.

CAL RSUTNU (nomsd, keyword, iocc, knum, nbordr, prec, crit, iret)

IN	nomsd	K8	name of the structure of data RESULT
IN	keyword d	K*	keyword ratio control
IN	iocc	I	number of occurrence of the keyword factor
IN	knum	K19	name of the vector JEVEUX ZI to write the list of the numbers
OUT	nbordr	I	many found sequence numbers
IN	prec	R8	precision requested
IN	crit	K8	search criterion
OUT	iret	I	code return of error, = 0: ok

Recovery starting from the keywords of the order, the sequence numbers in a structure SD_RESULTAT.

Note:

The routine RSUTNU above the list of the possible sequence numbers recovers of SD_RESULTAT, possibly filtered by the keywords of the process control language (TOUT= ' OUI ', LIST_INST=...). It is not sure for as much, that for all these sequence numbers, all fields (NOM_CHAM) or all the parameters really exist. When one wants to recover the list of the sequence numbers for which a field (or a parameter) really exists, the routines should be used RSUTN1 or RSUTN2 .

CAL RSTRAN (interp, nomsd, keyword, iocc, kinst, krang, nbinst, iret)

IN	interp	K4	type of interpolation wished
IN	nomsd	K8	name of the structure tran_gene
IN	keyword	K16	keyword ratio control
IN	iocc	I	number of occurrence of the keyword factor
IN	kinst	K19	name of the vector JEVEUX to write the list of the moments
IN	krang	K19	name of the vector JEVEUX to write the list of the numbers
OUT	nbinst	I	many moments read
OUT	iret	I	code return, = 0: ok

Recovery starting from the keywords of the order for a structure tran_gene Result, according to the type of interpolation:

- moments (INTERP = ' FLAX ', ' LOG '), user datum,
- moments and their associated sequence numbers (INTERP = ' NOT '), moments of calculation of tran_gene .

4 Routines of access to the structures SD_RESULTAT

CAL RSORAC (nomsd, access, ival, rval, kval, cval, prec, crit, nutrou, ndim, nbtrou)

IN	nomsd	K8	name of the structure of data RESULT
IN	access	K16	name of the variable of access.
IN	ival	I	value of the so whole variable of access
IN	rval	R8	value of the so real variable of access
IN	kval	K.	value of the variable of access if character
IN	cval	C16	value of the so complex variable of access
IN	prec	R8	precision requested
IN	crit	K8	criterion requested
IN	ndim	I	dimension of the list nutrou.
OUT	nutrou	I	list of the found sequence numbers.
OUT	nbtrou	I	many found sequence numbers. if the found number nbtrou is higher than ndim, then one returns nbtrou = - nbtrou

Recovery starting from a variable of access, sequence numbers of a structure SD_RESULTAT.

Convention:

- if access = 'LONUTI' : recovery working length of .ORDR
- if access = 'LONMAX' : recovery maximum length of .ORDR
- if access = 'LAST' : recovery of the last sequence number
- if access = 'FIRST' : recovery of the first sequence number
- if access = 'TOUT_ORDRE' : recovery of all the sequence numbers.

CAL RSEXCH (kstop, nomsd, nomsy, iordr, chextr, iret)

IN	kstop	K1	'F' / '' If 'F' and iret > 0 => fatal error for the user.
IN	nomsd	K8	name of the structure of give RESULT
IN	nomsy	K16	reference symbol
IN	iordr	I	sequence number
OUT	chextr	K19	name of the extracted field
OUT	iret	I	code return = 0 : the field is "possible" and there exists. =100: the field is "possible" but does not exist = 101: the reference symbol is prohibited. = 110: The sd_resultat is "full". It should be aggrandir. = 102: the sequence number IORDR is not allowed

Recovery starting from a reference symbol and of a sequence number, name of a field in a structure SD_RESULTAT.

This routine manages the fields in the structure SD_RESULTAT and fact call has the routine RSUTCH.

CAL RSNOCH (nomsd, nomsy, iordr)

IN	nomsd	K8	name of the structure of data RESULT
IN	nomsy	K16	reference symbol
IN	iordr	I	sequence number

To note (=stocker) starting from a reference symbol and of a sequence number, the name of a field in a structure SD_RESULTAT.

CAL RSADPA (nomsd, concealment, will npara, lpara, iordr, itype, ljeveu, ctype)

IN	nomsd	K8	name of the structure of data RESULT
IN	concealment	K1	access term with the parameters: 'L' : reading, 'E' : writing.
IN	will npara	I	many sought parameters
IN	will lpara	K16	list of the names of the parameters
IN	iordr	I	sequence number
IN	itype	I	code indicating that one wishes the type = 0 step of the type ? 0 one provide the type
OUT	ljeveu	I	list of the addresses JEVEUX in ZI, ZR,...
OUT	ctype	K4	list of the types of the addresses JEVEUX I entirety, R reality, C complex, 'K8' 'K16' 'K24' 'K32' 'K80' character.

Recovery starting from and a list sequence number of name (S) of parameter (S) or variable (S) of access, addresses JEVEUX of a structure SD_RESULTAT.

CAL RSNOPA (nomsd, icode, nomjv, nbacc, will nbpara)

IN	nomsd	K8	name of the structure of data RESULT
IN	nomjv	K19	name of the vector JEVEUX to write the list of the names
IN	icode	I	= 0, variables of access alone = 1, parameters alone = 2, variables of access and parameters
OUT	nbacc	I	many variables of access
OUT	will nbpara	I	many parameters

Recovery of the names and amongst parameters in a structure SD_RESULTAT as well as the names and the number of variables of access. These names are stored in a vector JEVEUX of name nomjv.

CAL RSUTNC (nomsd, nomsy, ndim, nomch, nuordr, nbtrou)

IN	nomsd	K8	name of the structure of data RESULT
IN	nomsy	K16	reference symbol
IN	ndim	I	dimension of the tables
OUT	nomch	K16	table of the field names
OUT	nuordr	I	table of the sequence numbers of the found fields
OUT	nbtrou	I	many found fields if the found number nbtrou is higher than ndim, then one returns nbtrou = - nbtrou

Recovery starting from a reference symbol, noted fields and their sequence numbers associated in a structure SD_RESULTAT.

CAL RSUTOR (nomsd, chextr, nomsy, nuordr, iret)

IN	nomsd	K8	name of the structure of data RESULT
IN	chextr	K19	name of the field to be searched
OUT	nomsy	K16	reference symbol
OUT	nuordr	I	sequence number
OUT	iret	I	code return: =1, the field was found, =0 if not

Recovery starting from a field, sequence number and possibly the reference symbol in a structure SD_RESULTAT.

5 Routine interpolation

CAL RSINCH (nomsd, nomsy, access, rval, chextr, proldr, prolga, istop, base, iret)

nomsd	IN	K8	name of the structure of data RESULT
nomsy	IN	K16	reference symbol
access	IN	K16	name of the variable of access
rval	IN	R8	actual value of the variable of access
chextr	IN	K19	name of the field has to create. (if there exists, it is destroyed)
proldr	IN	K8	type of prolongation wanted on the right: 'CONSTANT ', 'LINEAR ', 'EXCLUDED'
prolga	IN	K8	type of prolongation wanted on the left: 'CONSTANT ', 'LINEAR ', 'EXCLUDED '
istop	IN	I	in the event of interpolation error: = 0, do not write a message, do not make stop. = 1, written message, does not make stop. = 2, written message, made stop.
base	IN	K8	base field created
iret	OUT	I	code return: 1) the field is calculated: = 0, the field are interpolated between 2 values. = 1, the field is prolonged on the left. = 2, the field is prolonged on the right. 2) the field is not calculated: = 10, there does not exist any field for the interpolation. = 11, prolongation on the left prohibited. = 12, prolongation on the right prohibited. = 20, the variable of access is illicit.

Interpolation for a reference symbol and a variable of access, a field in a structure SD_RESULTAT

6 Routines of existence

CAL RSEXIS (nomsd, iret)

IN	nomsd	K8	name of the structure of data RESULT
OUT	iret	I	code return: = 0, the structure exist; ? 0, the structure do not exist.

Checking of the existence of a structure SD_RESULTAT.

Code_Aster

Version
default

Titre : Utilisation des SD_RESULTAT
Responsable : PELLET Jacques

Date : 01/08/2012 Page : 11/13
Clé : D6.05.01 Révision :
6bdf6c02a2

CAL RSEXPA (nomsd, icode, nompar, iret)

IN	nomsd	K8	name of the structure of data RESULT
IN	icode	I	= 0, variable of access = 1, parameter = 2, variable of access or parameter
IN	nompar	K16	name of the parameter or the variable of access
OUT	iret	I	code return: = 0, the name does not exist; = 100, the name exists.

Checking of the existence of a name of parameter or a name of variable of access in a structure SD_RESULTAT.

CAL RSVPAR (nomsd, iordr, nompar, ipar, rpar, kpar, iret)

IN	nomsd	K8	name of the structure of data RESULT
IN	iordr	I	sequence number
IN	nompar	K16	name of the parameter to be checked
IN	ipar	I	so whole value of the parameter
IN	rpar	R8	so real value of the parameter
IN	kpar	K.	value of the parameter if character
OUT	iret	I	code return: = 0, it is not a parameter = 100, the value of the parameter is correct = 110, the value of the parameter is not correct

Checking of the existence of a name of parameter and its value for a sequence number given in a structure SD_RESULTAT.

7 Example

Let us take the example of the storage of the modes of mechanical vibration (`mode_meca`), one calculated 5 modes and one stores them in a structure `SD_RESULTAT`.

Stage 1: creation of the structure `SD_RESULTAT`

```
NBMODE = 5  
CAL RSCRSO (MODES, 'MODE_MECA', NBMODE)
```

Stage 2: storage of the clean vectors

```
C 10 IM = 1, NBMODE  
...  
CAL RSEXCH ('', MODES, 'DEPL', IM, CHAMNO, 1ST)  
...  
CAL JEVEUO (CHAMNO//'.VALE', 'E', LVALE)  
C 12 IEQ = 1, NEQ  
    ZR (LVALE+IEQ-1) =...  
12 CONTINUOUS  
    CAL RSNOCH (MODES, 'DEPL', IM)  
...  
...
```

Stage 3: storage of the variables of access and modal parameters

```
CAL RSADPA (MODES, 'E', 1, 'FREQ', IM, 0, LFREQ, K8B)  
ZR (LFREQ) =...  
CAL RSADPA (MODES, 'E', 1, 'NUMÉRIQUE_MODE', IM, 0, LNUME, K8B)  
ZI (LNUME) =...  
CAL RSADPA (MODES, 'E', 1, 'STANDARD', IM, 0, LNORM, K8B)  
ZK24 (LNORM) =...  
...  
...
```