

/\*

001-Generated C code for functional specification 'DO\_ALL\_TAXES'

VERSION: 3.2.3.9 C-RAT

: 1.6.0.0

AUTHOR: Hamilton Technologies Inc. Copyright 1991-2015.

OPERATION: DO\_ALL\_TAXES

GENERATED: Tue Aug 25 15:34:47 2015

SCCSID: @(#) %M% %I% of %G%.

\*/

#include "FRIENDS.h"

#include "EMPLOYEES.h"

#include "PERSON.h"

#include "TAXDB.h"

#include "RAT.h"

#include "CHAR.h"

#include "NAT.h"

#include "BOOLEAN.h"

#include <stdio.h>

#include <math.h>

#include <errno.h>

#include "BOOLEAN.h"

#include "NAT.h"

fDO\_ALL\_TAXES(V0DB0,  
                  V0DB)

IDECLARE\_TAXDB(V0DB0)

ODECLARE\_TAXDB(V0DB)

{

                  /\* \_\_LOCAL\_VARIABLE\_DECLARATIONS\_\_ \*/

DECLARE\_PERSON(V0AFRIEND)

DECLARE\_PERSON(V0AFRIEND0)

```

DECLARE_FRIENDS(V0FRIENDS)
DECLARE_FRIENDS(V0FRIENDS0)
DECLARE_PERSON(V0ANEMP)
DECLARE_PERSON(V0ANEMP0)
DECLARE_EMPLOYEES(V0EMPS)
DECLARE_EMPLOYEES(V0EMPS0)
DECLARE_TAXDB(V0DB1)
DECLARE_RAT(V0TAXRATE)
DECLARE_EMPLOYEES(V1_0SETN)
DECLARE_CHAR(C14)
DECLARE_EMPLOYEES(V1_0SET2)
DECLARE_EMPLOYEES(V1_0SET1)
DECLARE_NAT(C17)
DECLARE_BOOLEAN(D1_0D4)
DECLARE_BOOLEAN(D1_1_0D1)
DECLARE_BOOLEAN(D1_1_0D2)
DECLARE_BOOLEAN(D1_1_0D3)
DECLARE_FRIENDS(V2_0SETN)
DECLARE_CHAR(C23)
DECLARE_FRIENDS(V2_0SET2)
DECLARE_FRIENDS(V2_0SET1)
DECLARE_NAT(C26)
DECLARE_BOOLEAN(D2_0D4)
DECLARE_BOOLEAN(D2_1_0D1)
DECLARE_BOOLEAN(D2_1_0D2)
DECLARE_BOOLEAN(D2_1_0D3)
/* __ITERATION_VARIABLE_DECLARATIONS__ */
int rec10REPEAT;
DECLARE_EMPLOYEES(R101_0SET1)
DECLARE_RAT(R100TAXRATE)
int rec29REPEAT;
DECLARE_FRIENDS(R292_0SET1)
DECLARE_RAT(R290TAXRATE)
/* __CONSTANT_DECLARATIONS_AND_ASSIGNMENTS__ */
DOT_K_CHAR('R',C14)
DOT_K_NAT(1,C17)

```

```

DOT_K_CHAR('R',C23)
DOT_K_NAT(1,C26)
          /* FUNCTION_SOURCE_CODE_BEGINNING */
MOVETO_RATE_TAXDB(V0DB0,V0TAXRATE)
MOVETO_EMPLOYEES_TAXDB(V0DB0,V0EMPS0)
IS_REJECT_EMPLOYEES(V0EMPS0,D1_1_0D1)
  if(D1_1_0D1<1)
{if(D1_1_0D1 == REJECT_BOOLEAN) {REJECT_TEST_BOOLEAN()}}
  IS_PLACE_EMPLOYEES(V0EMPS0,D1_1_0D2)
  if(D1_1_0D2<1)
{if(D1_1_0D2 == REJECT_BOOLEAN) {REJECT_TEST_BOOLEAN()}}
  IS_EMPTY_EMPLOYEES(V0EMPS0,D1_1_0D3)
  if(D1_1_0D3<1)
{if(D1_1_0D3 == REJECT_BOOLEAN) {REJECT_TEST_BOOLEAN()}}
  LOCATE_EMPLOYEES(C17,V0EMPS0,V1_0SET1)
R101_0SET1=V1_0SET1;
R100TAXRATE=V0TAXRATE;
rec10REPEAT=1;
while(rec10REPEAT--){
  ATNULL_EMPLOYEES(V1_0SET1,D1_0D4)
  if(D1_0D4<1)
{if(D1_0D4 == REJECT_BOOLEAN) {REJECT_TEST_BOOLEAN()}}
  MOVETO_EMPLOYEE_EMPLOYEES(V1_0SET1,V0ANEMP0)
  fCOMPUTE_TAX(V0ANEMP0,V0TAXRATE,&V0ANEMP);
  UPTO_EMPLOYEES_EMPLOYEE(V0ANEMP,V1_0SET2)
  NEXT_EMPLOYEES(C14,V1_0SET2,V1_0SETN)
  rec10REPEAT=1;
  V1_0SET1=V1_0SETN;
  V0TAXRATE=V0TAXRATE;
}/*FALSE*/
else{/*CLONE1_EMPLOYEES*/
  CLONE_EMPLOYEES(V1_0SET1,V0EMPS)
}/*TRUE*/
}
V1_0SET1=R101_0SET1;
V0TAXRATE=R100TAXRATE;

```

```

}/*FALSE*/
else{ /*CLONE1_EMPLOYEES*/
    CLONE_EMPLOYEES(V0EMPS0,V0EMPS)
}/*TRUE*/
}/*FALSE*/
else{ /*CLONE1_EMPLOYEES*/
    CLONE_EMPLOYEES(V0EMPS0,V0EMPS)
}/*TRUE*/
}/*FALSE*/
else{ /*CLONE1_EMPLOYEES*/
    CLONE_EMPLOYEES(V0EMPS0,V0EMPS)
}/*TRUE*/
    UPTO_TAXDB_EMPLOYEES(V0EMPS,V0DB1)
    MOVETO_FRIENDS_TAXDB(V0DB1,V0FRIENDS0)
    IS_REJECT_FRIENDS(V0FRIENDS0,D2_1_0D1)
        if(D2_1_0D1<1)
    {if(D2_1_0D1 == REJECT_BOOLEAN) {REJECT_TEST_BOOLEAN()}
    IS_PLACE_FRIENDS(V0FRIENDS0,D2_1_0D2)
        if(D2_1_0D2<1)
    {if(D2_1_0D2 == REJECT_BOOLEAN) {REJECT_TEST_BOOLEAN()}
    IS_EMPTY_FRIENDS(V0FRIENDS0,D2_1_0D3)
        if(D2_1_0D3<1)
    {if(D2_1_0D3 == REJECT_BOOLEAN) {REJECT_TEST_BOOLEAN()}
    LOCATE_FRIENDS(C26,V0FRIENDS0,V2_0SET1)
R292_0SET1=V2_0SET1;
R290TAXRATE=V0TAXRATE;
rec29REPEAT=1;
while(rec29REPEAT--){
    ATNULL_FRIENDS(V2_0SET1,D2_0D4)
        if(D2_0D4<1)
    {if(D2_0D4 == REJECT_BOOLEAN) {REJECT_TEST_BOOLEAN()}
    MOVETO_FRIEND_FRIENDS(V2_0SET1,V0AFRIEND0)
    fCOMPUTE_TAX(V0AFRIEND0,V0TAXRATE,&V0AFRIEND);
    UPTO_FRIENDS_FRIEND(V0AFRIEND,V2_0SET2)
    NEXT_FRIENDS(C23,V2_0SET2,V2_0SETN)
        rec29REPEAT=1;

```

```

    V2_0SET1=V2_0SETN;
    V0TAXRATE=V0TAXRATE;
}/*FALSE*/
else{/*CLONE1_FRIENDS*/
    CLONE_FRIENDS(V2_0SET1,V0FRIENDS)
}/*TRUE*/
}
V2_0SET1=R292_0SET1;
V0TAXRATE=R290TAXRATE;
}/*FALSE*/
else{/*CLONE1_FRIENDS*/
    CLONE_FRIENDS(V0FRIENDS0,V0FRIENDS)
}/*TRUE*/
}/*FALSE*/
else{/*CLONE1_FRIENDS*/
    CLONE_FRIENDS(V0FRIENDS0,V0FRIENDS)
}/*TRUE*/
}/*FALSE*/
else{/*CLONE1_FRIENDS*/
    CLONE_FRIENDS(V0FRIENDS0,V0FRIENDS)
}/*TRUE*/
    UPTO_TAXDB_FRIENDS(V0FRIENDS,*V0DB)

return;
}
/* ----- end of source -----*/

```