

```
/*
```

```
001-Generated C code for functional specification 'SIMULATOR'
```

```
VERSION: 3.2.3.9 C-RAT
```

```
: 1.6.0.0
```

```
AUTHOR: Hamilton Technologies Inc. Copyright 1991-2015.
```

```
OPERATION: SIMULATOR
```

```
GENERATED: Wed Aug 26 12:36:57 2015
```

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

```
*/
```

```
#include "SIMDB.h"
```

```
#include "SIMDB.h"
```

```
#include "SIMDB.h"
```

```
#include "STR.h"
```

```
#include "BOOLEAN.h"
```

```
#include "OS.h"
```

```
#include "OMAP.h"
```

```
#include "CHAR.h"
```

```
#include "TMAP.h"
```

```
#include <stdio.h>
```

```
#include <math.h>
```

```
#include <errno.h>
```

```
#include "BOOLEAN.h"
```

```
#include "NAT.h"
```

```
fSIMULATOR(V0OS0,  
            V0OS)
```

```
IDECLARE_OS(V0OS0)
```

```
ODECLARE_OS(V0OS)
```

```
{
```

```
/* __LOCAL_VARIABLE_DECLARATIONS__ */
```

```
DECLARE_SIMDB(V0SX)
```

```
DECLARE_SIMDB(V0SM)
DECLARE_STR(V0STOREPATH)
DECLARE_STR(C4)
DECLARE_SIMDB(V0SN)
DECLARE_SIMDB(V0S1)
DECLARE_BOOLEAN(V0ISF)
DECLARE_SIMDB(V0S)
DECLARE_SIMDB(V0S0)
DECLARE_STR(V0DISPLAYPATH)
DECLARE_STR(C11)
DECLARE_STR(V0OMAPPATH)
DECLARE_STR(C13)
DECLARE_BOOLEAN(D0D7)
DECLARE_OMAP(V1_OM3)
DECLARE_BOOLEAN(V1_B2)
DECLARE_OMAP(V1_OM2)
DECLARE_STR(V1_DFNM2)
DECLARE_STR(C21)
DECLARE_CHAR(C22)
DECLARE_STR(V1_DFNM1)
DECLARE_CHAR(C24)
DECLARE_SIMDB(V1_NEW_OBJECT)
DECLARE_OMAP(V1_OM0)
DECLARE_CHAR(C27)
DECLARE_CHAR(C28)
DECLARE_STR(V1_OMFNM1)
DECLARE_CHAR(C30)
DECLARE_STR(V1_OMFNM0)
DECLARE_CHAR(C32)
DECLARE_STR(V1_FNM0)
DECLARE_STR(C34)
DECLARE_STR(V1_TYPENM)
DECLARE_TMAP(V1_TM)
DECLARE_OS(V1_OS)
DECLARE_STR(V1_DFNM)
DECLARE_STR(V1_OMFNM2)
```

```

DECLARE_OMAP(V1_OM1)
DECLARE_BOOLEAN(D1_D13)
DECLARE_OMAP(V2_TOM1)
DECLARE_OMAP(V2_TOM)
DECLARE_STR(C44)
DECLARE_CHAR(C45)
DECLARE_STR(V2_OMFNM2)
DECLARE_CHAR(C47)
DECLARE_CHAR(C48)
DECLARE_STR(V2_OMFNM1)
DECLARE_CHAR(C50)
DECLARE_CHAR(C51)
DECLARE_STR(V2_FNM0)
DECLARE_STR(C53)
DECLARE_STR(V2_TYPENM)
DECLARE_STR(V2_OMFNM0)
DECLARE_TMAP(V2_TM)
DECLARE_OS(V2_OS)
DECLARE_STR(V2_DISPLAYFNM)
DECLARE_BOOLEAN(V3_B)
DECLARE_STR(V3_OMFNM2)
DECLARE_CHAR(C61)
DECLARE_CHAR(C62)
DECLARE_STR(V3_OMFNM1)
DECLARE_CHAR(C64)
DECLARE_STR(V3_OMFNM0)
DECLARE_CHAR(C66)
DECLARE_STR(V3_OMF0)
DECLARE_STR(C68)
DECLARE_STR(V3_TYPENM)
DECLARE_TMAP(V3_TM)
DECLARE_OS(V3_OS)
/* __ITERATION_VARIABLE_DECLARATIONS__ */
int rec28SIMULATE;
DECLARE_SIMDB(R280S0)
/* __CONSTANT_DECLARATIONS_AND_ASSIGNMENTS__ */

```

```

NEWSTACK_TRACE_IDSC("SIMULATOR");
DOT_K_STR("./output.",C4)
DOT_K_STR("./input.",C11)
DOT_K_STR("./input.",C13)
DOT_K_STR(".display",C21)
DOT_K_CHAR('>',C22)
DOT_K_CHAR('<',C24)
DOT_K_CHAR('l',C27)
DOT_K_CHAR('<',C28)
DOT_K_CHAR('<',C30)
DOT_K_CHAR('<',C32)
DOT_K_STR(".omap",C34)
DOT_K_STR(".display",C44)
DOT_K_CHAR('>',C45)
DOT_K_CHAR('L',C47)
DOT_K_CHAR('<',C48)
DOT_K_CHAR('<',C50)
DOT_K_CHAR('<',C51)
DOT_K_STR(".omap",C53)
DOT_K_CHAR('l',C61)
DOT_K_CHAR('<',C62)
DOT_K_CHAR('<',C64)
DOT_K_CHAR('<',C66)
DOT_K_STR(".omap",C68)
        /* __FUNCTION_SOURCE_CODE_BEGINNING__ */
K_STR(C13,V0OS0,V0OMAPPATH)
K_STR(C11,V0OS0,V0DISPLAYPATH)
K_OS(V0DISPLAYPATH,V1_OS)
READTMAP_SIMDB(V1_OS,V1_TM)
NODENAME_TMAP(V1_TM,V1_TYPENM)
K_STR(C34,V1_TYPENM,V1_FNM0)
MERGE_STR(C32,V1_TYPENM,V1_FNM0,V1_OMFNM0)
COPY_STR(V1_OMFNM0,V1_DFNM)
MERGE_STR(C30,V0DISPLAYPATH,V1_OMFNM0,V1_OMFNM1)
CASE_STR(C28,C27,V1_OMFNM1,V1_OMFNM2)
LOAD_OMAP(V1_OMFNM2,V1_TM,V1_OM0)

```

```

ISREJECT_OMAP(V1_OM0,D1_D13)
  if(D1_D13<1)
{if(D1_D13 == REJECT_BOOLEAN) {REJECT_TEST_BOOLEAN()}}
  CLONE_OMAP(V1_OM0,V1_OM1)
}/*FALSE*/
else{/*LOAD_REJECTED*/
  K_SIMDB(V1_OMFNM2,V1_NEW_OBJECT)
  CONVERT_OMAP_SIMDB(V1_NEW_OBJECT,V1_OM1)
}/*TRUE*/
  MERGE_STR(C24,V0OMAPPATH,V1_DFNM,V1_DFNM1)
  MERGE_STR(C22,C21,V1_DFNM1,V1_DFNM2)
  DISPLAY_OMAP(V1_DFNM2,V1_OM1,V1_OM2)
  STORE_OMAP(V1_OMFNM2,V1_OM2,V1_B2)
  CLONE_OMAP(V1_OM2,V1_OM3)
  CONVERT_SIMDB_OMAP(V1_OM3,V0S0)
R280S0=V0S0;
rec28SIMULATE=1;
while(rec28SIMULATE--){
  FIS_FINISHED(V0S0,&V0ISF);
  COPY_BOOLEAN(V0ISF,D0D7)
  if(D0D7<1)
{if(D0D7 == REJECT_BOOLEAN) {REJECT_TEST_BOOLEAN()}}
  FPROCESS_QUEUE(V0S0,&V0S1);
  FDO_EVENTS(V0S1,&V0SN);
  rec28SIMULATE=1;
  V0S0=V0SN;
}/*FALSE*/
else{/*CLONE1_SIMDB*/
  CLONE_SIMDB(V0S0,V0S)
}/*TRUE*/
}
V0S0=R280S0;
  K_STR(C4,V0OS0,V0STOREPATH)
  K_OS(V0STOREPATH,V2_OS)
  READTMAP_SIMDB(V2_OS,V2_TM)
  NODENAME_TMAP(V2_TM,V2_TYPENM)

```

```
K_STR(C53,V2_TYPENM,V2_FNM0)
MERGE_STR(C51,V2_TYPENM,V2_FNM0,V2_OMFNM0)
MERGE_STR(C50,V0STOREPATH,V2_OMFNM0,V2_OMFNM1)
CASE_STR(C48,C47,V2_OMFNM1,V2_OMFNM2)
MERGE_STR(C45,C44,V2_OMFNM2,V2_DISPLAYFNM)
CONVERT_OMAP_SIMDB(V0S,V2_TOM)
DISPLAY_OMAP(V2_DISPLAYFNM,V2_TOM,V2_TOM1)
CONVERT_SIMDB_OMAP(V2_TOM1,V0SM)
K_OS(V0SM,V3_OS)
READTMAP_SIMDB(V3_OS,V3_TM)
NODENAME_TMAP(V3_TM,V3_TYPENM)
K_STR(C68,V3_TYPENM,V3_OMF0)
MERGE_STR(C66,V3_TYPENM,V3_OMF0,V3_OMFNM0)
MERGE_STR(C64,V0STOREPATH,V3_OMFNM0,V3_OMFNM1)
CASE_STR(C62,C61,V3_OMFNM1,V3_OMFNM2)
STORE_SIMDB(V3_OMFNM2,V0SM,V3_B)
CLONE_SIMDB(V0SM,V0SX)
CLONE_OS(V0OS0,*V0OS)
ENDSTACK_IDSC(); /* STR Garbage Collector */

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