

# SimH Sample Software Packages

## 15-Jan-2006

### **COPYRIGHT NOTICE**

The following copyright notice applies to the SIMH source, binary, and documentation:

Original code published in 1993-2006, written by Robert M Supnik  
Copyright (c) 1993-2006, Robert M Supnik

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL ROBERT M SUPNIK BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of Robert M Supnik shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from Robert M Supnik.

1	PDP-8.....	3
1.1	ESI-X .....	3
1.2	FOCAL69 .....	3
1.3	PDP-8 OS/8.....	3
1.4	PDP-8 TSS/8.....	4
2	PDP-11 .....	5
2.1	UNIX V5, V6, V7.....	5
2.1.1	UNIX V5.....	5
2.1.2	UNIX V6.....	5
2.1.3	UNIX V7.....	5
2.2	RT-11 .....	6
2.2.1	RT-11 V4 .....	6
2.2.2	RT-11 V5.3 .....	6
3	Nova and Eclipse RDOS.....	6
4	PDP-1 .....	7
4.1	PDP-1 LISP.....	7
4.2	PDP-1 DDT.....	7
5	PDP-7 SIM8.....	7
6	PDP-9/PDP-15 .....	8
6.1	FOCAL .....	8
6.2	Advanced Software System/Keyboard Monitor .....	8
6.3	Advanced Software System/Foreground Background.....	9
6.4	DOS-15 .....	10
7	IBM 1401 .....	11
7.1	Single Card "Koans" .....	11
7.2	Diagnostic Tape .....	11
7.3	SPS.....	12
8	HP2116 16K BASIC.....	13
9	PDP-10 TOPS-10 7.03, TOPS-10 7.04, TOPS-20 V4.1.....	13
10	Interdata 32b Systems .....	14
10.1	Interdata 7/32 UNIX .....	14
10.1.1	UNIX V6.....	14
10.1.2	UNIX V7.....	14

This memorandum documents the sample software packages available to run with the SIMH simulators. Many of these packages are available under limited use licenses; please read the license terms included with the software.

## 1 PDP-8

### 1.1 ESI-X

ESI-X is an interactive program for technical computation. It can execute both immediate commands and stored programs (like BASIC). ESI-X is provided as both source and as a binary loader format paper-tape image. For more information see the documentation included with the program. My thanks to Dave Waks, who wrote the program, and to Paul Pierce and Tim Litt, who recovered the source from its archival medium.

To load and run ESI-X:

```
sim> load esix.bin
sim> run 5400
_TYPE 2+2.
      2+2 =      4
```

### 1.2 FOCAL69

FOCAL69 is an interactive program for technical computations. It can execute both immediate commands and stored programs (like BASIC). FOCAL69 is provided as a binary loader format paper-tape image. To load and run FOCAL69:

```
sim> load focal69.bin
sim> run 200
*TYPE 2+2
=      4.000*
```

### 1.3 PDP-8 OS/8

OS/8 is the PDP-8's mass storage-based operating system. It provides a program development and execution environment for assembler, BASIC, and FORTRAN programs. OS/8 is provided under license, as is, without fee, by Digital Equipment Corporation, for non-commercial use only. Please read the enclosed license agreement for full terms and conditions. This license agreement must be reproduced with any copy of the OS/8 disk images. My thanks to Doug Jones of the University of Iowa, who provided the disk images, and to Digital Equipment Corporation, which provided the license.

To boot and run OS/8:

```
sim> att rx0 os8_rx.dsk
sim> att rx1 os8f4_rx.dsk
sim> boot rx0

.DA dd-mmm-yy
.
```

Note that OS/8 only recognizes upper case characters. The first disk (drive 0) is the system disk; it also includes BASIC. The second disk (drive 1) includes FORTRAN.

## 1.4 PDP-8 TSS/8

TSS/8 is the PDP-8's timesharing system. It provides a program development and execution environment for assembler, BASIC, and FORTRAN programs. TSS/8 is provided as is, without fee, by Digital Equipment Corporation, for non-commercial use only. My thanks to John Wilson of Dbit Inc, who provided the disk image and the initialization tape source. Note: your environment must have a functioning second Teletype; that is, you cannot run TSS if your host system does not support the SIMH sockets library.

To load and run TSS/8:

- Load the paper-tape bootstrap:

```
sim> load tss8_init.bin
```

- Mount the TSS/8 disk image of the RF08:

```
sim> attach rf tss8_rf.dsk
```

- Assign a TCP/IP port to the Telnet listener for the extra terminals:

```
sim> attach ttix <port #>          4000 typically works
```

- Run the bootstrap:

```
sim> run 24200
```

- TSS/8 will boot and go through its startup dialog

```
LOAD, DUMP, START, ETC? START
MONTH-DAY-YEAR: mm:dd:yy          numeric, yy in range [74:85]
HR:MIN - hh:mm                   numeric, 24 hour format
(type cr to get attention)
```

.

- and is now ready for login. The list of accounts and passwords:

```
PPN  Password
[0,1]  VH3M
[0,2]  LXHE
[0,3]  SHUG
[77,77]
[1,10] WBCN
[20,1] DT
[20,2] PT
[20,3] TSS8
[20,4] EDIT
[20,5] 4TH
[1,50] JERK
```

- Login using one of the existing accounts. The login command won't echo:

```
.LOGIN 2 LXHE          privileged library account
```

```
TSS/8.24 JOB 01 [00,02] K00    23:23:06
SYSTEM IS DOWN, INC.
```

- The system is now ready for commands. To get a directory listing:

```
.R CAT
```

Other users can log in by connecting, from a Telnet client, to localhost on the port specified in the ATTACH TTIX command.

## 2 PDP-11

### 2.1 UNIX V5, V6, V7

UNIX was first developed on the PDP-7; its first widespread usage was on the PDP-11. UNIX provides a program development and execution environment for assembler and C programs. UNIX V5, V6, V7 for the PDP-11 is provided under license, as is, without fee, by Caldera Corporation, for non-commercial use only. Please read the enclosed license agreement for full terms and conditions. This license must be reproduced with any copy of the UNIX V5, V6, V7 disk images. My thanks to PUPS, the PDP-11 UNIX Preservation Society of Australia, which provided the disk images, and to Caldera, which provided the license.

#### 2.1.1 UNIX V5

UNIX V5 is contained on a single RK05 disk image. To boot UNIX:

```
sim> set cpu u18
sim> att rk0 unix_v5_rk.dsk
sim> boot rk
@unix
login: root
#ls -l
```

#### 2.1.2 UNIX V6

UNIX V6 is contained on four RK05 disk images. To boot UNIX:

```
sim> set cpu u18
sim> att rk0 unix0_v6_rk.dsk
sim> att rk1 unix1_v6_rk.dsk
sim> att rk2 unix2_v6_rk.dsk
sim> att rk3 unix3_v6_rk.dsk
sim> boot rk0
@unix
login: root
# ls -l
```

#### 2.1.3 UNIX V7

UNIX V7 is contained on a single RL02 disk image. To boot UNIX:

```
sim> set cpu u18
sim> set r10 RL02
sim> att r10 unix_v7_r1.dsk
sim> boot r10
```

```
@boot
New Boot, known devices are hp ht rk rl rp tm vt
: rl(0,0)rl2unix
#
```

A smaller image is contained on a single RK05 disk image. To boot UNIX:

```
sim> set cpu u18
sim> att rk0 unix_v7_rk.dsk
sim> boot rk0
@boot
New Boot, known devices are hp ht rk rl rp tm vt
: rk(0,0)rkunix
# STTY -LCASE
#
```

## 2.2 RT-11

RT-11 is the PDP-11's single user operating system. It provides a program development and execution environment for assembler, BASIC, and FORTRAN programs. RT-11 is provided under license, as is, without fee, by Mentec Corporation, for non-commercial use ONLY ON THIS SIMULATOR. Please read the enclosed license agreement for full terms and conditions. This license agreement must be reproduced with any copy of the RT-11 disk image. My thanks to John Wilson, a private collector, who provided the disk image for RT-11 V4; to Megan Gentry, of Digital Equipment Corporation, who provided the disk image for RT-11 V5.3; and to Mentec Corporation, which provided the license.

### 2.2.1 RT-11 V4

RT-11 is contained in a single RK05 disk image. To boot and run RT-11:

```
sim> att rk0 rtv4_rk.dsk
sim> boot rk0
```

For RL, HK, RM, and RP series disks, RT-11 expects to find a manufacturer's bad block table in the last track of the disk. Therefore, INITIALIZATION of a new (all zero's) disk fails, because there is no valid bad block table. To create a minimal bad block table, use the SET <unit> BADBLOCK command.

### 2.2.2 RT-11 V5.3

RT-11 is contained in a single RL02 disk image. To boot and run RT-11:

```
sim> set r10 r102
sim> att r10 rtv53_rl.dsk
sim> boot r10
```

This is a full RT-11 distribution kit. It expects the user to copy the distribution pack and generate a new system. This requires mounting a blank pack on RL1. When a blank pack is attached to the simulator, a bad block table must be created with the SET <unit> BADBLOCK command.

## 3 Nova and Eclipse RDOS

RDOS is the Nova's real-time mass storage operating system. It provides a program development and execution environment for assembler, BASIC, and FORTRAN programs. RDOS is provided under license, as is, without fee, by Data General Corporation, for non-commercial use only. Please read the enclosed license agreement for full terms and conditions. This license agreement must be reproduced with any copy of the RDOS disk image. My thanks to Carl Friend, a private collector, who provided the disk image, and to Data General Corporation, which provided the license.

To boot and run RDOS for the Nova:

```
sim> att dp0 rdos_d31.dsk
sim> set tti dasher
sim> boot dp0
FILENAME? (cr)
DATE (mm/dd/yy)? xx/yy/zz
TIME (hh:mm:ss)? hh:mm:ss
R
list/e
```

To boot and run RDOS for the Eclipse:

```
sim> att dp0 zrdos75.dsk
sim> set tti dasher
sim> boot dp0
FILENAME? (cr)
DATE (mm/dd/yy)? xx/yy/zz
TIME (hh:mm:ss)? hh:mm:ss
R
list/e
```

## 4 PDP-1

### 4.1 PDP-1 LISP

PDP-1 LISP is an interactive interpreter for the Lisp language. It can execute both interactive commands and stored programs. The startup instructions for LISP are complicated; the documentation included with the program provides information on loading and operating LISP. My thanks to Peter Deutsch, who wrote the program, to Gordon Greene, who typed it in from a printed listing, and to Paul McJones, who helped with the final debug process.

### 4.2 PDP-1 DDT

PDP-1 DDT is an interactive debugging tool for the PDP-1. It provides symbolic debugging capabilities for PDP-1 programs. The documentation included with the program provides information on loading and operating DDT. My thanks to Derek Peschel, who transcribed and debugged DDT.

## 5 PDP-7 SIM8

PDP-7 SIM8 is a PDP-8 simulator for the PDP-7. It implements an 8K PDP-8/I with keyboard, teleprinter, reader, punch, and line printer. It provides an interactive console environment for control and debug of the simulated PDP-8. For more information see the documentation included with the program. My thanks to

Dave Waks, who wrote the program, and to Paul Pierce and Tim Litt, who recovered the source from its archival medium.

To load and run SIM8:

```
sim> load sim8.rim
sim> run
AC/ 0000
```

## 6 PDP-9/PDP-15

### 6.1 FOCAL

FOCAL15 is an interactive program for technical computations. It can execute both immediate commands and stored programs (like BASIC). FOCAL15 is provided as a binary loader format paper-tape image. My thanks to Al Kossow, who provided the binary image. To load and run FOCAL15:

```
sim> load focal15.bin
sim> run
*TYPE FSQT(2),!
= 1.4142
*
```

### 6.2 Advanced Software System/Keyboard Monitor

The Advanced Software System Keyboard Monitor is the simplest mass storage monitor for the PDP-15. It offers single-user program development and execution capabilities. My thanks to David Gesswein, who provided the DECTape images of ADSS/KM. To load and run ADSS/KM-15:

- On the PDP-9 (only), initialize extend mode to on:

```
sim> d extm_init 1
```

- Load the paper-tape bootstrap into upper memory:

```
sim> load dec-15u.rim 77637
```

You *must* specify the load address.

- Verify that the bootstrap loaded correctly:

```
sim> ex pc
PC: 077646
```

- Change the default line printer to be the LP09 rather than the Type 647 (PDP-9) or LP15 (PDP-15):

```
sim> set lpt disabled
sim> set lp9 enabled
```

- Mount the Advanced Software System DECTape image on DECTape unit 0:

```
sim> attach dt adss15_32k.dtp
```



- Run the bootstrap:

```
sim> run
```

- The DECtape will boot and print out

```
KMS9-15 V5B000  
$
```

and is now ready for commands. Recognized commands include:

```
D          list system device directory  
I          list available commands  
R          list device assignments  
SCOM      list systems communication region
```

- To run Focal, assign unused DAT slot 10 and then load Focal:

```
$A LPA0 10  
$GLOAD  
LOADER V5B000  
>_FOCAL<altmode = control-[>  
FOCAL V9A  
*TYPE 2+2,!  
4.0000  
*
```

### 6.3 Advanced Software System/Foreground Background

The Advanced Software System/Foreground Background is a real-time monitor for the PDP-15. It offers foreground and background execution of programs. My thanks to David Gesswein, who provided the DECtape images of ADSS/FB.

Note: your environment must have a functioning second Teletype; that is, you cannot run Foreground/Background if your host system does not support the SIMH sockets library.

To load and run ADSS/FB:

- Load the paper-tape bootstrap into upper memory:

```
sim> load dec-15u.rim 77637
```

You *must* specify the load address.

- Verify that the bootstrap loaded correctly:

```
sim> ex pc  
PC: 077646
```

- Mount the Foreground/Background DECtape image on DECtape unit 0:

```
sim> attach dt fb15_32k.dtp
```

- Assign a TCP/IP port to the Telnet listener for the second terminal:

```
sim> assign tt1 <port #> 4000 typically works
```

- Start a Telnet client to act as the second terminal and connect to localhost on the specified port.
- Run the bootstrap:

```
sim> run
```

- The DECtape will boot, and the Foreground/Background Monitor will print on the second terminal

```
F9/15 V4A  
$
```

and is now ready for commands. Recognized commands include:

```
D 0      list system device directory  
R        list device assignments
```

- To activate the background, load IDLE into the foreground:

```
$A DTA0 -4  
$GLOAD  
FGLOAD V2A  
>_IDLE<altmode = control-[>
```

- Background terminal responds with

```
B9/15 V4A  
$
```

and is now ready for commands.

## 6.4 DOS-15

DOS-15 is a more polished version of ADSS-15 using the RF15 or RP15 as its system device. My thanks to Hans Pufal, who recovered DOS-15 from DECtape and deduced how to reconstruct disk images from DECtapes. To load and run DOS-15:

- Load the DOS-15 paper tape bootstrap into upper memory:

```
sim> load rfsboot.rim 77637
```

You *must* specify the load address.

- Mount the DOS-15 RF disk image on the RF15:

```
sim> att rf dosv2a_4p.rf
```

- Run the bootstrap:

```
sim> run
```

- The RF disk will boot and print out

```
DOS-15 V2A
```

ENTER DATE (MM/DD/YY) -

- Enter the date (the year must be between 70 and 99). DOS-15 will print its prompt and is now ready for commands. Recognized commands include:

I	system information
S	system configuration
R	system device assignments

## 7 IBM 1401

### 7.1 Single Card "Koans"

One of the art forms for the IBM 1401 was packing useful programs into a single punched card. Three samples are included:

i1401_ctolp.cd	prints a card deck on the line printer
i1401_ctopu.cd	copies a card deck to the card punch
i1401_hello.cd	prints "HELLO WORLD" on the line printer and stops

To use the reproduction cards, simply insert them at the beginning of a text file, terminated by newline. Attach the modified file to the card reader, attach a blank file to the output device, and boot the card reader.

### 7.2 Diagnostic Tape

The software and writeup were provided by Charles Owens.

This 1401 Diagnostics tape is a bootable tape containing a series of 1401 diagnostics dating from about 1962. The 1407 Inquiry console is not used; all control is via the front panel.

To run in the simulator, attach thusly:

```
sim> attach mt1 1401diag.mt
sim> attach lpt errorlist.txt
sim> boot mt1
```

The simulator will halt with IS = 433. At this point, you can set options through the sense switches and memory.

D 1252 "1"	Will cause headings to print for each test run. Otherwise no printing will occur unless there are errors.
D SSB 1	Loop if an error is detected.
D SSC 1	Prints all test cases not just errors.
D SSD 1	Repeat the test run over and over.
D SSE 1	Halt if any error is detected, otherwise continue.

When you continue from this halt (use C to CON), the simulator will halt at 3001. Enter C again and the tape will spin thru a series of basic CPU diagnostics.

## 7.3 SPS

The software and writeup were provided by Charles Owens.

sps1.obj and sps2.obj are the object card decks for the "Symbolic Programming System", a primitive assembler for the 1401 that predates the better known and more functional Autocoder.

To use SPS, write an SPS program using your favorite editor (two examples are provided, hello.sps and diaglist.sps). SPS decks are not free-format, but operands must be placed in columns:

```
1 - 5      Line Count (optional)
6 -        Count (number of characters when defining a constant).
8 - 13     Label (six characters must start with alphabetic).
14 - 16    Opcode: Examples:
           A = Add
           B = Branch (must be d-mod for conditional)
           BWZ = Branch if Wordmark or Zone
           C = Compare
           CC = Carriage Control (printer)
           CS = CLear Storage
           CU = Control Unit (e.g. tape)
           CW = Clear Workmark
           D = Divide
           DC = Define Constant (no wordmark)
           DCW = Define Constant (starts in 24, length in 6-7)
           END = End of program
           LCA = Load Characters
           H = Halt
           M = Multiply
           MCE = Move and Edit
           MCS = Move and Supress Zeros
           MCW = Move Characters
           MN = Move Numeric
           MZ = Move Zone
           ORG = Define Origin Point
           P = Punch Card
           R = Read Card
           S = Subtract
           SS = Stacker Select
           SW = Set Wordmark
           W = Write Line
           ZA = Zero and Add
           ZS = Zero and Subtract
           Tape:
           MCW %UX YYY W      Write Tape from addr YYY w/o wordmarks)
           LCA %UX YYY W      Write Tape, Unit X from addr YYY w/wordmarks)
           MCW %UX YYY R      Read Tape from addr YYY w/o wordmarks)
           LCA %UX YYY R      Read Tape, Unit X from addr YYY w/wordmarks)
           CU %UX M           Write Tape Mark
           CU %UX E           Skip and Blank Tape
           CU %UX B           Backspace Record
           CU %UX R           Rewind Tape
           CU %UX U           Rewind and Unlaod tape
17 - 22    Address for A-operand (label or 4-digit actual address)
23 - 23    blank, + or - to adjust A-operand by a constant
24 - 26    3-digit number to adjust A-operand by if 23 is + or -
```

27 - 27        Index (?)  
 28 - 33        Address for B-operand  
 34 - 34        Blank, + or -  
 35 - 37        3-digit number of adjust B-address by if 34 is + or -  
 38 - 38        Index (?)  
 39 - 39        D-modifier for this instruction.    Notes:  
               / = Compare is unequal  
               S = Branch if Compare Equal  
               T = B is less than A  
               U = B is greater than A  
               L = Tape read error  
               K = Tape end of reel  
 40 - 55        Comments

The SPS deck should start with an ORG operation to specify where in storage the program starts, and end with an END card, with an optional A-operand showing where to start execution.

To assemble an SPS program, place the SPS source between the SPS1.OBJ and the SPS2.OBJ deck, and another copy of the same source after the SPS2.OBJ deck (SPS is a two-pass assembler). SPS prints a listing on LPT and punches an object deck on CDP, ready to run.

A UNIX command script to assemble an SPS deck painlessly is in "sps". To use the script, enter "sps programname". The script creates programname.lst for the listing and programname.obj for the object deck. Windows users are out of luck, for now.

## 8 HP2116 16K BASIC

HP BASIC is a paper-tape centric implementation of BASIC for a 16KW HP2116. Device numbers correspond to the default simulator settings:

```

PTR   =    10
TTY   =    11
PTP   =    12

```

The program is a complete but early BASIC and has one unusual requirement: all programs must include a valid END statement to run correctly. My thanks to Jeff Moffatt for providing the program.

To load and run BASIC:

```

sim> load basic1.abs
sim> run 100
READY
10 PRINT SQR(2)
20 END
RUN
1.41421

```

## 9 PDP-10 TOPS-10 7.03, TOPS-10 7.04, TOPS-20 V4.1

TOPS-10 was the primary time-shared operating system for the PDP-10. TOPS-20 was a popular alternative derived from the BBN TENEX system. Installation and distribution tapes for TOPS-10 7.03, TOPS-10 7.04, and TOPS-20 4.1 are available at <http://pdp-10.trailing-edge.com>.

## 10 Interdata 32b Systems

### 10.1 Interdata 7/32 UNIX

Interdata UNIX V6 was the first port of UNIX from its native environment on the PDP-11. The port was done by Richard Miller and his colleagues at the University of Wollongong, Australia, in 1976-77. UNIX V6 and V7 for the Interdata 7/32 is provided under license, as is, without fee, by Caldera Corporation, for non-commercial use only. Please read the enclosed license agreement for full terms and conditions. This license must be reproduced with any copy of the UNIX V6 and V7 disk images. My thanks to Richard Miller, without whose guidance, knowledge, and patience the recreation of Interdata UNIX would not have been possible, and to Caldera, which provided the license.

#### 10.1.1 UNIX V6

UNIX V6 is contained on a single 10MB disk image. To boot UNIX V6:

```
sim> set ttp ena
sim> set pas dev=12
sim> set mt dev=85
sim> att -e dp0 iu6_dp0.dsk
sim> boot dp0
?unix
login: root
# ls -l
```

#### 10.1.2 UNIX V7

UNIX V7 is contained on two 10MB disk images. To boot UNIX V7:

```
sim> set ttp ena
sim> set pas dev=12
sim> set mt dev=85
sim> att -e dp0 iu7_dp0.dsk
sim> att -e dp1 iu7_dp1.dsk
sim> boot dp0
Boot
: dsk(1,0)unix
#
```

Type ^D to enable multiuser mode. The root password is root.