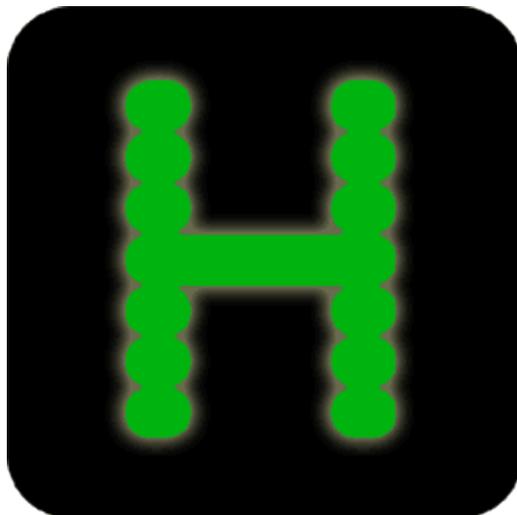


**Hercules System/370, ESA/390, z/Architecture Emulator**

# **Hercules – Operations and Utilities Guide**

*Version 4 Release 00*



**Draft - November 21, 2015**

**Hercules System/370, ESA/390, z/Architecture Emulator**

# **Hercules – Operations and Utilities Guide**

*Version 4 Release 00*



**First Edition, November 21, 2015**

**HEOP040000-00**

---

# Contents

Contents .....	3
Figures.....	5
Tables.....	7
1. Preface .....	8
1.1 Edition information .....	8
1.2 What this book is about.....	8
1.3 Who should read this book .....	8
1.4 What you need to know to understand this book.....	8
1.5 How to use this book.....	8
1.6 Revision Notice .....	8
1.7 Readers Comments .....	9
1.8 Legal Advice.....	9
1.9 Trademarks .....	9
1.10 Acknowledgements .....	10
2. Related Publications .....	11
2.1 Hercules – General Information .....	11
2.2 Hercules – Installation Guide .....	11
2.3 Hercules – User Reference Guide .....	11
2.4 Hercules – Operations and Utilities Guide.....	11
2.5 Hercules – Messages and Codes .....	11
2.6 Hercules – Reference Summary.....	11
3. Summary of changes .....	12
3.1 Version 4, First Edition (HEOP040000-00).....	12
Part I: Hercules Operations .....	13
4. Hercules Operations .....	14
4.1 Starting the Emulator .....	14
4.2 The Console Panel.....	14
4.3 The Device and Status Panel .....	18
4.4 The Web Browser Interface .....	24
4.5 Graphical User Interfaces (GUIs).....	36
Part II: Hercules Supplied Utilities .....	37
5. Hercules Supplied Utilities .....	38
5.1 Overview .....	38
5.2 CCKDCDSK (CCKD DASD file integrity verification, recovery and repair utility) .....	41
5.3 CCKDCOMP (CCKD DASD file compression utility) .....	43
5.4 CCKDDIAG (CCKD DASD file diagnostics utility) .....	45
5.5 CCKDSWAP (CCKD DASD file swap-endian program) .....	48
5.6 DASDCAT (Display PDS datasets and members) .....	49
5.7 DASDCONV (DASD image file conversion program).....	53
5.8 DASDCOPY (Copy DASD file to another DASD file) .....	55
5.9 DASDINIT (DASD image file creation).....	57
5.10 DASDISUP (Fix XCTL tables in SVCLIB) .....	60
5.11 DASDLOAD (DASD loader program).....	62
5.12 DASDLS (List datasets on a volume).....	70

5.13	DASDPDSU (PDS unload utility).....	74
5.14	DASDSEQ (Display sequential datasets) .....	76
5.15	HETGET (Extract files from an AWS or HET tape file) .....	78
5.16	HETINIT (Initialize an AWS or HET tape file).....	80
5.17	HETMAP (Show information about a HET or AWS tape file).....	82
5.18	HETUPD (Update and/or copy an AWS or HET tape file) .....	87
5.19	TAPECOPY (Copy a SCSI tape to or from an AWSTAPE disk file) .....	89
5.20	TAPEMAP (Show information about an AWS tape file) .....	91
5.21	TAPESPLT (Split an AWS tape file).....	93
5.22	DMAP2HRC (P/390 DEVMAP conversion program) .....	95
5.23	VMFPLC2 (VM formatted tape utility) .....	96
Part III: Third-Party Utilities.....		99
6.	Third Party Utilities .....	100
6.1	FTAPE (SCSI Tape Test Utility).....	100
6.2	HercPrt (Remote Hercules Printer Spooler) .....	108
6.3	PRTPUB (Mainframe Print Publishing Pprogram) .....	125
Part IV: Mainframe Utilities .....		129
7.	Mainframe Utilities.....	130
7.1	AWSSL (AWS virtual tape) .....	130
7.2	AWSUTIL (AWS format tape file generation utility) .....	135
7.3	RAWSTAPE (Reverse AWSTAPE utility) .....	137
7.4	VTTS (Virtual tape transportation system).....	138
Appendix A: Supported DASD Device Types.....		141
Appendix B. Syntax .....		144
B.1	Reading Syntax Descriptions .....	144
B.2	Reading Syntax Diagrams.....	145
B.3	Sample Syntax Description .....	147
B.4	Sample Syntax Diagram.....	148
Appendix C. Links.....		149

---

## Figures

Figure 1: Hercules Console Panel.....	14
Figure 2: Console Panel - Window Title .....	15
Figure 3: Console Panel – Log Area .....	16
Figure 4: Console Panel – Command Line .....	16
Figure 5: Console Panel – Status Line .....	17
Figure 6: Hercules Device and Status Panel.....	18
Figure 7: Hercules Help Panel .....	19
Figure 8: Device and Status Panel – Title Line .....	20
Figure 9: Device and Status Panel – Register Area.....	20
Figure 10: Device and Status Panel – Operator Area.....	21
Figure 11: Device and Status Panel – CPU Utilization Bar.....	22
Figure 12: Device and Status Panel – Peripherals Area.....	23
Figure 13: Web Browser Interface – Logon.....	24
Figure 14: Web Browser Interface – Main Screen .....	25
Figure 15: Web Browser Interface – System Log.....	26
Figure 16: Web Browser Interface – IPL .....	27
Figure 17: Web Browser Interface – Register Selection .....	27
Figure 18: Web Browser Interface – Registers Details .....	28
Figure 19: Web Browser Interface – Storage .....	29
Figure 20: Web Browser Interface – Miscellaneous.....	30
Figure 21: Web Browser Interface – Devices.....	31
Figure 22: Web Browser Interface – Subchannel Details .....	32
Figure 23: Web Browser Interface – Version Info .....	33
Figure 24: Web Browser Interface – CPU .....	33
Figure 25: Web Browser Interface – GPRs .....	34
Figure 26: Web Browser Interface – CRs.....	34
Figure 27: Web Browser Interface – PSW .....	35
Figure 28: Web Browser Interface – Documentation .....	35
Figure 29: CCKDCDSK utility output.....	42
Figure 30: CCKDCOMP utility output .....	44
Figure 31: CCKDDIAG utility output.....	47
Figure 32: CCKDSWAP utility output .....	48
Figure 33: DASDCAT utility output (create PDS member list) .....	50
Figure 34: DASDCAT utility output (list a member from a PDS) .....	51
Figure 35: DASDCAT utility output (list all members of a PDS).....	52
Figure 36: DASDCONV utility output.....	54
Figure 37: DASDCOPY utility output.....	56
Figure 38: DASDINIT utility output .....	59
Figure 39: DASDISUP utility output.....	61
Figure 40: DASDLOAD utility output (XMIT method, message level 1) .....	67
Figure 41: DASDLOAD utility output (XMIT method, message level 3).....	68
Figure 42: DASDLOAD utility output (SEQ method, message level 5).....	69
Figure 43: DASDLS utility output (extended information) .....	72
Figure 44: DASDLS utility output (extended information) .....	73

Figure 45: DASDPDSU utility output .....	75
Figure 46: DASDSEQ utility output .....	77
Figure 47: HETGET utility output .....	79
Figure 48: HETINIT utility output .....	81
Figure 49: HETMAP utility output (print all label and file information).....	85
Figure 50: HETMAP utility output (print only dataset information) .....	85
Figure 51: HETMAP utility output (TAPEMAP compatible output).....	86
Figure 52: HETUPD utility output .....	88
Figure 53: TAPECOPY utility output .....	90
Figure 54: TAPEMAP utility output (map AWS tape file) .....	92
Figure 55: TAPESPLT utility output.....	94
Figure 56: FTAPE utility output (display status) .....	107
Figure 57: HercPrt Program Options Page .....	109
Figure 58: HercPrt PDF Options Page .....	111
Figure 59: Job Seperator Control File Sample .....	123
Figure 60: HercPrt Report Sample .....	124
Figure 61: AWSSL JCL (Create AWS virtual tape) .....	133
Figure 62: AWSSL JCL (Create datasets from AWS virtual tape) .....	134
Figure 63: AWSUTIL utility JCL.....	136
Figure 64: RAWSTAPE utility JCL .....	137
Figure 65: IEBGENER JCL for RAWSTAPE utility.....	137
Figure 66: VTT2DISK utility JCL. ....	139
Figure 67: VTT2TAPE utility JCL .....	139
Figure 68: VTT2CNVU utility JCL.....	140
Figure 69: Sample Syntax Description .....	148
Figure 70: Sample Syntax Diagram .....	148

---

## Tables

Table 1: DASD image file maintenance utilites .....	38
Table 2: TAPE file maintenance utilities .....	39
Table 3: Miscellaneous utilities.....	39
Table 4: Third party utilities .....	39
Table 5: Mainframe utilities.....	40
Table 6: Sample Control Files .....	114
Table 7: HercPrt PDF and RTF Options.....	118
Table 8: HercPrt ASA Control Characters and ASCII Equivalentts.....	121
Table 9: Supported CKD DASD Devices .....	142
Table 10: Supported FBA DASD Devices .....	143
Table 11: Reading Syntax Descriptions .....	144
Table 12: Reading Syntax Diagrams.....	147

---

# 1. Preface

---

## 1.1 Edition information

This edition applies to the Hercules S/370, ESA/390 and z/Architecture Emulator, Release 4.00.0 and to all subsequent versions, releases and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of software you are using.

---

## 1.2 What this book is about

This book is a guide for using and operating the Hercules Emulator and Hercules utilities as well as third-party utilities. For guidance in installation of Hercules, for a general overview or for reference information, additional manuals are available.

Please note that some information can be found in more than one manual. This redundancy is not intended to unnecessarily expand the manuals, rather to help find all necessary information in one place.

---

## 1.3 Who should read this book

This book is mainly intended for people who are responsible for operating the Hercules Emulator. It may also be useful if you are responsible for installing the Hercules Emulator.

---

## 1.4 What you need to know to understand this book

To understand this book, you should be familiar using software under the Linux, Microsoft Windows or Mac OS X operating systems. You should also have experience with Linux command shells or native DOS (Microsoft Disk Operating System) and the Microsoft Windows command shell.

Last but not least you should be familiar with IBM mainframe environments (hardware and software) and the underlying ideas and concepts as Hercules emulates IBM mainframe hardware.

---

## 1.5 How to use this book

This book is designed as a reference for all utilities related to the Hercules Emulator. You can go through the book chapter by chapter or you can use the book as a reference for all questions regarding the Hercules utilities.

---

## 1.6 Revision Notice

Hercules Release:	Version 4 Release 00 Modification 0
Publication Number:	HEOP040000
SoftCopy Name:	HerculesOperationsandUtilities
Revision Number:	HEOP040000-00
Date:	November 21, 2015

---

## 1.7 Readers Comments

If you like or dislike anything of this book please send a mail or email to the address below. Feel free to comment any errors or lack of clarity. Please limit your comments on the information in this specific book and also include the "Revision Notice" just above. Thank you for your help.

Send your comments by email to the Hercules-390 discussion group:

[hercules-390@yahoogroups.com](mailto:hercules-390@yahoogroups.com)

---

## 1.8 Legal Advice

Hercules implements only the raw S/370, ESA/390, and z/Architecture instruction set, it does not provide any operating system facilities. This means that you need to provide an operating system or standalone program which Hercules can load from an emulated disk or tape device. You will have to write the operating system or standalone program yourself unless you possess a license from IBM to run one of their operating systems on your PC or use IBM programs and operating systems which have been placed in the public domain.

***NOTE: It is YOUR responsibility to comply with the terms of the license for the operating system you intend to run on the Hercules Emulator.***

---

## 1.9 Trademarks

The following is a list of trademark acknowledgments and copyright notices for product and company names mentioned in this book. Other product and company names in this book that are not listed below may be the trademarks or registered trademarks of their respective owners.

- IBM, System/370, ESA/390, z/Architecture, MVS, OS/390, z/OS, VM, VM/ESA, z/VM, VSE, VSE/ESA, z/VSE are trademarks or registered trademarks of International Business Machines Corporation (IBM).
- Windows XP, Windows Vista, Windows 7, Windows Server 2003, Windows Server 2008, Visual C++ Toolkit, Visual C++ Express are trademarks of Microsoft Corporation.
- Linux is a trademark owned by Linus Torvalds. The Linux Mark Institute is the exclusive licensor of the Linux trademark on behalf of its owner Linus Torvalds.
- WinPcap is copyrighted by NetGroup, Politecnico di Torino (Italy).
- Cygwin is copyrighted by Red Hat, Inc.
- Vista tn3270 is copyrighted by Tom Brennan Software.
- Pentium, XEON are trademarks or registered trademarks of Intel Corporation.
- Athlon, Opteron are trademarks or registered trademarks of Advanced Micro Devices (AMD), Inc.
- Xmit Manager is copyrighted by Neal Johnston-Ward.
- FLEX-ES is a registered trademark of Fundamental Software, Inc.

---

## 1.10 Acknowledgements

The Hercules manuals would not have been possible without the assistance of many people and I would like to thank all those who helped me. In particular I would like to thank:

- The Hercules developers for their documentation on various websites from which I derived a great deal of information.
- Roger Bowler and Fish for proof-reading the manuals.
- Loris Degoianni for allowing me to use parts of the original WinPcap documentation.
- Tom Brennan for allowing me to use parts of his Vista tn3270 documentation.
- My colleagues for working with early previews of the documentation, beginning with just a few pages.
- Mike Cairns for reviewing and editing the manuals.
- Robert Allan for providing the “Linux Installation” part.
- Lutz Mader for providing the “Mac OS X Installation” part.

If anyone feels they have been forgotten on this list please let me know.

Peter Glanzmann

---

## 2. Related Publications

---

### 2.1 Hercules – General Information

The Hercules “General Information” manual provides you an overview of the ideas and concepts of the Hercules Emulator as well as a documentation of the emulator’s functionality. It explains what Hercules does and does not do. It helps you decide if the software fits your needs fulfills your requirements.

---

### 2.2 Hercules – Installation Guide

The Hercules "Installation Guide" shows you how to install Hercules and all related optional and required software components under the Microsoft Windows, Linux and Apple Macintosh OS X operating systems.

After going through the installation guide you will have a working emulator environment ready to IPL a S370, S/390 or z/Architecture mainframe operating system.

---

### 2.3 Hercules – User Reference Guide

The Hercules “User Reference” leads you through all aspects of the emulator’s operation. It provides instruction in the operation of the Hercules Emulator with and without the Hercules GUIs. The usage details for the utilities are covered in the “Hercules Utilities” guide.

After reading this manual you should be able to work with Hercules, to create a configuration file and to use Hercules commands through the console.

---

### 2.4 Hercules – Operations and Utilities Guide

The Hercules “Operations and Utilities Guide” describes the operation of the Hercules Emulator and the usage of additional utilities that are delivered together with the Hercules. Selected utilities from third-party suppliers are also covered in this manual.

After reading this manual you should have the knowledge on how to work with Hercules and to use the right utility for a certain housekeeping task within the Hercules environment. You should also be able to create virtual devices and understand backup / restore procedures.

---

### 2.5 Hercules – Messages and Codes

The “Messages and Codes” manual provides a detailed explanation of all Hercules related messages. It is the primary source for troubleshooting and debugging if you experience problems with Hercules.

---

### 2.6 Hercules – Reference Summary

The Hercules "Reference Summary" booklet lists all the system parameters, device definitions, console commands, Hercules utilities etc. along with their arguments.

This booklet is intended as a quick reference guide for experienced users. Consult the Hercules "User Reference Guide" and “Utilities Guide” for more detailed and additional information.

---

## 3. Summary of changes

---

### 3.1 Version 4, First Edition (HEOP040000-00)

This section describes the various changes that have been made in the “Operations and Utilities Guide”. Because this is a new manual it has no previous editions. The most significant additions made in this edition of the manual are the following:

- Chapter 4 (Hercules Operations) added.
- Chapter 5 (Hercules Supplied Utilities) moved from the “User Reference Guide” to this manual.
- Chapter 6 (Third Party Utilities) moved from the “User Reference Guide” to this manual.
- Chapter 6 (Third Party Utilities): Section “HercPrt (Remote Hercules Printer Spooler)” added.
- Chapter 6 (Third Party Utilities): Section “FTAPE (SCSI Tape Test Utility)” added.
- Chapter 7 (Mainframe Utilities) moved from the “User Reference Guide” to this manual.
- Appendix C. Links: List of links added.

---

## Part I: Hercules Operations

---

## 4. Hercules Operations

---

### 4.1 Starting the Emulator

When the Hercules Emulator is started it expects a configuration file to build the hardware configuration. The name of this file can be provided in different ways.

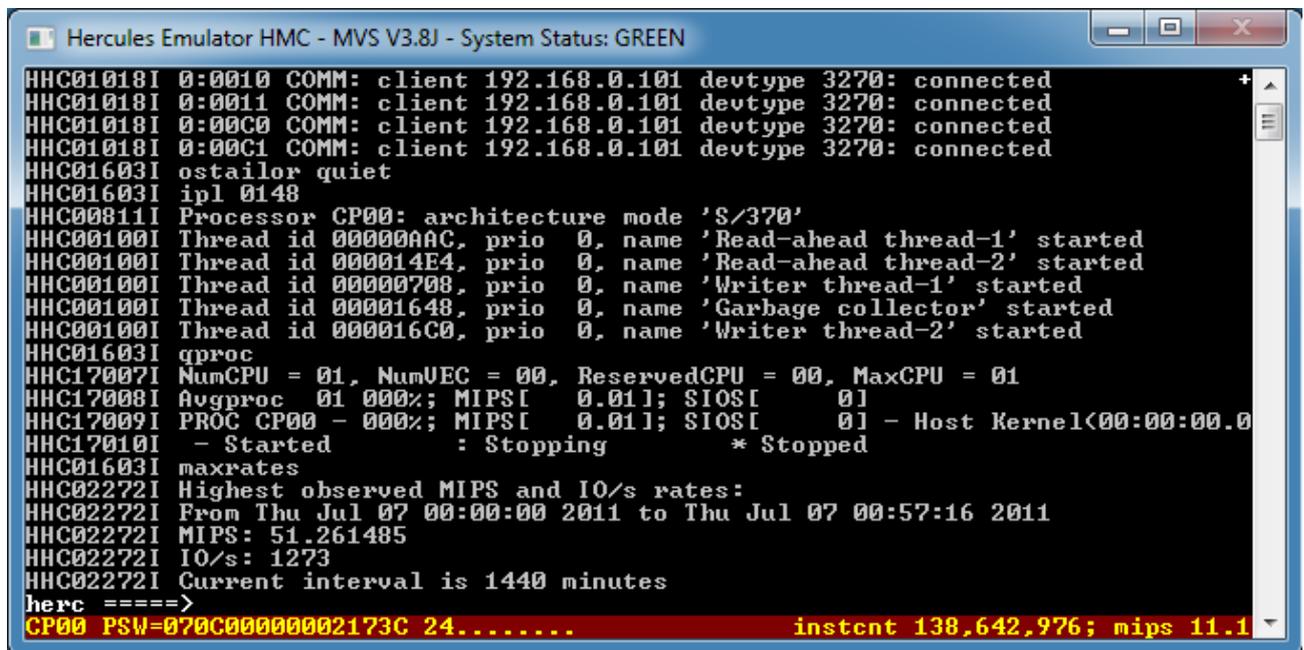
- If Hercules is started with option “-f *configfile*”, where *configfile* specifies the name (and optionally the path) of the desired configuration, then this file is used to build the configuration.
- If there is an environment variable called “HERCULES\_CNF” defined, which contains the name (and optionally the path) of a configuration file, then this file is used to build the configuration.
- Finally Hercules looks for a file with the default name “HERCULES.CNF” in the current directory where Hercules was run from.

If a valid configuration file could be found then Hercules builds the hardware configuration based on this file. If however no valid configuration file could be found then message HHC01432S is issued and Hercules is immediately terminated.

---

### 4.2 The Console Panel

After starting the emulator you are presented with the Hercules console panel. This panel displays the log and some status information and is used to enter Hercules console commands.



```
Hercules Emulator HMC - MVS V3.8J - System Status: GREEN
HHC01018I 0:0010 COMM: client 192.168.0.101 devtype 3270: connected
HHC01018I 0:0011 COMM: client 192.168.0.101 devtype 3270: connected
HHC01018I 0:00C0 COMM: client 192.168.0.101 devtype 3270: connected
HHC01018I 0:00C1 COMM: client 192.168.0.101 devtype 3270: connected
HHC01603I ostarlor quiet
HHC01603I ipl 0148
HHC00811I Processor CP00: architecture mode 'S/370'
HHC00100I Thread id 00000AAC, prio 0, name 'Read-ahead thread-1' started
HHC00100I Thread id 000014E4, prio 0, name 'Read-ahead thread-2' started
HHC00100I Thread id 00000708, prio 0, name 'Writer thread-1' started
HHC00100I Thread id 00001648, prio 0, name 'Garbage collector' started
HHC00100I Thread id 000016C0, prio 0, name 'Writer thread-2' started
HHC01603I qproc
HHC17007I NumCPU = 01, NumVEC = 00, ReservedCPU = 00, MaxCPU = 01
HHC17008I Augproc 01 000%; MIPS[ 0.01]; SIOS[ 0]
HHC17009I PROC CP00 - 000%; MIPS[ 0.01]; SIOS[ 0] - Host Kernel<00:00:00.0
HHC17010I - Started : Stopping * Stopped
HHC01603I maxrates
HHC02272I Highest observed MIPS and IO/s rates:
HHC02272I From Thu Jul 07 00:00:00 2011 to Thu Jul 07 00:57:16 2011
HHC02272I MIPS: 51.261485
HHC02272I IO/s: 1273
HHC02272I Current interval is 1440 minutes
herc =====>
CP00 PSW=070C00000002173C 24..... instcnt 138,642,976; mips 11.1
```

Figure 1: Hercules Console Panel

The console panel can be divided into four different areas:

- Window title
- Log area

- Command line
- Status line

Each of these areas is described in more detail in the following sections.

### 4.2.1 Window Title

The default window title is a string consisting of the following information:

```
"LPARNAME - SYSTYPE * SYSNAME * SYSPLEX - System Status: colour"
```

LPARNAME is the value taken from the correspondent system parameter or console command. SYSTYPE, SYSNAME and SYSPLEX are populated by the system call SCLP Control Program Identification. If any of these values is blank, then that field is not presented in the window title. The system status can either be green, amber or red and has the following meanings:

- GREEN            Everything is working correctly.
- AMBER           One or more CPUs are not running.
- RED              One or more CPUs are in wait state.

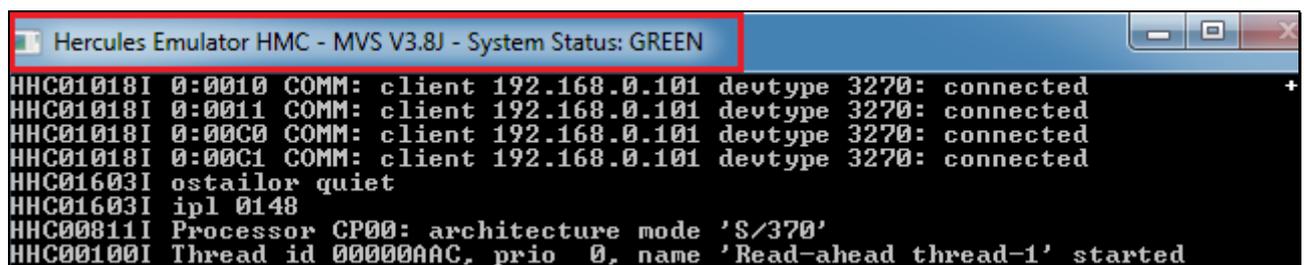


Figure 2: Console Panel - Window Title

The title of the Hercules window can be customized through the PANTITLE system parameter or console command. To create the window title shown in the figure above the following PANTITLE system parameter was used:

```
PANTITLE "Hercules Emulator HMC - MVS V3.8J"
```

The window title can also dynamically be changed through the PANTITLE console command which looks the same as the system parameter shown above.

### 4.2.2 Log Area

The log area displays Hercules' log output. This output can also be written to a log file, if requested. In the log area all Hercules messages are displayed, as well as every command entered and the response returned from Hercules.

This area may also show the system console output of the operating system running under Hercules, if there is no system console defined in the configuration file and the operating system console output is routed to the Hercules console instead.

```

Hercules Emulator HMC - MVS V3.8J - System Status: GREEN
HHC01018I 0:0010 COMM: client 192.168.0.101 devtype 3270: connected
HHC01018I 0:0011 COMM: client 192.168.0.101 devtype 3270: connected
HHC01018I 0:00C0 COMM: client 192.168.0.101 devtype 3270: connected
HHC01018I 0:00C1 COMM: client 192.168.0.101 devtype 3270: connected
HHC01603I oostailor quiet
HHC01603I ipl 0148
HHC00811I Processor CP00: architecture mode 'S/370'
HHC00100I Thread id 00000AAC, prio 0, name 'Read-ahead thread-1' started
HHC00100I Thread id 000014E4, prio 0, name 'Read-ahead thread-2' started
HHC00100I Thread id 00000708, prio 0, name 'Writer thread-1' started
HHC00100I Thread id 00001648, prio 0, name 'Garbage collector' started
HHC00100I Thread id 000016C0, prio 0, name 'Writer thread-2' started

```

Figure 3: Console Panel – Log Area

The amount of messages that are written to the log is customizable. Using the OSTAILOR system parameter or console command, messages that are considered normal for the type of operating system you are running can be suppressed.

The additional PGMTRACE console command allows further to fine tune the tracing of program interrupt exceptions that are displayed on the console. Finally the MSGLVL system parameter or console command can be used to set the level of displayed messages.

Error messages are displayed in red and are held at the top of the log window until they are cleared with a MSGHLD CLEAR or KD console command or the timeout for held messages expires. The default timeout value for held messages is two minutes and can be set or changed through the MSGHLD system parameter or console command.

### 4.2.3 Command Line

At the bottom of the log area is the command line (“herc =====>”). Here you can enter Hercules console commands or SCP / PSCP commands (commands intended for the operating system running under Hercules).

```

HHC01603I qproc
HHC17007I NumCPU = 01, NumVEC = 00, ReservedCPU = 00, MaxCPU = 01
HHC17008I Avgproc 01 000%; MIPS[ 0.011; SIOS[ 01
HHC17009I PROC CP00 - 000%; MIPS[ 0.011; SIOS[ 01 - Host Kernel<00:00:00.0
HHC17010I - Started : Stopping * Stopped
HHC01603I maxrates
HHC02272I Highest observed MIPS and IO/s rates:
HHC02272I From Thu Jul 07 00:00:00 2011 to Thu Jul 07 00:57:16 2011
HHC02272I MIPS: 51.261485
HHC02272I IO/s: 1273
HHC02272I Current interval is 1440 minutes
herc =====>
CP00 PSW=070C000000002173C 24..... instcnt 138,642,976; mips 11.1

```

Figure 4: Console Panel – Command Line

Given without prefix the entered command is assumed to be a Hercules command. Prefixed with a period (dot) or an exclamation mark the command is assumed to be a SCP reply or a SCP priority message respectively.

If you are mainly working with Hercules commands then this behavior is fine. However if you are entering a lot of SCP or PSCP commands then every time prefixing the command with a period or exclamation

mark can be somewhat cumbersome. In this case you can change the target for entered commands through the CMDTGT console command. As per default the command target is Hercules.

#### 4.2.4 Status Line

The status line shows on the left the CPU that is the current target for console commands. The target CPU can be changed through the “CPU” console command. Beneath the CPU information the status line displays the current PSW of the target CPU.

On the right side of the status line the overall instruction count is displayed together with the actual MIPS rate. The status line is refreshed at a fixed rate of 1000 milliseconds (once per second). This rate is independent of the PANRATE setting.

```
HHC01603I qproc
HHC17007I NumCPU = 01, NumUEC = 00, ReservedCPU = 00, MaxCPU = 01
HHC17008I Avgproc 01 000%; MIPS[ 0.011; SIOS[ 0]
HHC17009I PROC CP00 - 000%; MIPS[ 0.011; SIOS[ 0] - Host Kernel<00:00:00.0
HHC17010I - Started : Stopping * Stopped
HHC01603I maxrates
HHC02272I Highest observed MIPS and IO/s rates:
HHC02272I From Thu Jul 07 00:00:00 2011 to Thu Jul 07 00:57:16 2011
HHC02272I MIPS: 51.261485
HHC02272I IO/s: 1273
HHC02272I Current interval is 1440 minutes
here =====>
CP00 PSW=070C00000002173C 24..... instcnt 138,642,976; mips 11.1
```

Figure 5: Console Panel – Status Line

## 4.3 The Device and Status Panel

Pressing “Escape” (ESC) switches between the console panel and the device and status panel. This panel shows the current configuration, register status information, the CPU state and utilization and is used to perform different operator actions.

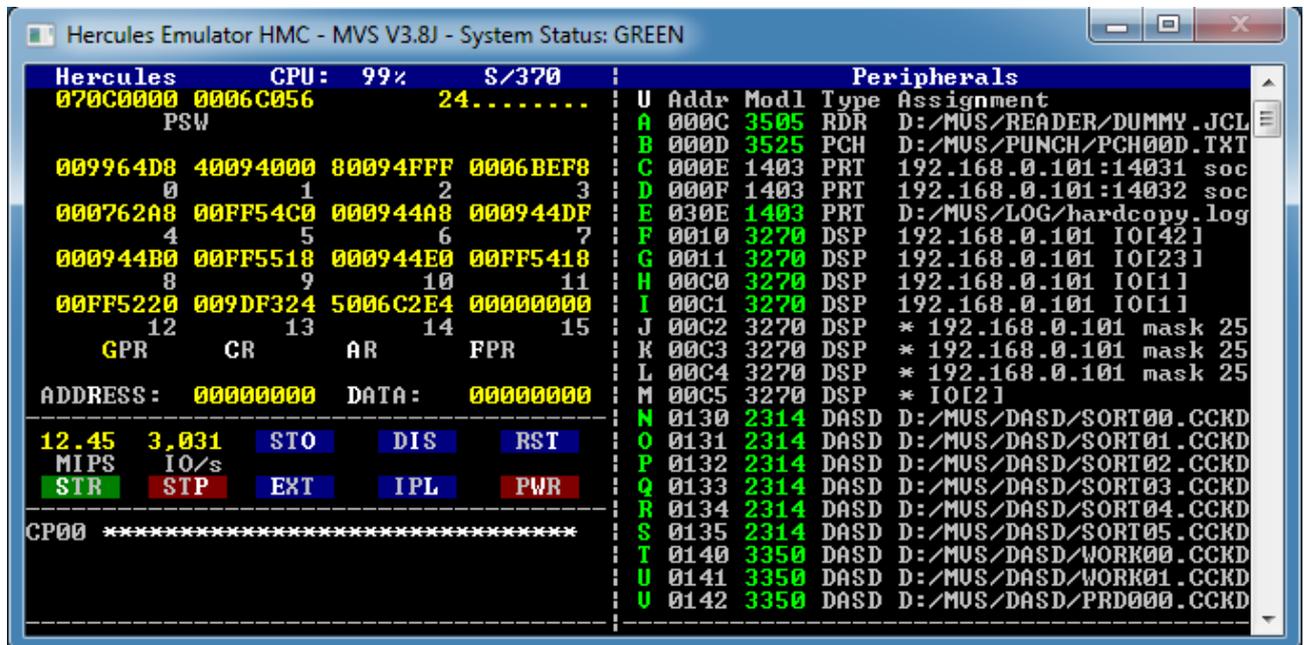


Figure 6: Hercules Device and Status Panel

The device and status panel can be divided in these areas:

- Window title
- Status panel heading
- Register area
- Operator area
- CPU utilization bar
- Peripherals area

All commands in the device and status panel consist of one character key controls. The various keys are highlighted by bright white versus the grey of the other letters. The following is an overview of all available keys and their related commands. They are described also in the respective areas sections where they belong to.

- Display Controls**
- G** - General purpose registers
  - C** - Control registers
  - A** - Access registers
  - F** - Floating point registers
  - I** - Display storage at 'address'

- CPU Controls**
- L - IPL
  - S - Start CPU
  - E - External interrupt
  - P - Stop CPU
  - W - Power down (exit Hercules)
  - T - Restart interrupt
- Storage Update**
- R - Enter address to be updated
  - D - Enter data to be updated at address
  - O - Place data value at address
- Peripherals**
- N - Enter new name for the device
  - U - Send an I/O attention interrupt

Pressing “?” displays a help page that explains briefly all the shortcuts available in the device and status panel. Pressing “ESC” again leaves the help page and returns to the device and status panel.

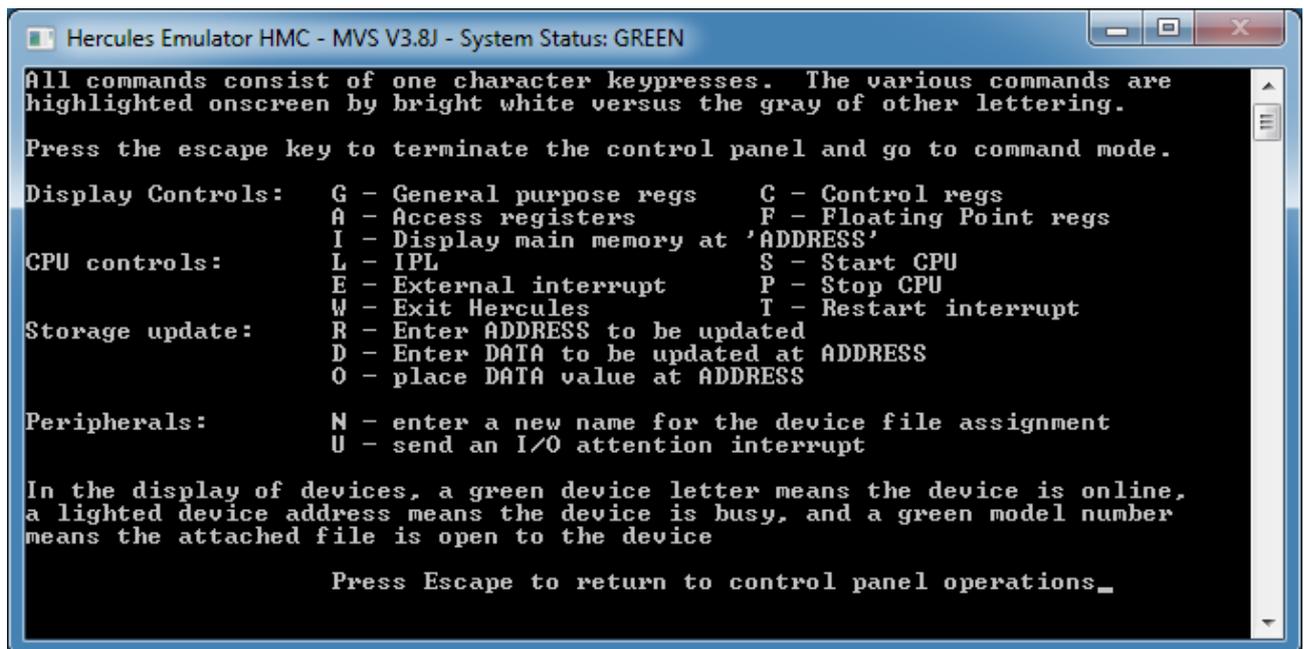


Figure 7: Hercules Help Panel

### 4.3.1 Window Title

The window title is the same as described under section 4.2.1. There is no possibility to have different window titles for the panels of the Hercules emulator.

### 4.3.2 Status Panel Heading

The heading of the device and status panel is the small blue line just below the window title.

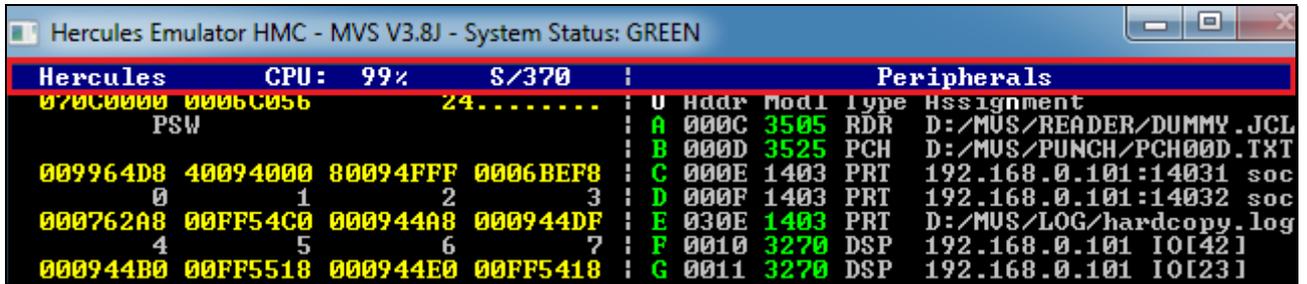


Figure 8: Device and Status Panel – Title Line

The heading consists of the following information:

Hercules - CPU: *nnn%* - *architecture mode* - Peripherals

“Hercules” and “Peripherals” is fix text, “CPU *nnn%*” displays the overall CPU utilization of the host system. The “*architecture mode*” field displays the current architecture mode as specified with the ARCHLVL (or ARCHMODE) system parameter or console command or the architecture mode that the operating system has switched to after the IPL.

### 4.3.3 Register Area

On top of the register area the current content of the Program Status Word (PSW) is displayed. The PSW cannot be altered from the device and status panel. To alter the individual fields of the PSW the corresponding “PSW” console command must be used after switching to the console panel.

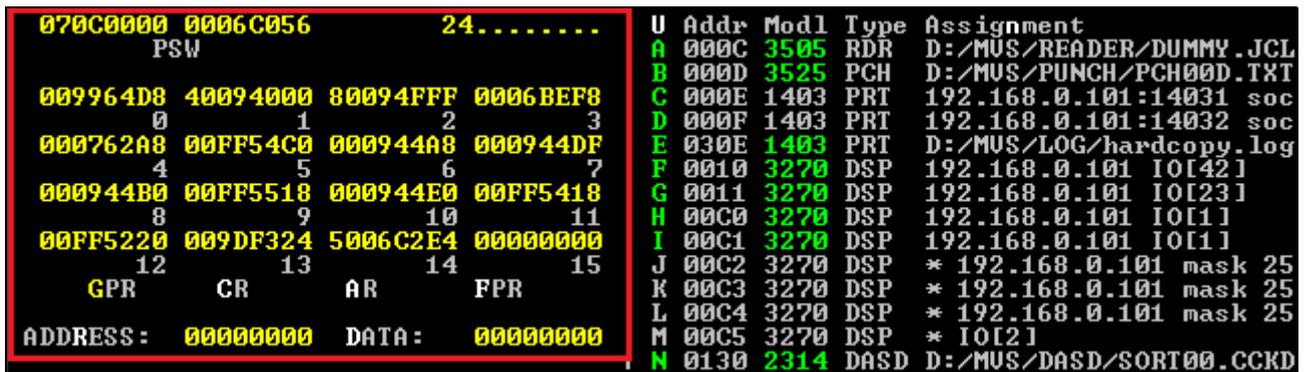


Figure 9: Device and Status Panel – Register Area

Below the PSW the registers are shown. The type of registers that are displayed can be selected through pressing the first (highlighted) letter of their abbreviation as shown in the description line directly below the registers:

- GPR General Purpose Registers

- CR Control Registers
- AR Access Registers
- FPR Floating Point Registers

The highlighted yellow letter indicates the currently displayed register type. The register types that are selectable are marked with a highlighted white letter.

At the end of the register area there are two fields that are used to alter storage (address and data). They are selectable through the highlighted white letters “R” and “D”. Pressing “R” allows it to specify the storage address to be updated. The default address is x’00000000’. Pressing “D” allows it to enter the data to be updated at the specified address. To place the new data value at the specified address press “O”.

The display refresh rate of the register area is depending on the PANRATE setting. This may be anything from 1 to 5000 milliseconds (between 0.001 and 5 seconds). The default refresh rate is 500 milliseconds. Please note that a fast refresh rate may negatively impact the performance, beneath the fact you won’t be able to read anything in the register area.

### 4.3.4 Operator Area

The operator area displays the current Hercules MIPS and I/O rates and has eight controls that can be selected through the highlighted letter of their respective label. These controls are:

- STO Place data value at address
- DIS Display main memory
- RST Restart interrupt
- STR Start all CPUs
- STP Stop all CPUs
- EXT Generate external interrupt
- IPL Initial Program Load
- PWR Power off Hercules



Figure 10: Device and Status Panel – Operator Area

For several of these controls you will be prompted to confirm the function. IPL requires specifying a device through its prefixed letter as shown in the peripherals area. If the IPL device is not within the first 26 devices then it has no letter assigned and the IPL must be performed from the console panel through the regular “IPL” console command.

### 4.3.5 CPU Utilization Bar

The CPU utilization bar shows one line per emulated CPU as specified in the configuration file. On the left there is the CPU number, on the right there is a utilization bar that shows the current CPU utilization. If a CPU is in the stopped state then the utilization bar displays “CPnn STOPPED”.



Figure 11: Device and Status Panel – CPU Utilization Bar

If capping is active (see the CAPPING system parameter or console command) and the CPU is currently capped then the utilization bar turns its color from white to red, as long as the CPU is in the capped state.

The CPU bar is refreshed at a fixed interval of 1000 milliseconds (once per second), independent of the PANRATE setting.

### 4.3.6 Peripherals Area

The peripherals area contains a list of all the devices of the current configuration. It is mostly the same as the device definition part of the configuration file. In the peripherals area however the devices are sorted by the device address and all symbols that may have been used in the configuration file are resolved. In addition the current number of IOs for each device is shown at the end of the assignment data column.

The peripherals area displays following information for each device:

**U** The “U” column contains a one letter shortcut for the device address (A-Z, beginning with the device with the smallest device address) for the first 26 devices of the peripherals list.

This alias letter can be used as shortcut when initiating an IPL from the device and status panel or when sending an I/O interrupt to a device. If the intended device is not within the first 26 devices of this list, then the shortcut can’t be used and the IPL or the I/O interrupt has to be performed with a console command from the console panel. A green device letter instead of a grey one indicates that the device is on-line.

**Addr** The “addr” column contains the device address as specified in the configuration file. Independent of the sequence in the configuration file the devices are sorted in ascending order in the peripherals list. A lighted device address indicates that the device is busy.

**Modl** This is the device type as defined in the configuration file. A green device type instead of a grey one indicates that the attached file is open to the device.

**Type** This describes the device class. Possible device classes are:

- CON (Console devices)
- CTCA (Channel-to-channel adapter)

- DASD (Disk devices)
- DSP (Terminals)
- LINE (Communication lines)
- PCH (Card punch devices)
- PRT (Printer devices)
- QETH (QETH devices)
- RDR (Card reader)
- TAPE (Tape drives)

**Assignment** This is the assignment made in the configuration file for the device. If any symbols have been used in the device definition then the assignment column displays the resolved value. At the end of each device assignment line the current number of I/Os to this device are shown in brackets.

Hercules				CPU: 99%	S/370	Peripherals			
070C0000	0006C056	PSW		24	U	Addr	Modl	Type	Assignment
009964D8	40094000	80094FFF	0006BEF8		A	000C	3505	RDR	D:/MUS/READER/DUMMY.JCI
000762A8	00FF54C0	000944A8	000944DF		B	000D	3525	PCH	D:/MUS/PUNCH/PCH00D.TXT
000944B0	00FF5518	000944E0	00FF5418		C	000E	1403	PRT	192.168.0.101:14031 soc
00FF5220	009DF324	5006C2E4	00000000		D	000F	1403	PRT	192.168.0.101:14032 soc
	GPR	CR	AR	FPR	E	030E	1403	PRT	D:/MUS/LOG/hardcopy.log
	12	13	14	15	F	0010	3270	DSP	192.168.0.101 IO[42]
					G	0011	3270	DSP	192.168.0.101 IO[23]
					H	00C0	3270	DSP	192.168.0.101 IO[1]
					I	00C1	3270	DSP	192.168.0.101 IO[1]
					J	00C2	3270	DSP	* 192.168.0.101 mask 25
					K	00C3	3270	DSP	* 192.168.0.101 mask 25
					L	00C4	3270	DSP	* 192.168.0.101 mask 25
					M	00C5	3270	DSP	* IO[2]
ADDRESS:	00000000	DATA:	00000000						

Figure 12: Device and Status Panel – Peripherals Area

In the peripherals area the following controls are available through pressing the highlighted letter:

- U Send an I/O interrupt to a device.
- N Enter a new name for the device file assignment.

---

## 4.4 The Web Browser Interface

Hercules can also be accessed through a web interface in addition to the application panels. This allows operating the emulator from a totally different machine than it is running on. The web server component is activated through HTTP statements in the configuration file or HTTP console commands (consult the “User Reference” manual for details on how to activate the built in web server). To access the web browser interface, start a web browser pointing to the machine where Hercules is running on. The URL is dependent on the port chosen on the HTTP statement to activate the web server:

```
http://ip_address      (port = 80)
http://ip_adress:port  (port <> 80)
```

If Hercules is running on the same machine as the web browser, then “localhost” can be used instead of the IP-address. Some valid examples to connect to the web server:

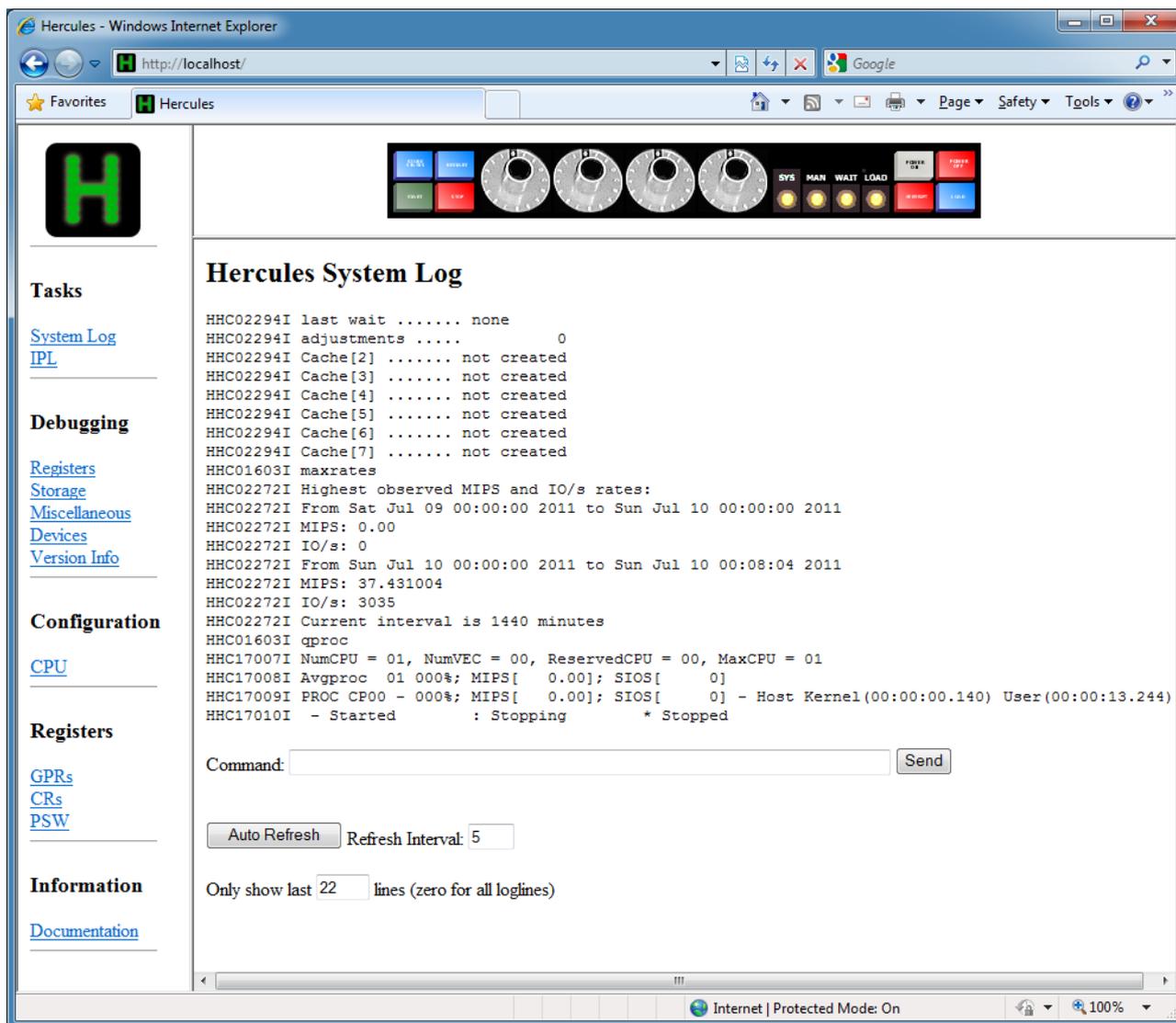
```
http://192.168.0.199
http://10.0.0.1:8080
http://localhost
http://localhost:8081
```

If logon security is enabled in the HTTP statement (through the AUTH parameter) then the user is first prompted to enter a user name and a password before to be able to connect to the web server. Please note that the user name and password necessary to access the web server are sent in an unsecure manner to the web server (no HTTPS is used) and are not saved encrypted in the configuration file.



**Figure 13: Web Browser Interface – Logon**

After accessing the web server the main screen of the web interface is presented. The web browser window is divided in three areas: the navigation area on the left, an operator area on top of the window (similar to the area in the device and status panel) and a display area.



**Figure 14: Web Browser Interface – Main Screen**

The navigation area contains links to all the other views grouped by functionality. Just click on the desired link to show the requested information in the display area. The following views are available:

- System Log (Tasks)
- IPL (Tasks)
- Registers (Debugging)
- Storage (Debugging)
- Miscellaneous (Debugging)
- Devices (Debugging)
- Version Info (Debugging)
- CPU (Configuration)
- GPRs (Registers)
- CRs (Registers)

- PSW (Registers)
- Documentation (Information)

Each of these views is explained in more detail in the following sections.

### 4.4.1 System Log

When accessing the web browser interface the system log is the first view displayed. It shows an excerpt of the current Hercules log.

Per default the last 22 lines of the Hercules log are displayed. There is an entry field in the last line of the log display to change number of lines. The number entered here specifies the number of log lines that are displayed in a scrollable window. Specifying zero allows browsing through all log lines.

## Hercules System Log

```

HHC00890I Facility(ENH_MONITOR           ) Disabled
HHC00890I Facility(LOAD_PROG_PARAM       ) Disabled
HHC00890I Facility(FPS_ENHANCEMENT       ) Disabled
HHC00890I Facility(DECIMAL_FLOAT         ) Disabled
HHC00890I Facility(DFP_HPERF             ) Disabled
HHC00890I Facility(FAST_BCR_SERIAL       ) Disabled
HHC00890I Facility(RES_REF_BITS_MUL      ) Disabled
HHC00890I Facility(ACC_EX_FS_INDIC       ) Disabled
HHC00890I Facility(MSA_EXTENSION_3       ) Disabled
HHC00890I Facility(MSA_EXTENSION_4       ) Disabled
HHC00890I Facility(MOVE_INVERSE         ) Enabled
HHC00890I Facility(MSA_EXTENSION_1       ) Disabled
HHC00890I Facility(MSA_EXTENSION_2       ) Disabled
HHC00890I Facility(PROBSTATE_DIAGF08     ) Disabled
HHC00890I Facility(SIGP_SETARCH_S370     ) Disabled
HHC00890I Facility(HOST_RESOURCE_ACCESS) Disabled
HHC01603I maxrates
HHC02272I Highest observed MIPS and IO/s rates:
HHC02272I From Wed Jul 20 00:00:00 2011 to Wed Jul 20 12:57:19 2011
HHC02272I MIPS: 26.624422
HHC02272I IO/s: 3060
HHC02272I Current interval is 1440 minutes

```

Command:

Refresh Interval:

Only show last  lines (zero for all loglines)

Figure 15: Web Browser Interface – System Log

Through clicking on the “Auto Refresh” button, the view is refreshed automatically. The refresh interval in seconds can be specified in the entry field beneath the “Auto Refresh” button. Once auto refresh is turned on you can stop it by clicking again on the same button, whose description meanwhile has changed to “Stop Refreshing”.

The command entry field can be used to enter any Hercules command. Click on the “Send” button to submit the command in the entry field.

### 4.4.2 IPL

The IPL view is used to perform an Initial Program Load (IPL). You can select the CPU, the device to IPL from and an optional IPL parameter. The IPL parameter is used for the IPL of the intended operating system. The parameter is operating system dependent, consult the relevant operating system documentation for details. Clicking on the “IPL” button performs the Initial Program Load with the chosen values.



Figure 16: Web Browser Interface – IPL

### 4.4.3 Registers

When displaying the registers view, the general purpose registers, the control registers and the access registers are hidden initially. To view the registers click on the according “Select xxxx Registers” button.

When the registers are displayed the button changes to “Hide xxxx Registers”. Click again on this button to hide the selected registers view. Depending on the current architecture mode some of the register sections may not be available.

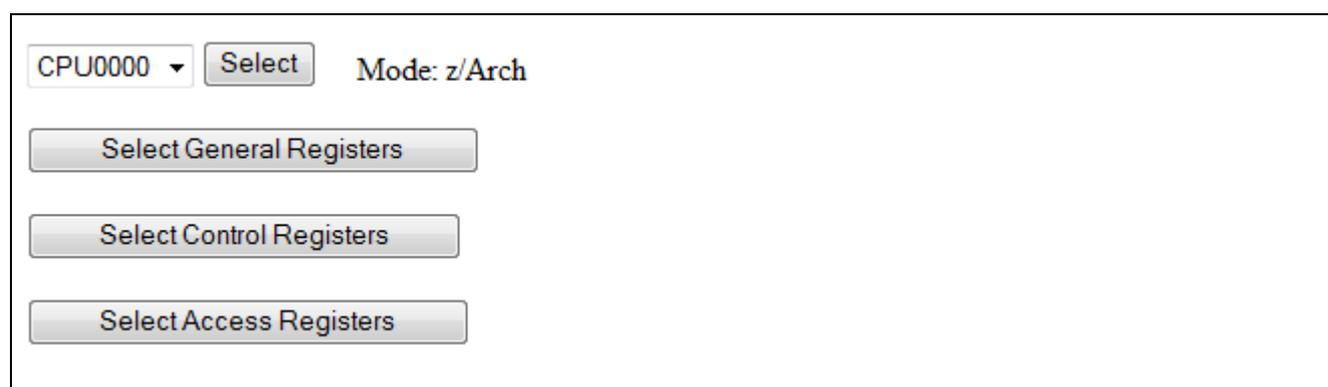


Figure 17: Web Browser Interface – Register Selection

On top there is a drop-down box to select the CPU for which the registers should be displayed. Click on the “Select” button after choosing the CPU from the drop-down list.

After each registers area there is a “Refresh” button to update the values for the selected registers. The values of the other registers are not affected. To alter the contents of a register just overwrite the value in the register box and click on the “Alter” button.

CPU0000 ▾
Select
Mode: S/370

GR0	<input type="text" value="00000000"/>	GR1	<input type="text" value="00000000"/>	GR2	<input type="text" value="00000000"/>	GR3	<input type="text" value="00000000"/>
GR4	<input type="text" value="00000000"/>	GR5	<input type="text" value="00000000"/>	GR6	<input type="text" value="00000000"/>	GR7	<input type="text" value="00000000"/>
GR8	<input type="text" value="00000000"/>	GR9	<input type="text" value="00000000"/>	GR10	<input type="text" value="00000000"/>	GR11	<input type="text" value="00000000"/>
GR12	<input type="text" value="00000000"/>	GR13	<input type="text" value="00000000"/>	GR14	<input type="text" value="00000000"/>	GR15	<input type="text" value="00000000"/>

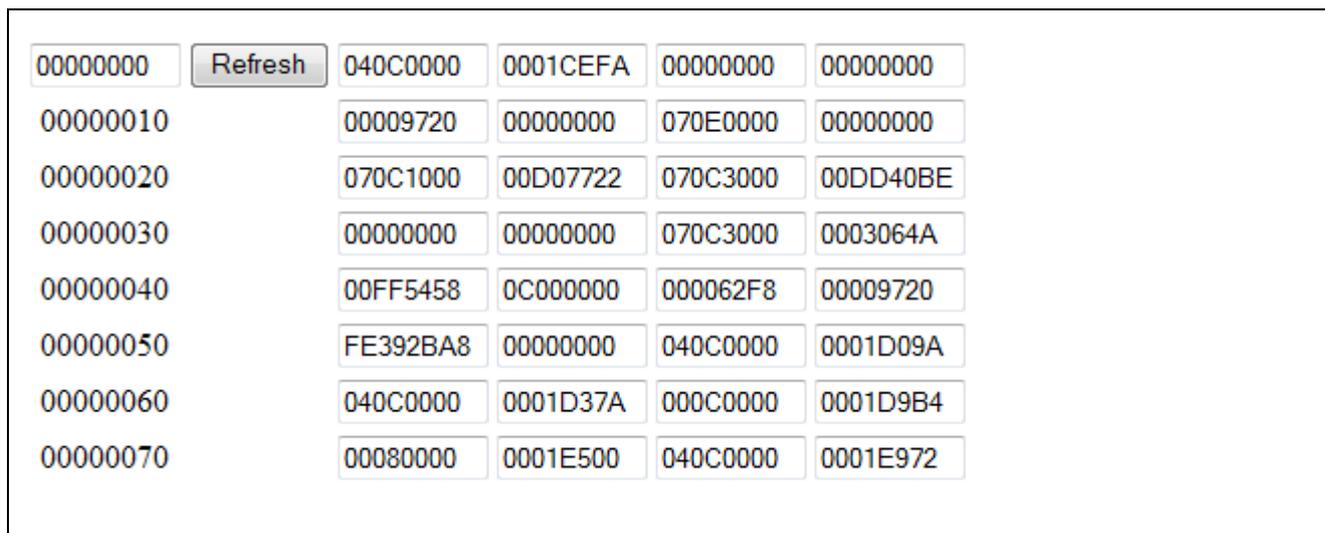
CR0	<input type="text" value="C080EC40"/>	CR1	<input type="text" value="0FFD7C00"/>	CR2	<input type="text" value="FFFFFFFF"/>	CR3	<input type="text" value="00000000"/>
CR4	<input type="text" value="00000000"/>	CR5	<input type="text" value="00000000"/>	CR6	<input type="text" value="00000000"/>	CR7	<input type="text" value="00000000"/>
CR8	<input type="text" value="00000000"/>	CR9	<input type="text" value="00000000"/>	CR10	<input type="text" value="00000000"/>	CR11	<input type="text" value="00000000"/>
CR12	<input type="text" value="00000000"/>	CR13	<input type="text" value="00000000"/>	CR14	<input type="text" value="EFC00000"/>	CR15	<input type="text" value="00FEBB10"/>

**Figure 18: Web Browser Interface – Registers Details**

#### 4.4.4 Storage

The storage view displays contiguous 128 bytes of storage. Each time the storage view is selected it displays storage beginning at address x'00000000'. The first column of each line shows the address in hexadecimal. The next four columns display the storage contents grouped by fullwords (4 bytes).

Between the address and the storage contents is a “Refresh” button. Clicking on this button refreshes the view with the current values.



**Figure 19: Web Browser Interface – Storage**

To view storage from another address in memory the first of the address fields can be overwritten with another value. After clicking the “Refresh” button, the contents of the next 128 bytes following the specified storage address is displayed.

Storage can only be displayed from this view. To alter storage use the “V” console command or the device and status panel.

#### **4.4.5 Miscellaneous**

The miscellaneous view displays various register information. The view is divided into three parts:

- Zone related registers
- Alternate measurement
- Address limit register

All of these values cannot be changed.

## Miscellaneous Registers

### Zone related Registers

Zone	CS Origin	CS Limit	ES Origin	ES Limit	Measurement Block	Key
00	00000000	03FFFFFF	FFF00000	FFFFFFFF	00000000	00
01	00000000	03FFFFFF	FFF00000	FFFFFFFF	00000000	00
02	00000000	03FFFFFF	FFF00000	FFFFFFFF	00000000	00
03	00000000	03FFFFFF	FFF00000	FFFFFFFF	00000000	00
04	00000000	03FFFFFF	FFF00000	FFFFFFFF	00000000	00
05	00000000	03FFFFFF	FFF00000	FFFFFFFF	00000000	00
06	00000000	03FFFFFF	FFF00000	FFFFFFFF	00000000	00
07	00000000	03FFFFFF	FFF00000	FFFFFFFF	00000000	00

### Alternate Measurement

Measurement Block	Key
00000000	00

### Address Limit Register

00000000

Figure 20: Web Browser Interface – Miscellaneous

### 4.4.6 Devices

The devices view displays a list of all devices in the current configuration. The list is sorted by the device address in ascending order. The device list contains the following columns:

- Number**            Number is the device address as specified in the configuration file. Independent of the sequence in the configuration file the devices are sorted by the device address in ascending order.
- Subchannel**        This is the subchannel number the device is attached to. This field contains a link to the subchannel details view (see below).
- Class**              This describes the device class. Possible device classes are:
- CON (Console devices)

- CTCA (Channel-to-channel adapter)
- DASD (Disk devices)
- DSP (Terminals)
- LINE (Communication lines)
- PCH (Card punch devices)
- PRT (Printer devices)
- QETH (QETH devices)
- RDR (Card reader)
- TAPE (Tape drives)

**Type** This is the device type. For a list of device types see the “General Information” or the “User Reference Guide” manuals.

**Status** The status column shows the device status.

<b>Attached Device List</b>				
<b>Number</b>	<b>Subchannel</b>	<b>Class</b>	<b>Type</b>	<b>Status</b>
000C	<a href="#">0000</a>	RDR	3505	
000D	<a href="#">0001</a>	PCH	3525	open
000E	<a href="#">0002</a>	PRT	1403	
000F	<a href="#">0003</a>	PRT	1403	
030E	<a href="#">0004</a>	PRT	1403	open
0010	<a href="#">0005</a>	DSP	3270	open
0011	<a href="#">0006</a>	DSP	3270	open
00C0	<a href="#">0007</a>	DSP	3270	open
00C1	<a href="#">0008</a>	DSP	3270	open
00C2	<a href="#">0009</a>	DSP	3270	
00C3	<a href="#">000A</a>	DSP	3270	
00C4	<a href="#">000B</a>	DSP	3270	
00C5	<a href="#">000C</a>	DSP	3270	
0130	<a href="#">000D</a>	DASD	2314	open
0131	<a href="#">000E</a>	DASD	2314	open
0132	<a href="#">000F</a>	DASD	2314	open
0133	<a href="#">0010</a>	DASD	2314	open

**Figure 21: Web Browser Interface – Devices**

Clicking on the subchannel number in the device list view opens the detailed subchannel information for the specific device. The information presented in the subchannel details is taken from the Subchannel-Information Block (SCHIB). The Subchannel-Information Block contains - beneath other areas - the Path-Management-Control Word (PMCW) and the Subchannel-Status Word (SCSW). The subchannel information shown here displays words 0-6 of the SCHIB (the Path-Management-Control Word). More details can be found in the “DS” console command description in the “User Reference” manual.

Clicking on the “Select / Refresh” button updates the information shown. Selecting another device from the drop-down box and clicking on the “Select / Refresh” button displays the subchannel details for the newly selected device without the necessity to go back to the device list first.

### Subchannel Details

Subchannel 0019 Device 0148 Select/Refresh

#### Path Management Control Word

Interruption Parameter											
00000000											
Q	0	ISC	00	A	E	LM	MM	D	T	V	DEVNUM
0	0		0	1	00	00	0	0	1		0148
LPM			PNOM			LPUM		PIM			
80			00			00		80			
MBI						POM		PAM			
0000						FF		80			
CHPID=0			CHPID=1			CHPID=2		CHPID=3			
01			00			00		00			
CHPID=4			CHPID=5			CHPID=6		CHPID=7			
00			00			00		00			
ZONE		00000		VISC		00000000		I	000000		S
00				0				0			0

Figure 22: Web Browser Interface – Subchannel Details

#### 4.4.7 Version Info

The version information displays details about the running Hercules software release. Beneath the current version (as the name implies) it shows additional information:

- Copyright information
- Hercules build date and time

- Options that have been built into Hercules
- Information about the host system

The information shown in this view is the same as returned by the VERSION console command.

```

Hercules Version Information

HHC01413I Hercules HTTP Server version 3.0.7.7541
HHC01414I (c) Copyright 1999-2011 by Roger Bowler, Jan Jaeger, and others
HHC01415I Built on Jul  3 2011 at 09:40:38
HHC01416I Build information:
HHC01417I Windows (MSVC) build for AMD64
HHC01417I Modes: S/370 ESA/390 z/Arch
HHC01417I Max CPU Engines: 8
HHC01417I Using fthreads Threading Model
HHC01417I Using Error-Checking Mutex Locking Model
HHC01417I Using FishIO
HHC01417I Dynamic loading support
HHC01417I Using shared libraries
HHC01417I HTTP Server support
HHC01417I No SIGABEND handler
HHC01417I Regular Expressions support
HHC01417I Automatic Operator support
HHC01417I Machine dependent assists: cmpxchg1 cmpxchg4 cmpxchg8
HHC01417I Running on GOOFY Windows-6.1.7601. NT , Intel(R) x64 MP=8

```

Figure 23: Web Browser Interface – Version Info

#### 4.4.8 CPU

The CPU view is used to configure CPUs online or offline. Each CPU in the current configuration is listed here with the CPU number, a drop-down list displaying the current CPU state and an “Update” button.

The example shown below is from a configuration with only one CPU.

```

Configure CPU

CPU0000

Online ▾ Update

```

Figure 24: Web Browser Interface – CPU

The CPU state can be changed through selecting “Online” or “Offline” in the drop-down list and clicking on the “Update” button.

#### 4.4.9 GPRs

The GPRs (General Purpose Register) view displays the current values of the general purpose registers. There is no possibility to change the values of the general purpose registers from this view. To alter the contents of the registers use the registers view instead.

<b>General Registers</b>			
GR00=00000000	GR01=00000000	GR02=00000000	GR03=00000000
GR04=00000000	GR05=00000000	GR06=00000000	GR07=00000000
GR08=00000000	GR09=00000000	GR10=00000000	GR11=00000000
GR12=00000000	GR13=00000000	GR14=00000000	GR15=00000000

Figure 25: Web Browser Interface – GPRs

#### 4.4.10 CRs

The CRs (Control Register) view displays the current values of the control registers. There is no possibility to change the values of the control registers from this view. To alter the contents of the control registers use the registers view instead.

<b>Control Registers</b>			
CR00=C080EC40	CR01=0FFD7C00	CR02=FFFFFFFF	CR03=00000000
CR04=00000000	CR05=00000000	CR06=00000000	CR07=00000000
CR08=00000000	CR09=00000000	CR10=00000000	CR11=00000000
CR12=00000000	CR13=00000000	CR14=EFC00000	CR15=00FEBB10

Figure 26: Web Browser Interface – CRs

#### 4.4.11 PSW

The PSW view displays the current contents of the Program Status Word (PSW). Like in the GPRs and CRs view the value cannot be changed. To alter the contents of the individual PSW fields the “PSW” console command must be used.

The “Auto Refresh” button has the same function as in the log view. Through clicking on “Auto Refresh” the view is refreshed automatically. The refresh interval in seconds can be specified in the entry field beneath the “Auto Refresh” button. Once auto refresh is turned on you can stop it by clicking again on the same button, whose description meanwhile has changed to “Stop Refreshing”.

## Program Status Word

Auto Refresh

Refresh Interval:

PSW=070E0000 00000000

Figure 27: Web Browser Interface – PSW

### 4.4.12 Documentation

The documentation view lets you browse through the Hercules online documentation. The information found here is exactly the same as the documentation on the Hercules website.

# The Hercules System/370, ESA/390, and z/Architecture Emulator

---

**Hercules** is an open source software implementation of the mainframe System/370 and ESA/390 architectures, in addition to the new 64-bit z/Architecture. Hercules runs under Linux, Windows (98, NT, 2000, and XP), Solaris, FreeBSD, and Mac OS X (10.3 and later).

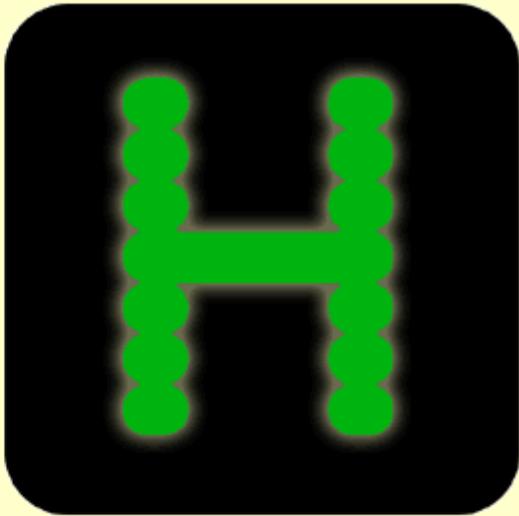


Figure 28: Web Browser Interface – Documentation

---

## 4.5 Graphical User Interfaces (GUIs)

Although Hercules has a semi-graphical user interface and can also be operated through its web server interface the possibilities in a non-GUI (command-line) application or in a web browser are somewhat limited.

For this reason several comfortable third-party GUIs have been developed. Currently the following GUIs are available:

- Hercules WinGUI (for Windows platforms, written by Fish)
- Hercules Studio (for Linux platforms, written by Jacob Dekel)
- Hercules Image Manager Hebe (for Linux platforms, written by Robin Atwood)

Because each of these products is an add-on to Hercules they are described in detail in Part III of this manual ("Third-Party Utilities").

---

## Part II: Hercules Supplied Utilities

---

## 5. Hercules Supplied Utilities

---

### 5.1 Overview

Several utility programs exist to support the Hercules Emulator. Most of them are used when working with DASD and TAPE files. Other utilities perform some special functions with configuration files.

All these utility programs are pure line-command mode programs (i.e. Windows DOS or Unix/Linux shell). They can be called manually from the command line prompt or built into batch files. This is generally the preferred method due to its greater flexibility and automation possibilities.

If the Hercules GUI (HercGUI) is used the utilities can also be called interactively via the GUI and using a graphical and menu driven interface.

#### 5.1.1 DASD Utilities

The following table shows the utilities that are used for DASD image file maintenance:

Utility Name	Function
CCKDCDSK	CCKD DASD file integrity verification, recovery and repair utility
CCKDCOMP	CCKD DASD file compression utility
CCKDDIAG	CCKD DASD file diagnostics utility
CKDSWAP	CCKD DASD file swap-endian program
DASDCAT	Display PDS datasets and members
DASDCONV	DASD image file conversion program
DASDCOPY	Copy DASD file to another DASD file
DASDINIT	DASD image file creation
DASDISUP	Fix XCTL tables in SVCLIB
DASDLOAD	DASD loader program
DASDLS	List datasets on a volume
DASDPDSU	PDS unload utility
DASDSEQ	Display sequential datasets

**Table 1: DASD image file maintenance utilites**

### 5.1.2 TAPE Utilities

The following table shows the utilities that are used for TAPE file maintenance:

Utility Name	Function
HETGET	Extract files from an AWS or HET tape file
HETINIT	Initialize an AWS or HET tape file
HETMAP	Show information about a HET or AWS tape file
HETUPD	Update and/or copy an AWS or HET tape file
TAPECOPY	Copy a SCSI tape to or from an AWSTAPE disk file
TAPEMAP	Show information about an AWS tape file
TAPESPLT	Split an AWS tape file
VMFPLC2	VM formatted tape utility

**Table 2: TAPE file maintenance utilities**

### 5.1.3 Miscellaneous Utilities

The following table shows utilities used for miscellaneous functions:

Utility Name	Function
DMAP2HRC	P/390 DEVMAP conversion program

**Table 3: Miscellaneous utilities**

### 5.1.4 Third Party Utilities

The following table shows “third party” utilities used for various functions:

Utility Name	Function
FTAPE	SCSI tape test utility
PRTPUB	Mainframe print publishing program

**Table 4: Third party utilities**

### 5.1.5 Mainframe Utilities

The utilities according to the next table will run on the mainframe and provide various functions that are useful for Hercules users:

Utility Name	Function
AWSSL	AWS virtual tape utility
AWSUTIL	AWS format tape file generation utility
RAWSTAPE	Reverse AWSTAPE utility
VTTS	Virtual tape transportation system

**Table 5: Mainframe utilities**

---

## 5.2 CCKDCDSK (CCKD DASD file integrity verification, recovery and repair utility)

### 5.2.1 Function

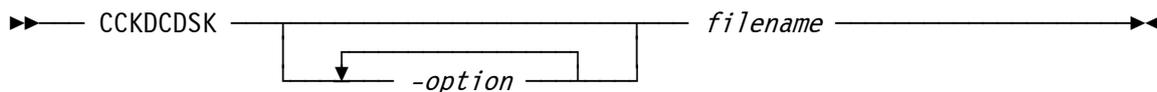
The CCKDCDSK utility performs compressed or shadowed CKD DASD emulation file integrity verification and recovery and repair. Calling the utility without any arguments will display a help information.

### 5.2.2 Syntax

#### Descriptive

```
CCKDCDSK [-option [-option ... ]] filename
```

#### Diagram



### 5.2.3 Parameter

#### *options:*

- v Display version information and exit.
- f Force check, even if OPENED bit is on.
- ro Open the DASD file read-only, no repairs are done.
- level The level of checking that is to be performed. The higher the level the longer the integrity check takes. The level is any number from 1 - 4.
  - Level 1 performs a minimal check. Device headers are verified, free space is verified, primary lookup table and secondary lookup tables are verified.
  - Level 2 performs a normal check. These are the same checks as in level 1 plus all 5-byte track headers are verified.
  - Level 3 performs the maximum checking. These are the same checks as in level 1 plus all track images are read, uncompressed and verified.
  - Level 4 recovers everything.

**arguments:**

**filename**            Name of the CCKD DASD image file to be checked.

## 5.2.4 Examples

**Example 1:**

Perform a minimal checking of a CCKD DASD file without doing any repairs.

```
CCKDCDSK -0 -ro D:\MVS\DASD\MVSRES.CCKD
```

**Example 2:**

Perform a maximum level check and repair of a CCKD DASD file and force the check even the OPENED bit is set on.

```
CCKDCDSK -r -f D:\MVS\DASD\MVSRES.CCKD
```

## 5.2.5 Sample Utility Output

**Sample 1:**

Perform a maximum level check and repair of a CCKD DASD file and force the check even the OPENED bit is set on.

```
D:\HERCULES>cckdcnsk -3 -ro D:\MVS\DASD\MVSRES.CCKD
size 301956072 used 301956072 free 0 imbed 0 first 0x0 number 0
```

**Figure 29: CCKDCDSK utility output**

---

## 5.3 CCKDCOMP (CCKD DASD file compression utility)

### 5.3.1 Function

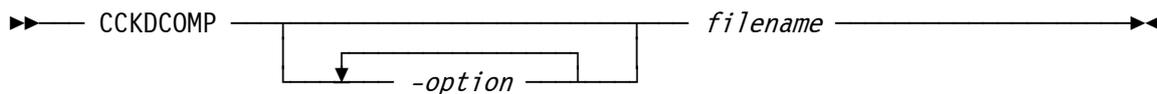
The CCKDCOMP utility removes all free space from a compressed or shadow CKD DASD emulation file. If *level* is specified then the CCKDCDSK utility is called first with the specified level; this is a short-hand method to call both functions in one utility call.

### 5.3.2 Syntax

#### Descriptive

CCKDCOMP [-option [-option ... ]] *filename*

#### Diagram



### 5.3.3 Parameter

#### **options:**

- v            Display version information and then exit.
- f            Force check even if OPENED bit is on.
- level        The level of checking that is to be performed. The higher the level the longer the integrity check takes. The level is any number from 1 - 4.
  - Level 1 performs a minimal check. Device headers are verified, free space is verified, primary lookup table and secondary lookup tables are verified.
  - Level 2 performs a normal check. These are the same checks as in level 1 plus all 5-byte track headers are verified.
  - Level 3 performs the maximum checking. These are the same checks as in level 1 plus all track images are read, uncompressed and verified.
  - Level 4 recovers everything.

#### **arguments:**

**filename**        Name of the DASD image file to be compressed.

## 5.3.4 Examples

### Example 1:

Compress a CCKD DASD file.

```
CCKDCOMP D:\MVS\DASD\MVSRES.CCKD
```

### Example 2:

Compress a CCKD DASD file and perform integrity checking and repair.

```
CCKDCOMP -3 D:\MVS\DASD\MVSRES.CCKD
```

## 5.3.5 Sample Utility Output

### Sample 1:

Compress a CCKD DASD file and perform integrity checking and repair.

```
D:\HERCULES>cckdcomp -3 D:\MVS\DASD\MVSRES.CCKD
cckdcomp: completed: moves 33665  moved 301552856  freed 407400
```

**Figure 30: CCKDCOMP utility output**

---

## 5.4 CCKDDIAG (CCKD DASD file diagnostics utility)

### 5.4.1 Function

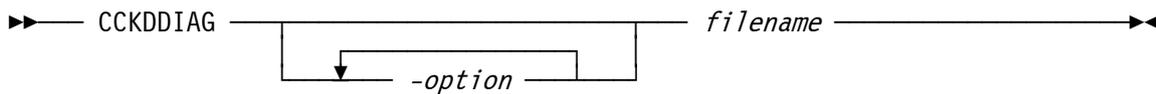
The CCKDDIAG utility is the diagnostic program which allows you to examine various aspects of a CCKD DASD image file.

### 5.4.2 Syntax

#### Descriptive

CCKDDIAG [-option [-option ... ]] filename

#### Diagram



### 5.4.3 Parameter

#### **options:**

- v Display version information and then exit.
- d Display DEVHDR.
- c Display CDEVHDR.
- 1 Display L1TAB.
- g Enable debug output.

#### **CKD track related options:**

- a *cc hh* Display absolute *cchh* data.
- r *tt* Display relative *TT* data.
- 2 Display L2TAB related to -a or -r.
- x Hex display track / key data.

**offset option:**

**-o oo ll** Hex display data at offset oo of length ll.

**arguments:**

**filename** The name of the DASD image file to be compressed.

### 5.4.4 Examples

**Example 1:**

Display the DEVHDR and CDEVHDR for a CCKD DASD image file.

```
CCKDDIAG -d -c D:\MVS\DASD\MVSRES.CCKD
```

**Example 2:**

Display the L1TAB and enable debug output for a CCKD DASD image file.

```
CCKDDIAG -l -g D:\MVS\DASD\MVSRES.CCKD
```

**Example 3:**

Display relative TT data for a CCKD DASD image file.

```
CCKDDIAG -r 15 D:\MVS\DASD\MVSRES.CCKD
```

### 5.4.5 Sample Utility Output

**Sample 1:**

Display the DEVHDR and CDEVHDR for a CCKD DASD image file and additionally enable debug output.

```
D:\Hercules>cckddiag -d -c -g D:\MVS\DASD\MVSRES.CCKD

READPOS seeking 0 (0x00000000)
READPOS reading buf addr 0012FA68 length 512 (0x00000200)

DEVHDR - 512 (decimal) bytes:
+0000 434B445F 43333730 1E000000 004C0000 C.D^C370.....<..
+0010 50000000 00000000 00000000 00000000 &.....
+0020 00000000 00000000 00000000 00000000 .....

3350 device has 30 heads/cylinder

READPOS seeking 512 (0x00000200)
READPOS reading buf addr 0012FC68 length 512 (0x00000200)
```

```
CDEVHDR - 512 (decimal) bytes:
+0000 00030141 42000000 00010000 0CFF5901 ...AB.....Y.
+0010 21F74901 19631500 EB071000 5CE90E00 !7I..c.....*Z..
+0020 0B000000 00000000 30020000 00010500 .....0.....
+0030 00000000 00000000 00000000 00000000 .....

MAKBUF malloc L1TAB buffer of 264 bytes at 003B2860

READPOS seeking 1024 (0x00000400)
READPOS reading buf addr 003B2860 length 264 (0x00000108)
```

**Figure 31: CCKDDIAG utility output**

---

## 5.5 CCKDSWAP (CCKD DASD file swap-endian program)

### 5.5.1 Function

The CCKDSWAP utility is the swap-endian program, it swaps the byte order of a CCKD file.

### 5.5.2 Syntax

#### Descriptive

CCKDSWAP *filename*

#### Diagram

▶— CCKDSWAP — *filename* —▶

### 5.5.3 Parameter

***filename***            The name of the CCKD DASD image file to have its byte order swapped.

### 5.5.4 Examples

#### Example 1:

Swap byte order for a CCKD DASD image file.

```
CCKDSWAP D:\MVS\DASD\MVSRES.CCKD
```

### 5.5.5 Sample Utility Output

#### Sample 1:

Swap byte order for a CCKD DASD image file.

```
D:\HERCULES>cckdswap D:\MVS\DASD\MVSRES.CCKD
Hercules cckd swap-endian program Version 3.05
(c)Copyright 1999-2005 by Roger Bowler, Jan Jaeger, and others
cckdswap: D:\MVS\DASD\MVSRES.CCKD changed from little-endian to big-endian
```

**Figure 32: CCKDSWAP utility output**

---

## 5.6 DASDCAT (Display PDS datasets and members)

### 5.6.1 Function

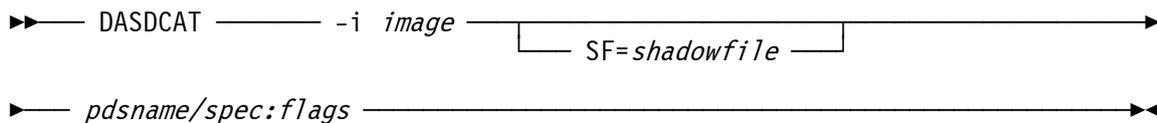
The DASDCAT utility displays PDS datasets and members from a DASD image file. It can generate a list of members in a given PDS dataset or can list the contents of a single member or all members in a PDS.

### 5.6.2 Syntax

#### Descriptive

```
DASDCAT -i image [SF=shadowfile] pdsname/spec:flags
```

#### Diagram



### 5.6.3 Parameter

- i *image***           The name (and optionally path) of the DASD image file.
- shadowfile***        The (optional) name of an associated shadow DASD file.
- pdsname***           The name of the partitioned dataset from which information will be extracted.
- spec***               *spec* can be one of the following:
- The name of a PDS member to be listed.
  - An asterisk (“\*”), meaning that all members in the PDS are to be listed.
  - A question mark (“?”) which causes the list of members in the PDS to be returned.
- flags***             *flags* can be one of the following:
- “c” (list the member(s) as card images)
  - “a” (list the member(s) in ASCII)

If no arguments are given the program displays help text and exits.

## 5.6.4 Examples

### Example 1:

Create a member list of PDS SYS1.PARMLIB on volume MVSRES.CCKD.

```
DASDCAT -i D:/MVS/DASD/MVSRES.CCKD SYS1.PARMLIB/?
```

### Example 2:

Print member IEASLP00 from PDS SYS1.PARMLIB on volume MVSRES.CCKD as card images.

```
DASDCAT -i D:/MVS/DASD/MVSRES.CCKD SYS1.PARMLIB/IEASLP00:c
```

### Example 3:

Print all members from PDS SYS1.PARMLIB on volume MVSRES.CCKD as card images.

```
DASDCAT -i D:/MVS/DASD/MVSRES.CCKD SYS1.PARMLIB/*:c
```

## 5.6.5 Sample Utility Output

### Sample 1:

Create a member list of PDS SYS1.PARMLIB on volume MVSRES.CCKD

```
D:\Hercules>dasdcat -i D:/MVS/DASD/MVSRES.CCKD SYS1.PARMLIB/?
Hercules DASD cat program Version 3.05
(c)Copyright 1999-2006 by Roger Bowler, Jan Jaeger, and others
commnd00
gtfparm
ieabd00
ieaapf00
ieabl00
ieadmp00
.
.
.
lnklst00
mvikey00
rpfkey00
smfprm00
tsokey00
vatlst00
```

Figure 33: DASDCAT utility output (create PDS member list)

## Sample 2:

Print member IEAIPS00 from PDS SYS1.PARMLIB on volume MVSRES.CCKD as card images.

```
D:\Hercules>dasdcat -i D:/MVS/DASD/MVSRES.CCKD SYS1.PARMLIB/IEAIPS00:c
Hercules DASD cat program Version 3.05
(c)Copyright 1999-2006 by Roger Bowler, Jan Jaeger, and others
CPU=10.0,IOC=5.0,MSO=3.0          /* DEFAULT IPS --IEAIPS00-- */
WKL=(1,50,99,100)
OBJ=1,SRV=(1000,*,*,0)           /* FIRST PERIOD BATCH (DMN1) */
OBJ=2,SRV=(1000,*,0)             /* SECOND PERIOD BATCH (DMN1) */
OBJ=3,SRV=(1000,*,*,0)           /* FIRST PERIOD TSO (DMN2) */
OBJ=4,SRV=(1000,*,*,0)           /* SECOND PERIOD TSO (DMN2) */
OBJ=5,SRV=(1000,*,*,0)           /* THIRD PERIOD TSO (DMN3) */
OBJ=6,SRV=(1000,1000,1000,1000) /* HOT BATCH (DMN1) */
OBJ=7,SRV=(1000,1,0)             /* LOW PRIORITY BATCH (DMN1) */
DMN=1,CNSTR=(1,50,1)             /* BATCH */
DMN=2,CNSTR=(1,50,255)           /* SHORT AND MEDIUM TSO */
DMN=3,CNSTR=(1,50,16)           /* LONG TSO */
PGN=1,(DMN=1,APG=6,DUR=10K,ISV=10K,OBJ=1,RTB=0) /* BATCH -SHORT */
      (DMN=1,APG=6,ISV=10K,OBJ=2,RTB=0) /* -----MEDIUM */
PGN=2,(DMN=2,APG=14,DUR=600,ISV=600,OBJ=3,RTB=0) /* TSO -SHORT */
      (DMN=2,APG=13,DUR=1400,ISV=2K,OBJ=4,RTB=0) /* -----MEDIUM */
      (DMN=3,APG=6,ISV=10K,OBJ=5,RTB=0) /* -----LONG */
PGN=3,(DMN=1,APG=9,ISV=100K,OBJ=6,RTB=0) /* HOT BATCH */
PGN=4,(DMN=1,APG=6,ISV=10K,OBJ=7,RTB=0) /* LOW PRTY BATCH*/
```

Figure 34: DASDCAT utility output (list a member from a PDS)

## Sample 3:

Print all members from PDS SYS1.PARMLIB on volume MVSRES.CCKD as card images.

```
D:\Hercules>dasdcat -i D:/MVS/DASD/MVSRES.CCKD SYS1.PARMLIB/*:c
Hercules DASD cat program Version 3.05
(c)Copyright 1999-2006 by Roger Bowler, Jan Jaeger, and others
> Member COMMND00
| TOD=NOPROMPT
> Member GTFPARM
| TRACE=SYS,USR,TRC,DSP,PCI,SRM,RR,EXT,PI,SVC,IO,SIO
> Member IEAABD00
| SDATA=(ALLSDATA),PDATA=(ALLPDATA)
> Member IEAAPF00
| SYS1.LINKLIB          MVSRES,
| SYS1.SVCLIB          MVSRES,
| SYS1.LPALIB          MVSRES,
| SYS1.VTAMLIB         MVSRES,
| SYS1.INDMAC          MVSRES
> Member IEABLD00
| SYS1.LINKLIB ALLOC,ALLOCATE,E,EDIT,HEWL,
|
|           IEWL,IEWLF128,IEWLF440,IEWLF880,IFOX01,IFOX02,IFOX03,
|           IFOX04,IFOX05,IFOX06,IFOX11,IFOX21,IFOX31,IFOX41,IFOX51,
|           IFOX61,IFOX62,LINK,LINKEDIT,LOADER,
```

```

|           LOGOFF,LOGON,SUBMIT,TEST
.
.
.
several lines not displayed
.
.
.
> Member TSOKEY00
| USERMAX=8,           /* MAXIMUM TSO USERS           */
| RECONLIM=60,        /* MAXIMUM DISCONNECT MINUTES  */
| BUFRSIZE=132,       /* VTIOC BUFFER SIZE           */
| HIBFREXT=48000,     /* MAX BUFFERS BEFORE SWAP OUT  */
| LOBFREXT=24000,     /* MINIMUM BUFFERS BEFORE SWAP IN */
| MODE=NOBREAK,       /* KEYBOARD LOCK OPTION        */
| MODESW=NO,          /* MODESWITCH FROM TERMINAL OPTION */
| CHNLEN=4,           /* NO. OF RU'S PER CHAIN       */
| SCRSIZE=1920        /* MAXIMUM SCREEN SIZE         */

```

**Figure 35: DASDCAT utility output (list all members of a PDS)**

---

## 5.7 DASDCONV (DASD image file conversion program)

### 5.7.1 Function

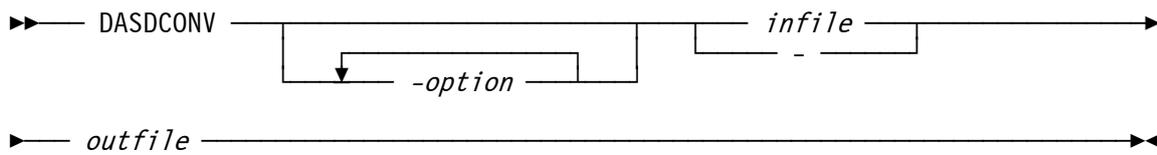
The DASD CKD image conversion utility is used to convert HDR-30 CKD image files or compressed (.gz) image files to the AWSCKD format that is used by the Hercules Emulator. The .gz compressed format was recently used by IBM to ship the ADCD DASD image files for the FLEX-ES emulator.

### 5.7.2 Syntax

#### Descriptive

```
DASDCONV [-option [-option ... ]] {infile | - } outfile
```

#### Diagram



### 5.7.3 Parameter

#### **options:**

- r Replace the output file if it already exists.
- lfs Output CKD file will be a single file even if it exceeds 2 GB in size.
- q The "-q" (quiet) option suppresses the progress messages.

#### **arguments:**

- infile** The name (and optionally path) of the input HDR-30 CKD image file or the name of the compressed (.gz) image file.
- If the hyphen is specified instead of a filename, then DASDCONV reads from stdin. This gives the possibility of piping the output from gunzip in the case that DASDCONV was generated without gzip support.
- outfile** The name of the AWSCKD image file to be created.

## 5.7.4 Examples

### Example 1:

Convert a .gz compressed DASD image file to AWSCKD format. Create a large file (> 2 GB) and replace the output file if it already exists.

```
DASDCONV -r -lfs D:\MVS\DASD\MVSRES.GZ D:\MVS\DASD\MVSRES.CKD
```

## 5.7.5 Sample Utility Output

### Sample 1:

Convert a .gz compressed DASD image file to AWSCKD format. Create a large file (> 2 GB) and replace the output file if it already exists.

```
D:\HERCULES>DASDCONV -r -lfs D:\MVS\DASD\MVSRES.GZ D:\MVS\DASD\MVSRES.CKD
Hercules DASD CKD image conversion program
Version 3.05
(c)Copyright 1999-2005 by Roger Bowler, Jan Jaeger, and others
Converting 3390 volume MVSRES: 3339 cyls, 15 trks/cyl, 56832 bytes/track
3339 cylinders successfully written to file MVSRES.ckd
DASD conversion successfully completed.
```

**Figure 36: DASDCONV utility output**

---

## 5.8 DASDCOPY (Copy DASD file to another DASD file)

### 5.8.1 Function

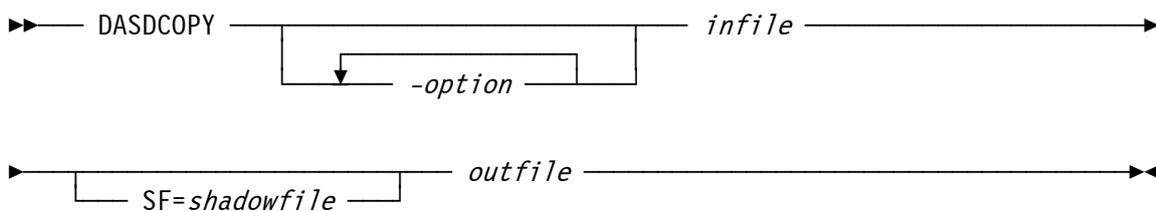
The DASDCOPY utility is used to copy a DASD image file to another DASD image file.

### 5.8.2 Syntax

#### Descriptive

```
DASDCOPY [-option [-option ... ]] infile [SF=shadowfile] outfile
```

#### Diagram



### 5.8.3 Parameter

#### *options:*

- v Display version information and help text.
- h Display usage summary and quit.
- q Quiet mode, do not display status.
- r Replace the output file if it already exists.
- z Compress output using zlib (default).
- bz2 Compress output using bzip2.
- 0 Do not compress output (0 = zero).
- blks *n* Size of output FBA file.
- cyls *n* Size of output CKD file.
- a Output CKD file will have alternate cylinders.

**-lfs** Output CKD file will be a single file even if it exceeds 2 GB in size.

**-o type** Output file type (CKD, CCKD, FBA, CFBA).

**arguments:**

**infile** The input DASD file.

**shadowfile** The (optional) name of an associated shadow DASD file.

**outfile** The output DASD file.

## 5.8.4 Examples

### Example 1:

Copy the compressed DASD file (CCKD) MVSRES.CCKD to SARESX.148, uncompress it (CKD file) and replace an already existing file with the same name.

```
DASDCOPY -r -lfs D:/MVS/DASD/MVSRES.CCKD D:/MVS/DASD/MVSRESX.148
```

### Example 2:

Copy the compressed DASD file (CCKD) MVSRES.CCKD to SARESX.148, uncompress it (CKD file) and create 2 separate files for systems which do not support large file support.

```
DASDCOPY D:/MVS/DASD/MVSRES.CCKD D:/MVS/DASD/MVSRESX.148
```

## 5.8.5 Sample Utility Output

### Sample 1:

Copy the DASD image file MVSRES.CCKD to the new DASD image file SARESX.148. Do not create a large image file (> 2 GB), create two separate files instead.

```
D:\HERCULES>dasdcopy D:/MVS/DASD/MVSRES.CCKD D:/MVS/DASD/SARESX.148
HHCDU044I Creating 3390 volume : 3339 cyls, 15 trks/cyl, 56832 bytes/track
HHCDU041I 2519 cylinders successfully written to file D:/MVS/DASD/SARESX_1.148
HHCDU041I 820 cylinders successfully written to file D:/MVS/DASD/SARESX_2.148
/ 100% 50085 of 50085
HHCDC010I Copy successful !!!
```

**Figure 37: DASDCOPY utility output**

---

## 5.9 DASDINIT (DASD image file creation)

### 5.9.1 Function

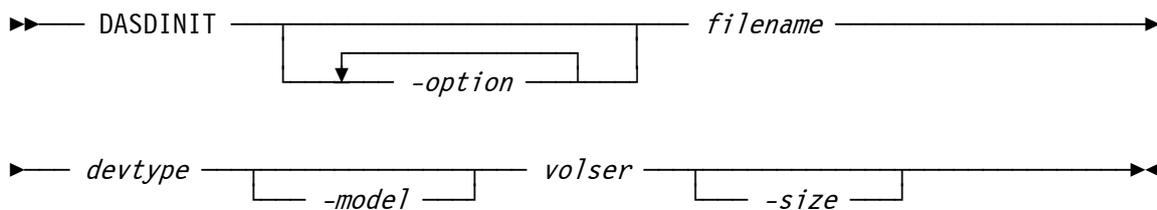
The Hercules DASD image file creation program (DASDINIT) is used to create an empty DASD image file.

### 5.9.2 Syntax

#### Descriptive

```
DASDINIT [-option [-option ... ]] filename devtype[-mode] volser [size]
```

#### Diagram



### 5.9.3 Parameter

#### **options:**

- v** Display version information and help text.
- z** Build compressed DASD image file using zlib.
- bz2** Build compressed DASD image file using bzip2.
- 0** Build compressed DASD image file with no compression (0 = zero).
- lfs** Build a large (uncompressed) DASD file (single file), if it is supported on the platform DASDINIT is running on.
- a** Build a DASD image file that includes alternate cylinders. This option is ignored if the size is manually specified.
- r** Build raw DASD image file (no VOL1 or IPL track).
- b** Make the wait PSW in the IPL1 record a BC-mode PSW. If this option is not specified, the wait PSW will be an EC-mode PSW.

- m** Enable the wait PSW in the IPL1 record for machine check interruptions. If this option is not specified, the wait PSW will be disabled for machine checks.
- linux** Null track images will look like Linux DASDFMT'ed images (3390 device type only).

**arguments:**

- filename** The name of the DASD image file to be created.
- devtype** Specifies the device type. Valid device types are:  
 FBA: 0671, 3310, 3370, 9313, 9332, 9335, 9336  
 CKD: 2311, 2314, 3330, 3340, 3350, 3375, 3380, 3390, 9345  
 A complete list of all Hercules supported DASD device types and models with additional information can be found in Appendix A.
- model** Specifies the device model (note – this implies size). A list of Hercules supported device types and models can be found in Appendix A.
- volser** Volume serial number (1-6 characters).
- size** Number of CKD cylinders or 512-byte FBA sectors (required if model not specified, otherwise optional).

For CKD volumes which exceed 2GB, such as the 3390-3, if the *-lfs* parameter is not specified the DASDINIT program will create multiple files by appending the characters *\_1*, *\_2*, *\_3* etc. to the file name specified on the command line. These characters are inserted before the first dot (.) occurring after the last slash (/). If there is no dot then the characters are appended to the end of the name. Each file contains a whole number of cylinders.

Hercules CKD support recognizes the files as belonging to a single logical volume. Specify the full name of just the first file in the Hercules configuration file (e.g. "filename\_1").

The DASDINIT program cannot create FBA volumes exceeding 2GB unless the *-lfs* parameter is specified and large file size is supported on your platform.

Note that the defaults for the wait PSW written to the IPL1 record have changed from earlier releases of Hercules. In the past, the wait PSW created by DASDINIT was a BC-mode PSW enabled for machine interrupts. The current default for the wait PSW is EC-mode, disabled for machine checks. To obtain the earlier behaviour, run DASDINIT with the *'-b'* and *'-m'* options.

## 5.9.4 Examples

### Example 1:

Create a 3390 Model 3 with volume serial USR001, the file(s) should be named USR001.300 and place them into directory D:\MVS\DASD\. Note that the utility creates actually two files called USR001\_1.300 and USR001\_2.300 because no large file support is requested.

```
DASDINIT D:\MVS\DASD\USR001.300 3390-3 USR001
```

**Example 2:**

Create the same DASD volume as in the previous example but create just one file (large file support) named USR001.300 in the same directory.

```
DASDINIT -lfs D:\MVS\DASD\USR001.300 3390-3 USR001
```

**Example 3:**

Create a 3390 Model 3 with volume serial SYS3A0 and alternate cylinder(s), the file should be compressed with ZLIB and the filename has to be TSTVOL.DSD. The DASD volume must be placed in directory D:\MVS\DASD\.

```
DASDINIT -z -a D:\MVS\DASD\TSTVOL.DSD 3390-3 SYS3A0
```

## 5.9.5 Sample Utility Output

**Sample 1:**

Create a 3390 model 3 DASD image file named SYS3A0.3A0 with volser=SYS3A0.

```
D:\HERCULES>DASDINIT D:\MVS\DASD\SYS3A0.3A0 3390-3 SYS3A0
HHCDU044I Creating 3390 volume SYS3A0: 3339 cyls, 15 trks/cyl, 56832 bytes/track
HHCDU041I 2519 cylinders successfully written to file D:/MVS/DASD/SYS3A0_1.3A0
HHCDU041I 820 cylinders successfully written to file D:/MVS/DASD/SYS3A0_2.3A0
HHCDI001I DASD initialization successfully completed.
```

**Figure 38: DASDINIT utility output**

---

## 5.10 DASDISUP (Fix XCTL tables in SVCLIB)

### 5.10.1 Function

The DASDISUP utility fixes the XCTL tables in SVCLIB on OS/360 systems. On an OS/360 system the OPEN / CLOSE / EOVS modules in SYS1.SVCLIB have XCTL tables embedded within them. These tables contain TTRs, pointing to other modules, and these TTRs need to be adjusted after loading SVCLIB to DASD.

OS/360 provides a program called IEHIOSUP to perform this function, however the issue is that you cannot run IEHIOSUP until you have the system up and running, and you cannot IPL the system until you have fixed the XCTL tables.

To circumvent this problem Hercules provides a utility program called DASDISUP which can be run from the Unix or Windows command line after running DASDLOAD and fixes the XCTL tables.

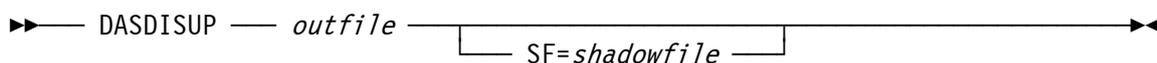
**Note:** Do not use this procedure except on OS/360 IPL volumes; other operating systems do not have XCTL tables.

### 5.10.2 Syntax

#### Descriptive

DASDISUP *outfile* [SF=*shadowfile*]

#### Diagram



### 5.10.3 Parameter

**outfile**            The name of the OS/360 IPL volume to be updated.

**shadowfile**        The (optional) name of the associated shadow DASD file.

### 5.10.4 Examples

#### Example 1:

Fix the XCTL tables in SVCLIB on volume OS36IP.148.

```
DASDISUP D:/MVS/DASD/OS36IP.148
```

## 5.10.5 Sample Utility Output

### Sample 1:

Fix the XCTL tables in SVCLIB on volume OS36IP.148.

```
D:\HERCULES>dasdisup D:/MVS/DASD/OS36IP.148
Hercules IEHIOSUP program Version 3.05
(c)Copyright 1999-2005 by Roger Bowler, Jan Jaeger, and others
HHCDS006W Member IGG019PZ is not single text record
HHCDS006W Member IGG019V6 is not single text record
HHCDS007W Member IGG019V6 size 0A20 exceeds X'7F8' bytes
HHCDS002I End of directory: 4 members selected
HHCDS010I Member IGG019PX skipped
HHCDS010I Member IGG019PY skipped
HHCDS010I Member IGG019PZ skipped
HHCDS010I Member IGG019V6 skipped
```

**Figure 39: DASDISUP utility output**

---

## 5.11 DASDLOAD (DASD loader program)

### 5.11.1 Function

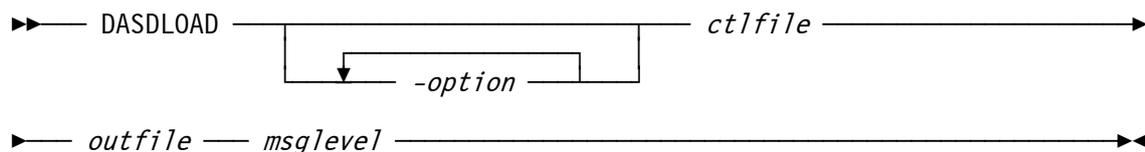
The Hercules DASD loader program is used to create new DASD file images and load them with data from unloaded PDS files (e.g. a list of TSO XMIT files or a sequential file).

### 5.11.2 Syntax

#### Descriptive

```
DASDLOAD [-option [-option ... ]] ctlfile outfile msglevel
```

#### Diagram



### 5.11.3 Parameter

#### **options:**

- z** Build compressed DASD image file using zlib.
- bz2** Build compressed DASD image file using bzip2.
- 0** Build compressed DASD image file with no compression (0 = zero).
- lfs** Build a large DASD image file (can exceed 2 Gb in size).
- a** Build DASD image file that includes alternate cylinders.
- b** For a volume without IPL text, make the wait PSW written to the IPL1 record a BC-mode PSW. If this option is not specified, the wait PSW will be an EC-mode PSW.
- m** For a volume without IPL text, make the wait PSW written to the IPL1 record enabled for machine checks. If this option is not specified, the wait PSW will be disabled for machine checks.

**arguments:**

- ctlfile** This is the name of the control file that specifies the datasets to be loaded onto the newly created volume.
- outfile** The name of the DASD image file to be created.
- msglevel** This can be a number from 0 to 5, it controls the detail level of messages issued during the load.

Note that the DASDLOAD defaults for the wait PSW written to the IPL1 record on non-IPLable volumes have changed from earlier releases of Hercules. In the past, the wait PSW created by DASDLOAD on volumes without IPL text was a BC-mode PSW enabled for machine check interrupts. The current default for the wait PSW is EC-mode, disabled for machine checks. To obtain the earlier behaviour, run the DASDLOAD utility with the '-b' and '-m' flags.

### 5.11.4 Control File

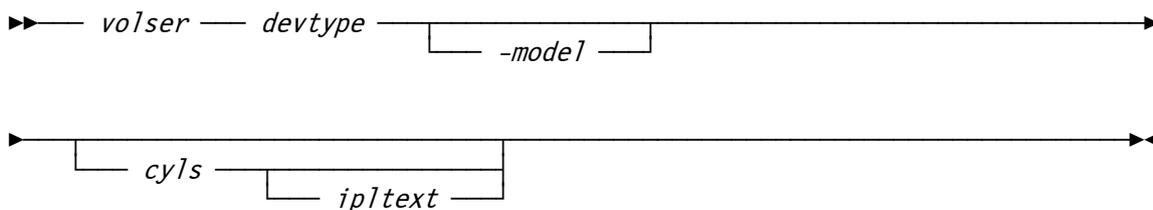
The control file required by the DASDLOAD program is an ASCII text file consisting of a *volume statement* followed by one *dataset statement* for each dataset to be created.

#### 5.11.4.1 Volume Statement

**Descriptive**

*volser devtype[-model] [cyls [ipltext]]*

**Diagram**



where:

- volser** Volser is the volume serial number of the newly created volume.
- devtype** This specifies the emulated device type (2311, 3330, 3350, 3375, 3380, 3390) of the new volume. FBA device types are not supported by the DASDLOAD program. A list of Hercules supported device types and models can be found in Appendix A.
- model** This specifies the device model. The model implies the size of the volume.

A list of Hercules supported device types and models can be found in Appendix A.

**cyls** Cyls specifies the size of the new volume in cylinders. If *cyls* is coded as "\*" (asterisk), as "0" or is omitted, then the default size for the device type and model is used. Cylinders is ignored for compressed devices.

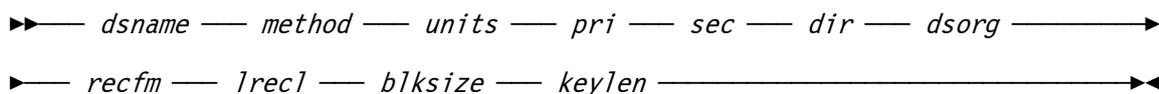
**ipltext** An optional parameter specifying the name of a file containing the IPL text which will be written to the volume. The file must be in the form of an object deck containing fixed length 80-byte records in the same format as expected by IBCDASDI or ICKDSF.

#### 5.11.4.2 Dataset Statement

##### Descriptive

*dsname method units pri sec dir dsorg recfm lrecl blksize keylen*

##### Diagram



where:

**dsname** The name of the dataset to be created.

**method** The dataset loading method can be one of the following:

**XMIT filename**

The dataset is loaded from an unloaded PDS created by the TSO XMIT command. The input is a binary file containing fixed length 80 byte records with no record delimiters.

**VS filename**

The dataset is loaded from an unloaded PDS created by IEBCOPY. The input is a binary file containing variable length spanned records with the record descriptor words omitted and with no record delimiters.

**SEQ filename**

The dataset is loaded from an binary file. ASCII / EBCDIC conversion is not currently supported. Also the DSORG must be 'PS' or 'DA' and RECFM must either be 'F' or 'FB'.

**XMSEQ filename**

The dataset is loaded from a dump of a sequential dataset created by the TSO XMIT command. The input is a binary file containing fixed length 80 byte records with no record delimiters.

**EMPTY**

The dataset is initialized with an end-of-file record (if DSORG is 'PS') or is an empty PDS directory (if DSORG is 'PO').

**DIP**

The dataset is initialized with a LOGREC header record.

**CVOL**

The dataset is initialized as an OS SYSCTLG containing the minimum entries needed to IPL an OS/360 system.

**VTOC filename**

Specifies the size and location of the VTOC. A dataset name must be coded on this statement although it is not used. If no VTOC statement is present, the VTOC will be placed after the last dataset on the volume and the size of the VTOC will be the minimum number of tracks necessary.

<b>units</b>	This specifies the space allocation units: 'TRK' or 'CYL'.
<b>pri</b>	This is the space allocation primary quantity.
<b>sec</b>	This is the space allocation secondary quantity.
<b>dir</b>	This is the number of directory blocks.
<b>dsorg</b>	This specifies the dataset organization: 'PS', 'PO', 'DA' or 'IS'.
<b>recfm</b>	This is the record format: 'F', 'FB', 'FBS', 'V', 'VB', 'VBS' or 'U'.
<b>lrecl</b>	This is the logical record length.
<b>blksize</b>	This is the block size.
<b>keylen</b>	This is the key length.

All parameters except the dsname and method are optional. Defaults of zero are supplied for DCB parameters. For datasets loaded with the XMIT method the DCB parameters are taken from the unloaded PDS, minimum space allocation required to load the dataset is used unless a larger quantity is specified. If space allocation is omitted the default is 'TRK 1 0 0'. If CYL is specified without any primary quantity then the default allocation is 1 cylinder or the minimum number of cylinders required to load the dataset, whichever is larger.

## 5.11.5 Examples

### Example 1:

Create a 2314 volume in a file called SYSRES.230 using the control file SYSRES.PLF with message level 2.

```
DASDLOAD SYSRES.PLF SYSRES230 2
```

The corresponding control file SYSRES.PLF looks like the following:

```
# Pack layout file for MFT system residence volume
sysres 2314 * ieaipl00.rdr
sys1.parmlib      xmit      /cdrom/os360/reslibs/parmlib.xmi
```

```

sys1.imagelib    xmit    /cdrom/os360/reslibs/imagelib.xmi
sysctlg         cvol    trk 1 0 0    ps f 256 256 8
sysvtoc         vtoc    trk 5
sys1.logrec     dip     trk 1 0 0
sys1.nucleus    xmit    /cdrom/os360/reslibs/nucleus.xmi cyl
sys1.svclib     xmit    /cdrom/os360/reslibs/svclib.xmi cyl
sys1.sysjobqe   empty   cyl 2 0 0    da f 176 176 0
sys1.dump       empty   cyl 10 0 0   ps u 0 3625 0

```

### Example 2:

Create a compressed 3390-3 volume in a file called LINUX.500 containing a bootable linux system for Linux/390 installation using the control file LINUX.PRM with message level 5.

```
DASDLOAD -z LINUX.PRM LINUX.500 5
```

The corresponding control file LINUX.PRM looks like the following:

```

#
# Build a bootable linux disk
# [Note: the dataset names (sys1.linux. ...) are hard-coded in
#       linuxipl.obj and cannot be changed without rebuilding it]
#
linux 3390-3 * linuxipl.obj
sys1.linux.parmfile SEQ images/redhat.prm trk 1 0 0 ps fb 1024 1024
sys1.linux.tapeipl.ikr SEQ images/kernel.img trk 200 0 0 ps fb 1024 1024
sys1.linux.initrd SEQ images/initrd.img trk 200 0 0 ps fb 1024 1024

```

## 5.11.6 Sample Utility Output

### Sample 1:

Create a 3390 model 3 in a compressed file called TST001.A80 containing 3 libraries (which have been downloaded from CBT tape as XMIT files) using the control file TST001.DAT with message level 1.

The control file TST001.DAT looks like follows:

```

TST001 3390-3 *
SYS2.RACF.UTIL XMIT D:\MVS\DASD\CBT728.XMI
SYS2.BATCH.CNTL XMIT D:\MVS\DASD\CBT357.XMI
SYS2.PL1.SAMPLES XMIT D:\MVS\DASD\CBT316.XMI

```

The corresponding output from the utility is the following:

```
D:\HERCULES>DASDLOAD -z D:\MVS\DASD\TST001.DAT D:\MVS\DASD\TST001.A80 1
Hercules DASD loader program Version 3.05
(c)Copyright 1999-2005 by Roger Bowler, Jan Jaeger, and others
----- TST001 3390-3 *
HHCDL006I Creating 3390 volume TST001: 15 trks/cyl, 56832 bytes/track
HHCDU044I Creating 3390 volume TST001: 3339 cyls, 15 trks/cyl, 56832 bytes/track
HHCDU041I 3339 cylinders successfully written to file D:\MVS\DASD\TST001.A80
HHCDL009I Loading 3390 volume TST001
----- SYS2.RACF.UTIL XMIT D:\MVS\DASD\CBT728.XMI
HHCDL012I Creating dataset SYS2.RACF.UTIL at cyl 0 head 1
HHCDL110I Processing file D:\MVS\DASD\CBT728.XMI

----- SYS2.BATCH.CNTL XMIT D:\MVS\DASD\CBT357.XMI
HHCDL012I Creating dataset SYS2.BATCH.CNTL at cyl 0 head 2
HHCDL110I Processing file D:\MVS\DASD\CBT357.XMI
----- SYS2.PL1.SAMPLES XMIT D:\MVS\DASD\CBT316.XMI
HHCDL012I Creating dataset SYS2.PL1.SAMPLES at cyl 10 head 0
HHCDL110I Processing file D:\MVS\DASD\CBT316.XMI
HHCDL057I VTOC starts at cyl 21 head 10 and is 1 track
HHCDL014I Free space starts at cyl 21 head 11
HHCDL016I Total of 3339 cylinders written to D:\MVS\DASD\TST001.A80
```

**Figure 40: DASDLOAD utility output (XMIT method, message level 1)**

### Sample 2:

Create a 3390 model 3 in a compressed file called TST001.A80 containing a library contained in a XMIT files using the control file TST001.DAT with message level 3.

The control file TST001.DAT looks like follows:

```
TST001 3390-3 *
SYS2.RACF.UTIL XMIT D:\MVS\DASD\CBT728.XMI
```

The corresponding output from the utility is the following:

```
D:\HERCULES>DASDLOAD -z D:\MVS\DASD\TST001.DAT D:\MVS\DASD\TST001.A80 3
Hercules DASD loader program Version 3.05
(c)Copyright 1999-2005 by Roger Bowler, Jan Jaeger, and others
----- TST001 3390-3 *
HHCDL006I Creating 3390 volume TST001: 15 trks/cyl, 56832 bytes/track
HHCDU044I Creating 3390 volume TST001: 3339 cyls, 15 trks/cyl, 56832 bytes/track
HHCDU041I 3339 cylinders successfully written to file D:\MVS\DASD\TST001.A80
HHCDL009I Loading 3390 volume TST001
----- SYS2.RACF.UTIL XMIT D:\MVS\DASD\CBT728.XMI
```

```

HHCDL012I Creating dataset SYS2.RACF.UTIL at cyl 0 head 1
HHCDL110I Processing file D:\MVS\DASD\CBT728.XMI
HHCDL078I File 1: DSNAME=SBGOLOB.CBT470.FILE728
HHCDL079I DSORG=PO RECFM=FB LRECL=80 BLKSIZE=5600 KEYLEN=0 DIRBLKS=6
HHCDL084I Original dataset: DSORG=PO RECFM=FB LRECL=80 BLKSIZE=5600 KEYLEN=0
HHCDL085I Dataset was unloaded from device type 3030200F (3390)
HHCDL086I Original device has 2226 cyls and 15 heads
HHCDL095I Member $$$#DATE TTR=000214 Userdata=044600000105311F .....
    0105311F0733000C 000C0000C3C2E360 F4F7F0404040 .....CBT-470
HHCDL095I Member $$NOTE1 TTR=00020E Userdata=010100160105311F .....
    0105311F07330009 000A0000E2C2C7D6 D3D6C2404040 .....SBGOLOB
HHCDL095I Member @FILE728 TTR=000210 Userdata=044600000105311F .....
    0105311F07330005 00050000C3C2E360 F4F7F0404040 .....CBT-470
HHCDL095I Member ICHPWX01 TTR=000109 Userdata=010100000105311F .....

    0105311F0711019A 019A0000D1D6E4E2 D4C140404040 .....JOUSMA
HHCDL090I End of directory
HHCDL013I Dataset SYS2.RACF.UTIL contains 1 track
HHCDL057I VTOC starts at cyl 0 head 2 and is 1 track
HHCDL014I Free space starts at cyl 0 head 3
HHCDL016I Total of 3339 cylinders written to D:\MVS\DASD\TST001.A80

```

**Figure 41: DASDLOAD utility output (XMIT method, message level 3)**

### Sample 3:

Create a 3390 model 3 in a compressed file called TST001.A80 containing a sequential file using the control file TST001.DAT with message level 5.

The control file TST001.DAT looks like follows:

```

TST001 3390-3 *
SYS2.SEQ.DATA SEQ D:\MVS\DASD\SEQ.BIN TRK 5 5 0 PS FB 80 3120

```

The corresponding output from the utility is the following:

```

D:\HERCULES>dasdload -z D:\MVS\DASD\TST001.DAT D:\MVS\DASD\TST001.A80 5
Hercules DASD loader program Version 3.05
(c)Copyright 1999-2005 by Roger Bowler, Jan Jaeger, and others
----- TST001 3390-3 *
HHCDL006I Creating 3390 volume TST001: 15 trks/cyl, 56832 bytes/track
HHCDU044I Creating 3390 volume TST001: 3339 cyls, 15 trks/cyl, 56832 bytes/track
HHCDU041I 3339 cylinders successfully written to file D:\MVS\DASD\TST001.A80
HHCDL009I Loading 3390 volume TST001
----- SYS2.SEQ.DATA SEQ D:\MVS\DASD\SEQ.BIN TRK 5 5 0 PS FB 80 3120
HHCDL012I Creating dataset SYS2.SEQ.DATA at cyl 0 head 1

```

```

HHCDL013I Dataset SYS2.SEQ.DATA contains 5 tracks
-----
HHCDL057I VTOC starts at cyl 0 head 6 and is 1 track
HHCDL058I Format 4 DSCB CCHHR=0000000601 (TTR=000001)
+0000 00000000 00000000 00000000 04040404 .....
+0010 04040404 04040404 04040404 04040404 .....
Line 0020 same as above
+0030 04040404 04040404 F4000000 0603002F .....4...../
+0040 0D0B0000 00008001 0000D0B 000FE5A2 .....Vs
+0050 00000030 0000322D 00000000 00000000 ...0..2-.....
+0060 00000000 00000000 00000000 00000000 .....
+0070 00000000 00010000 00000600 00000600 .....
+0080 00000000 00000000 00000000 00000000 .....
+0090 00000000 00000000 .....
HHCDL058I Format 5 DSCB CCHHR=0000000602 (TTR=000002)
+0000 00000000 00000000 00000000 05050505 .....

+0010 00000000 00000000 00000000 00000000 .....
Line 0020 same as above
+0030 00000000 00000000 F5000000 00000000 .....5.....
+0040 00000000 00000000 00000000 00000000 .....
Lines 0050 to 0080 same as above
+0090 00000000 00000000 .....
HHCDL058I Format 1 DSCB CCHHR=0000000603 (TTR=000003) SYS2.SEQ.DATA
+0000 00000000 00000000 00000000 E2E8E2F2 .....SYS2
+0010 4BE2C5D8 4BC4C1E3 C1404040 40404040 .SEQ.DATA
+0020 40404040 40404040 40404040 40404040
+0030 40404040 40404040 F1E3E2E3 F0F0F100 1TST001.
+0040 016A002C 00000001 4000C8C5 D9C3E4D3 .|.,.... .HERCUL

+0050 C5E24040 40404000 00000000 00004000 ES .....
+0060 90000C30 00500000 00A08000 00050000 ...0.&.....
+0070 02DCDE00 00010000 00000100 00000500 .....
+0080 00000000 00000000 00000000 00000000 .....
+0090 00000000 00000000 .....
HHCDL059I Format 0 DSCB CCHHR=0000000604 (TTR=000004)
HHCDL059I Format 0 DSCB CCHHR=0000000605 (TTR=000005)
.
several lines not displayed
.
HHCDL059I Format 0 DSCB CCHHR=0000000631 (TTR=000031)
HHCDL059I Format 0 DSCB CCHHR=0000000632 (TTR=000032)
HHCDL014I Free space starts at cyl 0 head 7
HHCDL016I Total of 3339 cylinders written to D:\MVS\DASD\TST001.A80
HHCDL017I Updating VTOC pointer 0000000601

```

**Figure 42: DASDLOAD utility output (SEQ method, message level 5)**

---

## 5.12 DASDLS (List datasets on a volume)

### 5.12.1 Function

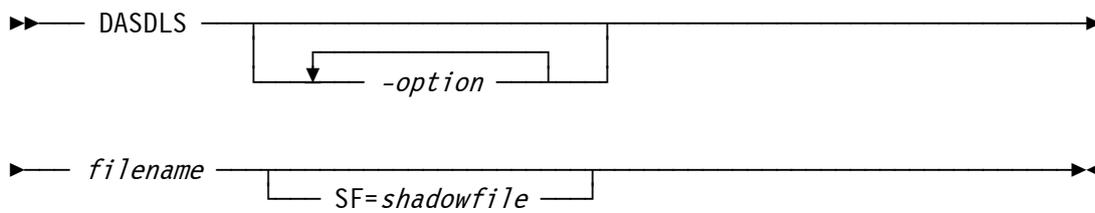
The DASDLS utility is used to list all datasets and their attributes on the specified DASD image file.

### 5.12.2 Syntax

#### Descriptive

```
DASDLS [-option [-option ... ]] filename [SF=shadowfile]
```

#### Diagram



### 5.12.3 Parameter

#### *options*

- info** Show Format 1 DSCB information. This option is implied if any of the other options are set.
- caldt** Display dates as YYYYMMDD. If this option is not specified then dates are displayed as YYDD.
- refdt** Display the last-referenced date (not applicable to MVT), otherwise it is omitted.
- expdt** Display the expiry date, otherwise it is omitted.
- hdr** Display column headers, otherwise they are omitted.
- dsnl[=*n*]** Reserve space on the output line for dataset names up to '*n*' characters. If '*n*' is not specified then space for 26 characters is reserved. If the `-dsnl` option is omitted, then space for 44 characters is reserved.
- yroffs[=*n*]** Add the year offset '*n*' (which may be negative) to dates before displaying them. If '*n*' is omitted then 28 is used. No checking is done that the value of '*n*' is sensible.

## **arguments**

- filename** Name of the DASD image file for which the datasets contained will be listed.
- shadowfile** Optional name of an associated shadow file (s) from which any datasets will be listed.

## **5.12.4 Examples**

### **Example 1:**

List all the datasets currently on the volume D:\MVS\DASD\MVSRES.CCKD.

```
DASDLS D:\MVS\DASD\MVSRES.CCKD
```

### **Example 2:**

List all the datasets currently on the volume D:\MVS\DASD\MVSRES.CCKD showing information from the Format 1 DSCB, with column headers and reserving only 20 characters for the dataset name.

```
DASDLS -dsnl=20 -hdr D:\MVS\DASD\MVSRES.CCKD
```

### **Example 3:**

List all the datasets currently on the volume D:\MVS\DASD\MVSRES.CCKD showing information from the Format1 DSCB, with column headers, calendar date format and reserving only 20 characters for the dataset name.

```
DASDLS -dsnl=20 -hdr -caldt D:\MVS\DASD\MVSRES.CCKD
```

## **5.12.5 Sample Utility Output**

### **Sample 1:**

List all the datasets currently on the volume D:\MVS\DASD\MVSRES.CCKD.

```
D:\Hercules>dasdls D:\MVS\DASD\MVSRES.CCKD
Hercules DASD list program Version 3.10.0
(c)Copyright 1999-2006 by Roger Bowler, Jan Jaeger, and others
D:\MVS\DASD\MVSRES.CCKD: VOLSER=MVSRES
SYS1.STAGE1.OUTPUT
SYS1.LPALIB
SYS1.LINKLIB
SYS1.SVCLIB
SYS1.NUCLEUS
SYS1.DCMLIB
SYS1.INDMAC
SYS1.CMDLIB
SYS1.HELP
SYS1.SAMPLIB
```

```

SYS1.MACLIB
SYS1.PROCLIB
SYS1.TELCMLIB
SYS1.UADS
SYS1.VTAMLIB
SYS1.IMAGELIB

```

.  
*several lines not displayed*  
.

```

SYS1.SMPCRQ
SYS1.SMPSCDS
SYS1.COBLIB
SYS1.FORTLIB
SYS1.PLLLIB
SYS1.SORTLIB

```

**Figure 43: DASDLS utility output (extended information)**

**Sample 2:**

List all the datasets currently on the volume D:\MVS\DASD\MVSRES.CCKD showing information from the Format1 DSCB, with column headers, expiration date, calendar date format with a year offset of 10 years and reserving only 20 characters for the dataset name.

```

D:\Hercules>dasdls -caldt -expdt -hdr -dsnl=20 -yroffs=10 D:\MVS\DASD\MVSRES.CCKD
Hercules DASD list program Version 3.10.0
(c)Copyright 1999-2010 by Roger Bowler, Jan Jaeger, and others
D:\MVS\DASD\MVSRES.CCKD: VOLSER=MVSRES
Dsnname          Created   Exp. Date  ORG  RECFM  LRECL  BLKSZ  Key   Trks%Use#Ext  2ndry_alloc
SYS1.STAGE1.OUTPUT 1983Apr28 ----- PS  FB      80 19040  0    52 98   2 TRK      30
SYS1.LPALIB        1983Apr28 1981Dec16 PO  U           19069  0    900 50   1 CYL      1
SYS1.LINKLIB        1983Apr28 1981Dec16 PO  U           19069  0   1200 38   1 CYL      1
SYS1.SVCLIB         1983Apr28 1981Dec16 PO  U           19069  0    60 24   1 CYL      1
SYS1.NUCLEUS        1983Apr28 1981Dec16 PO  U          19069 19069  0    480 12   1 CYL      0
SYS1.DCMLIB         1983Apr28 1981Dec16 PO  U           19069  0    120 1    1 CYL      0
SYS1.INDMAC         1983Apr28 1981Dec16 PO  FB      80 19040  0    180 1    1 CYL      1
SYS1.CMDLIB         1983Apr28 1981Dec16 PO  U           19069  0    180 46   1 CYL      1
SYS1.HELP           1983Apr28 1981Dec16 PO  FB      80 19040  0    180 33   1 CYL      1
SYS1.SAMPLIB        1983Apr28 1981Dec16 PO  F      80   80   0    240 100   1 CYL      1
SYS1.MACLIB         1983Apr28 1981Dec16 PO  FB      80 19040  0   1500 47   1 CYL      1
SYS1.PROCLIB        1983Apr28 1981Dec16 PO  FB      80 19040  0    180 5    1 CYL      1
SYS1.TELCMLIB       1983Apr28 1981Dec16 PO  U           19069  0    120 31   1 CYL      1
SYS1.UADS           1983Apr28 ----- PO  FB      80   800   0    30 16   1 CYL      1
SYS1.VTAMLIB        1983Apr28 1981Dec16 PO  U           19069  0    120 11   1 CYL      1
SYS1.IMAGELIB       1983Apr28 1981Dec16 PO  U           1024   0    60 3    1 CYL      0
SYS1.PARMLIB        1983Apr28 1981Dec16 PO  F      80   80   0    180 36   1 CYL      0
SYS1.BROADCAST      1983Apr28 ----- DA  F      129   1    30 100   1 CYL      0
SYS1.MANX           1983Apr28 1981Dec16 PS  VBS   4096 4096   0    150 51   1 CYL      0
SYS1.MANY            1983Apr28 1981Dec16 PS  VBS   4096 4096   0    150 0    1 CYL      0
SYS1.TCOMM          1983Apr28 1981Dec16 PO  FB      80 19040  0   2970 0    1 CYL     10
SYS1.DUMP00         1983Apr28 1981Dec16    U           0    0    900 0    1 CYL      0
SYS1.LOGREC         1983Apr28 ----- PS  U      40   40   0    30 97   1 TRK      0

```

SYS1.VTAMSRC	1983Apr29	1981Dec16	PO	FB	80	19040	0	30	6	1	CYL	1
SYS1.VTAMOBJ	1983Apr29	-----	PO	F	3152	3152	0	150	1	1	CYL	5
SYS1.VTAMLST	1983Apr29	-----	PO	FB	80	19040	0	150	1	1	CYL	5
SYS1.UMODMAC	1983Apr29	1981Dec16	PO	FB	80	19040	0	30	68	1	CYL	1
SYS1.UMODSRC	1983Apr29	1981Dec16	PO	FB	80	19040	0	30	28	1	CYL	1
SYS1.UMODOBJ	1983Apr29	-----	PO	FB	80	3120	0	30	3	1	CYL	1
SYS1.UMODLIB	1983Apr29	1981Dec16	PO	U		19069	0	30	2	1	CYL	1
SYS1.SMPCRQ	1983Apr29	-----	PO	FB	80	19040	0	26	42	1	BLK	200
SYS1.SMPSCDS	1983Apr29	-----	PO	FB	80	3120	0	30	7	1	CYL	1
SYS1.COBLIB	1983Apr29	1981Dec16	PO	U		19069	0	6	87	1	TRK	0
SYS1.FORTLIB	1983Apr29	1981Dec16	PO	U		19069	0	14	88	1	TRK	0
SYS1.PL1LIB	1983Apr29	1981Dec16	PO	U		19069	0	122	99	1	TRK	0
SYS1.SORTLIB	1983Apr29	1981Dec16	PO	U		19069	0	21	83	1	TRK	0

**Figure 44: DASDLS utility output (extended information)**

---

## 5.13 DASDPDSU (PDS unload utility)

### 5.13.1 Function

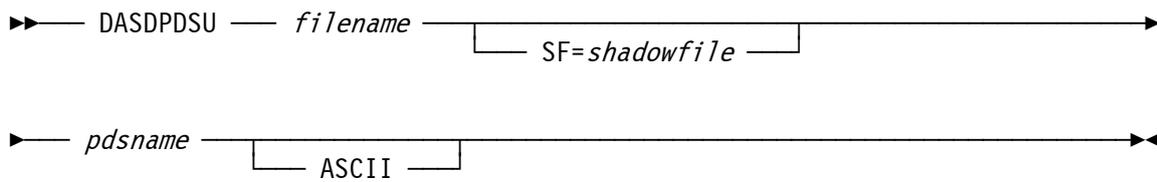
The DASDPDSU utility is a command that unloads PDS members from a DASD image and copies each member to a file *memname.mac* in the current working directory.

### 5.13.2 Syntax

#### Descriptive

```
DASDPDSU filename [SF=shadowfile] pdsname [ASCII]
```

#### Diagram



### 5.13.3 Parameter

- filename** Name of the DASD image file from which a PDS will be unloaded.
- shadowfile** Optional name of an associated shadow file.
- pdsname** Name of the partitioned dataset on the DASD image file from which the members will be unloaded.
- ASCII** If the optional keyword ASCII is specified the members will be unloaded as ASCII variable length text files. Otherwise the members are unloaded as fixed length EBCDIC binary files.

### 5.13.4 Examples

#### Example 1:

Unload the dataset "SYS1.PARMLIB" on DASD image file MVSRES.CCKD in ASCII format.

```
DASDPDSU D:\MVS\DASD\MVSRES.CCKD SYS1.PARMLIB ASCII
```

## 5.13.5 Sample Utility Output

### Sample 1:

Unload the dataset "SYS1.PARMLIB" on DASD image file MVSRES.CCKD in ASCII format.

```
D:\Hercules>dasdpdsu D:\MVS\DASD\MVSRES.CCKD SYS1.PARMLIB ASCII
Hercules PDS unload program Version 3.05
(c)Copyright 1999-2006 by Roger Bowler, Jan Jaeger, and others
Reading directory block at cyl 198 head 0 rec 1
Member COMMND00 TTR=001A27
Member GTFPARM TTR=001A33
Member IEAABD00 TTR=001A35
Member IEAAPF00 TTR=001A29
Member IEABLD00 TTR=001B0C
Member IEADMP00 TTR=001A31
Reading directory block at cyl 198 head 0 rec 2
Member IEADMR00 TTR=001A2F
Member IEAIPS00 TTR=000112
Member IEALOD00 TTR=000712
Member IEAOPT00 TTR=000126
Member IEAPAK00 TTR=00062E
Member IEASYS00 TTR=002D2E
Member IPCSPR00 TTR=00041B
Member IRBMF100 TTR=001B1E
Member JES2PARM TTR=003C07
Reading directory block at cyl 198 head 0 rec 3
Member LNKLST00 TTR=001804
Member MVIKEY00 TTR=001B12
Member RPFKEY00 TTR=000B42
Member SMFPRM00 TTR=001B3B
Member TSOKEY00 TTR=001B14
Member VATLST00 TTR=001B45
End of directory
```

**Figure 45: DASDPDSU utility output**

---

## 5.14 DASDSEQ (Display sequential datasets)

### 5.14.1 Function

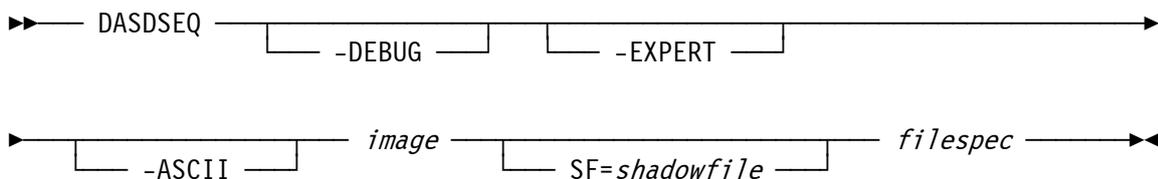
The DASDSEQ utility is used to extract and list sequential datasets from a DASD image file.

### 5.14.2 Syntax

#### Descriptive

```
DASDSEQ [-DEBUG] [-EXPERT] [-ASCII] image [SF=shadowfile] filespec
```

#### Diagram



### 5.14.3 Parameter

#### *options:*

- DEBUG**            Display additional debug options.
- EXPERT**         Display an additional help panel that describes the expert operands.
- ASCII**           Translate the output file to ASCII and trim trailing blanks.

#### *arguments:*

- image***            The file name (and optionally path) of the DASD image file.
- shadowfile***      The optional name (and path) of an associated shadow file.
- filespec***         The name of a sequential dataset (DSORG=PS) on the DASD image file. This name is also used as filename for the extracted data.

All expert facilities (as described in the -EXPERT help panel) are experimental and are therefore not yet described here.

## 5.14.4 Examples

### Example 1:

Extract and list the sequential dataset SYS2.SEQ.DATA on the DASD image file TST001.AA0.

```
DASDSEQ -ASCII D:\MVS\DASD\TST001.AA0 D:\MVS\DASD\SYS2.SEQ.DATA
```

## 5.14.5 Sample Utility Output

### Sample 1:

Extract and list the sequential dataset SYS2.SEQ.DATA on the DASD image file TST001.AA0.

```
D:\HERCULES>dasdseq -ascii D:\MVS\DASD\TST001.AA0 SYS2.SEQ.DATA
dasdseq 3.05 Copyright 1999-2005 Roger Bowler
Portions Copyright 2001-2005 James M. Morrison
dasdseq wrote 11 records to SYS2.SEQ.DATA
```

**Figure 46: DASDSEQ utility output**

---

## 5.15 HETGET (Extract files from an AWS or HET tape file)

### 5.15.1 Function

The HETGET utility extracts a file from an AWSTAPE or HET tape file. The output is in binary format and has to be displayed with an EBCDIC editor on the PC or has to be uploaded to the host operating system and opened with a mainframe editor.

### 5.15.2 Syntax

#### Descriptive

```
HETGET tapefile outfile filenum
```

#### Diagram

```
▶— HETGET — tapefile — outfile — filenum —▶
```

### 5.15.3 Parameter

- tapefile***            The name (and optionally the path) of the AWSTAPE or HET tape file from which a file should be extracted.
- outfile***            The name (and optionally the path) of the file to which the extracted file will be written.
- filenum***            The file number on the tape of the file to be extracted from the tape.

### 5.15.4 Examples

#### Example 1:

Extract file number 25 from the AWSTAPE file named T001003.AWS and write it to file FILE#25.BIN in directory D:/S390/UNLOAD/.

```
HETGET D:/MVS/TAPE/T001003.AWS D:/S390/UNLOAD/FILE#25.BIN 25
```

## 5.15.5 Sample Utility Output

### Sample 1:

Extract file number 10 from the AWSTAPE file named T38321A.AWS and write it to file FILE10.BIN.

```
D:\HERCULES>hetget D:/MVS/TAPE/T38321A.AWS D:/FILE10.BIN 10
Hercules HET extract files program Version 3.05
(c)Copyright 1999-2005 by Roger Bowler, Jan Jaeger, and others
File Info:
  DSN=BBC.BBC53XX.F5
DCB Attributes used:
  RECFM=V      LRECL=32756  BLKSIZE=32760
```

**Figure 47: HETGET utility output**

---

## 5.16 HETINIT (Initialize an AWS or HET tape file)

### 5.16.1 Function

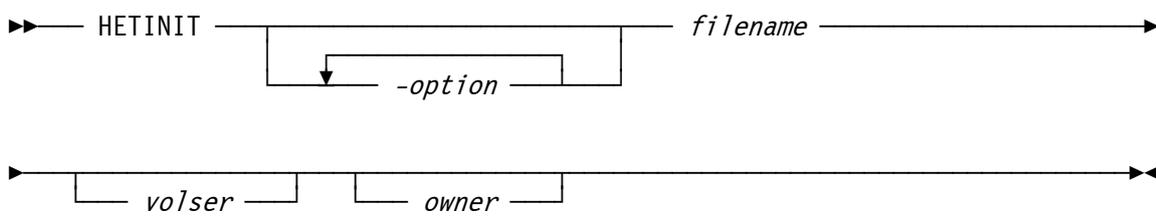
The HETINIT utility is used to initialize a tape file.

### 5.16.2 Syntax

#### Descriptive

```
HETINIT [-option [-option ... ]] filename [volser] [owner]
```

#### Diagram



### 5.16.3 Parameter

#### **options:**

- d** Disable compression (creates an AWSTAPE file).
- h** Display usage summary (help text).
- i** Create an IEHINITT formatted tape (set on by default).
- n** Create a NL (non labeled) tape.

#### **arguments:**

- filename** Name (and optionally path) of the tape file to be created.
- volser** The volser of the tape.
- owner** The owner of the tape.

## 5.16.4 Examples

### Example 1:

Create a new AWS tape file named NEWTAPE.AWS with volser NEWTAP and owner HERCULES in the directory D:/MVS/TAPE/.

```
HETINIT -d D:/MVS/TAPE/NEWTAPE.AWS NEWTAP HERCULES
```

## 5.16.5 Sample Utility Output

### Sample 1:

Create a new HET tape file named NEWTAPE.HET with volser NEWTAP and owner HERCULES in the directory D:/MVS/TAPE/.

```
D:\HERCULES>hetinit D:/MVS/TAPE/NEWTAPE.HET NEWTAP HERCULES
Hercules HET IEHINITT program Version 3.05
(c)Copyright 1999-2005 by Roger Bowler, Jan Jaeger, and others
```

**Figure 48: HETINIT utility output**

---

## 5.17 HETMAP (Show information about a HET or AWS tape file)

### 5.17.1 Function

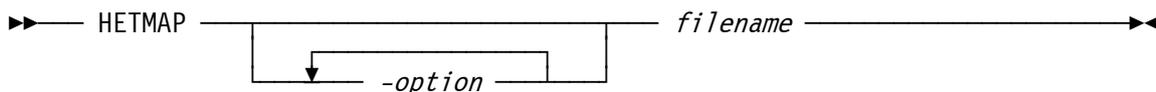
The Hercules HETMAP program displays information about a HET or an AWSTAPE tape file. By default, HETMAP shows label and file information. Optionally, HETMAP may be used to display data set information, or to display label information in a format similar to the output of TAPEMAP. HETMAP shows information about the tape and files on it, and does not display actual file contents.

### 5.17.2 Syntax

#### Descriptive

HETMAP [-option [-option ... ]] filename

#### Diagram



### 5.17.3 Parameter

#### *options:*

- a Print all label and file information (default).
- bn Print 'n' bytes per file. 'n' must be numeric. Option '-b' implies '-s'.
- d Print only dataset information.
- f Print only file information.
- h Display usage summary.
- l Print only label information.
- s Print dump of each data file (SLANAL format).
- t Print TAPEMAP-compatible format output.

Options are mutually exclusive. If more than one option is entered, the last one is used. If no options are specified, '-a' is the default.

**arguments:**

**filename**           The name (and optionally path) of the HET or AWS tape file about which information is to be displayed.

### 5.17.4 Examples

**Example 1:**

Show information about a HET tape file.

```
HETMAP D:/MVS/TAPE/RPF142.HET
```

### 5.17.5 Sample Utility Output

**Sample 1:**

Show information about a HET tape file (print all label and file information).

```
D:\Hercules>hetmap -a G:\MVS\TAPE\RPF142.HET
Hercules HET and AWS tape map program Version 3.0.7
(c)Copyright 1999-2009 by Roger Bowler, Jan Jaeger, and others
-----
Filename               : G:\MVS\TAPE\RPF142.HET
-----
Label                 : 'VOL1'
Volume Serial         : 'RPF142'
Improved Data Rec.   : ' '
Owner Code            : ' '
-----
Label                 : 'HDR1'
Dataset ID            : 'RPF.INST.SRPFASM '
Volume Serial         : 'RPF142'
Volume Sequence       : '0001'
Dataset Sequence      : '0001'
GDG Number            : ' '
GDG Version           : ' '
Creation Date         : ' 74186'
Expiration Date       : ' 00000'
Dataset Security      : '0'
Block Count Low       : '000000'
System Code           : 'IBM OS/VS 370'
Block Count High      : ' '
-----
Label                 : 'HDR2'
Record Format          : 'V'
Block Size            : '19060'
Record Length         : '19056'
Density               : '4'
Dataset Position      : '0'
Job/Step ID           : 'HERC01 /UNLOAD '
Recording Technique   : ' '
```

```

Control Character : ' '
Block Attribute  : 'S'
Device Serial    : ' 30001'
Checkpoint ID    : ' '
Large Block Length : '          '
-----
File #          : 1
Blocks         : 3
Min Blocksize  : 80
Max Blocksize  : 80
Uncompressed bytes : 240
Min Blocksize-Comp : 23
Max Blocksize-Comp : 68
Compressed bytes : 145
-----
File #          : 2
Blocks         : 196
Min Blocksize  : 20
Max Blocksize  : 19060
Uncompressed bytes : 2907140
Min Blocksize-Comp : 20
Max Blocksize-Comp : 4739
Compressed bytes : 579020
-----
Label           : 'EOF1'
Dataset ID      : 'RPF.INST.SRPFASM '
Volume Serial   : 'RPF142'
Volume Sequence : '0001'
Dataset Sequence : '0001'
GDG Number     : ' '
GDG Version    : ' '
Creation Date   : ' 74186'
Expiration Date : ' 00000'
Dataset Security : '0'
Block Count Low : '000196'
System Code     : 'IBM OS/VS 370'
Block Count High : ' '
-----
Label           : 'EOF2'
Record Format    : 'V'
Block Size      : '19060'
Record Length   : '19056'
Density         : '4'
Dataset Position : '0'
Job/Step ID     : 'HERC01 /UNLOAD '
Recording Technique : ' '
Control Character : ' '
Block Attribute  : 'S'
Device Serial    : ' 30001'
Checkpoint ID    : ' '
Large Block Length : '          '
-----
.
```

```

.
.
-----
File #           : 16
Blocks           : 0
Min Blocksize   : 0
Max Blocksize   : 0
Uncompressed bytes : 0
Min Blocksize-Comp : 0
Max Blocksize-Comp : 0
Compressed bytes : 0
-----
Summary         :
Files           : 16
Blocks         : 425
Uncompressed bytes : 3701156
Compressed bytes : 913267
Reduction      : 2787889

```

**Figure 49: HETMAP utility output (print all label and file information)**

**Sample 2:**

Show information about a HET tape file (print only dataset information).

```

G:\Hercules>hetmap -d G:\MVS\TAPE\RPF142.HET
Hercules HET and AWS tape map program Version 3.0.7
(c)Copyright 1999-2009 by Roger Bowler, Jan Jaeger, and others
vol=RPF142          owner=

seq=1              file#=2
dsn=RPF.INST.SRPFASM  crtdt=1974.186  expdt=0000.000  blocks=196
job=HERC01 /UNLOAD  recfm=VS          lrecl=19056     blksize=19060

seq=2              file#=5
dsn=RPF.INST.SRPF0BJ  crtdt=1974.186  expdt=0000.000  blocks=126
job=HERC01 /UNLOAD  recfm=VS          lrecl=3136     blksize=3140

seq=3              file#=8
dsn=RPF.INST.SRPFHELP crtdt=1974.186  expdt=0000.000  blocks=8
job=HERC01 /UNLOAD  recfm=VS          lrecl=19056     blksize=19060

seq=4              file#=11
dsn=RPF.INST.SRPFLOAD crtdt=1974.186  expdt=0000.000  blocks=55
job=HERC01 /UNLOAD  recfm=VS          lrecl=19085     blksize=19089

seq=5              file#=14
dsn=RPF.INST.CNTL    crtdt=1974.186  expdt=0000.000  blocks=19
job=HERC01 /UNLOAD  recfm=VS          lrecl=19056     blksize=19060

```

**Figure 50: HETMAP utility output (print only dataset information)**

### Sample 3:

Show information about a HET tape file in TAPEMAP format.

```
D:\Hercules>hetmap -t D:\MVS\TAPE\RPF142.HET
Hercules HET and AWS tape map program Version 3.0.7
(c)Copyright 1999-2009 by Roger Bowler, Jan Jaeger, and others
VOL1RPF1420
HDR1RPF.INST.SRPFASM RPF14200010001      74186 0000000000000000IBM OS/V5 370
HDR2V190601905640HERC01 /UNLOAD      S  30001
File 1: Blocks=3, block size min=80, max=80
File 2: Blocks=196, block size min=20, max=19060
EOF1RPF.INST.SRPFASM RPF14200010001      74186 0000000000196IBM OS/V5 370
EOF2V190601905640HERC01 /UNLOAD      S  30001
File 3: Blocks=2, block size min=80, max=80
HDR1RPF.INST.SRPF0BJ RPF14200010002      74186 0000000000000000IBM OS/V5 370
HDR2V031400313640HERC01 /UNLOAD      S  30001
File 4: Blocks=2, block size min=80, max=80
File 5: Blocks=126, block size min=60, max=3140
EOF1RPF.INST.SRPF0BJ RPF14200010002      74186 0000000000126IBM OS/V5 370
EOF2V031400313640HERC01 /UNLOAD      S  30001
File 6: Blocks=2, block size min=80, max=80
HDR1RPF.INST.SRPFHELPRPF14200010003      74186 0000000000000000IBM OS/V5 370
HDR2V190601905640HERC01 /UNLOAD      S  30001
File 7: Blocks=2, block size min=80, max=80
File 8: Blocks=8, block size min=60, max=11232
EOF1RPF.INST.SRPFHELPRPF14200010003      74186 00000000000008IBM OS/V5 370
EOF2V190601905640HERC01 /UNLOAD      S  30001
File 9: Blocks=2, block size min=80, max=80
HDR1RPF.INST.SRPFLOADRPF14200010004      74186 0000000000000000IBM OS/V5 370
HDR2V190891908540HERC01 /UNLOAD      S  30001
File 10: Blocks=2, block size min=80, max=80
File 11: Blocks=55, block size min=20, max=18864
EOF1RPF.INST.SRPFLOADRPF14200010004      74186 00000000000055IBM OS/V5 370
EOF2V190891908540HERC01 /UNLOAD      S  30001
File 12: Blocks=2, block size min=80, max=80
HDR1RPF.INST.CNTL RPF14200010005      74186 0000000000000000IBM OS/V5 370
HDR2V190601905640HERC01 /UNLOAD      S  30001
File 13: Blocks=2, block size min=80, max=80
File 14: Blocks=19, block size min=60, max=10672
EOF1RPF.INST.CNTL RPF14200010005      74186 00000000000019IBM OS/V5 370
EOF2V190601905640HERC01 /UNLOAD      S  30001
File 15: Blocks=2, block size min=80, max=80
File 16: Blocks=0, block size min=0, max=0
End of tape.
```

**Figure 51: HETMAP utility output (TAPEMAP compatible output)**

---

## 5.18 HETUPD (Update and/or copy an AWS or HET tape file)

### 5.18.1 Function

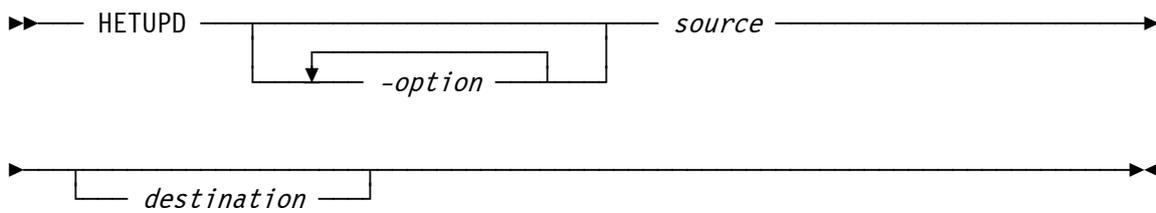
The HETUPD utility updates and/or copies emulated tape files.

### 5.18.2 Syntax

#### Descriptive

HETUPD [-option [-option ... ]] source [destination]

#### Diagram



### 5.18.3 Parameter

#### *options:*

- 1 ... -9**            Compression level (1=compress fast, 9=compress best).
- b**                    Use bzlib compression.
- c n**                Set chunk size to 'n'.
- d**                   Decompress source tape file.
- h**                   Display usage summary (help text).
- r**                   Rechunk tape file.
- s**                   Strict AWSTAPE specification (chunksize=4096, no compression).
- v**                   Verbose information (display usage summary).
- z**                   Use zlib compression.

**arguments:**

**source**           Name (and optionally path) of the source tape file.

**destination**      Name (and optionally path) of the destination tape file.

## 5.18.4 Examples

### Example 1:

Create an AWSTAPE file from an existing HET tape file.

```
HETUPD -s D:/MVS/TAPE/T001002.HET D:/MVS/TAPE/T001002.AWS
```

### Example 2:

Change the compression level to 2 with BZLIB compression and create a new HET tape file.

```
HETUPD -2 -b D:/MVS/TAPE/TLEV009.HET D:/MVS/TAPE/TLEV002.HET
```

## 5.18.5 Sample Utility Output

### Sample 1:

Change the compression level to 4 (best match between speed and compression rate) and create a new HET tape file.

```
D:\HERCULES>hetupd -4 D:/MVS/TAPE/T38321A.HET D:/MVS/TAPE/T38321B.HET
Hercules HET copy/update program Version 3.05
(c)Copyright 1999-2005 by Roger Bowler, Jan Jaeger, and others
```

**Figure 52: HETUPD utility output**

---

## 5.19 TAPECOPY (Copy a SCSI tape to or from an AWSTAPE disk file)

### 5.19.1 Function

The Hercules tape copy program copies a SCSI tape to or from an AWSTAPE disk file. TAPECOPY reads a SCSI tape and outputs an AWSTAPE file representation of the tape, or reads an AWSTAPE file and creates an identical copy of its contents on a tape mounted on a SCSI tape drive.

If the input file is a SCSI tape it is read and processed until physical EOD (end-of-data) is reached. That is, it does not stop whenever multiple tapemarks or filemarks are read rather it continues processing until the SCSI tape drive says there is no more data on the tape.

The resulting AWSTAPE output disk file may be specified for the filename on a Hercules tape device definition statement. It can then be used in order for the Hercules guest O/S to read the exact same data without having a SCSI tape drive physically attached to the host system. This allows you to easily transfer SCSI tape data to other systems that may not have SCSI tape drives attached to them.

The possible return codes and their meaning are:

- 0 Successful completion.
- 1 Invalid arguments or no arguments given.
- 3 Unable to open SCSI tape drive device file.
- 4 Unable to open AWSTAPE disk file.
- 5 Unrecoverable I/O error setting variable length block processing for SCSI tape device.
- 6 Unrecoverable I/O error rewinding SCSI tape device.
- 7 Unrecoverable I/O error obtaining status of SCSI device.
- 8 Unrecoverable I/O error reading block header from AWSTAPE disk file.
- 9 Unrecoverable I/O error reading data block.
- 10 AWSTAPE block size too large.
- 11 Unrecoverable I/O error writing tapemark.
- 12 Unrecoverable I/O error writing block header to AWSTAPE disk file.
- 13 Unrecoverable I/O error writing data block.

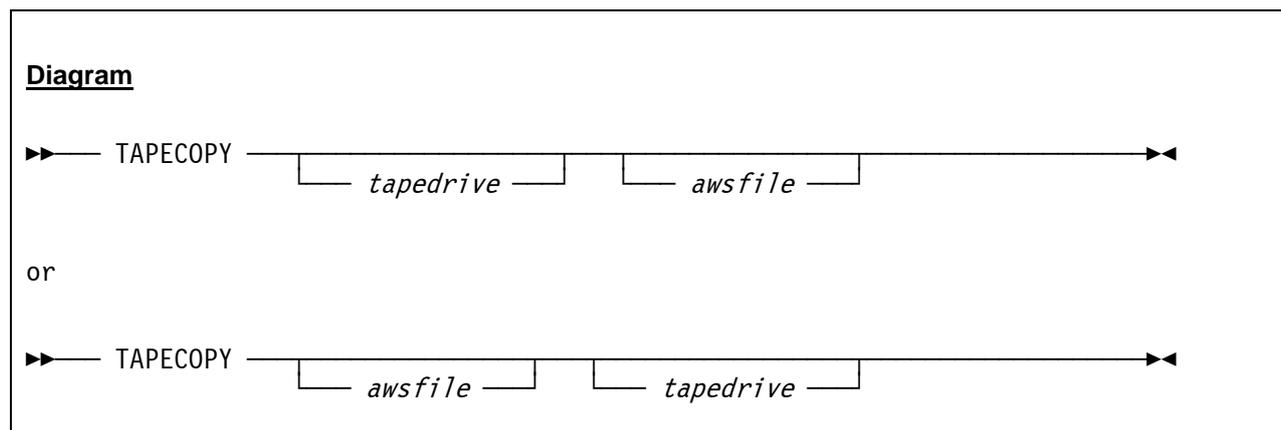
### 5.19.2 Syntax

#### Descriptive

```
TAPECOPY [tapedrive] [awsfile]
```

or

```
TAPECOPY [awsfile] [tapedrive]
```



### 5.19.3 Parameter

**tapedrive** Specifies the device filename of the SCSI tape drive. Must begin with */dev* to be recognized.

**awsfile** Specifies the filename of the AWSTAPE disk file.

The first filename is the input; the second one is the output.

### 5.19.4 Examples

#### Example 1:

Copy a SCSI file to an AWSTAPE disk file.

```
TAPECOPY /dev/nst0/scsifile.bin D:/MVS/TAPE/T3832AA.AWS
```

### 5.19.5 Sample Utility Output

#### Sample 1:

Copying an AWSTAPE disk file to a SCSI drive (this sample abends because the SCSI drive does not exist).

```
D:\HERCULES>tapecopy D:/T38321A.AWS /dev/nts0/scsifile.bin
Hercules tape copy program Version 3.05
(c)Copyright 1999-2005 by Roger Bowler, Jan Jaeger, and others
HHCTC001E Error opening /dev/nts0/scsifile.bin: errno=22: Invalid argument
HHCTC000I Abnormal termination
```

**Figure 53: TAPECOPY utility output**

---

## 5.20 TAPEMAP (Show information about an AWS tape file)

### 5.20.1 Function

The TAPEMAP program shows information about an AWS tape. It displays all header data contained on the tape. Please note that the utility does not display the actual contents of the files. Although the TAPEMAP utility does display the same data as the HETMAP utility, the output is formatted in a totally different way.

TAPEMAP produces correct output only for an AWS format tape. Use of TAPEMAP with any tape format other than AWS produces unpredictable results. Do not use it for HET tapes. The HETMAP program displays information about both AWS and HET tapes, and should be used instead of TAPEMAP.

### 5.20.2 Syntax

#### Descriptive

TAPEMAP *filename*

#### Diagram

▶— TAPEMAP — *filename* —▶

### 5.20.3 Parameter

***filename***            The name (and optionally path) of the AWS tape whose header data is to be displayed.

### 5.20.4 Examples

#### Example 1:

Print map of an AWS tape file.

```
TAPEMAP D:/MVS/TAPE/T38321A.AWS
```

## 5.20.5 Sample Utility Output

### Sample 1:

Print map of an AWS tape file.

```
D:\Hercules>tapemap G:\MVS\TAPE\38321A.AWS
Hercules tape map program Version 3.0.7
(c)Copyright 1999-2009 by Roger Bowler, Jan Jaeger, and others
File 1: Blocks=614, block size min=800, max=800
File 2: Blocks=38, block size min=6272, max=6272
File 3: Blocks=90, block size min=160, max=800
File 4: Blocks=29, block size min=720, max=800
File 5: Blocks=113, block size min=80, max=800
File 6: Blocks=1581, block size min=800, max=800
File 7: Blocks=597, block size min=160, max=800
File 8: Blocks=181, block size min=320, max=800
File 9: Blocks=126, block size min=640, max=640
File 10: Blocks=11, block size min=560, max=800
File 11: Blocks=512, block size min=800, max=800
File 12: Blocks=0, block size min=0, max=0
End of tape.
```

**Figure 54: TAPEMAP utility output (map AWS tape file)**

---

## 5.21 TAPESPLT (Split an AWS tape file)

### 5.21.1 Function

The TAPESPLT utility allows it to split an AWSTAPE file into a new AWSTAPE file. The number of the files to be copied defines the split point.

### 5.21.2 Syntax

#### Descriptive

TAPESPLT *infile outfile count*

#### Diagram

▶— TAPESPLT — *infile* — *outfile* — *count* —▶

### 5.21.3 Parameter

- infile***            The name of the input AWSTAPE disk file.
- outfile***           The name of the output AWSTAPE disk file.
- count***             The number of files that have to be copied to the output file.

### 5.21.4 Examples

#### Example 1:

Copy the first three files from AWSTAPE file TOLDTAP.AWS to the new AWSTAPE file TNEWTAP.AWS.

```
TAPESPLT D:/MVS/TAPE/TOLDTAP.AWS D:/MVS/TAPE/TNEWTAP.AWS
```

## 5.21.5 Sample Utility Output

### Sample 1:

Copy the first 10 files from AWSTAPE file T38321A.AWS to T4352AA.AWS.

```
D:\HERCULES>tapesplt D:/MVS/TAPE/T38321A.AWS D:/MVS/TAPE/T4352AA.AWS 10
Hercules tape split program Version 3.05
(c)Copyright 1999-2005 by Roger Bowler, Jan Jaeger, and others
Writing output file D:/MVS/TAPE/T4352AA.AWS.
VOL138321A
HDR1DOCLIB          38321A00010001      0030140000000000000000IBM OS/VS 370
HDR2V327523273600STP321VA/DOCLIB P   S   68216
File 1: Blocks=3, block size min=80, max=80
File 2: Blocks=16, block size min=20, max=32740
EOF1DOCLIB          38321A00010001      0030140000000000000016IBM OS/VS 370
EOF2V327523273600STP321VA/DOCLIB P   S   68216
File 3: Blocks=2, block size min=80, max=80
HDR1RIMLIB          38321A00010002      0030140000000000000000IBM OS/VS 370
HDR2V327523273600STP321VA/RIMLIB P   S   68216
File 4: Blocks=2, block size min=80, max=80
File 5: Blocks=47, block size min=20, max=5140
EOF1RIMLIB          38321A00010002      0030140000000000000047IBM OS/VS 370
EOF2V327523273600STP321VA/RIMLIB P   S   68216
File 6: Blocks=2, block size min=80, max=80
HDR1HOLDDATA        38321A00010003      0030140000000000000000IBM OS/VS 370
HDR2F031200008000STP321VA/HOLDDATA P   B   68216
File 7: Blocks=2, block size min=80, max=80
File 8: Blocks=794, block size min=3040, max=3120
EOF1HOLDDATA        38321A00010003      0030140000000000000794IBM OS/VS 370
EOF2F031200008000STP321VA/HOLDDATA P   B   68216
File 9: Blocks=2, block size min=80, max=80
HDR1PGMDIR          38321A00010004      0030140000000000000000IBM OS/VS 370
HDR2V327521065600STP321VA/PGMDIR P   S   68216
File 10: Blocks=2, block size min=80, max=80
```

Figure 55: TAPESPLT utility output

---

## 5.22 DMAP2HRC (P/390 DEVMAP conversion program)

### 5.22.1 Function

The P/390 DEVMAP to Hercules conversion program reads an IBM P/390 device map file, extracts the device definitions, and writes these as a valid Hercules configuration file to the standard output.

### 5.22.2 Syntax

#### Descriptive

DMAP2HRC *filename*

#### Diagram

▶— DMAP2HRC — *filename* —▶

### 5.22.3 Parameter

*filename*            The name of the DEVMAP input file containing P/390 device statements.

### 5.22.4 Examples

#### Example 1:

Convert the P/390 DEVMAP file DEVMAP.DVC to a Hercules configuration file named HERCULES.CNF.

```
D:/HERCULES/DMAP2HRC D:/S390/CONF/DEVMAP.DVC >D:/S390/CONF/HERCULES.CNF
```

### 5.22.5 Sample Utility Output

With the exception of error messages (if any) the utility does not produce any screen output.

---

## 5.23 VMFPLC2 (VM formatted tape utility)

### 5.23.1 Function

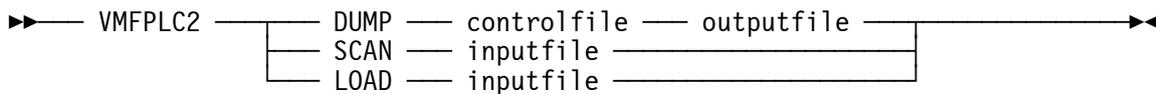
The VMFPLC2 utility is used to manipulate (create and read) VMFPLC2 (VM) formatted tape files for VM/370 use. The utility requires a function (dump, scan, load) followed by the name(s) of the files to be processed.

### 5.23.2 Syntax

#### Descriptive

```
VMFPLC2 {DUMP controlfile outputfile | SCAN inputfile |  
        LOAD inputfile}
```

#### Diagram



### 5.23.3 Parameter

<b>DUMP</b>	The dump function is used to create a VMFPLC2 formatted tape.
<b>SCAN</b>	The scan function is used to list the contents of a VMFPLC2 formatted tape.
<b>LOAD</b>	The load function is used to import the contents of a VMFPLC2 formatted tape onto the system.
<b><i>controlfile</i></b>	This specifies the name of a control file. The control file allows the dump function to determine what files to dump and how they should be interpreted on VM.
<b><i>outputfile</i></b>	This specifies the name of the output tape file.
<b><i>inputfile</i></b>	This specifies the name of the input tape file.

### 5.23.4 Control File

Each line of the control file has the following format:

```
filename filetype filemode recfm lrecl type tapefile
```

- filename** This is the 1 to 8 character name that represents the file name. The file name can be specified as lower case, but will be translated to upper case to follow CMS conventions. Allowed characters are [A-Z], [0-9], '\$' (dollar), '#' (pound), '@' (at), '+' (plus), '-' (hyphen), ':' (colon) and '\_' (underscore).
- filetype** This is the 1 to 8 character name that represents the file type. The file type can be specified as lower case, but will be translated to upper case to follow CMS conventions. Allowed characters are [A-Z], [0-9], '\$' (dollar), '#' (pound), '@' (at), '+' (plus), '-' (hyphen), ':' (colon) and '\_' (underscore).
- filemode** This is the 1 to 2 character that represents the file mode. The first character is a letter from A to Z and represents the "original" file mode when scanned on VM/370 (it does not force the file to be loaded on a certain disk). The second character is a digit from 0 to 6. The file mode number indicates specific behaviour for the file under CMS.
- recfm** Indicates the record format and should be 'F' (for Fixed) or 'V' (for Variable).
- lrecl** Indicates the logical record length. This should only be specified for RECFM F files.
- type** Indicates how the file is processed before being written to tape. The value can either be 'B' (Binary), 'T' (Text), or 'S' (Structured).
- Binary files are not translated. For RECFM F files, the file is cut into records of the size of the logical record length specified. For RECFM V files, the file is cut into records of 65535 bytes except for the last record which has a length of the remainder of the file.
- Textual files are translated from ASCII to EBCDIC and the trailing line termination character is removed. For RECFM F files the record may be truncated or padded with EBCDIC x'40' characters (white space, blanks). For RECFM V files, each record represents the length of the line up to 65535 characters.
- Structured files contain structured information which indicate the 16 bit length of each record in big endian format. For RECFM F files the record may be truncated or padded. For RECFM V files the records are stored as is.
- tapefile** The tape file is created or read as a HET (Hercules Emulated Tape) format. The DUMP function creates a file, while SCAN and LOAD functions only read the file.

### **5.23.5 Examples**

#### **Example 1:**

List the contents of a VMFPLC2 formatted tape.

```
VMFPLC2 SCAN D:/VM/TAPE/VMFPLC.HET
```

#### **Example 2:**

Import the contents of a VMFPLC2 formatted tape onto the system.

```
VMFPLC2 LOAD D:/VM/TAPE/VMFPLC.HET
```

#### **Example 3:**

Create a VMFPLC2 formatted tape.

```
VMFPLC2 DUMP D:/VM/TAPE/VMFPLC2.CTL D:/VM/TAPE/VMFPLC.HET
```

---

## Part III: Third-Party Utilities

---

## 6. Third Party Utilities

---

### 6.1 FTAPE (SCSI Tape Test Utility)

#### 6.1.1 Function

FTAPE is a command-line utility that allows to easily test the proper functioning of a Windows SCSI attached tape drive to make sure it's working properly before trying to add it to the Hercules device configuration. It supports functions like read and write, random or fixed-sized blocks of random data, fast-forward to the next or previous tape mark, back-space and forward-space blocks, write tape marks, locate blocks, etc. It will also display the data block it reads in either ASCII or EBCDIC to examine the data on an already existing tape.

It requires to already have the tape drive properly installed on the Windows system so that Windows can 'see' it (i.e. the SCSI adapter and SCSI tape device drivers are already installed such that device 'Tape0' appears in the Windows device list). When Windows can see the drive and FTAPE can access it, then the drive should be usable by Hercules.

Usually any "true" (non-ASPI) SCSI Tape device driver can be used, regardless of the tape drive model since all SCSI tape devices are required to support a minimum set of SCSI commands. The only exception found is IBM's device drivers. They are purposely coded to only work with their own tape drives unfortunately.

Also note that while you need to use a non-ASPI driver in order for the drive to work with Hercules, that sometimes the ASPI software that comes with a tape drive also includes a device-driver for the tape drive itself, such that by installing whatever ASPI software may come with the tape drive, the needed non-ASPI device-driver also gets installed. In other words, some ASPI software packages include not only a control DLL that allows their software to talk to the tape drive via ASPI, but also includes the necessary device-driver for the tape drive itself.

Further note that FTAPE does not currently support doing any type of I/O to the medium changer device (i.e. the cartridge loader). It is no problem if the drive has one, it just cannot be directly accessed by FTAPE or Hercules.

#### 6.1.2 Syntax

##### Descriptive

```
FTAPE [-f filename] function
```

where *function* can be:

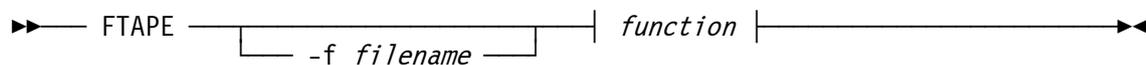
LOAD

UNLOAD

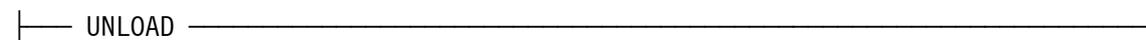
LOCK

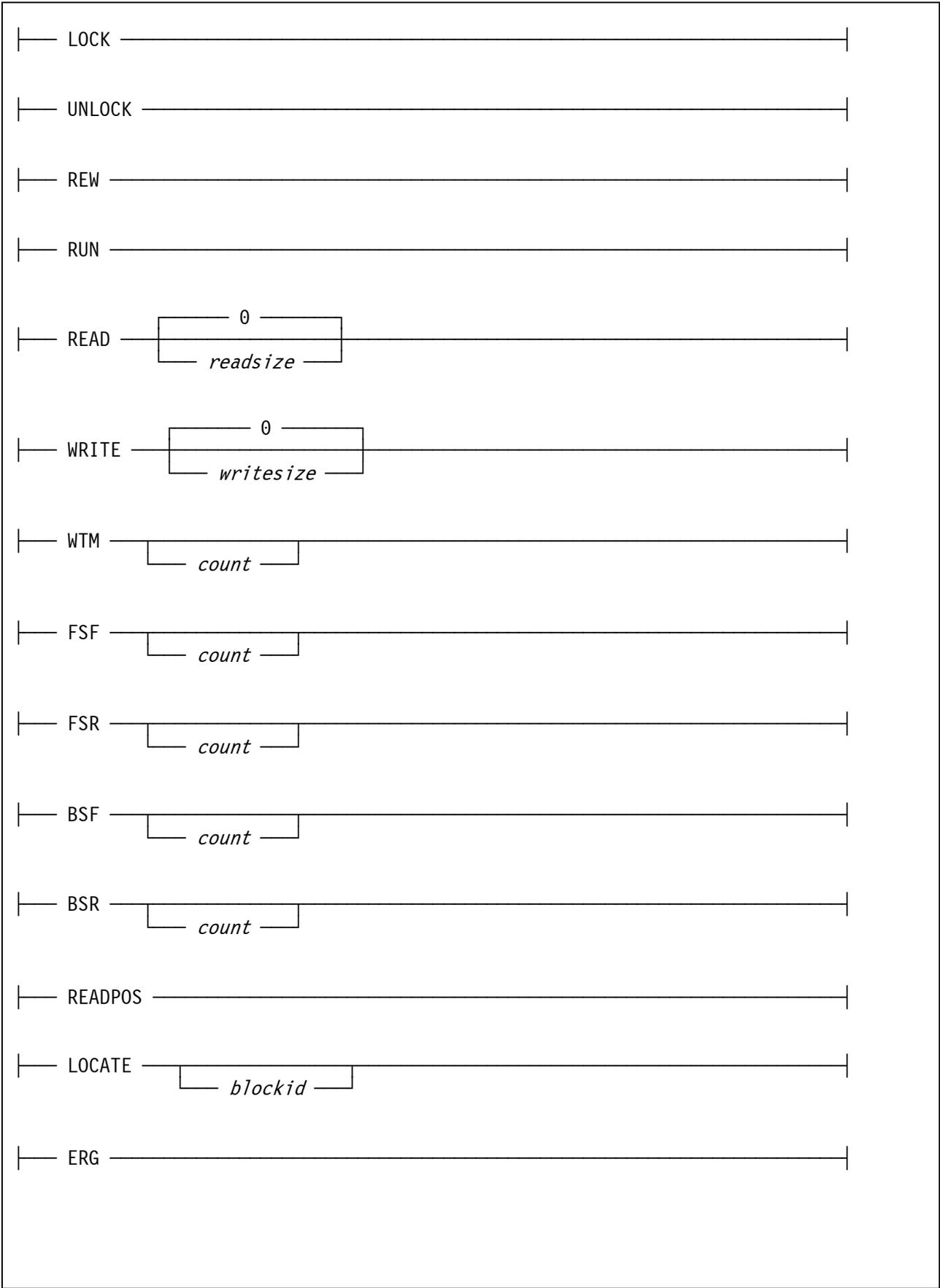
UNLOCK  
 REW  
 RUN  
 READ [*size* | 0]  
 WRITE [*size* | 0]  
 WTM [*count*]  
 FSF [*count*]  
 FSR [*count*]  
 BSF [*count*]  
 BSR [*count*]  
 READPOS  
 LOCATE [*blockid*]  
 ERG  
 COMP [*value* | 1]  
 ECC [*value* | 1]  
 MARGIN [*margin*]  
 STATUS  
 ASCII  
 EBCDIC  
 DUMP [*offset*[.*length*]]  
 ERASE  
 HELP | ?  
 EXIT | QUIT

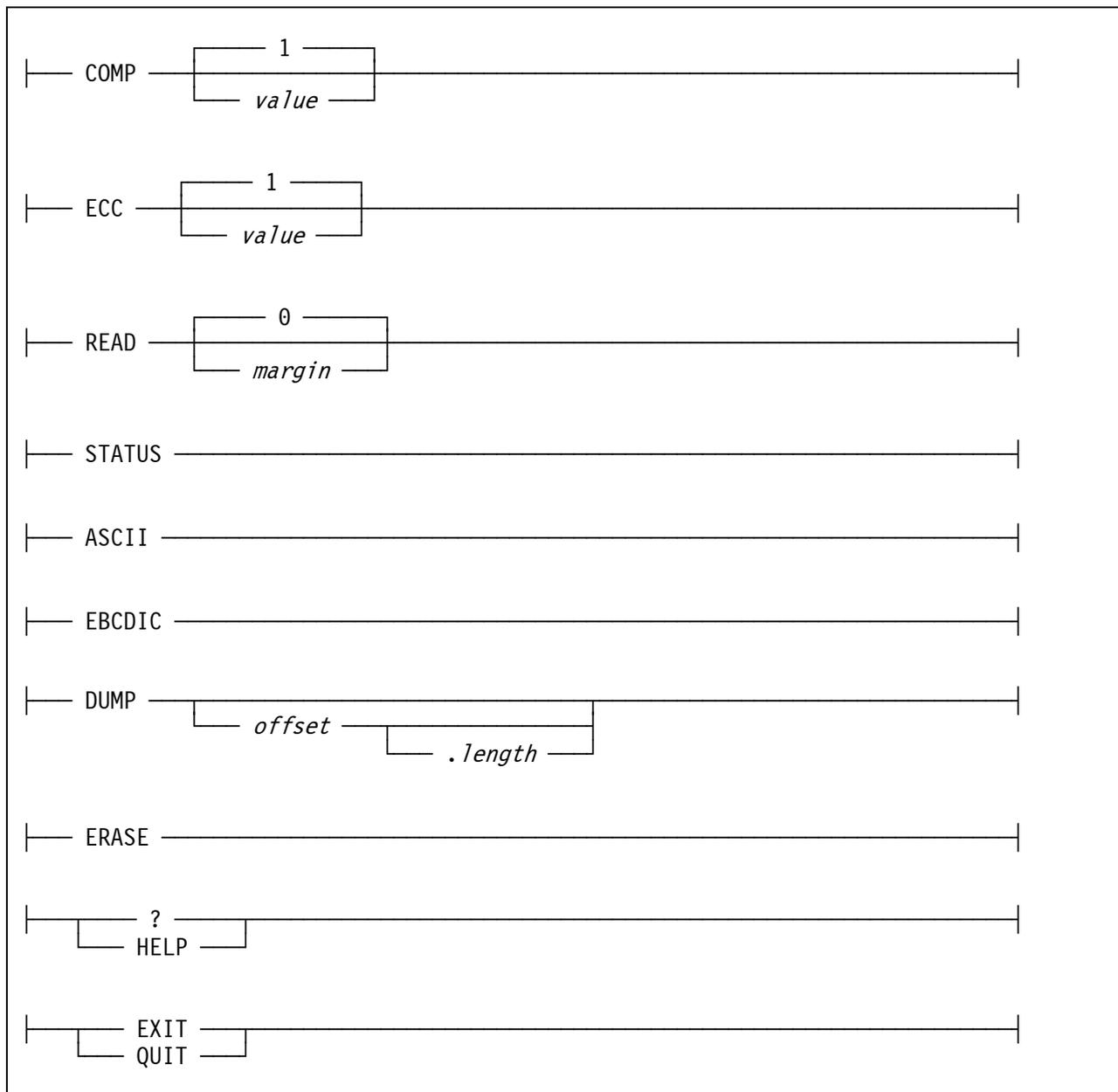
**Diagram**



where *function* can be:







### 6.1.3 Parameter

**filename** This specifies the name of the tape device. The default is "Tape0".

**function** The function that should be executed. For a list of available functions see below.

**functions:**

**LOAD** Load media.

**UNLOAD** Unload media.

<b>LOCK</b>	Lock media.
<b>UNLOCK</b>	Unlock media.
<b>REWIND</b>	Rewind tape.
<b>RUN</b>	Rewind and unload tape.
<b>READ</b>	Read ' <i>readsize</i> ' blocks.
<b>WRITE</b>	Write ' <i>writesize</i> ' blocks.
<b>WTM</b>	Write ' <i>count</i> ' tapemark(s).
<b>FSF</b>	Forward space ' <i>count</i> ' file(s).
<b>FSR</b>	Forward space ' <i>count</i> ' block(s).
<b>BSF</b>	Backward space ' <i>count</i> ' file(s).
<b>BSR</b>	Backward space ' <i>count</i> ' block(s).
<b>READPOS</b>	Read blockid.
<b>LOCATE</b>	Locate ' <i>blockid</i> '.
<b>ERG</b>	Erase gap.
<b>COMP</b>	Set compression on or off.
<b>ECC</b>	Set ECC on or off.
<b>MARGIN</b>	Set EOT margin ' <i>margin</i> '.
<b>STATUS</b>	Display status.
<b>ASCII</b>	Set ASCII data block display mode.
<b>EBCDIC</b>	Set EBCDIC data block display mode.
<b>DUMP</b>	Display data block at offset ' <i>offset</i> ' in length ' <i>length</i> '.
<b>ERASE</b>	Erase entire tape (data security erase).
<b>HELP or ?</b>	Display usage information.
<b>EXIT or QUIT</b>	Exit program.

***arguments:***

***readsize*** Specifies the number of blocks to read. *readsize* must be a value from 0 to 65535. A value of 0 (which is the default) means all blocks, whereas any other number

	specifies the actual count of blocks to read.
<b>writesize</b>	Specifies the number of blocks to write. <i>writesize</i> must be a value from 0 to 65535. A value of 0 (which is the default) means random, whereas any other number specifies the actual count of blocks to write.
<b>count</b>	Specifies the number of files, blocks or tape marks to process.
<b>blockid</b>	Specifies the ID of the block to be located.
<b>value</b>	Specifies the settings for compression and ECC. A value of '0' means OFF, whereas a value of '1' means ON. The default is '1' (ON).
<b>margin</b>	Specifies the EOT margin. <i>margin</i> must a value from 0 to 65535, the default is '0'.
<b>offset</b>	Specifies the offset from which data blocks are to be dumped.
<b>length</b>	Specifies the length of data blocks to be dumped.

## 6.1.4 Examples

### Example 1:

Display the status of SCSI tape device "Tape0".

```
FTAPE -f Tape0 STATUS
```

## 6.1.5 Sample Utility Output

### Sample 1:

Display the status of SCSI tape device "Tape0".

```
D:\Hercules>ftape

Fish's Win32 SCSI Tape Test Utility, v1.6.1.353
Copyright (C) 2004-2007, Software Development Laboratories
http://www.softdevlabs.com (fish@softdevlabs.com)

.
.
several lines not displayed
.
.

Command? status

Retrieving status...

Warning! Unable to request variable blocksize i/o
         due to media parameters retrieval failure!
```

Tape device: Tape0

Status: Not Ready (Not Loaded)

Drive information:

ECC	No
Compression	No
DataPadding	No
ReportSetmarks	No
DefaultBlockSize	32,768 bytes
MaximumBlockSize	65,536 bytes
MinimumBlockSize	1 bytes
MaximumPartitionCount	2 partitions
EOTWarningZoneSize	0 bytes
TAPE_DRIVE_ABS_BLK_IMMED	No
TAPE_DRIVE_ABSOLUTE_BLK	Yes
TAPE_DRIVE_END_OF_DATA	Yes
TAPE_DRIVE_FILEMARKS	Yes
TAPE_DRIVE_LOAD_UNLOAD	Yes
TAPE_DRIVE_LOAD_UNLD_IMMED	Yes
TAPE_DRIVE_LOCK_UNLOCK	Yes
TAPE_DRIVE_LOCK_UNLK_IMMED	No
TAPE_DRIVE_LOG_BLK_IMMED	No
TAPE_DRIVE_LOGICAL_BLK	Yes
TAPE_DRIVE_RELATIVE_BLK	Yes
TAPE_DRIVE_REVERSE_POSITION	Yes
TAPE_DRIVE_REWIND_IMMEDIATE	Yes
TAPE_DRIVE_SEQUENTIAL_FMKS	No
TAPE_DRIVE_SEQUENTIAL_SMKS	No
TAPE_DRIVE_SET_BLOCK_SIZE	Yes
TAPE_DRIVE_SET_COMPRESSION	No
TAPE_DRIVE_SET_ECC	No
TAPE_DRIVE_SET_PADDING	No
TAPE_DRIVE_SET_REPORT_SMKS	Yes
TAPE_DRIVE_SETMARKS	Yes
TAPE_DRIVE_SPACE_IMMEDIATE	No
TAPE_DRIVE_TENSION	No
TAPE_DRIVE_TENSION_IMMED	No
TAPE_DRIVE_WRITE_FILEMARKS	Yes
TAPE_DRIVE_WRITE_LONG_FMKS	No
TAPE_DRIVE_WRITE_MARK_IMMED	Yes
TAPE_DRIVE_WRITE_SETMARKS	Yes
TAPE_DRIVE_WRITE_SHORT_FMKS	No
TAPE_DRIVE_COMPRESSION	No
TAPE_DRIVE_CLEAN_REQUESTS	No
TAPE_DRIVE_ECC	No
TAPE_DRIVE_EJECT_MEDIA	Yes
TAPE_DRIVE_ERASE_BOP_ONLY	No
TAPE_DRIVE_ERASE_LONG	No
TAPE_DRIVE_ERASE_IMMEDIATE	Yes

TAPE_DRIVE_ERASE_SHORT	Yes
TAPE_DRIVE_FIXED	No
TAPE_DRIVE_FIXED_BLOCK	Yes
TAPE_DRIVE_INITIATOR	Yes
TAPE_DRIVE_PADDING	No
TAPE_DRIVE_GET_ABSOLUTE_BLK	Yes
TAPE_DRIVE_GET_LOGICAL_BLK	Yes
TAPE_DRIVE_REPORT_SMKS	Yes
TAPE_DRIVE_SELECT	No
TAPE_DRIVE_SET_EOT_WZ_SIZE	No
TAPE_DRIVE_SET_CMP_BOP_ONLY	No
TAPE_DRIVE_TAPE_CAPACITY	No
TAPE_DRIVE_TAPE_REMAINING	No
TAPE_DRIVE_VARIABLE_BLOCK	Yes
TAPE_DRIVE_WRITE_PROTECT	Yes
.	
.	
.	

**Figure 56: FTAPE utility output (display status)**

---

## 6.2 HercPrt (Remote Hercules Printer Spooler)

### 6.2.1 Function

HercPrt is a remote Hercules printer spooler written by Fish (David B. Trout), the author of the widely used HercGUI, CTCI-WIN, AWSBrowse and FTAPE packages. HercPrt is designed to receive text output from a Hercules “socket device” (sockdev) line printer and use it to create a disk file on the local Windows system.

Because HercPrt uses standard TCP/IP sockets to communicate with a Hercules line printer, the actual Hercules system whose line printer output HercPrt is spooling, can be physically located at any place reachable via standard IP networking.

HercPrt supports creation of either plain text files, HTML files, Rich Text Format files (RTF) or Portable Document Files (PDF) according to the options provided in a “Job Separator Control File”. Several sample job separator control files are provided with the HercPrt package.

The job separator control file tells HercPrt what the Hercules guest operating systems job separator pages look like, so that it can automatically create separate Windows files for each print job, each optionally named according to a choice of job accounting field values extracted directly from the job separator page itself. HercPrt will create these spooled print files in whatever directory you choose using your specified options.

Several HercPrt instances can be run simultaneously, each spooling a different Hercules line printer for either the same instance or a completely different instance of Hercules. The options used for each printer are saved under a user specific printer ID so that the same options can easily be specified the next time HercPrt is run, by simply selecting the desired printer ID from a provided list of previously defined printer IDs.

The main dialog user interface is fully resizable and can be completely hidden (minimized) to the system tray area for minimal interference with normal Windows use. Automatic connection and reconnection support (with a user configurable delay between retries) is provided as well as complete control over the display of popup balloon tooltips used to notify the user of either incoming print output or loss of connectivity.

### 6.2.2 Installation

The installation of HercPrt is beyond the scope of this manual. For detailed instructions on installing HercPrt on a Windows system please see the Hercules “Installation Guide”.

### 6.2.3 Hercules Configuration

HercPrt is designed to receive text output from a Hercules “socket device” (sockdev) line printer. In order to use HercPrt with Hercules a “sockdev” printer must first be added to the Hercules configuration.

A Hercules socket printer definition looks similar to the following device definition statement:

```
000E 1403 192.168.0.100:14031 sockdev
```

The value ‘192.168.0.100:14031’ is the TCP/IP address and the port number at which the Hercules socket printer will listen for incoming connections. The IP address is typically the IP address of the Windows system where HercPrt is running on and the port number can be any value from 1024 to 65535.

Note that Hercules socket printers do not support any other options besides “sockdev”, the “clrf” and “no-clear” options for example are invalid when defining a socket printer. For details on how to define printers in a Hercules configuration file please see the Hercules “User Reference” manual.

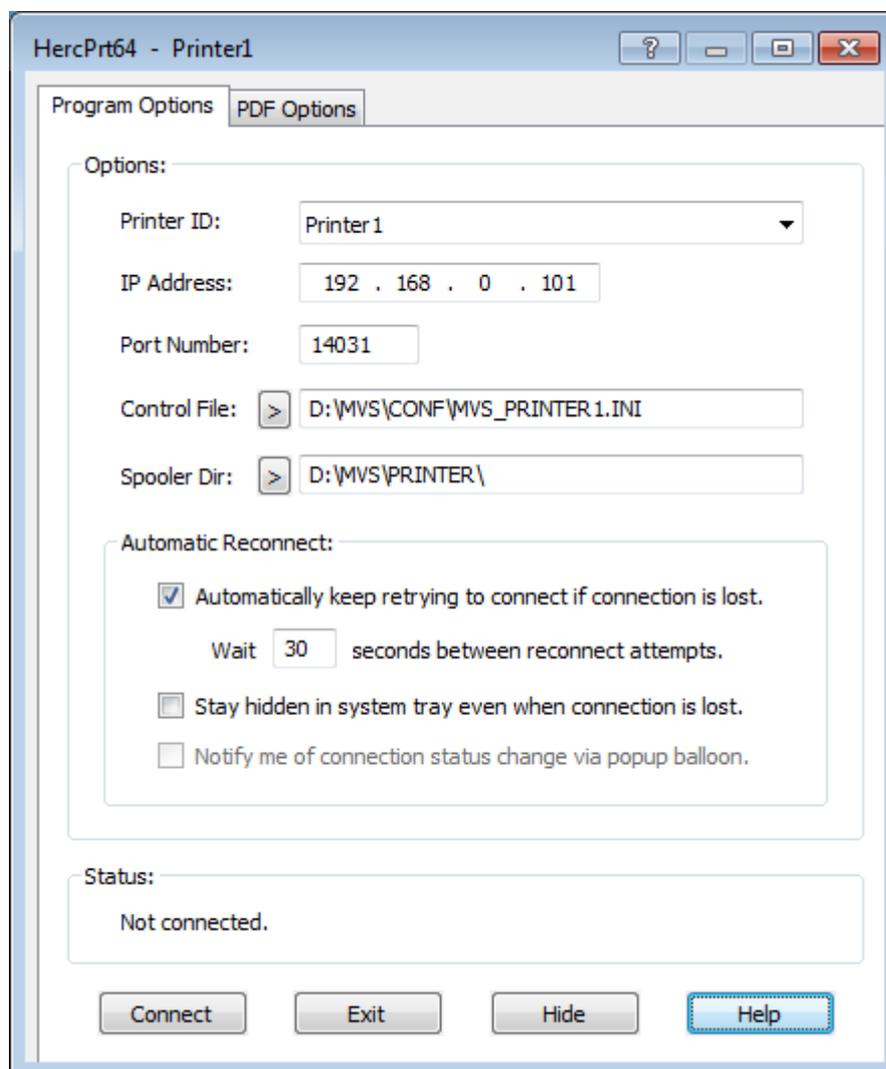
When Hercules is powered on the socket printer begins listening for incoming connections on the IP address and port number that are assigned to it. When you start HercPrt you define a printer at the same IP address and the same port number and click the connect button.

HercPrt should then establish a connection to that Hercules printer and immediately begins spooling reports from this printer if any are available.

Whether any reports are available or not on a given printer (as well as how to print them) depends on the guest operating system and associated spooler running under Hercules and is beyond the scope of this manual.

## 6.2.4 Program Options Page

The following figure shows the first of two available configuration pages from the HercPrt application – the “Program Options” page.



**Figure 57: HercPrt Program Options Page**

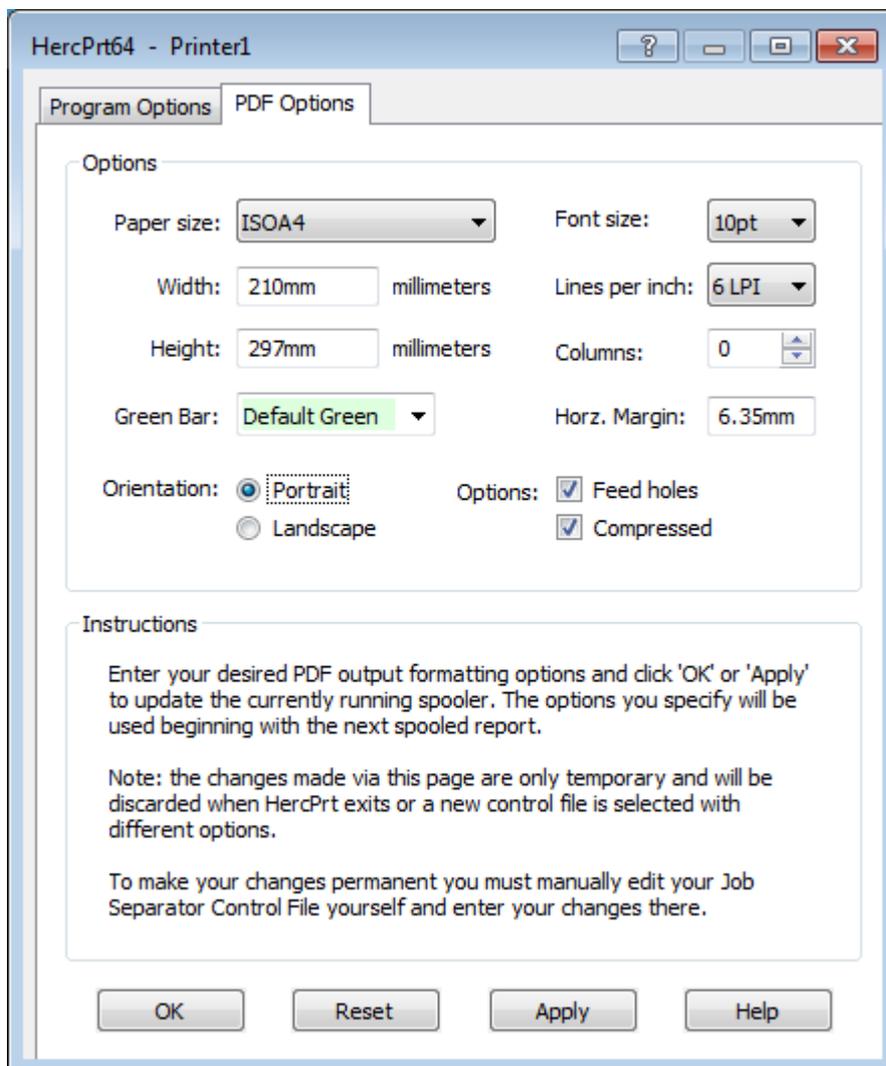
On the “Program Options” page you can configure the following:

<b>Printer ID</b>	This field is used to enter a descriptive name for the Hercules printer you are defining. The value can be anything that uniquely identifies the printer being defined. If you already have some printers defined you can select one of these previously defined printers from the dropdown list to automatically populate the remaining controls with the values for the chosen printer.
<b>IP Address</b>	This must be the IP address where your Hercules printer is listening for incoming connections.
<b>Port Number</b>	This must be the port number where your Hercules printer is listening for incoming connections.
<b>Control File</b>	<p>This is the filename of the “Job Separator Control File” for the printer. The job separator control file defines what your Hercules guest operating system’s job separator pages look like and allows HercPrt to detect where one print job ends and the next print job begins.</p> <p>This allows HercPrt to break spooled output into separate Windows files, one for each printout, and to name the files using information from the job accounting fields extracted from the print fields on the actual job separator page itself.</p> <p>Click on the arrow to display a standard Windows file open dialog where you can select the job separator control file you want to use for this printer.</p>
<b>Spooler Dir</b>	<p>Enter here the name of the directory where you want the print files to be placed.</p> <p>Click on the arrow to display a standard Windows file open dialog where you can select the directory for the print files of this printer.</p>
<b>Automatic Reconnect</b>	Check this box to automatically retry a failed connect attempt and to automatically keep retrying to reconnect whenever the connection is unexpectedly lost.
<b>Wait Seconds</b>	This is the number of seconds to pause between the time a (re-)connect fails and the time the (re-)connect attempt is tried again. The idea here is not to bombard the Hercules system with a steady stream of (re-)connect attempts and to wait a short while before trying again to connect.
<b>Stay Hidden</b>	Check this box to automatically hide the dialog once a connection has been made and to remain hidden even during reconnect retries whenever the connection is unexpectedly lost.
<b>Notify Me</b>	Check this box to have the system tray icon automatically display a popup balloon whenever the connection status changes (e.g. connecting, receiving, idle, etc.).
<b>Status</b>	This is the current status of the remote printer connection.
<b>Connect / Disconnect</b>	<p>Use this button to either connect to, stop retrying to connect to or disconnect from the specified Hercules printer.</p> <p>The connect button will be labelled “Connect” if HercPrt is not already connected to a Hercules printer or else will be labelled “Disconnect” if HercPrt is connected to a printer or in the process to connect.</p> <p>Clicking the button when it is labelled “Disconnect” will immediately disconnect from the specified printer if there is already a connection or else will immediately stop retrying to connect if HercPrt is currently in the process of connecting and “Automatic Reconnect” is enabled.</p>

- Exit**                    Clicking this button immediately disconnects from the Hercules printer and exits the program.
  
- Hide**                    Clicking this button hides the dialog. To redisplay it again double click on the system tray icon or right click and select “Show” from the appearing context menu. Single clicking the tray icon display a popup balloon showing the current status.
  
- Help**                    Displays the programs help information.

### 6.2.5 PDF Options Page

The next figure shows the second of two available configuration pages from the HercPrt application – the “PDF Options” page.



**Figure 58: HercPrt PDF Options Page**

On this page the desired PDF or RTF output formatting options can be specified to update the currently running spooler. The options specified here will be used beginning with the next spooled output.

The changes made on this page are only temporary and will be discarded when HercPrt exits or a new control file with different options is opened. To make this changes permanent you must manually edit the job separator control file and enter the changes there.

Please note that the options “Feedholes”, “Compression” and “Columns”, as well as all those related to the “Green Bar” are not supported for RTF but only to PDF files.

- Paper Size** This drop down allows to choose one of several predefined paper sizes. The pre-defined paper sizes are US Fan Fold, US Compact Fan Fold, US Letter, ISO A3, ISO A4, ISO B4, ISO B4 Extra and ISO C3.
- Paper Width** Enter here the desired custom form width in either fractional inches (the default) or whole millimetres. To specify millimetres append ‘mm’ to the size (e.g. ‘210mm’).
- Paper Height** Enter here the desired custom form height in either fractional inches (the default) or whole millimetres. To specify millimetres append ‘mm’ to the size (e.g. ‘297mm’).
- Green Bar** This dropdown is to select the desired “green bar” colour or a custom colour using the #RRGGBB format. Chose ‘No’ for no green bar or ‘Outline’ for an uncoloured “transparent” green bar.  
Note: The green bar option is ignored for RTF output.
- Orientation** Choose the desired paper orientation, either portrait or landscape.
- Font Size** Use this dropdown to choose the desired font size from the list of available sizes. Available font sizes are 9pt, 10pt, 11pt and 12pt. Any size may be used for reports printed at 6 LPI, but the 9pt font size is forced for reports printed at 8 LPI.  
The font itself is always “Courier New” and cannot be changed.
- Lines per Inch** The default line spacing is 6 lines per inch. If you choose to print your report at 8 LPI then the 9pt font size is forced.
- Columns** To squeeze long print lines onto narrow forms or stretch short print lines to fill wider forms enter here the number of columns you want to be able to fit onto your custom sized form.  
The recommended default value of zero uses the normal default pitch of 10 characters (columns) per inch.  
Note: The columns option is ignored for RTF output.
- Horizontal Margin** Specifies how far from the left edge of the form (not including the feed holes) you wish each print line to start.  
When the output is PDF the same value is also used as the right margin and print lines are prevented from printing beond it.  
When the output is RTF however, the value is only used as a left margin; the right margin is always set to zero and any print lines which would otherwise extend beyond the far right edge of the form are instead automatically wrapped onto the next line.
- Feed Holes** Select this option to cause a realistic looking set of tractor-feed holes to be drawn along the left and right sides of the report.  
Note: The feed holes option is ignored for RTF output.
- Compressed** Produces standard compressed PDF files instead of legacy uncompressed ones,

thereby reducing the additional disk space the reports would otherwise consume.

Note: The compressed option is ignored for RTF output.

- OK** Clicking the “OK” button validates and saves the settings and then automatically hides the dialog. It is the same as clicking the “Apply” button followed by the (x) “Close” button.
- Reset** Clicking the “Reset” button causes the changes made on this panel to be discarded, resetting all values back to those defined in the [PDF] section of the job separator control file.
- Help** Displays the programs help information.

### **6.2.6 Help System**

Clicking the [?] context sensitive help button turns the mouse cursor into a question mark shaped cursor indicating the system is now in context sensitive help mode.

Once in context sensitive help mode you can click with the mouse on any of the dialogs controls to cause a popup help balloon tooltip to be displayed containing a brief explanation of the control you clicked on.

To display the full help information for the control instead of just a popup balloon tooltip, then instead of clicking the [?] button first, simply press the F1 key while the control in question has the focus.

### **6.2.7 Tray Icon Context Menu**

The tray icon context menu appears whenever you right click on the HercPrt system tray icon. It has the following options:

- Show** Display the main dialog window. This is also the default action whenever the system tray icon is double clicked.
- Connect** Selecting “Connect” performs the same action as the “Connect” button on the main dialog. It connects to the defined Hercules printer at the specified IP address and port number.
- Disconnect** Selecting “Disconnect” performs the same action as the “Disconnect” button on the main dialog. It either disconnects from the Hercules printer or stops trying to reconnect, depending on whether “Automatic Reconnect Retry” is enabled or not.
- Help** Opens the applications help file.
- About** Displays the applications “About” dialog, which displays the program version number and copyright information as well as the Software Developments Laboratories support email address used when submitting problem reports.
- Exit** Selecting “Exit” performs the same action as the “Exit” button on the main dialog. It immediately disconnects from the Hercules printer and exits the program.

## 6.2.8 Job Separator Control File

The "Job Separator Control File" describes the layout of the guest's job separator page as well as other spooler options. HercPrt uses this control file to be able to decide when the end of the spooled report has been reached as well as how to name the resulting file.

Additionally the control file also contains various options related to how you want your spooled output to look like for those file types whose appearance can be customized (currently limited to Portable Document Files (PDF) and to a limited degree to Rich Text Format (RTF) files).

HercPrt provides sample control files already configured for three of the most popular Hercules guest operating systems:

Control File Name	Operating System
DOSVSR23.INI	DOS/VS Release 34
VM370R6.INI	VM/370 Release 6
MVSJES2.INI	MVS V3.8J

**Table 6: Sample Control Files**

The provided samples can either be used as is or can be customized to suit your needs.

### 6.2.8.1 [Separator] section

The [Separator] section contains the information which defines what your guest's end-of-job or beginning-of-job separator page looks like. Each "trigger" defines one line of the separator page, and *all* of them must match together as a group in order for the page to be considered a job separator page.

For most systems these lines would be those which appear on your guest's end-of-job separator page, not the beginning-of-job separator page, since it is the *end*-of-job separator page which defines where the current print job ends and thus where the next one begins.

Some operating systems however use a spooler which only outputs beginning-of-job separator pages, making it impossible to know with any degree of certainty when the current print job has ended (except in the special case where another print job immediately follows it; the next print job's beginning-of-job separator would thus also serve to define the end of the previous job). Nevertheless a special format is supported to try and accommodate these unusual spoolers.

#### 6.2.8.1.1 Pages=

The first 'n' value of the "Pages=(n,m)" string specifies the number of consecutively identical end-of-job separator pages there are and defines the number of consecutive times in a row that all triggers must match.

The 'm' value defines the number of additional end-of-job separator pages there are following the identical pages. This parameter is needed to accommodate spoolers such as DOS/VS's POWER/VS spooler those separator pages are not all the same.

For MVS's JES2 spooler (whose separator pages are all the same) you should specify "(2,0)", whereas for DOS/VS's POWER/VS spooler (whose separator pages are not) the value "(2,1)" should be used instead. Both values of course presume the usual two page default. If your guest uses only one separator page then you should of course use (1,0) or (1,1) instead.

To accommodate non-standard spoolers such as VM/370 R6 which only output beginning-of-job separator pages but not any end-of-job separator pages (instead it just stops printing!), the "(n,TIMEOUT=x)" format is used instead.

The first number 'n' defines how many beginning-of-job separator pages to look for, and the 'x' in "TIMEOUT=x" defines how many seconds after which no more additional spooled output is received should indicate that the end of the print job has been reached.

Using a timeout value like this to indicate when the end of the print job has been reached is a somewhat unreliable way to detect the end of a print job, but without an end-of-job separator page it's the best that can be done for now.

#### **6.2.8.1.2 Triggers=**

The "Triggers=" value defines how many trigger lines there are on your separator page. A trigger line is a line that exists on a job separator page that, together with other lines (triggers), identifies the page as a job separator page. The "Triggers=" value defines how many of these lines there are, and thus how many "[Trigger#]" sections there are following it.

#### **6.2.8.1.3 [Trigger#]**

Each "[Trigger#]" section defines one separator page line. The "#" character in each section's name should of course be replaced with the associated trigger number. That is, if there are three lines on your job separator page then you would specify "Triggers=3" and define three "[Trigger#]" sections called "[Trigger1]", "[Trigger2]" and "[Trigger3]".

#### **6.2.8.1.4 Line=, Column=, Value=**

The "Line=" value identifies which line of the separator page contains the text defined by the "Value=" value, and the "Column=" defines at which column of that line the text appears at. All line and column number values are 1-relative. Thus the first line on the page would be line 1 and the letter 'w' in "Hello world!" would be at column 7.

The text defined by "Value=" should be the exact text that appears at that line and column. All value comparisons are case sensitive and must match exactly. Enclose the value within double quotations if it contains any blanks.

There is no minimum or maximum value for the number of triggers you may define, but you should define enough of them to uniquely identify the page as a job separator page to prevent normal report pages from being erroneously detected as job separator pages.

### **6.2.8.2 [Field Names] section**

The [FieldNames] section defines where the job accounting fields appear on your job separator page, thus allowing you to use them in your spooled report's output filename via the [OutputFile] section.

The individual job accounting field values are extracted directly from the separator page's individual print lines based on the values defined in the [FieldNames] section.

#### **6.2.8.2.1 Names=, Name#=-**

The "Names=" value defines how many job accounting fields are being defined and how many "Name#=" entries immediately follow it. The "#" character in each "Name#=" entry should of course be replaced with the associated field number. For example, if you wish to define three job accounting fields you would specify "Names=3" followed by "Name1=*one*", "Name2=*two*" and "Name3=*three*", where "*one*", "*two*" and "*three*" are of course the specific job accounting field's unique name.

You can assign whatever name you wish to your defined job accounting fields as long as the name does not contain any blanks and does not match any of the following reserved section names: "FieldNames", "OutputFile", "PDF", "Separator", "Translations" and "Trigger#".

#### 6.2.8.2 [fieldname] Line=, Column=, Width=

Each "[fieldname]" sub-section following the primary [FieldNames] section should match one of the field names defined in one of the "Name#=" entries. Note that it is not considered an error if it does not match any of your defined "Name#=" entries. Rather, it will simply never be used. The reverse however is not true. For each field name defined by a "Name#=" entry, a corresponding "[fieldname]" section must exist or it is considered an error.

The "Line=", "Column=" and "Width=" values define where on the page that particular field is located. Just like the "Line=" and "Column=" values of the [Separator] section, the line and column number values are 1-relative. A "Width=" value of 0 is invalid, but a width value of -1 means the field extends through to the end of the print line.

Whenever a job separator page is detected (as defined by the [Separator] section) all job accounting field values are extracted and saved according to the information contained in your [fieldname] sections. All defined job accounting field names are case sensitive and all extracted field values have their leading and trailing blanks removed before being saved.

#### 6.2.8.3 [Output File] section

The [OutputFile] section contains a single "Name=" entry which defines how the job accounting fields must be used to construct the spooled report's filename. Job accounting fields are defined in the [FieldNames] section and their values are extracted directly from the job separator page and saved for use by the [OutputFile] section's "Name=" entry. Each percent sign enclosed string within the "Name=" value is substituted with one of the defined job accounting field values.

If, for example, have the following job accounting field defined in your [FieldNames] section and the contents of the job separator page's 58th line happens to be a "Name=" setting of "Job %JOBNAME%.txt" then this would result in the spooled report file being named "Job FOOBAR.txt".

Please note that Windows filenames cannot have any characters in them whose integer representation is within the range of 0 through 31 (i.e. x'00' to x'1F') nor can they contain any of the following reserved characters:

```
< > : " / \ | * ?
```

Further note that the length of a file's complete path specification cannot exceed the standard Windows operating system path length restriction of 260 characters total, so try to limit your output filename to the minimum needed while still ensuring that the resulting names remain different from one another.

After constructing an initial output filename based on your defined "Name=" value (which should be considered only a template), HercPrt then applies further string and character translations as defined in the [Translations] section before attempting to rename the file to its final name.

If during the construction of the output filename it is detected that a file already exists with that name, then "-2" is appended to the generated name and that is used instead. If a file with that name already exists then "-3" is appended, and so on, all the way to "-10". If a unique filename still cannot be constructed after trying "-10", then a warning is issued and the file is left with its original Windows assigned temporary filename (e.g. "PRT305.tmp").

The filename extension of your "Name=" value is used to determine what format the spooled file should be created in. If the "Name=" template ends with ".txt" the output file will be in plain text format (the default). If it ends with ".html" then it will be in HTML format, etc. Other supported filename extensions are ".rtf" and ".pdf". If the filename does not end with one of the supported filename extensions (txt, html, htm, rtf or pdf) then the file is created in text format by default.

If the output filename extension is either .rtf or .pdf then additional file formatting options may be specified via the [PDF] section.

#### 6.2.8.4 [Translations] section

The [Translations] section allows you to further customize your output filename by allowing you to define character and string substitutions to take place during output filename construction.

#### 6.2.8.4.1 Strings=, Strings#-=, Strings#+=

Each "String#-=" value defines an output filename substring that is to be replaced with the string defined in the corresponding "String#+=" entry (where '#' represents a digit in the range of 1 to n with 'n' being the value defined by the preceding "Strings=" entry).

The "Strings=" value defines how many corresponding "String#-=" or "String#+=" entries there are. The replacement strings defined by the "String#+=" entries do not have to be the same length as the "String#-=" entry, and may even be empty to cause the substring defined in the "String#-=" entry to be completely removed.

#### 6.2.8.4.2 Chars-=, Chars+=

The "Chars+" value defines the replacement characters which should be used to replace the corresponding characters defined at the same relative position in the "Chars-" value. Each character of the filename generated by the "Names=" entry of the [OutputFile] section matching one of the characters defined in the "Chars-=" value is replaced with the corresponding character defined at the same relative position in the "Chars+==" value. There can only be one "Chars-=" and "Chars+==" entry but their string values (which must be the same length) can be as long as you wish.

In this way the "String..." and "Chars..." entries together allow you to define additional output filename transformations to take place during the construction of the final output filename. String replacement is always performed first followed by character replacement second, and a final internal character translation of:

< > : " / \ | \* ?

to:

{ } . ' - \_ ! @ !

is always performed last to ensure that no invalid or illegal characters remain in the final output Windows filename.

#### 6.2.8.5 [PDF] section

The [PDF] section contains options used to define how you wish your output .pdf or .rtf file to be formatted. Please note that while all of the below options are supported for PDF output, not all of them are supported for RTF output.

The options which are not supported for .rtf files are: all of the GreenBar options (e.g. color) as well as the Feedholes, Compressed and Columns options. All of other PDF options (e.g. paper and font size, etc.) are supported for both PDF and RTF files:

Option	PDF	RTF
PaperSize= <i>formname</i>	Yes	Yes
PaperWidth= <i>width</i>	Yes	Yes
PaperHeight= <i>height</i>	Yes	Yes
Orientation={Portrait   Landscape}	Yes	Yes
HorzMargin= <i>margin</i>	Yes	Yes
FontSize={9   10   11   <u>12</u> }	Yes	Yes
LPI={6   <u>8</u> }	Yes	Yes

Columns={ <u>0</u>   cols}	Yes	No
GreenBar={No   Yes}	Yes	No
BarColor={colour   #rrggbb}	Yes	No
FeedHoles={No   Yes}	Yes	No
Compress={Yes   No}	Yes	No

**Table 7: HercPrt PDF and RTF Options**

### 6.2.8.5.1 PaperSize

Predefined "PaperSize=" values are "USFanFold" (default), "USCompactFanFold" (LPI=8), "USLetter", "ISOA3", "ISOA4", "ISOB4", "ISOB4Extra" and "ISOC3". When a predefined "PaperSize=" value is used then a "PaperWidth=" and "PaperHeight=" value need not be entered. Otherwise define your custom form size via "PaperWidth=" and "PaperHeight=".

### 6.2.8.5.2 PaperWidth / PaperHeight

"PaperWidth=" and "PaperHeight=" values are expressed in either fractional inches (the default) or whole millimeters. To express a value in millimeters append "mm" to your value (e.g. "297mm").

Note: when defining your form's width do not include the width of the feedholes. The width of the feedholes (0.5 inches on either side of your form) is automatically added to your specified paper width at run-time whenever the "FeedHoles=Yes" option is specified. Your "PaperWidth=" value should always be specified as the actual printable width of your form and should not include the width of the feedholes area.

### 6.2.8.5.3 Orientation

Specifying "Orientation=Landscape" or "Orientation=Portrait" will swap your specified "PaperWidth=" and "PaperHeight=" values if necessary in order to match your defined orientation.

Therefore defining "PaperWidth=8.5" and "PaperHeight=11" (or "PaperSize=USLetter") along with "Orientation=Landscape" will cause a paper width of 11 inches and a paper height of 8.5 inches to be used instead.

### 6.2.8.5.4 HorzMargin

The "HorzMargin=" value specifies how far from the left edge of the form the first character of the print line is. For PDF output the "HorzMargin=" value defines both the left and right margins (the right margin being the end of the print line, i.e. the position past which no character will print). For RTF output however, it defines only the left margin and the right margin is always set to zero. Additionally, for RTF output, print lines that extend beyond the right margin automatically wrap to the next print line whereas for PDF output they are simply truncated.

The vertical margin at the top of every page is fixed at 0.5 inches and cannot be changed. The first printable line on a page after performing a page eject is therefore always the 4th physical line on the page when printing at 6 lines per inch or the 5th physical line if printing at 8 lines per inch (even though in both cases it is still considered to be line number 1).

The only way to print within the top 0.5 inches of your form is to continue printing beyond the bottom of the previous page such that your print then "overflows" onto the first physical line of the next page. Be aware however that you cannot do this continuously. That is to say your page cannot be of unlimited length. When any page grows to be 500 lines long or longer a page-eject is automatically forced and this behavior cannot be overridden.

### 6.2.8.5.5 FontSize and LPI

"FontSize=" is always specified in typographic "points" (1/72 inches) with the default being 12pt for LPI=6 (Lines Per Inch) and 9pt for LPI=8. When 8 LPI is chosen then 9pt is forced. Valid font sizes are 9pt, 10pt, 11pt or 12pt.

The font that is used is always "Courier New" and cannot be changed.

### 6.2.8.5.6 Columns

The "Columns=" value should normally be specified as 0 (zero) to ensure a proper default character pitch of 10 characters per inch. Specifying a nonzero "Columns=" value causes all characters on your print line to either be squished or stretched in an effort to try and honor your specified value, usually resulting in a very ugly looking printout.

### 6.2.8.5.7 GreenBar, FeedHoles and BarColor

The default for "GreenBar=" and "FeedHoles=" is "No". "BarColor=" defaults to "Green". Other valid "BarColor=" values are: "Blue", "Yellow", "Red", "No" or "#rrggb".

Specifying "BarColor=No" will draw an outlined box around the area where the green bars would normally be, but without being filled in with any background color (i.e. the color of the bar is set to "transparent"). Custom green bar colors can be specified using the "#RRGGBB" html RGB color value format.

### 6.2.8.5.8 Compression

The "Compress=Yes" option (the default) performs normal ZLIB compression of the actual PDF page data stream in order to conserve Windows host disk space. "Compress=No" should only be used when reporting a problem viewing a HercPrt generated PDF file since such problems are usually the result of an improperly formatted PDF data stream and using ZLIB compression makes it difficult to see the actual PDF codes.

## 6.2.9 txt2pdf Utility

HercPrt also ships with a command-line tool called "txt2pdf" which will convert a text file into either a PDF or RTF file.

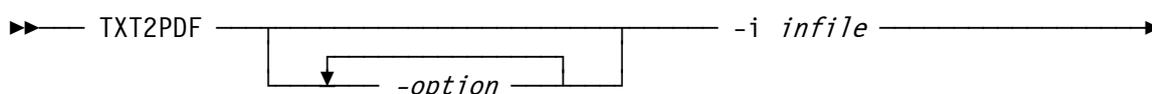
All of the PDF options that can be specified to HercPrt can also be specified to txt2pdf. Note however that just like with HercPrt, when converting text files to RTF files then not all of the PDF options are supported.

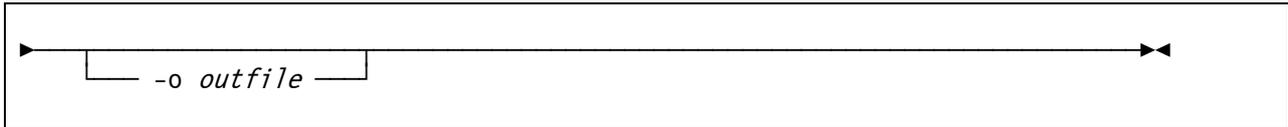
### 6.2.9.1 Syntax

#### Descriptive

```
TXT2PDF [-option [-option ... ]] -i infile [-o outfile]
```

#### Diagram





### 6.2.9.2 Parameter

#### options:

- a** Indicates that each input file line starts with an ASA/ANSI carriage control character (e.g. 1, 0, -, +, or <blank>).
- l *lpi*** Lines per inch: 6 or 8. (The default is 6 LPI).
- f *fontsize*** Font size: 9, 10 11 or 12 pt. (The default is 12 pt, if 6 LPI; otherwise 9pt).
- w *width*** Paper width in inches or millimeters. (Default is 13.875 inches).
- h *height*** Paper height in inches or millimeters. (Default is 11.0 inches).
- m *margin*** Left / right margin in inches or millimeters. (Default is 0.25 inches).
- t "*title*"** Document title property.
- s "*subject*"** Document subject property.
- r "*author*"** Document author property.
- i *infile*** The name (and optionally path) of the input ".txt" file to be converted.
- o *outfile*** The name (and optionally path) of the output ".pdf" or ".rtf" file (see notes).

#### PDF options:

- g** Green bar paper.
- c *color*** Green Bar color: 'Green' (default), 'Blue', 'Red', 'Yellow', or custom color specified as '#RRGGBB' HTML color value.
- n** No color (i.e. outlined "transparent" green bar).
- e** Feedholes.
- u** Uncompressed PDF output. (Default is compressed).
- # *cols*** Columns: 0-255. (Default is 0; see notes).

#### Notes:

- All PDF options are ignored when the output file is ".rtf" and not ".pdf".
- The -i switch is required in order to support specifying all options in any order. Options may precede the name of the input file or follow it or both, but the -i switch is required to identify the name of the actual input file.

- If -o is not specified the output file is named the same name as the input file but with a .PDF file extension. If you wish convert your file to .RTF instead of .PDF, then you MUST use the -o option to specify an output name ending with the ".RTF" file extension.
- The default paper size is US Standard Fanfold (13.875 x 11). Custom sizes should be specified in fractional inches or whole millimeters. To specify millimeters append 'mm' to your specified size (e.g. '287mm').
- The top margin size is fixed at 0.5 inches and cannot be changed.
- Do not include the width of the feedholes when specifying your paper width. The width of the feedholes (0.5 inches) is automatically added at runtime.
- The -n option is mutually exclusive with the -c option and overrides it when specified. Both are ignored if the -g option is not also specified.
- If the -# (columns) option is specified then each line is either squished or stretched in an attempt to fit exactly that number of columns of print onto each print line, usually resulting in a very ugly looking report. For best results it is recommended you not specify the -# option, or specify it as '0' (zero) for the default horizontal pitch of 10 characters per inch.

### 6.2.9.3 ASA Characters and their ASCII Equivalent

Unless the -a "ASA" option is specified, each line of the text input file that ends with a CR (carriage return) will cause that line to be overlaid with the text from the next one. Each line which ends with a LF (line feed or "new line" (NL) character) causes a skip to the next line after that line is printed. Skips to new pages only occur whenever a FF (form feed) character is encountered.

If the -a "ASA" option is specified, then each line of the input file is expected to start with an ASA (ANSI) carriage control character which identifies the action to be taken before that line is printed. Lines starting with a '1' cause a skip to the top of a new page before printing the line.

Lines starting with a blank cause a skip to a new line to occur before that line is printed. A '+' causes no spacing to occur thereby causing the line to overlay the previous one. '0' skips two lines before printing and '-' skips three lines before printing:

ASA Character	Action to take before printing	ASCII Equivalent
1	Advance to next page (Form Feed).	FF
+	Do not advance any lines before printing (overstrike the previous line).	CR
(blank)	Advance one line (single spacing).	CR LF
0	Advance two lines (double spacing).	CR LF CR LF
-	Advance three lines (triple spacing).	CR LF CR LF CR LF

**Table 8: HercPrt ASA Control Characters and ASCII Equivalent**

### 6.2.10 Samples

The first sample shows a "Job Separator Control File" for MVS 3.8J. It is based on the sample that is provided with the HercPrt Software.

Please note, that the sections "Separator" and "FieldNames" have been shortened to keep the sample within reasonable limits in this manual. Both sections show only the first and the last entry of the nume-

rous "Trigger" and "FieldNames" entries. Complete, working job separator control files are delivered with the HercPrt software for MVS 3.8J, for DOS/VS Release 34, and for VM/370 Release 6.

```
#-----  
#                               Separator  
#-----  
[Separator]  
Pages="(1,0)"  
Triggers=25  
  
[Trigger1]  
Line=31  
Column=1  
Value="**END*****END*****END*****END*****END*****END*****END*****END*****"  
.  
.  
.  
[Trigger25]  
Line=55  
Column=1  
Value="**END*****END*****END*****END*****END*****END*****END*****END*****"  
  
#-----  
#                               FieldNames  
#-----  
[FieldNames]  
Names=16  
Name1="JOBID"  
.  
.  
.  
Name16="SYSTEM"  
  
[JOBID]  
Line=33  
Column=18  
Width=62  
.  
.  
.  
[SYSTEM]  
Line=53  
Column=18  
Width=62  
  
#-----  
#                               OutputFile  
#-----  
[OutputFile]  
Name="%USERID%_%JOBNAME%_%JOBID%_%PRINTDATE%_%PRINTTIME%.pdf"
```

```

#-----
#                               Translations
#-----
[Translations]
Chars-="<>:"/\|?*"

Chars+="{}.'-_!!@"
Strings=6
String1-="      "
String1+="  "
String2-="      "
String2+="  "
String3-="      "
String3+="  "
String4-="      "
String4+="  "
String5-="      "
String5+="  "
String6-="PRT1"
String6+="000E"

#-----
#                               PDF / RTF Options
#-----
[PDF]
PaperSize=USFanFold
Orientation=Landscape
HorzMargin=0.25
FeedHoles=Yes
GreenBar=Yes
BarColor=Green
LPI=6
FontSize=12
Columns=0
Compress=Yes

```

**Figure 59: Job Separator Control File Sample**

The second sample shows a page from a report created through HercPrt using the predefined job separator control file for MVS 3.8J.

```

PPPPPPPPPP AAAAAAAAAA YY YY RRRRRRRRRR OOOOOOOOOO LL LL
PPPPPPPPPP AAAAAAAAAA YY YY RRRRRRRRRR OOOOOOOOOO LL LL
PP PP AA AA YY YY RR RR OO OO LL LL
PP PP AA AA YY YY RR RR OO OO LL LL
PPPPPPPPPP AAAAAAAAAA YYYY RRRRRRRRRR OO OO LL LL
PP AA AA YY RR RR OO OO LL LL
PP AA AA YY RR RR OO OO LL LL
PP AA AA YY RR RR OOOOOOOOOO LLLLLLLLLL LLLLLLLLLL
PP AA AA YY RR RR OOOOOOOOOO LLLLLLLLLL LLLLLLLLLL

JJJJJJJJJJ OOOOOOOOOO BBBBBBBBBB 00000000 5555555555 8888888888 6666666666 00000000
JJJJJJJJJJ OOOOOOOOOO BBBBBBBBBB 00000000 5555555555 8888888888 6666666666 00000000
JJ OO OO BB BB 00 00 00 55 88 88 66 66 00 00 00
JJ OO OO BB BB 00 00 00 55 88 88 66 66 00 00 00
JJ OO OO BBBBBBBBBB 00 00 00 55555555 88888888 6666666666 00 00 00
JJ OO OO BBBBBBBBBB 00 00 00 55555555 88888888 6666666666 00 00 00
JJ OO OO BB BB 00 00 00 55 88 88 66 66 00 00 00
JJ OO OO BB BB 0000 00 55 88 88 66 66 0000 00
JJ JJ OO OB BBBBBBBBBB 0000 00 55 88 88 66 66 000 00
JJJJJJJJ OOOOOOOOOO BBBBBBBBBB 00000000 5555555555 8888888888 6666666666 00000000
JJJJJJ OOOOOOOOOO BBBBBBBBBB 00000000 5555555555 8888888888 6666666666 00000000

**START**START**START**START**START**START**START**START**START**
*
* JOBID: JOB05860 *
* JOB NAME: PAYROLL *
* USER ID: IBMUSER *
* SYSOUT CLASS: P *
* OUTPUT GROUP: 1.1.1 *
* TITLE: *
*
* DESTINATION: LOCAL *
* NAME: IBM USER *
* ROOM: IBM *
* BUILDING: *
* DEPARTMENT: *
* ADDRESS: *
*
*
*
* PRINT TIME: 22:23:30 *
* PRINT DATE: 20 JUN 2011 *
* PRINTER NAME: PRT2 *
* SYSTEM: P390 *
*
**START**START**START**START**START**START**START**START**START**

```

Figure 60: HercPrt Report Sample

---

## 6.3 PRTPUB (Mainframe Print Publishing Program)

The mainframe print publishing program is written by Dan Nelson and can be obtained in the file section of the Hercules-390 discussion group (<http://groups.yahoo.com/group/hercules-390>).

### 6.3.1 Function

The mainframe print publishing program was designed to be used with Hercules' printing facilities. Hercules emulates line printers as simple text files containing printed output. This output can be viewed using a text editor or some other mechanism (such as being served via a web server).

This PRTPUB program converts the raw printer output to a more readable form by replacing form feed characters with line break lines, converting newlines to CR/LF pairs if desired (for Windows viewing), and placing the result in a specified file.

Several facilities are provided to allow for publishing the printed output in an automated way (e.g. to a web server) as follows.

- 1) You can choose to publish the output only if the input file size has changed since the last time the program was run. The file size is stored in a file for comparison on each run and the program terminates with return code 4 if the input file size has not changed since the last run. This allows efficient mirroring of the printed output to a separate file on a schedule, only updating the target file if the source file has changed. There is also an option that permits the polling for change to be carried out in this program, thus saving the resources that would otherwise be used to launch the program, parse the options, and so forth on each iteration.
- 2) You can choose to publish only the last part of the printed output. Hercules printer files tend to grow and grow in size and users are typically only interested in the last part, printed most recently. An option lets you specify the number of lines at the end of the file to publish to the target file, or lets you publish only the new data since the last run.
- 3) Locking is provided under UNIX type operating systems. As a courtesy this program will acquire an advisory read lock on the entire file before it starts processing. If the emulator uses an advisory write lock then this program can be scheduled to publish the output asynchronously to the emulator producing that output and the result will always be consistent.
- 4) If you want to publish the output as a web page without using a mechanism like server side includes, this program provides the option of supplying HTML tags to wrap the output, although this facility is only rudimentary. For sophisticated formatting server side includes are likely more appropriate to pull data into a page but this facility may be useful for simple scenarios.

The program should be used in a batch file. It is possible to use it for a single call or to build a loop for continuous processing (see examples below).

The possible return codes and their meaning are:

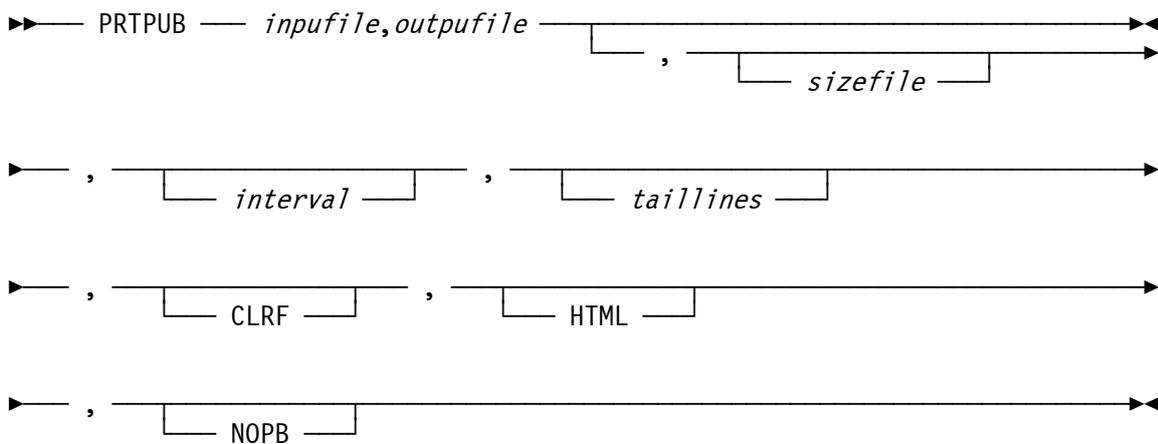
- |    |   |
|----|---|
| 0  | Translation was done.   |
| 4  | Translation was skipped because the file size did not change.   |
| 8  | Processing failed for an ordinary reason. An error message is sent to stderr indicating why.  |
| 16 | Processing failed due to some abnormality. An abend code is given for diagnostic purposes. This code will be the source line where the error occurred so you can use it with the source listing to diagnose the cause of the failure. |

## 6.3.2 Syntax

### Descriptive

```
PRT PUB inputfile,outputfile [, [sizefile] , [interval] , [taillines]  
      , [CRLF] , [HTML] , [NOPB]]
```

### Diagram



## 6.3.3 Parameter

The program takes a single long argument which must not contain spaces. If the file paths (on Windows) contain spaces then the entire argument must be enclosed in quotes. No spaces are allowed anywhere else, whether or not quotes are used. If quotes are used then they are only permitted as the first and last characters of the argument string.

The argument is divided into sub parameter separated by commas. All parameter are positional, if they are omitted a comma must mark their place.

- inputfile***            The file containing the emulated mainframe output.
- outputfile***          The file to which the result is to be published.
- sizefile***            The file containing the input file's size at the time of the previous run. If specified, this file is used to track the size of the input file, only publishing when it changes. The file will generally not exist the first time it is specified. In that case, the publishing is done and the file is written for use in subsequent runs. The file contains a number in binary form. It cannot be directly edited. If this parameter is not specified then the publishing is always done regardless of the file's size.
- interval***            The polling interval for use with the size file. If this argument is specified it must be a positive integer and it indicates a number of seconds between file size checks. In this mode if the input file's size has not changed since the last invocation (based on the size stored in the size file), the program goes into polling mode where it con-

tinually checks the size every interval seconds until a change is detected and then continues processing.

If this argument is not given the default behaviour is to return with a return code of 4 when the input file's size has not changed. Using this argument can improve the efficiency of simple polling in cases where nothing else is to be done on each polling interval. This is because it does not require the program to be invoked, arguments to be processed etc., on each polling interval. If no size file is specified this argument is checked for syntax but otherwise ignored.

### ***tail*lines**

The number of lines that should be published. If the input file contains more than the given number of lines then only the last <tail lines> lines are published. If this parameter is not specified the entire file is published. This number gives the number of printed lines published, including page separators. Note that the number may be off by one in either direction. If the final character of the file is a form feed then a trailing page separator is always output, which can add an extra line in some cases.

By the same token if the tail is broken on a form feed character then the entire form feed, which counts as two line breaks, is skipped, resulting in one fewer line of output. If this argument is 'ONLY' (in all caps) then the size file must also be specified. In that case the program translates only the part of the printer file that is new since the prior invocation, unless the new size is smaller than the old size, in which case the whole file is translated.

The purpose of this feature is to allow this program to be used as a sort of spool writer in conjunction with scripts that do something interesting with the incremental output, such as print it to a printer. In this mode empty output files are not produced. If the output is zero length and ONLY is specified, the program terminates with a return code of 4.

### **CRLF**

If specified then all linefeeds in the source file are translated to CR/LF pairs and CR/LF is used as the terminator for the page break line. If not specified no linefeed translation is done and the page break line is terminated with a newline character. This parameter is case sensitive (must be uppercase).

### **HTML**

If specified then HTML tags are added to the published output file. The file begins with <HTML><PRE> and ends with </PRE></HTML>. This parameter is case sensitive (must be uppercase).

### **NOPB**

If specified then form feed characters are not translated to page break lines. This is useful primarily for processing the published data after it is published, e.g. sending it to a real printer, etc. This argument must be in upper case. When NOPB is specified form feed characters are still bracketed with newlines to keep the maximum output record size consistent.

## **6.3.4 Examples**

### **Example 1:**

Publish the print output from print file PRT1.TXT and create a Word document named PRT1.DOC with carriage return/line feeds and do not translate form feed characters to page break lines.

```
D:\HERCUELS\PRTPUB D:\MVS\PRT\PRT1.TXT,D:\MVS\PRT\PRT1.DOC,,,,CRLF,,NOPB
```

**Example 2:**

Publish the print output from print file PRT2.TXT and create a HTML document named PRT2.HTML with carriage return/line feeds.

```
D:\HERCULES\PRTPUB D:\MVS\PRT\PRT2.TXT,D:\MVS\PRT\PRT2.HTML,,,CRLF,HTML
```

**Example 3:**

Publish the print output from print file PRT00E.TXT and create a HTML file called PUBL.HTML with carriage return / line feeds. The input file's size at the time of the previous run should be kept in file PRT00E.SIZ. The polling interval to check the file size is 10 seconds. The maximum number of lines that should be published is 1000.

```
@echo off
:LOOP
prtpub prt00e.txt,publ.html,prt00e.siz,10,1000,CRLF,HTML
if errorlevel 1 goto NOTDONE
echo PRINT FILE UPDATED
goto LOOP
:NOTDONE
echo ERROR - NO UPDATE DONE
goto LOOP
```

### **6.3.5 Sample Utility Output**

With the exception of error messages (if any) the utility does not produce any screen output.

---

## Part IV: Mainframe Utilities

---

## 7. Mainframe Utilities

---

### 7.1 AWSSL (AWS virtual tape)

AWSSL is written by Reed H. Petty and is available on the CBT MVS Utilities Tape (CBTTAPE) at [www.cbttape.org](http://www.cbttape.org) (File #585).

#### 7.1.1 Function

AWSSL moves MVS (OS390, etc.) datasets directly to/from AWS (and HET) virtual tape files. AWSSL strives to maintain compatibility with other AWS/HET utilities including AWSUTIL, VTTS, HETUTL and the Hercules Emulator.

AWSSL creates AWS and HET format virtual tapes which contain one to many datasets of any record format, with or without standard labels. The program will also retrieve datasets from AWS/HET files of any record format with the exception of spanned blocks. The retrieved datasets may be reblocked if necessary. If DCB attributes are omitted on the receiving dataset and if standard labels are present within the AWS structure, then the DCB attributes of the receiving dataset will be defaulted to those within the HDR1 label.

#### 7.1.2 Input Parameter

All input parameters are taken from control statements supplied by the AWSCNTL DD statement. The statements consist of a major function to be performed (e.g. TAPEVOL, EXPORT, etc.), and keywords which supply values to that function.

Control statement keywords may be continued to as many records as necessary. Continued statements are indicated by the last keyword argument suffixed with a comma and additional keywords on the next record. Additional keywords must not begin in column one.

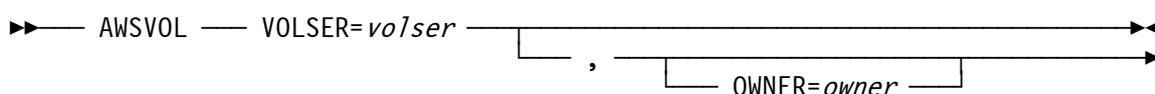
#### 7.1.3 AWSVOL Control Statement

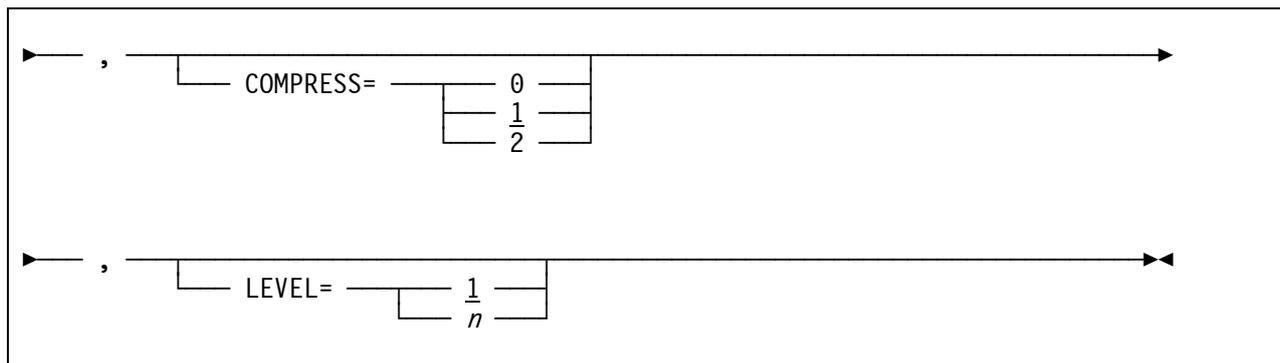
The AWSVOL control statement supplies characteristics of the virtual tape volume including volume serial number, owner and compression. AWSVOL must be the first control statement specified.

##### Descriptive

```
AWSVOL VOLSER=volser [,OWNER=owner] [,COMPRESS={0 | 1 | 2}] [,LEVEL={1 | n}]
```

##### Diagram





**VOLSER=volser** This specifies the volume serial number, volser is a 1 to 6 byte argument.

**OWNER=owner** The given owner will be included into the VOL1 owner field, owner is a 1 to 10 byte argument.

**COMPRESS=n** Compression, 0 = no compression, 1 = zlib, 2 = bzip. If COMPRESS=0 is coded then an AWS format is assumed. Default is HET format with COMPRESS=1 (zlib).

**LEVEL=n** Specifies the level of compression (0 – 9). Default is LEVEL=1.

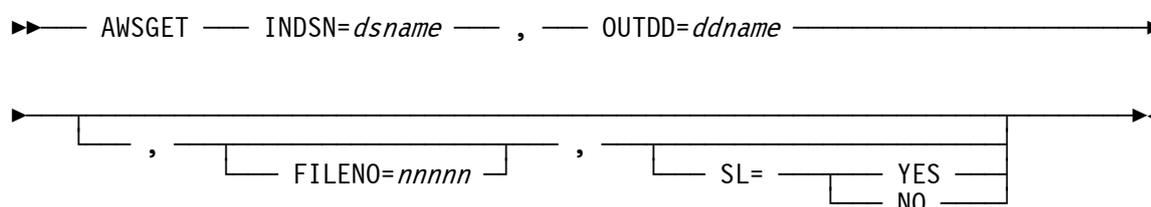
### 7.1.4 AWSGET Control Statement

The AWSGET control statement will supply values that are necessary to retrieve a dataset from an AWS or HET virtual tape volume. Decompression is invoked automatically when required.

#### Descriptive

AWSGET INDSN=*dsname*,OUTDD=*ddname* [,FILENO=*nnnnn*] [,SL={YES | NO}]

#### Diagram



**INDSN=dsname** This specifies the dataset name (up to 44 bytes) of a dataset stored inside of the AWS or HET virtual tape.

**OUTDD=ddname** This is the DDNAME representing the dataset to receive the data from the virtual tape.

**FILENO=nnnnn** Specifies the file number of the dataset inside of the AWS or HET virtual tape. This may be a standard label file number or an absolute file number depending on the value of the SL= keyword.

**SL=value** If the value is YES then standard label numbers are present, if the value is NO then the file numbers are absolute numbers. This keyword also impacts the meaning of the FILENO= keyword.

### 7.1.5 AWSPUT Control Statement

The AWSPUT control statement causes a dataset to be copied into the AWS or HET virtual tape file. Multiple AWSPUT statements may be specified. A set of standard labels are produced as each statement is processed. If necessary the dataset is staged into a temporary dynamically allocated dataset prior to insertion into the virtual tape.

The AWS or HET virtual tape output file may specify any DCB attributes that are meaningful in the users environment:

**RECFM=V** Variable length output, LRECL and BLKSIZE as specified. The records are in a format consistent with that produced by AWSUTIL.

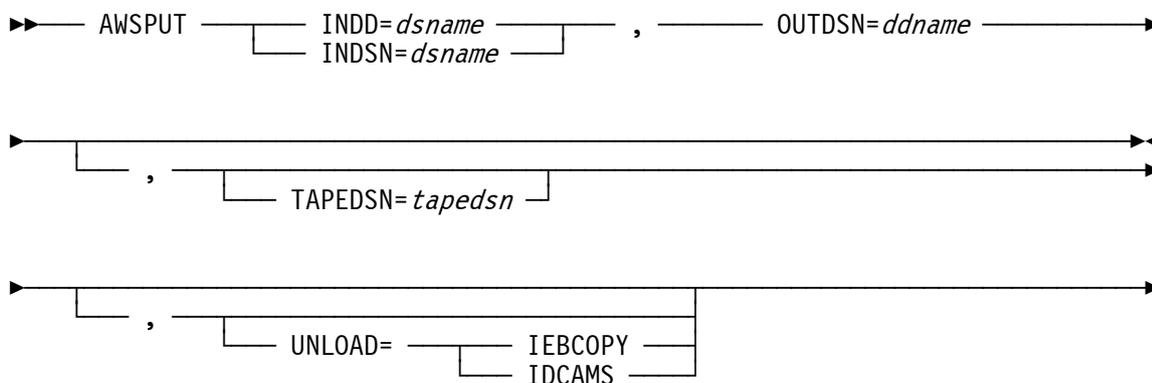
**RECFM=F** Fixed length output, LRECL and BLKSIZE as specified. The records are in a format consistent with that produced by the VTTS family of utilities.

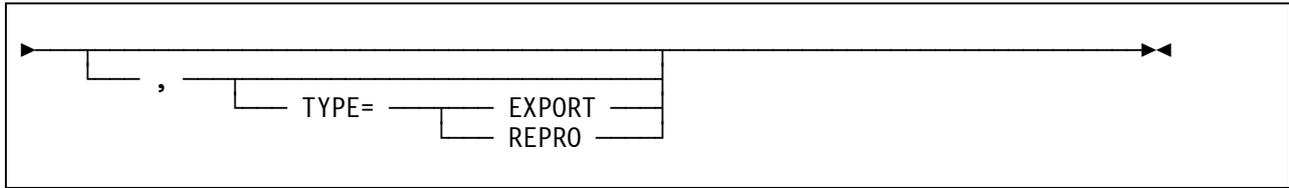
**RECFM=U** Undefined length output, BLKSIZE as specified. The records are written in an aggregated BLKSIZE length block. *Note:* This is the preferred method when the virtual tape is to be transported to other environments such as Hercules.

#### Descriptive

```
AWSPUT {INDD=ddname | INDSN=dsname},OUTDSN=dsname [,TAPEDSN=tapedsn]
      [,UNLOAD={IEBCOPY | IDCAMS}] [,TYPE={EXPORT | REPRO}]
```

#### Diagram





- INDD=ddname** Specifies the statically allocated DDNAME representing the file to be copied and placed into the AWS or HET virtual tape file.
- INDSN=dsname** Specifies the datasetname to be dynamically allocated and placed into the AWS or HET virtual tape file.
- OUTDSN=dsname** The dataset name (up to 44 bytes) to be placed into the labels which precede and follow the file on the virtual tape.
- TAPEDSN=tapedsn** The datasetname (up to 17 bytes) to be placed in the label.
- UNLOAD=value** Selects the utility (IEBCOPY or IDCAMS) to be called to stage the input dataset prior to insertion into the virtual tape.
- TYPE=value** If UNLOAD=IDCAMS is selected then the TYPE value specifies the method (EXPORT or REPRO) to be used to stage the dataset prior to insertion into the virtual tape.

## 7.1.6 Samples

### Sample 1:

Create an AWS virtual tape from several input datasets/input DDs:

```
//VIRTTAPE EXEC PGM=AWSSL
//STEPLIB DD DSN=your.loadlib,DISP=SHR
//AWSPRINT DD SYSOUT=*
//myddnam1 DD DSN=your.file1,DISP=SHR
//myddnam2 DD DSN=your.file2,DISP=SHR
//AWSFILE DD DSN=volser.aws,DISP=(NEW,CATLG,DELETE),
//          SPACE=(your_space_parms),
//          DCB=(your_DCB_parms)
//AWSCNTL DD *
AWSVOL VOLSER=mytape,COMPRESS=0
AWSPUT INDSN=your.first.dataset
AWSPUT INDSN=your.second.dataset,UNLOAD=IEBCOPY
AWSPUT INDD=myddnam1
AWSPUT INDD=myddnam1,UNLOAD=IEBCOPY
/*
```

**Figure 61: AWSSL JCL (Create AWS virtual tape)**

## Sample 2:

Create datasets from files on an AWS virtual tape:

```
//VIRTTAPE EXEC PGM=AWSSL
//STEPLIB DD DSN=your.loadlib,DISP=SHR
//AWSPRINT DD SYSOUT=*
//myddnam1 DD DSN=my.file1,DISP=NEW,CATLG,DELETE),
//          SPACE=(your_space_parms),
//          DCB=(your_DCB_parms)
//myddnam2 DD DSN=my.file2, DISP=NEW,CATLG,DELETE),
//          SPACE=(your_space_parms),
//          DCB=(your_DCB_parms)
//AWSFILE DD DSN=volser.aws,DISP=SHR
//AWSCNTL DD *
AWSVOL VOLSER=mytape
AWSGET OUTDD=myddnam1,INDSN=your.dataset1,FILENO=1,SL=YES
AWSGET OUTDD=myddnam2,INDSN=your.dataset2,FILENO=5,SL=YES
/*
```

**Figure 62: AWSSL JCL (Create datasets from AWS virtual tape)**

---

## 7.2 AWSUTIL (AWS format tape file generation utility)

AWSUTIL is written by Brandon Hill and is available on the CBT MVS Utilities Tape (CBTTAPE) at [www.cbttape.org](http://www.cbttape.org) (File #477).

### 7.2.1 Function

AWSUTIL is a comprehensive program for creating AWS format virtual tape files. The program can generate labels and write tape marks. Any dataset that can be accessed by BSAM can be copied to a virtual tape file. Any real tape device can have either a single file or the entire tape copied.

The commands for the utility are read from the SYSIN DD statement and status and errors are written to the SYSPRINT DD statement. The AWS tape image is written to SYSOUT as a VB file. This file is then suitable to be copied with FTP in binary mode to any other system.

Any number of DD statements representing sequential data and tapes can be specified and are read via the appropriate statement in SYSIN.

### 7.2.2 Control Statements

AWSUTIL supports these SYSIN control statements:

<b>READ <i>ddname</i></b>	Perform BSAM read from <i>ddname</i> and convert each block to AWS tape format. This works on whatever BSAM will read. The maximum BLKSIZE is 32760.
<b>GET <i>ddname</i></b>	Perform QSAM read from <i>ddname</i> and convert each block to AWS tape format. This works on whatever QBSAM will read. The maximum BLKSIZE is 32760.
<b>TAPEFILE <i>ddname</i></b>	Perform EXCP read from a real tape device and convert each block to AWS TAPE format. Only works on a tape device. Maximum BLKSIZE is 65535. This command stops reading when a tape mark is read. Therefore only the current tape file is processed. If BLP were used only the tape headers would be processed.
<b>TAPEALL <i>ddname</i></b>	Perform EXCP read from a real tape device and convert each block to AWS TAPE format. Only works on a tape device. Maximum BLKSIZE is 65535. This command rewinds the tape and reads until an error condition is returned. This effectively copies the entire contents of the tape. BLP is not necessary as a rewind command is sent before the tape is read. Tape marks are duplicated as they are encountered.  <i>NOTE:</i> Because end of reel reflectors are rarely effective this command can run a tape off the reel on a 3420 type drive. 3480 / 3490 / 3590 type drives have better stopping mechanisms.
<b>TAPEMARK</b>	Write a tape mark to the output file. This is usually used to separate individual files on a tape. Two consecutive tape marks sometimes signals the end of the tape.
<b>VERIFY</b>	Print a message after each command verifying that the command was executed ok. This merely prints a message. It has no effect on the integrity of the data. Because SYSOUT should be directed to disk there should be no data integrity problems anyway.

<b>ASCII</b>	All subsequent labels generated are converted to ASCII before they are written. This only affects labels. All data is copied exactly as it is read no matter the setting of this command.
<b>EBCDIC</b>	All subsequent labels generated are not to be converted before they are written. The implication is that the data is still in EBCDIC format. This only affects labels. All data is copied exactly as it is read no matter the setting of this command.
<b>VOL1</b>	These commands write 80 byte labels to the tape image. The format of the labels is controlled by the ASCII and EBCDIC commands. Except for the first 4 bytes of each input line none of the data is checked to make sure it is a valid tape label. It is up to the user to provide correctly formatted labels as desired. Essentially the entire 80 byte input record is written to the AWS TAPE image as is (or converted to ASCII if this flag is set).
<b>HDR1 - HDR2</b>	See VOL1 above.
<b>EOF1 - EOF2</b>	See VOL1 above.
<b>EOV1 - EOV2</b>	See VOL1 above.
<b>UHL1 - UHL8</b>	See VOL1 above.
<b>UTL1 - UTL8</b>	See VOL1 above.

### 7.2.3 Examples

The following example reads the first dataset from DASD and a second one dataset from tape and writes them to an AWS format dataset.

```
//MAKETAPE EXEC PGM=AWSUTIL
//STEPLIB DD DISP=SHR,DSN=my.load.library
//SYSPRINT DD SYSOUT=*
//SYSOUT DD DISP=(,CATLG),SPACE=(TRK,(15,15)),DSN=my.dsname
//INDATA DD DISP=SHR,DSN=my.input.sequential.data
//INTAPE DD DISP=OLD,DNS=my.real.tape.file
//SYSIN DD *
READ INDATA
TAPEMARK
TAPEFILE INTAPE
TAPEMARK
/*
```

**Figure 63: AWSUTIL utility JCL**

---

## 7.3 RAWSTAPE (Reverse AWSTAPE utility)

The RAWSTAPE program is written by Jan Jaeger and is available on the CBT MVS Utilities Tape (CBTTAPE) at [www.cbttape.org](http://www.cbttape.org) (File #478).

### 7.3.1 Function

The RAWSTAPE utility converts an AWSTAPE file back to a blocked OS dataset such that the original blocking is restored. As an AWSTAPE file does not contain DCB parameters this utility will not restore the original DCB parameters, these will have to be added manually with a utility such as IEBGENER.

Upload the AWSTAPE file to OS/390 or z/OS in binary format, do not use CR/LF or any other blocking feature when doing the file transfer. The file must be uploaded to a RECFM=U dataset, either preallocated or allocated using IND\$FILE. BLKSIZE=4096 is reasonable but any other blocksize will work also.

An AWS tape file can consist of more than 1 file and as such the file number of the tape must be specified. This file number is equal to the value used when reading a tape with bypass label processing (BLP). Thus, on a labeled tape file 1 will always be the VOL1 record.

Run RAWSTAPE with SYSUT1 pointing to AWS input file and SYSUT2 to the output file, for example copying a standalone dump tape.

### 7.3.2 Examples

The following example copies an AWS tape format dataset back to a blocked OS dataset:

```
//CONVERT EXEC PGM=RAWSTAPE , PARM=3
//STEPLIB DD DSN=IBMUSER.LOAD , DISP=SHR
//SYSUDUMP DD SYSOUT=*
//SYSUT1 DD DISP=SHR , DSN=IBMUSER.SADUMP.AWS
//SYSUT2 DD DSN=IBMUSER.SADUMP , DISP=(NEW,CATLG) ,
// UNIT=SYSALLDA , SPACE=(TRK,1200,RLSE)
```

**Figure 64: RAWSTAPE utility JCL**

After the previous example the IBMUSER.SADUMP contains the correct data but still has invalid DCB parameters in the DSCB, these can be corrected with IEBGENER as following:

```
//SETDCB EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT1 DD DUMMY , DCB=(DSORG=PS,RECFM=FBS,LRECL=4160,BLKSIZE=29120)
//SYSUT2 DD DISP=MOD,DCB=(*.SYSUT1),DSN=IBMUSER.SADUMP
```

**Figure 65: IEBGENER JCL for RAWSTAPE utility**

---

## 7.4 VTTS (Virtual tape transportation system)

The programs presented in this section are part of the VTTS (Virtual tape transportation system) and are copyrighted and owned by Sam Golob, however the programs on CBTTAPE File #533 can be used freely. The CBT MVS Utilities Tape (CBTTAPE) can be found at [www.cbttape.org](http://www.cbttape.org).

File #533 of the CBT tape contains the following programs:

- VTT2DISK
- VTT2TAPE
- VTT2CNVU
- VTT2FK2T
- VTT2T2FK

### 7.4.1 Function

The VTT2TAPE program converts AWS-format tape files to real tapes, while the VTT2DISK program creates an AWS-format virtual tape file from a real tape. The VTT2CNVU program converts a VB-format AWS-format tape (such as the one produced by the AWSUTIL program) to FB-80 format on MVS so that VTT2TAPE can be used subsequently to convert the data to a real tape.

The VTT2T2FK program is similar to VTT2DISK except that a real tape is converted to a FAKETAPE (tm) format tape (see below). VTT2FK2T is similar to VTT2TAPE except that a FAKETAPE tape image, folded over on MVS into FB-80 format, is converted into a real tape. Unless otherwise mentioned AWS-format virtual tape files on an MVS system have been folded over into fixed blocked 80-byte record format. AWS-format files on other systems are just long strings of data.

The FAKETAPE file format is a published interface of Fundamental Software Inc. Fundamental Software reserves the right to change the format at any time. FAKETAPE and FLEX-ES are registered trademarks of Fundamental Software Inc.

The AWS Tape format is produced by IBM's P/390 and Hercules systems to create disk files which an MVS system reads and writes as if they were actual tapes. Another name for these disk files is "virtual tapes", as the MVS system looking at these files "thinks" that they really are tapes.

There are programs like AWSUTIL which can read real tapes and create virtual tapes from them. These virtual tapes can be loaded on a Hercules system somewhere and be read as a tape.

VTT2DISK performs this function also, by reading a real tape and producing an FB-80 folded AWS-format tape as an MVS disk file. While reading the tape the VTT2DISK program produces a lot of statistics about the tape. The advantage of the VTT2DISK program is that it can be run on mainframe-based MVS systems that cannot normally produce AWS-format virtual tapes.

Another issue is encountered when transferring data in the other direction. Suppose you have an MVS system that cannot read a tape in AWS format. Most MVS systems which are not P/390 or Hercules machines are in this position. For these you have to convert an AWS virtual tape disk file to some format that the MVS system can convert into a REAL tape. That is the purpose of the VTT2TAPE program.

### 7.4.2 Control Statements

The VTT2DISK program allows the following keywords via the SYSIN DD Statement:

- |                       |   |
|-----------------------|---|
| <b>CHUNKSIZE=nnnn</b> | This specifies the chunksize. Any chunksize up to the limit of 65535 bytes is allowed. If not specified the default is 65535 bytes. |
| <b>NEWVOL=volser</b>  | This keyword changes the VOLSER on VOL1 label to <i>volser</i> .  |
| <b>READ</b>           | The READ keyword produces a read only run, no AWSOUT is created.  |

**IDRCOFF**

This keyword turns off the "data is compressed" indicators in the tape labels.

### 7.4.3 Examples

#### Example 1:

Read a tape and write it to an AWS format file.

```
//*****  
//* VTT2DISK - COPY TAPE TO DISK (AWS TAPE FORMAT) *  
//*****  
//AWSWRITE EXEC PGM=VTT2DISK  
//STEPLIB DD DSN=your.loadlib,DISP=SHR  
//AWSOUT DD DSN=your.aws.file,DISP=(NEW,CATLG),  
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=27920),  
// SPACE=(CYL,(750,750),RLSE)  
//TAPIN DD DISP=OLD,UNIT=3490,VOL=SER=xxxxxxx,  
// LABEL=(1,BLP,EXPDT=98000)  
//SYSPRINT DD SYSOUT=*  
//SYSFILES DD SYSOUT=*  
//SYSTRACE DD SYSOUT=*  
//SYSMOVED DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*
```

**Figure 66: VTT2DISK utility JCL.**

#### Example 2:

Read an AWS format file and write it to a tape.

```
//*****  
//* VTT2TAPE - COPY DISK (AWS TAPE FORMAT) TO REAL TAPE *  
//*****  
//AWSREAD EXEC PGM=VTT2TAPE  
//STEPLIB DD DSN=your.loadlib,DISP=SHR  
//AWSIN DD DSN=your.aws.file,DISP=SHR  
//TAPOUT DD DISP=OLD,UNIT=3490,VOL=SER=xxxxxxx,  
// LABEL=(1,BLP,EXPDT=98000)  
//SYSPRINT DD SYSOUT=*  
//SYSFILES DD SYSOUT=*  
//SYSTRACE DD SYSOUT=*  
//SYSMOVED DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*
```

**Figure 67: VTT2TAPE utility JCL**

### Example 3:

Convert an AWSUTIL file (VB) to standard AWS format (FB).

```
//*****  
//* VTT2CNVU - CONVERT AWSUTIL FILE (VB) TO STANDARD AWS FORMAT (FB) *  
//*****  
//AWSCNVT EXEC PGM=VTT2CNVU  
//STEPLIB DD DSN=your.loadlib,DISP=SHR  
//AWSVBIN DD DSN=your.vbaws.file,DISP=SHR  
//AWSFBOUT DD DSN=your.fbaws.file,DISP=(NEW,CATLG),  
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=27920),  
// SPACE=(CYL,(300,50),RLSE)  
//SYSPRINT DD SYSOUT=*  
//SYSFILES DD SYSOUT=*  
//SYSTRACE DD SYSOUT=*  
//SYSMOVED DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*
```

Figure 68: VTT2CNVU utility JCL

---

## Appendix A: Supported DASD Device Types

The following tables show the supported DASD device types and models with their sizes. The symbol “[\*]” in the size column means that any size can be specified, else the size defaults to the first listed model.

### CKD Devices

Devicetype-Model	Cylinders	Alternate Cylinders
IBM 2311	[*]	
IBM 2311-1	200	2
IBM 2314	[*]	
IBM 2314	200	3
IBM 3330	[*]	
IBM 3330-1	404	7
IBM 3330-2	808	7
IBM 3330-11	808	7
IBM 3340	[*]	
IBM 3340-1	348	1
IBM 3340-35	348	1
IBM 3340-2	696	2
IBM 3340-70	696	2
IBM 3350	[*]	
IBM 3350-1	555	5
IBM 3375	[*]	
IBM 3375-1	959	1
IBM 3380	[*]	
IBM 3380-1	885	1
IBM 3380-A	885	1
IBM 3380-B	885	1

Devicetype-Model	Cylinders	Alternate Cylinders
IBM 3380-D	885	1
IBM 3380-J	885	1
IBM 3380-2	1770	2
IBM 3380-E	1770	2
IBM 3380-3	2665	3
IBM 3380-K	2665	3
EMC 3380 K+	3339	3
EMC 3380 K++	3993	3
IBM 3390	[*]	1
IBM 3390-1	1113	1
IBM 3390-2	2226	2
IBM 3390-3	3339	1
IBM 3390-9	10017	3
IBM 3390-27	32760	3
IBM 3390-54	65520	3
IBM 9345	[*]	
IBM 9345-1	1440	0
IBM 9345-2	2156	0

**Table 9: Supported CKD DASD Devices**

## FBA Devices

Devicetype-Model	Blocks
IBM 3310	[*]
IBM 3310-1	125664
IBM 3370	[*]
IBM 3370-A1	558000
IBM 3370-B1	558000
IBM 3370-A2	712752
IBM 3370-B2	712752
IBM 9313	[*]
IBM 9313-1	246240
IBM 9332	[*]
IBM 9332-200	360036
IBM 9332-400	360036
IBM 9336-600	554800
IBM 9335	[*]
IBM 9335-1	804714
IBM 9336	[*]
IBM 9336-10	920115
IBM 9336-20	1672881
IBM 9336-25	1672881
IBM 0671-08	513072
IBM 0671	574560
IBM 0671-04	624456

**Table 10: Supported FBA DASD Devices**

---

## Appendix B. Syntax

This book uses two kinds of describing the syntax of the utilities. These are:

- Syntax descriptions
- Syntax diagrams

---

### B.1 Reading Syntax Descriptions

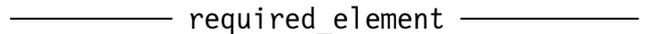
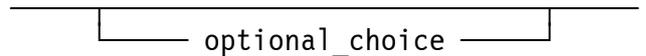
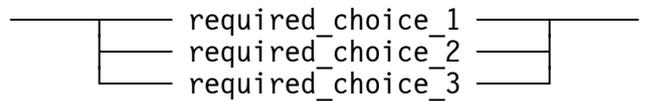
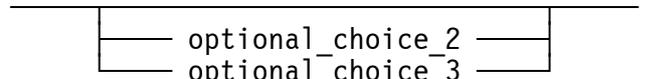
All syntax descriptions in this book use a common structure as described in the following table.

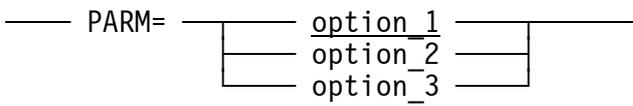
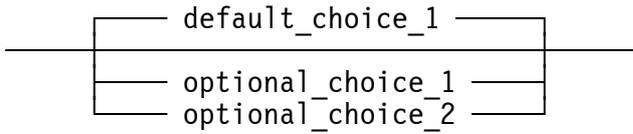
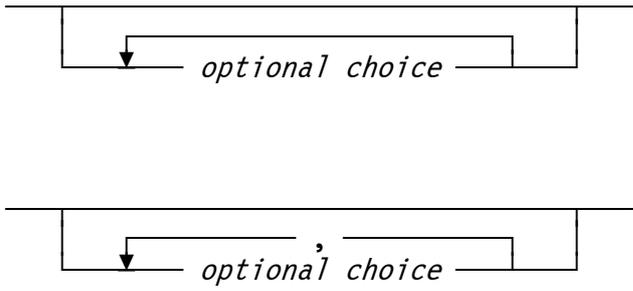
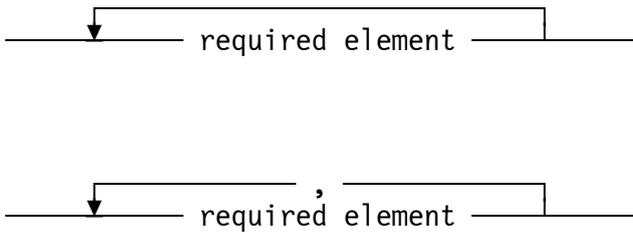
Syntax Element	Description
KEYWORDS	Keywords are denoted with upper case letters. Obey the spelling. In the actual statements or commands they can be coded in upper case or lower case letters.
<i>variables</i>	All user defined values are denoted with lower case italic letters. In the actual statements or commands they can be coded in upper case or lower case letters.
{ }	Signifies that all, or some portion, of the code elements between the braces are required elements. Note that the braces are not part of the statements and must be not coded.
[ ]	Signifies that all, or some portion of the code elements between the square brackets can optionally appear but are not required elements. Note that the square brackets are not part of the statements and must be not coded.
	The OR symbol signifies that you may use only one of the code elements or values from the possible choices. Note that the OR symbol is not part of the statements and must be not coded.
xxx, ...	Signifies that there can be more than one value in a comma delimited list. Note that the dots are not part of the statements and must be not coded.
xxx ...	Signifies that there can be more than one value in a blank space delimited list. Note that the dots are not part of the statements and must be not coded.

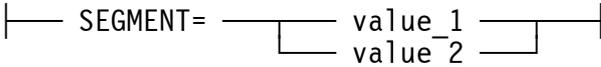
**Table 11: Reading Syntax Descriptions**

## B.2 Reading Syntax Diagrams

All syntax diagrams in this book (configuration statements, console commands and utilities) use a common structure as described in the following table.

Symbol	Description
	<p>This symbol indicates the beginning of a syntax diagram.</p>
	<p>This symbol indicates the end of a syntax diagram.</p>
	<p>This symbol indicates that the syntax diagram is continued on the next line.</p>
	<p>This symbol indicates that the syntax diagram is a continuation from the previous line.</p>
	<p>A required element (keyword or variable) appears on the main path of the horizontal line. You must specify this element.</p>
	<p>An optional element (keyword or variable) appears below the main path of the horizontal line. You may or may not specify this element.</p>
	<p>A required choice (keyword or variable) appears vertically stacked in the main path of the horizontal line. You must choose one of the available options.</p>
	<p>An optional choice (keyword or variable) appears vertically stacked below the main path of the horizontal line. You may choose one of the available options.</p>

Symbol	Description
	<p>A keyword with options. Only one of the available options may be specified. The underscored option is the default if the whole keyword statement is not coded.</p>
	<p>An optional choice (keyword or variable) with default appears vertically stacked with the default value above the main path of the horizontal line and the remaining optional elements below the main path of the horizontal line. Only one of the available options may be specified. If none of these elements is explicitly specified, the default above the main line is taken.</p>
	<p>An arrow returning to the left of an element below the main path of the horizontal line indicates an optional repeatable item. A character within the arrow path means that repeated items have to be separated by that character. If there is no character within the arrow path then the items are separated by a blank.</p>
	<p>An arrow returning to the left of an element on the main path of the horizontal line indicates an required repeatable item. A character within the arrow path means that repeated items have to be separated by that character. If there is no character within the arrow path then the items are separated by a blank.</p>
	<p>This symbol is a reference to a syntax segment, which is described separately below the main syntax diagram. Complex syntax diagrams are occasionally broken into separated simpler segments.</p>

Symbol	Description
	<p>This symbol indicates a syntax segment which is referenced from a main syntax diagram that is shown above the syntax segment.</p>
<p>KEYWORDS</p>	<p>Keywords are denoted with upper case letters. Obey the spelling. Lower case letters are optional and can be omitted (for example DISable). In the actual statements or commands the keywords can be coded in upper case or lower case letters.</p>
<p><i>variables</i></p>	<p>All user defined values are denoted with lower case italic letters. They represent user supplied names or values. In the actual statements or commands they can be coded in upper case or lower case letters.</p>

**Table 12: Reading Syntax Diagrams**

## B.3 Sample Syntax Description

The following figure shows a sample of a complex syntax description. This is not an example of an existing system parameter or panel command. It is used mainly to demonstrate the “look and feel” of syntax descriptions.

### Syntax

```

CMDNAME required_argument [optional_argument]
      {required_choice_1 | required_choice_2 | required_choice_3}
      [optional_choice_1 | optional_choice_2]
REQUIRED_KEYWORD=variable
[OPTIONAL_KEYWORD=variable]
[DEFAULT_KEYWORD | KEYWORD_1 | KEYWORD_2]
[KEYWORD=default choice | KEYWORD={choice_1 | choice_2}]
repeatable_item_n {repeatable_item_n ...}
repeatable_item_n {,repeatable_item_n ,...}
fragment_name
(variable_1, variable_2)

```

### fragment\_name

[DEFAULT\_KEYWORD | KEYWORD\_1 | KEYWORD\_2 | KEYWORD\_3 | KEYWORD\_4]

KEYWORD={*choice\_1* | *choice\_2* | *choice\_3* | *choice\_4*}

Figure 69: Sample Syntax Description

## B.4 Sample Syntax Diagram

The next figure shows a sample of a complex syntax diagram. It shows the same example as in the syntax description in the previous section. Like in the example before it is not based on an existing system parameter or panel command. It is used mainly to demonstrate the “look and feel” of syntax descriptions.

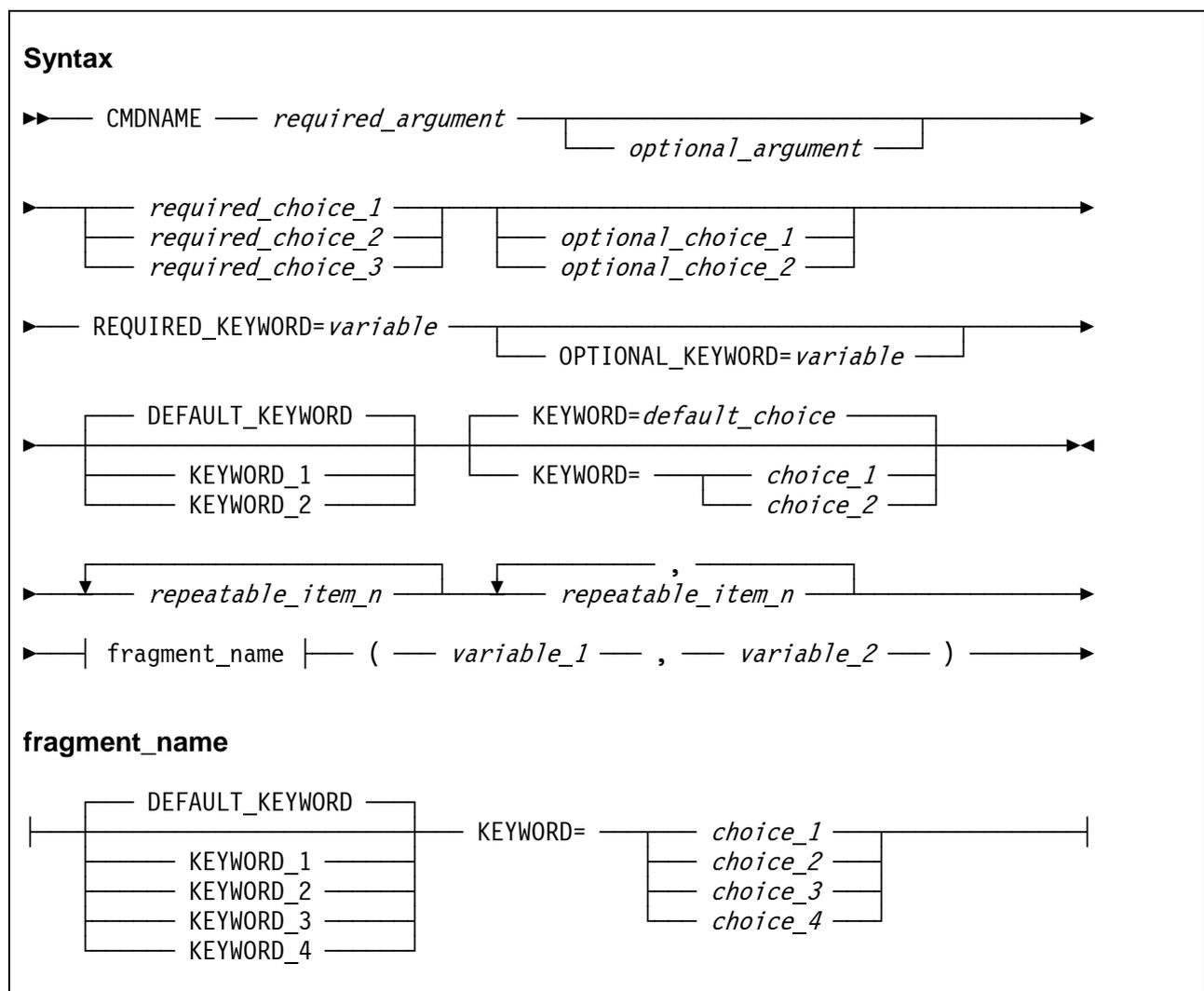


Figure 70: Sample Syntax Diagram

---

## Appendix C. Links

- **The Hercules System/370, ESA/390, and z/Architecture Emulator**

<http://www.hercules-390.eu>

- **Hercules source code repositories**

<https://github.com/rbowler/spinhawk> (release 3.xx development stream)

<https://github.com/rbowler/sandhawk> (release 4.xx development stream)

<https://github.com/hercules-390/hyperion> (cutting-edge developer sandbox)

- **Hercules Developer Snapshots (Dave Wade)**

<http://www.smrcc.org.uk/members/g4ugm/snapshots/>

- **Hercules PDF Documentation (Peter Glanzmann)**

<http://hercdoc.glanzmann.org>

- **The MVS Tur(n)key System, Version 3 (Volker Bandke)**

<http://www.bsp-gmbh.com/turnkey/index.html>

- **Hercules WinGUI (“Fish”, David B. Trout)**

<http://www.softdevlabs.com/Hercules/hercgui-index.html>

- **CTCI-WIN (“Fish”, David B. Trout)**

<http://www.softdevlabs.com/Hercules/CTCI-WIN-index.html>

- **Hercules Studio (Jacob Dekel)**

<http://www.mvsdasd.org/hercstudio>

- **Hebe – Hercules Image Manager (Robin Atwood)**  
<http://kde-apps.org/content/show.php/Hebe?content=126738>
- **WinPcap, Politecnico di Torino**  
<http://www.winpcap.org>
- **Vista tn3270, Tom Brennan Software**  
<http://www.tombrennansoftware.com>
- **X3270, Paul Mattes**  
<http://x3270.bgp.nu>
- **AWSBROWSE (“Fish”, David B. Trout)**  
<http://www.softdevlabs.com/Hercules/hercgui-index.html>
- **XMIT Manager**  
[www.cbttape.org](http://www.cbttape.org)
- **CBT MVS Utilities Tape (CBTTAPE)**  
[www.cbttape.org](http://www.cbttape.org)
- **Microsoft Visual C++ 2008 Express**  
<http://www.microsoft.com/express/download/>

- **ZLIB**

<http://www.zlib.net>

<http://www.softdevlabs.com/Hercules/ZLIB1-1.2.3-bin-lib-inc-vc2008-x86-x64.zip>

- **BZIP2**

<http://www.bzip.org>

<http://www.softdevlabs.com/Hercules/BZIP2-1.0.5-bin-lib-inc-vc2008-x86-x64.zip>

- **PCRE**

<http://www.pcre.org>

<http://www.softdevlabs.com/Hercules/PCRE-6.4.1-bin-lib-inc-vc2008-x86-x64.zip>

- **Regina REXX**

<http://regina-rexx.sourceforge.net/>

- **Open Object Rexx (ooRexx)**

<http://www.oorexx.org/>

Hercules Emulator

<p data-bbox="644 954 703 1348">Hercules System/370, ESA/390, z/Architecture Emulator</p> <p data-bbox="778 853 887 1447"><b>Operations and Utilities Guide</b></p> <p data-bbox="962 1021 986 1283">Version 4 Release 00</p>
HEOP040000-00