```
/*

     001-Generated C code for functional specification 'SCHEDULE_EVENT'

       VERSION: 3.2.3.9 C-RAT
              : 1.6.0.0
        AUTHOR: Hamilton Technologies Inc. Copyright 1991-2015.
     OPERATION: SCHEDULE_EVENT
     GENERATED: Tue Aug 25 16:55:14 2015

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

*/
#include "EVENTS.h"
#include "NAT.h"
#include "CHAR.h"
#include "AN_EVENT.h"
#include "BOOLEAN.h"

#include <stdio.h>
#include <math.h>
#include <errno.h>
#include "BOOLEAN.h"
#include "NAT.h"

fSCHEDULE_EVENT(V0RUNTIME,V0EVENT,V0EVENTS,
                V0ES)
IDECLARE_NAT(V0RUNTIME)
IDECLARE_AN_EVENT(V0EVENT)
IDECLARE_EVENTS(V0EVENTS)
ODECLARE_EVENTS(V0ES)
{
                  /* __LOCAL_VARIABLE_DECLARATIONS__  */
DECLARE_CHAR(C1)
DECLARE_AN_EVENT(V0ME3)
DECLARE_CHAR(C3)
```

```
DECLARE_AN_EVENT(V0ME2)
DECLARE_EVENTS(V0ES3)
DECLARE_CHAR(C6)
DECLARE_NAT(V0NEWRT)
DECLARE_CHAR(C8)
DECLARE_EVENTS(V0ES4)
DECLARE_AN_EVENT(V0E1)
DECLARE_AN_EVENT(V0E_)
DECLARE_NAT(V0ETX)
DECLARE_AN_EVENT(V0E)
DECLARE_NAT(V0NEWET)
DECLARE_AN_EVENT(V0ME1)
DECLARE_NAT(V0ET)
DECLARE_AN_EVENT(V0RE)
DECLARE_EVENTS(V0ES1)
DECLARE_CHAR(C19)
DECLARE_BOOLEAN(D0D1)
DECLARE_BOOLEAN(D0D4)
DECLARE_BOOLEAN(D0D8)
                /* __ITERATION_VARIABLE_DECLARATIONS__  */
int rec4MOVE_TO_INSERT;
DECLARE_NAT(R40RUNTIME)
DECLARE_AN_EVENT(R40EVENT)
DECLARE_EVENTS(R40ES1)
                /* __CONSTANT_DECLARATIONS_AND_ASSIGNMENTS__  */
DOT_K_CHAR('R',C1)
DOT_K_CHAR('L',C3)
DOT_K_CHAR('R',C6)
DOT_K_CHAR('L',C8)
DOT_K_CHAR('<',C19)
                /* __FUNCTION_SOURCE_CODE_BEGINNING__  */
   ISEMPTY_EVENTS(V0EVENTS,D0D1)
     if(D0D1<1)
 {if(D0D1 == REJECT_BOOLEAN) {REJECT_TEST_BOOLEAN()}
   NEXT_EVENTS(C19,V0EVENTS,V0ES1)
R40RUNTIME=V0RUNTIME;
```

```
R40EVENT=V0EVENT;
R40ES1=V0ES1;
rec4MOVE_TO_INSERT=1;
while(rec4MOVE_TO_INSERT--){
   ATNULL_EVENTS(V0ES1,D0D4)
     if(D0D4<1)
 {if(D0D4 == REJECT_BOOLEAN) {REJECT_TEST_BOOLEAN()}
   MOVETO_EVENTS(V0ES1,V0RE)
   MOVETO_TIME_AN_EVENT(V0RE,V0ET)
   GT_NAT(V0RUNTIME,V0ET,D0D8)
     if(D0D8<1)
 {if(D0D8 == REJECT_BOOLEAN) {REJECT_TEST_BOOLEAN()}
   PUT_TIME_AN_EVENT(V0RUNTIME,V0EVENT,V0ME1)
   SUB_NAT(V0ET,V0RUNTIME,V0NEWET)
   MOVETO_EVENTS(V0ES1,V0E)
   GET_TIME_AN_EVENT(V0E,V0ETX,V0E_)
   PUT_TIME_AN_EVENT(V0NEWET,V0E_,V0E1)
   UPTO_EVENTS_AN_EVENT(V0E1,V0ES4)
   INSERT_EVENTS(C8,V0ME1,V0ES4,*V0ES)
}/*FALSE*/
else{/*ADVANCE_BY_EVENT*/
   SUB_NAT(V0RUNTIME,V0ET,V0NEWRT)
   NEXT_EVENTS(C6,V0ES1,V0ES3)
     rec4MOVE_TO_INSERT=1;
     V0RUNTIME=V0NEWRT;
     V0EVENT=V0EVENT;
     V0ES1=V0ES3;
}/*TRUE*/
}/*FALSE*/
else{/*BEYOND_ALL_EVENTS*/
   PUT_TIME_AN_EVENT(V0RUNTIME,V0EVENT,V0ME2)
   INSERT_EVENTS(C3,V0ME2,V0ES1,*V0ES)
}/*TRUE*/
}
V0RUNTIME=R40RUNTIME;
V0EVENT=R40EVENT;
```

```
V0ES1=R40ES1;
}/*FALSE*/
else{/*NO_EVENTS*/
   PUT_TIME_AN_EVENT(V0RUNTIME,V0EVENT,V0ME3)
   INSERT_EVENTS(C1,V0ME3,V0EVENTS,*V0ES)
}/*TRUE*/

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