

COMMANDE D'APPEL POUR INCREMENTS DE BOUCLE ET COULEURS DS SOURCE RPGLE

CMD : INBRPGLE

```
100 /* COMMANDE D'APPEL POUR INCREMENTS DE BOUCLE ET COULEURS DS SOURCE RPGLE */
200             CMD             PROMPT(' Traitement des Sources RPGLE ')
300 /* Bibliothèque */
400             PARM            KWD(BIBL) TYPE(*CHAR) LEN(10) DFT(*LIBL) +
500                     PROMPT('Bibliothèque . . . . .')
600 /* Fichier */
700             PARM            KWD(FICH) TYPE(*CHAR) LEN(10) DFT(QRPGLESRC) +
800                     PROMPT('Fichier . . . . .')
900 /* Membre */
1000            PARM            KWD(MBR) TYPE(*CHAR) LEN(10) +
1100                     PROMPT('Source RPGLE . . . . .')
1200 /* Type de Membre */
1300            PARM            KWD(TYPE) TYPE(*CHAR) LEN(10) DFT(RPGLE) +
1400                     PROMPT('Type de source . . . . .')
1500 /* Texte du Membre */
1600            PARM            KWD(TEXTE) TYPE(*CHAR) LEN(50) +
1700                     PROMPT('Texte du source . . . . .')
```

PF : INPRPGILE

```
A*             Physical *SRC : INPRPGILE
A*             Incréments de boucles et couleurs ds source RPGLE
A*
A             R RPGILE             TEXT('Incréments de boucles et coul-
A                                     eurs ds source RPGLE')
A             SRCSEQ              6S 2
A             SRCDAT              6S 0
A             SRCDTA              100
```

CL : : INCRPGLE

```
100 /* * * * * * * * * * * * * * * * * * * * * * * * * * * * */
200 /* Nom du programme . . . . . : INCRPGLE */
400 /* Fichier source initial . . . . . : QCLPSRC */
500 /* */
600 /* Texte . . : INCREMENTS DE BOUCLE ET COULEURS DANS SOURCES RPG */
700 /* */
800 /* Option PDM : LE ==> INBRPGLE &L &F &N &S &X */
900 /* */
1000 /* * * * * * * * * * * * * * * * * * * * * * * * * * * * */
1100
1200             PGM             PARM(&BIBL &FICH &MBR &RPG &TEXTE)
1300
1400             DCL VAR(&BIBL) TYPE(*CHAR) LEN(10) /* Biblio */
1500             DCL VAR(&FICH) TYPE(*CHAR) LEN(10) /* Fichier */
1600             DCL VAR(&MBR) TYPE(*CHAR) LEN(10) /* Membre */
1700             DCL VAR(&RPG) TYPE(*CHAR) LEN(10) /* Type */
1800             DCL VAR(&TEXTE) TYPE(*CHAR) LEN(50) /* Texte */
1900             DCL VAR(&MSGF) TYPE(*CHAR) LEN(70)
2000             DCL VAR(&NBR) TYPE(*CHAR) LEN(4)
2100
2200 /* Le type de membre doit être RPGLE */
2300             IF COND(&RPG *NE 'RPGLE') THEN(GOTO +
2400                     CMDLBL(£ERREUR))
2500
2600 /* Le Membre doit exister */
2700             CHKOBJ OBJ(&BIBL/&FICH) OBJTYPE(*FILE) MBR(&MBR)
2800             MONMSG MSGID(CPF0000) EXEC(GOTO CMDLBL(£ERREUR))
2900
3000 /* Création du Source de Travail dans QTEMP */
3100             CHKOBJ OBJ(QTEMP/INPRPGILE) OBJTYPE(*FILE)
3200             MONMSG MSGID(CPF9801) EXEC(GOTO CMDLBL(£CREAT))
3300             GOTO CMDLBL(£CLEAR)
3400
3500 £CREAT:      CRTDUPOBJ OBJ(INPRPGILE) FROMLIB(GBTEST) +
3600                     OBJTYPE(*FILE) TOLIB(QTEMP)
3700 £CLEAR:      CLRPFM FILE(QTEMP/INPRPGILE)
```

```

3800
3900  /* Copy du Membre origine dans Travail de QTEMP */
4000          CPYSRCF      FROMFILE(&BIBL/&FICH) +
4100                      TOFILE(QTEMP/INPRPGILE) FROMMBR(&MBR) +
4200                      TOMBR(*FIRST)
4300
4400  /* Traitement en cours */
4500          CHGVAR        VAR(&MSGF) VALUE(&MBR *CAT &BIBL *CAT &TEXTE)
4600          SNDPGMMMSG    MSGID(SYS0001) MSGF(GBTEST/INERMSGF) +
4700                      MSGDTA(&MSGF) TOPGMQ(*EXT) +
4800                      TOMSGQ(*TOPGMQ) MSGTYPE(*STATUS)
4900
5000          OVRDBF        FILE(INPRPGILE) TOFILE(QTEMP/INPRPGILE)
5100          CALL          PGM(GBTEST/INRRPGLE) PARM(&NBR)
5200          DLTOVR        FILE(*ALL)
5300
5400  /* Remplacement du Membre origine */
5500          CPYSRCF        FROMFILE(QTEMP/INPRPGILE) TOFILE(&BIBL/&FICH) +
5600                      FROMMBR(*FIRST) TOMBR(&MBR)
5700
5800          CHGVAR        VAR(&MSGF) VALUE(&NBR *CAT &MBR *CAT &FICH +
5900                      *CAT &BIBL)
6000          SNDPGMMMSG    MSGID(SYS0004) MSGF(IKETEST/INERMSGF) +
6100                      MSGDTA(&MSGF) TOPGMQ(*PRV) +
6200                      TOMSGQ(*TOPGMQ) MSGTYPE(*COMP)
6300          GOTO          CMDLBL(£FIN)
6400
6500 £ERREUR:  /* Membre non RPGLE ou non Trouvé */
6600          CHGVAR        VAR(&MSGF) VALUE(&MBR *CAT &FICH *CAT &BIBL)
6700          SNDPGMMMSG    MSGID(SYS0002) MSGF(IKETEST/INERMSGF) +
6800                      MSGDTA(&MSGF) TOPGMQ(*PRV) +
6900                      TOMSGQ(*TOPGMQ) MSGTYPE(*COMP)
7000
7100 £FIN:      RCLRSC
7200          ENDPGM

```

RPGLE : INRRPGLE

FINPRPGILE	UP	E	DISK	
*-----				
D OP		S	5	DIM(100) Codes opérations
D SQ		S	4	DIM(100) N° Ligne Source
*-----				
D DATA		DS		
D LIG			1 4	
D TABLE			1 3	
D BIT			5 5	
D CARTE			6 6	
D COMM			7 7	
D COMMB			7 8	
D FICH			7 14	
D £FACT1			12 25	
D £DS			24 25	
D OPER			26 31	
D OPER4			26 29	
D OPER5			26 30	
D OPIF			26 27	
D OPI			26 28	
D OPCAS			26 28	
D OPEND			26 28	
D OPENDZ			29 30	
D OPELSE			26 29	
D £PTR			40 40	
D £TAB			44 46	
D £BASED			44 48	
D INHI			71 72	
D INLO			73 74	
D INEQ			75 76	
D WFIN			89 100	
D WFINDO			89 90	
D WFIND			90 90	

D	WFINOP	91	95	
D	WFINSQ	96	100	
D	INBLAN	95	99	
D	POS80	100	100	
*				
D	INDIC	DS		
D	HI	1	1	
D	LO	3	3	
D	EQ	5	5	
D	BL	6	6	
*-----				
C	*ENTRY	PLIST		
C		PARM	NBRA	4
*-----				
C	N10	EXSR	\$INIT	
*				
C		MOVE	SRCDTA	DATA
C		CLEAR		INDIC
C		MOVE	' '	BITD 1
C		MOVE	' '	INCR 4
D001 C	CPT	IFNE	000	
001 C		MOVE	CPT	INCR 4
F001 C		END		< IFNE 0062
C	TABLE	CABEQ	'** '	£FIN 99 Table compil. =
*				
* Commentaires				
D001 C	COMMB	IFEQ	'*/'	
001 C		GOTO	£FIN	
F001 C		END		< IFEQ 0068
D001 C	COMM	IFEQ	'*'	
001 C		MOVE	' '	INCR 4
001 C		MOVE	BIT22	BITD 1 * Blanc
001 C		GOTO	£LIGF	
F001 C		END		< IFEQ 0071
*				
* Test Carte				
D001 C		IF	CARTE <> 'H' AND CARTE <> 'F' AND CARTE <>	
001 C			'D' AND CARTE <> 'I' AND CARTE <> 'C' AND	
001 C			CARTE <> 'O'	
001 C		GOTO	£FIN	
F001 C		END		< IF 0072
* Carte < F >				
D001 C		IF	CARTE = 'F' AND FICH <> ' '	
001 C		MOVE	BIT34	BITD Turquoise, Sép, Sou
F001 C		END		< IF 0079
* Carte < I >				
D001 C		IF	CARTE = 'I' AND FICH <> ' '	
001 C		MOVE	BIT38	BITD * Rose
F001 C		END		< IF 0084
* Carte < D >				
* DS				
D001 C		IF	CARTE = 'D' AND FICH <> ' '	' AND £DS = 'DS'
001 C		MOVE	BIT38	BITD * Rose
F001 C		END		< IF 0090
* Table				
D001 C		IF	CARTE = 'D' AND £TAB = 'DIM'	
001 C		MOVE	BIT36	BITD *
F001 C		END		< IF 0095
* Pointeur				
D001 C		IF	CARTE = 'D' AND £PTR = '**'	
001 C		MOVE	BIT24	BITD *
F001 C		END		< IF 0099
D001 C		IF	CARTE = 'D' AND £BASED = 'BASED'	
001 C		MOVE	BIT3A	BITD *
F001 C		END		< IF 0102
* Carte < C >				
D001 C		IF	CARTE = 'C'	
* Incréments de Boucle				
D002 C	OPIF	IFEQ	'DO'	DO...
002 C	OPIF	OREQ	'IF'	IF..
002 C	OPI	OREQ	'DOW'	

002 C	OPI	OREQ	'DOU'			
002 C	OPI	OREQ	'FOR'			
002 C	OPER	OREQ	'SELECT'		SELEC	
002 C	OPIF	OREQ	'Do'			
002 C	OPIF	OREQ	'If'		IF..	
002 C	OPI	OREQ	'Dow'			
002 C	OPI	OREQ	'Dou'			
002 C	OPI	OREQ	'For'			
002 C	OPER	OREQ	'Select'		SELEC	
002 C		ADD	001	CPT		
002 C		MOVEL	'D'	INCR	Début	
002 C		MOVE	CPT	INCR		
D003 C	CPT	IFGT	0			
003 C		MOVEL	OPER	OP(CPT)	Mémoirese Opération	
003 C		MOVEL	SRCSEQ	SQ(CPT)	" N° ligne	
F003 C		END			< IFGT	0118
F002 C		END			< IFEQ	0110
D002 C	OPCAS	IFEQ	'CAS'			
002 C	OPCAS	OREQ	'Cas'			
002 C		MOVEL	'CAS '	INCR		
002 C		MOVE	'*'	BITCAS	1	
F002 C		END			< IFEQ	0123
D002 C	OPELSE	IFEQ	'ELSE'			
002 C	OPELSE	OREQ	'Else'			
002 C		MOVEL	'X'	INCR		
002 C		MOVE	CPT	INCR		
F002 C		END			< IFEQ	0127
	*					
D002 C	OPEND	IFEQ	'END'			
002 C	OPENDZ	ANDNE	'SR'			
002 C	OPEND	OREQ	'End'			
002 C	OPENDZ	ANDNE	'SR'			
D003 C	BITCAS	IFEQ	'*'			
003 C		MOVEL	'CAS '	INCR		
003 C		MOVE	' '	BITCAS		
X003 C		ELSE				
003 C		MOVEL	'F'	INCR	Fin IF ou DO OU SELE	
003 C		MOVE	CPT	INCR		
D004 C	WFIN	IFEQ	*BLANKS		Mise en place	
004 C	WFINDO	OREQ	'< '		< IF... DO...	
004 C	WFIND	OREQ	'< '		< IF... DO...	
004 C		MOVEL	'< '	WFINDO		<
D005 C	CPT	IFGT	0			
005 C		MOVEL	OP(CPT)	WFINOP	Opcde	
005 C		MOVE	SQ(CPT)	WFINSQ	125	
F005 C		END			< IFGT	0143
004 C		MOVE	WFIN	SRCDTA		
F004 C		END			< IFEQ	0140
D004 C	CPT	IFGT	0			
004 C		MOVE	*BLANKS	OP(CPT)		
F004 C		END			< IFGT	0149
003 C		SUB	001	CPT		
F003 C		END			< IFEQ	0134
F002 C		END			< IFEQ	0132
	* indicateurs					
D002 C		IF	INBLAN = *BLANKS			
D003 C		IF	INHI >= '01' and INHI <= '99'			
003 C		MOVEL	'>' HI			
F003 C		END			< IFNE	0157
D003 C		IF	INLO >= '01' and INLO <= '99'			
003 C		MOVEL	'<' LO			
F003 C		END			< IFNE	0160
D003 C		IF	INEQ >= '01' and INEQ <= '99'			
003 C		MOVEL	'=' EQ			
F003 C		END			< IFNE	0163
D003 C	INDIC	IFNE	*BLANKS			
003 C		MOVE	INDIC	SRCDTA		
F003 C		END			< IFNE	0166
F002 C		END			< IF	0156
	* IF					
D002 C		IF	OPER = 'IF' AND £FACT1 = *BLANKS OR			

```

002 C          OPER = 'If' AND £FACT1 = *BLANKS
002 C          MOVE      BIT30          BITD
F002 C          END
002 C          IF          OPI = 'DOW' AND £FACT1 = *BLANKS OR
002 C          OPI = 'Dow' AND £FACT1 = *BLANKS
002 C          MOVE      BIT3A          BITD
F002 C          END
002 C          IF          OPI = 'DOU' AND £FACT1 = *BLANKS OR
002 C          OPI = 'Dou' AND £FACT1 = *BLANKS
002 C          MOVE      BIT3A          BITD
F002 C          END
002 C          IF          OPI = 'FOR' AND £FACT1 = *BLANKS OR
002 C          OPI = 'For' AND £FACT1 = *BLANKS
002 C          MOVE      BIT3A          BITD
F002 C          END
002 C          IF          OPIF = 'DO' OR
002 C          OPIF = 'Do'
002 C          MOVE      BIT3A          BITD
F002 C          END
002 C          IF          OPIF = 'IF' OR
002 C          OPIF = 'If'
002 C          MOVE      BIT30          BITD
F002 C          END
* Exsr
D002 C          IF          OPER = 'EXSR' OR OPER = 'ExSr'
002 C          MOVE      BIT3A          BITD
F002 C          END
* Klist
D002 C          IF          OPER = 'KLIST' OR OPER = 'Klist'
002 C          MOVE      BIT3A          BITD
F002 C          END
* Else
D002 C          IF          OPER = 'ELSE' OR OPER = 'Else'
002 C          MOVE      BIT30          BITD
F002 C          END
D002 C          IF          OPER = 'SELECT' OR OPER = 'Select'
002 C          MOVE      BIT33          BITD
F002 C          END
D002 C          IF          OPER = 'ENDSL' OR OPER = 'EndSl'
002 C          MOVE      BIT33          BITD
F002 C          END
* Begsr,Endsr
D002 C          IF          OPER = 'BEGSR' OR OPER = 'ENDSR' OR
002 C          OPER = 'BegSr' OR OPER = 'EndSr'
002 C          MOVE      BIT28          BITD
F002 C          END
* Delete, Except, Exfmt, Lookup, Unlock, Update, Write
D002 C          IF          OPER = 'DELETE' OR OPER = 'EXCEPT' OR OPER =
002 C          'EXFMT' OR OPER = 'UPDATE' OR OPER = 'UNLOCK'
002 C          OR OPER = 'WRITE' OR
002 C          OPER = 'Delete' OR OPER = 'Except' OR OPER =
002 C          'ExFmt' OR OPER = 'Update' OR OPER = 'Unlock'
002 C          OR OPER = 'Write'
002 C          MOVE      BIT31          BITD
F002 C          END
* Chain, Setll, Setgt, Read, Readpe, Reade (attention au "CHAIN(n)")
D002 C          IF          OPER5= 'CHAIN' OR OPER5= 'SETLL' OR OPER5=
002 C          'SETGT' OR OPER4= 'READ' OR OPER5= 'READE'
002 C          OR OPER = 'READPE' OR OPER5= 'READC'
002 C          OR OPER5= 'READP' OR
002 C          OPER5= 'Chain' OR OPER5= 'Setll' OR OPER5=
002 C          'SetGt' OR OPER4= 'Read' OR OPER5= 'Reade'
002 C          OR OPER = 'ReadPe' OR OPER5= 'ReadC'
002 C          OR OPER5= 'ReadP'
002 C          MOVE      BIT33          BITD
F002 C          END
D002 C          IF          OPER = 'LOOKUP' OR OPER = 'SORTA' OR
002 C          OPER = 'Lookup' OR OPER = 'SortA'
002 C          MOVE      BIT30          BITD
F002 C          END
* Call, Callb, Plist

```

```

D002 C          IF          OPER = 'CALL' OR OPER = 'CALLB' OR OPER =
002 C          'PLIST' OR
002 C          OPER = 'Call' OR OPER = 'CallB' OR OPER =
002 C          'PList' OR OPER = 'CallP'
002 C          MOVE        BIT39          BITD          * Rose inv.
F002 C          END          < IF          0220
*
F001 C          END          < IF          0107
*
* Carte < 0 >
D001 C          IF          CARTE = 'O' AND FICH <> '          '
001 C          MOVE        BIT36          BITD
F001 C          END          < IF          0228
* Sortie enreg.
C          £LIGF          TAG
D001 C          POS80          IFEQ          BIT20
001 C          MOVE        '          '          SRCDTA
F001 C          END          < IFEQ          0233
C          MOVE        INCR          Z5          5
C          MOVE        BITD          Z5
C          MOVE        Z5          SRCDTA
*
C          UPDATE        RPGILE
* -----
C          £FIN          TAG
C          ADD          0001          NBR
C          MOVE        NBR          NBRA
C          99          RETURN
* -----
* $INIT :
* -----*
C          $INIT          BEGSR
C          Z-ADD          000          CPT          3 0
C          Z-ADD          0000          NBR          4 0
C          BITOFF        '01234567'          BIT20          1
C          MOVE        BIT20          BIT22          1
C          MOVE        BIT20          BIT24          1
C          MOVE        BIT20          BIT28          1
C          MOVE        BIT20          BIT30          1
C          MOVE        BIT20          BIT31          1
C          MOVE        BIT20          BIT33          1
C          MOVE        BIT20          BIT34          1
C          MOVE        BIT20          BIT36          1
C          MOVE        BIT20          BIT38          1
C          MOVE        BIT20          BIT39          1
C          MOVE        BIT20          BIT3A          1
C          BITON          '2'          BIT20          * Normal
C          BITON          '26'          BIT22          * X 2
C          BITON          '25'          BIT24          * X 2
C          BITON          '24'          BIT28          * Rouge
C          BITON          '23'          BIT30
C          BITON          '237'          BIT31          * Turquoise, inv.
C          BITON          '236'          BIT33          * Jaune, inv.
C          BITON          '235'          BIT34          * Sép.Soul,Turquoise
C          BITON          '2356'          BIT36          * Jaune Sép.soul X 2
C          BITON          '234'          BIT38          * Rose (Sép, Clign)
C          BITON          '2347'          BIT39          * Rose Inv.
C          BITON          '2346'          BIT3A          * Bleu
*
C          SETON          10          >
C          ENDSR

```