

HP OpenVMS

V8.3 リリース・ノート【翻訳版】

注文番号: BA322-90060

2006 年 10 月

本書は、本ソフトウェアに追加された変更点について説明します。また、インストール、アップグレード、互換性の情報についても説明します。さらに、新規および既存のソフトウェアの問題点と制限事項、ソフトウェアとドキュメントの訂正についても説明します。

改訂 / 更新情報:

ソフトウェア・バージョン:

新規マニュアルです。

OpenVMS I64 Version 8.3

OpenVMS Alpha Version 8.3

© Copyright 2006 Hewlett-Packard Development Company, L.P.

本書の著作権は Hewlett-Packard Development Company, L.P. が保有しており、本書中の解説および図、表は Hewlett-Packard Development Company, L.P. の文書による許可なしに、その全体または一部を、いかなる場合にも再版あるいは複製することを禁じます。

また、本書に記載されている事項は、予告なく変更されることがありますので、あらかじめご承知おきください。万一、本書の記述に誤りがあった場合でも、弊社は一切その責任を負いかねます。

本書で解説するソフトウェア (対象ソフトウェア) は、所定のライセンス契約が締結された場合に限り、その使用あるいは複製が許可されます。

日本ヒューレット・パカードは、弊社または弊社の指定する会社から納入された機器以外の機器で対象ソフトウェアを使用した場合、その性能あるいは信頼性について一切責任を負いかねます。

Intel および Itanium は、米国およびその他の国における、Intel Corporation またはその関連会社の登録商標です。

Java は、米国における Sun Microsystems, Inc. の商標です。

Oracle は、米国における Oracle Corporation, Redwood City, California の登録商標です。

OSF および Motif は、米国およびその他の国における The Open Group の商標です。UNIX は、The Open Group の登録商標です。

Microsoft, Windows, Windows NT, および MS Windows は、米国における Microsoft Corporation の登録商標です。

X/Open は、英国およびその他の国における X/Open Company Ltd. の登録商標です。X device は、英国およびその他の国における X/Open Company Ltd. の商標です。

ZK6677

原典：HP OpenVMS Version 8.3 Release Notes

© 2006 Hewlett-Packard Development Company, L.P.

本書は、日本語 VAX DOCUMENT V 2.1を用いて作成しています。

目次

まえがき	xv
1 OpenVMS ソフトウェアのインストールおよびアップグレードに関する注 意事項	
1.1 弊社ソフトウェアのテクニカル・サポート方針	1-1
1.2 アプリケーションの互換性	1-3
1.3 修正キットの入手方法	1-3
1.4 ネットワーク・オプション	1-3
1.5 古いバージョンの OpenVMS と互換性のないディスク	1-4
1.6 HP DECwindows Motif for OpenVMS	1-5
1.7 Integrity サーバ用の OpenVMS ユーザ向けの注意事項	1-5
1.7.1 Version 8.3 での A6825A Gigabit Ethernet コントローラと RX4640 の 制限	1-5
1.7.2 rx7620 および rx8620 での U160 SCSI のサポート	1-6
1.7.3 Integrity Server での System Event Log (SEL) のクリア	1-6
1.7.4 Integrity サーバ用のファームウェア	1-7
1.7.5 インストール DVD からのブート	1-9
1.7.6 HP DECwindows Motif に関する注意事項	1-9
1.7.6.1 キーボードのサポート	1-10
1.7.6.2 サーバ起動前の周辺デバイスの接続	1-10
1.7.6.3 スタートアップ中に表示されるカウントダウン・メッセー ジ	1-10
1.8 OpenVMS Alpha 向けの注意事項	1-10
1.8.1 OpenVMS Alpha Version 8.3 のファームウェア	1-11
1.8.2 アップグレード・パス	1-11
1.9 CDSA: 他社製品の Secure Delivery 署名について OpenVMS V8.3 のリリース後に 使用可能	1-13
1.10 SYS\$MANAGER:SYSTARTUP_VMS.COM から暗号化や SSL を開始する行を削 除	1-13
1.11 Kerberos for OpenVMS	1-13
1.12 OpenVMS オペレーティング・システムとともにインストールされる Encryption for OpenVMS	1-16
1.13 HP DECram V3.n: アップグレード前に削除する	1-16
1.14 アップグレード後に LANCP デバイス・データベースを変換する	1-17
1.15 DECnet-Plus の新しいバージョンが必要	1-17
1.16 SHADOW_MAX_UNIT のデフォルト設定とメモリ使用量	1-18

1.17	アップグレード前に TIE Kit の削除が必要	1-19
1.18	レイヤード・プロダクトを代替デバイスや代替ディレクトリにインストールすると失敗する	1-19
2	OpenVMS の関連製品に関する注意事項	
2.1	関連製品のサポート	2-1
2.2	追加された CMAP ファイル	2-2
2.3	COBOL: I/O 実行時診断と RMS 特殊レジスタの変更	2-2
2.4	COM for HP OpenVMS (Alpha のみ)	2-2
2.4.1	COM for OpenVMS のサポート	2-3
2.4.2	高負荷時に発生するアプリケーションのレジストリ・アクセス・エラー	2-3
2.5	OpenVMS Version 8.3 では DECdfs Version 2.4 が必要	2-3
2.6	DEC PL/I: OpenVMS での RTL のサポート	2-3
2.7	FMS キット	2-4
2.8	Graphical Configuration Manager (GCM) (Alpha のみ)	2-4
2.9	HP DCE RPC for OpenVMS	2-4
2.9.1	DCE RPC による FAILSafe IP のサポート	2-4
2.9.2	RPC ソケットにおけるバッファ・サイズ・チューニングのサポート	2-5
2.9.3	RTI (Remote Task Invocation) RPC (I64 のみ)	2-5
2.9.4	Microsoft LAN Manager RPC はテストされていない (I64 のみ)	2-5
2.9.5	utc_multitime 係数指数の型	2-5
2.9.6	G_FLOAT 浮動小数点型と IEEE 浮動小数点型のサポート	2-6
2.10	HP DECnet-Plus での大文字小文字混在のパスワードのサポート	2-7
2.11	HP DECnet-Plus for OpenVMS: X.25 データ・リンクがサポートされていない (I64 のみ)	2-8
2.12	HP DECram	2-8
2.12.1	DECram は OpenVMS Version 8.2 に含まれる	2-8
2.12.2	DECRYPT DCL コマンドとの競合	2-9
2.13	HP DECwindows Motif for OpenVMS	2-9
2.13.1	DECwindows のポーズ画面でロック解除できない問題 (Alpha のみ)	2-9
2.13.2	新しいロケールの追加	2-10
2.13.3	ユーザが作成したトランスポートはサポートされない	2-10
2.14	HP Secure Web Server のサポート	2-10
2.15	HP Pascal for OpenVMS Alpha Systems	2-11
2.15.1	STARLET ライブラリの作成には V5.8A (以降) が必要 (Alpha のみ)	2-11
2.15.2	アップグレード後の HP Pascal のインストール (Alpha のみ)	2-12
2.16	I64 システムでの WEBES および SEA のサポート	2-12
2.17	NetBeans Version 3.6 では Java Standard Edition, Development Kit v 1.4.2-x が必要	2-12

3 一般ユーザ向けの注意事項

3.1	OpenVMS Freeware CD	3-1
3.1.1	フリーウェア・メニューが利用できない (I64 のみ)	3-2
3.2	DCL コマンド	3-2
3.2.1	CREATE/MAILBOX: 一時的な制約	3-2
3.2.2	DIAGNOSE コマンドはサポートされない	3-2
3.3	Open Source Tools CD での DECmigrate の提供について	3-3
3.4	HP Secure Web Browser	3-3
3.4.1	必要メモリ量の増加	3-3
3.4.2	ODS-2 ディスク・ボリュームで発生するインストール・エラー (I64 のみ)	3-3
3.5	ドキュメントの訂正	3-4
3.5.1	『HP OpenVMS System Analysis Tools Manual』	3-4
3.5.2	『HP OpenVMS Programming Concepts Manual』	3-4
3.5.3	『HP OpenVMS デバッガ説明書』: クライアント/サーバ・インタフェースの訂正	3-4
3.5.4	『HP OpenVMS DELTA/XDELTA Debugger Manual』 の更新	3-5
3.5.5	『HP OpenVMS Utility Routines Manual』 の更新	3-6
3.5.6	『OpenVMS I/O User's Reference Manual』: PTD\$READ の明確化	3-6
3.5.6.1	PTD\$疑似端末ドライバの訂正	3-7
3.5.7	『HP OpenVMS Version 8.2 新機能説明書』: Librarian ユーティリティの訂正	3-7
3.5.7.1	/REMOVE 修飾子の訂正	3-7
3.5.7.2	ELF オブジェクト・ライブラリへのアクセスについての訂正	3-7
3.5.8	『OpenVMS RTL Library (LIB\$) Manual』 の訂正	3-8
3.5.8.1	『OpenVMS RTL Library (LIB\$) Manual』: LIB\$CVT_DX_DX の丸め規則の明確化	3-9
3.5.9	『OpenVMS RTL Library (LIB\$) Manual』: プラットフォームの制限の明確化	3-9
3.5.10	『OpenVMS システム管理者マニュアル』: IPC コマンドの制限	3-10
3.5.11	『OpenVMS System Services Reference Manual』: \$PUTMSG システム・サービスの訂正	3-10
3.5.12	『Volume Shadowing for OpenVMS 説明書』: メモリ要件の訂正	3-11
3.6	Version 8.2 から Version 8.2-1 へのネットワーク・アップデートの制限	3-11
3.7	同期型データ・リンクはサポートされていない	3-12
3.8	LAN ドライバから報告される二重モード不一致エラー	3-12
3.9	ブートの際の AUDIT_SERVER の起動失敗	3-13

4 システム管理に関する注意事項

4.1	Monitor ユーティリティの変更	4-1
4.1.1	MONITOR データのバージョン間での互換性	4-1
4.1.2	記録ファイルからのデータのブレイバック	4-1
4.1.3	VMS クラスタでの生のリモート・データの監視	4-2
4.2	ファイル・セキュリティ属性の推奨値の更新	4-3
4.3	システム管理についての注意事項	4-3
4.3.1	CIMSERVER プロセスの推奨事項 (I64 のみ)	4-3

4.4	システムのハングアップまたはクラッシュからの回復 (I64 のみ)	4-4
4.5	Oracle 8iおよび 9iでの DECdtm/XA (Alpha のみ)	4-4
4.6	デバイス・ユニットの最大数の増加	4-5
4.7	ECP Data Collector と Performance Analyzer V5.5 (Alpha のみ)	4-5
4.8	EDIT/FDL: 推奨バケット・サイズの変更	4-5
4.9	EFI\$CP ユーティリティ: 使用は推奨できない	4-6
4.10	EFI シェルで共用システム・ディスクまたはシャドウ・システム・ディスクを操作する場合の注意事項	4-6
4.11	Error Log Viewer (ELV) ユーティリティ: TRANSLATE/PAGE コマンド	4-8
4.12	外部認証	4-8
4.12.1	I64 外部認証サポート	4-8
4.12.2	DECterm 端末セッションでの SET PASSWORD の動作	4-8
4.12.3	ワークステーションではパスワードの有効期限切れは通知されない	4-9
4.13	OpenVMS Cluster システム	4-9
4.13.1	OpenVMS I64 でのクラスタのサポート	4-9
4.13.2	現時点での制限事項	4-10
4.13.3	サテライト・ブートと LAN フェールオーバー	4-10
4.13.4	Integrity サーバのサテライト・システムでのエラー・ログ・ダンプ・ファイルの作成	4-11
4.14	クラスタ互換性パッチ・キット	4-11
4.14.1	クラスタの互換性のために必要なパッチ・キット	4-12
4.14.2	Fibre Channel および SCSI マルチパスと、他社製品との非互換性を修正する API	4-14
4.14.3	DDT Intercept Establisher ルーチンとデバイス構成通知結果	4-15
4.14.4	CI と LAN との間の回線切り替えによるクラスタの性能の低下	4-15
4.14.5	マルチパス・テープ・フェールオーバーの制限事項	4-16
4.14.6	SCSI マルチパス媒体チェンジャでは自動フェールオーバーは行われ ない	4-17
4.14.7	Availability Manager AVAIL_MAN_BASE キットがロック競合情報の 収集に失敗する	4-17
4.15	OpenVMS Galaxy (Alpha のみ)	4-17
4.15.1	Galaxy の定義	4-17
4.16	セル型システムでの複数の nPartitions	4-18
4.16.1	OpenVMS Graphical Configuration Manager	4-18
4.16.2	ES40 上の Galaxy: 非圧縮ダンプの制限事項	4-19
4.16.3	ES40 上の Galaxy: Fast Path の無効化	4-19
4.17	OpenVMS Management Station	4-19
4.18	OpenVMS Registry は Version 2 フォーマットのデータベースを壊すことがあ る	4-19
4.19	システム・パラメータ	4-20
4.19.1	新しいシステム・パラメータ	4-20
4.19.2	廃止されたシステム・パラメータ	4-20
4.19.3	システム・パラメータの変更	4-21
4.19.4	ドキュメントの誤り: LCKMGR_CPUID システム・パラメータ	4-22
4.19.5	MMG_CTLFLAGS: ドキュメントの誤り	4-22
4.20	Terminal Fallback Facility (TFF)	4-22

4.21	User Environment Test Package (UETP) (I64 のみ)	4-23
4.22	推奨するキャッシュ方式	4-24
4.23	Volume Shadowing for OpenVMS	4-24
4.23.1	デバイス名の必要条件	4-24
4.23.2	DCL コマンド・プロシージャ内での SET SHADOW と SHOW SHADOW の使用についての警告	4-25
4.23.3	書き込みビットマップと異種デバイス・シャドウイング (DDS) の注意事 項	4-25
4.23.4	KZPDC (Smart Array 5300) の制限事項	4-27
4.23.5	シャドウ・セット・マージ遅延の算出方法の変更	4-28
4.23.6	/MINICOPY を使用したシャドウ・セット・メンバのディスマウン ト	4-28
4.24	Authentication and Credential Management Extension (ACME)	4-29
4.24.1	新しい機能と変更された機能	4-29
4.25	OpenVMS と Kerberos ACME エージェントのデフォルトの起動順序は必須	4-30
4.26	トレースバック API の問題の修正	4-31
4.27	WBEM Services for OpenVMS Version 2.0 の注意事項	4-31
4.27.1	ベースは OpenPegasus 2.5	4-31
4.27.2	nPartitions と iCAP のサポート	4-31
4.27.3	共通システム・ディスクのクラスタ向けに設計されていない	4-32
4.27.4	OpenVMS 上でプロバイダをアンロードするには cimserver.exe を再起 動する	4-32
4.27.5	コマンド行オプションは引用符で囲む	4-32
4.27.6	アンインストール後の WBEMCIM リポジトリ・ディレクトリ・ツリー のクリーン・アップ	4-32
4.28	I64 では \$SIGNAL_ARRAY_64 システム・サービスがない	4-32
4.29	『HP OpenVMS System Analysis Tools Manual』	4-33
4.29.1	DUMPSTYLE の各ビットの定義	4-33
4.29.2	システム・ダンプの保存	4-33
4.29.3	システムをリブートする際の SDA の起動	4-34
4.29.4	SDA CONTEXT	4-34
4.29.5	SDA シンボルの初期化	4-35
4.29.6	ANALYZE	4-35
4.29.7	Crash_Dump	4-35
4.29.8	SDA コマンド	4-35
4.29.9	COPY コマンド	4-36
4.29.10	DUMP コマンド	4-37
4.29.11	EVALUATE コマンド	4-37
4.29.12	MAP コマンド	4-37
4.29.13	SET CPU コマンド	4-38
4.29.14	SET PROCESS コマンド	4-38
4.29.15	READ コマンド	4-38
4.29.16	SEARCH コマンド	4-39
4.29.17	SHOW_CALL_FRAME コマンド	4-39
4.29.18	SHOW CLUSTER コマンド	4-40
4.29.19	SHOW CRASH コマンド	4-40
4.29.20	SHOW DUMP コマンド	4-40
4.29.21	SHOW PROCESS コマンド	4-42
4.29.22	SHOW RESOURCES コマンド	4-43
4.29.23	SHOW SPINLOCKS コマンド	4-43

4.29.24	SHOW MEMORY コマンド	4-44
4.29.25	SHOW GCT コマンド	4-44
4.29.26	SHOW CPU コマンド	4-44
4.29.27	SHOW SYMBOL コマンド	4-44
4.29.28	SHOW UNWIND コマンド	4-45
4.29.29	SHOW SWIS コマンド	4-45
4.29.30	CLUE CONFIG コマンド	4-45
4.29.31	CLUE REGISTER コマンド	4-46
4.29.32	I64 ISD_Labels インデックスの表	4-46
4.29.33	SDA 拡張機能のコンパイルとリンク	4-47
4.29.34	拡張機能のデバッグ	4-47
4.29.35	呼び出し可能ルーチンの概要	4-47
4.29.36	ターゲット・システムでの接続の設定	4-48
4.29.37	OpenVMS Alpha システム・ダンプ・デバッグ	4-61
4.29.38	OpenVMS Alpha ウォッチポイント・ユーティリティ	4-63
4.29.39	SET PROCESS/LOG コマンド	4-64
5	プログラミングに関する注意事項	
5.1	システム・サービスの変更点	5-1
5.1.1	追加	5-1
5.2	システム・サービス\$GETQUI および\$SNDJBC のジョブ・リミット・アイテム・コードの範囲の拡大	5-2
5.3	特権プログラムの再コンパイルが必要 (Alpha のみ)	5-2
5.4	特権データ構造体の変更	5-2
5.4.1	KPB 拡張	5-3
5.4.2	CPU の名前空間	5-4
5.4.3	64 ビットの論理ブロック番号 (LBN)	5-4
5.4.4	動的スピンロックのフォーク	5-4
5.4.5	UCB と DDB のアップデート	5-5
5.4.6	PCBST_TERMINAL のサイズの拡張	5-5
5.4.7	スレッド単位のセキュリティは特権付きコードとデバイス・ドライバに影響する	5-6
5.5	浮動小数点型データを使用するアプリケーション	5-8
5.5.1	IEEE 浮動小数点フィルタ (I64 のみ)	5-8
5.5.2	Ctrl/C と STOP ボタンを使用する場合の制限 (OpenVMS Alpha)	5-9
5.5.3	Ada イベントのサポート (I64 のみ)	5-9
5.5.4	SHOW SYMBOL/TYPE がアレイ・サイズを正しく報告するようになった (Alpha および I64)	5-9
5.5.5	EXAMINE/INSTRUCTION %PREVLOC コマンドが修正された (I64 のみ)	5-10
5.5.6	SHOW MODULE コマンドがモジュール・サイズを計算するようになった (I64 のみ)	5-10
5.5.7	C++ 言語の問題 (I64 のみ)	5-10
5.6	Ada コンパイラ (I64 のみ)	5-11
5.7	Backup API: ジャーナリング・コールバック・イベントの制限事項	5-11
5.8	C プログラム: CASE_LOOKUP=SENSITIVE を設定したコンパイル	5-11
5.9	C ランタイム・ライブラリ	5-12
5.9.1	C RTL TCP/IP ヘッダ・ファイルのアップデート	5-12
5.9.2	toascii 関数の追加	5-13

5.9.3	64 ビットの sigaction の問題の修正	5-13
5.9.4	一部の算術関数への、64 ビット・ポインタ機能の追加	5-13
5.9.5	2 GB の malloc で何も表示せずに失敗することがなくなった	5-13
5.9.6	exec* のメモリ・リークの修正	5-14
5.9.7	execl 失敗後の exit 動作の修正	5-14
5.9.8	confstr の拡張	5-14
5.9.9	fopen の失敗の修正	5-15
5.9.10	ファイル・ポインタ・ロックのハングアップの修正	5-15
5.9.11	バックポート・ライブラリが出荷されなくなった	5-15
5.9.12	ヘッダ・ファイル<time.h>の変更	5-15
5.9.13	ヘッダ・ファイル<time.h>での非 ANSI の*_r 関数の参照	5-16
5.9.14	ヘッダ・ファイル<decc\$types.h>: time_t int 宣言	5-17
5.9.15	新しい DECC\$SHRP.EXE イメージ	5-17
5.9.16	ヘッダ・ファイル<wchar.h>と C++ %CXX-W-ENVIRSTKDIRTY メッセージ	5-17
5.9.17	ヘッダ・ファイル<builtins.h>の__CMP_SWAP*と_Interlocked*が C++ から参照可能	5-17
5.9.18	fcntl への余分なパラメータの無視	5-18
5.9.19	システムの MAXBUF パラメータが大きい場合、stdout への fwrite が失敗する問題	5-18
5.9.20	64K バイトを超えるソケット転送の読み取りと書き込みの問題	5-18
5.9.21	I64 システムでの nanosleep の問題	5-18
5.9.22	I64 システムへのビルトイン __fci の追加	5-19
5.9.23	_FAST_TOUPPER マクロの追加	5-19
5.9.24	atof("NaN") の呼び出しで算術トラップが生成されなくなった	5-19
5.9.25	DECC\$*.OLB オブジェクト・ライブラリに新しいエントリがない	5-20
5.10	呼び出し標準規則とローテートするレジスタ (I64 のみ)	5-20
5.11	Common Data Security Architecture (CDSA) に関する考慮	5-20
5.11.1	Secure Delivery	5-21
5.11.2	インストールと初期化に関する考慮	5-21
5.12	デバッグ・モード: CPUSPINWAIT バグ・チェックの回避	5-22
5.13	Delta/XDelta デバッガ	5-22
5.13.1	XDelta のレジスタ表示に関する考慮 (I64 のみ)	5-22
5.14	ファイル・アプリケーション: 『Guide to OpenVMS File Applications』の訂正	5-23
5.15	RMS 構造体についての HP BLISS コンパイラの警告 (I64 のみ)	5-23
5.16	RMS の Must-Be-Zero エラーの可能性: FAB 内に新しいファイル・オプション用の場所を確保	5-24
5.17	HP COBOL ランタイム・ライブラリ (RTL)	5-25
5.18	I64 用の HP Fortran	5-25
5.19	OpenVMS 用 HP MACRO	5-27
5.19.1	OpenVMS I64 用 HP MACRO	5-27
5.19.2	OpenVMS Alpha システム用の HP MACRO	5-28
5.19.3	/TIE 修飾子のデフォルトは Alpha と I64 で異なる	5-29
5.19.4	/OPTIMIZE=VAXREGS 修飾子は I64 ではサポートされない	5-29
5.19.5	浮動小数点数のゼロ除算エラーが検出されない (I64 のみ)	5-29
5.20	Hypersort ユーティリティ	5-29
5.20.1	弊社への問題の報告	5-30
5.20.2	ラージ・ファイルの制限事項	5-30

5.20.3	Hypersort と VFC ファイルの制限事項	5-30
5.20.4	/FORMAT=RECORD_SIZE の制限事項	5-30
5.20.5	Hypersort と検索リスト, および論理名の使用	5-31
5.20.6	作業ファイルの空き領域不足	5-31
5.20.7	入力アスタリスク (*) の制限事項	5-31
5.20.8	最適化されたワーキング・セット・エクステントとページ・ファイル・クォータの設定	5-31
5.21	Intel®アセンブラ (I64 のみ)	5-32
5.22	Librarian ユーティリティ	5-32
5.22.1	data-reduced ELF オブジェクト・ライブラリとのリンクは推奨できない (I64 のみ)	5-32
5.22.2	I64 ライブラリへの.STB ファイルの挿入または置き換えの失敗 (I64 のみ)	5-32
5.22.3	プロセス・クォータが低すぎると Librarian がエラーを通知しない問題	5-33
5.23	OpenVMS 用の Analyze ユーティリティ (I64 のみ)	5-33
5.23.1	I64 Analyze イメージの/SELECT でのプロセス・リソースの解放についての修正	5-34
5.23.2	デバッグ行情報の選択的出力の修正	5-34
5.24	Command Definition ユーティリティ (I64 のみ)	5-34
5.24.1	I64 イメージのレコード属性の修正	5-34
5.25	OpenVMS Alpha 用 Linker ユーティリティ	5-35
5.25.1	多数のファイルを指定した場合に Linker がハングアップしたように見える	5-35
5.25.2	ライブラリ・チェックにおける Linker のデフォルト動作の変更	5-35
5.25.3	スタックのエLEMENT数は最大 25 に制限	5-36
5.26	OpenVMS I64 用 Linker ユーティリティ	5-36
5.26.1	正しくないイメージ間デバッグ・フィックスアップをリンクがデバッグ・シンボル・ファイルに書き込む	5-36
5.26.2	OpenVMS I64 のオブジェクト・モジュールとイメージ・ファイルの情報が現在利用できない	5-37
5.26.3	/SELECTIVE_SEARCH がトランスファー・アドレスを誤って無視することがある	5-37
5.26.4	I64 リンカと Alpha リンカの違い	5-39
5.26.5	LINK_ORDER セクション・ヘッダ・フラグはサポートされていない	5-39
5.26.6	data-reduced ELF オブジェクト・ライブラリとのリンクは推奨できない	5-40
5.26.7	初期化されたオーバーレイ・プログラム・セクションの取り扱いについての誤りの修正	5-40
5.26.8	リンクの修飾子/EXPORT_SYMBOL_VECTOR と/PUBLISH_GLOBAL_SYMBOLS の削除	5-40
5.26.9	オプションでの長いシンボル名のサポート	5-41
5.26.10	リンクが作成したコード・スタブのメモリの使用方法の改善	5-41
5.26.11	コンパイラでのデマングル化されたシンボル名のサポート	5-41
5.26.12	最大セクション数	5-42
5.26.13	マップ・ファイル中の共用可能イメージの作成日が正しくない	5-42
5.27	LTDRIVER: CANCEL SELECTIVE の制限事項	5-42
5.28	Mail ユーティリティ: 呼び出し可能メールのスレッドの制限事項	5-43
5.29	OpenVMS Debugger	5-43

5.29.1	デバッガのコマンド行ヘルプに STEP/SEMANTIC_EVENT に関する情報がない	5-43
5.29.2	このリリースで修正された状況と問題点 (I64 のみ)	5-43
5.29.3	全般的な問題点と回避方法 (I64 のみ)	5-44
5.29.4	Basic 言語での問題 (I64 のみ)	5-45
5.29.5	C++ 言語での問題 (I64 のみ)	5-45
5.29.6	COBOL 言語での問題 (I64 のみ)	5-45
5.29.7	Fortran 言語での問題 (I64 のみ)	5-45
5.29.8	Pascal 言語での問題 (I64 のみ)	5-46
5.29.9	SET SCOPE コマンド: 動作の変更	5-46
5.29.10	SHOW IMAGE コマンドの変更	5-46
5.29.11	クライアント/サーバ・インタフェース: 以前のバージョンはサポートされない (Alpha のみ)	5-46
5.30	OpenVMS のシステム・ダンプ・アナライザ (SDA)	5-47
5.30.1	CLUE コマンドは OpenVMS I64 に移植されていない	5-47
5.31	OpenVMS I64 Version 8.2 に含まれない PL/I ライブラリ	5-47
5.32	POSIX スレッド・ライブラリ	5-48
5.32.1	例外処理中のスタック・オーバフロー (I64 のみ)	5-48
5.32.2	I64 システムでの THREADCP コマンドの動作	5-49
5.32.3	浮動小数点のコンパイルと例外 (I64 のみ)	5-49
5.32.4	C 言語コンパイル・ヘッダ・ファイルの変更	5-49
5.32.5	新しい優先順位調整アルゴリズム	5-50
5.32.6	プロセス・ダンプ	5-51
5.32.7	動的 CPU 構成の変更	5-51
5.32.8	デバッガ計測機能は動作しない	5-52
5.33	RTL ライブラリ (LIB\$)	5-52
5.33.1	RTL ライブラリ (LIB\$) のヘルプ	5-52
5.33.2	RTL Library (LIB\$): 呼び出し標準ルーチン (I64 のみ)	5-52
5.34	Screen Management (SMG\$) のドキュメント	5-53
5.35	SORT32 ユーティリティ	5-54
5.35.1	DFS サービス・ディスクでの CONVERT の問題	5-54
5.35.2	一時作業ファイルが削除されないことがある	5-54
5.35.3	複合条件のある SORT/SPECIFICATION: 要件	5-55
5.35.4	可変長レコードでの性能の問題	5-55
5.35.5	作業ファイル・ディレクトリの制約事項	5-55
5.36	システム・サービス	5-55
5.37	タイマ・キュー・エントリ (TQE)	5-56
5.38	Watchpoint ユーティリティ (I64 のみ)	5-56
5.39	プログラム全体の浮動小数点モード (I64 のみ)	5-56
5.40	HP OpenVMS Debugger の Heap Analyzer の問題点と回避策 (I64 のみ)	5-57
6	ハードウェアに関する注意事項	
6.1	MP コンソールと BMC コンソールの制約事項 (I64 のみ)	6-1
6.1.1	入力デバイス, 出力デバイス, およびエラー・デバイスの制約事項	6-2
6.1.2	削除キーへの Ctrl/H の再マッピング	6-2
6.2	AlphaServer 2100	6-2
6.2.1	コンソール表示	6-2

6.2.2	SCSI コントローラの制限事項	6-3
6.3	AlphaServer 8200/8400: FRU テーブル・エラー	6-4
6.4	AlphaServer ES47/ES80/GS1280 システム	6-4
6.4.1	ES47/ES80/GS1280 ソフト・パーティションでの INIT コンソール・コマンドの使用	6-4
6.4.2	RAD のサポート	6-4
6.4.3	ライセンス要件	6-5
6.4.4	STOP/CPU およびシャットダウン動作	6-5
6.4.5	MBM での時刻の設定	6-5
6.5	AlphaServer GS シリーズ・システム	6-5
6.5.1	AlphaServer GS80/160/320 システム: デバイスの制限事項	6-6
6.5.2	OpenVMS Galaxy ライセンスの実行	6-6
6.5.3	ライセンスのインストール	6-6
6.5.4	AlphaServer GS60/GS60E/GS140 複数 I/O ポート・モジュール構成の制限事項	6-8
6.6	AlphaStation 200/400: ISA_CONFIG.DAT の変更が必要	6-9
6.7	AlphaStation 255: PCI 構成の制限事項	6-9
6.8	ATI RADEON 7000 グラフィック (I64 のみ)	6-10
6.8.1	I64 のグラフィック・サポート	6-10
6.8.2	RADEON 7000 ではハードウェア・アクセラレーション 3D グラフィックはサポートされていない	6-10
6.9	ATI RADEON 7500 グラフィック	6-11
6.9.1	リソースの要件	6-11
6.9.2	DECW\$OPENGLSHR_RADEON.EXE を DECW\$MESA3DSHR_RADEON.EXE にリネーム	6-12
6.9.3	サポートの制限事項	6-12
6.9.4	高リフレッシュ・レートでの画像への影響	6-12
6.9.5	OpenGL は IEEE 演算のみをサポート	6-13
6.9.6	グラフィック・コンソール (OPA0) に出力する際に DECwindows サーバがハングアップ	6-13
6.9.7	モニタは初期化時に接続されていなければならない	6-13
6.9.8	ブート・リセットの推奨 (Alpha のみ)	6-14
6.9.9	オーバーレイ・プレーンはサポートされない	6-14
6.9.10	単一カラーマップのみサポート	6-14
6.9.11	すべてのウィンドウで同じビット深度	6-14
6.9.12	読み取り/書き込みカラー・マップのピクセル深度	6-15
6.9.13	backing store と save unders は 3D ウィンドウをサポートしていない	6-15
6.9.14	スレッドの制限事項	6-15
6.9.15	シングル・バッファ・ビジュアルがサポートされていない	6-15
6.9.16	カラー・インデックス・モードでの 3D がサポートされていない	6-16
6.9.17	タイマ・メカニズム	6-16
6.10	DECwindows X11 ディスプレイ・サーバ (Alpha のみ)	6-17
6.10.1	S3 マルチヘッド・グラフィック	6-17
6.10.2	OpenVMS V8.3 の DECwindows では複数の USB キーボードとマウスがサポートされない	6-17
6.11	DIGITAL Modular Computing Components (DMCC)	6-17
6.11.1	Alpha 5/366 および 5/433 PICMG SBC の制限事項	6-17
6.11.2	SRM コンソールの更新	6-18

6.12	Digital Personal Workstation: OpenVMS V7.3-1 およびそれ以降のブート	6-18
6.13	I/O 負荷が重い場合, LUN が多数あるデュアル・コントローラ HSGnnに障害が発生することがある	6-19
6.14	Open3D グラフィックのライセンス方式の変更	6-19
6.15	OpenVMS 用の PowerStorm 300/350 PCI グラフィック・サポート	6-20
6.16	RFnn DSSI ディスク・デバイスとコントローラ・メモリのエラー	6-21
6.17	RZnn ディスク・ドライブの考慮事項	6-23
6.17.1	RZ25M と RZ26N ディスク・ドライブ: 推奨事項	6-24
6.17.2	RZ26N および RZ28M ディスク: 推奨ファームウェア・サポート	6-24
6.17.3	RZ26L および RZ28 ディスク: マルチホストで使用するために必要なファームウェア	6-24
6.17.3.1	ファームウェア・リビジョン・レベル 442 の必要条件	6-25
6.17.3.2	ファームウェア・リビジョン・レベル 442 のインストール手順	6-25
6.18	sx1000 Integrity Superdome	6-25
6.19	ZLX グラフィック・ボードのサポート	6-26
6.20	OpenVMS デバイス・ドライバの再コンパイルと再リンク	6-26
6.20.1	Alpha および VAX の SCSI デバイス・ドライバ	6-26
6.20.2	OpenVMS Alpha デバイス・ドライバ	6-27
6.21	MON バージョンのデバイス・ドライバの処理	6-27
6.22	スレッド単位のセキュリティが Alpha デバイス・ドライバに与える影響	6-27
6.23	OpenVMS Alpha ドライバのデバイス IPL の設定	6-28
6.24	CRCTX ルーチンの機能の強化	6-28
6.25	アダプタについての注意事項	6-29
6.25.1	Fibre Channel の EFI ドライバとファームウェアの要件	6-29
6.25.2	複数の Fibre Channel ブート・エントリを使用したブート	6-30

A リタイア製品情報

A.1	Compaq Open3D レイヤー・プロダクトは Version 8.2 ではサポートされない	A-1
A.2	Open3D グラフィックのライセンス方式の変更	A-2
A.3	DECamds は OpenVMS Version 8.2 ではサポートされない	A-2
A.4	DECevent はサポートされない	A-2
A.5	2004 年 12 月に DECwrite のサービス期間が終了	A-3
A.6	Error Log Report Formatter (ERF) のサポート終了	A-3
A.7	ISA_CONFIG.DAT のサポートの将来のリリースでの中止	A-4
A.8	POSIX 1003.4a Draft 4 インタフェースのサポート中止	A-4
A.9	NetBeans Version 3.6 のサポートは OpenVMS Version 8.3 で終了	A-4
A.10	アーカイブ扱いのマニュアル	A-5

B インターロックされたメモリ命令の使用 (Alpha のみ)

B.1	必要なコード・チェック	B-1
B.2	コード分析ツール (SRM_CHECK) の使用	B-2
B.3	規則に準拠していないコードの特徴	B-3
B.4	コーディングの必要条件	B-4
B.5	コンパイラのバージョン	B-6
B.6	ALONONPAGED_INLINE または LAL_REMOVE_FIRST によるコードの再コンパイル	B-7

索引

表

1-1	サポートされている DECwindows Motif のバージョン	1-5
1-2	エントリ・クラスの Integrity サーバ用のファームウェア・バージョン	1-7
4-1	クラスタ互換性のために必要なパッチ・キット	4-12
4-2	クラスタの互換性のために必要なパッチ・キット	4-13
4-3	Galaxy の定義	4-18
4-4	TFF 文字フォールバック・テーブル	4-23
4-5	I64 ISD_Labels インデックス	4-46
6-1	デバイス記述ブロックの変更点	6-9
6-2	サポートされるマイクロコードのレベル	6-22
6-3	DSSI ディスク・デバイスのマイクロコードを更新するコマンド	6-23
6-4	リビジョン・レベル 442 ファームウェアの互換性	6-25
B-1	OpenVMS コンパイラのバージョン	B-6

対象読者

本書は、HP OpenVMS Alpha または HP OpenVMS I64 Integrity サーバ Version 8.3 オペレーティング・システムのすべてのユーザを対象にしています。OpenVMS Version 8.3 のインストール、アップグレード、あるいはご使用前にお読みください。

本書の構成

本書の構成は以下のとおりです。

- 第1章は、OpenVMS Alpha オペレーティング・システムのアップグレードおよびインストールと、OpenVMS I64 のインストールに関する注意事項について説明します。
- 第2章は、OpenVMS の関連製品のインストールおよびサポート情報について説明します。
- 第3章は、OpenVMS オペレーティング・システムの一般的な使用に関する注意事項について説明します。
- 第4章は、特に OpenVMS のシステム管理に関する注意事項について説明します。
- 第5章は、OpenVMS システムでのプログラミングに関する注意事項について説明します。コンパイラ、リンカ、およびランタイム・ライブラリ・ルーチンを使用する際の注意事項も含まれています。
- 第6章は、OpenVMS オペレーティング・システムが実行されるハードウェアについての情報と、OpenVMS のデバイス・サポートについての情報を示します。
- 付録 A は、本リリースではサポートされなくなった OpenVMS 製品、または廃止予定の OpenVMS 製品についての情報を示します。
- 付録 B は、プロセッサ Alpha 21264 (EV6) を使用する際に非常に重要な、インターロックされたメモリ命令の正しい使用方法について説明します。

可能な限り、機能または製品名ごとに注意事項が分類されています。

本書に記載されている注意事項には、今回のリリースから適用されるものと、以前のリリースから引き続き適用されるものが含まれています。各注意事項が最初に適用されたバージョンが、項目タイトルとして (V8.2のように) 示されています。または、古

い注意事項が更新された場合はそのバージョンが示されています (たとえば , V8.2 の注意事項で , Version 8.3 で改訂されたものには , V8.3という見出しが付きます)。

次のような場合も , 以前のバージョンの注意事項が記載されます。

- その情報が OpenVMS ドキュメント・セットのどのマニュアルにも記載されておらず , その注意事項の内容がまだ適用される場合。
- 注意事項がマルチ・バージョンの OpenVMS Cluster システムに適用される可能性がある場合。

OpenVMS ソフトウェアのインストールおよびアップグレードに関する注意事項

この章では、OpenVMS Version 8.3 のインストールやアップグレードに必要な情報について説明します。Alpha および I64 システムのユーザに共通のトピックを最初に説明し、その後の節で、特定のプラットフォームのユーザ向けに説明します。

OpenVMS Version 8.3 のインストールまたはアップグレードの際には次のマニュアルをすべてお読みください。

- 『HP OpenVMS V8.3 リリース・ノート[翻訳版]』(本書)
- 『HP OpenVMS V8.3 新機能説明書』
- 『HP OpenVMS V8.3 インストレーション・ガイド[翻訳版]』

ハードウェアに関する注意事項については第 6 章、関連製品については第 2 章を参照してください。

1.1 弊社ソフトウェアのテクニカル・サポート方針

弊社では、OpenVMS オペレーティング・システム・ソフトウェアの最新版 (現在出荷されているバージョン) と、直前のバージョンの製品について、ソフトウェアのテクニカル・サポートを行います。各バージョンは、リリース日より 24 カ月間、または次の次のバージョンがリリースされるまでの期間 (いずれか長い方)、サポートされます。「バージョン」は、新しい機能と拡張を含むリリースと定義されます。サポート方針としては、パッチ・キットやメンテナンス専用のリリースは、この「バージョン」の定義に当てはまりません。

これらのガイドラインに基づいて、OpenVMS のバージョンごとに、OpenVMS オペレーティング・システム・ソフトウェアの、現行バージョン・レベルのサポート (標準サポート (SS)) と旧バージョンのサポート (PVS) が行われます。OpenVMS Alpha と OpenVMS VAX の最近のバージョンに対する現在のサポート・レベルは、次の Web ページから入手してください。

http://h71000.www7.hp.com/openvms/openvms_supportchart.html

オペレーティング・システムのサポート方針は、すべての OpenVMS メジャー・リリース、新機能リリース、および拡張リリースに適用されます。各リリースの定義は、次のとおりです。

- メジャー・リリースには、根本的な新機能が含まれます。バージョン番号の整数部分に、次に大きい数字が使用されます (例: 6.2-1H1 から 7.0 へ)。

アプリケーションへの影響: OpenVMS の内部インタフェースが変更されています。大半のアプリケーションではバイナリ互換性が維持されますが、独立系ソフトウェア・ベンダ (ISV) は、新しいバージョンでテストを行い、場合によっては新しいアプリケーション・キットをリリースする必要があります。アプリケーション・パートナーは、オペレーティング・システムの新しい機能を活用するために、新しいアプリケーション・キットをリリースすることもできます。

- 新機能リリースには、新機能、拡張機能、およびメンテナンス上の更新が含まれます。バージョン番号の小数第 1 位に、次に大きい数字が使用されます (例: 7.2 から 7.3 へ)。

アプリケーションへの影響: このリリースでは、公開されたアプリケーション・プログラミング・インタフェース (API) は廃止されません。ただし、重要な新機能の追加や性能の向上により、OpenVMS の内部インタフェースが変更されていることがあります。マニュアルに記載されている API を使用するアプリケーションのほとんど (95%) については、新しいアプリケーション・キットは不要です。デバイス・ドライバとカーネル・レベルのアプリケーション (標準で規定されていない API または説明されていない API を使用するアプリケーション) については、適合性テストが必要になることがあります。

- 拡張リリースには、既存の機能に対する拡張機能やメンテナンス上の更新が含まれます。バージョン番号のダッシュ (-) 以降に、次に大きい数字が使用されます (例: 7.3-2)。

アプリケーションへの影響: このリリースには、新しいハードウェア・サポート、ソフトウェアの拡張機能、およびメンテナンス上の更新が含まれていますが、この変更は公開された API を使用するアプリケーションに影響を及ぼしません。ISV では新しいリリースをテストしたり、新しいアプリケーション・キットを作成したりする必要はありません。

- ハードウェア・リリースは、特定のハードウェア・サポートを含む次のリリースが出されてから 12 カ月間、現行バージョン・レベルでサポートされます。ハードウェア・リリースは、新しいハードウェアを購入いただいた場合にのみ送付されます。サービス契約だけの場合は、ご利用いただけません。

OpenVMS の次の主な製品は、同時に出荷されたオペレーティング・システムのバージョンと同じ期間だけ同じレベル (標準サポートまたは旧バージョンのサポート) でサポートされます。

- HP Advanced Server for OpenVMS
- HP DECnet (フェーズ IV)
- HP DECnet-Plus for OpenVMS
- HP OpenVMS Cluster Client Software
- HP OpenVMS Cluster Software for OpenVMS

- HP PathWorks または HP PATHWORKS for OpenVMS
- HP RMS Journaling for OpenVMS
- HP TCP/IP Services for OpenVMS
- HP Volume Shadowing for OpenVMS
- HP DECram for OpenVMS

これらの製品のサポート・サービスを依頼するには、個別のサポート契約が必要です。オペレーティング・システムのサポート契約には含まれていません。

1.2 アプリケーションの互換性

OpenVMS では、公開された API は、どのリリースでも一貫してサポートされます。通常、公開された API を使用するアプリケーションであれば、OpenVMS の新しいリリースをサポートするために変更が必要になることはありません。廃止された API は、マニュアルから削除されますが、OpenVMS では引き続き API として使用できます。

1.3 修正キットの入手方法

弊社製品の修正キットは、HP IT リソース・センタ (ITRC) からオンラインで入手できます。ITRC パッチ・ダウンロード・サイトを使用するには、ユーザ登録とログインが必要です。すべてのユーザが登録可能で、サービス契約は不要です。次の URL で、登録とログインができます。

<http://www2.itrc.hp.com/service/patch/mainPage.do>

また、FTP を使用して、次の場所からパッチを入手することもできます。

ftp://ftp.itrc.hp.com/openvms_patches

1.4 ネットワーク・オプション

V8.3

OpenVMS では、使用するネットワーク・プロトコルを柔軟に選択できます。DECnet が必要な場合も、TCP/IP が必要な場合も、OpenVMS ではネットワークにとって最適なプロトコルあるいは複数のプロトコルの組み合わせを選択できます。OpenVMS では、弊社のネットワーク製品と他社製ネットワーク製品のどちらも使用できます。

OpenVMS Version 8.3 のインストール時に、サポートされている次の HP ネットワーク・ソフトウェアをインストールすることができます。

- HP DECnet-Plus Version 8.3 for OpenVMS または HP DECnet フェーズ IV Version 8.3 for OpenVMS (2 つの DECnet 製品を同時に実行することはできません)

DECnet-Plus には、DECnet フェーズ IV 製品のすべての機能に加えて、TCP/IP または OSI プロトコルを介して DECnet を実行する機能も含まれています。

または、OpenVMS をインストールした後で、OpenVMS Version 8.3 で動作する他社製ネットワーク製品をインストールすることもできます。

インストールの後で HP ネットワーキング・ソフトウェアを構成したり管理したりする方法については、TCP/IP、DECnet-Plus、または DECnet の各マニュアルを参照してください。

1.5 古いバージョンの OpenVMS と互換性のないディスク

V8.3

OpenVMS Version 8.3 のインストール手順では、ターゲット・ディスクをボリューム拡張付き (INITIALIZE/LIMIT) で初期化します。これにより、Version 7.2 よりも前の OpenVMS とはディスクの互換性がなくなります。多くの場合、特に問題は発生しません。ただし、この新しいディスクを Version 7.2 よりも前の OpenVMS にマウントする場合は、そのバージョンのオペレーティング・システムと互換性を持つように設定しなければなりません。設定手順の詳細は、『HP OpenVMS V8.3 インストール・ガイド[翻訳版]』を参照してください。

これらの手順を実行すると、新しいシステム・ディスクの最小割り当てサイズ (/CLUSTER_SIZE で定義) は、以前より大きくなることがあります。その結果、小さなファイルが必要以上のスペースを専有するようになります。そのため、この手順は、Version 7.2 より前の OpenVMS にマウントしなければならないシステム・ディスクに対してのみ実行してください。

注意

ODS-5 ディスクも、Version 7.2 より前の OpenVMS とは互換性がありません。

1.6 HP DECwindows Motif for OpenVMS

V8.3

次の表に、HP OpenVMS 8.3 オペレーティング・システムでサポートされている DECwindows Motif のバージョンを示します。

表 1-1 サポートされている DECwindows Motif のバージョン

OpenVMS のバージョン	DECwindows Motif のバージョン
OpenVMS I64 8.3	DECwindows Motif for OpenVMS I64 V1.6
OpenVMS Alpha 8.3	DECwindows Motif for OpenVMS Alpha V1.6

DECwindows Motif ソフトウェアは、特定のバージョンの OpenVMS サーバおよびデバイス・ドライバのイメージを必要とします。表 1-1 に示すように、オペレーティング・システム環境に適したバージョンの DECwindows Motif をインストールまたはアップグレードするようにしてください。

以前のバージョンの DECwindows Motif のサポート情報については、『日本語 HP DECwindows Motif for OpenVMS リリース・ノート』を参照してください。

DECwindows Motif ソフトウェアのインストールについての詳細は、『日本語 HP DECwindows Motif for OpenVMS インストレーション・ガイド』を参照してください。

1.7 Integrity サーバ用の OpenVMS ユーザ向けの注意事項

以降の説明は、主に Integrity サーバ用の OpenVMS システムのユーザ向けです。

1.7.1 Version 8.3 での A6825A Gigabit Ethernet コントローラと RX4640 の制限

V8.3

HP Integrity サーバの HP sx1000 チップセットは、HP Integrity rx7620 サーバ、HP Integrity 8620 サーバ、および HP Integrity Superdome サーバの、CPU、メモリ、および I/O サブシステムを制御します。sx1000 チップセット・アーキテクチャでは、セル・コントローラが 4 個の CPU チップと結合されて、コンピューティング・セルを構成しています。セル・コントローラ・チップには、I/O デバイスやオフ・セル・メモリへのパスも用意されています。

rx7620 サーバ、rx8620 サーバ、および Superdome サーバでは、さまざまな数の sx1000 チップセット・セルを提供します。rx7620 は最大 2 セル (8 CPU)、rx8620 は最大 4 セル (16 CPU)、また Superdome は最大 16 セル (64 CPU) を提供します。

OpenVMS I64 Version 8.3 には、2 つの主要なストレージ・インターコネクトがあります。

- SCSI ストレージ・タイプは U320 です。これは一部の Integrity Server の Core I/O で使用しており、また A7173A U320 SCSI アダプタでも使用しています。外部の SCSI ストレージとの接続については、ストレージ・シェルフとして DS2100 と MSA30 をサポートしています。
- 外部の Fibre Channel ストレージとの接続には、dual-port 2GB Fibre Channel Universal PCI-X アダプタ (A6826A) を使用します。このアダプタは、OpenVMS がサポートしている SAN ベースの Fibre Channel ストレージのいずれとも接続できます。

以前の評価版キットや、フィールド・テスト版キットをご使用のユーザは、下記の重要な考慮事項にご注意ください。

- SCSI アダプタのうち、U160 アダプタ (A6829A) は、OpenVMS I64 Version 8.3 では公式にサポートしていませんし、2005 年にサポート期間が終わりました。ただし、システムを現在の構成のままにしておく限り、このアダプタを既存のハードウェア用に使用することは可能です。アダプタを追加したり、別のサーバに移行する場合は、U320 タイプの SCSI アダプタが必要になります。
- Fibre Channel については、AB232A または KGPSA-EA FC アダプタをご使用の場合があります。これらのアダプタは、OpenVMS I64 Version 8.3 ではサポートされませんので、Version 8.2 上で製品版のアプリケーションを実行する前に、A6826A FC アダプタにアップグレードする必要があります。

1.7.2 rx7620 および rx8620 での U160 SCSI のサポート

V8.3

rx7620 と rx8620 には、コア I/O としてシステムに U160 (SCSI) が内蔵されています。SCSI ディスクのラックへの内部接続 (システム・ボックスの前面にあります) が OpenVMS によってサポートされます。内部ボックスには 2 つの外部ポートもありますが、これらのポートをケーブルで SCSI ラックに接続することはサポートされていません。

1.7.3 Integrity Server での System Event Log (SEL) のクリア

V8.3

HP Integrity サーバは、システム・コンソールのストレージ内に System Event Log (SEL) を保存しており、OpenVMS I64 システムは自動的に SEL の内容を OpenVMS エラー・ログに転送します。コンソールから操作しているとき、正常なブート処理中に BMC (Baseboard Management Controller) SEL が満杯である旨のメッセージが表示されることがあります。BMC SEL が満杯の場合でも、プロンプトに

従って操作すれば、OpenVMS は SEL の内容を処理し、運用を継続できます。手動で SEL をクリアしたい場合は、EFI シェルのプロンプトで、次のコマンドを入力してください。

```
Shell> clearlogs SEL
```

このコマンドにより、SEL の内容がクリアされます。このコマンドは、最新のシステム・ファームウェア・バージョンで使用できます。

ご使用の Integrity サーバに MP (Management Processor) が構成されていて、MP コンソールと接続しているときに BMC イベント・ログの警告メッセージが表示された場合は、MP を使用して BMC イベント・ログをクリアすることができます。Ctrl/B を押して、MP>プロンプトを表示させます。MP>プロンプトで(メイン・メニューから) SL を入力し、C オプションでログをクリアします。

最新のシステム・ファームウェアをロードして使用することをお勧めします。システム・ファームウェアのアップデートの詳細については『HP OpenVMS V8.3 インストレーション・ガイド[翻訳版]』を参照してください。

1.7.4 Integrity サーバ用のファームウェア

V8.3

OpenVMS I64 Version 8.3 は、最新のファームウェアがインストールされた各 Integrity サーバでテストされています。次の表に、推奨するファームウェア・バージョンの一覧を示します。

表 1-2 エントリ・クラスの Integrity サーバ用のファームウェア・バージョン

システム	システム・ファームウェア	BMC ファームウェア	MP ファームウェア	DHCP ファームウェア
rx1600	2.18	3.48	E.03.15	適用外
rx1620	2.18	3.48	E.03.15	適用外
rx2600	2.31	1.53	E.03.15	適用外
rx2620	3.17	3.47	E.03.15	適用外
rx4640	3.17	3.49	E.03.15	1.10

サーバのファームウェア・バージョンを確認するには、次の例のように、EFI (Extensible Firmware Interface) のコマンド INFO FW を使用します。(EFI のアクセス方法および使用方法については、『HP OpenVMS V8.3 インストレーション・ガイド[翻訳版]』を参照してください)。

OpenVMS ソフトウェアのインストールおよびアップグレードに関する注意事項

1.7 Integrity サーバ用の OpenVMS ユーザ向けの注意事項

```
Shell> INFO FW
FIRMWARE INFORMATION

Firmware Revision: 2.13 [4412]          1
PAL_A Revision: 7.31/5.37
PAL_B Revision: 5.65
HI Revision: 1.02

SAL Spec Revision: 3.01
SAL_A Revision: 2.00
SAL_B Revision: 2.13

EFI Spec Revision: 1.10
EFI Intel Drop Revision: 14.61
EFI Build Revision: 2.10

POSSE Revision: 0.10

ACPI Revision: 7.00

BMC Revision: 2.35                      2
IPMI Revision: 1.00
SMBIOS Revision: 2.3.2a
Management Processor Revision: E.02.29  3
```

- 1 システムのファームウェア・リビジョンは、2.13 です。
- 2 BMC のファームウェア・リビジョンは、2.35 です。
- 3 MP のファームウェア・リビジョンは、E.02.29 です。

HP Integrity rx4640 サーバには、アップグレード可能ファームウェアを備えた、Dual Hot Plug Controller (DHPC) ハードウェアが含まれています。DHPC ファームウェアの現在のバージョンを確認するには、次の例のように、EFI の INFO CHIPREV コマンドを使用します。ホット・プラグ・コントローラのバージョンが表示されます。0100 という表示は、Version 1.0 を意味します。0110 という表示は、Version 1.1 を意味します。

```
Shell> INFO CHIPREV
CHIP REVISION INFORMATION
```


OpenVMS ソフトウェアのインストールおよびアップグレードに関する注意事項
1.7 Integrity サーバ用の OpenVMS ユーザ向けの注意事項

Chip Type	Logical ID	Device ID	Chip Revision
-----	-----	-----	-----
Memory Controller	0	122b	0023
Root Bridge	0	1229	0023
Host Bridge	0000	122e	0032
Host Bridge	0001	122e	0032
Host Bridge	0002	122e	0032
Host Bridge	0004	122e	0032
HotPlug Controller	0	0	0110
Host Bridge	0005	122e	0032
HotPlug Controller	0	0	0110
Host Bridge	0006	122e	0032
Other Bridge	0	0	0002
Other Bridge	0	0	0008
Baseboard MC	0	0	0235

ファームウェアのアップグレード手順については、『HP OpenVMS V8.3 インストール・ガイド[翻訳版]』を参照してください。

1.7.5 インストール DVD からのブート

V8.2

サポートする最小メモリ構成 (512MB) の I64 システムで、インストール DVD からブートすると次のようなメッセージが表示されます。

```
***** XFC-W-MemmgInit Misconfigure Detected *****
XFC-E-MemMisconfigure MPW_HILIM + FREEGOAL > Physical Memory and no reserved memory for XFC
XFC-I-RECONFIG Setting MPW$GL_HILIM to no more than 25% of physical memory
XFC-I-RECONFIG Setting FREEGOAL to no more than 10% of physical memory
***** XFC-W-MemMisconfigure AUTOGEN should be run to correct configuration *****
```

このメッセージは、システム・キャッシュ (XFC) の初期設定で、インストール中のキャッシュ処理を有効にするために、SYSGEN パラメータの MPW_HILIM と FREEGOAL を正常に調整したことを意味しています。インストールは続行できます。

1.7.6 HP DECwindows Motif に関する注意事項

HP DECwindows Motif に関する以降の注意事項は、OpenVMS I64 ユーザを対象としています。

1.7.6.1 キーボードのサポート

V8.2

HP DECwindows Motif for OpenVMS I64 システムでサポートされているキーボード・モデルは、LK463 (I64 用 AB552A) キーボードだけです。他のタイプのキーボードも OpenVMS I64 環境で動作する可能性があります、弊社では現在それらはサポートしていません。

1.7.6.2 サーバ起動前の周辺デバイスの接続

V8.2

システムを DECwindows X ディスプレイ・サーバとして適切に構成するには、起動前に、次のすべての周辺コンポーネントが接続されていなければなりません。

- モニタ
- USB マウス
- USB キーボード

接続されていない場合、サーバ・システムは初期化処理を正しく完了できないことがあります。たとえば、入力デバイス (マウスおよびキーボード) なしでサーバ・システムがスタートアップすると、ブランク・スクリーンになることがあります。

この問題を解決するには、すべての周辺コンポーネントを切り離してから再接続してください。

1.7.6.3 スタートアップ中に表示されるカウントダウン・メッセージ

V8.2

クライアント・オンリー・モード (サーバ無しの構成) で DECwindows Motif を実行すると、起動中に次のようなメッセージが表示されることがあります。

```
Waiting for mouse...  
Waiting for keyboard...
```

これらのメッセージは、デバイス・ポーリングの実行中だという情報を示しているにすぎません。15 秒経過すると、これらのメッセージは消えます。

このメッセージが表示されないようにするには、システムを起動する前に、入力デバイス (USB マウスおよび USB キーボード) を接続してください。

1.8 OpenVMS Alpha 向けの注意事項

以降のリリース・ノートは、OpenVMS Alpha ユーザ用です。

1.8.1 OpenVMS Alpha Version 8.3 のファームウェア

V8.3

OpenVMS Alpha Version 8.3 は、Alpha Systems Firmware CD Version 6.8 に収められているプラットフォーム固有ファームウェアでテストされています。ファームウェア CD に含まれなくなった古いプラットフォームについては、最新のファームウェア・バージョンで OpenVMS Alpha Version 8.2 がテストされています。OpenVMS をアップグレードする前に、最新のファームウェアにアップグレードすることをお勧めします。

OpenVMS Alpha Version 8.3 キットには、Alpha Systems Firmware CD-ROM とリリース・ノートが同梱されています。ファームウェアをインストールする前に、ファームウェアのリリース・ノートに目を通してください。

Version 6.8 と、最新のファームウェアの情報については、次の Web サイトを参照してください (URL の大文字と小文字は区別されます)。

<http://ftp.digital.com/pub/DEC/Alpha/firmware/>

1.8.2 アップグレード・パス

V8.3

OpenVMS Alpha Version 8.3 では、下記のバージョンからのアップグレード・インストールをサポートしています。

Alpha の場合:

Version 7.3-2 から Version 8.3 へ
Version 8.2 から Version 8.3 へ

I64 の場合:

Version 8.2 から Version 8.3 へ
Version 8.2-1 から Version 8.3 へ

現在、OpenVMS Alpha Version 6.2x ~ 7.3 を実行している場合は、まず Version 7.3-2 にアップグレードした後、Version 8.3 にアップグレードする必要があります。なお、OpenVMS Alpha Version 7.3-1 システムの標準サポートは、2005 年 3 月 31 日で終了しました。これ以降、OpenVMS Alpha V7.3-1 システムは、PVS (Prior Version Support) の対象とはなりません。OpenVMS オペレーティング・システムのサポートの詳細については、次の Web サイトの表を参照してください。

http://h71000.www7.hp.com/openvms/openvms_supportchart.html

これ以外のサポートされていないバージョンの OpenVMS を実行している場合は、以前のバージョンのドキュメントに記載されているアップグレード・パスに沿って、複数回のアップグレード作業を行わなければなりません。

クラスタの同時アップグレード

クラスタを一度にアップグレードする場合には、クラスタ全体をシャットダウンして、各システム・ディスクをアップグレードします。各コンピュータでアップグレードおよびリブートが完了するまでクラスタを使用することはできません。各コンピュータがリブートされると、それぞれのコンピュータではアップグレード後のオペレーティング・システムが実行されます。

クラスタのローリング・アップグレード

クラスタのローリング・アップグレードを実行する場合は、各システム・ディスクを個別にアップグレードします。この場合、新旧バージョンのオペレーティング・システムを同じクラスタ内で同時に実行できます (複合バージョン・クラスタ)。ただし、この場合システム・ディスクが複数必要です。アップグレード中でないメンバ・システムは使用可能です。

OpenVMS Alpha Version 8.3 を含む混合バージョン・クラスタでサポートされる OpenVMS Alpha と OpenVMS VAX のバージョンは、次のとおりです。

Version 7.3-2 (Alpha)

Version 7.3 (VAX)

クラスタ環境でアップグレードを行う場合、ローリング・アップグレードは OpenVMS Alpha オペレーティング・システム Version 7.3-2 および 7.3-1 からのアップグレードをサポートします。クラスタ内にこの他のバージョンが存在する場合は、それらのバージョンをサポートされているバージョンにアップグレードするまで、ローリング・アップグレードは実行できません。

どのバージョンのオペレーティング・システムも、V8.2 以降のシステムとの混合クラスタをサポートするには、修正キットをインストールする必要があります。詳細は、第 4.14.1 項を参照してください。

注意

現在 1 つのクラスタでは、アーキテクチャにかかわらず、一度に 2 つのバージョンの OpenVMS の実行のみがサポートされています。1 つの OpenVMS Cluster 内でサポートされるアーキテクチャは、2 つだけです。warranted サポートが提供されているのは、OpenVMS I64 Version 8.3 との組み合わせです。詳細は、『HP OpenVMS V8.3 インストレーション・ガイド[翻訳版]』を参照してください。

OpenVMS オペレーティング・システムで warranted サポートが提供される組み合わせ、migration サポートが提供される組み合わせ、あるいは OpenVMS Alpha Version 8.3 のインストール/アップグレード手順、OpenVMS I64 Version 8.3 のイン

ストール手順などの詳細は、『HP OpenVMS V8.3 インストレーション・ガイド[翻訳版]』を参照してください。

1.9 CDSA: 他社製品の Secure Delivery 署名について OpenVMS V8.3 のリリース後に使用可能

V8.3

OpenVMS Version 8.3 に含まれる CDSA Version 2.2 では、新しい OpenVMS Secure Delivery 機能を使用して、OpenVMS Engineering 以外の組織が OpenVMS キットへの署名や検証を行うことができます。キットに署名して検証するための機能は CDSA Version 2.2 に含まれていますが、Secure Delivery 機能による他社製品の署名については現在行なわれていません。他社による署名についても、もまもなく可能になる予定です。HP Developer & Solution Partner Program の Web サイト (<http://www.hp.com/dspp>) で、このプログラムが利用可能になったことが発表されていないか確認してください。

1.10 SYS\$MANAGER:SYSTARTUP_VMS.COM から暗号化や SSL を開始する行を削除

V8.3

暗号化と SSL のためのスタートアップ・コマンド・プロシージャは、VMS\$LPBEGIN-050_STARTUP.COM プロシージャから呼び出されるようになりました。これらの製品がインストールされている以前のバージョンの OpenVMS からアップグレードする場合は、SYSTARTUP_VMS.COM を編集して SYS\$STARTUP:ENCRYPT_START.COM および SYS\$STARTUP:SSL\$STARTUP.COM の呼び出しを削除してください。これにより、これらのコマンド・プロシージャが 2 回実行されるのを防ぐことができます。

1.11 Kerberos for OpenVMS

V8.3

Kerberos を構成または起動する前に、HP TCP/IP ローカル・ホスト・データベースをチェックし、ホスト名定義が短い名前 (たとえば、node1) と、完全修飾ドメイン名 (FQDN) (たとえば、node1.hp.com) のどちらかを確認してください。

ホスト名定義が短い名前の場合、TCPIP\$CONFIG を実行して、定義を完全修飾名に変更しなければなりません。

次の例に、ホスト名が短い名前である場合について示します。

```
$ TCPIP SHOW HOST/LOCAL NODE1  
LOCAL database  
Host address    Host name  
1.2.3.4    node1
```

以下のログは、ホスト名定義を FQDN に変更する方法の例です。

```
$ @SYS$STARTUP:TCPIP$CONFIG  
TCP/IP Network Configuration Procedure  
This procedure helps you define the parameters required  
to run HP TCP/IP Services for OpenVMS on this system.  
Checking TCP/IP Services for OpenVMS configuration database files.  
HP TCP/IP Services for OpenVMS Configuration Menu  
Configuration options:  
1 - Core environment  
2 - Client components  
3 - Server components  
4 - Optional components  
5 - Shutdown HP TCP/IP Services for OpenVMS  
6 - Startup HP TCP/IP Services for OpenVMS  
7 - Run tests  
A - Configure options 1 - 4  
[E] - Exit configuration procedure  
Enter configuration option: 1  
HP TCP/IP Services for OpenVMS Core Environment Configuration Menu  
Configuration options:  
1 - Domain  
2 - Interfaces  
3 - Routing  
4 - BIND Resolver  
5 - Time Zone  
A - Configure options 1 - 5  
[E] - Exit menu  
Enter configuration option: 2  
HP TCP/IP Services for OpenVMS Interface & Address Configuration Menu  
Hostname Details: Configured=node1, Active=node1  
Configuration options:  
1 - WE0 Menu (EWA0: TwistedPair 1000mbps)  
2 - 1.2.3.4/21    node1          Configured,Active  
3 - IE0 Menu (EIA0: TwistedPair 100mbps)  
I - Information about your configuration
```

OpenVMS ソフトウェアのインストールおよびアップグレードに関する注意事項
1.11 Kerberos for OpenVMS

```
[E] - Exit menu
Enter configuration option: 2
    HP TCP/IP Services for OpenVMS Address Configuration Menu
    WE0 1.2.3.4/21 node1 Configured,Active WE0
Configuration options:
    1 - Change address
    2 - Set "node1" as the default hostname
    3 - Delete from configuration database
    4 - Remove from live system
    5 - Add standby aliases to configuration database (for failSAFE IP)
[E] - Exit menu
Enter configuration option: 1
    IPv4 Address may be entered with CIDR bits suffix.
    E.g. For a 16-bit netmask enter 10.0.1.1/16
Enter IPv4 Address [1.2.3.4/21]:
Enter hostname [node1]: node1.hp.com
Requested configuration:
    Address   : 1.2.3.4/21
    Netmask   : 255.255.248.0 (CIDR bits: 21)
    Hostname  : node1.hp.com
* Is this correct [YES]:
    "node1" is currently associated with address "1.2.3.4".
    Continuing will associate "node1.hp.com" with "1.2.3.4".
* Continue [NO]: YES
Deleted host node1 from host database
Added hostname node1.hp.com (1.2.3.4) to host database
* Update the address in the configuration database [NO]: YES
Updated address WE0:1.2.3.4 in configuration database
* Update the active address [NO]: YES
WE0: delete active inet address node1.hp.com
Updated active address to be WE0:1.2.3.4
```

次に E を 3 回入力して、TCP/IP サービスの構成メニューを終了し、DCL (\$) プロンプトに戻ります。

変更した結果を確認するには、次のコマンドを入力します。

```
$ TCPIP SHOW HOST/LOCAL NODE1
    LOCAL database
Host address   Host name
1.2.3.4  node1.hp.com
```

システム上で以前のバージョンの Kerberos を構成したことがない場合や、上記のように TCP/IP ホスト名定義を FQDN に変更した場合は、Kerberos を起動する前に Kerberos 構成プログラムを実行しなければなりません。

Kerberos を再構成するには、次のコマンドを入力します。

```
$ @SYS$STARTUP:KRB$CONFIGURE
```

構成が正しい状態になった後、次のコマンドで Kerberos を起動します。

```
$ @SYS$STARTUP:KRB$STARTUP.COM
```

詳細は、『Kerberos for OpenVMS Installation Guide and Release Notes』を参照してください。

1.12 OpenVMS オペレーティング・システムとともにインストールされる Encryption for OpenVMS

V8.3

OpenVMS をインストールまたはアップグレードするときに、Encryption for OpenVMS は独自の ENCRYPT コマンドと DECRYPT コマンドを作成します。Encryption for OpenVMS は、自動的に起動されます (SSL for OpenVMS の自動起動後)。Encryption for OpenVMS についての詳細は、『HP OpenVMS V8.3 新機能説明書』を参照してください。

注意

OpenVMS の Version 8.3 では、DCL コマンド DECRAM は、新しい DECRYPT コマンドと矛盾するため削除されます (DECRYPT は、DECram の起動に使用されている可能性のある、DECRAM の既存の定義を上書きします)。DECRAM コマンドを使用しているコマンド・プロシージャを、DCL のフォーリン・コマンド形式を使用するようにアップデートする必要があります。例を次に示します。

```
$ DECRAM == "$MDMANAGER"
```

この変更により影響があるのは、DCL コマンドでの使用だけです。これを除き、DECram 製品の使い方は、前と同じです。OpenVMS Alpha システム上に古いバージョンの DECram がある場合は、アップグレードする前にそれを削除しなければなりません。第 1.13 節を参照してください。

1.13 HP DECram V3.n: アップグレード前に削除する

V8.2

OpenVMS Alpha Version 8.2 および OpenVMS I64 Version 8.2 以降、DECram は System Integrated Product (SIP) として、OpenVMS オペレーティング・システムとともに提供されます。OpenVMS Alpha システムを OpenVMS Version 7.3-2 から Version 8.3 へアップグレードする場合は、古いバージョンの DECram を削除しなけ

ればなりません。詳細は、『HP OpenVMS V8.3 インストレーション・ガイド[翻訳版]』を参照してください。

DECram の注意事項については、第 2.12 節も参照してください。

1.14 アップグレード後に LANCP デバイス・データベースを変換する

V8.3

OpenVMS Version 7.3-2 から OpenVMS Alpha Version 8.3 へアップグレードするとき、アップグレード後の最初の LANACP の実行で LAN デバイス・データベースが自動的に Version 8.3 形式に変換されない場合は、この変換を実行する必要があります。

データベースを変換するには、次の LANACP コマンドを入力してデバイス・データベースを変換後 LANACP を停止させ、新しいデータベースで再起動できるようにします。

```
$ LANCP
LANCP> CONVERT DEVICE_DATABASE
LANCP> SET ACP/STOP
LANCP> EXIT
$ @SYS$STARTUP:LAN$STARTUP
```

1.15 DECnet-Plus の新しいバージョンが必要

V7.3-2

インストールまたはアップグレードを行って OpenVMS Alpha Version 7.3-2 以降のバージョンにするときには、DECnet-Plus の新しいバージョンもインストールしなければなりません。このインストールが必要な理由の 1 つに、Version 7.3-2 で導入された、AUTOGEN の動作の変更があります。

以前のバージョンの動作と異なり、DECnet-Plus for OpenVMS Version 7.3-2 以降では、AUTOGEN が必要としているように、製品情報を NEWPARAMS.DAT レコードに格納するようになりました。AUTOGEN は、DECnet-Plus にこの変更が行われていることを前提としているため、CLUSPARAMS.DAT から「不正」レコードを削除する際に、警告を出力しません。AUTOGEN は、これらのレコードは古い DECnet-Plus キットが作成したものなので、新しい DECnet-Plus キットが置き換えるものと見なします。このため通常の状態では、OpenVMS Version 7.3-2 以降のインストールまたはアップグレードの際に、動作に大きな違いは見られません。

ただし、他の製品が NEWPARAMS.DAT レコードに製品情報を格納しない場合、AUTOGEN は、レポートと、ユーザの SYS\$OUTPUT デバイスの両方に警告メッセージを出力します。これは、AUTOGEN がこの製品情報を必要とするようにな

ったためです。この警告は、(製品名が付加されていないため) NEWPARAMS.DAT にあるパラメータ割り当てを AUTOGEN が受け入れられないことと、レコードが CLUSPARAMS.DAT に追加されないことを示します。レコードが追加されないため、期待されている追加や変更が SYSGEN パラメータに行われず、リソースが足りなくなる可能性があります。ソフトウェア製品の開発者とテスト者は、この要件を意識しなければなりません。また、システム管理者にも関係する場合があります。

この新しい動作は、レイヤード・プロダクトのユーザと提供者の両方を守ることを目的としています。適切にアップデートされるようにこの情報を適切な順序にすると、不正なアップデートによって起こる問題を最小限にできます。

NEWPARAMS.DAT と CLUSPARAMS.DAT の説明は、『OpenVMS システム管理ユーティリティ・リファレンス・マニュアル』の AUTOGEN の章を参照してください。

1.16 SHADOW_MAX_UNIT のデフォルト設定とメモリ使用量

V7.3-2

この注意事項は、以前の注意事項を更新するものです。以前の注意事項は、このシステム・パラメータのデフォルト設定について説明していましたが、デフォルト設定で消費されるメイン・メモリの量については説明していませんでした。

OpenVMS Alpha Version 7.3 で、HP Volume Shadowing for OpenVMS でのミニコピーがサポートされました。ミニコピー機能の一環として、新しいボリューム・シャドウイング・システム・パラメータ、SHADOW_MAX_UNIT が導入されました。OpenVMS Alpha システムでは、このシステム・パラメータのデフォルト値は 500 で、メイン・メモリを 24 KB 消費します。OpenVMS VAX システムでは、デフォルト値は 100 で、メイン・メモリを 5 KB 消費します。

Volume Shadowing for OpenVMS を使用する予定がない場合は、この設定を最小値の 10 に変更することができます (この場合、消費するメイン・メモリは 480 バイトです)。デフォルト値から最小値に変更すると、OpenVMS Alpha システムでは 23.5 KB のメイン・メモリ、VAX システムでは 4.5 KB のメイン・メモリを解放することができます。

注意

SHAD_MAX_UNIT は、静的なシステム・パラメータです。新しい設定値を有効にするには、システムをリブートしなければなりません。

ボリューム・シャドウイングの SHADOW_MAX_UNIT 設定の推奨値については、『Volume Shadowing for OpenVMS 説明書』を参照してください。

1.17 アップグレード前に TIE Kit の削除が必要

V8.2-1

Translated Image Environment (TIE) は OpenVMS I64 Version 8.2-1 に統合されました。詳細は、HP OpenVMS Systems Migration Software の Web サイトを参照してください。

<http://www.hp.com/go/openvms/products/omsais>

OpenVMS I64 Version 8.2 または Version 8.2-1 に Translated Image Environment (TIE) PCSI キット (HP-I64VMS-TIE) のいずれかのバージョンをインストールしてある場合は、OpenVMS I64 Version 8.3 にアップグレードする前に手動で TIE キットを削除する必要があります。

TIE 製品キットを削除するには次のコマンドを使用します。

```
$ PRODUCT REMOVE TIE
```

TIE 製品キット HP I64VMS TIE V1.0 は、OpenVMS I64 Version 8.2-1 や Version 8.3 にインストールしないでください。

1.18 レイヤード・プロダクトを代替デバイスや代替ディレクトリにインストールすると失敗する

V8.3

デフォルトでは、PRODUCT INSTALL コマンドはレイヤード・プロダクトをシステム・デバイスの SYS\$COMMON ディレクトリ・ツリーにインストールします。/DESTINATION=dev:[dir] 修飾子を使用するか、論理名 PCSI\$DESTINATION を定義することで、レイヤード・プロダクトを代替デバイスまたは代替ディレクトリにインストールすると、ファイル [SYSLIB]DCLTABLES.EXE、[SYSHLP]HELPLIB.HLB、[SYSLIB]STARLET*.* のいずれかが見つからないことを示すエラー・メッセージが表示され、インストールに失敗することがあります。その場合は、"Do you want to terminate? [YES]" という質問に YES と答え、/NORECOVERY_MODE 修飾子を使用してインストールを再試行してください。

OpenVMS の関連製品に関する注意事項

この章では、OpenVMS 関連製品について説明します。関連製品のインストール/アップグレードに特に関連する事項については、第 1 章を参照してください。

コンパイラ、リンカ、ランタイム・ライブラリ・ルーチンの使用に関する注意事項については、第 5 章を参照してください。

2.1 関連製品のサポート

Software Public Rollout Reports for OpenVMS には、OpenVMS Alpha および OpenVMS VAX 用の Software Products Library キット (CD-ROM)、および OpenVMS I64 用の Layered Product Library キット (DVD) で提供されている弊社のソフトウェア製品の情報が記載されています。

レポートには製品名とバージョン、製品をサポートするのに必要なオペレーティング・システムのバージョン、製品のボリューム出荷日が示されています。レポートの情報は今後も追加され、変更される可能性があります。レポートは公開され、毎月更新されます。レポートの情報はたえず変化するので、リリース・ノートには含まれません。

Software Public Rollout Reports for OpenVMS は、次の Web サイトから入手できます。

<http://h71000.www7.hp.com/openvms/os/swroll/>

インターネットにアクセスできない場合は、四半期ごとに提供される Software Products Libraries からオペレーティング・システム・サポート情報を検索できます。このライブラリは、次のファイルにあります。

[README]SW_COMPAT_MATRIX.PS
[README]SW_COMPAT_MATRIX.TXT

また、Software Public Rollout Reports は弊社のサポート担当者から入手することもできます。

OpenVMS Version 7.3-2 およびそれ以降の変更により、V1.5A より前のバージョンの BASIC では、インストール中に BASIC\$STARLET ライブラリ・ファイルを作成できません。

STARLET ライブラリ・ファイルの構築を要求しなければ、以前のバージョンの BASIC を OpenVMS Version 7.3-2 およびそれ以降の上にインストールすることができます。また、以前にインストールされた BASIC コンパイラと、以前に作成された STARLET ライブラリ・ファイルは、古い OpenVMS システムを Version 7.3-2 およびそれ以降にアップグレードした後も動作します。

OpenVMS Version 7.3-2 およびそれ以降で動作しないのは、BASIC\$STARLET ライブラリ・ファイルの作成だけです。BASIC V1.5A キットには、STARLET ライブラリ・ファイルを OpenVMS Version 7.3-2 およびそれ以降上で正しく構築する拡張インストール・プロシージャが含まれています。

BASIC V1.6 は、最新のコンソリデートド・レイヤード・プロダクト CD で入手できます。

2.2 追加された CMAP ファイル

V8.2

次の新しい CMAP ファイルが、OpenVMS Version 8.2 国際化データ・キットで提供されています。

DECKANJI2000
GB18030
ISO8859-1-EURO
UTF8-20
UTF8-30

2.3 COBOL: I/O 実行時診断と RMS 特殊レジスタの変更

V7.3

OpenVMS Alpha Version 7.2 で Extended File Support が追加されたことにより、OpenVMS Alpha Version 7.2 以降での I/O 実行時診断メッセージおよび RMS 特殊レジスタの処理が変わりました。特に、Version 7.2 より前のバージョンの OpenVMS では、長いファイル名により RMS\$_FNM が発生しましたが、OpenVMS Alpha Version 7.2 以降のバージョンでは RMS\$_CRE が発生するようになりました。これらの RMS に関する違いをサポートするために、新しい ODS-5 サポートを使用する必要があります。

2.4 COM for HP OpenVMS (Alpha のみ)

ここでは、COM for HP OpenVMS に関する注意事項について説明します。

2.4.1 COM for OpenVMS のサポート

V8.2

現在 COM Version 1.4 for OpenVMS は、OpenVMS Alpha Version 7.3-2 および 8.2 でサポートされています。COM for OpenVMS についての最新情報は次の Web サイトを参照してください。

<http://h71000.www7.hp.com/openvms/products/dcom/>

2.4.2 高負荷時に発生するアプリケーションのレジストリ・アクセス・エラー

V7.3-2

CTLPAGES 値に 256 以下の値を設定して、COM for OpenVMS アプリケーションを負荷の高い状態で実行した場合、Error accessing registry database, contact system manager (0x000025fc) メッセージが出力されることがあります。この問題を回避するには、CTLPAGES 値に 512 を設定してください。

2.5 OpenVMS Version 8.3 では DECdfs Version 2.4 が必要

V8.3

OpenVMS Version 8.3 には、DECdfs Version 2.4 が必要です。古いバージョンの DECdfs を使用すると、エラー・メッセージが表示されます。

2.6 DEC PL/I: OpenVMS での RTL のサポート

V7.3

OpenVMS オペレーティング・システムで配布される PL/I RTL と、Kednos Corporation が所有し配布する新しい PL/I RTL の間には、互換性はありません。OpenVMS オペレーティング・システムと同時に配布された旧バージョンが、より新しいバージョンを上書きしてしまうことがあります。該当するイメージ・ファイルは、SYS\$LIBRARY:DPLI\$RTL\$SHR.EXE です。

OpenVMS では、DCL の ANALYZE/IMAGE コマンドで次のように表示されるバージョンのファイルが配布されます。

Image Identification Information

image name: "DPLI\$RTL\$SHR"

image file identification: "V4.0-6"

OpenVMS Version 7.3 以降にアップグレードする前に ANALYZE/IMAGE コマンドを実行し、より新しいバージョンの DPLISRTL.SHR.EXE が存在する場合、コピーしておきアップグレード後に復元するか、後で PL/I キットを再インストールしてください。

第 5.31 節の、関連する注意を参照してください。

2.7 FMS キット

V8.3

次の FMS キットのいずれか (またはそれより新しいもの) を OpenVMS Alpha および OpenVMS I64 システムにインストールできます。

フルキット: HPFMS025

実行時キット: HPFMSRT025

FMS V2.4 は OpenVMS Alpha V8.3 でサポートされます。

FMS V2.5 は OpenVMS V8.2 以降のシステム (Alpha および I64) でサポートされません。

2.8 Graphical Configuration Manager (GCM) (Alpha のみ)

Graphical Configuration Manager (GCM) は、オペレーティング・システムと一緒に出荷される Layered Products CD に含まれています。ただし、GCM は頻繁にアップデートされます。次の Web ページを定期的にチェックして、新しいバージョンのソフトウェアがリリースされていないか確認してください。

<http://h71000.www7.hp.com/openvms/products/gcm/>

2.9 HP DCE RPC for OpenVMS

ここでは、OpenVMS Version 8.2 の HP DCE RPC に関する注意事項について説明します。

2.9.1 DCE RPC による FAILSafe IP のサポート

V8.2

DCE RPC は FAILSafe IP の環境で動作するように拡張されました。

2.9.2 RPC ソケットにおけるバッファ・サイズ・チューニングのサポート

V8.2

DCE RPC は、論理名RPC_USE_DEFAULT_SOCKET_BUFFER_SIZEを使用してソケットのバッファ・サイズをチューニングできるようになりました。この論理名を設定することで、RPC ランタイム・ライブラリで、システムのデフォルトのソケット・バッファ・サイズを使用できるようになります。論理名をシステム全体に対して定義するには、次のコマンドを実行します。

```
$ DEFINE/SYSTEM RPC_USE_DEFAULT_SOCKET_BUFFER_SIZE 1
```

RPC ランタイム・ライブラリの動作をオリジナルに戻すには、論理名RPC_USE_DEFAULT_SOCKET_BUFFER_SIZEの割り当てを解除してください。

2.9.3 RTI (Remote Task Invocation) RPC (I64 のみ)

V8.2

I64 Version 8.2 の RPC は RTI RPC をサポートしていません。詳細については、キットに付属しているドキュメント『HP DCE for OpenVMS Alpha and OpenVMS I64 Release Notes』を参照してください。

2.9.4 Microsoft LAN Manager RPC はテストされていない (I64 のみ)

V8.2

NTLM (Microsoft 社の LAN Manager プロトコル) 上の Authenticated RPC は、OpenVMS I64 ではテストされていません。これは、DCE RPC が依存しているインフラストラクチャが、OpenVMS I64 上では使用できないためです。

2.9.5 utc_mulftime 係数引数の型

V8.2

DTSS API ルーチンの入力引数である係数utc_mulftimeは、I64 システムではIEEE_FLOAT 型、Alpha システムではG_FLOAT 型でなければなりません。CVT\$FTOF または CVT\$CONVERT_FLOAT を使用すると、utc_mulftimeを呼び出す前に、この係数引数を適切な浮動小数点型に変換することができます。

2.9.6 G_FLOAT 浮動小数点型と IEEE 浮動小数点型のサポート

V8.2

DCE RPC for OpenVMS は現在、I64 プラットフォームと Alpha プラットフォーム上で、G_FLOAT 浮動小数点型と IEEE 浮動小数点型の両方をサポートしています。

1 つの RPC アプリケーション内では、同じ浮動小数点型を使用してください。異なる RPC アプリケーションであれば、1 つのシステム内でそれぞれ異なる浮動小数点型を使用しても適切に動作します。

I64 システム

I64 システム上では、DCE はデフォルトで IEEE_FLOAT 型を使用します。つまり、I64 システム用にビルドされた DCE アプリケーションは、IEEE_FLOAT 浮動小数点型を使用します。

C または C++ 言語で開発された RPC アプリケーションで G_FLOAT 浮動小数点型を使用するには、次の手順に従います。

1. RPC の実行時関数を呼び出す前に、新しい API 関数 `rpc_set_local_float_drep(RPC_APPLICATION_FLOAT_TYPE, &status)` を呼び出します。RPC_APPLICATION_FLOAT_TYPE 定数に対して、コンパイラのコマンド行修飾子で指定された浮動小数点型が自動的に定義されます。詳細は、『Release Notes for OpenVMS DCE V3.2』を参照してください。
2. コンパイラの修飾子/FLOAT=G_FLOAT を使用して、RPC アプリケーション・プログラムをコンパイルします。
3. スタブのビルド時に、以下のうち適切な IDL コンパイル・オプションを使用します。

C アプリケーションの場合: `-CC_CMD "CC/FLOAT=G_FLOAT"`

C++ アプリケーションの場合: `-CPP_CMD "CXX/FLOAT=G_FLOAT"`

4. 以下のうち適切な DCE オプション・ファイルを使用して、RPC アプリケーションをリンクします。

C アプリケーションの場合: `DCE.OPT`

C++ アプリケーションの場合: `DCE_CXX.OPT`

C または C++ 言語で開発された RPC アプリケーションで IEEE_FLOAT 浮動小数点型を使用するには、次の手順に従います。

1. コンパイラの修飾子/FLOAT=IEEE_FLOAT (デフォルトのオプション) を使用して、RPC アプリケーション・プログラムをコンパイルします。
2. DCE.OPT または DCE_CXX.OPT を使用して RPC アプリケーションをリンクします。

Alpha システム

デフォルトでは、Alpha システム上でビルドされた DCE アプリケーションは、G_FLOAT 浮動小数点型を使用します。

C または C++ 言語で開発された RPC アプリケーションで IEEE_FLOAT 浮動小数点型を使用するには、次の手順に従います。

1. RPC の実行時関数を呼び出す前に、新しい API 関数 `rpc_set_local_float_drep(RPC_APPLICATION_FLOAT_TYPE, &status)` を呼び出します。RPC_APPLICATION_FLOAT_TYPE 定数に対して、コンパイラのコマンド行修飾子で指定された浮動小数点型が自動的に定義されます。詳細は、『Release Notes for OpenVMS DCE V3.2』を参照してください。
2. コンパイラの修飾子/FLOAT=IEEE_FLOAT を使用して、RPC アプリケーション・プログラムをコンパイルします。
3. スタブのビルド時に、以下のうち適切な IDL コンパイル・オプションを使用します。

C アプリケーションの場合: `-CC_CMD "CC/FLOAT=IEEE_FLOAT"`

C++ アプリケーションの場合: `-CPP_CMD "CXX/FLOAT=IEEE_FLOAT"`

4. 以下のうち適切な DCE オプション・ファイルを使用して、RPC アプリケーションをリンクします。

C アプリケーションの場合: `DCE.OPT`

C++ アプリケーションの場合: `DCE_CXX.OPT`

C または C++ 言語で開発された RPC アプリケーションで G_FLOAT 浮動小数点型を使用するには、次の手順に従います。

1. C または C++ コンパイラの修飾子/FLOAT=G_FLOAT (デフォルトのオプション) を使用して、RPC アプリケーション・プログラムをコンパイルします。
2. DCE.OPT または DCE_CXX.OPT を使用して RPC アプリケーションをリンクします。

2.10 HP DECnet-Plus での大文字小文字混在のパスワードのサポート

V8.3

OpenVMS Versions 7.3-2 およびそれ以降では、ログイン・フラグとして PWDMIX が指定されている場合、パスワードに大文字小文字を混在させることや、拡張文字を使用することができます。DECnet-Plus V8.3 より前では、DECnet はタスク間通信や、DECnet を使用するリモート・ファイル・アクセスで、大文字小文字混在のパスワードをサポートしていませんでした。

2.11 HP DECnet-Plus for OpenVMS: X.25 データ・リンクがサポートされていない (I64 のみ)

V8.2

HP X.25 for OpenVMS Alpha ソフトウェアは移植作業中で、まだサポートされていません。このため、HP DECnet-Plus for OpenVMS I64 Version 8.2 は、X.25 データ・リンクをサポートしていません。

2.12 HP DECram

ここでは、DECram に関する注意事項について説明します。

注意

HP DECram についての詳細は、第 1.13 節を参照してください。

2.12.1 DECram は OpenVMS Version 8.2 に含まれる

V8.2

OpenVMS Alpha および OpenVMS I64 の Version 8.2 以降、DECram は System Integrated Product (SIP) として OpenVMS オペレーティング・システムに統合されて提供されます。ユーザは、DECram のライセンスを引き続き必要とします。DECram ドライバは、SYS\$COMMON:[SYS\$LDR]にあります。Alpha のユーザは、システム固有のディレクトリ ([SYSx.SYS\$LDR]) 内に残っている SYS\$MDDRIVER イメージを削除する必要があります。Version 8.2 へアップグレードする前に DECram の旧バージョンを削除する手順は、『HP OpenVMS Version 8.2 Upgrade and Installation Manual』を参照してください。

旧バージョンの DECram をロードしようとする、次のエラー・メッセージが表示されます。

SYSTEM-W-SYSVERDIF, system version mismatch; please relink

OpenVMS Version 8.2 では、旧バージョンの DECram はサポートされていません。

DECram Version 2.5 は、引き続き VAX システムでのみサポートされます。

2.12.2 DECRYPT DCL コマンドとの競合

V8.2

Encryption for OpenVMS Alpha レイヤード・プロダクトでは、インストール時に専用の DCL コマンド DECRYPT を生成します。DECRYPT は、DCL コマンド DECR のデフォルトの定義を上書きしますが、これはユーザが DECram を呼び出すために使用している可能性があります。

両方の製品がインストールされた場合、次のような DCL フォーリン・コマンド・シンボルを定義することにより DECram インタフェースにアクセスすることができます。

```
$ DECram == "$MDMANAGER"
```

2.13 HP DECwindows Motif for OpenVMS

ここでは、HP DECwindows Motif for OpenVMS に関する注意事項について説明します。

2.13.1 DECwindows のポーズ画面でロック解除できない問題 (Alpha のみ)

V8.3

V8.3 よりも前のバージョンでは、外部認証環境で以下の条件が揃った場合、ユーザは DECwindows のワークステーションのロックを解除できませんでした。

- ユーザが SYSUAF データベース上で EXTAUTH としてマークされている。
- ユーザがユーザ名/LOCAL_PASSWORD 修飾子を使用して DECwindows のワークステーションにログインしている。
- ポーズ画面が起動されている。

この状況下では、ワークステーションは有効なパスワードを認識できないため、パスワードの入力を要求し続けます。この状況から抜け出すには、別のソース (たとえばネットワークや LAT) からログインし、@SYS\$STARTUP:DECW\$STARTUP RESTART と入力して DECwindows サーバを再起動する必要がありました。

この問題は V8.3 で修正されています。外部認証環境では次の点に注意してください。

1. user_name/local_pass としてログインするには、SYSUAF データベースで VMSAUTH のフラグがあることを確認してください (OpenVMS V8.2 ではこれは必要ありませんでした)。

2. user_name/local_passでログインしている場合であっても、DECwindows デスクトップで "Screen Lock" をオフにしてください。

2.13.2 新しいロケールの追加

V8.2

ローカライズされた DECwindows Motif ソフトウェアが使用する次の新しいロケールが、OpenVMS Version 8.2 国際化データ・キットに追加されました。

iw_IL.utf-8 (ヘブライ、イスラエル、UTF-8)
ko_KR.utf-8 (韓国、UTF-8)
zh_CN.utf-8 (中国、PRC、UTF-8)
zh_HK.utf-8 (中国、香港、UTF-8)
zh_TW.utf-8 (中国、台湾、UTF-8)

2.13.3 ユーザが作成したトランスポートはサポートされない

V7.3-2

DECwindows Motif Version 1.3 for OpenVMS Alpha では、クライアント間交換 (ICE) プロトコル、低帯域幅 X (LBX) プロキシ・サーバ、および入力メソッド・サーバのマルチスレッドおよび通信のニーズに応じて、DECwindows Motif のトランスポート・ライブラリが大きく変更されました。この結果、弊社は、DECwindows Motif Version 1.3 以降を実行しているシステムでの、ユーザ作成ネットワーク・トランスポートのサポートを中止しました。

既存のトランスポート (DECnet、TCP/IP、LAT、および LOCAL) はそのまま使用でき、問題なく機能しますが、更新されたトランスポート・インタフェースに基づくユーザ作成トランスポートの設計および実装に対するサポートは今後行われません。『VMS DECwindows Transport Manual』はアーカイブに入れられ、新しいライブラリは公開されません。

ユーザが作成したトランスポートを実装し、そのトランスポートを DECwindows Motif Version 1.5 以降の環境に移行する場合は、移行戦略の作成方法について、弊社のサポート担当者にお問い合わせください。

2.14 HP Secure Web Server のサポート

V8.2

OpenVMS Alpha Version 7.3-2 および OpenVMS Version 8.2 (Alpha および I64) は、Secure Web Server (SWS) Version 1.3-* をサポートする最後のリリースです。

OpenVMS Alpha Version 7.3-2 は、SWS Version 2.0 をサポートする最後のリリースです。

SWS Version 1.3-*およびSWS Version 2.0 の代替機能は、SWS Version 2.1 です。今後のSWSの新機能や機能拡張は、Apache 2.0.*オープン・ソース・コードをベースにしたSWS 2.1 で提供されます。

OpenVMS Alpha Version 7.3-2 がPVS (Prior Version Support) の状態である限り、SWS Version 1.3-*およびSWS Version 2.0 に対するサポートは継続します。また、OpenVMS Version 8.2 がサポートされている限り、SWS Version 1.3-*に対するサポートは継続します。これらのSWSバージョンのサポートには、必要と判断された修正パッチ、セキュリティ・パッチが含まれます。

2.15 HP Pascal for OpenVMS Alpha Systems

ここでは、OpenVMS Alpha システム上のHP Pascalに関する注意事項について説明します。

2.15.1 STARLET ライブラリの作成にはV5.8A (以降) が必要 (Alpha のみ)

V7.3-2

OpenVMS Version 7.3-2 の変更により、V5.8A より前のバージョンのPascalでは、インストール中にSTARLET ライブラリ・ファイルを作成できません。

STARLET ライブラリ・ファイルを作成してインストールするかという質問に"NO"とこたえと、OpenVMS Version 7.3-2 以降上に、以前のバージョンのPascalをインストールできます。また、以前にインストールされたPascalコンパイラと、以前に作成されたSTARLET ライブラリ・ファイルは、古いOpenVMS システムをVersion 7.3-2 以降にアップグレードした後も動作します。

OpenVMS Version 7.3-2 以降で動作しないのは、Pascal のインストールの、STARLET ライブラリ作成の部分だけです。Pascal V5.8A キットには、STARLET ライブラリ・ファイルをOpenVMS Version 7.3-2 以降上で正しく構築する拡張インストール・プロシージャが含まれています。

Pascal V5.8A は、最新のコンソリデートッド・レイヤード・プロダクトCDで入手できます。

2.15.2 アップグレード後の HP Pascal のインストール (Alpha のみ)

V7.3

この注意事項は、すべてのバージョンの HP Pascal と、すべてのバージョンの OpenVMS Alpha オペレーティング・システムにあてはまります。

OpenVMS のアップグレード後、HP Pascal を再インストールして、新しいバージョンの STARLET.PAS ファイルと他の定義ファイルを作成し、アップグレード後のシステムと一致させなければなりません。

OpenVMS のアップグレード後に HP Pascal を再インストールしない場合でも、コンパイラは正しく起動しますが、STARLET.PAS ファイルと他の定義ファイルに、OpenVMS のアップグレードで提供される新しい定義や修正後の定義が含まれません。

2.16 I64 システムでの WEBES および SEA のサポート

V8.3

最新版の WEBES (WEBased Enterprise Services) は、次の WEBES のホームページから入手できます。

<http://h18000.www1.hp.com/support/svctools/>

2.17 NetBeans Version 3.6 では Java Standard Edition, Development Kit v 1.4.2-x が必要

V8.3

OpenVMS Alpha および OpenVMS I64 用の NetBeans Version 3.6 は、Java Platform, Standard Edition, Development Kit (JDK) v 1.4.2-x 上での動作だけがサポートされます。

一般ユーザ向けの注意事項

この章では、OpenVMS オペレーティング・システムのすべてのユーザのための情報をまとめます。一般に使用するコマンドとユーティリティに関する情報が含まれています。

オペレーティング・システムのこのバージョンで提供される新機能については、『HP OpenVMS V8.3 新機能説明書』を参照してください。

3.1 OpenVMS Freeware CD

V8.3

OpenVMS Version 8.3 メディア・キットには、OpenVMS Freeware Version 8.0 CD が付属しています。Freeware CD には、アプリケーションの作成や、OpenVMS システムの使用あるいは管理のためのフリー・ソフトウェア・ツールとユーティリティが収録されています。

システムに Freeware CD をマウントし内容を表示するには、CD を CD ドライブに挿入し、次のコマンドを入力します。

```
$ MOUNT/OVERRIDE=IDENTIFICATION ddcu:
```

このコマンドでは、ddcu:の箇所に、OpenVMS システム上の CD ドライブまたは DVD ドライブのデバイス名を指定します。このデバイス名は OpenVMS システムごとに異なります。

```
$ TYPE ddcu:[FREWARE]FREWARE_README.TXT
```

このファイルのコピーは、Freeware 8.0 ディストリビューションの各ボリュームにも格納されており、TYPE コマンドや使い慣れたテキスト・エディタでその内容を参照できます。

フリーウェアについての詳細は、FREWARE_README.TXT ファイルを参照してください。

適切なデバイスをマウントしたら、DIRECTORY や COPY などの標準の DCL コマンドを使用して、キット・ディレクトリに直接アクセスできます。各ディスクの[FREWARE]ディレクトリに、フリーウェアの要約を記述したテキスト・ファイル、その他の資料があります。

[FREEWARE]FREEWARE.COM フリーウェア・メニュー・システム・インタフェースは、Freeware 8.0 ディストリビューションから削除されました。

3.1.1 フリーウェア・メニューが利用できない (I64 のみ)

V8.2

OpenVMS Freeware V7.0 CD-ROM に含まれている [FREEWARE]FREEWARE.COM はフリーウェア・メニュー・システム・インタフェースですが、これは OpenVMS I64 システムでは動作しません。

このメニュー・システム・インタフェースは、OpenVMS Alpha および OpenVMS VAX システム上で動作します。

OpenVMS I64, OpenVMS Alpha, または OpenVMS VAX システムの、DIRECTORY や COPY などの DCL コマンドを使用して、Freeware V7.0 CD-ROM の内容に直接アクセスできます。これは、Freeware V7.0 ディストリビューションにアクセスする際の望ましい方法です。

フリーウェアと Freeware ディストリビューションについての情報は [FREEWARE]FREEWARE_README.TXT に収められています。このファイルは Freeware V7.0 ディストリビューションの各ボリュームに格納されており、内容は TYPE コマンドまたはテキスト・エディタで確認できます。

3.2 DCL コマンド

ここでは、DCL コマンドに関する注意事項について説明します。

3.2.1 CREATE/MAILBOX: 一時的な制約

V8.2

CREATE/MAILBOX/TEMPORARY には、現在 CMEXEC 特権が必要です。この制約は将来のリリースでは解除されます。

3.2.2 DIAGNOSE コマンドはサポートされない

V8.2

3.3 Open Source Tools CD での DECmigrate の提供について

V8.2

OpenVMS Migration Software for VAX to Alpha (DECmigrate) は、OpenVMS Version 8.2 インストール・キットに含まれている Open Source Tools CD では提供されていません。OpenVMS Version 7.3-2 用のメディアにはこのソフトウェア・キットが含まれていましたが、OpenVMS Version 8.2 ではテストが完了していません。以前のバージョンの OpenVMS 用のソフトウェアについては、次の Web サイトから引き続き利用できます。

<http://h71000.www7.hp.com/openvms/products/omsva/omsva.html>

OpenVMS Version 8.2 でのサポートが可能になれば、アップデート版が掲載されます。

3.4 HP Secure Web Browser

ここでは、HP Secure Web Browser に関する注意事項について説明します。

3.4.1 必要メモリ量の増加

V7.3-1

OpenVMS ワークステーションで、Mozilla ベースの HP Secure Web Browser (SWB) を使用している場合は、256 MB 以上のメモリが必要です。ただし、処理を安定させるために、メモリを 512 MB にすることをお勧めします。

3.4.2 ODS-2 ディスク・ボリュームで発生するインストール・エラー (I64 のみ)

V8.2

OpenVMS I64 用 HP Secure Web Browser (CSWB) Version 1.4 を ODS-2 ディスク・ボリュームにインストールすると、次のように PCSI エラーとなります。

```
%PCSI-E-OPENIN, error opening
ODS2$DISK:[SYS0.SYSCOMMON.][CSWB.RES]SAMPLE^.UNIXPSFONTS.PROPERTIES;* as input
-RMS-E-FND, ACP file or directory lookup failed
-SYSTEM-W-BADFILEVER, bad file version number
%PCSI-E-OPFAILED, operation failed
```

Do you want to terminate?とプロンプトが出たら "NO" と答えることでインストールを継続できます。インストールは正常に継続できます。

代替手段として、ODS-5 ディスク・ボリュームに HP Secure Web Browser をインストールすることもできます。

3.5 ドキュメントの訂正

ここでは OpenVMS ドキュメント・セット内の各種マニュアルの訂正と追加について説明しています。

3.5.1 『HP OpenVMS System Analysis Tools Manual』

『HP OpenVMS System Analysis Tools Manual』に対する変更と更新については、第 4 章を参照してください。

3.5.2 『HP OpenVMS Programming Concepts Manual』

『HP OpenVMS Programming Concepts Manual』では以下の点が訂正されています。

31.2 項の「Writing a Privileged Routine (User-Written System Service)」の段落に対して、以下の変更を行う必要があります。

「作成したプログラムは、保護されたイメージであるため、オペレーティング・システムのプログラミング環境全体を自由に使用することはできません。モジュールにプレフィックス SYSS\$または EXES\$がないかぎり、内部モードからモジュールを呼び出すことは避けてください。特に、LIB\$GET_VM や LIB\$RET_VM は内部モードから呼び出さないでください。OpenVMS RMS のルーチンは、エグゼクティブ・モードから呼び出すことはできますが、カーネル・モードからは呼び出すことができません。」

LIB\$GET_VM は間違いで、正しくは LIB\$FREE_VM です。これらの LIBRTL ルーチンを直接呼び出すことも、これらのルーチンを現在または将来間接的に呼び出すルーチンを呼び出すこともできません。これには、LIBRTL 内の他のルーチンや、ユーザモードの C ライブラリなどが含まれます。

3.5.3 『HP OpenVMS デバッガ説明書』：クライアント/サーバ・インタフェースの訂正

Version 8.2 の『HP OpenVMS デバッガ説明書』には、PC クライアント・インタフェース・キットに関して間違った情報が記載されています。第 11.1 節と第 11.2 節に対する訂正を以下に示します。

デバッグ・サーバは OpenVMS システム上で動作します。クライアントはデバッグに対するユーザ・インタフェースであり、そこからユーザがデバッグ・コマンドを入力しサーバに送られます。サーバはコマンドを実行し、結果がクライアントに送り返されて表示されます。クライアントは、Microsoft® Windows® 95、Windows 98、Windows NT®、Windows 2000、Windows XP®上で動作します。

上記のクライアント・プラットフォームに対する正しいクライアント・キットは次のとおりです。

[DEBUG_CLIENTS011.KIT] DEBUGX86011.EXE

OpenVMS 上で動作するコンポーネントについては、特別なインストール手順はありません。システム管理者は上記の OpenVMS デバッグ・クライアント・キットを OpenVMS ディストリビューション・メディアから PATHWORKS シェアや FTP サーバなど PC ユーザがアクセスできる場所に移動する必要があります。クライアント・キットは自己解凍形式の.EXE ファイルです。

適切な実行可能ファイルを PC に転送したら、そのファイルを実行してデバッグ・クライアントを PC にインストールします。InstallShield のインストール・プロセスによって、インストール手順が案内されます。

デフォルトでは、デバッグ・クライアントは\Program Files\OpenVMS の Debugger フォルダにインストールされますが、[参照]をクリックして、別の場所を選択することもできます。

インストール手順によって、OpenVMS Debugger のプログラム・フォルダが作成されます。そこには、以下の項目へのショートカットが格納されます。

- デバッグ・クライアント
- デバッグ・クライアントのヘルプ・ファイル
- READMEファイル
- アンインストール・プロセス

3.5.4 『HP OpenVMS DELTA/XDELTA Debugger Manual』の更新

V8.3

HP DELTA デバッグが OpenVMS I64 Version 8.2-1 で利用可能になりました。『HP OpenVMS DELTA/XDELTA Debugger Manual』が本リリースで改訂され、OpenVMS I64 システムで DELTA を利用するための情報が追加されました。

3.5.5 『HP OpenVMS Utility Routines Manual』の更新

V8.3

『HP OpenVMS Utility Routines Manual』が改訂され、以下の情報が追加されました。

- I64 システムと Alpha システムでの Traceback ファシリティと TBK\$ルーチンに関する新しい章を追加
- Library ユーティリティの章を改訂し、LBR\$ルーチンの追加および変更を実施
- I64 での AES 暗号化について新しい章を追加

3.5.6 『OpenVMS I/O User's Reference Manual』：PTD\$READ の明確化

『OpenVMS I/O User's Reference Manual』では、読み込みリクエストの動作が十分に説明されていません。次の段落は、第 6.5.1 項の最初の段落を置き換えるものです。

制御プログラムは、データを擬似端末から読み込むために、PTD\$READ ルーチンを使います。PTD\$READ ルーチンが呼び出されると、オペレーティング・システムによって、読み込み操作がキューに登録されます。読み込み操作は、擬似端末が出力しようとする文字があれば終了します。読み込みリクエストは TTDRIVER に対して、返すべきデータがあるかどうかを照会します。データがあった場合には、その文字列が返されます。読み込みリクエストが発行されてもデータが存在しない場合には、読み込みリクエストはキューに登録され、終了するのは後になります。この場合には、ルーチンは少なくとも 1 文字を返します。読み込みリクエストは、出力として返す文字がなくても終了することがあります。TTDRIVER が出力として返すデータがないことを示し、実際にデータがないという稀な状況では、読み込み操作は 0 バイトのデータで終了します。

以下の段落は、第 D.4.4 項を置き換えるものです。

PTD\$READ ルーチンは擬似端末からデータを読み込みます。読み込みリクエストは最小で 1 文字、そして最大で readbuf_len 引数で指定した数の文字を読み込んで終了します。

PTD\$READ ルーチンが呼び出されると、オペレーティング・システムによって、読み込み操作がキューに登録されます。読み込み操作は、擬似端末が出力しようとする文字があれば終了します。読み込みリクエストは TTDRIVER に対して、返すべきデータがあるかどうかを照会します。データがあった場合には、その文字列が返されます。読み込みリクエストが発行されてもデータが存在しない場合には、読み込みリクエストはキューに登録され、終了するのは後になります。この場合には、ルーチンは少なくとも 1 文字を返します。読み込みリクエストは、出力として返す文字がなくても終了することがあります。TTDRIVER が出力として返すデータがないことを示

し、実際にデータがないという稀な状況では、読み込み操作は 0 バイトのデータで終了します。

3.5.6.1 PTD\$疑似端末ドライバの訂正

V8.3

疑似端末ドライバの制御接続ルーチンに関して、プレフィックス PTD\$は正しくありません。正しいプレフィックスは PTDS\$です。

3.5.7 『HP OpenVMS Version 8.2 新機能説明書』： Librarian ユーティリティの訂正

以下のリリース・ノートは、OpenVMS I64 Librarian ユーティリティに関する訂正情報です。

3.5.7.1 /REMOVE 修飾子の訂正

『HP OpenVMS Version 8.2 新機能説明書』の第 4.8.2.3 項にある、Librarian の拡張された /REMOVE 修飾子についての説明は誤っています。正しい説明は次のとおりです。

I64 Librarian ユーティリティでは、/REMOVE 修飾子の機能が拡張されました。拡張された形式では削除するシンボルのモジュール・インスタンスを指定できるようになりました。拡張された /REMOVE 修飾子では、LIBRARY コマンドに対して、オブジェクト・ライブラリのグローバル・シンボル・テーブルから 1 つまたは複数のエントリを削除するように要求します。

3.5.7.2 ELF オブジェクト・ライブラリへのアクセスについての訂正

『HP OpenVMS Version 8.2 新機能説明書』の第 4.8.3.2 項には誤った説明があります。以下の文章はその項の説明を置き換えるものです。

ELF オブジェクト・ライブラリへのアクセス

OpenVMS Alpha オブジェクト、テキスト・モジュールなどは、シーケンシャル・アクセス・モジュールですが、ELF オブジェクト・モジュールは、本質的に、ランダム・アクセス・モジュールです。ランダムにアクセスできるように、1 つの新しいライブラリ・ルーチンが作成されました。このルーチンを使うと、ELF オブジェクト・モジュールがプロセスの P2 空間にマップされ、アプリケーションはランダム・アクセスのクエリを実行できるようになります。このマッピングから仮想アドレス空間を解放するために、このマッピングを削除するための別のライブラリ・ルーチンも作成されました。これらの新しいルーチン (LBR\$MAP_MODULE と LBR\$UNMAP_MODULE) は、ELF オブジェクト・ライブラリの処理にのみ使用できます。これらのルーチンは P2 空間を参照するため、エントリ・ポインタは 64 ビット・インタフェースです。

ELF オブジェクト・ファイルはランダムにアクセスされるものなので、次に示す操作は ELF オブジェクト・ライブラリに対しては実行できません。

LBR\$GET_RECORD

LBR\$SET_LOCATE
LBR\$SET_MOVE

ライブラリにモジュールを挿入する操作はシーケンシャル操作なので、ELF オブジェクト・ライブラリに対して LBR\$PUT_RECORD を実行することはできません。ELF オブジェクト・モジュールはレコード単位にセグメント化されていないので、モジュールをライブラリに書き込む際に、LBR\$PUT_MODULE を呼び出すとき、または LBR\$PUT_RECORD を最初に呼び出すときに、ディスク上でのモジュールのサイズを指定する必要があります。

オブジェクト・モジュールを挿入するために LBR\$PUT_RECORD を使用する方法を次の C コードの例に示します。

```
bufdesc->dsc$a_pointer = &p0_buffer ;
bytes_to_transfer = module_size ;
while ( bytes_to_transfer ) {
    transfer = MIN ( bytes_to_transfer ,
                    ELBR$C_MAXRECSIZ ) ;
    bufdesc->dsc$w_length = transfer ;
    status = lbr$put_record ( library_index ,
                             & bufdesc ,
                             & txtrfa ,
                             module_size ) ;
    if ( (status & 1) == 0 )
        break ;
    bytes_to_transfer      -= transfer ;
    bufdesc->dsc$a_pointer += transfer ;
} ;
if ( (status & 1) == 1 )
    status = lbr$put_end ( library_index ) ;
```

LBR\$PUT_RECORD を何度も呼び出さなくてよいように、新しいライブラリ・ルーチン LBR\$PUT_MODULE が作成されました。

3.5.8 『OpenVMS RTL Library (LIB\$) Manual』の訂正

V8.2-1

ここでは、Version 8.2 の『OpenVMS RTL Library (LIB\$) Manual』に対する追加と訂正を説明します。

3.5.8.1 『OpenVMS RTL Library (LIB\$) Manual』: LIB\$CVT_DX_DX の丸め規則の明確化 V8.2-1

『OpenVMS RTL Library (LIB\$) Manual』の LIB\$CVT_DX_DX ルーチンの説明では、「Guidelines for Using LIB\$CVT_DX_DX」の下にある下記の段落に丸め規則に関する具体的な説明を追加する必要があります。

結果は、常に、(切り捨てられるのではなく) 丸められます。ただし、次に述べる状況は例外です。精度や範囲が失われることは、変換先のデータ型や NBDS 変換先のサイズによっては本質的に避けられないことに注意してください。変換先のデータ型のせいで精度や範囲が失われてもエラーは表示されません。

この段落は次のように変更する必要があります。

結果は、常に、(切り捨てられるのではなく) 丸められます。ただし、変換元と変換先の両方が NBDS で、スケーリングが要求されていない場合は例外です。この場合については、後述する規則で詳しく説明されます。LIB\$CVT_DX_DX は VAX_ROUNDING 規則を使用します。精度や範囲が失われることは、変換先のデータ型や NBDS 変換先のサイズによっては本質的に避けられないことに注意してください。変換先のデータ型のせいで精度や範囲が失われてもエラーは表示されません。VAX_ROUNDING 規則の詳細は、CVT\$CONVERT_FLOAT の説明を参照してください。

3.5.9 『OpenVMS RTL Library (LIB\$) Manual』: プラットフォームの制限の明確化 V8.2-1

『OpenVMS RTL Library (LIB\$) Manual』では、以下のルーチンが Alpha と I64 の両方で利用できると説明されていますが、これは誤りです。これらのルーチンは Alpha でしか利用できません。

- LIB\$GET_CURR_INVO_CONTEXT
- LIB\$GET_INVO_CONTEXT
- LIB\$GET_INVO_HANDLE
- LIB\$GET_PREV_INVO_CONTEXT
- LIB\$GET_PREV_INVO_HANDLE
- LIB\$PUT_INVO_REGISTERS

また、『OpenVMS RTL Library (LIB\$) Manual』では、LIB\$GET_UIB_INFO ルーチンは I64 でのみ利用できることを説明する必要があります。

I64 でのみ利用できる呼び出しコンテキストと呼び出しハンドルに関連するルーチンは、次のとおりです。

- LIB\$I64_CREATE_INVO_CONTEXT
- LIB\$I64_FREE_INVO_CONTEXT
- LIB\$I64_GET_CURR_INVO_CONTEXT
- LIB\$I64_GET_CURR_INVO_HANDLE
- LIB\$I64_GET_INVO_CONTEXT
- LIB\$I64_GET_INVO_HANDLE
- LIB\$I64_GET_PREV_INVO_CONTEXT
- LIB\$I64_GET_PREV_INVO_HANDLE

これらのルーチンについての詳細は、『OpenVMS Calling Standard』を参照してください。

3.5.10 『OpenVMS システム管理者マニュアル』：IPC コマンドの制限

V8.2-1

『OpenVMS システム管理者マニュアル (上巻)』の第 9.15 節「IPC (割り込み優先順位レベル C) の使用」では、I64 とすべての Alpha で IPC コマンドを使用できると説明していますが、これは誤りです。このドキュメントは訂正され、次の文章が追加されました。

OpenVMS Versions 8.2 と 8.2-1 では、グラフィック・コンソールからブートした場合には、I64 システム、または ES47 あるいは GS1280 Alpha システムで IPC コマンドを使うことはできません。

3.5.11 『OpenVMS System Services Reference Manual』：\$PUTMSG システム・サービスの訂正

V8.2-1

Version 8.2 の『OpenVMS System Services Reference Manual』では、\$PUTMSG のプロトタイプの説明に誤りがあります。正しいプロトタイプは次のとおりです。

C プロトタイプ

```
int sys$putmsg (void *msgvec, int (*actrtn)(__unknown_params),  
void *facnam, unsigned __int64 actprm);
```

*actrtnからの戻り値は実際にチェックされ、メッセージが入力されたかどうかの判定に使用されることに注意してください。

ドキュメントのソース・ファイルが訂正され、その訂正の内容は次版の『OpenVMS System Services Reference Manual』とオンライン・ヘルプでリリースされる予定です。

3.5.12 『Volume Shadowing for OpenVMS 説明書』：メモリ要件の訂正

V8.2-1

Version 7.3-2 の『Volume Shadowing for OpenVMS 説明書』の第 1 章の第 1.3.1 項「メモリ要件」で説明されているように、OpenVMS Version 7.3 からは、Volume Shadowing for OpenVMS を実行するには追加メモリが必要になりました。以下の訂正があることに注意してください。

- 「たとえば、メンバごとに 200 GB のストレージがあるシャドウ・セットでは、クラスタ内の各々のノードの書き込みビットマップには、420 KB のメモリが必要になります。」という説明がありますが、420 KB ではなく 400 KB が正しい値です。
- ここから 3 段落先に、「たとえば、10 個のシャドウ・セットがマウントされているシステムで、各々のシャドウ・セットに 50 GB のメンバ・ディスクがある場合、追加で 1,119 KB のメモリが必要ですが、」という説明がありますが、その下に示されている計算方法に従って計算されるように、1,119 KB ではなく 1,069 KB が正しい値です。

これらは、『Volume Shadowing for OpenVMS 説明書』の次版で訂正される予定です。

3.6 Version 8.2 から Version 8.2-1 へのネットワーク・アップデートの制限

V8.2-1

OpenVMS Version 8.2-1 は、Version 8.2 から Version 8.2-1 へのネットワーク・アップデートをサポートしています。ネットワーク・アップデートは OpenVMS Version 8.2 でサポートされるシステム上のコア I/O LAN カードを使用する形態だけがサポートされています。詳細は、『HP OpenVMS Version 8.2-1 for Integrity Servers Upgrade and Installation Manual』を参照してください。

また、ネットワーク・ブートにはハードウェア構成上の制限もあります。ネットワーク・アダプタの速度や二重モードの設定をコンソールから実行できる Alpha コンソールとは異なり、Integrity サーバのコンソールとネットワーク・ブート・ドライバは自動ネゴシエーションしか実行できません。正常にネットワーク・ブートを実行するためには、Integrity サーバのブート・クライアントの最も近くにあるネットワーク・スイッチに自動ネゴシエーションを設定する必要があります。スイッチを自動ネ

ゴシエーションに設定しないと、ネットワーク・ブート・プロセスを実行することができません。

3.7 同期型データ・リンクはサポートされていない

V8.2-1

OpenVMS Version 8.2-1 では、Integrity サーバの同期型データ・リンク・ハードウェアはサポートしていません。

3.8 LAN ドライバから報告される二重モード不一致エラー

V8.3

二重モードの不一致状態は、LAN デバイスが全二重モードで動作し、ケーブルの反対側のデバイス (一般にはスイッチのポート) が半二重モードで動作している場合に起こります。また、逆の場合にも不一致となります。二重モードの不一致状態の原因となる一般的なネットワーク構成エラーは、スイッチ・ポートの速度と二重設定をオートネゴシエーションに設定し、LAN デバイスを全二重の固定設定とした場合に発生します。この構成では、スイッチによるオートネゴシエーションにより、半二重モードが選択され、LAN デバイスは全二重モードに設定されるため、二重モードの不一致が発生します。

二重モードの不一致が発生すると、通常、性能が低下します。また、オートネゴシエーション処理について記述されている IEEE 802.3 の規格では、二重モードが一致しないと、データが壊れる可能性があることが示唆されています。ほとんどの LAN デバイスでは、二重モードの不一致による唯一の影響は性能の低下です。LAN デバイスによっては、CRC が正常な状態でパケット・データが壊れ、LAN サブシステムでパケットの破壊が検出されないこともあります。そのようなデバイスとしては、Broadcom ベースのすべての NIC と、埋め込み LOM チップがあります。Alpha システムでは、DEGPA、DEGXA、AlphaServer DS25 上の BCM5703 LOM、デュアルポート BCM5704 チップを使用したすべての実装が含まれます。Integrity システムでは、A6847A、A6725A、A9782A、A9784A、AB465A、および rx2600 上の BCM5701 LOM、その他のシステム上の BCM5703 LOM、A6794A が含まれます。

以前のバージョンの OpenVMS から、LAN ドライバが二重モードの不一致状態を検出しようとします。この状態になると、1 時間に一度コンソール・メッセージとエラー・ログ・メッセージの警告を出力します。

OpenVMS Version 8.3 では、Broadcom ベースの LAN デバイスでの、メッセージの頻度が 1 時間に 1 回から 36 秒に 1 回に増えました。Broadcom 以外の LAN デバイスでは、頻度は 1 時間に 1 回のままです。また、これらのメッセージがよく見える

ように、このコンソール・メッセージが OPCOM および LANACP ログ・ファイル (SYSS\$MANAGER:LAN\$ACP.LOG) に送られます。

この注記の目的は、二重モードの不一致を避けることの重要性を強調することです。特に、この状態になると、Broadcom ベースのデバイスでは検出されないデータ破壊が発生します。

LAN ドライバは、デバイスのエラーを監視することで二重モードの不一致状態を検出することに注意してください。検出は完全ではないため、LAN ドライバはこの状態を「二重モードの不一致の可能性あり」と表します。このようなメッセージが表示されたら、システム管理者やネットワーク管理者は、LAN カウンタと LAN デバイスの設定を調べて、二重モードが不一致になっていないか確認してください。

3.9 ブートの際の AUDIT_SERVER の起動失敗

V8.3

システム管理者は、ブートの際に AUDIT_SERVER が何らかの理由で起動に失敗していないかどうか注意してください。この状態になると、起動を阻害している状態が解消されて AUDIT_SERVER が正常に起動するまで、起動プロセスが再試行ループに陥ります。この動作は意図的なもので、セキュリティが低い状態でシステムが動作しないようにするためにこのようになっています。

起動を阻害する要因には以下のものがあります。

- 監査サーバ・データベース中の不正なジャーナル・ファイルの指定
- 監査サーバ・データベース (SYSS\$MANAGER:VMS\$AUDIT_SERVER.DAT) の破壊
- オブジェクト・サーバ・データベース (SYSS\$SYSTEM:.VMS\$OBJECTS.DAT) の破壊
- ターゲット・ディスクへのアクセス失敗
- クラスタ上の他の監査サーバとの同期の失敗
- クラスタ上の他の監査サーバによる必要なリソースの占有

この状態を解消するには、手動での介入が必要な場合があります。必要な対処は、障害の原因に依存します。対処方法としては、他のクラスタ・ノード上での AUDIT_SERVER の再起動や、影響のあるノードの MINIMUM 状態でのリブート、障害への手動での対処があります。データベース・ファイルが破壊されているかどうかを確認するには、ファイル名を変更して AUDIT_SERVER を再起動します。サーバは不足しているファイルを再度作成し、その中にシステムのデフォルト・エントリを設定します。

ブート・オプションについての詳細は、『OpenVMS システム管理者マニュアル (上巻)』の第 4 章を参照してください。

システム管理に関する注意事項

この章では、システムの保守と管理、性能の管理、ネットワーキングに関連する情報をまとめます。

このバージョンで提供される新機能の詳細については、『HP OpenVMS V8.3 新機能説明書』を参照してください。

4.1 Monitor ユーティリティの変更

V8.3

OpenVMS Version 7.3-2 以降、Monitor ユーティリティ (MONITOR) にはいくつかの変更が行われています。変更の大半は、記録ファイルのフォーマットの改良と、クラス・データの追加に関するものです。これらの変更により、あるバージョンの MONITOR が収集したデータを他のバージョンで処理したときに、互換性の問題が発生します。ここでは、これらの問題について説明します。

4.1.1 MONITOR データのバージョン間での互換性

MONITOR が収集するデータの本体はリリースごとに変更される可能性があるため、あるバージョンで収集した MONITOR データを別のバージョンで必ず表示できるとは限りません。

リリース間での互換性の程度は、記録されているバイナリ・データをファイルから調べるか (つまり、プレイバック)、他のクラスタ・ノードからの生データを調べるかにより異なります。一般的に、記録されているデータをプレイバックする方法の方が、生のリモート・データを監視する方法よりも互換性が高くなります。

4.1.2 記録ファイルからのデータのプレイバック

MONITOR バイナリ・データが記録されている各ファイルは、MONITOR 記録ファイル構造レベル ID で識別されます。ファイルに対して DCL コマンド DUMP /HEADER /PAGE を実行すると、この ID を表示できます。最近のいくつかの MONITOR バージョンと、それに関連する構造レベル ID を、次の表に示します。

オペレーティング・システムのバージョン	MONITOR 記録ファイル構造 ID
OpenVMS Version 7.3-2 (修正キットあり) ¹	MON32050
OpenVMS Versions 8.2, 8.2-1 (修正キットあり) ¹	MON01060

¹これらの修正キットは、互換性を改善することのみを目的とした、推奨修正キットです。

通常、単独の MONITOR 記録ファイルをプレイバックするためには、構造レベル ID の下 2 桁が、実行中の MONITOR のバージョンの下 2 桁と一致しなければなりません。たとえば、OpenVMS Version 7.3-2 を実行している場合は、Version 7.3-2 のファイルはプレイバックできますが、Version 8.2 のファイルはプレイバックできません。

ただし、MONITOR Version 8.2 およびそれ以降は、"50"で終わる構造レベル ID の記録ファイルを読み取るように特別に作成されています。さらに、SYS\$EXAMPLES 内の MONITOR_CONVERT.C というユーティリティは、MONxx060 ファイルを MON31050 ファイルに変換します。これにより、変換後のファイルは、Version 8.2 よりも前のバージョンで読み取ることができるようになります。このプログラムの構築手順と実行手順については、MONITOR_CONVERT.C を参照してください。

ファイルのプレイバックが許可されている場合でも、ファイル内の一部の MONITOR データ・クラスが利用できないこともあります。この現象は、新しいバージョンの MONITOR で作成されたファイルを、古いバージョンの MONITOR でプレイバックする場合に発生することがあります。

最後に、いくつかの記録ファイルからマルチファイル要約を作成するときには、すべてのファイルの構造レベル ID の 8 文字すべてが一致しなければなりません。

4.1.3 VMS クラスタでの生のリモート・データの監視

V8.3

記録ファイル構造レベル ID の他に、MONITOR の各バージョンには、関連する「サーバ・バージョン番号」があります。サーバ・バージョン番号は、OpenVMS Cluster の 1 つのノードから別のノードへの生のデータを扱うことができるように、MONITOR データのバージョンを識別します。他のクラスタ・ノードのデータを監視する場合は、監視するノードと監視対象のノードのサーバ・バージョン番号が同じでなければなりません。バージョン番号が同じでない場合は、次のエラー・メッセージが表示されます。

```
%MONITOR-E-SRVMISMATCH, MONITOR server on remote node is an
incompatible version
```


最近のいくつかの MONITOR バージョンと、それに関連するサーバ・バージョン番号を、次の表に示します。

オペレーティング・システムのバージョン	MONITOR サーバ・バージョン番号
OpenVMS Version 7.3-2	5
OpenVMS Version 7.3-2 (修正キットあり) ¹	7
OpenVMS Version 8.2, 8.2-1 (修正キットあり) ¹	8

¹これらの修正キットは、互換性を改善することのみを目的とした、推奨修正キットです。

バージョンに互換性がないためにノード間での生の監視ができない場合は、ファイルへ記録してプレイバックすることにより、統計情報を表示することができます。

4.2 ファイル・セキュリティ属性の推奨値の更新

V8.3

次の表に、『HP OpenVMS システム・セキュリティ・ガイド』にリストされているファイルについて、プロテクション・プロファイルの更新された推奨値を示します。(ファイル VMSPASSWORD_HISTORY.DATA は、マニュアルの現在のバージョンには記載されていませんが、次のリビジョンでは記載されます。)

ファイル	保護
RIGHTSLIST.DAT	S:RWE,O:RWE,G,W
SYSUAF.DAT	S:RWE,O:RWE,G,W
VMSSOBJECTS.DAT	S:RWE,O:RWE,G:RE,W
VMSPASSWORD_HISTORY.DATA	S:RWE,O:RWE,G,W

ファイルの所有者は、システム範囲 (MAXSYSGROUP システム・パラメータ未満) のグループの UIC でなければなりません。推奨値は、[1,1]または[SYSTEM] (1,4) です。

4.3 システム管理についての注意事項

ここでは、システム管理およびメンテナンスへのアップデートについて説明します。

4.3.1 CIMSERVER プロセスの推奨事項 (I64 のみ)

V8.3

最適な性能を得るために、CIMSERVER プロセスを実行するアカウント (通常は SYSTEM アカウント) の PGFLQUOTA は、少なくとも 1 GB (PGFLQUOTA = 2000000) にすることをお勧めします。

この制限は、WBEM Services for OpenVMS の次のリリースで撤廃されます。

4.4 システムのハングアップまたはクラッシュからの回復 (I64 のみ)

V8.2

システムがハングアップしたため、強制的にクラッシュさせたいときは、コンソールから Ctrl/P を押します。クラッシュ・ダンプの強制方法は、XDELTA がロードされているかどうかによって異なります。

XDELTA がロードされている場合、Ctrl/P を押すと、システムが XDELTA に入ります。システムは命令ポインタと、現在の命令を表示します。次の例のように、;C を入力することで XDELTA からクラッシュを強制できます。

```
$  
Console Brk at 8068AD40  
8068AD40!      add      r16 = r24, r16 ;; (New IPL = 3)  
;C
```

XDELTA がロードされていない場合、Ctrl/P を押すと、システムがプロンプト “Crash? (Y/N)” で応答します。Y を入力すると、システムがクラッシュします。これ以外の文字を入力すると、システムでは何も起こりません。

4.5 Oracle 8i および 9i での DECdtm/XA (Alpha のみ)

V7.3-2

DECdtm/XA を使用して Oracle® 8i/9i XA 準拠リソース・マネージャ (RM) のトランザクションを調整する場合は、XA switch (xaoswd) の動的登録は使用しないでください。動的登録をサポートしている Version 9.0.1.0.0 の Oracle 共有ライブラリは、動作しません。必ず静的登録 XA switch (xaosw) を使用して、Oracle RM を DECdtm/XA Veneer にバインドしてください。

DECdtm/XA V2.1 Gateway は、クラスタ単位のトランザクション回復をサポートするようになりました。クラスタ単位の DECdtm Gateway Domain Log を使用するアプリケーションのトランザクションは、単一ノード障害から回復できるようになりました。残りのクラスタ・ノードで実行されているゲートウェイ・サーバは、障害の発生したノードの代わりに、トランザクション回復プロセスを起動できます。

4.6 デバイス・ユニットの最大数の増加

V8.2

以前のバージョンの OpenVMS では、10,000 個を超えるクローン・デバイス・ユニットを作成できませんでした。また、ユニット番号は、9999 の後は 0 に戻っていました。このことは、メールボックスや TCPIP ソケットなどの一部のデバイスにとっては、制限事項となっていました。

OpenVMS Version 7.3-2 からは、UCBSL_DEVCHAR2 の DEV\$V_NNM ビットがクリアされており、DEVICE_NAMING システム・パラメータのビット 2 がクリアされている場合、OpenVMS は最大 32,767 個のデバイスを作成します。デバイス・ドライバの変更は必要ありません。

しかし、最大のデバイス番号が 9999 であるという前提でコーディングされているプログラムやコマンド・プロシージャは、変更が必要になる場合があります。

4.7 ECP Data Collector と Performance Analyzer V5.5 (Alpha のみ)

V8.2

OpenVMS Alpha Version 8.2 およびそれ以降用の Enterprise Capacity and Performance (ECP) Analyzer の推奨バージョンは、Version 5.5 です。Version 5.5 は、OpenVMS Version 6.2 以降のバージョンと下位互換性があります。

OpenVMS Version 8.2 からは、ECP Collector は Performance Data Collector (TDC) Version 2.1 に置き換えられています。ECP Analyzer は、TDC Version 2.1 以降で生成された収集ファイルを解析できます。

OpenVMS I64 では、ECP Analyzer を現時点ではサポートしていません。

4.8 EDIT/FDL: 推奨バケット・サイズの変更

V7.3

OpenVMS Version 7.3 より前のバージョンでは、EDIT/FDL の実行時に計算されるバケット・サイズ (最大バケット・サイズは 63) が、常に最も近いディスク・クラスタのバウンダリに切り上げられていました。そのため、ディスク・クラスタ・サイズが大きい場合に、ファイルの元々のバケット・サイズは小さいが、バケット・サイズが必要以上に大きく切り上げられるという問題が発生することがありました。バケット・サイズが大きくなるほど、レコードとバケット・ロックの争奪が増加し、性能に大きく影響します。

OpenVMS Version 7.3 以降では、推奨バケット・サイズを計算するためのアルゴリズムが変更され、ディスク・クラスタが大きい場合に、より妥当なサイズが提案されます。

4.9 EFI\$CP ユーティリティ: 使用は推奨できない

V8.2

OpenVMS EFI\$CP ユーティリティは、現在ドキュメント化されておらず、サポートしないことになっています。このユーティリティは、使用しないでください。このユーティリティ内で行われている一部の特権操作により、OpenVMS I64 がブートできなくなることがあります。

4.10 EFI シェルで共用システム・ディスクまたはシャドウ・システム・ディスクを操作する場合の注意事項

V8.2-1

Integrity サーバのシステム・ディスクには、OpenVMS ブート・ローダ、EFI アプリケーション、ハードウェア診断ツールを含む FAT パーティションが最大 2 つ存在します。OpenVMS のブートストラップ・パーティションと診断パーティション (存在する場合) は、それぞれ OpenVMS システム・ディスク内の次のコンテナ・ファイルにマップされます。

```
SYS$LOADABLE_IMAGES:SYS$EFI.SYS  
SYS$MAINTENANCE:SYS$DIAGNOSTICS.SYS
```

これらの FAT パーティションの内容は、コンソールの EFI Shell>プロンプトに、*fsn:* デバイスとして現れます。これらの *fsn:* デバイスは、EFI Shell>プロンプトでのユーザ・コマンド入力、または EFI コンソール・アプリケーションや EFI 診断アプリケーションによって直接変更することができます。システム・ディスクを共用する OpenVMS 環境または EFI コンソール環境のいずれにも、パーティションの変更は通知されません。OpenVMS 環境と EFI コンソール環境は、これらのコンソールによる変更を全く意識しません。そのため、OpenVMS コンソールや使用する他の任意の EFI コンソールでは、この変更適切に連携して同期を取る必要があります。

次のどちらかまたは両方の手段で、構成内のコンソールを変更するときには注意が必要です。

- OpenVMS I64 システム・ディスクに対する OpenVMS のホスト・ベースのポリシー・シャドウイング
- システム・ディスクを共用している Integrity 環境間での、共用システム・ディスクと EFI コンソールへの同時アクセス

4.10 EFI シェルで共用システム・ディスクまたはシャドウ・システム・ディスクを操作する場合の注意事項

このような OpenVMS システム・ディスク環境は、前もって、単一メンバのホスト・ベース・ボリューム・シャドウセット、または非シャドウ・システム・ディスクに移行し、さらに、次のような操作で `fsn:` デバイスをシェル・レベルで変更するときには、Shell>プロンプトへの同時アクセスを行わないようにアクセスを調整する必要があります。

- 診断パーティション内での、診断ツールのインストールまたは操作
- パーティション内またはリムーバブル・メディアから実行する診断ツールに、OpenVMS I64 システム・ディスク上のブート・パーティションまたは診断パーティションの変更を許す
- あるいは、これらの環境の Shell>プロンプトから、ブート・パーティションまたは診断パーティションを直接的または間接的に変更

上記の予防措置をとらなかった場合には、ブート・パーティションに対応する `fsn` デバイスでの変更や、診断パーティションに対応するデバイスでの変更は直ちに、または次回の OpenVMS のホスト・ベース・ボリューム・シャドウイングのフル・マージ操作の後に、書き換えられて失われます。

たとえば、シャドウ・システム・ディスクのいずれかの物理メンバでコンテナ・ファイルの内容が EFI コンソールのシェルによって変更されても、ボリューム・シャドウイング・ソフトウェアは物理デバイスへの書き込みがあったことは認識できません。システム・ディスクが複数のメンバからなるシャドウ・セットの場合には、シャドウ・セットのメンバである他の物理デバイスのすべてに対して同じ変更を行う必要があります。そうしないと、システム・ディスクでフル・マージ操作が行われたときに、これらのファイルの内容が元に戻ってしまいます。マージ操作は、EFI での変更が行われてから数日後、または数週間後に行われることもあります。

また、シャドウ・システム・ディスクでフル・マージがアクティブになっている場合には、いずれのファイルもコンソールの EFI シェルを使って変更してはなりません。

進行中のフル・マージ操作を停止する方法や、シャドウ・セットのメンバ構成を調べる方法については、『Volume Shadowing for OpenVMS におけるホストベース・ミニマージ』を参照してください。

これらの注意事項は、ホスト・ベースのボリューム・シャドウイング用に構成されている Integrity システム・ディスク、または複数の OpenVMS I64 システムで構成されて共用されているシステム・ディスクにのみ適用されます。コントローラ・ベースの RAID を使用している構成、システム・ディスクでホスト・ベースのシャドウイングを使用していない構成、別の OpenVMS I64 システムと共用していない構成では影響を受けません。

4.11 Error Log Viewer (ELV) ユーティリティ: TRANSLATE/PAGE コマンド

V7.3-2

TRANSLATE コマンドで/PAGE 修飾子を使用してレポートを参照している際にメッセージが通知された場合、表示が乱れることがあります。この問題を回避するには、Ctrl/W を使用して、表示をリフレッシュします。

メッセージが通知された直後に Ctrl/Z を押すと、プログラムが突然終了します。この問題を回避するには、通知されたメッセージを越えてスクロールした後に Ctrl/Z を押します。

4.12 外部認証

ここでは、外部認証に関する注意事項について説明します。外部認証は OpenVMS Version 7.1 で導入されたオプションの機能であり、この機能を利用すると、OpenVMS システムは外部のユーザ ID とパスワードを使用して、指定されたユーザを認証できます。外部認証の使用方法についての詳細は、『OpenVMS システム・セキュリティ・ガイド』を参照してください。また、外部認証に関連するリリース・ノートについては、第 2.13.1 項を参照してください。

4.12.1 I64 外部認証サポート

V8.2

Advanced Server for OpenVMS V7.3A ECO4 (およびそれ以降) の製品キットには、OpenVMS Cluster 内の I64 システムに対するスタンドアロン外部認証ソフトウェアが含まれています。

I64 が動作している OpenVMS Cluster メンバのノードで NT LAN Manager の外部認証を可能にする場合は、Advanced Server がインストールされている Alpha システムから I64 メンバ・ノードへ I64 スタンドアロン外部認証イメージをコピーする必要があります。また、Advanced Server キットのリリース・ノートで説明されているように正しくセットアップされている必要があります。

4.12.2 DECterm 端末セッションでの SET PASSWORD の動作

V7.2

DECterm 端末セッションでは、ログインで使用する外部ユーザ名にアクセスすることができないため、SET PASSWORD 操作の際に外部ユーザ名を入力しなければなりません。外部ユーザ名のデフォルトは、プロセスの OpenVMS ユーザ名です。デフ

ォルトが適切でない場合 (つまり, 外部ユーザ名と, それに対応する OpenVMS ユーザ名が異なる場合), 正しい外部ユーザ名を入力しなければなりません。

次の例に, 外部ユーザ名が JOHN_DOE であるユーザが開始した SET PASSWORD 操作を示します。マッピングされた OpenVMS ユーザ名は JOHNDOE であり, これは SET PASSWORD 操作で使用されるデフォルトです。この場合, デフォルトは正しくないので, 実際の外部ユーザ名がユーザによって指定されています。

```
$ set password
External user name not known; Specify one (Y/N) [Y]? Y
External user name [JOHNDOE]: JOHN_DOE
Old password:
New password:
Verification:
%SET-I-SNDEXTAUTH, Sending password request to external authenticator
%SET-I-TRYPWDSYNCH, Attempting password synchronization
$
```

4.12.3 ワークステーションではパスワードの有効期限切れは通知されない

V7.1

LAN Manager ドメインでは, パスワードの有効期限が切れると, ログインすることはできません。

PC のユーザには, 外部ユーザ・パスワードの有効期限が間もなく切れることが通知されるので, 有効期限が切れる前にパスワードを変更できます。ところが, 外部認証を使用して OpenVMS ワークステーションからログインする場合, ログイン・プロセスは外部パスワードの有効期限が間もなく切れるかどうか判断できません。したがって, パスワードの有効期限が設定されていて, ユーザの大半が PC を使用していないサイトでは, ワークステーション・ユーザに対して外部認証を使用しない方が賢明です。

4.13 OpenVMS Cluster システム

ここでは, OpenVMS Cluster システムに関する注意事項について説明します。

4.13.1 OpenVMS I64 でのクラスタのサポート

V8.2

一部の例外を除き, OpenVMS Cluster ソフトウェアは, OpenVMS Alpha および OpenVMS VAX システムでの機能と同じ機能を, OpenVMS I64 システムでも提供します。

4.13.2 現時点での制限事項

V8.2

一時的に、以下の例外があります。

- サポートされる、I64 システムを含む運用クラスタには、VAX システムを含めることはできません。VAX システムは、開発や移行を目的とするクラスタに含めることはできますが、クラスタ内に VAX システムが存在するために問題が発生した場合は、VAX システムと I64 システムのどちらかを削除する必要があることを理解した上で使用する必要があります。詳細は、OpenVMS Cluster ソフトウェア SPD を参照してください。
- 現時点では、Open VMS Cluster システムの運用環境でサポートされるのは2つのアーキテクチャのみです。サポートされるクラスタ構成のリストは、『HP OpenVMS Version 8.2 Upgrade and Installation Manual』を参照してください。
- デバイスの命名 (ポート割り当てクラスとも言います) を使用する I64 Satellite システムは、このリリースで正しく動作させるためには追加の手順が必要です。サテライト・ブートのサーバ・ノードで、ファイル・デバイスを編集します。

```
[SYSn.SYSCOMMON.SYS$LDR]SYS$MEMORYDISK.DAT
```

*device**n*はサテライトのルートを含むディスクで、*n*はサテライト・システムのルートです。また、次の行をファイルに追加します。

```
SYS$SYSTEM:SYS$DEVICES.DAT, text
```

この場合は、ファイルの最上部にある "Do Not Edit" コメントを無視して構いません。SYS\$MEMORYDISK.DAT 内のファイルのリストは、順序に依存しません。この問題は、最終リリースで解決される予定です。

4.13.3 サテライト・ブートと LAN フェールオーバー

V8.3

Alpha および Integrity のクラスタ・サテライトでは、ネットワーク・ブート・デバイスを LAN フェールオーバー・セットのメンバ候補とすることはできません。たとえば、LAN フェールオーバー・セットを作成し、EWA および EWB で LLA を構成し、システムのブート時に LLA をアクティブにすると、LAN デバイス EWA または EWB を通してサテライトとしてシステムをブートすることはできません。

4.13.4 Integrity サーバのサテライト・システムでのエラー・ログ・ダンプ・ファイルの作成

V8.3

Integrity サーバのサテライト・システムでは、システム・クラッシュ・ダンプ・ファイルとシステム・エラー・ログ用に DOSD (Dump Off the System Disk) が必要です。AUTOGEN は、DOSD を有効にすると、適切なディスク上にシステム・ダンプを作成しますが、エラー・ログ・ダンプ・ファイル (SYS\$ERRORLOG.DMP) は作成しようとしません。システム障害が起きた場合でもエラー・ログ・エントリを保持するためには、エラー・ログ・ダンプ・ファイルを手動で作成する必要があります。

DOSD を有効にする方法については、『OpenVMS システム管理者マニュアル (下巻)』を参照してください。AUTOGEN を実行して適切なデバイス上にダンプ・ファイルを作成した後、以下のようにしてエラー・ログ・ダンプを作成します。

1. SYSGEN を使用して、システム・パラメータ ERRORLOGBUF_S2 および ERLBUFFERPAG_S2 の値を取得します。
2. $(\text{ERRORLOGBUF_S2} * \text{ERLBUFFERPAG_S2}) + 10$ を計算します。
3. SYSGEN CREATE コマンドを使用して、次のようにしてファイルを作成します。

```
SYSGEN> CREATE dev:[SYSn.SYSEX] SYS$ERRLOG.DMP/SIZE=filesize
```

ここで、各項目の意味は次のとおりです。

dev = DOSD リスト中のデバイス
n = サテライトのシステム・ルート
filesize = 手順 2 で計算した値

将来的には AUTOGEN が拡張され、この操作を行うようになる予定です。

4.14 クラスタ互換性パッチ・キット

V8.3

OpenVMS Version 8.2-1 システムを既存の OpenVMS Cluster システムに導入する場合は、事前に旧バージョンの OpenVMS で動作しているシステムにいくつかのパッチ・キット (修正キット) を適用する必要があります。これらのキットは特定のバージョンに対応していることに注意してください。

表 4-1 にリストしたバージョンは、動作が保証された (warranted) 構成でサポートされます。動作が保証された構成についての詳細は、『HP OpenVMS Version 8.2-1 for Integrity Servers Upgrade and Installation Manual』を参照してください。

表 4-1 に、パッチ・キットを必要とする機能とパッチ・キットのファイル名をリストします。各パッチ・キットには対応する readme ファイルが付属しています。readme ファイルの名前はパッチ・キットの名前にファイル拡張子.README を付加した形式になっています。

パッチ・キットは次の Web サイトからダウンロードすることができます。または、弊社のサポート担当に連絡して、ご使用中のシステムに適したメディアに格納されたパッチ・キットを入手してください。

<http://www2.itrc.hp.com/service/patch/mainPage.do>

注意

パッチ・キットは必要に応じて定期的に更新されます。必ず、機能ごとの最新のパッチ・キットを使用してください。バージョン番号はキットの readme ファイルに記載されています。各キットの最新バージョンは、Web サイトで提供されているバージョンです。

表 4-1 クラスタ互換性のために必要なパッチ・キット

機能	パッチ・キットのファイル名
OpenVMS Alpha Version 7.3-2	
ここにリストされているパッチ・キットを除く大部分のパッチ・キットを含むアップデート・キット	VMS732_UPDATE-V0600
C RTL	VMS732_ACRTL-V0100
ドライバ	VMS732_DRIVER-V0200
PCSI	VMS732_PCSI-V0100
OpenVMS Alpha Version 8.2	
	VMS82A_UPDATE-V0200
DECnet-Plus for OpenVMS Alpha ECO1	DNVOSIECO01_V82 ¹
OpenVMS I64 Version 8.2	
	VMS82I_UPDATE-V0200
¹ ご使用中の構成でこのソフトウェアを使用している場合には、このキットが必要です。	

4.14.1 クラスタの互換性のために必要なパッチ・キット

V8.2

OpenVMS Version 8.2 (またはそれ以降の) システムを既存の OpenVMS Cluster システムに導入する前に、以前のバージョンの OpenVMS を実行しているシステムに、パッチ・キット (修正キットとも呼ばれます) を適用しなければなりません。Fibre Channel, XFC, または Volume Shadowing を使用している場合は、追加のパッチ・キットも必要です。これらのキットは、各バージョンに固有のものであります。

表 4-2 に示されているバージョンは、warranted サポート構成でも、migration サポート構成でもサポートされます。これらの構成についての詳細は、『OpenVMS Cluster システム』または『HP OpenVMS V8.3 インストレーション・ガイド[翻訳版]』を参照してください。

表 4-2 に、パッチ・キットの適用が必要な機能と、そのパッチ ID 名を示します。各パッチ・キットには、同じ名前の、対応する readme ファイルがあります (ファイル拡張子は README です)。

次の Web サイトから、パッチ・キットをダウンロードしてください (「software patches」オプションの OpenVMS を選択します)。または、弊社のサポート担当者に連絡して、ご使用のシステムに合ったメディアでパッチ・キットを入手してください。

<http://h18007.www1.hp.com/support/files/index.html>

注意

パッチ・キットは、必要に応じて、定期的にアップデートされます。各機能に対する最新のパッチ・キット (キットの readme ファイルにバージョン番号が示されています) を必ず使用してください。各キットの最新バージョンは、Web サイトに掲載されているバージョンです。

表 4-2 クラスタの互換性のために必要なパッチ・キット

機能	パッチ ID
OpenVMS Alpha Version 7.3-2	
この項に示されているパッチ・キットを除く、すべてのパッチ・キットを持つアップデート・キット	VMS732_UPDATE-V0600
OpenVMS VAX Version 7.3 ¹	
Audit Server	VAXAUDS01_073
Cluster	VAXSYSL01_073
DECnet-Plus	VAX_DNVOSIECO04-V73
DECwindows Motif	VAXDWMOTMUP01_073

¹ クラスタ内で VAX システムを使用する際の運用ガイドについては、第 4.13.2 項を参照してください。

(次ページに続く)

表 4-2 (続き) クラスタの互換性のために必要なパッチ・キット

機能	パッチ ID
OpenVMS VAX Version 7.3 ¹	
DTS	VAXDTSS01_073
Files 11	VAXF11X02_073
MAIL	VAXMAIL01_073
MIME	VAXMIME01_073
MOUNT	VAXMOUN01_073
RMS	VAXRMS01_073
RPC	VAXRPC02_073
Volume Shadowing	VAXSHAD01_073
System	VAXSYS01_073

¹ クラスタ内で VAX システムを使用する際の運用ガイドについては、第 4.13.2 項を参照してください。

VAX システムは I64 システムと同じクラスタに含めることはできませんのでご注意ください。クラスタ内で保証されている組み合わせの詳細については『HP OpenVMS V8.3 インストレーション・ガイド[翻訳版]』を参照してください。

4.14.2 Fibre Channel および SCSI マルチパスと、他社製品との非互換性を修正する API

V7.3-2

システムと、SCSI デバイスまたは Fibre Channel デバイスの間に存在する複数のパス間でのフェールオーバーをサポートするマルチパス機能は、OpenVMS Alpha Version 7.2-1 で導入されました。OpenVMS Alpha Version 7.3-1 では、Fibre Channel マルチパス・テープ・デバイス間でのフェールオーバーのサポートが導入されました。

このマルチパス機能は、他社のディスク・キャッシング、ディスク・シャドウイング、または類似の機能を持つ製品との互換性がないことがあります。この機能がソフトウェアの製造元でサポートされるようになるまでは、そのようなソフトウェアを、マルチパス・フェールオーバー用に構成された SCSI デバイスまたは Fibre Channel デバイスでは使用しないでください。

OpenVMS Alpha SCSI ディスク・クラス・ドライバ (SYSSDKDRIVER.EXE)、OpenVMS Alpha SCSI テープ・クラス・ドライバ (SY\$MKDRIVER.EXE)、または SCSI 汎用クラス・ドライバ (SY\$GKDRIVER) の Driver Dispatch Table (DDT) の変更依存している他社製品で SCSI マルチパス機能が正常に動作するようにするには、製品を変更する必要があります。

このようなソフトウェアの作成者は、OpenVMS Alpha Version 7.3-2 で導入された DDT Intercept Establisher ルーチンを使用して、ソフトウェアを変更できるようになりました。これらのルーチンの詳細は、『HP OpenVMS Alpha Version 7.3-2 新機能説明書』を参照してください。

注意

他社製のディスク・キャッシュ製品や、ディスク・シャドウイング・アプリケーションを使用している場合は、アプリケーションがこれらの新しいルーチンを使用するように改訂されるまで、OpenVMS SCSI マルチパス構成や Fibre Channel マルチパス構成でこれらの製品を使用しないでください。

OpenVMS Alpha SCSI マルチパス機能と Fibre Channel マルチパス機能の詳細は、『OpenVMS Cluster 構成ガイド』を参照してください。

4.14.3 DDT Intercept Establisher ルーチンとデバイス構成通知結果

V8.3

一部のルーチンは、正しく動作するために、パッチ・キットを必要とします。必要なパッチ・キットを使用せずにこれらのルーチンを使用すると、システムがハングまたはクラッシュしたり、データが破壊されるおそれがあります。また、弊社ではこのような使用はサポートしていません。

これらのルーチンについての詳細は、『HP OpenVMS Alpha Version 7.3-2 新機能説明書』を参照してください。

4.14.4 CI と LAN との間の回線切り替えによるクラスタの性能の低下

V7.3-1

CI と、複数の FDDI、100 Mb/s または Gb/s のイーサネット・ベースの CIRCUIT の両方を含む OpenVMS Cluster 構成では、SCS 接続が CI 回線と LAN 回線の間を約 1 分単位で移動することがまれにあります。この頻繁な回線の切り替えが原因で、クラスタの性能が低下したり、シャドウ・セット・メンバのマウント確認が行われる場合があります。

PEdriver では、数秒間継続している LAN 輻輳を検出し、対処することができます。LAN パスでの遅延時間の大幅な増加やパケットの損失が検出されると、PEdriver はそのパスを使用しなくなります。パスの性能が回復したことが確認されると、そのパスを再度使用するようになります。

限界条件下では、LAN パスにクラスタ・トラフィックで使用する負荷が追加されると、遅延やパケットの損失が容認できる限界を超える場合があります。クラスタの負

荷が取り除かれると、パスの性能は再度使用できる状態まで回復できる場合があります。

LAN 回線の負荷クラスに限界 LAN パスを割り当てると、その回線の負荷クラスが増加して CI の負荷クラス値 140 を超えて限界パスが対象となる場合 (また、反対に LAN 回線の負荷クラスが減少して 140 を下回り限界パスが除外される場合) に、SCS 接続は CI 回線と LAN 回線の間を移動します。

LAN 回線と CI 回線間の接続の移動を確認するには、CONNECTION クラスと CIRCUITS クラスを追加した SHOW CLUSTER を使用します。

回避方法

接続の移動が頻繁に行われている場合は、次のいずれかの回避方法を使用してください。

- SCACP または Availability Manager を使用して、使用する回線またはポートにより高い優先順位を割り当て、自動的な接続割り当てと移動を無効にします。

SCACP コマンドの例を次に示します。

```
$ MC SCACP
SCACP> SET PORT PNA0 /PRIORITY=2      ! This will cause circuits from local
                                         ! CI port PNA0 to be chosen over
                                         ! lower priority circuits.

SCACP> SET PORT PEA0 /PRIORITY=2      ! This will cause LAN circuits to be
                                         ! chosen over lower priority circuits.
```

- SCACP SHOW CHANNEL コマンドを使用して、使用の切り替えが行われているチャンネルを確認します。次に、SCACP を使用して、特定のチャンネルに目的のチャンネルよりも低い値を割り当てて、そのチャンネルを明示的に除外することもできます。たとえば、次のように指定します。

```
SCACP> SET CHANNEL LARRY /LOCAL=EWB/REMOTE=EWB /PRIORITY=-2
```

max, max-1 の範囲内にある CHANNEL および LAN デバイスの優先順位値は等価とみなされます。つまり、この両方のデバイスに、最大優先順位値が指定されているものとみなされます。チャンネルまたは LAN デバイスを使用対象から外す場合は、優先順位値に 2 以上の差をつける必要があります。

4.14.5 マルチパス・テープ・フェールオーバーの制限事項

V7.3-1

Fibre Channel マルチパス・テープ・セット内の 1 つのデバイスで INITIALIZE コマンドを実行している間は、そのセットの別のメンバへマルチパス・フェールオーバーを実行できません。別のマルチパス・テープ・デバイスが初期化されている間に、現在のパスで障害が発生した場合は、テープ・デバイスが機能しているパスへフェールオーバーした後に、INITIALIZE コマンドを再試行してください。

この制限は、今後のリリースで無くなる予定です。

4.14.6 SCSI マルチパス媒体チェンジャでは自動フェールオーバーは行われない

V7.3-1

Fibre と SCSI 間のテープ・ブリッジを使用して Fibre Channel に接続されている SCSI 媒体チェンジャ (テープ・ロボット) 向けの OpenVMS Alpha Version 7.3-1 以降には、パスの自動切り替えが実装されていません。そのようなデバイスに対しては複数のパスを構成できますが、別のパスに切り替える場合は、SET DEVICE /SWITCH コマンドを使用してパスの手動切り替えを使用する方法しかありません。

この制限は、今後のリリースで無くなる予定です。

4.14.7 Availability Manager AVAIL_MAN_BASE キットがロック競合情報の収集に失敗する

OpenVMS Version 8.3 をインストールすると、AVAIL_MAN_BASE SIP キットもインストールされます。さらにこのキットは、Availability Manager Data Collector をインストールします。

AVAIL_MAN_BASE SIP キットの Data Collector には、ロック競合データを正しく収集しない原因となる不具合があります。この問題を修正するには、AVAIL_MAN_COL Version 2.6 キットをインストールしてください。このキットは、オペレーティング・システムのアップデート媒体で提供されています。

4.15 OpenVMS Galaxy (Alpha のみ)

ここでは、OpenVMS Galaxy システムに関する注意事項について説明します。

OpenVMS Galaxy がサポートされているのは、OpenVMS Alpha システムだけです。

4.15.1 Galaxy の定義

V8.2

『OpenVMS Alpha パーティショニングおよび Galaxy ガイド』は、このリリースではアップデートされていません。ここでは、前後関係により異なる、Galaxy という言葉の定義を詳しく説明します。

表 4-3 Galaxy の定義

Galaxy が使われる 文脈	説明
ライセンス	1 台のコンピュータで OpenVMS の複数のインスタンスを作成および実行するために必要です。このライセンスがないと、1 台のコンピュータで実行できる OpenVMS のインスタンスは 1 つだけです。
システム・パラメータ	メモリ共用を設定します。GALAXY に 1 を設定すると、ハード・パーティションにそのパラメータが設定されている OpenVMS インスタンスで、ハード・パーティション内の複数のソフト・パーティション間でメモリを共用することを指定します (1 つのハード・パーティション内で 3 つ以上のソフト・パーティションを実行でき、そのすべてではメモリを共用したくない場合があります)。このパラメータでは、ノードで共用メモリを使用するかどうかだけを指定します。連携して動作する OpenVMS の複数のインスタンスを実行するためにこのパラメータを使用する必要はありません。メモリの共用のための設定は、コンソールで希望の構成ツリーを設定することで実現できます。GALAXY に 0 を設定すると、メモリは共用されません (デフォルト)。
ソフト・パーティション	CPU の移動、API の使用、メモリの共用などが可能なように、複数の OpenVMS インスタンスを 1 台のコンピュータ内で連携させて動作させることができるようにします。プラットフォームのパーティショニング機能により、リソースを複数のソフト・パーティションに分けることができ、それぞれのパーティションで OS のインスタンスを実行することができます。ソフト・パーティションは、その中で実行されている OS インスタンスが参照および使用することができる、リソースのサブセットです。

4.16 セル型システムでの複数の nPartitions

V8.2-1

HP Integrity rx7620, HP Integrity rx8620, HP Integrity Superdome などのサーバに複数の nPartitions を構成し、マルチ・オペレーティング・システム環境で動作させ、そのうち 1 つの nPartitions で OpenVMS を動作させる場合、OpenVMS のブート時に他のオペレーティング・システムのいずれかが SEL (システム・イベント・ログ) にエラーまたはイベントを記録することがあります。OpenVMS は FRU (Field Replaceable Unit) テーブルの生成が完了するまで SEL を保持し続けるため、それが原因で、他のオペレーティング・システムがエラーまたはイベントを記録することがあります。

4.16.1 OpenVMS Graphical Configuration Manager

V8.2

AlphaServer ES47/ES80/GS1280 の Galaxy 構成で OpenVMS Graphical Configuration Manager (GCM) がサポートされるようになりました。以前は、Graphical Configuration Utility (GCU) だけがサポートされていました。

4.16.2 ES40 上の Galaxy: 非圧縮ダンプの制限事項

永続的な制限事項

AlphaServer ES40 Galaxy システムでは、インスタンス 1 のメモリが 4 GB (物理) 以上から始まっている場合、インスタンス 1 から raw (非圧縮) ダンプを書き出すことはできません。代わりに、圧縮ダンプを書き出さなければなりません。

4.16.3 ES40 上の Galaxy: Fast Path の無効化

永続的な制限事項

AlphaServer ES40 システムで Galaxy を使用する場合、インスタンス 1 で Fast Path を無効化する必要があります。そのためには、そのインスタンスで SYSGEN パラメータ FAST_PATH を 0 に設定します。

インスタンス 1 で Fast Path を無効化しないと、インスタンス 0 のリブート時にインスタンス 1 での入出力がハングします。この状態は、PCI バスをリセットし、インスタンス 1 をリブートするまで続きます。共有する SCSI または Fibre Channel がある場合、共有ノードでの入出力がハングし、これらのデバイスへのすべてのパスが無効になります。

4.17 OpenVMS Management Station

V8.2

OpenVMS I64 Version 8.2 およびそれ以降と、OpenVMS Alpha Version 8.2 およびそれ以降での OpenVMS Management Station の推奨バージョンは、Version 3.2D です。ただし、OpenVMS Management Station は、OpenVMS Version 6.2 以降に対して下位互換性を維持しています。

OpenVMS をインストールすると OpenVMS Management Station Version 3.2D がインストールされます。

4.18 OpenVMS Registry は Version 2 フォーマットのデータベースを壊すことがある

V7.3-2

キー・ツリーに揮発性のサブキーを 8 個以上作成して、スタンドアロン・システムやクラスタをリブートした場合、リブート後にサーバが起動すると、OpenVMS Registry サーバは、Version 2 フォーマットの Registry データベースを壊すことがあります。

この問題を回避するには、以下のいずれかを実行します。

- 揮発性のキーを使用しない。
- Version 1 フォーマットのデータベースを使用する。

Advanced Server for OpenVMS と COM for OpenVMS は、揮発性のキーを作成しません。

4.19 システム・パラメータ

ここでは、システム・パラメータに関する注意事項について説明します。

4.19.1 新しいシステム・パラメータ

V8.3

新しいシステム・パラメータについては、『HP OpenVMS V8.3 新機能説明書』を参照してください。

4.19.2 廃止されたシステム・パラメータ

V8.3

以下のシステム・パラメータは、OpenVMS Version 8.3 で「廃止」扱いになりました。

- SMP_CPUS
- SMP_CPUSH
- IO_PREFER_CPU
- IO_PREFER_CPUS
- NPAG_AGGRESSIVE
- NPAG_GENTLE
- SCH_CTLFLAGS
- SHADOW_REC_DLY
- TTY_SILOTIME
- BALSETCNT
- BREAKPOINTS
- MMG_CTLFLAGS
- MULTITHREAD

- NISCS_MAX_PKTSZ
- NISCS_PORT_SERV
- SECURITY POLICY
- SHADOW_HBMM_RTC
- SHADOW_PSM_RDLY
- SHADOW_SYS_DISK
- WBM_MSG_UPPER

以下の新しいパラメータで、それぞれ該当する以前のパラメータが置き換えられています。

- SMP_CPU_BITMAP
- IO_PRCPU_BITMAP

これらの新しいシステム・パラメータについての詳細は、『OpenVMS システム管理ユーティリティ・リファレンス・マニュアル』またはオンライン・ヘルプを参照してください。

4.19.3 システム・パラメータの変更

V8.3

OpenVMS Version 8.3 では、以下のシステム・パラメータが変更されました。詳細は、『OpenVMS システム管理ユーティリティ・リファレンス・マニュアル』を参照してください。

- BALSETCNT - 説明が変更されました。
- BREAKPOINTS - 動的になりました。
- MMG_CTLFLAGS - 追加のビットが定義され、言い回しに変更されました。
- MULTITHREAD - I64 のサポートが追加されました。
- NISCS_MAX_PKTSZ - 言い回しに変更されました。
- NISCS_PORT_SERV - ビット定義が変更されました。
- SECURITY POLICY - ビット 13 および 14 が定義されました。
- SHADOW_HBMM_RTC - 言い回しに変更されました。
- SHADOW_PSM_RDLY - つづりが訂正されました (元は SHADOW_PSM_DLY)。
- SHADOW_SYS_DISK - 言い回しに変更されました。
- WBM_MSG_UPPER - デフォルトが変更されました。

これらのパラメータについての詳細は、オンライン・ヘルプまたは『OpenVMS システム管理ユーティリティ・リファレンス・マニュアル』を参照してください。

4.19.4 ドキュメントの誤り: LCKMGR_CPUID システム・パラメータ

V8.3

『OpenVMS Performance Management』では、複数の個所で LCKMGR_CPUID システム・パラメータが LOCKMGR_CPU と記載されています。後者のつづりは誤りで、このマニュアルが次回アップデートされるときに訂正されます。

4.19.5 MMG_CTLFLAGS: ドキュメントの誤り

V8.2

『OpenVMS Performance Management』の、MMG_CTLFLAGS システム・パラメータのビット 1 の記述に誤りがあります。正しい記述は、次のとおりです。

「LONGWAIT 秒より長い時間アイドルであったプロセスを外部にスワップすることによる再生が有効になります。これは、空きリストのサイズが FREEGOAL の値より小さくなったときに発生します。」

4.20 Terminal Fallback Facility (TFF)

V8.2

OpenVMS Alpha システムの Terminal Fallback Facility (TFF) には、フォールバック・ドライバ (SYS\$FBDriver.EXE)、共有イメージ (TFFSHR.EXE)、Terminal Fallback ユーティリティ (TFU.EXE)、フォールバック・テーブル・ライブラリ (TFF\$MASTER.DAT) が含まれます。

注意

ユーザが呼び出し可能なインタフェースとして公開されているものではないため、TFFSHR は IMAGELIB から削除されています。ただし、イメージは現在でも SYS\$LIBRARY: ディレクトリにあります。

TFF を起動するには、次のように SYS\$MANAGER にある TFF スタートアップ・コマンド・プロシージャを起動します。

```
$ @SYS$MANAGER:TFF$SYSTARTUP.COM
```

フォールバックを有効にしたり、フォールバック属性を変更するには、次のように Terminal Fallback ユーティリティ (TFU) を起動します。

```
$ RUN SYS$SYSTEM:TFU
TFU>
```

端末に対するデフォルトのフォールバックを有効にするには、次の DCL コマンドを入力します。

```
$ SET TERMINAL/FALLBACK
```

OpenVMS Alpha TFF は、次の点で OpenVMS VAX TFF と異なります。

- Alpha システムでは、TFF フォールバック・ドライバの名前は SYSS\$FBDRIVER.EXE です。VAX システムでは、TFF フォールバック・ドライバの名前は FBDRIVER.EXE です。
- Alpha システムでは、TFF は 16 ビット文字のフォールバックを処理できます。OpenVMS Alpha フォールバック・テーブル・ライブラリ (TFF\$MASTER.DAT) には、VAX ライブラリより 4 つ多い 16 ビット文字テーブルが含まれます。表 4-4 に、これらの追加テーブルを示します。

表 4-4 TFF 文字フォールバック・テーブル

テーブル名	ベース	説明
BIG5_HANYU	BIG5	BIG5 を CNS 11643 (SICGCC) 端末/プリンタへ
HANYU_BIG5	CNS	CNS 11643 (SICGCC) を BIG5 端末/プリンタへ
HANYU_TELEX	CNS	CNS 11643 を MITAC TELEX-CODE 端末へ
HANGUL_DS	KS	KS を DOOSAN 200 端末へ

これらのテーブルは主にアジア地域で使用されます。また、このテーブルの形式は、16 ビット文字フォールバックのサポートのために変更されています。

- Alpha システムでは、TFU コマンド SHOW STATISTICS はフォールバック・ドライバ (SYSS\$FBDRIVER.EXE) のサイズを表示しません。

RT 端末は TFF ではサポートされません。

Terminal Fallback Facility の詳細については、OpenVMS ドキュメント Web サイトで、アーカイブ扱いになった『OpenVMS Terminal Fallback Utility Manual』を参照してください。

<http://www.hp.com/go/openvms/doc>

このマニュアルを参照するには、左側のサイドバーに表示されている“Archived documents”をクリックしてください。

4.21 User Environment Test Package (UETP) (I64 のみ)

V8.2

User Environment Test Package (UETP) を使用する際には、以下の点に注意が必要です。

- ロード・フェーズ中に、UETMEMORY01 でアクセス・バイオレーションが時折発生します。これによって、実行が終了したり、実行結果の有効性に影響を与えることはありません。UETP は、それでも利用可能で、正しい結果を出力します。
- デバイス・フェーズは現在、アクセス・バイオレーションが発生して実行が完了しません。
- DECnet フェーズは問題なく動作します。クラスタ・フェーズは、まだテスト中です。このフェーズは問題なく動作しているように見えますが、何点か注意事項があります。また、他のシステムの名前が正しく出力されません。

4.22 推奨するキャッシュ方式

永続的な制限事項

Virtual I/O キャッシュ (VIOC) (別名 VAX Cluster キャッシュ (VCC)) は、OpenVMS I64 では利用できません。I64 システムでは、SYSGEN パラメータの VCC_FLAGS に 1 を設定しても、0 を設定したのと同じになります。したがって、キャッシュはロードされません。

Alpha システムと I64 システムの両方でお勧めできるキャッシュ方式は、拡張ファイル・キャッシュ (XFC) です。XFC の詳細は『OpenVMS システム管理者マニュアル』を参照してください。

OpenVMS Alpha の将来のリリースでは、VIOC はサポートされなくなります。

4.23 Volume Shadowing for OpenVMS

ここでは、HP Volume Shadowing for OpenVMS (ホスト・ベース・ボリューム・シャドウイング (HBVS) とも言います) に関する注意事項について説明します。

4.23.1 デバイス名の必要条件

V7.3-2

Volume Shadowing for OpenVMS は、完全デバイス名 \$allocclass\$ddcu: の ddc 部分が 3 文字のデバイス名をサポートしています。

このリリースより前では、完全デバイス名の ddc 部分が長いデバイス名 (たとえば、\$1\$DECRAM10:) を作成でき、これらのデバイスを正しくマウントできました。ただし、このようなデバイスをシャドウ・セットの一部としてマウントする

と、操作上の問題 (たとえば、他のディスクをシャドウ・セットに追加した場合の%MOUNT-F-XSMBRS エラー) が発生します。

OpenVMS Alpha Version 7.3-2 からは、Mount ユーティリティは、デバイスを最初にマウントする際に、完全デバイス名のddc 部分に対し 3 文字という必要条件を強制するようになりました。この必要条件を満たしていない名前のデバイスをマウントしようとすると、次のエラー・メッセージが表示されます。

```
MOUNT-F-NOTSHDWDEV, not a valid shadow set member
```

4.23.2 DCL コマンド・プロシージャ内での SET SHADOW と SHOW SHADOW の使用についての警告

V7.3-2

新しい DCL コマンド SET SHADOW および SHOW SHADOW は、今後も進化します。将来のリリースでは、SHOW SHADOW/FULL 表示のデフォルトの表示および実装により、現在の表示フォーマットが変更されます。このため、DCL コマンド・プロシージャ内で現在の出力フォーマットを解析することでシャドウ・セットの情報を取得することは避けるようにお勧めします。SHOW SHADOW コマンドが表示する項目の大部分は、FSGETDVI レキシカル関数を使用して取得することができます。

さらに、SET SHADOW コマンドの動作も変更されます。SET SHADOW を使用してシステム上のすべてのシャドウ・セットの特性を同時に設定する場合は、他の新しい修飾子に加え、新しい/ALL 修飾子が必要です。

これらの新しいコマンドを使用する DCL コマンド・プロシージャを作成する場合は、これらの変更を念頭に置いてください。

4.23.3 書き込みビットマップと異種デバイス・シャドウイング (DDS) の注意事項

V7.3-2

Volume Shadowing for OpenVMS を使用するときには、書き込みビットマップと異種デバイス・シャドウイング (DDS) の間に相互作用があります。

DDS (OpenVMS Version 7.3-2 の新機能) を使用すると、異なるサイズのディスク・デバイスからなるシャドウ・セットを構築できます。(DDS についての詳細は、『HP OpenVMS Alpha Version 7.3-2 新機能説明書』と『Volume Shadowing for OpenVMS 説明書』を参照してください。)

書き込みビットマップは、完全コピーのオーバーヘッドなしでメンバを仮想ユニットに戻せるように、シャドウ・セットの仮想ユニットに対して行われたアプリケーションの書き込みを追跡します。ユーザがシャドウ・セット・メンバに対して DISMOUNT

/POLICY=MINICOPY コマンドを実行した場合や、MOUNT/POLICY=MINICOPY コマンドを使用してシャドウ・セットをマウントした場合に、書き込みビットマップが作成されます。このビットマップが作成されるときサイズは、ボリュームの現在のサイズに依存します。

シャドウ・セットがマウントされるとき、そのシャドウ・セットの仮想ユニットの論理サイズは、最小のメンバ・ユニットのサイズになります。シャドウ・セットのメンバが削除された場合、仮想ユニットの論理サイズは、セット内に残っているメンバのサイズを基にして、再計算されます。その結果、仮想ユニットの論理サイズは、大きくなる場合があります。

シャドウ・セットに書き込みビットマップが作成されるとき、そのサイズは、シャドウ・セットの仮想ユニットの現在のサイズによって決まります。仮想ユニットのサイズが後で大きくなると、ビットマップは仮想ユニット全体をカバーできなくなります。その後、ビットマップを使用してミニコピー操作でシャドウ・セット・メンバを戻すと、仮想ユニット内でビットマップがカバーしていない部分は、フル・コピー操作でコピーされます。

この問題を、次の例で示します。

- シャドウ・セット DSA1: は、次の 3 つのメンバからなります。

\$1SDGA20: (18 GB)

\$1SDGA21: (36 GB)

\$1SDGA22: (36 GB)

- 次のコマンドを使用して、ミニコピー・ビットマップ付きで、シャドウ・セットから \$1SDGA22: を削除します。

```
$ DISMOUNT/POLICY=MINICOPY $1SDGA22:
```

書き込みビットマップのサイズは、シャドウ・セットの仮想ユニットの現在のサイズである、18 GB を基にして決められます。

- \$1SDGA20: をシャドウ・セットから削除します。ファイル・システムで残りのメンバの 36 GB 全体を利用できるようにするには、次のコマンドを使用します。

```
$ SET VOLUME/SIZE DSA1
```

\$1SDGA20 は、新しいボリューム・サイズよりも小さいため、このシャドウ・セットでは使用できなくなります。

- 次のコマンドを使用して、\$1SDGA22: をシャドウ・セットに戻します。

```
$ MOUNT/SYSTEM DSA1:/SHADOW=$1SDGA22: label
```

DSA1: の論理サイズは 36 GB のままですが、ビットマップがカバーしているのは、最初の 18 GB だけです。

- \$1SDGA22: の最初の 18 GB はビットマップを使用してミニコピーでコピーされ、残りの 18 GB は、フル・コピー操作でコピーされます。

小さいシャドウ・セット・メンバの削除を予定している場合は、ミニコピー・ビットマップ付きで大きなシャドウ・セット・メンバを削除する前に小さいメンバを削除すれば、大きなビットマップが作成され、短いビットマップで性能へ悪影響を及ぼすのを避けることができます。(上記の例では、\$1SDGA22: を削除する前に\$1SDGA20: を削除します。)

4.23.4 KZPDC (Smart Array 5300) の制限事項

永続的な制限事項

すべてのシャドウ・セット・メンバが、次のようなフォールト・トレラント・デバイスからなるデバイスを使用して形成されているという前提で、Volume Shadowing for OpenVMS は、KZPDC コントローラ (Smart Array 5300) とともに使用できません。

- RAID 1。コントローラ・ベース・ミラーリングとも呼ばれます。
- RAID 5。パリティ付きのストライピングです。
- RAID ADG (Advanced Data Guarding)。複数のパリティ・デバイスを用いたストライピングです。

KZPDC (Smart Array 5300) コントローラでのフォールト・トレラント・デバイスは、基礎となる LUN のいずれかでメディア障害が発生した場合に、データ・エラーを修復できるデバイスです。

OpenVMS Alpha Version 7.3-2 およびそれ以降は、総ブロック数が異なるメンバからなるシャドウ・セットをサポートしています。この新しい機能は、異種デバイス・シャドウイング (DDS) と言います。DDS を使用すると、KZPDC デバイスは、サポートされている任意のコントローラ下のデバイスとシャドウ化できます。

以前のすべてのバージョンの OpenVMS では、HBVS で複数メンバのシャドウ・セットを作成するためには、すべてのデバイスは、総ブロック数が同じでなければなりません。構成ユーティリティは、KZPDC デバイスや MSA1000 デバイス上のブロックの総数として、作成可能で、要求されたサイズに最も近い値を設定します。KZPDC と MSA1000 は同じ計算方法を使用するため、同じ要求サイズで両者に作成されたデバイスには、同じサイズが設定されます。これにより、HBVS で複数メンバのシャドウ・セットを作成できます。

注意

フォールト・トレラント・デバイスが使用されていない場合は、HBVS を使用して複数メンバのシャドウ・セットを作成できないこともあります。たとえば、単一メンバのシャドウ・セットが、1つのデバイス (物理ディスクまたは非フォールト・トレラント・デバイス) を使用して形成されたとします。その後このデバイスで回復不可能なデータ・エラーが発生した場合、HBVS を使用してこのシャドウ・セットに他のメンバを追加することはできません。2つ目のメンバがシャドウ・セットに追加されると、HBVS はソース・デバイス全体を読み取り、ターゲット・デバイスへコピーします。基本シャドウ・

セット・メンバ(つまりソース・シャドウ・セット・メンバ)からデータ・エラーが読み取られると、HBVSは、現在のすべてのシャドウ・セット・メンバ(ソース・メンバとコピー・ターゲット)に、「バッド・スポット」を作成しようとします。いずれかのシャドウ・セット・メンバで、バッド・スポットを作成する要求が失敗すると、シャドウ・セットは、1メンバに縮小されます。

4.23.5 シャドウ・セット・マージ遅延の算出方法の変更

V7.3-2

シャドウ・セットの補助なしマージ操作の実行中には、アプリケーションの読み込み I/O の性能が、次の 2 つの理由で低下します。

- すべての読み込み I/O で、データー貫性チェックを行う必要がある。
- シャドウ・セット・マージ操作により、I/O 帯域幅の競合が発生する。

シャドウ・セット・マージ操作では、アプリケーションへのマージ I/O の影響を少なくするために、絞り込みメカニズムを採用しています。マージ処理は、システムの負荷が検出されたときにマージ I/O 操作間に遅延を入れることにより、絞り込まれます。この遅延を算出するロジックが、OpenVMS Alpha Version 7.3-2 用に再設計されました。新しいマージ遅延の計算では、デフォルトのパラメータ設定で、HSG80 などの一部の I/O コントローラでのマージ速度が速くなります。詳細は、『Volume Shadowing for OpenVMS 説明書』を参照してください。

4.23.6 /MINICOPY を使用したシャドウ・セット・メンバのディスマウント

V7.3

シングル・サイトまたはマルチ・サイトの OpenVMS Cluster 構成では、クライアント・システムでシャドウ・セットのメンバのディスマウントに、/MINICOPY 修飾子を使用して DISMOUNT コマンドを実行すると、DISMOUNT コマンドが失敗することがあります。

回避方法

最初の DISMOUNT コマンドが失敗した場合、次の例のようにコマンドを繰り返します。

```
$! The following commands are NOT executed on the WILD3 system.
$
$ SHOW DEVICE DSA5555
Device              Device      Error   Volume      Free  Trans Mnt
Name               Status      Count   Label        Blocks Count Cnt
DSA5555:           Mounted      0   $80$DKA107:  7994646      1  18
$80$DKA107:        (WILD3)    ShadowSetMember  0 (member of DSA5555:)
$80$DKA302:        (WILD3)    ShadowSetMember  0 (member of DSA5555:)
$80$DKA303:        (WILD3)    ShadowSetMember  0 (member of DSA5555:)
$
$
$ DISMOUNT/POLICY=MINICOPY $80$DKA302:
%DISM-W-CANNOTDMT, $80$DKA302: cannot be dismounted
%DISM-F-SRCMEM, only source member of shadow set cannot be dismounted
$
$
$ DISMOUNT/POLICY=MINICOPY $80$DKA302:
$
```

この問題は、今後のリリースで修正される予定です。

4.24 Authentication and Credential Management Extension (ACME)

4.24.1 新しい機能と変更された機能

V8.3

以下に Version 8.3 での新機能と変更された機能を示します。

- ACME_SERVER の再起動の変更

以前のリリースでは、ACME_SERVER プロセスはすべての構成コマンドを、ステージング・ファイル SYSS\$SYSTEM:ACME\$SERVER_CONFIG.TMP に記録していました。再起動に際し、サーバはこのファイルを使用して、再起動時に処理されるステージング・ファイルから選択された構成情報を含む、データ・ファイル SYSS\$SYSTEM:ACME\$SERVER_RESTART.DAT を作成します。

再起動後にサーバの構成状態を復元する方法は使用されなくなり、サイト固有のスタートアップ・プロシージャ SYSS\$MANAGER:ACME\$START.COM に置き換えられました。このファイルの検索には、エグゼクティブ・モードの論理名 ACME\$START が使用されます。

SYSS\$MANAGER:ACME\$START.COM は、次のいずれかの条件の結果として実行されます。

- SET SERVER ACME/START=AUTO コマンドが実行された。
- SET SERVER ACME/RESTART コマンドが実行された。
- 自動サーバ再起動が行われる、予期しない状態となった。

SYSSMANAGER:ACMES\$START.LOG ファイルには、障害時の再起動で生成された情報が含まれます。

再起動時に構成されるエージェントを定義したり、他のサーバ構成オプションを定義するために、ユーザは SYSSMANAGER:ACMES\$START.COM ファイルを変更することができます。システムのアップグレード時には、このファイルは置き換えられません。このファイルの、弊社が提供しているバージョンは、SYSSMANAGER:ACMES\$START.TEMPLATE ファイルに入っています。

- 新しい SET SERVER ACME/START[=AUTO] キーワード
オプションのキーワード AUTO を使用すると、サーバは SYSSMANAGER:ACMES\$START.COM プロシージャを使用して、自身の起動および構成を行います。デフォルトでは、サーバは OpenVMS ACME エージェントだけが構成されている状態で起動されます。
- Integrity サーバ上での SET SERVER ACME/CONFIG=THREAD_MAX の無視
このリリースの Integrity サーバでは、SET SERVER ACME /CONFIG=THREAD_MAX コマンドは無視されます。アクティブなワーカ・スレッドは、1 個だけです。
- SYSSACM 対応の LOGINOUT.EXE イメージと SETP0.EXE イメージのためのオプション・キット
SYSSACM 対応の LOGINOUT.EXE イメージと SETP0.EXE (SET PASSWORD) イメージは、オプションのインストール・キットとして提供されています。OpenVMS 以外の ACME エージェントとともに使用すると、これらのイメージは、追加の認証機能を必要とするサイトに対して追加の認証機能を提供します。

このキットの動作上の要件とサポートされる ACME エージェントの一覧については、SYSSHLP:ACME_DEV_README.TXT を参照してください。

4.25 OpenVMS と Kerberos ACME エージェントのデフォルトの起動順序は必須

V8.3

OpenVMS ACME エージェントを起動するためのデフォルトの設定では、正常なログインを許可します。OpenVMS と Kerberos ACME エージェントのデフォルトの起動順序を変更し、Kerberos ACME を先に起動するようにすると、Kerberos レルム内にはないアカウントでログインできなくなる可能性があります。デフォルトの起動順序の変更は、OpenVMS の本リリースではサポートされていません。

起動順序が変更されている場合は、以下の手順を実行することでデフォルトの順序に戻すことができます。

SYSSMANAGER:ACME\$START.COM を編集し、コマンド・プロシージャの最後の付近にある、エージェントの順序を指定できるセクションを探します。AGENT_LIST で始まる最後の行を変更し、プロシージャ内に現れるようにします。

```
$! A specific agent ordering can be specified in AGENT_LIST.
$!
$! If the list is empty, the agents will be enabled in the order that
$! they were configured. Some agent startup procedures may alter
$! the agent order. You can override that ordering here. Consult the
$! agent documentation you are using to ensure that the ordering you
$! specify is supported by that agent.
$!
$! For example
$!
$!     AGENT_LIST = "VMS,MSV1_0"
$!
$! will enable the VMS and MSV1_0 agents (and only those agents) in
$! that order.
$!
$ AGENT_LIST = "VMS,ACME_KRB_DOI"
```

4.26 トレースバック API の問題の修正

OpenVMS I64 Version 8.2 では、pc_rel (相対 PC 値) または image_desc (イメージ名文字列記述子) の引数にゼロを指定すると、エラーが発生しました。本リリースでは、トレースバック・ファシリティはこれらの引数を正しく無視し、トレースバック処理が継続されるようになりました。

4.27 WBEM Services for OpenVMS Version 2.0 の注意事項

以下に、WBEM Services for OpenVMS Version 2.0 に関する最新情報を示します。

4.27.1 ベースは OpenPegasus 2.5

WBEM Services for OpenVMS Version 2.0 は、The Open Group の Pegasus オープン・ソース・プロジェクトの OpenPegasus 2.5 コード・ストリームをベースにしています。

4.27.2 nPartitions と iCAP のサポート

WBEM Services for OpenVMS Version 2.0 は、ローカルの nPartitions プロバイダと iCAP プロバイダをサポートしています。これらのプロバイダが必要とする機能だけがサポートされています。

4.27.3 共通システム・ディスクのクラスタ向けに設計されていない

UNIX ベースの製品である Pegasus は、共通システム・ディスクのクラスタ向けに設計されていません。1 つのクラスタ共通システム・ディスク上で複数のノードに対して cimserver を実行するには、リポジトリ、トレース、ログ・ファイル・ディレクトリが衝突しないように、製品を個別のルートにインストールしてください。

4.27.4 OpenVMS 上でプロバイダをアンロードするには cimserver.exe を再起動する

cimprovider -r コマンドを入力した後は、プロバイダの置き換え処理を完了させるために、cimserver を停止して再起動する必要があります。(OpenVMS では、動的にロードされたイメージのアンロードはサポートされていません。)

4.27.5 コマンド行オプションは引用符で囲む

コマンド行オプションの大文字と小文字の区別を保持するには、次の例のように、必ず引用符で囲んでください。

正:

```
$ cimmo fl "-E" "--xml"
```

誤:

```
$ cimmo fl -E -xml
```

4.27.6 アンインストール後の WBEMCIM リポジトリ・ディレクトリ・ツリーのクリーン・アップ

DCL コマンド `PRODUCT REMOVE WBEMCIM` は、WBEM_SERVICES\$SETUP.EXE がインストール後に作成するリポジトリ・ディレクトリ・ツリーを削除しません。SYS\$COMMON:[WBEM_SERVICES.VAR...]*.*,*を手動で削除する必要があります。

このディレクトリ・ツリーを削除しないと、WBEM をアップグレードまたは再インストールするときに、WBEM_SERVICES\$SETUP.EXE を実行すると複数の "File already exists" エラーが報告されます。

4.28 I64 では \$SIGNAL_ARRAY_64 システム・サービスがない

V8.3

システム・サービス \$SIGNAL_ARRAY_64 は、I64 にはありません。I64 でこのサービスを呼び出すと、戻り状態値 `SS$_NOT_LOADED` (10 進数で 4026) が返されます。このサービスは、OpenVMS I64 の将来のリリースで追加される予定です。

この問題を回避するには、メカニズム配列内のオフセット `chf$ph_mch_sig64_addr` にある、64 ビットのシグナル配列のアドレスを見つけます。たとえば C 言語で次の文があるとしたします。

```
status = sys$signal_array_64(mech_array, &sig64_array);
```

これを、次の文に置き換えます。

```
sig64_array = ((CHFDEF2 *)mech_array)->chf$ph_mch_sig64_addr;
```

4.29 『HP OpenVMS System Analysis Tools Manual』

V8.3

『HP OpenVMS System Analysis Tools Manual』については、以下の訂正があります。このマニュアルは、OpenVMS Version 8.3 用に更新されていませんが、ここに示す訂正と、『HP OpenVMS V8.3 新機能説明書』で説明する追加情報は、SDA ユーティリティと、ANALYZE およびシステム・サービス・ログに関連するコマンドのオンライン・ヘルプに取り込まれています。

4.29.1 DUMPSTYLE の各ビットの定義

表 2-1 の最後の行は、次の行に置き換える必要があります。

5	32	0=選択型ダンプに、すべてのプロセス・ページとグローバル・ページを書き込む。 1= 1=選択型ダンプに主要なプロセス・ページとグローバル・ページだけを書き込む。フル・ダンプを書き込む際には (ビット 0 = 0)、このビットは無視される。このビットは、『システム管理者マニュアル (下巻)』の説明に従って、優先プロセスが正しく設定されている場合にだけ設定する。
---	----	---

4.29.2 システム・ダンプの保存

以降の段落が第 2.2.2 項に追加となります。

ダンプを分析する際、システムのクラッシュ時にダンプ・ファイルに書き込むことができないデータが利用できると便利です。このデータには、ファイル識別子に関連した完全なファイル指定が含まれます。I64 システムでは、プロセス内でアクティブ化されたイメージのアンワインド・データも含まれます。

ダンプが書き込まれたシステムでダンプを分析する場合は、COLLECT コマンドを使用して、現在の SDA セッションで使用するためにこのデータを収集することができます。分析のために別の場所にダンプをコピーしている場合は、COPY/COLLECT コマンドを使用してデータを収集し、書き込まれるコピーに追加することができます。

す。COPY/COLLECT コマンドを COLLECT コマンドの後に使用すると、すでに収集済みのデータがダンプ・コピーに追加されます。

デフォルトでは、システムのクラッシュ時に書き出されるオリジナルのダンプのコピーには、収集内容が含まれます。この動作は、COPY/NOCOLLECT を使用して変更することができます。逆に、以前 SDA で収集を行わずに (COPY/NOCOLLECT) コピーしたダンプのコピーには、収集内容は含まれません。この動作は、COPY /COLLECT を使用して変更することができます。

追加された収集内容がすでに含まれているダンプをコピーすると、常に収集内容が含まれます。

すべてのファイルとアンワインド・データを正常に収集するためには、システムのクラッシュ時にマウントされていたすべてのディスクを再マウントし、SDA を実行しているプロセスからアクセスできるようにする必要があります。ダンプの内容を保存するため、起動時の早い時期に SDA を起動し (たとえば CLUESSITE_PROC を使用。第 2.2.3 項を参照)、バッチ・ジョブが動作するまでにディスクがマウントされていない場合は、コマンド・プロシージャ CLUESSITE_PROC で COPY /NOCOLLECT コマンドを使用する必要があります。いったんすべてのディスクがマウントされたら、COPY/COLLECT を使用してファイルとアンワインド・データを保存することができます。

COPY 操作と COLLECT 操作を一度の手順で実行できない場合は、COLLECT /SAVE コマンドを入力すると、収集内容が個別のファイルに書き出されます。このファイルは、後でダンプ・ファイルとともに使用することができます。後でコピーを行うことにより 2 つのファイルが結合されます。

4.29.3 システムをリブートする際の SDA の起動

第 2.2.3 項で、CLUE HISTORY コマンドについての段落の最後に次の内容が追加となります。

COPY コマンドで /NOCOLLECT 修飾子を指定しなくてはならない場合があります。詳細については、前の項を参照してください。

4.29.4 SDA CONTEXT

第 2.5 項で、SET PROCESS コマンドと SHOW PROCESS コマンドの最初のリストに、SHOW PROCESS/SYSTEM 行と SHOW PROCESS/NEXT 行が追加となります。また、以下の行が一覧に追加となります。

```
VALIDATE PROCESS/POOL プロセス名
VALIDATE PROCESS/POOL/ADDRESS=PCB アドレス
VALIDATE PROCESS/POOL/INDEX=nn
VALIDATE PROCESS/POOL/NEXT
VALIDATE PROCESS/POOL/SYSTEM
```


この項の最後で、SET PROCESS コマンドと SHOW PROCESS コマンドの一覧の最後に、以下の行が追加となります。

```
VALIDATE PROCESS/POOL プロセス名  
VALIDATE PROCESS/POOL/ADDRESS=PCB アドレス  
VALIDATE PROCESS/POOL/INDEX=nn  
VALIDATE PROCESS/POOL/NEXT
```

4.29.5 SDA シンボルの初期化

第 2.6.1.4 項で、次の文が第 4.29.5 項の見出しの直前に追加となります。

シンボルは小文字を含むことができます。シンボルを操作するコマンド (DEFINE, SHOW SYMBOL, UNDEFINE など) では、小文字を含むシンボルを引用符で囲む必要があります。

4.29.6 ANALYZE

第 3 章のパラメータの項で、ファイル指定の説明が次のように変更となります。

分析するダンプが格納されているファイルの名前です。ANALYZE/CRASH_DUMP コマンドでファイルを指定しないと、デフォルトは、バージョン番号が最も大きな SYS\$SYSTEM:SYSDUMP.DMP となります。このファイルが存在しないと、ファイル名の入力を促すプロンプトが表示されます。ファイル指定のいずれかのフィールドを指定すると、残りフィールドは、デフォルト・ディレクトリ、SYSDUMP.DMP、最も大きなバージョン番号となります。

パラメータの項に次の内容が追加となります。

コレクション・ファイル名

SDA が使用するファイルの名前で、ファイル識別子変換データとアンwind・データが格納されています。

4.29.7 Crash_Dump

第 3 章の形式の項で filespec は [filespec] に変更する必要があります (つまり、filespec パラメータは省略可能になりました)。

4.29.8 SDA コマンド

第 4 章の導入部で、FLT が削除され、以下のコマンドが追加となります。

COLLECT
SHOW EFI
SHOW VHPT
VALIDATE POOL
VALIDATE PROCESS

4.29.9 COPY コマンド

第 4 章で、COPY コマンドの説明に次の内容が追加となります。

ダンプを分析する際、システムのクラッシュ時にダンプ・ファイルに書き込むことができないデータが利用できると便利です。このデータには、ファイル識別子に関連した完全なファイル指定が含まれます。また、OpenVMS I64 では、プロセス内でアクティブ化されたイメージのアンワインド・データも含まれます。

ダンプが書き込まれたシステムでダンプを分析する場合は、COLLECT コマンドを使用して、現在の SDA セッションで使用するためにこのデータを収集することができます。分析のために別の場所にダンプをコピーしている場合は、COPY/COLLECT コマンドを使用してデータを収集し、書き出されるコピーに追加することができます。COPY/COLLECT コマンドを COLLECT コマンドの後に使用すると、すでに収集済みのデータがダンプ・コピーに追加されます。

デフォルトでは、システムのクラッシュ時に書き出されるオリジナルのダンプのコピーには、収集内容が含まれます。この動作は、COPY/NOCOLLECT を使用して変更することができます。逆に、以前 SDA で収集を行わずにコピー (COPY/NOCOLLECT) したダンプのコピーには、収集内容は含まれません。この動作は、COPY/COLLECT を使用して変更することができます。追加された収集内容がすでに含まれているダンプをコピーすると、常に収集内容が含まれます。

すべてのファイルとアンワインド・データを正常に収集するためには、システムのクラッシュ時にマウントされていたすべてのディスクを再マウントし、SDA を実行しているプロセスからアクセスできるようにする必要があります。ダンプの内容を保存するため、起動中の早い時期に SDA を起動し (たとえば CLUE\$SITE_PROC を使用。第 2.2.3 項を参照)、バッチ・ジョブが動作するまでにディスクがマウントされていない場合は、コマンド・プロシージャ CLUE\$SITE_PROC で COPY/NOCOLLECT コマンドを使用する必要があります。いったんすべてのディスクがマウントされたら、COPY/COLLECT を使用してファイルとアンワインド・データを保存することができます。

COPY コマンドと COLLECT コマンドを一度の手順で実行できない場合は、COLLECT /SAVE コマンドを実行すると、収集内容が個別のファイルに書き出されます。このファイルは、後でダンプ・ファイルとともに使用することができます。後で COPY コマンドを実行することで 2 つのファイルが結合されます。

4.29.10 DUMP コマンド

第 4 章で、DUMP コマンドの説明が以下のように変更となります。

- フォーマットの項の最後で、/BYTE | /WORD | が/LONGWORDおよび[/NOSUPPRESS]の前に追加となります。
- rangeパラメータの説明に、次の文章が追加となります。

範囲の長さは、データ項目サイズ (/INDEX_ARRAYが指定されている場合はインデックス配列サイズ) の倍数になっている必要があります。

- /RECORD_SIZEパラメータの説明に次の文章が追加となります。

レコード・サイズを指定せず、範囲の長さが 512 バイトを超えない場合は、指定された範囲を含み、レコード番号フィールドのない、単一のレコードが出力されます。範囲の長さは、データ項目サイズ (/INDEX_ARRAYが指定されている場合はインデックス配列サイズ) の倍数になっている必要があります。

4.29.11 EVALUATE コマンド

第 4 章で、EVALUATE コマンドの説明が以下のように変更となります。

- フォーマットの項で、[= filter]を/ [NO] SYMBOLSの後に追加する必要があります。
- /SYMBOLS修飾子と/NOSYMBOLS修飾子の説明が、次の内容に置き換えられます。

EVALUATE コマンドのデフォルトの動作では、評価した式に等しいことが分かっているシンボルが 5 個まで表示されます。フィルタなしで/SYMBOLSを指定すると、すべてのシンボルがアルファベット順に表示されます。/NOSYMBOLSを指定すると、16 進数値と 10 進数値だけが表示されます。フィルタ付きで/SYMBOLSを指定すると、フィルタに一致するシンボルだけが表示されます。フィルタは、PCB\$*のように、ワイルドカードを含んだ文字列です。

4.29.12 MAP コマンド

第 4 章で、説明の項の最後に次の文が追加となります。

OpenVMS for Integrity サーバでは、MAP コマンドによってシステム空間内のアドレスの追加データを出力することもできます。アドレスが、ロードされた実行可能なイメージまたは常駐の共用可能イメージのコード・セクションにあり、イメージ・ファイルにアクセス可能で、/TRACEBACK付きでリンクされている場合は、トレースバック・データを使用してモジュール名とルーチン名の情報が取得され表示されます。

第 4 章で、MAP コマンドの説明に次の例が追加となります。

```
11. SDA> EVALUATE 2F0/SYMBOL=PCB*
      Hex = 00000000.000002F0   Decimal = 752      PCB$L_INITIAL_KTB
                                           PCB$L_PCB
```

この例は、シンボル・フィルタの使用方法を示します。値が 2F0 で名前が PCB で始まるシンボルだけが表示されます。

4.29.13 SET CPU コマンド

第 4 章で、SET PROCESS コマンドと SHOW PROCESS コマンドの一覧に以下の行を追加する必要があります。

```
VALIDATE PROCESS/POOL プロセス名
VALIDATE PROCESS/POOL/ADDRESS=PCB アドレス
VALIDATE PROCESS/POOL/INDEX=nn
VALIDATE PROCESS/POOL/NEXT
```

4.29.14 SET PROCESS コマンド

第 4 章で、SET PROCESS コマンドと SHOW PROCESS コマンドの一覧に以下の行を追加する必要があります。

```
VALIDATE PROCESS/POOL プロセス名
VALIDATE PROCESS/POOL/ADDRESS=PCB アドレス
VALIDATE PROCESS/POOL/INDEX=nn
VALIDATE PROCESS/POOL/NEXT
VALIDATE PROCESS/POOL/SYSTEM
```

4.29.15 READ コマンド

第 4 章で、説明の項の最後に次の文が追加となります。

directory-spec パラメータの説明

で、SYS\$LOADABLE_IMAGES と SYS\$LIBRARY は、SYS\$LOADABLE_IMAGES、SYS\$LIBRARY、および SYS\$SYSTEM に変更となります。

表 4-1 の脚注 3 で、次の文を削除する必要があります。

これらは SYS\$SYSTEM 内にあり、READ/EXEC コマンドを実行しても自動的に読み込まれません。

4.29.16 SEARCH コマンド

第 4 章のSEARCHコマンドの説明は以下のように変更となります。

- /LENGTH修飾子と/MASK修飾子の説明の最後に、次の文が追加となります。
この修飾子は、文字列検索では無視されます。
- /STEPS修飾子の説明で、"expression" (式) の後に "or the given string" (または指定された文字列) を挿入する必要があります。
- /STEPS修飾子の説明の最後の文の最後に、 "...for value searches, and a step factor of a byte for string searches"が追加 (最後の文が「/STEP 修飾子を指定しないと、値の検索ではロングワードのステップ係数が使用され、文字列の検索ではバイトのステップ係数が使用されます。」に変更) となります。
- 説明の項で、"value" (値) の後に "or string" (または文字列) という記述が挿入となります。

4.29.17 SHOW_CALL_FRAME コマンド

第 4 章で、starting-addressパラメータの説明は、次の文に置き換えとなります。

Alpha の場合

表示するプロシージャ呼び出しフレームの開始アドレスを表す式です。開始アドレスの値を指定しないと、SDA の現在のプロセスのフレーム・ポインタ (FP) レジスタの内容がデフォルトの開始アドレスとなります。 pthread を使用するプロセスの場合は、次の SDA コマンドを使用してすべての pthread の開始アドレスを表示することができます。

```
SDA> pthread thread -o u
```

OpenVMS for Integrity サーバの場合

以下のいずれかを表す式です。

- フレームの呼び出しコンテキスト・ハンドル
- 例外フレームのアドレス。これは、次の SDA コマンドと同等です。

```
SDA> SHOW CALL_FRAME /EXCEPTION_FRAME=starting-address
```

- スレッド環境ブロック (TEB) のアドレス。プロセスのすべての TEB の一覧を表示するには、次の SDA コマンドを使用します。

```
SDA> pthread thread -o u
```

starting-addressを指定しないと、デフォルトの開始アドレスは、SDA の現在のプロセス内の現在のプロシージャの呼び出しコンテキスト・ハンドルとなります。

4.29.18 SHOW CLUSTER コマンド

第 4 章の SHOW CLUSTER コマンドの説明は、以下のように変更となります。

フォーマットの項で、"/CIRCUIT=PB アドレス | "が "/CSID"の前に追加となります。

/ADDRESS 修飾子の説明で、"/CSID=csid"および "/NODE=名前"は、
"/CIRCUIT=PB アドレス", "/CSID=csid", および "/NODE=名前"に置き換えとなります。

/CSID 修飾子と/NODE 修飾子の説明で、"/CIRCUIT=PB アドレス"が
/ADDRESS=n"の後に追加となります。

次の文が説明に追加となります。

修飾子/CIRCUIT=PB アドレスを指定すると、SHOW CLUSTER コマンドは指定されたパス・ブロックから得た情報だけを表示します。

4.29.19 SHOW CRASH コマンド

第 4 章の SHOW CRASH コマンドの説明は、以下のように変更となります。

フォーマットの項は、"SHOW CRASH [/ALL | /CPU = n]"に変更となります。

説明の項で、後ろから 2 つ目の段落に次の内容が追加となります。

/ALL を指定しないと、バグチェック CPUEXIT または DBGCPUEXIT を持つ CPU
について、レジスタ (Alpha 上) または例外フレームの内容 (I64 上) が表示から省略されます。

説明の項で、最後の段落から "and additionally displays all CPU database
addresses in system dumps" (また追加でシステム・ダンプ内のすべての CPU
データベース・アドレスを表示します) が削除となります。

例 3 のコマンドは、次のように変更となります。

```
SDA> SHOW CRASH /ALL
```

4.29.20 SHOW DUMP コマンド

第 4 章で、SHOW DUMP コマンドの説明は、次のように変更となります。

フォーマットの項は次の内容に置き換えとなります。

```
SHOW DUMP [ /ALL
           | /BLOCK [ = m [ { : | ; } n ] ]
           | /COLLECTION [ = { ALL | n } ]
           | /COMPRESSION_MAP [ = m [ : n [ : p ] ] ]
           | /ERROR_LOGS
           | /FILE = { COLLECTION | DUMP }
           | /HEADER
           | /LMB [ = { ALL | n } ]
           | /MEMORY_MAP
           | /SUMMARY ]
```

説明の項で、"the memory map, and the file identification, and/or unwind data collection" (メモリ・マップ、ファイル識別子、アンワインド・データの収集内容) は、"and the memory map" (およびメモリ・マップ) に置き換えとなります。

次の例が SHOW DUMP コマンドに追加となります。

3. SDA> SHOW DUMP/COLLECTION

File and unwind data collection

```
Collection start VBN:      0002155B
Collection end VBN:       00022071
Collection block count:   00000B17
```

VBN	Blocks>	Contents
0002155B	000000C1	Unwind data segment 00000001 of _\$30\$DKB200: [VMS\$COMMON.SYSEXEXE] DCL.EXE;1
0002161C	00000001	Unwind data segment 00000001 of _\$30\$DKB200: [VMS\$COMMON.SYSEXEXE] USB\$UCM_SERVER.EXE;1
0002161D	0000000C	Unwind data segment 00000008 of _\$30\$DKB200: [VMS\$COMMON.SYSEXEXE] USB\$UCM_SERVER.EXE;1
.		
.		
.		
0002200F	0000001F	Unwind data segment 00000007 of _\$30\$DKB200: [VMS\$COMMON.SYSEXEXE] LATACP.EXE;1
0002202E	00000006	Unwind data segment 0000000B of _\$30\$DKB200: [VMS\$COMMON.SYSEXEXE] LATACP.EXE;1
00022034	00000001	Unwind data segment 00000002 of _\$30\$DKB200: [BISHOP] CMEXEC_LOOP.EXE;1
00022035	00000001	File data for _\$30\$DKA0:
00022036	0000003B	File data for _\$30\$DKB200:
00022071	00000001	Disk data

この SHOW DUMP/COLLECTION コマンドの例は、SDA コマンド COPY /COLLECT を使用してシステム・ダンプをコピーしたときに追加されたファイル識別子およびアンワインド・データの収集内容を表示します。アンワインド・データ・セグメントは、OpenVMS I64 システムのシステム・ダンプだけにある点に注意してください。

4.29.21 SHOW PROCESS コマンド

第 4 章の SHOW PROCESS コマンドの説明は、以下のように変更となります。

フォーマットの項は、次のように変更となります。

- "{ =P0 | =P1 | =ALL }"で、"=IMGACT"を "=ALL"の前に挿入し、"(D)"を "=ALL"に追加
- "[/BRIEF { /FREE | /UNUSED } | ...]"で、"/CHECK"を "/BRIEF"の後に挿入
- "/UNWIND_TABLE [=ALL]"を "/UNWIND_TABLE [= { ALL | 名前}]"に変更

修飾子/ALL の説明で、"/STATISTICS"が "/POOL/HEADER/RING_BUFFER"の行に追加となります。

/POOL 修飾子の説明の見出しと最初の段落は、次のように変更となります。

/POOL [= { P0 | P1 | IMGACT | ALL (D) }]

プロセスの P0 (プロセス) 領域、P1 (制御) 領域、イメージ・アクティベータの予約ページ、またはオプションで、あるアドレス範囲内の動的なストレージ・プールを表示します。デフォルトでは、動的なストレージ・プールをすべて表示します。

/UNWIND_TABLE 修飾子の説明は、次の内容に置き換えとなります。

/UNWIND_TABLE [= { ALL | 名前}] (I64 のみ)

キーワードなしで指定すると、プロセスのマスタ・アンワインド・テーブルが表示されます。SHOW PROCESS/UNWIND=ALL は、すべてのプロセス・アンワインド記述子の詳細を表示します。SHOW PROCESS/UNWIND=名前 は、指定した名前 (ワイルドカードが使用可能) を持つイメージのすべてのアンワインド記述子の詳細を表示します。プロセス空間内の特定の PC のアンワインド・データを表示するには、SHOW UNWIND アドレス を使用します。

あるイメージのいくつかまたはすべてのアンワインド・データがシステム・ダンプに含まれていない場合は (たとえば、システムのクラッシュ時にプロセスのワーキング・セットになかった場合)、アンワインド・データにアクセスできないため、SHOW PROCESS/UNWIND コマンドは%SDA-W-NOREAD エラーで失敗します。アンワインド・データを収集 (コマンド COLLECT および COPY/COLLECT を参照) してもこの状況は変わりません。収集されたアンワインド・データは SHOW UNWIND アドレス および SHOW CALL でのみ使用されるためです。

"For I64" (I64 の場合) の下の 2 番目の箇条書き項目は、次のように変更となります。

特殊な用途のレジスタ (PC, PSR, ISR)。PC は、PSR から得たスロット番号と IP の組み合わせであることに注意してください。

4.29.22 SHOW RESOURCES コマンド

第 4 章の SHOW RESOURCES コマンドの説明は、次のように変更となります。

次のキーワードが/STATUS 修飾子に追加となります。

キーワード	意味
RM_FORCE	強制的にツリーを移動する。
RM_FREEZE	このノードのリソース・ツリーを凍結する。
RM_INTEREST	マスタに関心がないため、再マスタリングを行う。
XVAL_VALID	最後の値ブロックは長いブロックだった。

4.29.23 SHOW SPINLOCKS コマンド

第 4 章の SHOW SPINLOCKS コマンドの説明は、次のように変更となります。

フォーマットの項は次のように変更となります。

```
SHOW SPINLOCKS { [name] | /ADDRESS = 式 | /INDEX = 式 }  
                [ /OWNED | /DYNAMIC | /STATIC  
                | /CACHE PCB | /MAILBOX | /PCB | /PORT | /PSHARED ]  
                [ { /BRIEF | /COUNTS | /FULL } ]
```

nameパラメータの説明で、"PCB, or cached PCB" (PCB またはキャッシュされた PCB) は、"PCB, cached PCB, or process-shared" (PCB, キャッシュされた PCB, またはプロセス共用された) に変更となります。

/ADDRESS 修飾子の説明で、"PCB, or cached PCB" (PCB またはキャッシュされた PCB) は、"PCB, cached PCB, or process-shared" (PCB, キャッシュされた PCB, またはプロセス共用された) に変更となります。

/DYNAMIC 修飾子の説明で、"PCB, or cached PCB" (PCB またはキャッシュされた PCB) は、"PCB, cached PCB, or process-shared" (PCB, キャッシュされた PCB, またはプロセス共用された) に変更となります。/PORT 修飾子の後に次の新しい修飾子が追加となります。

/PSHARED

プロセス共用された (pthread) スピンロックをすべて表示します。

説明の項の 3 番目の段落で、"PCB, and cached PCB" (PCB およびキャッシュされた PCB) は、"PCB, cached PCB, and process-shared" (PCB, キャッシュされた PCB, およびプロセス共用された) に変更となります。

4.29.24 SHOW MEMORY コマンド

第 4 章の SHOW MEMORY コマンドの説明は、次のように変更となります。

/SLOTS 修飾子の説明で、"partition" (パーティション) という語は "process" (プロセス) に変更となります。

4.29.25 SHOW GCT コマンド

第 4 章の SHOW GCT コマンドの説明は、以下のように変更となります。

フォーマットの項で、"[CHILDREN] |"は "[/CHILDREN] |"に変更となり、その後 "[/FULL] |"が追加となります。

修飾子の項で、次の/FULL 修飾子の説明が/CHILDREN の後に追加となります。

/FULL

/FULL 修飾子を/CHILDREN、/OWNER=n、/TYPE=タイプ のいずれかとともに使用すると、SDA は各ノードについて詳細に表示します。

/OWNER と/TYPE の説明で、"Provides a detailed display of all nodes" (すべてのノードについて詳細に表示します) は "Displays all nodes" (すべてのノードを表示します) に変更となります。

/TYPE の説明で、"CORE"、"SOCKET"、および "THREAD"が、有効なタイプの一覧にアルファベット順で追加となります。

4.29.26 SHOW CPU コマンド

第 4 章の SHOW CPU コマンドの説明に、次の内容が追加となります。

- I64 では例外フレームの概要

4.29.27 SHOW SYMBOL コマンド

symbol-nameパラメータの説明に、次の文が追加となります。

小文字を含むシンボルは、引用符で囲む必要があります。

4.29.28 SHOW UNWIND コマンド

フォーマットの項は次のように変更となります。

/IMAGE

SHOW UNWIND [アドレス | /ALL | /IMAGE=名前]

4.29.29 SHOW SWIS コマンド

第 4 章の SHOW SWIS コマンドの説明は、以下のように変更となります。

- フォーマットの項は次のように変更となります。

SHOW SWIS [/RING_BUFFER [/CPU = (m, n,...)]]

- /CPU修飾子の説明は次のように変更となります。

/CPU = (m, n,...)

/RING_BUFFER とともに使用すると、指定した CPU のエントリだけが表示されます。CPU を 1 つだけ指定する場合は、括弧は省略してかまいません。

- 説明の項の最後の文は、次のように変更となります。

/CPU = (m, n) を指定すると、指定した CPU のレコードだけが表示されます。

4.29.30 CLUE CONFIG コマンド

第 5 章の CLUE CONFIG コマンドの説明に、次の修飾子が追加となります。

/CPU

システム、メモリ、CPU に関する情報が格納されたシステム構成の部分だけを表示します。

/ADAPTER

システム上のアダプタとデバイスに関する情報が格納されたシステム構成の部分だけを表示します。

次の文が説明の項に追加となります。

修飾子を全く指定しないと、システム構成全体が表示されます。

4.29.31 CLUE REGISTER コマンド

第 5 章で、CLUE REGISTER コマンドの説明が以下のように変更となります。

フォーマットの項は次のように変更となります。

```
CLUE REGISTER [/CPU [cpu-id|ALL]
               | /PROCESS[/ADDRESS=n|INDEX=n
               | /IDENTIFICATION=n|プロセス名|ALL]]
```

パラメータと修飾子は、CLUE_CALL_FRAME コマンドと同じです。

説明の項に次の文が追加となります。

/CPU と/PROCESS のどちらも指定しないと、パラメータ (cpu-id やプロセス名) は無視され、SDA の現在のプロセスのレジスタが表示されます。

4.29.32 I64 ISD_Labels インデックスの表

第 10 章の I64 ISD_Labels インデックスの表は、次の表に置き換わります。

表 4-5 I64 ISD_Labels インデックス

インデックス	名前	意味
0	SDA_CIO\$K_FIX	フィックスアップ
1	SDA_CIO\$K_PROMO_CODE	プロモート (コード)
2	SDA_CIO\$K_PROMO_DATA	プロモート (データ)
3	SDA_CIO\$K_INIT_CODE	初期化 (コード)
4	SDA_CIO\$K_INIT_DATA	初期化 (データ)
5	SDA_CIO\$K_CODE	コード
6	SDA_CIO\$K_SHORT_RW	short 型データ (読み書き)
7	SDA_CIO\$K_SHORT_RO	short 型データ (読み込み専用)
8	SDA_CIO\$K_RW	データ (読み書き)
9	SDA_CIO\$K_RO	データ (読み込み専用)
10	SDA_CIO\$K_SHORT_DZ	short 型データ (デマンド・ゼロ)
11	SDA_CIO\$K_SHORT_TDZ	short 型データ (後続のデマンド・ゼロ)
12	SDA_CIO\$K_DZERO	デマンド・ゼロ
13	SDA_CIO\$K_TR_DZERO	後続のデマンド・ゼロ

4.29.33 SDA 拡張機能のコンパイルとリンク

第 10.2.1 項の注 2 で、2 箇所ある "ALPHA\$LIBRARY" は、"SYS\$LIBRARY" に変更となります。

4.29.34 拡張機能のデバッグ

10.3 項の例で、2 箇所ある "alpha\$library" は "sys\$library" に変更となります。

4.29.35 呼び出し可能ルーチンの概要

10.4 項で、"SDA\$NEWPAGE" は "SDA\$NEW_PAGE" に変更となります。

現在の一覧に以下のルーチンがアルファベット順に追加となります。

SDA\$CBB_BOOLEAN_OPER
SDA\$CBB_CLEAR_BIT
SDA\$CBB_COPY
SDA\$CBB_FFC
SDA\$CBB_FFS
SDA\$CBB_INIT
SDA\$CBB_SET_BIT
SDA\$CBB_TEST_BIT
SDA\$DELETE_PREFIX
SDA\$FID_TO_NAME
SDA\$GET_FLAGS

"So, for example," (そのため、たとえば) で始まるこの項の最後の段落は、"Some routines expect..." (いくつかのルーチンは... 期待します。) で始まる最後の箇条書き項目の中に移動となります。

次の箇条書き項目がその項の最後に追加となります。

- CBB (Common Bitmask Block) ルーチン SDA\$CBB_xxx は、システムで使用している CPU について記述する CBB 構造体のローカルなコピーに対して使用するように設計されています。CBB 構造体の長さは、少なくとも CBB\$K_STATIC_BLOCK バイトと想定されます。これらのルーチンで使用されるさまざまな CBB 定数とフィールド名の定義は、SYS\$LIBRARY:SYS\$LIB_C.TLB 内の CBBDEF.H にあります。

一連のルーチンは、考えられる CBB 関連の操作をすべて網羅することを目的としたものではなく、必要であることが分かっている操作を提供するためのものです。CPU について記述すること以外の目的で設定された CBB 構造体では、これらのルーチンが期待どおりに動作しないことがあります。

4.29.36 ターゲット・システムでの接続の設定

第 11.3 項のブート・コマンドの説明で、"with boot command" (ブート・コマンドを使用して) という文は、"with the boot command" (boot コマンドを使用して) に変更となります。

SCD 構成ファイルの説明の前の段落で、最後の文は "See the Boot Option Maintenance Menu, as described in the *HP OpenVMS System Manager's Manual, Volume 1: Essentials*." (『OpenVMS システム管理者マニュアル (上巻)』に記載されているブート・オプション保守メニューを参照してください。) に変更となります。

第 11.3 項で、XDELTA コマンドの説明の最初の箇条書き項目は、次のように変更となります。

- n\xxxx\R

第 11.3 項のシステム・パラメータの説明に、以下の 2 つの箇条書き項目が追加となります。

- BREAKPOINTS

このパラメータはビットマップになっており、以下の状況で OpenVMS 内に存在する INISBRK 呼び出しが可能となります。

- ビット 0: INIT の先頭
- ビット 1: INIT の最後
- ビット 2: INIT の中で、セカンダリ CPU を開始する直前
- ビット 3: INISBRK が外部モードから呼び出された場合
- ビット 4: 新たにロードされたエグゼクティブ・イメージの初期化ルーチンを呼び出す前
- ビット 5-31: 弊社での使用のために予約

注意

1. BREAKPOINTS のビット 3 がオフの場合にエグゼクティブ・モードから INISBRK を呼び出すと、プロセスが終了するか、SSRVEXCEPT バグチェック (SYSTEM_CHECK と BUGCHECKFATAL のどちらかも設定されている場合) となります。
 2. BREAKPOINTS をデフォルト値の 3 から変更すると、システムのセキュリティが低下するおそれがあるため、変更は慎重に行う必要があります。
-

- TIME_CONTROL

このパラメータはビットマスクになっており、OpenVMS の特定の時間制御機能を無効にします。

- ビット 0: システム・クロックを無効にする
- ビット 1: CPU サニティ・タイムアウトを無効にする
- ビット 2: CPU スピンウェイト・タイムアウトを無効にする

XDELTA または SCD をロードする際に (ブート・フラグのビット 1 またはビット 15 がオンになる)、TIME_CONTROL の値がデフォルト値のゼロから 6 (CPU サニティ・タイムアウトと CPU スピンウェイト・タイムアウトが無効) に変わります。これは、システムがブレークポイントで待っているときにこれらのタイムアウトが発生しないようにするためです。これらの設定を変更する必要がある場合は、SYSGEN を使用するか、XDELTA または SCD でDEPOSITコマンドを使用します。ビット 0 はオンにしないでください。

第 11.3.2 項の最後に、次の内容を追加する必要があります。

I64 での同等のテクニックは、SCD フラグ (ビット 15) だけをオンにしてシステムをブートすることです。エラーが発生したのに気づいたら、コンソールで Ctrl/P を押します。この動作によって、XDELTA に制御が渡り (XDELTA ブート・フラグがオンでない場合でも)、1;R と入力できるようになります。ターゲット・カーネルが制御権を獲得し、SCD の接続を待ちます。

また、以下の新しい第 11.3.3 項が追加となります。

ターゲット・カーネルは、イーサネット・デバイスを排他的に使用できる必要があります。DECnet など、システム・コンポーネントによっては SCD がロードされると起動しないものがあります。複数のイーサネット・デバイスがあり、ターゲット・カーネルに対して SCD イーサネット・デバイスの排他的なアクセスを許可するようにシステムが構成されている場合は、影響のあるシステム・コンポーネントが通常通り起動するように、論理名 DBGTK\$OVERRIDE を定義する必要があります。この論理名はシステム全体で定義することも、システム・コンポーネントの起動コマンドを実行するプロセス内で定義することもできます。

第 11.11.1 項で、2 番目の段落の最後の文 "To remove symbols..." (シンボルを削除するには...) は削除となります。

第 11.11.3 項で、箇条書きの 2 番目と 3 番目の項目は削除となり、最初の項目 "Access to All Executive Image Symbols" (すべてのエグゼクティブ・イメージ・シンボルへのアクセス) は箇条書きから通常の文に変更となります。

第 11.12 項で、例 11-1 の直前に次の新しい段落が追加となります。

例 11-5 以降の表示例は、すべて OpenVMS I64 システムでの例です。OpenVMS Alpha システムでは、出力がいくらか異なりますが、入力するコマンドはどちらのプラットフォームでも同じです (付随する文章に記載されている 1 つの例外を除く)。

第 11.12 項の "V8.2-014" は , "V8.3-003" に変更となります。

例 11-5 は次の内容に置き換えとなります。

```
DBG> connect %node_name TSTSYS
%DEBUG-I-INIBRK, target system interrupted
DBG> show image
image name                set    base address    end address
ERRORLOG                  no     0000000000000000  FFFFFFFFFFFFFFFF
EXEC_INIT                  no     0000000000000000  FFFFFFFFFFFFFFFF
SYS$ACPI                   no     0000000000000000  FFFFFFFFFFFFFFFF
*SYS$BASE_IMAGE            yes    0000000000000000  FFFFFFFFFFFFFFFF
SYS$DKBTDRIVER             no     0000000000000000  FFFFFFFFFFFFFFFF
SYS$DKBTDRIVER             no     0000000000000000  FFFFFFFFFFFFFFFF
SYS$DKBTDRIVER             no     0000000000000000  FFFFFFFFFFFFFFFF
SYS$EGBTDRIVER             no     0000000000000000  FFFFFFFFFFFFFFFF
SYS$OPDRIVER               no     0000000000000000  FFFFFFFFFFFFFFFF
SYS$PKMBTDRIVER            no     0000000000000000  FFFFFFFFFFFFFFFF
SYS$PKMBTDRIVER            no     0000000000000000  FFFFFFFFFFFFFFFF
SYS$PKMBTDRIVER            no     0000000000000000  FFFFFFFFFFFFFFFF
SYS$PLATFORM_SUPPORT      no     0000000000000000  FFFFFFFFFFFFFFFF
SYS$PUBLIC_VECTORS         no     0000000000000000  FFFFFFFFFFFFFFFF
SYS$SRBTDRIVER             no     0000000000000000  FFFFFFFFFFFFFFFF
SYSTEM_DEBUG               no     0000000000000000  FFFFFFFFFFFFFFFF
SYSTEM_PRIMITIVES          no     0000000000000000  FFFFFFFFFFFFFFFF
SYSTEM_SYNCHRONIZATION    no     0000000000000000  FFFFFFFFFFFFFFFF

total images: 18
DBG>
```

例 11-7 は次の内容に置き換えとなります。

```
DBG> set image system_debug
%DEBUG-I-DYNLNGSET, setting language IMACRO
DBG> show module
module name                symbols  language  size
AUX_TARGET                 no       C          0
BUF$RV_TARGET              no       C          0
BUGCHECK_CODES             no       BLISS      0
C_TEST_ROUTINES            no       C          0
LIB$UNWIND_WEAK            no       BLISS      0
LIB$EF                      no       IMACRO     0
LIB$MALLOC                 no       C          0
LIB$MALLOC_64              no       C          0
LINMGR_TARGET              no       C          0
OBJMGR                     no       C          0
PLUMGR                     no       C          0
POOL                       no       C          0
PROTOMGR_TARGET            no       C          0
SOCMGR                     no       C          0
SYS$DOINIT                 yes      IMACRO    122526
TMRMGR_TARGET              no       C          0

total modules: 16
DBG> set module c_test_routines
DBG> show module c_test_routines
module name                symbols  size
```

```
C_TEST_ROUTINES          yes          5672
```

```
total C modules: 1
```

```
DBG> set language c
```

```
DBG> show symbol test_c_code*
```

```
routine C_TEST_ROUTINES\test_c_code
```

```
routine C_TEST_ROUTINES\test_c_code2
```

```
routine C_TEST_ROUTINES\test_c_code3
```

```
routine C_TEST_ROUTINES\test_c_code4
```

```
routine C_TEST_ROUTINES\test_c_code5
```

```
DBG> set break test_c_code
```

例 11-8 の最後の行は次の内容に置き換えとなります。

```
113:    x = c_test_array[0];
```

例 11-9 は次の内容に置き換えとなります。

```
DBG> Set Mode Screen; Set Step Nosource
```

```
- SRC: module C_TEST_ROUTINES -scroll-source-----
```

```
 98:    c_test_array[5] = in64;
```

```
 99:    c_test_array[6] = in32;
```

```
100:    if (c_test_array[9] > 0)
```

```
101:        *pVar = (*pVar + c_test_array[17]) & c_test_array[9];
```

```
102:    else
```

```
103:        *pVar = (*pVar + c_test_array[17]);
```

```
104:    c_test_array[7] = test_c_code3(10);
```

```
105:    c_test_array[3] = test;
```

```
106:    return c_test_array[23];
```

```
107: }
```

```
108: void test_c_code(void)
```

```
109: {
```

```
110:     int x,y;
```

```
111:     __int64 x64,y64;
```

```
112:
```

```
-> 113:    x = c_test_array[0];
```

```
114:    y = c_test_array[1];
```

```
115:    x64 = c_test_array[2];
```

```
116:    y64 = c_test_array[3];
```

```
117:    c_test_array[14] = test_c_code2(x64+y64,x+y,x64+x,&y64);
```

```
118:    test_c_code4();
```

```
119:    return;
```

```
120: }
```

```
- OUT -output-----
```

```
- PROMPT -error-program-prompt-----
```

DBG>

例 11-10 の直前の段落で、参照している 46 行目は 93 行目に置き換えとなります。

例 11-10 は、次の内容に置き換えとなります。

```
- SRC: module C_TEST_ROUTINES -scroll-source-----
 80: void test_c_code4(void)
 81: {
 82:     int i,k;
 83:     for(k=0;k<1000;k++)
 84:     {
 85:         test_c_code5(&i);
 86:     }
 87:     return;
 88: }
 89: int test_c_code3(int subrtnCount)
 90: {
 91:     subrtnCount = subrtnCount - 1;
 92:     if (subrtnCount != 0)
 93:         subrtnCount = test_c_code3(subrtnCount);
 94:     return subrtnCount;
 95: }
 96: int test_c_code2(__int64 in64,int in32, __int64 test, __int64* pVar)
 97: {
 98:     c_test_array[5] = in64;
 99:     c_test_array[6] = in32;
100:     if (c_test_array[9] > 0)
101:         *pVar = (*pVar + c_test_array[17]) & c_test_array[9];
102:     else
- OUT -output-----

- PROMPT -error-program-prompt-----

DBG> Scroll/Up
DBG> set break %line 93
DBG>
```

例 11-11 は、次の内容に置き換えとなります。

```
- SRC: module C_TEST_ROUTINES -scroll-source-----
  82:   int i,k;
  83:   for(k=0;k<1000;k++)
  84:   {
  85:       test_c_code5(&i);
  86:   }
  87:   return;
  88: }
  89: int test_c_code3(int subrtnCount)
  90: {
  91:     subrtnCount = subrtnCount - 1;
  92:     if (subrtnCount != 0)
-> 93:         subrtnCount = test_c_code3(subrtnCount);
  94:     return subrtnCount;
  95: }
  96: int test_c_code2(__int64 in64,int in32, __int64 test, __int64* pVar)
  97: {
  98:     c_test_array[5] = in64;
  99:     c_test_array[6] = in32;
 100:     if (c_test_array[9] > 0)
 101:         *pVar = (*pVar + c_test_array[17]) & c_test_array[9];
 102:     else
 103:         *pVar = (*pVar + c_test_array[17]);
 104:     c_test_array[7] = test_c_code3(10);
- OUT -output-----
break at C_TEST_ROUTINES\test_c_code3\%LINE 93

- PROMPT -error-program-prompt-----

DBG> Scroll/Up
DBG> set break %line 93
DBG> go
DBG>
```

例 11-12 の直前の段落で、参照している 147 行目は 94 行目に置き換えとなり、146 行目は 93 行目に置き換えとなります。

例 11-12 は、次の内容に置き換えとなります。

```
- SRC: module C_TEST_ROUTINES -scroll-source-----
  82:   int i,k;
  83:   for(k=0;k<1000;k++)
  84:   {
  85:       test_c_code5(&i);
  86:   }
  87:   return;
  88: }
  89: int test_c_code3(int subrtnCount)
  90: {
  91:     subrtnCount = subrtnCount - 1;
  92:     if (subrtnCount != 0)
-> 93:         subrtnCount = test_c_code3(subrtnCount);
  94:     return subrtnCount;
  95: }
  96: int test_c_code2(__int64 in64,int in32, __int64 test, __int64* pVar)
  97: {
  98:     c_test_array[5] = in64;
  99:     c_test_array[6] = in32;
 100:     if (c_test_array[9] > 0)
 101:         *pVar = (*pVar + c_test_array[17]) & c_test_array[9];
 102:     else
 103:         *pVar = (*pVar + c_test_array[17]);
 104:     c_test_array[7] = test_c_code3(10);
- OUT -output-----
break at C_TEST_ROUTINES\test_c_code3\%LINE 93
break at C_TEST_ROUTINES\test_c_code3\%LINE 93
```

例 11-13 の直前の段落で、次の文が閉じ括弧の直前に追加となります。

エグゼクティブ・イメージがスライスされている場合は、接尾辞_CODE0 が付加されます。

例 11-13 は、次の内容に置き換えとなります。

```
- SRC: module C_TEST_ROUTINES -scroll-source-----
  82:   int i,k;
  83:   for(k=0;k<1000;k++)
  84:   {
  85:       test_c_code5(&i);
  86:   }
  87:   return;
  88: }
  89: int test_c_code3(int subrtnCount)
  90: {
  91:     subrtnCount = subrtnCount - 1;
  92:     if (subrtnCount != 0)
-> 93:         subrtnCount = test_c_code3(subrtnCount);
  94:     return subrtnCount;
  95: }
  96: int test_c_code2(__int64 in64,int in32, __int64 test, __int64* pVar)
  97: {
  98:     c_test_array[5] = in64;
  99:     c_test_array[6] = in32;
 100:     if (c_test_array[9] > 0)
 101:         *pVar = (*pVar + c_test_array[17]) & c_test_array[9];
 102:     else
 103:         *pVar = (*pVar + c_test_array[17]);
 104:     c_test_array[7] = test_c_code3(10);
- OUT -output-----
C_TEST_ROUTINES\test_c_code3\subrtnCount:      8
  module name      routine name      line      rel PC      abs PC
*C_TEST_ROUTINES test_c_code3        93      000000000000DC0 FFFFFFFF800BAFC0
*C_TEST_ROUTINES test_c_code3        93      000000000000DE0 FFFFFFFF800BAFE0
*C_TEST_ROUTINES test_c_code2       104      000000000000F40 FFFFFFFF800BB140
*C_TEST_ROUTINES test_c_code       117      00000000000010B0 FFFFFFFF800BB2B0
                  XDT$INIT          00000000000015C0 FFFFFFFF880955C0
*SYS$DOINIT      EXE$INITIALIZE    1973      000000000000360 FFFFFFFF88094360
SHARE$EXEC_INIT_CODE0      000000000005C240 FFFFFFFF803BB640
SHARE$EXEC_INIT_CODE0      0000000000057F20 FFFFFFFF803B7320
SHARE$EXEC_INIT_CODE0      0000000000047850 FFFFFFFF803A6C50
SHARE$EXEC_INIT_CODE0      0000000000042E90 FFFFFFFF803A2290
- PROMPT -error-program-prompt-----
DBG> set break %line 93
DBG> go
DBG> Step
DBG> examine subrtnCount
DBG> show calls
DBG>
```

例 11-14 の直前の段落で、参照されている 147 行目は 94 行目に置き換えとなり、
146 行目は 93 行目に置き換えとなります。

例 11-14 は、次の内容に置き換えとなります。

```
- SRC: module C_TEST_ROUTINES -scroll-source-----
  83:   for(k=0;k<1000;k++)
  84:   {
  85:       test_c_code5(&i);
  86:   }
  87:   return;
  88: }
  89: int test_c_code3(int subrtnCount)
  90: {
  91:     subrtnCount = subrtnCount - 1;
  92:     if (subrtnCount != 0)
  93:         subrtnCount = test_c_code3(subrtnCount);
-> 94:     return subrtnCount;
  95: }
  96: int test_c_code2(__int64 in64,int in32, __int64 test, __int64* pVar)
  97: {
  98:     c_test_array[5] = in64;
  99:     c_test_array[6] = in32;
 100:     if (c_test_array[9] > 0)
 101:         *pVar = (*pVar + c_test_array[17]) & c_test_array[9];
 102:     else
 103:         *pVar = (*pVar + c_test_array[17]);
 104:     c_test_array[7] = test_c_code3(10);
 105:     c_test_array[3] = test;
- OUT -output-----
module name      routine name      line      rel PC      abs PC
*C_TEST_ROUTINES test_c_code3      93      000000000000DC0 FFFFFFFF800BAFC0
*C_TEST_ROUTINES test_c_code3      93      000000000000DE0 FFFFFFFF800BAFE0
*C_TEST_ROUTINES test_c_code2     104      000000000000F40 FFFFFFFF800BB140
*C_TEST_ROUTINES test_c_code     117      00000000000010B0 FFFFFFFF800BB2B0
                  XDT$INIT      00000000000015C0 FFFFFFFF880955C0
*SYS$DOINIT      EXE$INITIALIZE  1973      000000000000360 FFFFFFFF88094360
SHARE$EXEC_INIT_CODE0 0000000000005C240 FFFFFFFF803BB640
SHARE$EXEC_INIT_CODE0 00000000000057F20 FFFFFFFF803B7320
SHARE$EXEC_INIT_CODE0 00000000000047850 FFFFFFFF803A6C50
SHARE$EXEC_INIT_CODE0 00000000000042E90 FFFFFFFF803A2290
stepped to C_TEST_ROUTINES\test_c_code3\%LINE 94
- PROMPT -error-program-prompt-----
DBG> Step
DBG> examine subrtnCount
DBG> show calls
DBG> cancel break/all
DBG> go
DBG>
```

例 11-15 の前にある STEP/RETURN の段落で、"R0 register" (R0 レジスタ) の部分が、"R0 register on Alpha, or the R8 register on I64" (Alpha では R0 レジスタ、I64 では R8 レジスタ) に変更となります。

例 11-15 は、次の内容に置き換えとなります。

```
- SRC: module C_TEST_ROUTINES -scroll-source-----
83:     for(k=0;k<1000;k++)
84:     {
85:         test_c_code5(&i);
86:     }
87:     return;
88: }
89: int test_c_code3(int subrtnCount)
90: {
91:     subrtnCount = subrtnCount - 1;
92:     if (subrtnCount != 0)
93:         subrtnCount = test_c_code3(subrtnCount);
-> 94:     return subrtnCount;
95: }
96: int test_c_code2(__int64 in64,int in32, __int64 test, __int64* pVar)
97: {
98:     c_test_array[5] = in64;
99:     c_test_array[6] = in32;
100:    if (c_test_array[9] > 0)
101:        *pVar = (*pVar + c_test_array[17]) & c_test_array[9];
102:    else
103:        *pVar = (*pVar + c_test_array[17]);
104:    c_test_array[7] = test_c_code3(10);
105:    c_test_array[3] = test;
- OUT -output-----
*C_TEST_ROUTINES test_c_code3      93      0000000000000DE0 FFFFFFFF800BAFE0
*C_TEST_ROUTINES test_c_code2     104      0000000000000F40 FFFFFFFF800BB140
*C_TEST_ROUTINES test_c_code      117      00000000000010B0 FFFFFFFF800BB2B0
          XDT$INIT                  00000000000015C0 FFFFFFFF880955C0
*SYS$DOINIT      EXE$INITIALIZE    1973      0000000000000360 FFFFFFFF88094360
SHARE$EXEC_INIT_CODE0              0000000000005C240 FFFFFFFF803BB640
SHARE$EXEC_INIT_CODE0              00000000000057F20 FFFFFFFF803B7320
SHARE$EXEC_INIT_CODE0              00000000000047850 FFFFFFFF803A6C50
SHARE$EXEC_INIT_CODE0              00000000000042E90 FFFFFFFF803A2290
stepped to C_TEST_ROUTINES\test_c_code3\%LINE 94
stepped on return from C_TEST_ROUTINES\test_c_code3\%LINE 94 to C_TEST_ROUTINES -
\test_c_code3\%LINE 94+17
C_TEST_ROUTINES\test_c_code3\%R8:      0
- PROMPT -error-program-prompt-----
DBG> show calls
DBG> cancel break/all
DBG> go
DBG> step/return
DBG> examine r8
DBG>
```

例 11-16 の前にある最初の段落で、"for address 80002010" (アドレス 80002010 の) が削除となり、"this image or module" (このイメージまたはモジュール) が "INISBRK"に置き換えとなります。

現在の例 11-16 は削除となります。現在の例 11-17 が例 11-16 となり、次の内容に置き換えとなります。

```
- SRC: module C_TEST ROUTINES -scroll-source-----
83:   for(k=0;k<1000;k++)
84:   {
85:       test_c_code5(&i);
86:   }
87:   return;
88: }
89: int test_c_code3(int subrtnCount)
90: {
91:     subrtnCount = subrtnCount - 1;
92:     if (subrtnCount != 0)
93:         subrtnCount = test_c_code3(subrtnCount);
-> 94:     return subrtnCount;
95: }
96: int test_c_code2(__int64 in64,int in32, __int64 test, __int64* pVar)
97: {
98:     c_test_array[5] = in64;
99:     c_test_array[6] = in32;
100:    if (c_test_array[9] > 0)
101:        *pVar = (*pVar + c_test_array[17]) & c_test_array[9];
102:    else
103:        *pVar = (*pVar + c_test_array[17]);
104:    c_test_array[7] = test_c_code3(10);
105:    c_test_array[3] = test;
- OUT -output-----
SYS$UTC_SERVICES          no      0000000000000000    FFFFFFFFFFFFFFFF
SYS$VM                    no      0000000000000000    FFFFFFFFFFFFFFFF
SYS$XFCACHE_MON           no      0000000000000000    FFFFFFFFFFFFFFFF
SYSDEVICE                 no      0000000000000000    FFFFFFFFFFFFFFFF
SYSGETSYI                 no      0000000000000000    FFFFFFFFFFFFFFFF
SYSLDR_DYN                no      0000000000000000    FFFFFFFFFFFFFFFF
SYSLICENSE                no      0000000000000000    FFFFFFFFFFFFFFFF
SYSTEM_DEBUG              yes     0000000000000000    FFFFFFFFFFFFFFFF
SYSTEM_PRIMITIVES         no      0000000000000000    FFFFFFFFFFFFFFFF
SYSTEM_SYNCHRONIZATION    no      0000000000000000    FFFFFFFFFFFFFFFF

total images: 53
- PROMPT -error-program-prompt-----
DBG> go
%DEBUG-I-INIBRK, target system interrupted
%DEBUG-I-DYNIMGSET, setting image SYS$BASE_IMAGE %DEBUG-W-SCRNOSRCLIN, No source line -
for address: FFFFFFFF80000310
DBG> show image
DBG> go
```

4.29.37 OpenVMS Alpha システム・ダンプ・デバugga

第 12 章のタイトルは "OpenVMS System Dump Debugger" (OpenVMS システム・ダンプ・デバugga) となり, "Alpha Only" (Alpha のみ) という注記は削除となります。

第 12.4 項の最初の段落は, 次の内容に置き換えとなります。

SDD がこの検索パスでイメージのいずれかを見つけることができないと, 警告メッセージが表示されます。SDD は, 少なくとも 2 個のイメージを見つけるまで初期化を続行します。SDD が, OpenVMS オペレーティング・システムの主要なイメージ・ファイルである SYS\$BASE_IMAGE ファイルと SYS\$PUBLIC_VECTORS ファイルを見つけることができない場合, エラー・メッセージが表示され, デバugga が終了します。

第 12.10 項の手順 1 と 2 は次の内容に置き換えとなります。

1. 第 11.12 項の手順のうち, 例 11-9 までを実行します (SET MODE SCREEN コマンドを使用します)。
2. 次の OpenVMS Debugger コマンドを実行します。

```
DBG> set break test_c_code5
DBG> go
DBG> deposit k=0
DBG> go
```

次の新しい段落が, 例 12-1 の直前に追加となります。

例 12-1 以降の表示例は, すべて OpenVMS I64 システムでの例です。OpenVMS Alpha システムでは, 出力がいくらか異なりますが, 入力するコマンドはどちらのプラットフォームでも同じです。

例 12-1 で, "Alpha" は "I64" に変更となり, "V7.2-019" は "V8.3-003" に変更となります。

例 12-3 は, 次の内容に置き換えとなります。

```
DBG> Set Mode Screen; Set Step Nosource
```

```
- SRC: module C_TEST_ROUTINES -scroll-source-----
  67:
  68:      /* We want some global data cells */
  69: volatile __int64 c_test_array[34];
  70:
  71: void test_c_code5(int *k)
  72: {
  73:     int i;
  74:     char str[100];
  75:     for(i=0;i<100;i++)
  76:         str[i]= 'a';
  77:     str[99]=0;
-> 78:     *k = 9;
  79: }
  80: void test_c_code4(void)
  81: {
  82:     int i,k;
  83:     for(k=0;k<1000;k++)
  84:     {
  85:         test_c_code5(&i);
  86:     }
  87:     return;
  88: }
  89: int test_c_code3(int subrtnCount)
- OUT -output-----

- PROMPT -error-program-prompt-----

%DEBUG-I-SCRNOTORIGSRC, original version of source file not found for display in SRC
      file used is SYS$COMMON:[SYSHLP.EXAMPLES]C_TEST_ROUTINES.C;1
DBG>
```

例 12-4 は、次の内容に置き換えとなります。

```
- SRC: module C_TEST_ROUTINES -scroll-source-----
67:
68:      /* We want some global data cells */
69: volatile __int64 c_test_array[34];
70:
71: void test_c_code5(int *k)
72: {
73:     int i;
74:     char str[100];
75:     for(i=0;i<100;i++)
76:         str[i]= 'a';
77:     str[99]=0;
-> 78:     *k = 9;
79: }
80: void test_c_code4(void)
81: {
82:     int i,k;
83:     for(k=0;k<1000;k++)
84:     {
85:         test_c_code5(&i);
86:     }
87:     return;
88: }
89: int test_c_code3(int subrtnCount)

- OUT -output-----
C TEST ROUTINES\test_c_code5\i: 100
C TEST ROUTINES\test_c_code5\k: 0
  module name      routine name      line      rel PC      abs PC
*C TEST_ROUTINES  test_c_code5        78      000000000000CD0 FFFFFFFF800BAED0
*C TEST_ROUTINES  test_c_code4        85      000000000000D60 FFFFFFFF800BAF60
*C TEST_ROUTINES  test_c_code        118     00000000000010D0 FFFFFFFF800BB2D0
                  XDT$INIT          00000000000015C0 FFFFFFFF880955C0
*SYS$DOINIT       EXE$INITIALIZE    1973    0000000000000360 FFFFFFFF88094360
SHARE$EXEC_INIT_CODE0
0000000000005C240 FFFFFFFF803BB640
SHARE$EXEC_INIT_CODE0
00000000000057F20 FFFFFFFF803B7320
SHARE$EXEC_INIT_CODE0
00000000000047850 FFFFFFFF803A6C50
SHARE$EXEC_INIT_CODE0
00000000000042E90 FFFFFFFF803A2290
- PROMPT -error-program-prompt-----

%DEBUG-I-SCRNOTORIGSRC, original version of source file not found for display in SRC
      file used is SYS$COMMON:[SYSHLP.EXAMPLES]C_TEST_ROUTINES.C;1
DBG> examine i,k
DBG> show calls
DBG>
```

4.29.38 OpenVMS Alpha ウォッチポイント・ユーティリティ

第 III 部のページで, "Alpha Only" (Alpha のみ) という注記の内容は, 次の内容に置き換えとなります。

このユーティリティは, OpenVMS Alpha システムでのみ動作します。

第 13 章の最初のページで，"Alpha Only" (Alpha のみ) という注記の内容は，次の内容に置き換えとなります。

このユーティリティは， OpenVMS Alpha システムでのみ動作します。

4.29.39 SET PROCESS/LOG コマンド

フォーマットの項で， ARGV は ARGUMENTS に置き換えとなります。

パラメータ FLAGS の説明で， ARGV は ARGUMENTS に置き換えとなります。

これらの変更は『OpenVMS DCL ディクショナリ: N-Z』にも影響する点に注意してください。

プログラミングに関する注意事項

この章では、OpenVMS システムでのアプリケーション・プログラミングとシステム・プログラミングに関する注意事項について説明します。

5.1 システム・サービスの変更点

V8.3

ここでは、本リリースにおけるシステム・サービスの変更点について説明します。

5.1.1 追加

以下のシステム・サービスに対して追加と変更が行われています。

- \$GETJPI- アイテム・コード JPI\$_TERMINAL および JPI\$_SCHED_CLASS_NAME
- \$GETRMI- 説明を次のように変更。「\$GETRMI は、非同期システム・サービスであり、要求した情報が利用可能であることを保証するためには、\$SYNCH サービスまたは他の待ち状態同期メカニズムを使用する必要があります。このシステム・サービスの同期待ち形式はありません。」
- \$GETRMI のアイテム・コードを次のように変更。「RMI\$_MODES は、システムがブートされてから、現在アクティブな CPU が各プロセッサ・モードで消費した時間の量を 10 ms 単位で返します。あるモードに対して返される時間の値は、そのモードで CPU を費やした時間 (10 ms 単位) を表します。アクティブな CPU とは、OpenVMS インスタンスが実行するプロセッサのスケジューリングに参加している CPU のことです。詳細については、『OpenVMS System Services Reference Manual』の \$GETRMI システム・サービスを参照してください。」
- \$GETSYI- Intel Itanium 2 および Intel Itanium 3 用のプロセッサ・コードを追加
- \$GETSYI- 以下のアイテム・コードの定義を変更

```
SYI$_ACTIVE_CPU_MASK
SYI$_CPUCONF
SYI$_IO_PREFER_CPU
SYI$_POTENTIAL_CPU_MASK
SYI$_POWERED_CPU_MASK
SYI$_PRESENT_CPU_MASK
```

- \$INIT_VOL: いくつかのアイテム・コードを、ODS-2 だけでなく ODS-5 にも適用

5.2 システム・サービス\$GETQUI および\$SNDJBC のジョブ・リミット・アイテム・コードの範囲の拡大

V8.3

お客様からの要求に応え、バッチ環境の定義を簡単にするために、OpenVMS Version 8.3 では、2 つのシステム・サービスのジョブ・リミット・アイテム・コードが拡大されました。

- \$GETQUI システム・サービスでは、QUI\$_JOB_LIMIT アイテム・コードの範囲が 1 ~ 65535 になりました。
- \$SNDJBC システム・サービスでは、SJC\$_JOB_LIMIT アイテム・コードの範囲が 1 ~ 65535 になりました。

5.3 特権プログラムの再コンパイルが必要 (Alpha のみ)

V8.2

メジャー・バージョン・リリースである OpenVMS Alpha Version 8.2 では、多くの特権データ構造体に変更されています。このため OpenVMS の内部データ構造体やルーチンを参照する/SYSEXEC でリンクされた特権アプリケーションは、再コンパイルおよび再リンクを行う必要があります。

イメージの起動時やデバイス・ドライバのロード時に SYSVERDIF エラー・メッセージが出力された場合は、特権イメージや特権ドライバが、以前のバージョンのオペレーティング・システムでコンパイルおよびリンクされていることを示しています。このイメージやドライバを OpenVMS Alpha Version 8.2 で実行するためには、再コンパイルおよび再リンクが必要になります。

5.4 特権データ構造体の変更

V8.2

OpenVMS Version 8.2 では、多数の特権データ構造体に変更されています。これらの変更は、Alpha システムと I64 システムの両方で実施されています。これらのデータ構造体の変更の大半は、オペレーティング・システムでの将来のスケーリングおよび性能強化の計画を実現するためのものです。この変更の結果、ベース・オペレーティング・システムに対してリンクされているイメージやドライバ(つまり、LINK コマンドで/SYSEXEC を使用するもの)は、OpenVMS Version 8.2 で動作させるために再コンパイルと再リンクが必要となります。

特権データ構造体の変更は、すべての特権イメージやドライバに影響するとは限りません。変更されたサブシステムにリンクされているイメージやドライバだけが影響を受けます。変更されたサブシステムに関しては、サブシステムに対応するメジャー・バージョン識別番号が変更されています。変更されたサブシステムは、次のとおりです。

```
SYSSK_IO  
SYSSK_MEMORY_MANAGEMENT  
SYSSK_CLUSTERS_LOCKMGR  
SYSSK_FILES_VOLUMES  
SYSSK_CPU  
SYSSK_MULTI_PROCESSING  
SYSSK_PROCESS_SCHED
```

注意: I64 システムでは、これらのサブシステムは SYSSK_VERSION_xxxx のように表示されます。

これらのサブシステムは、各種の特権システム・ルーチンおよびデータ・セルを使用しているイメージのリンク時に、そのサブシステムのバージョン番号がイメージに記録されています。ANALYZE/IMAGE ユーティリティを使用すると、特権イメージがどの特権サブシステムにリンクされているかを調べることができます。例を次に示します。

```
$ ANALYZE/IMAGE IMAGE.EXE /OUTPUT=IMAGE.TXT  
$ SEARCH IMAGE.TXT "SYSSK_"
```

変更されたサブシステムが表示された場合、OpenVMS Version 8.2 は、イメージの実行に失敗し、オペレーティング・システムの以前のバージョンにリンクされたイメージに対して SSS_SYSVERDIF (システム・バージョン不一致エラー) を出力します。

5.4.1 KPB 拡張

V8.2

OpenVMS の以前のバージョンでは、IPL 2 より上のカーネル・モードでの KPB の使用をサポートしていました。I64 への移行を簡単にするために、KPB の使用が、アウター・モードとすべての IPL に拡張されました。この Alpha と I64 での変更により、Alpha と I64 の両方で、以前はプライベート・スレッド・パッケージを持っていた一部のコードが、KPB を使用できるようになりました。カーネル・プロセスでのこれらの変更をサポートするために、KPB 構造体に変更を行う必要がありました。既存の Alpha コードでは、ソース・プログラムを変更する必要はありません。

5.4.2 CPU の名前空間

V8.2

OpenVMS には現在、最大の CPU ID が 31 であるという、アーキテクチャ上の制限があります。各種の内部データ構造体およびデータ・セルでは、CPU マスクに 32 ビットを割り当てています。将来のリリースでより多くの CPU ID をサポートできるように、これらのマスクに割り当てられるスペースを、Alpha では 64 ビット、I64 では 1024 ビットに増やしました。既存のロングワードの CPU マスクのシンボルおよびデータ・セルは、今後も維持されます。

当面は、特権イメージおよびドライバには影響はありません。ただし、将来的には、CPU マスクを参照する特権付き製品が、31 個より多くの CPU ID を持つシステムをサポートするために、コードをどのように変更しなければならないかを明記する予定です。

5.4.3 64 ビットの論理ブロック番号 (LBN)

V8.2

OpenVMS はこれまで 31 ビットの LBN をサポートしています。このため、ディスク・ボリュームのサポートが 1 TB に制限されます。内部の LBN フィールドに割り当てられるスペースが 64 ビットに拡張されており、より大きなディスク・ボリュームを将来サポートできるようになります。既存のロングワードの LBN シンボルは今後も維持され、クォドワード・シンボルでオーバーレイされます。

5.4.4 動的スピンロックのフォーク

V8.2

OpenVMS オペレーティング・システムを大規模な SMP システムへ拡張するために、オペレーティング・システム内の多くの部分で、限られた数の静的スピンロックではなく、動的スピンロックを使用するようになっていきます。フォークする機能と、フォーク・ディスパッチャが動的スピンロックと同期をとる機能が必要とされます。FKB 構造体のサイズを拡張し、FKBSL_SPINLOCK フィールドをこの構造体の最後に追加することによって、この機能を OpenVMS Version 8.2 に追加しています。このスピンロック・フィールドは、FKBSB_FLCK に値 SPL\$C_DYNAMIC が設定されている場合にのみ参照されます。FKB 構造体は、他の多くのシステム・データ構造体に埋め込まれているため、この変更は、多数の特権データ構造体のサイズとレイアウトに影響します。

FKBSB_FLCK フィールドを、OpenVMS が作成した構造体から他の FKB ヘコピーするアプリケーションは、FKBSL_SPINLOCK フィールド内のデータもコピーする必要があります。

FKB 構造体を割り当て、ハードコード化された値 32 をサイズとして使用していないか、特権コードをチェックしてください。コードでは、FKB のサイズとして、シンボル FKB\$C_LENGTH を使用しなければなりません。

5.4.5 UCB と DDB のアップデート

V8.2

UCB 構造体と DDB 構造体には、多数のアップデートが行われました。

DDB に対応する UCB のリストは、現在は単方向リストです。UCB を作成したり削除する場合は、適切な位置が見つかるまで、このリストをたどらなければなりません。OpenVMS Version 8.2 では、UCB は双方向リストで DDB にリンクされるようになりました。さらに DDB は、新しいユニットを作成するときに検索を開始する位置を示すシード・ポインタを維持しており、デバイスの作成が速くなります。テンプレート UCB 内でユニット・シード・ポインタを操作しているドライバには、デバイスの作成が速くなるという利点はありません。

DDB の UCB リストを操作するコードは、正しく動作しなくなります。UCB のリンクとリンク解除には、提供されている内部ルーチンを使用してください。UCB リストを前方にたどるコードは、引き続き正しく動作します。

UCB\$W_UNIT フィールドは現在、16 ビット・ワード・フィールドです。このフィールドには、32 ビットが割り当てられるようになりました。UCB\$W_UNIT フィールドは引き続き維持されるため、ソース・コードの変更は必要ありません。今後のリリースでは、OpenVMS はより大きいユニット番号をサポートする可能性があります。この機能をサポートできるドライバに対してだけ、この変更が行われる予定です。

ターミナル・ドライバの UCB 拡張内のバイト・フィールドとワード・フィールドは、ロングワード境界に揃えられるようになりました。

5.4.6 PCB\$T_TERMINAL のサイズの拡張

V8.2

Process Control Block (PCB) 構造体には、PCB\$T_TERMINAL というフィールドが含まれています。このフィールドは 8 バイトで、会話型プロセスのデバイス名 (LTA123:, RTA7:, NVA456: など) を保持します。このフィールドは、文字数付きの ASCII 文字列で、1 バイト目が文字列の長さ、残りの 7 バイトがデバイス名です。デバイス名が 3 文字の場合、ユニット番号には 4 桁しか使用できないため、999 より大きいユニット番号ではコロンは取り除かれます。OpenVMS Version 8.2 では、このフィールドは 16 バイトに拡張されたため、より大きなユニット番号のデバイス名を保持できるようになりました。

JPI\$_TERMINAL アイテム・コードを指定して\$GETJPI を呼び出してこのフィールドをフェッチする場合、特に影響はありません。ただし、システム・サービスに渡すバッファを、最大 16 バイト保持できるように拡張しても構いません。

5.4.7 スレッド単位のセキュリティは特権付きコードとデバイス・ドライバに影響する

永続的な変更

セキュリティ・プロファイルを I/O Request packet (IRP) に添付するために使用する方法が、Version 7.2 で変更されました。

Version 7.2 より前のバージョンの OpenVMS では、IRP 構造体には、要求者のプロセス単位の Access Rights Block (ARB) セキュリティ構造体のアドレスが含まれていました。OpenVMS Alpha Version 7.2 以降、新しいセキュリティ・プロファイル構造体 (Persona Security Block (PSB)) のアドレスが、ARB アドレスの機能置換として、IRP に追加されました。

I/O サブシステムは PSB へのアクセスを、PSB 内のリファレンス・カウンタを通して管理します。I/O サブシステムは、このリファレンス・カウンタを、IRP の作成時にカウントアップし、IRP の I/O 後処理時にカウントダウンします。このカウンタが 0 になったとき、PSB 構造体は割り当て解除されます。

1 つの要求に対して複数の I/O 操作を行うために IRP のコピーを作成またはクローン化して、コピーした IRP を後処理のために I/O サブシステムに渡すデバイス・ドライバは、コードを変更して、追加した IRP 内の PSB への余分な参照に対処しなければなりません。この処理は、コピーされた IRP 内の PSB アドレスを、NSA_STD\$REFERENCE_PSB ルーチンに渡すことで行います。インクルード・ファイルと、NSA_STD\$REFERENCE_PSB の呼び出しは、次のとおりです。

```
#include <security-macros.h>

/* Increment REFCNT of PSB that is now shared by both IRPs */
nsa_std$reference_psb( irp->irp$ar_psb );
```

デバイス・ドライバは、次の状況でこの変更を行わなければなりません。

- デバイス・ドライバが次の状態の場合
 1. デバイス・ドライバが、既存の IRP を複製することで新しい IRP を作成する場合、かつ
 2. IOC_STD\$SIMREQCOM または IOC_STD\$DIRPOST1 を呼び出すことで、I/O 後処理のためにオリジナル IRP と複製 IRP の両方をキューに登録する場合デバイス・ドライバは IRP を複製した後、I/O 後処理のためにキューに登録する前に、NSA_STD\$REFERENCE_PSB を呼び出さなければなりません。

- デバイス・ドライバが次の状態の場合
 1. デバイス・ドライバが、既存の IRP を複製することで新しい IRP を作成する場合、かつ
 2. コピーまたはオリジナル IRP の IRPSL_PID セルにプロシージャ記述子のアドレスを格納しない場合、かつ
 3. IOC_STD\$REQCOM, COM_STD\$POST, COM_STD\$POST_NOCNT, IOC_STD\$POST_IRP を呼び出すことで、I/O 後処理のためにオリジナル IRP と複製 IRP の両方をキューに登録する場合

デバイス・ドライバは IRP を複製した後、I/O 後処理のためにキューに登録する前に、NSA_STD\$REFERENCE_PSB を呼び出さなければなりません。

これらのステップを実行するデバイス・ドライバは、たいていはプロシージャ記述子のアドレスを IRPSL_PID に格納しています。したがって、IRP を複製するほとんどのデバイス・ドライバは、ソース・コードの変更、再リンク、再コンパイルを行わなくても、OpenVMS Version 7.2 以降で正しく機能するはずです。

この状態で NSA_STD\$REFERENCE_PSB を呼び出さないと、PSB 内のトラッキング情報が壊れ、システム障害となることがあります。

NSA_STD\$REFERENCE_PSB を呼び出すようにデバイス・ドライバのコードを変更する場合は、OpenVMS Version 7.2 またはそれ以降で動作するように、ドライバを再コンパイルおよび再リンクしなければなりません。

OpenVMS フォーク実行スレッドを作成するためには、いくつかのルーチンを特権コードで使います。これらのルーチンは、プロセスとは独立にシステム・コンテキストで実行されます。これらのルーチンには 4 つの形式があり、どの形式を使用するかは、直系のフォークとキューに入れられるフォークのどちらが必要かと、使用している言語インタフェースで決まります。

- EXE\$QUEUE_FORK
- EXE_STD\$QUEUE_FORK
- EXE\$PRIMITIVE_FORK
- EXE_STD\$PRIMITIVE_FORK

これらのルーチンは、実行中に、誤って別の CPU に再スケジュールされないようにするため、IPL\$_RESCHED 以上で呼び出す必要があります。このような再スケジュールが行われると、システムがハングする可能性があります。

OpenVMS V7.3-1 では、SYSTEM_CHECK の値を 1 にすると、これらのルーチンによって、まずシステムの IPL がチェックされます。IPL が IPL\$_RESCHED の値よりも小さい場合、システムは SPLINVIPL バグ・チェックで失敗します。

性能上の理由から、SYSTEM_CHECK の値を 0 にすると (デフォルト)、IPL は検証されません。不正なコードを使用すると、プロセス・コンテキストでこれらのルーチンの実行中に別の CPU への再スケジュールが発生したときに (IPL\$ RESCHED よりも小さい値を指定した場合など)、システムがハングする可能性があります。

5.5 浮動小数点型データを使用するアプリケーション

V8.2

Itanium®アーキテクチャでは、IEEE 単精度および IEEE 倍精度を含む、IEEE 浮動小数点形式を用いた、ハードウェアによる浮動小数点演算を実装しています。

Alpha ハードウェアでは、IEEE 浮動小数点形式と VAX 浮動小数点形式のいずれもサポートしています。OpenVMS Alpha では、コンパイラはデフォルトで VAX 形式のコードを生成し、オプションで IEEE 形式のコードを生成します。

OpenVMS I64 では、コンパイラはデフォルトで IEEE 形式のコードを生成し、オプションで VAX 形式のコードを生成します。Integrity サーバでは、VAX や Alpha システムで生成された VAX 形式の浮動小数点バイナリ・データをアプリケーションで取り扱う必要がある場合を除き、IEEE 形式を使用することをお勧めします。OpenVMS I64 で VAX 形式を使用する場合の詳細については、次の Web サイトのホワイト・ペーパー『Intel® Itanium®における OpenVMS 浮動小数点演算について』を参照してください。

<http://h50146.www5.hp.com/products/software/oe/openvms/technical/>

5.5.1 IEEE 浮動小数点フィルタ (I64 のみ)

V8.3

浮動小数点例外が IEEE-Std 754-1985 に完全に準拠するように、Intel では、IEEE フィルタと呼ばれる関数を提供しています。この関数を使用したいアプリケーション開発者は、通常の OpenVMS 例外ハンドラの中からこの関数を呼び出すことができます。例外が発生すると、このフィルタが、例外の原因となった浮動小数点命令、IEEE の丸めモードと精度をデコードし、例外の原因となったオペランドを特定することができます。

このフィルタのコピーを入手するには、以下の Intel の Web サイトにアクセスし、OpenVMS の見出しを探してください。

<http://www.intel.com/cd/software/products/asmo-na/eng/219748.htm>

このサイトにあるアプリケーション・ノートでは、フィルタについて詳細に説明しています。ソース・コード例とフィルタのオブジェクト・ライブラリは、OpenVMS のバックアップ・セーブ・セットとして提供されています。

このフィルタは、浮動小数点例外の詳細を IEEE 規格に準拠させるためにのみ必要です。通常の浮動小数点演算には必要ありません。

5.5.2 Ctrl/C と STOP ボタンを使用する場合の制限 (OpenVMS Alpha)

V8.3

問題: デバッガの制御下で実行されているプログラムでは、Ctrl/C による割り込みは 1 回しか動作しません。以後は、Ctrl/C による割り込みは無視されます。DECwindows の STOP ボタンも同様です。最初にこのボタンを押したときだけその動作が認識されます。

回避策: なし。

5.5.3 Ada イベントのサポート (I64 のみ)

V8.3

Ada イベントのサポート (SET BREAK/EVENT=ada_event。ただし、ada_eventは SHOW EVENT で説明されるイベントのいずれか) が、OpenVMS I64 で有効になりました。ただし、このサポートは不完全です。

イベント・ブレークポイントで問題が発生した場合は、回避策として、pthread イベント (SET EVENT_FACILITY pthread) に切り替えてください。すべての Ada イベントに同等の pthread 機能があるわけではありません。

5.5.4 SHOW SYMBOL/TYPE がアレイ・サイズを正しく報告するようになった (Alpha および I64)

V8.3

SHOW SYMBOL コマンドは、アレイ・タイプの全体サイズを正しく報告するようになりました。以前の SHOW SYMBOL/TYPE の動作では、次のように、誤ったサイズが報告されていました。例を次に示します。

```
DBG> sho sym/type z
data MATINV\Z
    noncontiguous array descriptor type, 2 dimensions, bounds: [0:10,0:10], size: 4 bytes
    cell type: atomic type, IEEE single precision floating point, size: 4 bytes
DBG>
```

正しい動作は、次のとおりです。

```
DBG> sho symbol/type z
data MATINV\MATINV\Z
    noncontiguous array descriptor type, 2 dimensions, bounds: [0:10,0:10], size: 484 bytes
    cell type: atomic type, IEEE single precision floating point, size: 4 bytes
DBG>
```

5.5.5 EXAMINE/INSTRUCTION %PREVLOC コマンドが修正された (I64 のみ)

V8.3

EXAMINE/INSTRUCTION %PREVLOC コマンド (および代替の EXAMINE /INSTRUCTION ^コマンド) は、期待したとおりに動作するようになりました。以前はデバッガが PC をデクリメントしなかったため、同じ命令が表示されていました。

5.5.6 SHOW MODULE コマンドがモジュール・サイズを計算するようになった (I64 のみ)

V8.3

SHOW MODULE コマンドは、モジュールのサイズとしてゼロでない値を報告するようになりました。この値は推定値なので、モジュールの相対サイズを比較する際のめやすとしてだけ使用してください。

5.5.7 C++ 言語の問題 (I64 のみ)

V8.2

問題: SHOW CALLS コマンドは、C++ のデマングル化された名前ではなく、C++ のマングル化された名前を表示することがあります。

回避策: なし。

V8.3

問題: デバッガは、/OPTIMIZE でコンパイルされた C++ プログラムのデバッグをサポートしていません。

回避策: C++ プログラムを /NOOPTIMIZE でコンパイルしてください。

5.6 Ada コンパイラ (I64 のみ)

V8.2

GNAT Pro (Ada 95) は、AdaCore から入手できます。詳細は、www.adacore.com または sales@adacore.com の AdaCore にお問い合わせください。

5.7 Backup API: ジャーナリング・コールバック・イベントの制限事項

永続的な制限事項

アプリケーションがジャーナリング・イベントのいずれかに対してコールバック・ルーチンを登録する場合は、すべてのジャーナリング・コールバック・イベントに対してコールバック・ルーチンを登録しなければなりません。ジャーナリング・コールバック・イベントは次のとおりです。

```
BCK_EVENT_K_JOURNAL_OPEN  
BCK_EVENT_K_JOURNAL_WRITE  
BCK_EVENT_K_JOURNAL_CLOSE
```

コールバック・ルーチンの登録の詳細については、『HP OpenVMS Utility Routines Manual』の Backup API に関する章を参照してください。

5.8 C プログラム: CASE_LOOKUP=SENSITIVE を設定したコンパイラ

永続的な制限事項

特性として CASE_LOOKUP=CASE=SENSITIVE が設定されているプロセスで C プログラムをコンパイルすると、.h ファイル・タイプ (小文字の「h」) で指定された C プログラム内の #include ファイルは、検出および実行されません。また、システムの #include ファイルが他の .h ファイル・タイプの #include ファイルを使用している場合、この #include ファイルは検出されず、エラーが出力されます。

この動作を防ぐには、大文字と小文字を区別しないように設定します。case=sensitive を設定する必要がある場合は、C プログラム内の #include ファイルにファイル・タイプを指定しないか (例 #include <stdio>)、または大文字の H ファイル・タイプを指定してください (例 #include <stdio.H>)。

ただし、stdlib.h などのシステム・インクルード・ファイルが、その中から .h ファイル・タイプのインクルード・ファイルを使用している場合は、エラーを回避できませんので注意してください。

5.9 C ランタイム・ライブラリ

ここでは、C ランタイム・ライブラリ (RTL) の変更や修正について説明します。

5.9.1 C RTL TCP/IP ヘッド・ファイルのアップデート

V8.3

C RTL では、TCP/IP を呼び出すユーザ向けのヘッド・ファイルを出荷しています。これらのヘッドには多くの問題があるため、簡易 TCP/IP プログラミングの域を越えて使用できないものがあります。

以前のいくつかの TCP/IP リリースでは、訂正済みのヘッドがプログラミング例と同じ場所で提供されていました。OpenVMS Version 8.3 では、訂正済みのヘッドが、C RTL のヘッド・ライブラリ (DECC\$RTLDEF.TLB) に置かれています。

このファイルへの変更の例としては、以下のものがあります。

- メンバのアラインメントが訂正されました。この訂正により、ヘッドに定義されている構造体が、必要とされているパック化されたアラインメントに戻されました。
- 一部の定数マクロ定義が、TCP/IP の構築で使用される定数と正しく一致するように変更されました。
- SOCKET.H では、MSG_DONTWAIT と MSG_DONTROUTE の値が正しくなるように入れ替えられました。この問題を回避するようにコーディングしていたユーザは、この変更によるハングアップを避けるために、変更を行わなければならないことがあります。
- 古いプログラムはデータを TCP/IP に正しく渡していなかったため、以前は、古いヘッドでコンパイルされた古いプログラムとの互換性はありませんでした。ユーザが TCP/IP と通信できるように、データ構造体に変更されました。これらの変更を行わなければ、このヘッド・ファイルは使用できませんでした。ユーザは、新しい定義を利用するために再コンパイルを行う必要があります。
- 製品が準拠している TCP/IP の標準で、ヘッド・ファイルの変更が必要でした。
- TCP/IP の新しい機能に合わせて、新しい型、マクロ、およびヘッド・ファイルが追加されました。

以下の C RTL ヘッド・ファイルが変更されました。

BITYPES.H	IN.H	NAMESER_COMPAT.H	RESOLV.H
IF.H	IN6.H	NETDB.H	SOCKET.H
IF_ARP.H	INET.H	PCAP-BPF.H	STROPTS.H
IF_TYPES.H	NAMESER.H	PCAP.H	TCP.H

5.9.2 toascii 関数の追加

V8.3

X/Open 仕様で必要とされているtoascii関数は、<ctype.h>内でマクロとして定義されているだけでした。

本リリースでは、toascii関数のエントリ・ポイントが DECC\$SHR イメージに追加されています。

5.9.3 64 ビットの sigaction の問題の修正

V8.3

sigaction関数に対して 64 ビット・ポインタを使用すると、コンパイル時に警告が出力されていました。

この修正により、通常のコンパイル、long ポインタのコンパイル (/POINTER=LONG)、および short ポインタのコンパイル (/POINT=SHORT) が可能になりました。

5.9.4 一部の算術関数への、64 ビット・ポインタ機能の追加

V8.3

/POINTER_SIZE 修飾子を指定してコンパイルすることで、以下の C RTL 算術関数に 64 ビット・ポインタを渡せるようになりました。

frexp	modf
frexpf	modff
frexpl	modfl

5.9.5 2 GB の malloc で何も表示せずに失敗することがなくなった

V8.3

C RTL のmalloc関数は、unsigned int (size_t) をパラメータとして受け付けます。lib\$vm_mallocは、正の符号付き整数をパラメータとして受け付けます。

以前のリリースでは 2 GB (0x80000000) のメモリを割り当てようとすると、正しい量のメモリは割り当てられず、エラー通知も返されませんでした。本リリースでは、関数malloc、calloc、およびreallocにチェックが追加され、2 GB 以上のサイズを要求すると呼び出しが失敗するようになりました。

5.9.6 exec*のメモリ・リークの修正

V8.3

exec*のメモリ・リークが修正されました。

5.9.7 execl 失敗後の exit 動作の修正

V8.3

C RTL には、execlが失敗した後に_exitを呼び出して終了してはならないのに、呼び出して終了するという問題に対する修正が含まれています。

OpenVMS のvforkの実装では、子プロセスを起動する大半の UNIX システムと異なり、実際の子プロセスは起動されません。ただし、C RTL は、子プロセスの機能をまねるために、「子コンテキスト」という内部データ構造体を作成します。

vfork後、子コンテキストの中にいるときにexec関数の呼び出しが失敗すると、_exitが呼び出されることでバグが発生します。UNIX システムでは、exec呼び出しの失敗後、子プロセスは引き続き動作します。これ以降の_exitの呼び出しで、子プロセスが終了します。OpenVMS の実装では、exec呼び出しの失敗後、子コンテキストは終了します。これ以降の_exitの呼び出しでは、親プロセスが終了します。

C RTL の修正は、機能論理名スイッチ DECC\$EXIT_AFTER_FAILED_EXEC で有効になります。この機能論理名を有効にすると、子コンテキストが引き続き動作します。

DECC\$EXIT_AFTER_FAILED_EXEC が無効か定義されていない場合、現在の動作のままとなります。

5.9.8 confstr の拡張

V8.3

X/Open 仕様に準拠するために、confstr関数に長さがゼロのバッファを渡すことができるようになりました。

また、confstrは、<unistd.h>ヘッダ・ファイルに追加された、次の3つの HP-UX シンボリック定数をサポートするようになりました。

```
_CS_MACHINE_IDENT  
_CS_PARTITION_IDENT  
_CS_MACHINE_SERIAL
```

5.9.9 fopen の失敗の修正

V8.3

以前は、ユーザが `fopen(file, "wb+")` を呼び出すと、`file` が、引用符で囲まれた DECNET 指定で、リモート・ファイルが存在しなかった場合、RMS はファイルをアップデート用にオープンする (新しいバージョンを作成する) のではなく、ファイル指定の構文エラーを報告していました。

この問題は、バージョン 8.3 で修正されました。

5.9.10 ファイル・ポインタ・ロックのハングアップの修正

V8.3

OpenVMS Version 8.2 では、マルチスレッドの C プログラムに対して、ファイル・ポインタ・ロック (`flockfile` およびそのシリーズ) を導入しました。C RTL は、X/Open 仕様に準拠するために、ファイル・ポインタを内部的にロックします。

データの事前ロード中のエラー・リターンによりファイル・ポインタのロック解除が行われなかった場合、問題が発生します。

この問題が発生すると、アプリケーションは、ファイル・ポインタ I/O を使用するマルチスレッド・プログラム内でハングアップすることがあります。プログラムの状態を分析すると、ファイル・ポインタ・ロックの実装に使用されている TIS 再帰ミューテックス上でスレッドがデッドロックしていることが分かります。

この問題は、バージョン 8.3 で修正されました。

5.9.11 バックポート・ライブラリが出荷されなくなった

V8.3

以前は、古いバージョンの OpenVMS で開発者が最新の C RTL 関数を使用できるように、コンパイラ配布キットに C RTL バックポート・オブジェクト・ライブラリが含まれていました。このバックポート・オブジェクト・ライブラリは、出荷されなくなりました。

5.9.12 ヘッダ・ファイル<time.h>の変更

V8.3

以下で説明している問題が修正されました。この問題がまだ発生している場合は、正しい動作が行われるように、アプリケーションを再コンパイルする必要があります。

C RTL <time.h>ヘッダ・ファイルでは、struct tm構造体と、関数gmtime、localtime、gmtime_r、およびlocaltime_rが定義されています。

これらの関数のいずれかを呼び出したときに、アプリケーションとC RTLが認識しているstruct tmのサイズが異なると、ユーザ・アプリケーションでデータが壊れるか、アクセス違反が発生することがあります。

tm構造体には、BSD 拡張用のオプションのメンバtm_gmtoffおよびtm_zoneが入っています。これらのメンバは、ANSI 仕様やPOSIX 仕様では、古いコンパイルとの互換性のため、定義されていません。

C RTL では、上記で説明した関数に対して、3 種類の異なる定義があります。

- ローカル・タイム (プレフィックスなし)
- UTC 時間 (OpenVMS V7.0 以降)
 - プレフィックス__UTC_ (BSD 拡張なし)
 - プレフィックス__UTCTZ_ (BSD 拡張を使用)

プレフィックス__UTCTZ_付きの関数は、BSD 拡張が定義されている長いtm構造体だけを割り当てることを前提とします。

<time.h>ヘッダ・ファイルとC RTLで、認識しているstruct tm構造体のメンバの数が異なっていると、問題が発生します。たとえば、_ANSI_C_SOURCEを意味するコンパイル時マクロ(_POSIX_C_SOURCEなど)を使用してC++コンパイルを行うと、プレフィックス__UTC_を使用して上記のC RTL関数がマッピングされるため、問題が発生します。関数は短いtmデータ構造体を期待しますが、ユーザ・プログラムは長いtm構造体定義を使用します。関数から戻る際に行われるデータのコピー・バックで、C RTLが割り当てていないtmデータ・メンバへのアクセスが試みられます。これにより、意図していないメモリの使用やアクセス違反など、予期しない動作が発生することがあります。

OpenVMS Version 8.3 では、<time.h>が変更され、上記の関数に対してC RTLが適切なプレフィックスを選択するようになりました。

5.9.13 ヘッダ・ファイル<time.h>での非ANSIの*_r関数の参照

V8.3

X/Open 仕様で定義されているctime_r、gmtime_r、およびlocaltime_rというC RTL関数はISO/ANSI C 標準では規定されておらず、厳密にANSI 準拠のコンパイルでは参照できないようになっていなければなりません。以前のC RTLでは、これらの関数を参照できていました。

この問題は修正されました。<time.h>ヘッダにチェックが追加され、これらの関数は、ANSI 準拠のコンパイルでないときのみ参照できるようになりました。

5.9.14 ヘッダ・ファイル<decc\$types.h>: time_t int 宣言

V8.3

OpenVMS システムでは、ヘッダ・ファイル<decc\$types.h>で定義されているtime_t構造体は、従来からunsigned long intとして宣言されていました。UNIX プラットフォームではこれを符号付きの型として宣言することが多く、OpenVMS 上に移植されたプログラムで問題が発生する可能性があります。

UNIX との互換性のために、コンパイル・マクロ__SIGNED_INT_TIME_Tを使用し、time_tをintとして宣言することもできるように修正されました。

古いプログラムとの互換性を保つために、デフォルトはunsigned long intのままです。

5.9.15 新しいDECC\$SHRP.EXE イメージ

V8.3

保護モードを必要とする C RTL 関数を実装するために、OpenVMS は新しい共用イメージ DECC\$SHRP.EXE をインストールします。この共用イメージは、すべての Alpha プロセッサと Integrity プロセッサにインストールされ、DECC\$SHR.EXE 共用イメージまたは DECC\$SHR_EV56.EXE 共用イメージから呼び出されます。

5.9.16 ヘッダ・ファイル<wchar.h>と C++ %CXX-W-ENVIRSTKDIRTY メッセージ

V8.3

<wchar.h>ヘッダ・ファイルは、_XOPEN_SOURCE マクロが定義されている状態でコンパイルしたときに C++ Version 7.1 コンパイラが%CXX-W-ENVIRSTKDIRTY 警告を出力するという問題を解決するために、修正されました。

5.9.17 ヘッダ・ファイル<builtins.h>の__CMP_SWAP*と_Interlocked*が C++ から参照可能

V8.3

<builtins.h>にある比較とスワップのビルトイン (__CMP_SWAP*と_Interlocked*) は、OpenVMS Alpha C++ コンパイラを対象としていませんでした。HP C++ Version 7.1 はこれらの関数を必要とするため、条件付きコンパイルが変更されこれらのビルトインを参照できるようになりました。

5.9.18 fcntl への余分なパラメータの無視

V8.3

期待されているパラメータが 2 個だけのとき (3 番目のパラメータを期待しないコマンド) に、オプションの 3 番目のパラメータを指定して `fcntl` を呼び出すと、以前はエラーが返されていました。

この問題は修正されました。必要のない 3 番目のパラメータは、無視されるようになりました。

5.9.19 システムの MAXBUF パラメータが大きい場合、stdout への fwrite が失敗する問題

V8.3

システムの MAXBUF SYSGEN パラメータが 33278 以上の場合、stdout への `fwrite` を使用すると、以前はエラーとなっていました。

失敗した `fwrite` 呼び出しの後に `perror` を使用すると、次のメッセージ (`perror` で表示されるとおり) が返されていることがあります。

```
Error writing output: : non-translatable vms error code: 0x186A4
%rms-f-rsz, invalid record size
```

この問題は修正されました。

5.9.20 64K バイトを超えるソケット転送の読み取りと書き込みの問題

V8.3

以下のソケット・ルーチンに対し、64K バイトを超えるソケット転送のサポートが追加されました。

<code>send</code>	<code>recv</code>	<code>read</code>
<code>sendto</code>	<code>recvfrom</code>	<code>write</code>
<code>sendmsg</code>	<code>recvmsg</code>	

5.9.21 I64 システムでの nanosleep の問題

V8.3

OpenVMS I64 システムでは、`nanosleep` 関数は正しくスリープしても、不正なステータス -1 を返し、`errno` に `EINTR` (割り込まれたシステム・コール) を設定していました。

この問題は修正されました。

5.9.22 l64 システムへのビルトイン __fci の追加

V8.3

<builtins.h>ヘッダ・ファイルがアップデートされ、新しいビルトイン __fci (fc.i 命令用のビルトイン) のプロトタイプが追加され、HP C コンパイラでサポートされるようになりました。

5.9.23 _FAST_Toupper マクロの追加

V8.3

OpenVMS Version 8.3 から、C99 ANSI 標準および X/Open 規格に準拠するため、_tolower マクロと _toupper マクロはデフォルトではパラメータを 2 回以上評価しないようになりました。これらのマクロは、単にそれぞれ tolower 関数と toupper 関数を呼び出します。これにより、式を評価する回数によって結果が異なるような副作用 (i++ や関数呼び出しの副作用など) を避けることができます。

_tolower マクロと _toupper マクロで、以前の最適化された動作を行うには、/DEFINE=_FAST_Toupper を指定してコンパイルします。これにより、以前のリリースと同様に、これらのマクロは実行時呼び出しのオーバーヘッドを避けるように呼び出しが最適化されます。ただし、結果を計算する方法を決定するために、マクロのパラメータは 2 回以上評価されます。これにより、望まない副作用が発生する可能性があります。

5.9.24 atof("NaN") の呼び出しで算術トラップが生成されなくなった

V8.3

OpenVMS Alpha Version 8.2 以降では、atof("NaN") に対する次の呼び出しで算術トラップが生成されます。

```
d = atof("NaN");  
%SYSTEM-F-HPARITH, high performance arithmetic trap, Imask=00000000,  
Fmask=00000002, summary=02, PC=FFFFFFFF80BB23D4, PS=0000001B  
-SYSTEM-F-FLTINV, floating invalid operation, PC=FFFFFFFF80BB23D4,  
PS=0000001B
```

この問題は OpenVMS Version 8.3 で修正され、算術トラップは生成されなくなりました。

5.9.25 DECC\$.OLB オブジェクト・ライブラリに新しいエントリがない

V8.3

OpenVMS Alpha および OpenVMS I64 の Version 8.2 から、DECC\$.OLB オブジェクト・ライブラリに新しいエントリ・ポイントが追加されなくなりました。これは、新しい C RTL エントリ・ポイントはこれらのライブラリを通じてプレフィックス付きのエントリにマッピングされないことを意味します。たとえば、新しい OpenVMS Version 8.3 のエントリ・ポイント `crypt` は、`/PREFIX=NONE` 付きでコンパイルすると、`crypt` から `decc$crypt` にマッピングされません。

5.10 呼び出し標準規則とローテートするレジスタ (I64 のみ)

V8.3

ここでの説明は『HP OpenVMS Calling Standard』の情報を補足するものです。

呼び出し標準規則の ICB (invocation context block) (『HP OpenVMS Calling Standard』の表 4-16 を参照) およびメカニズム・ベクタ (『HP OpenVMS Calling Standard』の図 8-7 と表 8-6 を参照) は常に、あたかも、レジスタ・リネーム・ベース (CFM.rrb) とローテート・サイズ (CFM.sor) がいずれも 0 であったかのように、汎用レジスタ、浮動小数点レジスタ、およびプレディケート・レジスタを記録しています。つまり、ローテートするレジスタを使用しているときに、ローテーションの効果が無視されます。このことは、`LIB$I64_PUT_INVO_REGISTERS` ルーチン (『HP OpenVMS Calling Standard』の第 4.8.3.13 項を参照) で使用するレジスタ・マスクについても同様です。というのは、これらのマスクは、ICB 構造体のフィールドによって定義されるからです。

以前は、補足的なアクセス・ルーチン `LIB$I64_GET_FR`、`LIB$I64_SET_FR`、`LIB$I64_GET_GR` および `LIB$I64_SET_GR` (『HP OpenVMS Calling Standard』の第 4.8.4 項を参照) が、レジスタ・リネーム・ベース・レジスタとローテート・サイズ・レジスタの効果を調整しないで、不適切に、そのレジスタ番号パラメータを解釈していました。この誤りは修正されました。

5.11 Common Data Security Architecture (CDSA) に関する考慮

ここでは、CDSA に関する注意事項について説明します。

5.11.1 Secure Delivery

V8.3

インターネット経由でファイルをダウンロードすることは、手軽にソフトウェアのアップデートをしたいと思う OpenVMS ユーザにとっては、必須条件になりつつあります。一方で、これらのユーザは、セキュリティの脆弱性に用心深くなっています。Secure Delivery は、公開鍵と電子署名技術を用いており、OpenVMS ユーザは HP および他社の OpenVMS ベンダーからダウンロードするファイルが本物であることを確認することができます。

Secure Delivery は、OpenVMS Version 8.3 では完全に機能します。詳細は、『HP OpenVMS V8.3 新機能説明書』を参照してください。

5.11.2 インストールと初期化に関する考慮

V7.3-2

CDSA は、オペレーティング・システムのインストール時に自動的にインストールされます。

- 新しいバージョンの CDSA を、オペレーティング・システムのアップグレードとは別にインストールする場合は、CDSA のアップグレード・プロシージャを実行する必要があります。

```
$ @SYS$STARTUP:CDSA$UPGRADE
```

アップグレード・プロシージャを実行する前に、すべての CDSA アプリケーションをシャットダウンしてください。

システムをリブートするときに、アップグレード・プロシージャを再実行する必要はありません。また、OpenVMS スタートアップ・プロシージャにアップグレード・プロシージャを追加する必要もありません。

- 使用中のシステムから CDSA を削除しないでください。CDSA を削除するように見えるオプションがありますが、CDSA に対する PCSI コマンドの PRODUCT REMOVE はサポートされていません。(このオプションはインストールに使用する PCSI ユーティリティの機能です)。CDSA とオペレーティング・システムは、同時にインストールされ、密接に結合しているので、OpenVMS から CDSA を削除しようとする、正常に動作しないで、望ましくない影響が発生する可能性があります。CDSA を削除しようとする、次のようなメッセージが表示されます。

```
%PCSI-E-HRDREF, product CPQ AXPVMS CDSA V2.2 is referenced by DEC AXPVMS OPENVMS V8.3  
-PCSI-E-HRDRF1, the two products are tightly bound by this software dependency
```

5.12 デバッグ・モード: CPUSPINWAIT バグ・チェックの回避

永続的な条件

OpenVMS オペレーティング・システムには、複雑なハードウェアの問題やソフトウェアの問題をデバッグするのに役立つように、多くの特殊操作モードが準備されています。一般には、これらの特殊モードを使用すれば、特別なレベルでトレース、データの記録、一貫性チェックを行うことができ、このような機能は、問題があるハードウェア構成要素やソフトウェア構成要素を突き止めるのに役立ちます。これらの操作モードは、システム・パラメータ MULTIPROCESSING, POOLCHECK, BUGCHECKFATAL, SYSTEM_CHECK によって制御されます。

一般に I/O 負荷の高い特定の状況で、これらの特殊モードのいずれかを使用している場合は (たとえば、デバイス・ドライバや他の複雑なアプリケーションをデバッグする場合など)、CPUSPINWAIT バグ・チェックが発生することがあります。特に、スピンロックのある状態で長期間実行する特権コードに対して CPUSPINWAIT バグ・チェックが発生します。スピンロックは、クリティカル・セクションのエントリ・ポイントとイグジット・ポイントを区切るために使われ、この場合のように連続的に使うことはできません。

CPUSPINWAIT バグ・チェックを防止するには、これらのシステム・パラメータに対して、システムのデフォルト設定を使用するか、またはシステムの負荷を低下させます。

何らかの理由でデフォルトの設定を変更しなければならない場合は、SMP_LNGSPINWAIT システム・パラメータを 9000000 に設定することで、問題が発生する可能性を減らせます。

5.13 Delta/XDelta デバッグ

ここでは、OpenVMS Alpha および I64 システム上で動作する OpenVMS Delta および XDelta デバッグに関する注意事項について説明します。

OpenVMS Debugger に関する注意事項は第 5.29 節を参照してください。

5.13.1 XDelta のレジスタ表示に関する考慮 (I64 のみ)

V8.2

OpenVMS I64 上の XDelta は、あたかも、レジスタ・リネーム・ベース (CFM.rrb) とローテート・サイズ (CFM.sor) がいずれも 0 であるかのように、汎用レジスタ、浮動小数点レジスタ、およびプレディケート・レジスタを表示します。言い換えると、ローテートするレジスタを使用しているときには、ローテーションの効果は無視され

ます。この状況は、今後のリリースで修正される予定です。詳細は第 5.10 節を参照してください。

5.14 ファイル・アプリケーション: 『Guide to OpenVMS File Applications』の訂正

永続的な訂正

マニュアル『Guide to OpenVMS File Applications』が改訂されるときには、下記の訂正が盛り込まれます。

- 表 1-4 は誤解を招く可能性があります。ファイル・フォーマットの比較表で、ODS-2 や ODS-5 のファイル・フォーマットでのディレクトリの制限として 256 としてあります。この説明は、256 ディレクトリ・レベルとし、ユーザが、ディレクトリ数が 256 に制限されているものと誤解しないようにする必要があります。

『OpenVMS システム管理者マニュアル』の類似した表は Version 8.2 で訂正されました。

- 第 6.6.3 項の第 4 パラグラフ (Note はカウントしない) を、以下の内容に置き換えます。

「プログラム内で使用するルート・デバイス論理名を、SET DEFAULT コマンドで定義する際には、DCL コマンドの DEFINE または ASSIGN で /TRANSLATION_ATTRIBUTES=CONCEALED 修飾子を使用して、この論理名が隠し装置の論理名であることを指定します。隠し装置の論理名をルート・デバイス論理名として定義するためには、ルート・ディレクトリがピリオド (.) で終わっていなければなりません (例: DUA22:[ROOT.])。また、デバイスの指定が、物理デバイス名でなければなりません。ルート・デバイス論理名の等価名には、他の論理名を含めることはできません。ディレクトリを指定するときには、ルート・ディレクトリに対して、後のピリオドだけを使用することができます。」

5.15 RMS 構造体についての HP BLISS コンパイラの警告 (I64 のみ)

永続的な条件

RMS ユーザ構造体 (たとえば、FAB、RAB) の割り当てに使用できる BLISS マクロ (\$xxx_DECL) に、キーワード・アラインメントが追加されました。プロセッサが高速になるほど、アラインメント・フォルトは性能に悪影響を及ぼします。アラインメントをマクロ内に直接実装することにより、これらのマクロを使用する、BLISS で書かれた多数の OpenVMS ユーティリティおよびユーザ・アプリケーションは、性能が改善されます。

該当するマクロは、\$FAB_DECL、\$NAM_DECL、\$NAML_DECL、\$RAB_DECL、\$RAB64_DECL、\$XABALL_DECL、\$XABDAT_DECL、\$XABFHC_DECL、\$XABITM_DECL、\$XABJNL_DECL、\$XABKEY_DECL、\$XABPRO_DECL、\$XABRDT_DECL、\$XABRU_DECL、\$XABTRM_DECL、および\$XABSUM_DECLです。

RMS マクロに追加されたアラインメントにより、コンパイラがアラインメント競合の警告を出力することがあります。コンパイラの警告があるプログラムでも、正しくリンクして、実行することができます。ただし、ソースに簡単な変更を加えて、警告を取り除くことをお勧めします。

これらのマクロを BLISS アプリケーション内で使用し、宣言に ALIGN 属性が含まれている場合、BLISS コンパイラは“conflicting or multiply specified attribute”という警告を出力します。たとえば、FAB: \$FAB_DECL ALIGN(2) という宣言に対して、警告が出力されます。この警告は、クォードワード・アラインメント (ALIGN(3)) を指定したとしても出力されます。これらのマクロに関連する明示的な ALIGN 属性を削除する必要があります。

さらに、これらの割り当てが、ALIGN(3) と競合する明示的なアラインメント (ALIGN(3) 未満のもの) を持つ PSECT に含まれている場合、BLISS コンパイラは、“align request negative or exceeds that of psect”という警告を出力します。たとえば、次の宣言に対して警告が出力されます。

```
PSECT OWN = $OWN$ (... , ALIGN(2), ...)
OWN
    FAB = $FAB_DECL, ...
```

BLISS アプリケーションの再コンパイル時に PSECT のアラインメントに関する警告が表示された場合、PSECT のアラインメントを ALIGN(3) (またはそれ以上) に調整してください。まれに、PSECT 間でデータが隣接していると仮定しているアプリケーションが存在することがあります。この変更により、このような仮定が成り立たなくなることがあります。そのため、コードでこのような仮定を行っていないかチェックし、必要な変更を行ってください。

多くの OpenVMS ユーティリティが BLISS で記述されていますが、OpenVMS 全体のビルドで発生した警告はわずかでした。この警告を取り除くために OpenVMS に加えた変更は他にはありませんでした。このため、修正が必要なユーザ・アプリケーションは、ほとんどないと考えられます。

5.16 RMS の Must-Be-Zero エラーの可能性: FAB 内に新しいファイ ル・オプション用の場所を確保

V8.3

5.16 RMS の Must-Be-Zero エラーの可能性: FAB 内に新しいファイル・オプション用の場所を確保

RMS のユーザ・ファイル・アクセス・ブロック (FAB) には、未割り当ての領域がごくわずかしかありません。FAB を通じて実装されるファイル処理オプションを将来拡張できるように、FAB (FAB\$L_FOP) 内のファイル処理オプション・フィールドの最後の未割り当てビットが、FAB\$V_EXTEND_FOP オプションとして定義されました。このオプションは、FAB 内で以前使用されていなかった領域に置かれる以下の 2 つの新しい FAB フィールドへの割り当てを要求し検査するために、将来使用されます。

- FAB\$W_FOPEXT: ワード・フィールド FOPEXT は、将来実装が必要になる新しいファイル処理オプションの定義用として予約されています。
- FAB\$W_RESERVED_MBZ: ワード・フィールド RESERVED_MBZ は、将来の使用のために予約されています。その用途は現在決定されておらず、将来定義されます。

将来のリリースで新しいファイル処理オプションを実装するときには、FOPEXT フィールドの新しいオプションを利用するために、FAB\$L_FOP フィールドの FAB\$V_EXTEND_FOP ビットをオンにする必要があります。ただし、このビットをオンにすると、FOPEXT フィールドの未定義のビットがオフで、FAB\$W_RESERVED_MBZ にゼロが格納されていることも示します。この条件が満たされていない状態でこのビットをオンにすると、RMS のすべてのファイル・レベル・サービスで、回復不可能なエラー (RMS-F-FOPEXTMBZ) が返されます。

EXTEND_FOP オプションの追加は完全に上位互換性があります。ただし、ユーザが FAB\$L_FOP フィールドのこの最後のビットを誤ってオンにし、この 2 つの新しい FAB フィールドのいずれかがゼロでない場合は例外です。以前は、FAB\$L_FOP フィールドのこの最後のビットが誤ってオンになっていた場合でも、RMS はそれを無視していました。

RMS-F-FOPEXTMBZ エラーが返される場合は、FAB\$L_FOP フィールドの EXTEND_FOP オプションをオフにするか、2 つの新しいフィールド FAB\$W_FOPEXT および FAB\$W_RESERVED_MBZ をクリアする必要があります。

5.17 HP COBOL ランタイム・ライブラリ (RTL)

V8.3

OpenVMS Alpha および OpenVMS I64 Version 8.3 では、HP COBOL RTL (DEC\$COBRTL) は V2.8-781 にアップデートされました。

5.18 I64 用の HP Fortran

V8.3

OpenVMS Integrity サーバ用の HP Fortran コンパイラは、OpenVMS Alpha の HP Fortran 90 を移植したものです。このコンパイラは OpenVMS I64 システム上で動作し、OpenVMS I64 システム用のオブジェクトを生成します。このオブジェクトは、OpenVMS I64 上の標準リンカを使用してリンクされます。このコンパイラは、OpenVMS I64 Version 8.2 以降を必要とします。

OpenVMS Integrity サーバ用の HP Fortran のコマンド行オプションと言語機能は、以下の例外を除いて、OpenVMS Alpha システム用の HP Fortran 90 のものと同じです。

- 浮動小数点の計算

要点は、ホワイト・ペーパー『Intel® Itanium®アーキテクチャにおける OpenVMS 浮動小数点演算について』を参照してください。このドキュメントは、次の Web サイトで参照できます。

<http://h71000.www7.hp.com/openvms/integrity/resources.html>

- デフォルトの浮動小数点データ型は IEEE です。(つまり、/FLOAT=IEEE_FLOAT がデフォルトです)。
- /IEEE_MODE 修飾子のデフォルト値は、/IEEE_MODE=DENORM_RESULTS です。
- ユーザは/FLOAT の値を 1 つと、/IEEE_MODE の値を 1 つ選択し、アプリケーション全体でその値を維持しなければなりません。
- F90 コンパイラだけがサポートされています。以前/OLD_F77 修飾子で起動されていた F77 コンパイラは利用できません。FDML や CDD のサポートなど、Alpha F77 コンパイラに含まれていて、Alpha F90 コンパイラでは利用できない一部の機能が、I64 F90 コンパイラに実装されています。詳細は、Fortran V8.0 または V8.1 製品のリリース・ノートを参照してください。
- /ARCH 修飾子と/TUNE 修飾子での値 Alpha は、コンパイル・アンド・ゴーの互換性を保つために、コンパイラ起動コマンドで利用することができます。これらの値を無視したことを示す情報メッセージが表示されます。

インストール手順など、このリリースについての詳細は、Fortran V8.0 または V8.1 製品のリリース・ノートを参照してください。リリース・ノートを抽出するには、Fortran PCSI キットが置かれているディレクトリをデフォルトとして設定し、次のコマンドのいずれかを入力します。

```
$ PRODUCT EXTRACT RELEASE NOTES FORTRAN ! For TXT file  
$ PRODUCT EXTRACT FILE FORTRAN/SELECT=FORTRAN_RELEASE_NOTES.PS ! For PS file
```


5.19 OpenVMS 用 HP MACRO

OpenVMS MACRO コンパイラは、OpenVMS VAX システム用に記述された Macro-32 ソース・コード (VAX MACROアセンブラ) をコンパイルし、OpenVMS Alpha および OpenVMS I64 システムで動作する機械語コードに変換します。ここでは、MACRO コンパイラに関する注意事項について説明します。

5.19.1 OpenVMS I64 用 HP MACRO

V8.3

HP MACRO for OpenVMS I64 コンパイラには、以下の注意事項があります。

- OpenVMS Version 8.3 よりも前のバージョンでは、コンパイラは HALT 命令に対して誤ったコードを生成していました。I64 プラットフォームでは、HALT 命令は Itanium の break 命令と、予約リテラル値 BREAK\$C_SYS_HALT を使用して実装されます。コンパイラの構築環境のバグにより、Macro-32 コンパイラが誤ったリテラル値を使用していました。この問題はバージョン 8.3 で修正されました。HALT 命令を使用しているすべてのコードを、バージョン 8.3 のコンパイラで再コンパイルする必要があります。バージョン 8.3 よりも前のシステムでは、次のようにして正しい動作を実現することができます。

```
$BREAKDEF  
IA64_HALT #BREAK$C_SYS_HALT ; Issue break instruction with correct literal  
HALT ; Use HALT builtin to inform compiler that this ends the flow of control
```

- コンパイラは、HALT、BPT、および EVAX_BUGCHK の前の命令を最適化によって除去する可能性があります。オブティマイザは、これらの命令が、ダンプ・ファイルを書き出したり、デバッグ環境に制御を渡すことで、すべてのレジスタを暗黙に読み込む特殊な命令になっていることを認識しません。使用されていないように見える命令は、誤って削除されます。これらの命令の特殊な動作についてコンパイラに通知することができます。任意の命令を最終的なオブジェクト・ファイルに残すように指示する構文はありませんが、IA64_LD8_A ビルトインを回避策として使用できます。このビルトインについては、次の Macro-32 の段落を参照してください。また、/NOOPTIMIZE を指定すると、使用されていないように見える命令も残りますが、コードの実行速度が遅くなり、またコードのサイズが大きくなります。
- コンパイラは、使用されていないように見えるメモリのロードを最適化によって除去することがあります。コンパイラのオブティマイザは、メモリのロード結果が使用されるかどうかを認識することができます。結果が使用されていないと判断されると、オブティマイザはメモリ・ロードを削除します。しかし、コードによっては、たとえば IPL を上げる前にページをメモリにページ・インさせるためにメモリのロードを使用している場合があります。そのような場合は、命令が除去されるとページがメモリにページ・インされなくなり、高い IPL での以降のコードで、高い IPL 例外での回復不可能なページ・フォルトが発生します。任意の

命令を最終的なオブジェクト・ファイルに残すように指示する構文はありませんが、IA64_LD8_A ビルトインを回避策として使用できます。バージョン 8.3 で新たに追加された IA64_LD8_A ビルトインは、特別な形式の Itanium "ld8"命令を生成し、フェッチされたアドレスを ALAT (Advanced Load Address Table) に格納します。コンパイラはこの特別な形式の "ld8"を、副作用を持ち、結果が使用されていないように見える場合でも最終的なオブジェクト・ファイルから削除できないものとして認識します。ALAT にアドレスを挿入しても、問題が発生したり、他の変更が必要になることはありません。最終的なオブジェクト・ファイルに残す必要がある、使用されないメモリのロードについては、次のように変更します。

```
MOVL (Rn), Rn
```

から

```
IA64_LD8_A Rn, (Rn), #0
```

将来のリリースでは、最終的なオブジェクト・ファイルに残す必要がある命令を指定するための新しい構文がコンパイラに追加される予定です。

バージョン 8.3 よりも前のシステムでは、IA64_LD8_A ビルトインはありません。唯一の回避策は、/NOOPTIMIZEを使用することです。

5.19.2 OpenVMS Alpha システム用の HP MACRO

V8.3

コンパイラは、使用されていないように見えるメモリのロードを最適化によって除去することがあります。コンパイラの新しいオプティマイザは、メモリのロード結果が使用されるかどうかを認識することができます。結果が使用されていないと判断されると、オプティマイザはメモリ・ロードを削除します。しかし、コードによっては、たとえば IPL を上げる前にページをメモリにページ・インさせるためにメモリのロードを使用している場合があります。そのような場合は、命令が除去されるとページがメモリにページ・インされなくなり、高い IPL での以降のコードで、高い IPL 例外での回復不可能なページ・フォルトが発生します。唯一の回避策は、/NOOPTIMIZEを使用するか、最適化機能を備えていない以前の Macro-32 コンパイラに戻ることです。

新しい Macro-32 コンパイラの名前はSYS\$SYSTEM:MACRO.EXEであり、DCL MACRO コマンドではこのイメージがデフォルトで起動されます。古いコンパイラはSYS\$SYSTEM:ALPHA_MACRO.EXEにあります。DCL コマンド MACRO で古いコンパイラを使用するには、論理名 MACRO を次のように定義します。

```
$ DEFINE MACRO SYS$SYSTEM:ALPHA_MACRO.EXE
```

5.19.3 /TIE 修飾子のデフォルトは Alpha と I64 で異なる

V8.2

/TIE 修飾子のデフォルトは Alpha システムと I64 システムで異なります。Alpha では、デフォルトは/TIE です。I64 では、デフォルトは/NOTIE です。

5.19.4 /OPTIMIZE=VAXREGS 修飾子は I64 ではサポートされない

V8.2

Alpha システムでサポートされていた/OPTIMIZE=VAXREGS 修飾子は、I64 システムではサポートされません。残念ながら、関連するコードすべてがコマンド行処理から削除されてはいません。I64 システムで/OPTIMIZE=ALL を指定すると、サポートされていない VAXREGS 最適化を誤って起動することになります。今後のリリースで、コマンド行プロセスを修正して VAXREGS 最適化を起動しないようにする予定です。

5.19.5 浮動小数点数のゼロ除算エラーが検出されない (I64 のみ)

V8.2

Macro-32 浮動小数点数サポート・ルーチンは、浮動小数点数のゼロ除算を検出しません。サポート・ルーチンでは、VAX 浮動小数点数を IEEE 浮動小数点数に変換して除算を実行します。チェック処理無しで、除算で IEEE の NaN 値 (非数値) が生成されます。サポート・ルーチンは、次に NaN 値を VAX 浮動小数点数に戻そうとします。この操作で、不正浮動小数点数のエラーになります。今後のリリースで、サポート・ルーチンが修正され、正しく浮動小数点数のゼロ除算エラーを検出ようになります。

5.20 Hypersort ユーティリティ

ここでは、OpenVMS Alpha および OpenVMS I64 Version 8.2 用の Hypersort V08-006 に関する注意事項について説明します。

Hypersort で修正されていない問題を回避する場合、または Hypersort に実装されていない機能を使用する場合には、従来どおり SORT32 を使用してください。SORT32 に関する注意事項は第 5.35 節を参照してください。

5.20.1 弊社への問題の報告

永続的な条件

SORT や MERGE で問題を発見した場合は、問題を報告する前に、次のコマンドを実行してください。

```
$ WRITE SYS$OUTPUT "WSEXTENT ="'F$GETJPI("","WSEXTENT")' "  
$ WRITE SYS$OUTPUT "PGFLQUOTA=" 'F$GETJPI("","PGFLQUOTA")' "  
$ SHOW LOGICAL SORTSHR  
$ SORT/STATISTICS (or MERGE/STATISTICS)
```

問題を再現する入力ファイルとともに、この出力を問題の報告に含めてください。

5.20.2 ラージ・ファイルの制限事項

V8.2

Hypersort V08-010 は、メモリ割り当てのアルゴリズムが改良されましたが、大規模な入力ファイルが使用されると、ハングアップしたり、ACCVIO が発生することがあります。ハングアップや ACCVIO が発生する可能性を低くするには、第 5.20.8 項に記載されているとおりにページ・ファイル・クォータとワーキング・セット・エクステンツを設定します。Hypersort でハングアップしたり ACCVIO が発生する場合は、代わりに SORT32 を使用してください。

5.20.3 Hypersort と VFC ファイルの制限事項

V7.3-2

Hypersort で VFC ファイルを使用するには、/FORMAT=RECORD_SIZE:n が必要です。

5.20.4 /FORMAT=RECORD_SIZE の制限事項

永続的な制限事項

Hypersort では、SORT と MERGE の両方で使用する /FORMAT=RECORD_SIZE:n がサポートされます。ただし、次の 2 つの制限事項があります。

- すべての場合において、コマンドで指定した RECORD_SIZE の値が入力ファイル内の任意のレコードの最大レコード長 (LRL) よりも小さい場合、長すぎるレコードは、ソートされた出力ファイルで RECORD_SIZE のサイズまで切り捨てられ、診断メッセージ %SORT-E-BAD_LRL が発行されます。この場合は、出力ファイルを破棄し、ソートを再実行する必要があります。SORT コマンドの RECORD_SIZE パラメータの値を、DIR/FULL コマンドを実行して表示される入力ファイルの最大レコードのサイズに合わせて修正してください。

- SORT や MERGE によって、入力索引順編成ファイルから出力順編成ファイルが作成されます。この場合、%SORT-E-BAD_LRL 診断メッセージも発行される場合があります。

5.20.5 Hypersort と検索リスト、および論理名の使用

永続的な制限事項

Hypersort では、検索リスト、および入力ファイルと作業ファイルで使用される論理名のサポートが十分ではありません。この問題を検出した場合は、SORT32 を使用してください。

5.20.6 作業ファイルの空き領域不足

永続的な制限事項

すべてのソート作業ファイルで空き領域が無くなると、Hypersort が正しく終了しません。この問題を防ぐには、次のいずれかの処理を実行してください。

- ソートの作業ファイル用に使用するデバイスに、十分な空き領域を割り当てる。
- SORT32 を使用して、作業ファイルの領域を使い果たしたことを検出する。

5.20.7 入力アスタリスク (*) の制限事項

永続的な制限事項

Hypersort では、入力ファイル指定にアスタリスク (*) を使用できません。

5.20.8 最適化されたワーキング・セット・エクステントとページ・ファイル・クォータの設定

永続的な制限事項

SORT32 と Hypersort は、異なるソート・アルゴリズムと作業ファイル・アルゴリズムを使用します。これらのユーティリティの相対的な速度は、入力ファイルと、メモリ、ディスク、および CPU の構成によって異なります。ワーキング・セット・エクステントが、ページ・ファイル・クォータの 3 分の 1 を超えないようにしてください。SORT32 と Hypersort はいずれも、個々の入力ファイルに合ったワーキング・セット・エクステントを使用することで、最高の性能を発揮します。

5.21 Intel®アセンブラ (I64 のみ)

永続的な制限事項

Intel アセンブラ言語で記述されたすべてのモジュールには、`-Xunwind` フラグによる自動生成を行うか、ソース・モジュールに明示的に `unwind` ディレクティブを記述する方法で、適切な `unwind` ディレクティブが含まれていなければなりません。正確な `unwind` 情報なしでは、オペレーティング・システムの条件処理と例外ディスパッチが動作せず、予期しない状態でプログラムが失敗することがあります。正確な `unwind` 情報を持たないプログラムは、OpenVMS ではサポートされていません。この前提条件は、永続的な要件となります。

5.22 Librarian ユーティリティ

ここでは、Librarian ユーティリティと Library Service ルーチンに関する注意事項について説明します。

5.22.1 data-reduced ELF オブジェクト・ライブラリとのリンクは推奨できない (I64 のみ)

V8.2

DCX data-reduced ELF オブジェクト・ライブラリは、ライブラリ内の連続領域には格納されません。その結果、最初にデータの展開を行う必要があるため、モジュールをプロセスの P2 空間に直接マッピングすることはできません。LBR\$MAP_MODULE ライブラリ・サービスが、モジュールを展開しプロセスの P2 空間にコピーします。この動作により、結果としてできた P2 空間内のページが、プロセス・クォータとしてカウントされます。

LBR\$UNMAP_MODULE ライブラリ・サービスは、これらのページを回復しますが、これらのページはヒープ領域解放リストに残り、プロセス・クォータとしてカウントされ続けます。そのため、Linker 操作の前に、あらかじめ DCX data-reduced ELF オブジェクト・ライブラリを展開することをお勧めします。

5.22.2 I64 ライブラリへの.STB ファイルの挿入または置き換えの失敗 (I64 のみ)

V8.2

OpenVMS VAX および Alpha では、.STB ファイルをオブジェクト・ライブラリに格納することができます。OpenVMS I64 では、Librarian はこの動作を行いません。例を次に示します。

```
$ LIBR/CREATE OBJ$:SOME_LIBRARY OBJ$:SOME_FILE.STB
Librarian T01-23
%LIBRAR-E-NOTELFFILE, TPSSWRKD$: [TPSS.TRACE.IA64.LPIOBJ]TRACMSG.STB;1
is not an ELF object or image file
```

.STB ファイルはオブジェクトでもイメージでもないため、現在 Librarian は、このファイルを I64 システムの.OLB ファイルには格納しません。

ただし、Alpha と VAX では、.STB ファイルはオブジェクト・ファイルと同様の形式になっています。VAX では、.STB ファイルを Linker への入力として使用することもできます。Alpha では、.STB ファイルの形式は、.OBJ ファイルの形式と同じです（つまり、ファイル拡張子が.STB であること以外は、これらのファイルに違いはありません）。そのため、Alpha の Librarian はこのファイルを受け付けますが、Linker の入力として使用することはできません。

I64 では、.STB ファイルにファイル・タイプ (ET_VMS_LINK_STB) が埋め込まれます。これにより、Librarian と Linker が、.STB ファイルを処理できないことを判断できます。これは永続的な状態です。

5.22.3 プロセス・クォータが低すぎると Librarian がエラーを通知しない問題

永続的な制限事項

OpenVMS Alpha および I64 の Librarian は圧縮、データ・リダクション、データ拡張操作でエラーを通知しないことがあります。この問題が発生するのは、Librarian が動作しているアカウントまたはプロセスの PGFLQUOTA プロセス・クォータが低い場合です。\$PUTMSG システム・サービスは、エラーが発生した場合でも必ず SSS_NORMAL というステータスを返すので、操作エラーがただちに明らかになりません。しかし、エラーが発生した場合には、Librarian は Success 以外のステータスを返します。

この問題を回避するには、PGFLQUOTA を増加させてから圧縮、データ・リダクション、データ拡張操作を実行します。さらに、コマンド・プロシージャで LIBRARY コマンドからの戻りステータスを確認するようにしてください。

5.23 OpenVMS 用の Analyze ユーティリティ (I64 のみ)

ここでは、OpenVMS I64 上の Analyze ユーティリティに関する注意事項について説明します。

5.23.1 I64 Analyze イメージの/SELECT でのプロセス・リソースの解放についての修正

V8.3

I64 イメージ・ファイルに対する ANALYZE/SELECT でプロセス・リソースが解放されませんでした。ワイルド・カードを使用した場合は、SECTBLFUL (または他のクォータ超過のメッセージ) で失敗することがありました。

この問題は、バージョン 8.3 で修正されました。

5.23.2 デバッグ行情報の選択的出力の修正

V8.3

I64 イメージ・ファイルでは、ANALYZE/IMAGE/SELECT=DEBUG=LINE で対応するデバッグ行情報が出力されませんでした。行の選択は、/TRACE の場合だけ動作していました。

この問題は、バージョン 8.3 で修正されました。

5.24 Command Definition ユーティリティ (I64 のみ)

ここでは、OpenVMS I64 上の Command Definition ユーティリティに関する注意事項について説明します。

5.24.1 I64 イメージのレコード属性の修正

V8.3

I64 イメージ (コマンド・テーブル) では、Command Definition ユーティリティ (CDU) はレコード属性を設定しますが、それにより、イメージ・ファイルのコピーまたはアペンドの際に警告メッセージが表示されることがありました。警告メッセージが表示されても、コピーまたはアペンドされたファイルは正しい内容となっています。

バージョン 8.3 では、警告メッセージが表示されことなくこれらのファイル操作が完了するように、CDU は互換性のある属性を設定します。

5.25 OpenVMS Alpha 用 Linker ユーティリティ

ここでは、すべての OpenVMS プラットフォーム上での Alpha Linker ユーティリティに関する注意事項について説明します。I64 Linker に関する注意事項は第 5.26 節を参照してください。

5.25.1 多数のファイルを指定した場合に Linker がハングアップしたように見える

V8.3

RMS_RELATED_CONTEXT リンカ・オプションがオン (デフォルトは RMS_RELATED_CONTEXT=YES) で、存在しないファイルが LINK コマンドのファイル・リストに指定されていた場合、リンカによる LIB\$FIND_FILE の呼び出しは完了するまでに長時間かかり、リンカがハングアップしたように見えることがあります。リンクしているファイルの数と、ファイル指定での論理名の使用状況に応じて、リンカの処理が完了するまでに数時間かかることもあります。これは LIB\$FIND_FILE が、不明ファイルについてプレフィックスの組み合わせをすべて探してから、"file not found"メッセージを表示するためです。リンカが LIB\$FIND_FILE を呼び出した後は、Ctrl/Y を押してもリンカを終了させることはできません。

どのファイルが不明かを調べるには、『*Linker Manual*』の第 4 部「LINK Command Reference」の RMS_RELATED_CONTEXT=オプションの説明に記載されている手順を使用します。

5.25.2 ライブラリ・チェックにおける Linker のデフォルト動作の変更

V7.3-1

これまでの Linker では、ライブラリと共有イメージ間の一致条件が厳密に検証されていましたが (正確な日時を照合し、該当するものがない場合は、LINK-I-DATMISMCH シグナル通知を発行)、このリリースでは、イメージ・アクティベータと同じ検証 (GSMATCH 条件を使用して互換性を検証) だけが実行されます。

以前の動作 (日時の照合) を実行する場合は、LINK\$SHR_DATE_CHECK 論理名を設定してください。

詳細は、『*Linker Manual*』の第 4 部「LINK Command Reference」の/LIBRARY 修飾子を参照してください

共用可能イメージ・ライブラリには、イメージのコピーは格納されていません。格納されているのは、イメージの名前、イメージの識別情報、イメージのユニバーサル・シンボルが格納されたテーブルです。識別情報は、共用可能イメージをリンクする際に GSMATCH=オプションで指定します。詳細は、『*Linker Manual*』の第 4 部「LINK Command Reference」の GSMATCH=オプションを参照してください。

共用可能イメージを再リンクしても、ライブラリが更新されない場合があります。このような場合に対処するために、リンカは互換性を確認します。リンカは、イメージ・アクティベータが行うのと同じように、GSMATCH条件を使用して互換性を確認します。

VAX では、リンカは日付と時刻も比較し、違っている場合にはLINK-I-DATMISMCHを生成します。

Alpha では、リンカの最初の動作は VAX のリンカと同じでした。しかし、このチェックは厳し過ぎると判断され、デフォルトでは GSMATCH 条件だけを使用するように変更されました。論理名LINK\$SHR_DATE_CHECKを定義すると、VAX と同じ以前の動作になります。

5.25.3 スタックの要素数は最大 25 に制限

永続的な制限事項

オブジェクト・ファイルを作成する開発者は、Linker の内部スタックの要素数が最大 25 に制限されていることに注意しなければなりません。どのような計算も、この制限の範囲内で実行しなければなりません。

5.26 OpenVMS I64 用 Linker ユーティリティ

V8.3

ここでは、I64 Linker ユーティリティに関する注意事項について説明します。

Alpha Linker に関する注意事項は、第 5.25 節を参照してください。

5.26.1 正しくないイメージ間デバッグ・フィックスアップをリンカがデバッグ・シンボル・ファイルに書き込む

V8.3

状況によっては、リンカは OpenVMS デバッガ用のイメージ間フィックスアップを作成します。イメージ間デバッグ・フィックスアップは、リンカが解決できない、コンパイラが生成したデバッグ再配置の結果です。つまり、共用可能イメージに格納されている値を実行時に調べるために、デバッガはこれらのフィックスアップを必要とします。デバッガ用にイメージ間フィックスアップをリンカに作成させることが必要となる頻度は、コンパイラによって異なります。C コンパイラはイメージ間デバッグ・フィックスアップをめったに使用しませんが、C++ コンパイラは頻繁に使用します。このようなイメージを/DEBUG 付きでリンクすると、リンカはデバッグ情報の後にイメージ間デバッグ・フィックスアップを書き込みます。/NODSF (省略時の指定) を使

用すると、情報はイメージ・ファイルに正しく書き込まれますが、/DSF を指定すると、情報は誤って DSF ファイルに書き込まれることがあります。

たとえば、次の例の DEBUG 情報と DEBUG エラーは、リンカが誤って DSF ファイルへ書き込んだために表示されます。

```
$ RUN/DEBUG MINIREF
%DEBUG-I-INFODWARF, error reading Dwarf info: Section 0a extends outside file
%DEBUG-I-INFODWARF, error reading Dwarf info: Section 0c extends outside file
%DEBUG-I-INFODWARF, error reading Dwarf info: SHT_VMS_FIXUP section 10 size 17eb
e0 not multiple of 18
%DEBUG-I-INFODWARF, error reading Dwarf info: SHT_VMS_FIXUP section 12 size 17ec
30 not multiple of 18

OpenVMS I64 Debug64 Version V8.3-003

%DEBUG-F-ACCVIO, access violation, reason mask=00, virtual address=000000000014A
000, PC=000000007BD73100, PS=0000001B
%DEBUG-I-INITIAL, Language: C, Module: MINIREF

DBG>
```

イメージファイルには影響がなく、RUN コマンドを使用して問題なく実行できます。

```
$ RUN MINIREF
```

このエラーは I64 リンカの次のリリースで修正される予定です。

5.26.2 OpenVMS I64 のオブジェクト・モジュールとイメージ・ファイルの情報が現在利用できない

V8.3

ELF (Executable and Linkable Format。I64 でのオブジェクト・モジュールとイメージ・ファイルの形式) に対する OpenVMS I64 の拡張についての情報は、現時点では公開しておりません。将来のリリースで提供されます。

Alpha または VAX のオブジェクト言語形式についての詳細は、『OpenVMS Alpha /VAX Version 7.3 OpenVMS Linker Utility Manual』のそれぞれに対応した付録を参照してください。このこのマニュアルは、次の URL で入手できます。

http://h71000.www7.hp.com/doc/os732_index.html

5.26.3 /SELECTIVE_SEARCH がトランスファー・アドレスを誤って無視することがある

V8.3

トランスファー・アドレスが含まれている I64 オブジェクト・モジュールがあり、
/SELECTIVE_SEARCH 修飾子を指定したリンク操作にそのモジュールを含めると、
リンカはそのトランスファー・アドレスを検出しません。

次の例では、オブジェクト・モジュール (MAIN.OBJ) にトランスファー・アドレスが
含まれていますが、/SELECTIVE_SEARCH によって無視されます。

```
$ LINK MAIN/SELECTIVE_SEARCH  
%LINK-W-USRTFR, image USER:[JOE]MAIN.EXE;1 has no user transfer address
```

この状態になるのは、プログラムのトランスファー・アドレスを提供することを意図
した I64 オブジェクト・モジュールが、SELECTIVE_SEARCH 属性を使用したリ
ンク操作に含まれている場合だけです。次の例のように、LINK コマンドでオブジ
ェクト・モジュールに/SELECTIVE_SEARCH 修飾子を指定すると、SELECTIVE_
SEARCH 属性がオブジェクト・モジュールに与えられます。

```
$ LINK MAIN/SELECTIVE_SEARCH
```

次のような LIBRARY と LINK コマンドの例も考えられます。

```
$ LIBRARY/INSERT LIB.OLB MAIN.OBJ/SELECTIVE_SEARCH
```

このライブラリに含まれているモジュールを、リンク操作で参照を解決するために使
用します。暗黙的に使用する例は次のとおりです。

```
$ LINK/EXECUTABLE=MAIN SUBROUTINES.OBJ, LIB.OLB/LIBRARY
```

明示的に使用する例は次のとおりです。

```
$ LINK/EXECUTABLE=MAIN SUBROUTINES.OBJ, LIB.OLB/INCLUDE=MAIN
```

この問題は、以下の 2 つのうちのどちらかの形で明らかになります。

- 上記の例のように、リンカが警告メッセージを表示する場合。この状態になるの
は、リンク操作の他のオブジェクト・モジュールが weak かどうかにかかわらず
トランスファー・アドレスを提供しない場合です。
- リンカがメッセージを表示しない場合。この状態になるのは、リンク操作の他の
オブジェクト・モジュールが weak かどうかにかかわらずトランスファー・アド
レスを提供する場合です。選択的に検索されたオブジェクト・モジュールからト
ランスファー・アドレスを見つけることができないと、リンカは他のオブジェク
ト・モジュールのトランスファー・アドレスを選択します。そのトランスファ
ー・アドレスは、意図せずにそのイメージのメイン・エントリ・ポイントとなり
ます。マップ・ファイルには、リンカが正しくない遷移モジュールとトランスフ
ァー・アドレスを割り当てたことが出力されますが、実際にアプリケーションを
実行するまで問題に気づかない可能性があります。

5.26.4 I64 リンカと Alpha リンカの違い

V8.3

OpenVMS I64 Linker と OpenVMS Alpha Linker の違いに関する詳細な説明については、『HP OpenVMS V8.3 新機能説明書』を参照してください。『HP OpenVMS V8.3 新機能説明書』には、OpenVMS I64 Linker の新機能についての説明もあります。

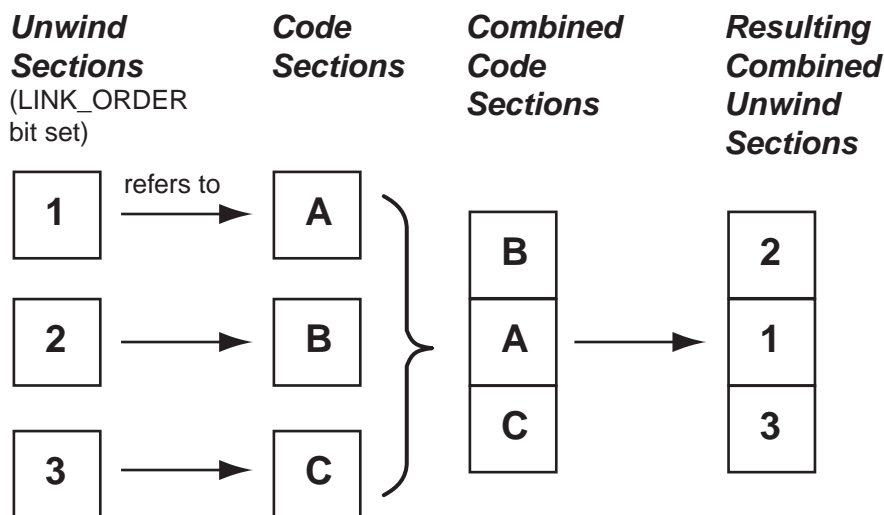
5.26.5 LINK_ORDER セクション・ヘッダ・フラグはサポートされていない

V8.2

Linker は、ELF セクション・ヘッダ・フラグ LINK_ORDER をサポートしていません。ただし、このフラグが設定されている unwind セクションは、正しい順序で配置されます。

このフラグは、セクションがイメージ・ファイル内の他のセクションと結合される場合、参照先のセクションが結合される順序と同じ順序で結合しなければならないことを示します。unwind セクションは、特別なケースとして扱われ、正しい順序で現れます。

次の図に、この概念を図示します。



VM-1134A-AI

5.26.6 data-reduced ELF オブジェクト・ライブラリとのリンクは推奨できない

V8.2

DCX data-reduced ELF オブジェクト・ライブラリは、ライブラリ内の連続領域には格納されません。その結果、最初にデータの展開を行う必要があるため、モジュールをプロセスの P2 空間に直接マッピングすることはできません。LBR\$MAP_MODULE ライブラリ・サービスが、モジュールを展開しプロセスの P2 空間にコピーします。この動作により、結果としてできた P2 空間内のページが、プロセス・クォータとしてカウントされます。

LBR\$UNMAP_MODULE ライブラリ・サービスは、これらのページを回復しますが、これらのページはヒープ領域解放リストに残り、プロセス・クォータとしてカウントされ続けます。そのため、Linker 操作の前に、あらかじめ DCX data-reduced ELF オブジェクト・ライブラリを展開することをお勧めします。これは永続的な状態です。

5.26.7 初期化されたオーバーレイ・プログラム・セクションの取り扱いについての誤りの修正

V8.3

以前のバージョンの I64 Linker では、0 で初期化されたオーバーレイ・プログラム・セクションは、誤って、互換性のある初期化と見なされていました。OpenVMS Version 8.2 と Version 8.2-1 の I64 Linker では 0 で初期化されたセクションは 0 以外の初期値を持つセクションと互換性があると見なされます。そのため、リンク操作時のモジュールの順番によっては、イメージに異なる初期値が設定されることがあります。

この問題は、Version 8.3 で修正されました。

5.26.8 リンカの修飾子/EXPORT_SYMBOL_VECTOR と/PUBLISH_GLOBAL_SYMBOLS の削除

V8.3

すべてのグローバル・シンボルをエクスポートするだけで共用可能イメージを作成しても、それだけではうまくいきません。以前のバージョンの I64 Linker は、/EXPORT_SYMBOL_VECTOR 修飾子と/PUBLISH_GLOBAL_SYMBOLS 修飾子を提供しており、すべてのグローバル・シンボルのシンボル・ベクタ・オプションをモジュール単位に作成できました。ただし、これらの修飾子を使用すると、異なるイメージに格納されている同じユニバーサル・シンボルがエクスポートされます。異なるイメージに格納されている同じ C++ テンプレートがエクスポートされてしまうことを除外できないため、これは問題になります。

どちらの修飾子も Version 8.3 から削除されました。

この問題は、OpenVMS for I64 の将来のリリースの I64 Linker で修正される予定です。

5.26.9 オプションでの長いシンボル名のサポート

V8.3

オプション用として、リンカの内部ライン・バッファが、2048 文字に拡大されました。これにより、最大サポート長が 1024 文字のシンボル名とともにオプションを指定できるようになりました。

5.26.10 リンカが作成したコード・スタブのメモリの使用方法の改善

V8.3

I64 Linker は、OpenVMS Debugger でのコードのステップ実行でも参照可能であるコード・スタブを生成します。コードはルーチン呼び出しに対して挿入され、そのサイズは異なることがあります。しかし、以前のリンカは、必要となりうる最大コード・サイズを使用して、固定サイズのコード・セクションとしてコードを追加していました。

バージョン 8.3 のリンカは、コード・スタブを実サイズで書き込みます。これにより、スタブ間の未使用スペースが無くなるため、未使用のメモリが解放される可能性があります。

5.26.11 コンパイラでのデマングル化されたシンボル名のサポート

V8.3

シンボル名を固有のものにするために、一部のプログラミング言語では名前のマングル化が使用されています。バージョン 8.3 からは、リンカはソース・コード名と呼ばれる、デマングル化された名前を表示しようとします。これらの名前は、リンカのメッセージに使用されます。また、要求によって、リンカ・マップに表示されるマングル化されたグローバル・シンボルの変換テーブルでも使用されます (『HP OpenVMS V8.3 新機能説明書』を参照)。この機能は、コンパイラでの名前のデマングル化のサポートに依存しています。

5.26.12 最大セクション数

V8.3

65280 を超えるセクションに対しては、ELF 形式は拡張された番号付け方式を使用します。これは、現在の I64 リンカでは実装されていません。そのため、単一の入力オブジェクト・モジュールまたは共用可能イメージを持つことのできるセクションの数が制限されます。通常リンカは複数のセクションを持つ共用可能イメージを作成するため、この制限は共用可能イメージを作成する際にも適用されます。ここで、ELF セクションは、C++ テンプレートや共用セクションをエクスポートするために使用されます。つまり、共用可能イメージ中のそのようなインタフェースの数は 65280 未満でなければなりません。

5.26.13 マップ・ファイル中の共用可能イメージの作成日が正しくない

V8.2

OpenVMS I64 プラットフォームでは、リンカ・マップ中の共用可能イメージの作成日が正しくない場合があります。誤った日付は、通常 3686 と表示されます。この状態になるのは、リンカが共用可能イメージを入力ファイルとして処理し、日付フィールドを抽出してマップに出力した場合です。ANALYZE/IMAGE で表示されるイメージ自体の日付は正しい内容になっています。

この問題は、I64 Linker の将来のリリースで修正されます。

5.27 LTDRIVER: CANCEL SELECTIVE の制限事項

永続的な制限事項

OpenVMS Version 6.1 より前のリリースでは、LTDRIVER は「拡張 DDT」ビットをセットしていませんでした。したがって、POSIX 関数 CANCEL SELECTIVE は LTDRIVER で動作しませんでした。この問題は解決されましたが、まだ制限事項が残っています。

この修正により、\$QIO 読み込みと書き込みを選択的に取り消すことができるようになりましたが、ポート・ドライバに対して行った \$QIO (つまり、LAT 接続 \$QIO などのように IOS_TTY_PORT 関数修飾子を使用して行ったもの) は、CANCEL SELECTIVE によって取り消すことができません。

5.28 Mail ユーティリティ: 呼び出し可能メールのスレッドの制限事項

V7.1

OpenVMS 呼び出し可能メール・ルーチンはスレッド・セーフではありません。スレッド化されたアプリケーション内での非スレッド・セーフ・ルーチンの呼び出しの詳細については、『Guide to the POSIX Threads Library』を参照してください。

呼び出し可能メールのコンテキスト情報は、プロセス単位 (スレッド単位ではない) で管理されるので、コンテキスト・ベースの処理を実行する複数のスレッドは相互に同期をとり、特定のタイプのメール・コンテキストが一度に1つだけアクティブになるようにしなければなりません。この条件が満たされないと、1つのスレッドが他のスレッドのメール操作を妨害する可能性があります。

OpenVMS Alpha システムでは、マルチスレッド環境でカーネル・スレッドが有効に設定されている場合、この他にも追加制限事項があります。この環境では、呼び出し可能メールは初期スレッドでのみ使用しなければなりません。

5.29 OpenVMS Debugger

以降の注意事項では、OpenVMS I64 システムと OpenVMS Alpha システムでの、OpenVMS Debugger に固有の注意事項について説明しています。

DELTA デバッガと XDELTA デバッガの注意事項については、第 5.13 節を参照してください。

5.29.1 デバッガのコマンド行ヘルプに STEP/SEMANTIC_EVENT に関する情報がない

V8.3

デバッガのコマンド行ヘルプのSTEPコマンドの下に/SEMANTIC_EVENT 修飾子に関する情報が正しく表示されません。この修飾子の情報は、『HP OpenVMS デバッガ説明書』のコマンド・リファレンスのセクションの、STEPコマンドの下にあります。

5.29.2 このリリースで修正された状況と問題点 (I64 のみ)

V8.3

次のリストは、このリリースで修正された問題点です。

全般的な問題点: 以前は、境界にアラインされていない要素が存在する配列は、正しく表示されませんでした。この問題は、プログラミング言語の構文を使用して不自然なデータ・アラインメントを強制した (たとえば、C の #pragma

nomember_alignや、Pascal のPACKEDキーワード) 場合に発生していました。この問題は修正されました。

COBOL 言語での問題: EDIT picture 節で宣言したデータは、部分的にしかサポートされていませんでした。この問題は修正されました。

Fortran 言語での問題: REAL*16 型のデータ項目に値を保存すると、実際の値ではなくゼロが格納されていました。この問題は修正されました。

Pascal 言語での問題:動的境界で宣言された配列 (たとえば "array [1..n]") が、正しく処理されないことがありました。この問題は修正されました。

5.29.3 全般的な問題点と回避方法 (I64 のみ)

V8.2

次に、確認されている全般的な状況と、ユーザの回避方法を示します。

問題点:要素が境界にアラインされていない配列は、正しく表示されません。これは、プログラミング言語の構文を使用して、不自然なデータ・アラインメントを強制した場合に発生します (たとえば、C 言語での#pragma nomember_align、Pascal でのPACKEDキーワード)。

回避方法:ありません。

問題点:添字がスカラ (整数) の配列を調べる時、デバッガは配列のインデックス値を 10 進数ではなく、現在定義されている基数で出力します。

回避方法: SET RADIX DECIMAL コマンドを実行します。

問題点: I64 システムの OpenVMS Debugger は、あたかもレジスタ・リネーム・ベース (CFM.rrb) とローテート・サイズ (CFM.sor) がいずれも 0 であるかのように、汎用レジスタ、浮動小数点レジスタ、およびプレディケート・レジスタを表示します。言い換えると、ローテートするレジスタを使用しているときは、ローテーションの効果は無視されます。この問題は、今後のリリースで修正される予定です。さらに詳細な説明は第 5.10 節を参照してください。

注意

この現象は、C++ およびアセンブラ言語のプログラムの、特殊な状況でまれに発生します。多くのプログラムは、この問題の影響を受けません。

回避方法: CFM レジスタを調べ、(ゼロでない) CFM.rrb フィールドと CFM.sor フィールドの値を考慮して、手作業で EXAMINE コマンドを調整します。

5.29.4 Basic 言語での問題 (I64 のみ)

V8.2

問題点: 多次元配列の範囲を調べるとき、デバッガは要素のアドレスを適切にシンボル化できません。つまり、配列の行と列の値が正しくありません。

回避方法: ありません。

5.29.5 C++ 言語での問題 (I64 のみ)

V8.2

問題点: SHOW CALLS コマンドが、C++ のマングル化されていない名前ではなく、マングル化された名前を表示することがあります。

回避方法: ありません。

5.29.6 COBOL 言語での問題 (I64 のみ)

V8.2

問題点: EDIT picture 節で宣言したデータは、部分的にしかサポートされません。EXAMINE コマンドを使用して項目を調べると、位取りなどの調整なしに生データが表示されます。整数や 10 進数の定数を保存することで項目を変更しようとする (DEPOSIT コマンドを使用) とエラーになります。

回避方法: 適切な位取り因数を設定し、表示される値を手作業で調整します。値を変更する必要がある場合は、別の変数または引用符付きの文字列をソースとして DEPOSIT コマンドを実行します。

5.29.7 Fortran 言語での問題 (I64 のみ)

V8.2

問題点: REAL*16 型のデータ項目に値を保存すると、実際の値ではなく常にゼロが格納されます。

回避方法: ありません。

5.29.8 Pascal 言語での問題 (I64 のみ)

V8.2

問題点:動的境界で宣言された配列 (たとえば "array [1..n]") が、正しく処理されないことがあります。配列を調べている際に、デバッガが INVARRDSC や IVALOUTBNDS のエラーを表示することがあり、SHOW SYMBOL/TYPE で配列境界が正しく表示されないことがあります。

回避方法:ありません。

5.29.9 SET SCOPE コマンド: 動作の変更

V8.2

SET SCOPE コマンドの動作は、このリリースで変更されました。SET SCOPE では、デバッガの言語設定を、指定されたスコープの言語に変更するようになりました。以前は、SET SCOPE は言語設定に影響を与えませんでした。SET LANGUAGE NODYNAMIC コマンドを実行することで、この機能を無効にできます。

5.29.10 SHOW IMAGE コマンドの変更

V8.2

SHOW IMAGE コマンドは、プロセスのイメージが占めるアドレス空間の概要を表示するようになりました。イメージ・セクション情報をすべて表示するには、SHOW IMAGE/FULL を使用するか、イメージ名を指定します (たとえば、SHOW IMAGE FOO)。

5.29.11 クライアント/サーバ・インタフェース: 以前のバージョンはサポートされない (Alpha のみ)

V7.3

OpenVMS Alpha Version 7.3 およびそれ以降の Debugger では、以前のバージョンのクライアント/サーバ・インタフェースをサポートしません。次の表に従って、配布メディアのキットにあるクライアント/サーバ・インタフェースをインストールしてください。

CPU	オペレーティング・システム	クライアント・キット
Intel	Microsoft Windows 95 , 98 , NT , Me , 2000 , XP	[DEBUG_CLIENTS011.KIT] DEBUGX86011.EXE
Alpha	Microsoft Windows NT	[DEBUG_CLIENTS011.KIT] DEBUGALPHA011.EXE

これらのクライアント・キットは、自己解凍形式の.EXE ファイルです。

適切な実行形式ファイルを PC に転送したら、そのファイルを実行して、PC 上にデバッグ・クライアントをインストールすることができます。InstallShield インストール・プロシージャのガイドに従ってインストールを行います。

5.30 OpenVMS のシステム・ダンプ・アナライザ (SDA)

ここでは、システム・ダンプ・アナライザ (SDA) に関する注意事項について説明します。

5.30.1 CLUE コマンドは OpenVMS I64 に移植されていない

V8.2

次の CLUE コマンドは、OpenVMS I64 にはまだ移植されていないため、その旨のメッセージを返します。

```
CLUE CALL
CLUE ERRLOG
CLUE FRU
CLUE REGISTER
```

5.31 OpenVMS I64 Version 8.2 に含まれない PL/I ライブラリ

V8.2

PL/I ライブラリ (ネイティブ・ライブラリと変換されたライブラリ) は、OpenVMS Alpha には含まれていましたが OpenVMS I64 には含まれていません。PL/I コア・イメージのうち OpenVMS I64 にはないものは以下のとおりです。

```
[SYSLIB]DPLI$RTL$SHR.EXE
[SYSMMSG]PLI$MSG.EXE
[SYSLIB]PLIRTL.IIF
[SYSLIB]PLIRTL_D56_TV.EXE
```

PL/I ライブラリを参照するアプリケーションや共有イメージは、Integrity サーバ用の OpenVMS I64 では動作しません。(一般的に、PL/I で記述したコードを含むアプリケーションや共有イメージがそうです。) この制約は、ネイティブ・コードにも、変換された VAX および Alpha イメージにも、適用されます。

第 2.6 節の、関連する注意事項を参照してください。

5.32 POSIX スレッド・ライブラリ

ここでは、POSIX スレッド・ライブラリ (旧名称は、DECthreads) に関する注意事項について説明します。

第 A.8 節の関連する注意事項も参照してください。

5.32.1 例外処理中のスタック・オーバフロー (I64 のみ)

V8.2

I64 システムでの例外処理には、Alpha システムの場合よりもかなり大きなスタック領域が必要です。OpenVMS からアプリケーションを移植するときに、例外処理を使用するスレッドに未使用スタック領域が十分ないと、I64 での例外処理中に、このスレッドでスタック・オーバフローが発生することがあります。通常は、次の操作のいずれかに関連する ACCVIO の処理が不適切であったように見えます。

- RAISE
- pthread_cancel
- pthread_exit
- スレッドにアクティブな TRY または pthread_cleanup_push ブロックがあり、OpenVMS の状態がシグナル通知された (たとえば、ハードウェア例外や、LIB\$SIGNAL または LIB\$STOP の使用など)。

このような問題が発生した場合は、スレッドに割り当てられるスタックのサイズを数ページずつ増やしてみてください。最初は、スタックのサイズを 24 KB 大きくすることをお勧めします。

デフォルトのスタック・サイズは、I64 システム上でのスタック使用量が多いことに対応するため、24 KB 大きくされました。アプリケーションが多数のスレッドをデフォルトのスタック・サイズで作成している場合、アプリケーションのメモリ・リソースが不足することがあります。このような状況になった場合は、プロセス・クォータを大きくするか、アプリケーションを変更して同時に存在するスレッドの数を減らしてください。

5.32.2 I64 システムでの THREADCP コマンドの動作

V8.2

OpenVMS I64 システム上の DCL コマンド THREADCP は、スレッド関連の 2 つのメイン・イメージ・ヘッダ・フラグ、UPCALLS と MULTIPLE_KERNEL_THREADS の問い合わせや変更には使用できません。代わりに、I64 システムでのスレッド・ヘッダ・フラグの設定や参照を行うための DCL コマンド SET IMAGE および SHOW IMAGE を使用する必要があります。

Alpha システムのユーザは、引き続き THREADCP コマンドを使用してください。

5.32.3 浮動小数点のコンパイルと例外 (I64 のみ)

V8.2

次の 2 つの古い cma スレッド・ライブラリ・ルーチンのいずれかを呼び出すソース・モジュールは、OpenVMS I64 上で使用するために/FLOAT=G_FLOAT コンパイラ修飾子 (または、言語固有の同等の修飾子) を指定してコンパイルしなければなりません。

```
cma_delay()  
cma_time_get_expiration()
```

これらのルーチンは、VAX 形式の浮動小数点数だけを引数として受け入れます。通常、OpenVMS I64 コンパイラは、デフォルトで VAX 形式を使用する OpenVMS Alpha コンパイラとは異なり、デフォルトで IEEE 形式の浮動小数点数を使用します。この 2 つの cma スレッド・ルーチンは、Alpha と I64 のどちらでも、VAX 形式の浮動小数点引数だけを受け入れます。これらのルーチンの呼び出しを適切にコンパイルしないと、IEEE 形式の浮動小数点数が実行時に誤って渡され、予期しない例外が発生することがあります。

5.32.4 C 言語コンパイル・ヘッダ・ファイルの変更

V7.3-2

OpenVMS Version 7.3-2 では、次の変更が行われました。

- INTS.H

OpenVMS の以前の一部のリリースでは、POSIX スレッドの C 言語ヘッダ・ファイル PTHREAD_EXCEPTION.H に、C ヘッダ・ファイル INTS.H の #include が誤って含まれていました。この問題は、OpenVMS Version 7.3-2 で修正されました。PTHREAD_EXCEPTION.H を使用しても、コンパイルで INTS.H がインクルードされなくなりました。アプリケーションによっては、コンパイル時に

INTS.H が必要で、PTHREAD_EXCEPTION.H でインクルードされることを誤って前提としているものがあるかもしれません。

このようなアプリケーション・ソース・モジュールを OpenVMS Version 7.3-2 システムで再コンパイルすると、C コンパイラから診断メッセージが出力されます。これらのメッセージは、INTS.H で定義されるシンボルやデータ型 (たとえば `int32`) が未定義であることを示します。このようなアプリケーション・ソース・モジュールを修正するには、該当するシンボルや型を最初に使用している位置より前で、直接 `<ints.h>` を `#include` します。

- `timespec_t` typedef

以前のリリースの OpenVMS では、POSIX スレッドの C 言語ヘッダ・ファイル PTHREAD.H に、`timespec_t` という名前の typedef 定義が入っていました。このシンボルは、PTHREAD.H に属さない、非標準のシンボルです。(この typedef は、TIME.H や TIMERS.H などの C RTL ヘッダ・ファイルの内容との関連で残されていました。) OpenVMS Version 7.3-2 では、C RTL とスレッド・ヘッダ・ファイルの標準への準拠性が改善されました。この結果、PTHREAD.H から `timespec_t` の typedef が削除されました。

アプリケーションによっては、`timespec_t` の typedef がコンパイル時に必要で、PTHREAD.H で直接的または間接的 (たとえば、TIS.H の使用による) に定義されることを誤って前提としているものがあるかもしれません。このようなアプリケーション・ソース・モジュールを OpenVMS Version 7.3-2 システムで再コンパイルすると、C コンパイラから、`timespec_t` を未定義のシンボルまたは型としてリストする診断メッセージが出力されます。このようなアプリケーション・ソース・モジュールを修正するには、`timespec_t` を使用している部分を `timespec` 構造体に置き換えるか、`timespec_t` シンボルを最初に使用している位置より前で、C RTL ヘッダ・ファイル TIMERS.H をインクルードします。

アプリケーションの構築環境で古い C RTL やスレッド・ヘッダ・ファイルのプライベート・コピーを使用していたり、`timespec` 構造体や `timespec_t` typedef を含む抜粋を使用している場合 (どちらの方法もお勧めできません) は、もっと多くのコンパイル・エラーが表示されることがあります。コンパイラは、`timespec` 構造体が再定義されている (2 回定義されている) ことを示すメッセージを表示することがあります。このような場合、システムが提供している C RTL およびスレッド・ヘッダ・ファイルを使用するように戻したり、`timespec` 構造体に関連して個別に抜粋した箇所を、システムが提供している TIME.H ヘッダ・ファイルのインクルードに変更します。

5.32.5 新しい優先順位調整アルゴリズム

V7.3-2

OpenVMS Version 7.3-2 では、『Guide to the POSIX Threads Library』で説明されている適応型スレッド・スケジューリング動作が、新しい優先順位調整アルゴリズムとともに実装されました。場合によっては、新しいアルゴリズムでは、優先順位が

異なる、スループット方針のスレッドが同期オブジェクトを共用することによる問題を回避できます。優先順位の調整により、アプリケーションのスループットや、システム全体の使用状況も改善できます。スループット・スケジューリング方針のスレッドの優先順位調整は、自動で、透過的に行われます。

5.32.6 プロセス・ダンプ

V7.3

POSIX スレッド・ライブラリで実行時に修正不能な重大エラー (アプリケーション内のデータ破損によって損傷したデータ構造など) が検出されると、ライブラリにより実行中のイメージが終了されることがあります。終了中に、ライブラリによりプロセス・ダンプ・ファイルの作成がトリガーされます (このファイルは、ANALYZE/PROCESS_DUMP によりエラー診断に使用されます)。このようなプロセス・ダンプ・ファイルのサイズは、エラー時のプロセスのアドレス空間に依存するため、非常に大きくなることがあります。

5.32.7 動的 CPU 構成の変更

V7.3

OpenVMS Version 7.3 以降、POSIX スレッド・ライブラリは、マルチプロセッサ Alpha システムを実行する CPU の数の動的変化に対応するようになりました。1 つのイメージに対して、複数のカーネル・スレッドが使用できるように指定 (LINK /THREADS_ENABLE 修飾子または THREADCP コマンド動詞により) すると、POSIX スレッド・ライブラリが、アプリケーションの明白な並列処理を監視して、利用可能な CPU の数を最大とする数のカーネル・スレッドを作成します。それぞれのカーネルスレッドは、OpenVMS エグゼクティブによってスケジューリングされて別々の CPU で実行されるので、同時に実行することができます。

アプリケーションの実行中、オペレータは CPU を個別に停止または開始することができます。このような動的変化を反映して、これ以降にイメージがアクティブ化されたときに作成できるカーネル・スレッドの数が増えます。また、現在実行中のイメージにも反映されるようになりました。

CPU を追加または除去すると、スレッド・ライブラリは、追加、除去後のアクティブな CPU の数を照会し、プロセスが現在使用しているカーネル・スレッドの数と比較します。現在 CPU がカーネル・スレッドよりも多い場合、ライブラリは既存の POSIX スレッドを CPU まで延長します (必要に応じて、すぐに、または後に新しいカーネル・スレッドを作成します)。逆に CPU がカーネル・スレッドよりも少ない場合、ライブラリは余分のカーネル・スレッドを強制的にハイバネートさせ、残りのカーネル・スレッド上で POSIX スレッドを再度スケジューリングします。これにより、プロセスに関する限り、利用可能な数以上のカーネル・スレッドが、CPU リソースを奪い合うということがなくなります。

5.32.8 デバッガ計測機能は動作しない

V7.0

POSIX スレッド・デバッガの計測機能は動作しません。

『Guide to the POSIX Threads Library』の C.1.1 に記載されている、動作中のプログラムをデバッグする手順を使用すると、プロセスが ACCVIO メッセージで失敗する可能性があります。

5.33 RTL ライブラリ (LIB\$)

ここでは、LIB\$ランタイム・ライブラリに関する注意事項について説明します。

5.33.1 RTL ライブラリ (LIB\$) のヘルプ

V8.2

OpenVMS Version 8.2 の LIB\$ランタイム・ライブラリのヘルプ・ファイルには、LIB\$LOCK_IMAGE ルーチンのヘルプがありません。この問題は、今後のリリースで修正される予定です。当面は、このルーチンの詳細な説明は『OpenVMS RTL Library (LIB\$) Manual』を参照してください。

5.33.2 RTL Library (LIB\$): 呼び出し標準ルーチン (I64 のみ)

V8.2

この注意事項では、ローテートするレジスタが、以下の呼び出し標準ルーチンでどのように取り扱われるかを明確化します。

LIB\$I64_GET_FR
LIB\$I64_SET_FR
LIB\$I64_GET_GR
LIB\$I64_SET_GR
LIB\$I64_PUT_INVO_REGISTERS

呼び出し標準規則の ICB (invocation context block) およびメカニズム・ベクタは常に、あたかも、レジスタ・リネーム・ベース (CFM.rrb) とローテート・サイズ (CFM.sor) がいずれも 0 であったかのように、汎用レジスタ、浮動小数点レジスタ、およびプレディケート・レジスタを記録しています。言い換えると、ローテートするレジスタを使用しているときに、ローテーションの効果が無視されます。このことは、LIB\$I64_PUT_INVO_REGISTERS ルーチンが使用するレジスタ・マスクについても同様です。というのは、これらのマスクは、ICB 構造体のフィールドによって定義されるからです。

現在は、補足的なアクセス・ルーチン LIB\$I64_GET_FR, LIB\$I64_SET_FR, LIB\$I64_GET_GR および LIB\$I64_SET_GR が、レジスタ・リネーム・ベース・レジスタとローテート・サイズ・レジスタの効果を調整しないで、不適切に、そのレジスタ番号パラメータを解釈しています。これは、誤りであり今後のリリースで修正される予定です。

それまでは、ICB またはメカニズム・ベクタ内の汎用レジスタ、浮動小数点レジスタ、およびプレディケート・レジスタを調べるプログラムや、実行時に見えるレジスタを探して内容を解釈するプログラムでは、保存された CFM レジスタを調べて、自身で適切に調整する必要があります。

5.34 Screen Management (SMG\$) のドキュメント

『OpenVMS RTL Screen Management (SMG\$) Manual』の最後にある参照情報のトピックに、次の情報を追加します。

V7.2

- ルーチン SMG\$DELETE_VIRTUAL_DISPLAY の「Condition Values Returned (返される条件値)」に、次の説明を追加してください。

"Any condition value returned by the \$ DELPRC system service"
(\$DELPRC システム・サービスから返された条件値)

- ルーチン SMG\$GET_TERM_DATA の説明の「Arguments (引数)」の部分で、capability-data 引数の説明が誤っています。正しい説明は次のとおりです。

access: write-only
mechanism: by reference, array reference

- ルーチン SMG\$SET_OUT_OF_BAND_ASTS の説明の「引数 (AST-argument)」の部分で、AST-argument 引数の説明に誤りがあります。構造体の図のシンボル名が誤っています。この図の下にある段落に示されているシンボル名は正しい名前です。正しいシンボル名と誤ったシンボル名は次のとおりです。

誤っているシンボル名	正しいシンボル名
SMG\$P_PASTEBOARD_ID	SMG\$P_PBD_ID
SMG\$P_ARG	SMG\$P_USER_ARG
SMG\$B_CHARACTER	SMG\$B_CHAR

V7.1

- SMG\$READ_COMPOSED_LINE ルーチンの説明で、flags 引数の説明に次の文を追加してください。

"The terminal characteristic /LINE_EDITING should be set for your terminal for these flags to work as expected. /LINE_EDITING is the default."

(「これらのフラグが正しく機能するには、端末に対して端末属性/LINE_EDITING を設定しなければなりません。/LINE_EDITING は省略時の設定です。」)

- ルーチン SMG\$SET_KEYPAD_MODE の説明に、次の注意を追加してください。

注意

キーボード・モードを変更すると、物理端末の設定も変更されます。これは、keyboard-id 引数によって指定される仮想キーボードだけでなく、すべての仮想キーボードに対するグローバルな変更です。

5.35 SORT32 ユーティリティ

ここでは、OpenVMS Alpha および OpenVMS I64 Version 8.2 用の、SORT32 V08-010 に関する注意事項について説明します。詳細は、第 5.20.8 項と第 5.20.1 項を参照してください。

Hypersort で修正されていない問題を回避する場合、または Hypersort に実装されていない機能を使用する場合に SORT32 を使用することをお勧めします。Hypersort の注意事項については、第 5.20 節を参照してください。

5.35.1 DFS サービス・ディスクでの CONVERT の問題

V8.2

SORT, MERGE, および CONVERT 操作は、UNIX がサービスする DFS マウント・ディスクが出力先になっている場合、%SORT-E-BAD_LRL エラーを返します。

この制約事項を回避するには、次のいずれかを実行します。

- 出力ファイルを OpenVMS ディスクに書き込んでから、その出力ファイルを UNIX がサービスする DFS マウントディスクにコピーする。
- CONVERT/FDL コマンドを使用する。このとき、FDL には、出力ファイルに必要な LRL (最大レコード長) を指定する。

5.35.2 一時作業ファイルが削除されないことがある

V7.3-2

SORT32 は、一時作業ファイルを削除しないことがあります。SY\$SCRATCH や、SORT32 の作業ファイルを置いている場所を定期的にチェックし、削除されていない

作業ファイルを削除してディスク・スペースを空けることができないかを調べてください。

5.35.3 複合条件のある SORT/SPECIFICATION: 要件

V7.3-1

SORT32 では、キー指定ファイルの複合条件が括弧で囲まれていない場合、複合条件に対する診断メッセージを出力しません。例を次に示します。

誤り:

```
/CONDITION=(NAME=TEST1, TEST=(Field2 EQ "X") AND (Field3 EQ "A"))
```

正しい:

```
/CONDITION=(NAME=TEST1, TEST=((Field2 EQ "X") AND (Field3 EQ "A")))
```

5.35.4 可変長レコードでの性能の問題

V7.3-1

SORT32 では、入力ファイル内の最大レコード長 (LRL) 情報に基づいて、ソート作業ファイルの固定長のスロットが割り当てられます。性能を向上させるには、実際の最大レコード長に最も近い LRL 情報を入力ファイルに設定します。初期性能が低い場合は、C プログラムによって作成されたファイルをソートしており、LRL が必要以上に大きく (32767 まで) 設定されていることが原因と考えられます。

5.35.5 作業ファイル・ディレクトリの制約事項

V7.3

SORT32 の作業ファイルは、必要な数の作業ファイルを複数のファイル・バージョンにわたって格納できるディレクトリに作成する必要があります。

5.36 システム・サービス

ここでは、OpenVMS のシステム・サービスに関する注意事項について説明します。

5.37 タイマ・キュー・エントリ (TQE)

永続的な制限事項

OpenVMS Alpha Version 7.3-1 では、タイマ・キュー・エントリの管理方法が変更され、多くの TQE を使用するシステムの性能が大きく向上しました。この変更は、非特権アプリケーションにとっては無関係です。

また、特権コードで TQE を直接操作することはできません。特に TQE キュー・ヘッダ (TQESL_TQFL/TQESL_TQBL) 内のポインタに直接アクセスすると、通常はアクセス違反になります。ただし、特権コードで内部ルーチン `exe_std$instimq` / `exe$instimq` と `exe_std$rmvtimq` / `exe$rmvtimq` を使用して、タイマ・キュー・エントリを入力または削除することは可能です。

5.38 Watchpoint ユーティリティ (I64 のみ)

V8.2

Watchpoint ユーティリティは、OpenVMS I64 に移植されていません。弊社では、このユーティリティを今後のリリースで移植する予定です。

5.39 プログラム全体の浮動小数点モード (I64 のみ)

V8.3

OpenVMS Alpha では、浮動小数点丸め動作、例外動作、および精度制御は、コンパイル時に定義されます。各モジュールは、それぞれの浮動小数点動作の設定で、個別にコンパイルされます。たとえば、計算のオーバフローでオーバフロー例外がシグナル通知されるディレクティブで 1 つのモジュールをコンパイルし、別のモジュールを、計算のオーバフローで例外をシグナル通知するのではなく、値を InfinityT とするディレクティブでコンパイルすることができます。これらの 2 つのモジュールがコンパイルされ実行された場合、モジュールのコードは、コンパイル時に指定されたオーバフロー動作をします。

OpenVMS I64 では、浮動小数点丸め動作、例外動作、および精度制御は実行時に定義され、プログラム全体の浮動小数点モードの概念で制御されます。プログラム全体の浮動小数点モードでは、プログラムのメイン・エントリ・ポイント (リンカが決定したもの) を含むモジュールが、デフォルトの浮動小数点丸め動作、例外動作、および精度制御を定義するモジュールです。

大半のプログラムには、この相違点の影響はありません。要点は、ホワイト・ペーパー『Intel® Itanium®アーキテクチャにおける OpenVMS 浮動小数点演算について』を参照してください。このドキュメントは、次の Web サイトで参照できます。

<http://h71000.www7.hp.com/openvms/integrity/resources.html>

5.40 HP OpenVMS Debugger の Heap Analyzer の問題点と回避策 (I64 のみ)

V8.2-1

問題点: Heap Analyzer の起動時, また I64 用のデバッガ上で Heap Analyzer の START コマンド (START HEAP_ANALYZER) を実行したとき, アップコールを有効にしているマルチスレッド・アプリケーションがあるとハングします。

回避策: スレッドまたは AST を使っているアプリケーションに対しては, デバッグ・イベントを設定する前, またはデバッグ・イベントを無効またはキャンセルした後に Heap Analyzer を起動するようにしてください (Heap Analyzer を起動した後, START コマンドがデバッガの制御をユーザに戻した後にイベントを有効化/リセットすることができます)。

ハードウェアに関する注意事項

この章では、以下のハードウェア製品についての情報を示します。

- MP コンソールと BMC コンソール (第 6.1 節)
- AlphaServer 2100 (第 6.2 節)
- AlphaServer 8200/8400 (第 6.3 節)
- AlphaServer ES47/ES80/GS1280 システム (第 6.4 節)
- AlphaServer GS シリーズ (第 6.5 節)
- AlphaStation 200/400 (第 6.6 節)
- AlphaStation 255 (第 6.7 節)
- ATI RADEON 7000 グラフィック (第 6.8 節)
- ATI RADEON 7500 グラフィック (第 6.9 節)
- DECwindows X11 ディスプレイ・サーバ (第 6.10 節)
- DIGITAL モジュラ・コンピューティング・コンポーネント (第 6.11 節)
- Digital パーソナル・ワークステーション (第 6.12 節)
- デュアル・コントローラ HSGnnデバイス (第 6.13 節)
- Open3D グラフィック (第 6.14 節)
- PowerStorm 300/350 PCI グラフィック・コントローラ (第 6.15 節)
- RFnn DSSI ディスク・デバイス (第 6.16 節)
- RZnnディスク・デバイス (第 6.17 節)
- sx1000 Integrity Superdome (第 6.18 節)
- ZLX グラフィック・ボード (第 6.19 節)

デバイス・ドライバの使用についての注意事項も、この章の最後にまとめてあります。

6.1 MP コンソールと BMC コンソールの制約事項 (I64 のみ)

ここでは、MP コンソールと BMC コンソールに関する注意事項について説明します。

6.1.1 入力デバイス，出力デバイス，およびエラー・デバイスの制約事項

V8.2

現在，OpenVMS オペレーティング・システムでは，各 MP コンソールや BMC コンソールの入力デバイス，出力デバイス，およびエラー・デバイスは，同じシリアル回線コンソールに対応していなければなりません。システムに MP コンソールがある場合は，このコンソールを使用してください。

指定されたコンソールからブートしないと，VMS_LOADER に警告が送信され，ブート中にその他のエラーが表示されることがあります。また，通常のブート時に表示される情報が出力されないこともあります。

6.1.2 削除キーへの Ctrl/H の再マッピング

V8.2

文字 0X7F を DEL/RUBOUT 用に使用する OpenVMS オペレーティング・システムとは異なり，MP コンソール，BMC コンソール，および EFI コンソール環境では，Ctrl/H を使用します。VTxxx 端末上の削除キーを押したり，ターミナル・エミュレータで 0X7F を送信するようにマッピングされているキーを押しても，文字は削除されません。

注意: ドライバは，次の条件では再マッピングを行いません。

- 端末が IOS_READALL 状態の場合
- 端末が IOS_READPBLK 状態の場合
- 端末属性に PASSALL が設定されている場合
- 端末属性に PASTHRU が設定されている場合
- Ctrl/V が押されたことにより，ドライバが，次の文字を渡し，再マッピング・チェックを行わないようにされた場合

6.2 AlphaServer 2100

ここでは，AlphaServer 2100 シリーズのコンピュータ固有の情報をまとめます。

6.2.1 コンソール表示

V7.2

AlphaServer 2100 システムと 2100A システムで表示される次のようなコンソール表示は正常であり，システム・エラーを示しているわけではありません。

```
P00>>>SET CONSOLE SERIAL
P00>>>INIT

VMS PALcode X5.48-112, OSF PALcode X1.35-81

starting console on CPU 0
initialized idle PCB
initializing semaphores
initializing heap
initial heap 1c0c0
memory low limit = 132000
heap = 1c0c0, 13fc0
.
.
.
probing hose 0, PCI
probing PCI-to-EISA bridge, bus 1
probing PCI-to-PCI bridge, bus 2
*** unable to assign PCI base address
*** bus 2, slot 7, function 0, size 00001000 (16 bit I/O)
bus 1, slot 1 -- fra -- DEFEA
bus 1, slot 2 -- vga -- Compaq Qvision
bus 1, slot 3 -- pua -- KFESA
bus 2, slot 1 -- pka -- NCR 53C810
bus 2, slot 6 -- pkb -- NCR 53C810
bus 2, slot 7 -- pkc -- DEC KZPSA
bus 0, slot 7 -- ewa -- DECchip 21041-AA
initializing keyboard
Memory Testing and Configuration Status


| Module | Size | Base Addr | Intlv Mode | Intlv Unit | Status |
|--------|------|-----------|------------|------------|--------|
| 0      | 64MB | 00000000  | 1-Way      | 0          | Passed |


Total Bad Pages 0
Testing the System
Testing the Disks (read only)
Testing the Network
econfig: 20041 99
econfig: 20042 04
econfig: 20043 00
AlphaServer 2100A Console V4.3-130, built on Oct 26 1996 at 19:44:57
P00>>>P
```

この表示では、KZPSA アダプタは正しくインストールされていますが、次のエラー・メッセージが表示されます。

```
*** unable to assign PCI base address
*** bus 2, slot 7, function 0, size 00001000 (16 bit I/O)
```

6.2.2 SCSI コントローラの制限事項

V6.2

1 GB 以上のメモリを装備した AlphaServer 2100 システムでは、Adaptec 1740 /1742 SCSI コントローラ (PB2HA-SA) はサポートされていません。コントローラがこのようなシステムに接続されていると、次のメッセージがオペレータのコンソールに表示されます。

```
%PKJDRVR-E- The direct DMA window does not map all of memory.  
Port is going OFFLINE.
```

6.3 AlphaServer 8200/8400: FRU テーブル・エラー

V7.2

エラー・ログ・バッファのサイズはシステム・パラメータ ERLBUFFERPAGES で制御され、最大値は 32 ページレットです。AlphaServer 8200/8400 または 4100 システムで、OpenVMS Alpha オペレーティング・システムのブート時に FRU (Field Replaceable Unit) テーブルがこの上限を超える場合には、エントリはエラー・ログ・ファイルに書き込まれません。

6.4 AlphaServer ES47/ES80/GS1280 システム

ここでは、AlphaServer ES47/ES80/GS1280 システムに関する注意事項について説明します。第 6.5.2 項の注意事項も、これらのシステムに関係します。

6.4.1 ES47/ES80/GS1280 ソフト・パーティションでの INIT コンソール・コマンドの使用

V8.2

同じハード・パーティション内で他のソフト・パーティションがブートされ、OpenVMS が実行されている場合には、ES47/ES80/GS1280 ソフト・パーティションでの INIT コンソール・コマンドの使用はサポートされません。INIT コマンドを実行すると、OpenVMS を実行中の他のインスタンスでシステム・クラッシュが発生することがあります。INIT コマンドを実行する前に、他のインスタンスをシャット・ダウンしてください。

コンソールの INIT の処理中に、同一ハード・パーティション内の他のインスタンスに対してブートコマンドを発行しないでください。INIT コマンドの完了までお待ちください。

6.4.2 RAD のサポート

V7.3-2

OpenVMS でのリソース・アフィニティ・ドメイン (RAD) のサポート (NUMA サポートまたは NUMA 対応とも言います) は、AlphaServer ES47/ES80/GS1280 システム用の OpenVMS Alpha Version 7.3-2 ではテストされていません。RAD サポートについての詳細は、『OpenVMS Alpha パーティショニングおよび Galaxy ガイド』を参照してください。

6.4.3 ライセンス要件

V7.3-2

AlphaServer ES47/ES80/GS1280 システムには、少なくとも 2 つの OpenVMS ソフトウェア・ライセンス (ベース・サポート用と、最初の 2 つのプロセッサのデュアル SMP サポート用) が必要です。この要件は、OpenVMS AlphaServer SMP システムの以前のライセンス付与方法から変更されています。OpenVMS のデュアル SMP ライセンスは、OpenVMS システムを購入したとき、または OpenVMS システムの追加 CPU モジュールを購入したときに、CPU モジュールに含まれています。

6.4.4 STOP/CPU およびシャットダウン動作

V7.3-2

ハードウェアの制約により、I/O ドロアを装備した AlphaServer ES47/ES80/GS1280 システム上の CPU は、DCL コマンドの STOP/CPU を使用して停止することができません。一方、I/O ドロアを装備していないシステム上の CPU は、このコマンドで停止できます。

I/O ドロアを装備した ES47/ES80/GS1280 システムでシャットダウン・プロシージャを起動すると、次のようなエラー・メッセージが表示されることがあります。

```
%SYSTEM-W-WRONGSTATE, CPU 5 is in the wrong state for the requested operation
```

このメッセージは無視して構いません。シャットダウンは正常に実行されます。

6.4.5 MBM での時刻の設定

V7.3-2

AlphaServer ES47/ES80/GS1280 システムでは、MBM で正しい時刻と日付を設定しなければなりません。正しく設定しないと、OpenVMS は誤った日時を表示することがあります。

6.5 AlphaServer GS シリーズ・システム

ここでは、AlphaServer GS シリーズ・システムの大半のユーザに関係する注意事項について説明します。

6.5.1 AlphaServer GS80/160/320 システム: デバイスの制限事項

永続的な制限事項

OpenVMS Alpha Version 7.3 以降では、1 つのパーティションあたり、従来型のバス・アダプタ上のデバイスが 1 セットだけ設定され、サポートされます。このようなデバイスは、次のとおりです。

- シリアル・ポート COM1 および COM2
- パラレル・ポート
- キーボード
- マウス

従来型のバス・アダプタが複数ある場合には、コンソール・ポートを含むアダプタだけが設定され、サポートされます。

6.5.2 OpenVMS Galaxy ライセンスの実行

V7.3

OpenVMS Galaxy のコンピューティング環境では、システム・スタートアップ時やインスタンス間での CPU の再割り当てが発生するつど、OPENVMS-GALAXY ライセンス・ユニットがチェックされます。

CPU を起動しようとしたときに十分な OPENVMS-GALAXY ライセンス・ユニットがないと、CPU はインスタンスの設定されたセットの中に引き続き残りますが、動作は停止します。その後、適切なライセンス・ユニットをロードすれば、停止している CPU をシステムの実行中に起動することができます。これは、CPU が単独の場合でも複数の場合でも同じです。

6.5.3 ライセンスのインストール

V7.3-1

Version 7.3-1 以降にアップグレードする前に、次の手順を実行して、共通ライセンス・データベースが、ハード・パーティションとソフト・パーティション間でライセンス・ユニットを共有できるようにします。

1. 必要なユニット数を計算します。
 - OpenVMS 基本ライセンスをロードする。
 - SMP ライセンスをロードする。

- 次のコマンドを使用して、ライセンス・ユニットの数が適切か確認する。

```
$ SHOW LICENSE /UNIT_REQUIREMENTS /CLUSTER
```

注意

OpenVMS 基本ライセンスでは、パーティションではなく物理システムごとに、1 人の操作ユーザのみを割り当てることができます。ただし、各パーティションの OPA0: からいつでもログインできます。操作ユーザを追加する場合は、追加のライセンス・ユニットが必要です。必要なライセンス・ユニット数については、弊社のサポート担当者にお問い合わせください。

2. 共通ライセンス・データベースにライセンスを追加します。たとえば、次のように入力します。

```
$ LICENSE REGISTER license-name /ISSUER=DEC -  
_ $ /AUTHORIZATION=USA123456 -  
_ $ /PRODUCER=DEC -  
_ $ /UNITS=1050 -  
_ $ /AVAILABILITY=H -  
_ $ /OPTIONS=(NO_SHARE) -  
_ $ /CHECKSUM=2-BGON-IAMA-GNOL-AIKO
```

LICENSE REGISTER コマンドで/INCLUDE 修飾子を使用して、ライセンスの NO_SHARE 属性を変更することはできません。

3. ライセンスを編集して、LICENSE REGISTER /INCLUDE=(*node-name-list*) コマンドで PAK の NO_SHARE 属性を変更します。たとえば、次のように入力します。

```
$ LICENSE MODIFY OPENVMS-ALPHA /INCLUDE=(NODEA, NODEB, NODEC)
```

4. 各パーティションで実行している OpenVMS のインスタンスで OpenVMS Alpha のライセンス・ユニットを利用できるようにするために、SRM 環境変数 SYS_SERIAL_NUM が各パーティションで同じであるか確認します。確認するには、次の手順を実行します。
 - a. 各パーティションのマスタ・コンソール (通常はコンソール・ライン 0) で、SHOW SYS_SERIAL_NUM コマンドを使用して、システムのシリアル番号を表示します。たとえば、次のように入力します。

```
P00>>>  
P00>>>SHOW SYS_SERIAL_NUM  
sys_serial_num G2A105
```

SYS_SERIAL_NUM の値がブランクの場合は、他の各パーティションのマスタ・コンソールで SHOW SYS_SERIAL_NUM コマンドを使用して、ブランクでないシステム・シリアル番号がないかチェックします。

注意

すべてのパーティション・コンソールに SYS_SERIAL_NUM の値がブランクで表示される場合は、0 以外の値を 12 文字以内で作成してください。作成

するシステム・シリアル番号が、同じ OpenVMS Cluster の他の AlphaServer GS80/160/320 で使用されていないことを確認してください。

- b. システムのシリアル番号が決まったら、各パーティションのマスタ・コンソールで SET SYS_SERIAL_NUM コマンドを使用して、SYS_SERIAL_NUM の値を適切な値に変更します。たとえば、次のように変更します。

```
P00>>>  
P00>>>SET SYS_SERIAL_NUM G2A105
```

この手順は、すべてのハード・パーティションとソフト・パーティションで実行する必要があります。

5. OpenVMS Cluster ライセンス・データベースを正しく更新するために、すべての OpenVMS Cluster 共通ノードをシャットダウンしてからリブートすることをお勧めします。ローリング・アップグレード・タイプのブートでは、共通ライセンス・データベースは適切に更新されません。

注意

システムが、共通ライセンス・データベースを共有する OpenVMS Cluster の一部である場合、AlphaServer GS80/160/320 のハード・パーティションとソフト・パーティションの数を再構成する際には、必ずすべてのパーティションの SYS_SERIAL_NUM を同じ値にしてください。

パーティション間で NO_SHARE ライセンスを共有しているパーティション化可能なマシンでは、システムのブート時に次のエラー・テキストが表示される場合があります。

```
%LICENSE-E-NOAUTH, DEC OPENVMS-ALPHA use is not authorized on this node  
-LICENSE-F-EXCEEDED, attempted usage exceeds active license limits  
-LICENSE-I-SYSMGR, please see your system manager  
Startup processing continuing...
```

このエラー・テキストは無視して構いません。このテキストは、OPENVMS-ALPHA PAK を共有しているシステムにユーザがログインして、使用中になった場合に表示されます。この問題は今後のリリースで修正される予定です。

6.5.4 AlphaServer GS60/GS60E/GS140 複数 I/O ポート・モジュール構成の制限事項

V7.2-1

複数の I/O ポート・モジュール (KFTHA-AA または KFTIA-AA) がある AlphaServer GS60/GS60E/GS140 構成では、システム障害が発生することがあります。

GS60/GS60E/GS140 システムの複数 I/O ポート・モジュールを持つ OpenVMS Galaxy および非 Galaxy の AlphaServer 8200/8400 構成をアップグレードするときには、Compaq Action Blitz # TD 2632 に記載されているように、最低でもリビジョン B02 KN7CG-AB EV6 CPU (E2063-DA/DB rev D01) のモジュールを 1 つインストールしなければなりません。

この制限事項と解決方法についての詳細は、Compaq Action Blitz # TD 2632 を参照してください。

6.6 AlphaStation 200/400: ISA_CONFIG.DAT の変更が必要

V7.1

AlphaStation 200/400 ファミリ・システムで ISA 装置を構成する場合には、各デバイスのノード情報が各デバイス記述ブロックの最後に格納されるように、SYS\$MANAGER:ISA_CONFIG.DAT ファイルを変更しなければなりません。

重要

OpenVMS Version 6.2 または 7.0 システムからアップグレードする場合は、アップグレード手順を開始する前に、この変更を行わなければなりません。

表 6-1 に、デバイス記述ブロックの変更点を示します。

表 6-1 デバイス記述ブロックの変更点

Version 7.1 より前	Version 7.1 以降
[AUA0]	[AUA0]
NAME=AU	NAME=AU
NODE=3	DRIVE=SYS\$MSBDriver
DRIVER=SYS\$MSBDriver	IRQ=9
IRQ=9	DMA=(0,1)
DMA=(0,1)	PORT=(388:4,530:8)
PORT=(388:4,530:8)	NODE=3

SYS\$MANAGER:ISA_CONFIG.DAT ファイルを使用している場合には、第 A.7 節を参照してください。

6.7 AlphaStation 255: PCI 構成の制限事項

V7.1

AlphaStation 255 シリーズ・システムの OpenVMS Alpha オペレーティング・システムでは、PCI スロット 0 に PCI オプション・カードを構成することはできません。

PCI スロット 0 は、AlphaStation 255 シリーズ・システム上の、最下位の物理 PCI オプション・スロットです。このスロットの割り込みシグナルは、組み込みイーサネット・ポートと共用されます。OpenVMS Alpha オペレーティング・システムは現在、PCI デバイスが割り込みラインを共用することを許していないため、スロット 0 に PCI デバイスを設置すると、正しく動作しないか、組み込みのイーサネット・ポートでエラーが発生する原因となることがあります。この制限事項があるため、AlphaStation 255 シリーズ・システムがサポートする PCI オプション・カードの数は、最大で 2 枚 (スロット 1 とスロット 2 に構成) です。

6.8 ATI RADEON 7000 グラフィック (I64 のみ)

V8.2

ここでは、OpenVMS I64 システムで、組み込み型 ATI RADEON 7000 グラフィック・アダプタを使用する際の注意事項について説明します。

注意: 第 6.9.1 項で説明しているリソースの要件は、組み込み型 ATI RADEON 7000 グラフィック・アダプタにも適用されます。

6.8.1 I64 のグラフィック・サポート

V8.2

OpenVMS I64 システム上で現在サポートされているグラフィック・アダプタは、組み込み型 RADEON 7000 PCI アダプタだけです。

ATI RADEON 7500 PCI グラフィック・アダプタは OpenVMS I64 Version 8.2 で近い将来サポートされる予定です。現在テストが進行中です。このグラフィック・カードのサポートが可能になったら、下記の Web サイトで発表されます。

<http://h71000.www7.hp.com/new/>

6.8.2 RADEON 7000 ではハードウェア・アクセラレーション 3D グラフィックはサポートされていない

V8.2

ハードウェア・アクセラレーションによる 3D (OpenGL) レンダリングは、組み込み型 RADEON 7000 PCI アダプタではサポートされていません。

6.9 ATI RADEON 7500 グラフィック

V7.3-2

ここでは、ATI RADEON 7500 グラフィックのリソース要件、拡張機能、修正点および制限事項について説明します。HP DECwindows Motif for OpenVMS に関するドキュメントを参照したい場合、特に『Managing DECwindows Motif for OpenVMS Systems』を参照したい場合は、このドキュメントと、その他のドキュメントを、次の Web サイトから入手できます。

<http://www.hp.com/go/openvms/doc>

OpenVMS Version 8.2 より、3D サポートのライセンス (使用権) は、OpenVMS のライセンスの一部として含まれています。詳細については、第 6.14 節を参照してください。

6.9.1 リソースの要件

RADEON グラフィックのサポートには、次に示す、システム単位のリソースが必要です。

- 2 個のグローバル・セクション
- 296 KB のグローバル・メモリ

さらに、RADEON カードごとに、次のリソースが必要です。

- 3 個のグローバル・セクション
- 16 MB プラス 1 ページのグローバル・メモリ
- 16 MB プラス 1 ページのバッファ・オブジェクト・スペース (32 ビット System Space Windows)

グローバル・セクションの必要量は GBLSECTIONS システム・リソースに加算され、16M バイト強のグローバル・メモリは GBLPAGES リソースと GBLPAGFIL リソースに加算されます。

たとえば、1 枚の RADEON カードには、次のリソースが必要です。

- 5 個のグローバル・セクション
- 16 MB + 8 KB + 296 KB のグローバル・メモリ

これらの数は、次の値に相当します。

GBLSECTIONS	5
GBLPAGES	33376 (GBLPAGES の単位は、512 バイト・ページレット)

GBLPAGFIL 2086 (GBLPAGFIL の単位は、8192 バイトの Alpha ページ)

4 カード構成の場合は、次のリソースが必要です。

- 14 個のグローバル・セクション
- $296 \text{ KB} + 4 \times 16 \text{ MB} + 4 \times 8 \text{ KB} = 64 \text{ MB} + 328 \text{ KB}$ のグローバル・メモリ

これらの数は、次の値に相当します。

GBLSECTIONS 14

GBLPAGES 131728 (GBLPAGES の単位は、512 バイト・ページレット)

GBLPAGFIL 8233 (GBLPAGFIL の単位は、8192 バイト Alpha ページ)

6.9.2 DECW\$OPENGLSHR_RADEON.EXE を DECW\$MESA3DSHR_RADEON.EXE にリネーム

V8.2

共有ライブラリ SYSS\$LIBRARY:DECW\$OPENGLSHR_RADEON.EXE は、Mesa 3D コードをベースとしていることを反映させるために、リネームされて SYSS\$LIBRARY:DECW\$MESA3DSHR_RADEON.EXE になりました。API と機能は以前のリリースと同じです。新しいファイル指定の共有ライブラリを指定するための論理名 DECW\$OPENGLSHR が定義されています。

6.9.3 サポートの制限事項

V7.3-2

次の機能は、サポートされていません。

- S ビデオ出力
- デジタル出力
- デュアル・ヘッド動作

DVI ポートとアナログ VGA (CRT) ポートの両方にモニタを接続すると、両方のスクリーンに同じビデオ出力が得られます。これを、クローン・ビデオと言います。各ポートに独立したビデオ表示を行う本当の意味でのデュアル・ヘッド動作は、将来のリリースでサポートされます。

6.9.4 高リフレッシュ・レートでの画像への影響

V8.2

高解像度 (たとえば, 1920 × 1440 や 2048 × 1536), 高リフレッシュ・レート, 高負荷の状況では, 使用している RADEON カードやモニタによっては, 画像に影響が出ることがあります。このような状況が発生したら, リフレッシュ・レートを下げてみてください。

6.9.5 OpenGL は IEEE 演算のみをサポート

V8.2

OpenGL ライブラリは, IEEE 浮動小数点形式のみをサポートしています。VAX 浮動小数点形式はサポートしていません。アプリケーションのコンパイルでは, /FLOAT=IEEE_FLOAT オプションを使用してください。

6.9.6 グラフィック・コンソール (OPA0) に出力する際に DECwindows サーバがハングアップ

V8.2

DECwindows サーバの起動後に, グラフィック・コンソール (OPA0) に出力がある場合, DECwindows サーバはハングアップし, 画面はフリーズします。CTRL/F2 を押すと DECwindows サーバは実行状態に戻ります。

通常, コンソールに表示されるメッセージは, 直接 OPA0 に書き込むのではなく, OPCOM と Console Window アプリケーションを使用してメッセージ表示することをお勧めします。このアプリケーションを有効にするには, コマンド・プロシージャ SYSSMANAGER:DECW\$PRIVATE_APPS_SETUP.COM を編集し, 次のグローバル・シンボル定義を追加します。

```
$ DECW$CONSOLE_SELECTION == "WINDOW"
```

SYSSMANAGER:DECW\$PRIVATE_APPS_SETUP.COM が存在しない場合は, SYSSMANAGER:DECW\$PRIVATE_APPS_SETUP.TEMPLATE から作成することができます。

SYSSMANAGER:DECW\$PRIVATE_APPS_SETUP.COM の編集後, 次のコマンドを入力して DECwindows を再起動します。

```
$ @SYSSMANAGER:DECW$STARTUP RESTART
```

6.9.7 モニタは初期化時に接続されていなければならない

RADEON 7500 グラフィック・カードは, ビデオ出力ポートを 2 つ (デジタルとアナログ) 備えています。デジタル・インタフェースは, 現在サポートされていません。ただし, デジタル・ツー・アナログ・アダプタを使用すると, アナログ・モニタをデジタル・ポートに接続でき, アナログ・ポートと同じ出力を得ることができます。デ

デジタル・ポートを使用する場合、ポートを正しく動作させるためには、システムの電源投入より前にモニタが接続されていなければなりません。

6.9.8 ブート・リセットの推奨 (Alpha のみ)

コンソール変数 `boot_reset` に ON を設定することをお勧めします。

6.9.9 オーバレイ・プレーンはサポートされない

ハードウェア・オーバレイ・プレーンはサポートされていません。

6.9.10 単一カラーマップのみサポート

RADEON 7500 グラフィックス・コントローラがサポートするハードウェア・カラーマップは、1 つだけです。デフォルトのビジュアルが PseudoColor の場合、8 ビット・カラーに変更するときは、このことに留意してください。複数の PseudoColor カラーマップを一度に使用しようとすると、カラーマップがフラッシュします。

注意

3D (OpenGL) レンダリングは、8 ビットの PseudoColor ビジュアルではサポートされていません。(第 6.9.16 項も参照してください。)

アプリケーションは、自分でカラーマップのインストールやデインストールを行ってはいけません。これらの動作は、ウィンドウ・マネージャが行います。ただし、アプリケーションは、どのカラーマップをインストールまたはデインストールするかについてのヒントをウィンドウ・マネージャに知らせなければなりません。この情報は、Xlib 関数の `XsetWMColormapWindows()` を使用して渡します。この関数は、指定されたウィンドウの `WM_COLORMAP_WINDOWS` プロパティを設定します。

6.9.11 すべてのウィンドウで同じビット深度

RADEON 7500 カードを使用している場合、特定のヘッド上のすべてのウィンドウは、同じビット深度でなければなりません。RADEON 7500 カードは、それぞれのグラフィックス・ヘッドで、8、16、および 24 ビット/ピクセルのビット深度をサポートしています。しかし、DECwindows サーバが特定のヘッド上で一旦ビット深度を確立すると、そのビット深度のウィンドウやビジュアルだけが作成可能となります。

6.9.12 読み取り/書き込みカラー・マップのピクセル深度

デフォルトでは、RADEON 7500 には、読み取り専用の TrueColor カラー・マップが 1 つある 24 プレーンのピクセル深度が用意されています。DECwindows Paint などの一部のアプリケーションでは、読み取り/書き込みのカラー・マップが必要です。Paint は、読み取り専用カラー・マップで実行されると、次のエラー・メッセージを出力して失敗します。

```
Error: Visual Not Supported  
Supported Visuals are {PseudoColor, GrayScale, StaticGray}
```

読み取り/書き込みのカラー・マップを使用するには
SYSSMANAGER:DECW\$PRIVATE_SERVER_SETUP.COM ファイル (このファイルが存在しない場合は、SYSSMANAGER:DECW\$PRIVATE_SERVER_SETUP.TEMPLATE から作成します) を編集して、次の論理名定義を追加します。

```
$ DEFINE/EXECUTIVE/SYSTEM/NOLOG DECW$SERVER_PIXEL_DEPTH 8,8,8,8,8,8,8,8
```

そして、次のコマンドを使用して DECwindows を再起動します。

```
$ @SYSSMANAGER:DECW$STARTUP RESTART
```

この変更により、(マルチ・カード構成が可能な、最大 8 枚のグラフィック・カードに) ピクセル深度として 8 プレーンが設定され、サーバが PseudoColor ビジュアルを提供できるようになります。

6.9.13 backing store と save unders は 3D ウィンドウをサポートしていない

RADEON 7500 X11 サーバの backing store と save unders の実装では、3D ウィンドウをサポートしていません。

6.9.14 スレッドの制限事項

OpenVMS 用 RADEON 7500 OpenGL ライブラリは、スレッド・セーフではありません。ただし、OpenGL の使用がプログラム内の単一のスレッドに限定されている場合は、OpenGL をマルチスレッド・プログラムで 사용할 수 있습니다。

6.9.15 シングル・バッファ・ビジュアルがサポートされていない

RADEON 7500 DECwindows サーバは、ダブル・バッファ・ビジュアルだけをサポートしています。シングル・バッファリングを行うには、ダブル・バッファ・ビジュアルを選択して、アプリケーション内で `glDrawBuffer(GL_FRONT)` を呼び出さなければなりません。

6.9.16 カラー・インデックス・モードでの 3D がサポートされていない

DECwindows サーバが 8 ビット深度で起動された場合、8 ビット・ビジュアルが 2D 操作でサポートされますが、OpenGL レンダリングは、8 ビット・ビジュアルではサポートされません。

6.9.17 タイマ・メカニズム

環境によっては、プロセスがハードウェア・ロックを所有している間に、そのプロセスに割り込みが発生することがあります。その結果、DECwindows サーバがハングアップしているように見えることがあります。

このような状況を検出し、一時停止中のプロセス内でイメージを強制的に一時停止から抜け出させるか、場合によってはプロセスを削除してこのような状況から回復させるために、タイマ・メカニズムが実装されました。

タイマ・メカニズムは、次の 2 つの論理名を使用して構成できます。これらの論理名は、DECW\$PRIVATE_SERVER_SETUP.COM ファイルで指定しなければなりません。

- DECW\$DRM_TIMER_PERIOD (デフォルト: 5.0 秒)
クロック・ティックの間隔を秒数で指定します。浮動小数点数です。
- DECW\$DRM_TIMEOUT (デフォルト: 6)
タイムアウトが発生して、DECwindows サーバが RADEON カードの制御を奪うまでに待つクロック・ティック数を指定します。

デフォルト値は、グラフィック・アプリケーションの性能にタイマが与える影響を最小限にするように選択されています。DECwindows サーバが応答を再開するまでの時間を短くしたい場合は、DECW\$DRM_TIMER_PERIOD の値を小さくします。

次のいずれかの状況では、ハードウェア・ロックの保持中に、プロセスに割り込みが発生することがあります。

- プロセスは、別システム上に表示された端末を使用して、リモートでログインされている。
- プロセスは、通常の終了処理が実行されない状態で、他のプロセスによって一時停止または終了させられたサブプロセスである。

どちらの状況も発生しないと思われる構成では、時間間隔にゼロを設定して、タイマ・メカニズムを完全に無効にすることができます。

```
$ DEFINE/SYSTEM DECW$DRM_TIMER_PERIOD 0
```


ECW\$DRM_TIMER_PERIOD の値を変更するたびに、DECwindows サーバを再起動するか、システムをリブートして、変更内容を有効にしなければなりません。DECwindows サーバを再起動するには、次のコマンドを使用します。

```
$ @SYS$STARTUP:DECW$STARTUP RESTART
```

6.10 DECwindows X11 ディスプレイ・サーバ (Alpha のみ)

ここでは、OpenVMS Alpha システムに対応した DECwindows X11 ディスプレイ・サーバに関する注意事項について説明します。

6.10.1 S3 マルチヘッド・グラフィック

永続的な制限事項

S3 Trio32 または Trio64 グラフィック・カードを装備している Alpha コンピュータでは、シングル・スクリーン・ディスプレイだけがサポートされます。マルチヘッド・グラフィックはサポートされません。

6.10.2 OpenVMS V8.3 の DECwindows では複数の USB キーボードとマウスがサポートされない

V8.3

DECwindows は、1 つのマウスとキーボードからの入力だけをサポートします。システムのブート時に他のキーボードやマウスが接続されていると、KBD0 および MOU0 として構成されているデバイスだけが機能し、その他のデバイスは動作しません。DECwindows の起動後に他のキーボードやマウスを接続すると、システムがクラッシュする可能性があります。システム・クラッシュは、本リリースの後に提供される ECO パッチで修正される予定です。

6.11 DIGITAL Modular Computing Components (DMCC)

ここでは、DMCC に関する注意事項について説明します。

6.11.1 Alpha 5/366 および 5/433 PICMG SBC の制限事項

永続的な制限事項

KZPDA SCSI コントローラおよび PBXGA グラフィック・カードは、DIGITAL Modular Computing Components (DMCC) Alpha 5/366 および 5/433 PICMG SBC のブリッジの後ろにあるスロットに挿入することはできません。

6.11.2 SRM コンソールの更新

永続的な制限事項

Alpha 4/233 (21064a) , 4/266 (21164a) , 5/366 , 5/433 DMCC システムで SRM コンソールを更新するには、SRM コンソールまたは AlphaBIOS セットアップを選択しなければなりません。格納できるコンソールは 1 つだけです。

- これらのシステムで OpenVMS を実行している場合は、SRM コンソールだけを更新する。
- これらのシステムで Windows NT を実行している場合は、AlphaBIOS セットアップだけを更新する。

SRM と AlphaBIOS コンソールの両方を誤って更新すると、AlphaBIOS セットアップ・メニューが表示されます。この後、SRM コンソールに戻るオプションはありません。AlphaBIOS セットアップ・メニューを終了し、SRM コンソールに戻るには、次のインターネット・サイトにある Firmware Update ユーティリティを使用しなければなりません。

<ftp://ftp.digital.com/pub/Digital/Alpha/firmware/index.html>

6.12 Digital Personal Workstation: OpenVMS V7.3-1 およびそれ以降のブート

V7.3-1

Digital Personal Workstation 433au , 500au , および 600au シリーズのシステムを使用するときは、コントローラ・チップが Cypress PCI Peripheral Controller の場合、IDE CD から OpenVMS Version 7.3-1 以降をブートできます。Intel Saturn I/O (SIO) 82378 チップが実装されている Digital Personal Workstation au シリーズ・システムでは、IDE CD から OpenVMS をブートすることはできません。Intel SIO チップが実装されている場合は、SCSI CD を使用しなければなりません。

どちらの IDE チップが実装されているか調べるには、次の SRM コンソール・コマンドを入力します。

SHOW CONFIGURATION

「Cypress PCI Peripheral Controller」と表示された場合は、OpenVMS をブートできます。

「Intel SIO 82378」と表示された場合は、SCSI CD を使用して CD からブートする必要があります。

6.13 I/O 負荷が重い場合，LUN が多数あるデュアル・コントローラ HSGnnに障害が発生することがある

V7.3-2

ドライバの性能が改善されたことと，システムの高速化によって，比較的多数の LUN が構成されているデュアル・コントローラ HSGnnストレージ・アレイで扱うことができる I/O 量の制限が取り払われました。この制限値に達した場合，アレイは，コントローラ間の通常の定期的な同期処理を完了できないほど，I/O 処理がビジーになることがあります。これにより，コントローラのハングアップまたは障害が発生し，手作業でコントローラをリセットするまで，一部またはすべての LUN がホストからアクセスできなくなることがあります。このようなコントローラ障害が発生したときには，HSG に報告される Last Failure Code は，多くの場合 01960186，01942088，および 018600A0 になります。

大半の HSGnnデバイスは，問題なく動作し続けます。負荷が重く，約 24 個以上の LUN が HSG に構成されていて，HSG コントローラのハングアップや障害が発生するサイトでは，コントローラを再構成して，LUN の数を少なくするか，HSG の負荷が重くならないように I/O を分散させると，ハングアップや障害を防止できる可能性があります。

この問題は，弊社のエンジニア・グループで調査を行っています。

6.14 Open3D グラフィックのライセンス方式の変更

V8.2

これまで，3D グラフィック表示機能は，OpenVMS オペレーティング・システムとは別にライセンスしていました。Open3D レイヤード・プロダクトは，最初の提供時から，オペレーティング・システムとは別のライセンスが必要でした。Open3D ソフトウェアを OpenVMS オペレーティング・システムの一部として出荷を開始したときも，3D グラフィック表示機能は，別のライセンス機能とされました。このようなライセンス方式の例としては，3X-PBXGG-AA ATI RADEON 7500 PCI 2D/3D グラフィック・アダプタによるグラフィック表示機能をサポートするための Open3D ライセンスがあります。

OpenVMS Version 8.2 からは，AlphaServers と Integrity サーバのいずれについても，3D グラフィック表示機能はオペレーティング・システムとともにライセンスされます。したがって，OpenVMS Version 8.2 には Open3D の個別のライセンスはありません。以前のバージョンの OpenVMS で 3D 表示操作を行うためには，Open3D ライセンスをシステムにインストールする必要があります。

弊社は、標準契約または Mature Product Support の OpenVMS Version 7.3-2 とともに出荷された 3D デバイス・ドライバについては、サポート契約内容に従ってサポートを継続します。以下のアダプタのデバイス・ドライバは、OpenVMS Version 7.3-2 とともに出荷されています。

- PowerStorm 300 および 350 PCI グラフィック・アダプタ (SN-PBXGD)
- ATI RADEON 7500 PCI および AGP グラフィック・アダプタ (3X-PBXGG)

これらのアダプタは OpenVMS Version 8.2 でも 3D グラフィック表示が可能です。が、ライセンスは不要です。さらに、OpenVMS Version 8.2 では、以下の 2D グラフィック・アダプタもサポートを継続します。

- ELSA Gloria Synergy (SN-PBXGK)
- 3Dlabs Oxygen VX1 (SN-PBXGF)

ATI RADEON 7500 PCI グラフィック・アダプタは、OpenVMS I64 Version 8.2 でも近日中にサポートされる予定です。現在はテストを実施中です。このグラフィック・カードがサポートされるようになった時点で、次の Web サイトでお知らせします。

<http://h71000.www7.hp.com/new/>

6.15 OpenVMS 用の PowerStorm 300/350 PCI グラフィック・サポート

V8.2

OpenVMS Alpha を実行している Compaq Workstation での PowerStorm 300 /350 PCI グラフィック・コントローラのサポートに関する注意事項については、『PowerStorm 300/350 OpenVMS Graphics Release Notes Version 2.0』を参照してください。リリース・ノート、およびインストール・ガイドは、グラフィック・カードとともに発送されます。

Open3D のライセンスはチェックされない

OpenVMS Version 8.2 以降では、3D (OpenGL) サポートのライセンスは、OpenVMS のライセンスの一部として含まれています。詳細は、第 6.14 節を参照してください。

論理名 DECW\$OPENGL_PROTOCOL の定義

3D グラフィック・アプリケーションを実行して、PowerStorm 350/300 グラフィック・カードを備えたシステムに出力を表示する場合、アプリケーションを実行しているシステム上で、論理名 DECW\$OPENGL_PROTOCOL が次のように定義されていることを確認してください。

```
$ DEFINE DECW$OPENGL_PROTOCOL DECW$OPENGL_PROTOCOL_V11
```

問題の修正

P350 はセッションの終了時に、再初期化に失敗することがありました。

この問題は以下の 2 点の変更で修正されました。

- デバイス固有の riCloseScreen 関数に vmsCloseScreen の呼び出しを追加しました。riCloseScreen 関数は、CDE セッションの終了時などに呼び出されます。vmsCloseScreen は GB デバイスへのチャネルの割り当てを解除し、ドライバがボードを正しく再初期化できるようにします。
- デバイス・ドライバのピクセル・コンバータ同期アルゴリズムが大きく改善されました。

6.16 RFnn DSSI ディスク・デバイスとコントローラ・メモリのエラ

V6.2

RF31T, RF31T+, RF35, RF35+, RF73, RF74 DSSI ディスク・デバイスの以前のバージョンのマイクロコードには問題があります。この問題が原因で、データが失われる可能性があり、これらのデバイスからデータを読み込むときに、デバイスでコントローラ・メモリ・エラー (エラー検出/訂正 (EDC) エラーとも呼ばれる) が発生していた場合、問題が発生することがあります。このエラーは仮想サーキットの閉鎖やハードウェアの障害が原因で発生している可能性があります。

これらのデバイスを使用する場合は、マイクロコードのリビジョン・レベルを確認してください。マイクロコードのリビジョン・レベルが表 6-2 に示されている値より低い場合は、マイクロコードを更新してください。

RF31T, RF31T+, RF35+ 以外のすべてのモデルのマイクロコードは、最新の OpenVMS バイナリ・ディストリビューション CD にあります。

DSSI ディスク・デバイスのマイクロコード・リビジョン・レベルを表示するユーティリティ・プログラムである RF_VERS ユーティリティも同じ CD に格納されています。このユーティリティ・プログラムの使用法とマイクロコードの更新方法については、ここで説明します。

注意

RF31T, RF31T+, RF35+ ディスク・ドライブを使用し、マイクロコードのバージョンがサポートされないバージョンであり (表 6-2 を参照), サポート契約を結んでいるお客様の場合には、弊社のサポート担当者にお問い合わせください。サポート契約を結んでいないお客様の場合には、公認代理店にお問い合わせください。

DSSI ディスクのマイクロコードのうち、表 6-2 に示したリビジョン・レベル以上がサポートされます。

表 6-2 サポートされるマイクロコードのレベル

デバイス・タイプ	サポートされるマイクロコードの最小リビジョン・レベル
RF31T	T387E
RF31T+	T387E
RF35	T392D
RF35+	T392D
RF36	V427P
RF73	T392D
RF74	V427P

DSSI ディスク・デバイスのマイクロコード・リビジョン・レベルを表示するには、次の手順を実行します。

1. SYSTEM アカウントにログインするか、または CMKRNL 特権、DIAGNOSE 特権、SYSPRV 特権がある他のアカウントにログインします。
2. 次のコマンドを入力します。

```
$ SET PROCESS /PRIVILEGE=(DIAGNOSE,CMKRNL,SYSPRV)
$ SHOW DEVICE FYA0:
```

VAX システムで SHOW DEVICE コマンドを実行してエラーが発生した場合には、次のコマンドを入力します。

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> CONN FYA0/NOADAP
SYSGEN> ^Z
```

Alpha システムで SHOW DEVICE コマンドを実行してエラーが発生した場合には、次のコマンドを入力します。

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> IO CONNECT FYA0: /NOADAP
SYSGEN> ^Z
```

次の例に RF_VERS コーティリティが出力する内容を示します。

```
Program Name:   RF_VERS
Revision Level: V1.2s

NOTICE: This program does not currently support the RF72 or any
        HSDxx controllers. See next version for support.

DSSI disks currently on this system as seen by RF_VERS

Device      Node      Status      Hardware      Firmware
Name        Name                               Type          Version
```

6.16 RFnn DSSI ディスク・デバイスとコントローラ・メモリのエラー

_ \$22\$DIA7:	R4JL2I	mounted	RF73	T387A
_ \$22\$DIA6:	R4I0BG	mounted	RF73	T387A
_ \$22\$DIA8:	R4XLWE	mounted	RF73	T387A
_ \$22\$DIA2:	R4FCZK	mounted	RF73	T387A
_ \$22\$DIA3:	R4CKCG	mounted	RF73	T387A
_ \$22\$DIA4:	R4ZKUE	mounted	RF73	T387A
_ \$22\$DIA9:	R4GYII	mounted	RF73	T387A
_ \$22\$DIA1:	R4XRYI	mounted	RF73	T387A

デバイスのマイクロコードを更新するには、表 6-3 でデバイスとプラットフォームに対応するコマンドを確認し、使用します。

重要

マイクロコードを更新する場合は、あらかじめディスクのバックアップを作成してください。

表 6-3 DSSI ディスク・デバイスのマイクロコードを更新するコマンド

デバイス・タイプ	プラットフォーム	コマンド
RF35	Alpha	\$RUN SYS\$ETC:RF35_T392F_DEC_ALPHA.EXE
RF35	VAX	\$RUN SYS\$ETC:RF35_T392F_DEC.EXE
RF36	Alpha	\$RUN SYS\$ETC:RF36_V427P_DEC_ALPHA.EXE
RF36	VAX	\$RUN SYS\$ETC:RF36_V427P_DEC.EXE
RF73	Alpha	\$RUN SYS\$ETC:RF73_T392F_DEC_ALPHA.EXE
RF73	VAX	\$RUN SYS\$ETC:RF73_T392F_DEC.EXE
RF74	Alpha	\$RUN SYS\$ETC:RF74_V427P_DEC_ALPHA.EXE
RF74	VAX	\$RUN SYS\$ETC:RF74_V427P_DEC.EXE

重要

SCSI_INFO.EXE、RF_VERS.EXE をはじめ、表 6-3 に示されているファイルは絶対に削除しないでください。これらのファイルを削除すると、VAX システムでは VMSKITBLD.COM がファイルを検索することができなくなります。同様に、Alpha システムでは AXPVMS\$PCSI_INSTALL と AXPVMS\$PCSI_INSTALL_MIN での PRODUCT INSTALL コマンドが失敗します。

6.17 RZnn ディスク・ドライブの考慮事項

ここでは、各種の RZ ディスク・ドライブに関する注意事項について説明します。

6.17.1 RZ25M と RZ26N ディスク・ドライブ: 推奨事項

V7.1

DWZZA とロング・ディファレンシャル SCSI バスを含む構成を使用して、弊社がサポートする SCSI ディスク・ドライブをテストしたところ、2 台のドライブ (RZ25M と RZ26N) でバス・フェーズに関する問題が検出されました。そのため、DWZZA を接続するディファレンシャル・バスの長さが 20 メートル以上の構成では、これらのドライブを使用しないでください。

この勧告は RZ25M および RZ26N ドライブにのみ適用されます。OpenVMS SPD に、サポートされるドライブとしてリストされている他のすべてのディスク・ドライブは、SCSI-2 仕様の上限のバスの長さまで使用できます。

6.17.2 RZ26N および RZ28M ディスク: 推奨ファームウェア・サポート

V6.2-1H3

RZ26N および RZ28M ディスクを使用する場合には、ファームウェアのリビジョン・レベルは 0568 以上をお勧めします。

これらのディスクで最新のファームウェア・リビジョン・レベルが使用されていない場合には、問題が発生する可能性があります。

6.17.3 RZ26L および RZ28 ディスク: マルチホストで使用するために必要なファームウェア

V6.2

OpenVMS Cluster のマルチホスト SCSI バスに RZ26L または RZ28 ディスクを取り付ける場合、ディスクに必要なファームウェア・リビジョンは、最低でも 442 です。

ここでは、一部の RZ26L および RZ28 ドライブでファームウェアを更新するために使用する手順について説明します。この手順を使用できるのは、ドライブがホスト・システムの SCSI アダプタに直接接続されている場合だけです。インテリジェント・コントローラ (HSZ40 や KZPSC など) を介して接続されているドライブは、この手順では更新できません。ファームウェアの別の更新手順があるかどうかについては、インテリジェント・コントローラのドキュメントを参照してください。

重要

ファームウェア・リビジョン・レベル 442 に安全にアップグレードできるのは、特定の RZ26L および RZ28 ファームウェア・リビジョンだけです。使用中のディスクをファームウェア・リビジョン・レベル 442 にアップグレードできるかどうか判断するには、第 6.17.3.1 項を参照してください。ディスクがファームウェア・リビジョン・レベル 442 をサポートできる場合は、第

6.17.3.2 項で説明している RZTOOLS ユーティリティを使用して、ディスクのファームウェアを更新します。

6.17.3.1 ファームウェア・リビジョン・レベル 442 の必要条件

ファームウェア・リビジョン・レベル 442 に安全にアップグレードできるのは、表 6-4 に示したディスク・ドライブとファームウェア・リビジョン・レベルの組み合わせだけです。他の組み合わせで更新手順を実行すると、ディスクを永久に破損する可能性があります。

表 6-4 リビジョン・レベル 442 ファームウェアの互換性

ディスク・ドライブ	ファームウェア・リビジョン	ディスク・ファイル名
RZ26L	440C	RZ26L_442D_DEC.FUP
RZ28	441C または D41C 435 または 436	RZ28_442D_DEC2104.FUP RZ28P4_442C_DEC.FUP

6.17.3.2 ファームウェア・リビジョン・レベル 442 のインストール手順

使用しているディスクでリビジョン・レベル 442 ファームウェアが必要かどうか、安全にアップグレードできるかどうかを判断した後、次の手順を実行してファームウェアを更新します (アップグレードするディスクのファイル名については、表 6-4 を参照してください)。

```
$ RZTOOLS_ALPHA := $SYS$ETC:RZTOOLS_ALPHA
$ RZTOOLS_ALPHA DKB500 /LOAD=$SYS$ETC:filename.FUP
Read in 262144 bytes.
Current FW version - X440C
Upgrading to      - DEC0
Loading code .....
New code has been sent to the drive.
```

6.18 sx1000 Integrity Superdome

V8.3

HP Integrity Superdome は、Core I/O LAN カードを通してサテライトとしてブートできません。この LAN カードを BOOT_OPTION.COM のブート・オプションとして指定し、オペレーティング・システムをシャットダウンすると、LAN カードが EFI に現れません。この問題は、ファームウェアの将来のリリリースで修正される予定です。

6.19 ZLX グラフィック・ボードのサポート

V8.2

次のグラフィック・コントローラ・ボード・ファミリは、OpenVMS Version 8.2 ではサポートされていません。

- ZLX-M シリーズ (PixelVision): ZLX-M1 (PMAGC-AA) , ZLX-M2 (PMAGC-BA)
- ZLX-L シリーズ (PixelVision Lite): ZLX-L1 (PMAGC-DA) , ZLX-L2 (PMAGC-EA)
- ZLXp-L シリーズ (PixelVision PCI): ZLXp-L1 (PBXGC-A) , ZLXp-L2 (PBXGC-B)

OpenVMS Version 8.2 では、以下のグラフィック・コントローラ・ボード・ファミリ用に、OpenVMS と一緒に出荷される基本的な 2D 機能を使用した 2D サポートのみが提供されます。以下のボード用に 2D サポートを利用する目的で Open3D をインストールしないでください。

- ZLX-E シリーズ (FFB): ZLX-E1 (PMAGD-AA) , ZLX-E2 (PMAGD-BA) , ZLX-E3 (PMAGD-CA)
- ZLXp-E シリーズ (TGA): ZLXp-E1 (PBXGA-A) , ZLXp-E2 (PBXGA-B) , ZLXp-E3 (PBXGA-C)
- ZLX2-E シリーズ (TGA2): PowerStorm 3D30 (PBXGB-AA) , PowerStorm 4D20 (PBXGB-CA)

第 A.1 節の関連する注意事項も参照してください。

6.20 OpenVMS デバイス・ドライバの再コンパイルと再リンク

ここでは、OpenVMS デバイス・ドライバの再コンパイルと再リンクに関する注意事項について説明します。

第 5.4 節の関連する注意事項も参照してください。

6.20.1 Alpha および VAX の SCSI デバイス・ドライバ

V7.3-1

OpenVMS の以前のバージョンのすべての OpenVMS Alpha SCSI デバイス・ドライバが OpenVMS Version 7.3-1 以降で正しく動作するには、再コンパイルと再リンクが必要です。

OpenVMS Alpha Version 7.0 より前のバージョンからアップグレードしている OpenVMS Alpha SCSI ドライバがある場合は、第 6.20.2 項を参照してください。

OpenVMS Version 7.1 では、すべての OpenVMS VAX SCSI デバイス・ドライバの再コンパイルと再リンクが必要でした。OpenVMS Version 7.1 で実行できるように再コンパイルと再リンクされた OpenVMS VAX デバイス・ドライバは、OpenVMS Version 7.3 以降でも正しく動作します。

6.20.2 OpenVMS Alpha デバイス・ドライバ

V7.1

OpenVMS Alpha Version 7.0 で実行できるように再コンパイルおよび再リンクされたデバイス・ドライバは、OpenVMS Alpha Version 7.1 以降で実行できるようにするためにソース・コードを変更したり、再コンパイルや再リンクしたりする必要がありません (ただし、Alpha SCSI ドライバについては、再コンパイルと再リンクが必要です。第 6.20.1 項を参照してください)。

OpenVMS Alpha Version 7.0 より前のリリースのデバイス・ドライバのうち、OpenVMS Alpha Version 7.0 に対応するよう再コンパイルおよび再リンクされていないデバイス・ドライバを OpenVMS Alpha Version 7.1 以降で実行するには、再コンパイルと再リンクが必要です。

OpenVMS Alpha Version 7.0 では、OpenVMS Alpha 特権インタフェースと構造体が大幅に変更されました。これらの変更の結果、OpenVMS Alpha Version 7.0 より前のリリースのデバイス・ドライバでは、OpenVMS Alpha Version 7.0 以降で正しく動作するように、ソース・コードを変更する必要があります。カスタマが作成したドライバのソースの変更が必要となる OpenVMS Alpha Version 7.0 の変更点の詳細については、『HP OpenVMS Guide to Upgrading Privileged-Code Applications』を参照してください。

6.21 MON バージョンのデバイス・ドライバの処理

V7.3

OpenVMS Version 7.3 では、SYSTEM_CHECK を有効化すると、SYS\$nnDRIVER_MON.EXE という形式の名前を持つデバイス・ドライバ・イメージがシステム・ローダによって自動的に読み込まれます。対応する MON バージョンが存在しない場合は、デフォルトのイメージ名 SYS\$nnDRIVER.EXE が使用されます。

6.22 スレッド単位のセキュリティが Alpha デバイス・ドライバに与える影響

V7.2

スレッド単位のセキュリティが OpenVMS Alpha デバイス・ドライバに与える影響については、第 5.4.7 項を参照してください。

6.23 OpenVMS Alpha ドライバのデバイス IPL の設定

V6.2

PCI, EISA, ISA バスをサポートする Alpha ハードウェア・プラットフォームでは、20 または 21 という異なる IPL で I/O デバイスへの割り込みが発生します。デバイスへの割り込みが発生する IPL は、デバイスをプラットフォーム間で移動したときに変わる可能性があります。ドライバがデバイス IPL を 20 であると宣言した後、I/O デバイスへの割り込みが IPL 21 で発生するマシンでそのドライバを実行すると、問題が発生します。

この問題に対する最も簡単な対処法は、PCI, EISA, ISA のデバイス・ドライバで IPL 21 を使用することです。この方法は、I/O デバイスへの割り込みが IPL 20 で発生するプラットフォームでも、I/O デバイスへの割り込みが IPL 21 で発生するプラットフォームでも、正しく動作します。

6.24 CRCTX ルーチンの機能の強化

V7.1-2

Counted Resource Context Block (CRCTX) 構造体の管理に使用できるシステム・ルーチンが強化されました。次のルーチンが、CRCTX 構造体のステータス (CRCTX\$V_ITEM_VALID) を設定およびチェックするようになりました。

- IOC\$DEALLOC_CRCTX
- IOC\$ALLOC_CNT_RES
- IOC\$DEALLOC_CNT_RES
- IOC\$LOAD_MAP

これらのルーチンは次のように変更されました。

有効な CRCTX ステータス (CRCTX\$V_ITEM_VALID を 1 に設定) で IOC\$DEALLOC_CRCTX を呼び出すと、サービスは不正なステータスを返します。SYSBOOT パラメータ SYSTEM_CHECK が設定されている場合には、システム障害が発生します。このため、割り当てが解除されていない有効なリソースがあるときに、ユーザが CRCTX の割り当てを誤って解除することを防止できます。

IOCSALLOC_CNT_RES は、無効な CRCTX ステータス (CRCTXSV_ITEM_VALID を 0 に設定) で呼び出さなければなりません。有効なステータスでこのルーチンと呼び出すと、OpenVMS はこの CRCTX によってマップされたリソースをユーザが手放すものと解釈します。OpenVMS は新しいリソースを割り当てず、不正なステータスを返します。SYSTEM_CHECK が設定されている場合には、システム障害が発生します。IOCSALLOC_CNT_RES は有効ビットを設定してから戻ります。

IOCSDEALLOC_CNT_RES は、有効な CRCTX ステータス (CRCTXSV_ITEM_VALID を 1 に設定) で呼び出さなければなりません。無効な CRCTX で IOCSDEALLOC_CNT_RES を呼び出すと、OpenVMS は他のパラメータが有効でないものと解釈し、不正ステータスを返します。SYSTEM_CHECK がセットされている場合には、システム障害が発生します。IOCSDEALLOC_CNT_RES は有効ビットをクリアしてから戻ります。

IOCSLOAD_MAP は有効な CRCTX で呼び出さなければなりません。無効な CRCTX (CRCTXSV_ITEM_VALID を 0 に設定) で呼び出すと、他のパラメータも無効であると解釈され、不正ステータスが返されます。SYSBOOT パラメータ SYSTEM_CHECK がセットされている場合には、システム障害が発生します。

これらの変更により、デバイス・サポート・アプリケーションや特権付きコード・アプリケーションの開発者は、OpenVMS で汎用リソースとして取り扱われる scatter gather レジスタの割り当てを解除する必要があるかどうか判断できます。CRCTXSV_ITEM_VALID ビットがセットされている場合は、IOCSDEALLOC_CNT_RES を呼び出さなければなりません。

6.25 アダプタについての注意事項

V8.2-1

ここでは OpenVMS Version 8.2-1 でサポートされるアダプタについての注意事項を説明します。

6.25.1 Fibre Channel の EFI ドライバとファームウェアの要件

OpenVMS I64 Version 8.3 for Integrity サーバでは、HP A6826A 2 GB Fibre Channel ホストベース・アダプタとその改良版を使用するためには、EFI ドライバのバージョン 1.47 以降、RISC ファームウェアのバージョン 3.03.154 以降が必要です。また、HP AB378A および AB379A 4 GB Fibre Channel ホストベース・アダプタを使用するためには、EFI ドライバのバージョン 1.05、RISC ファームウェアのバージョン 4.00.70 以降が必要です。

RISC ファームウェアと EFI ドライバについて、現在サポートされている最新のバージョンを調べるには、HP IPF Offline Diagnostics and Utilities CD に入っている README テキスト・ファイルを参照してください。このファイルを見つけるには、

2 GB Fibre Channel HBA では `\efi\hp\tools\io_cards\fc2` ディレクトリに、4 GB HBA では `\efi\hp\tools\io_cards\fc4` ディレクトリに移動してください。ドライバとファームウェアをアップデートするには、HBA の種類に応じて `fc2_update2.nsh` または `fc4_update4.nsh` を実行してください。Offline Diagnostics and Utilities CD の入手方法については、『HP OpenVMS V8.3 インストレーション・ガイド[翻訳版]』を参照してください。

6.25.2 複数の Fibre Channel ブート・エントリを使用したブート

セル・ベース・システムと、新しいエントリ・レベル・システムでは、最初の Fibre Channel ブート・エントリ・リストが唯一有効なブート・エントリです。他の Fibre Channel I64 システム・ディスクからブートするには、EFI Shell を開き、`"search all"` を実行し、EFI Shell を終了した後、指定のブート・エントリを選択します。この操作は、複数のメンバにシャドウイングされたシステム・ディスクからブートする場合にも必要です。

リタイア製品情報

この付録では、すでにサポートが中止された OpenVMS 製品およびサポートの中止が予定されている OpenVMS 製品についてお知らせします。また、アーカイブ扱いになったマニュアルを検索する方法についても説明します。

フリーウェア

製品のサポートが中止されると、弊社はそれらの製品に対する問題に関する報告を受け付けず、またそのような報告への対処も行いません。しかし、独自の開発やサポートを目的に、以前の製品のソース・コードを必要とされるお客様に対しては、契約上その他の状況で受け入れ可能な場合、多くの旧製品のインストール・キットやソース・コードを OpenVMS Freeware として次の形態で提供しています。

- OpenVMS オペレーティング・システムに添付される Freeware の媒体
- 次のアドレスの Web サイト

<http://www.hp.com/go/openvms/freeware/>

A.1 Compaq Open3D レイヤー・プロダクトは Version 8.2 ではサポートされない

V8.2

OpenVMS Version 8.2 では、レイヤー・プロダクトの Compaq Open3D for OpenVMS Alpha V4.9B およびそれ以前のバージョンは、サポートされなくなりました。ただし、Mature Product Support (MPS) の下で OpenVMS Versions 6.2 および 7.3-2 でのサポートを継続します。この製品と、OpenVMS Version 8.2 に統合され、PowerStorm 300、PowerStorm 350、および ATI RADEON などの新しいカードをサポートする Open3D 製品とを混同しないようにしてください。サポートされなくなった製品は、OpenVMS Version 7.3-2 以降のリリースでの PixelVision、FFB、TGA、および TGA2 ベースのグラフィックの 3D サポートです。第 6.19 節の関連する注意事項を参照してください。

サポートされていない Open3D 製品がシステムにインストールされている場合は、問題が発生する可能性があります。たとえば、DECwindows ディスプレイ・サーバは、CPU を多用するループ内でハングアップすることがあります。使用中のバージョンの OpenVMS に Open3D がすでにインストールされている場合に、OpenVMS Version 8.2 へアップグレードする予定であれば、ターゲット・システムに次のグラフィック・コントローラ・ボードがインストールされていないことをまず確認しなければなりません。

- ZLX-M シリーズ (PixelVision): ZLX-M1 (PMAGC-AA) , ZLX-M2 (PMAGC-BA)
- ZLX-L シリーズ (PixelVision Lite): ZLX-L1 (PMAGC-DA) , ZLX-L2 (PMAGC-EA)
- ZLXp-L シリーズ (PixelVision PCI): ZLXp-L1 (PBXGC-A) , ZLXp-L2 (PBXGC-B)

A.2 Open3D グラフィックのライセンス方式の変更

V8.2

OpenVMS Version 8.2 から , 3D グラフィック表示機能のライセンスは , AlphaServer と Integrity サーバのいずれの場合も , オペレーティング・システムのライセンスに含まれます。したがって , OpenVMS Version 8.2 では Open3D の単独のライセンスは入手できません。

詳細については第 6.14 節を参照してください。

A.3 DECamsds は OpenVMS Version 8.2 ではサポートされない

V8.2

OpenVMS Version 8.2 では , DECamsds はサポートされません。弊社では , DECamsds の代わりに Availability Manager を使用することをお勧めします。Availability Manager についての情報は , 次の Web サイトを参照してください。

<http://h71000.www7.hp.com/openvms/products/availman/>

A.4 DECevent はサポートされない

V8.2

DECevent と DIAGNOSE コマンドは , OpenVMS Version 8.2 ではサポートされません。

DECevent の代替機能としては , 次のツールがあります。

- System Event Analyzer (SEA)

SEA は , OpenVMS とそれ以降のハードウェア・プラットフォーム (たとえば , すべての I64 プラットフォーム, および Alpha の DSnn , ESnn , および大半の GSnn プラットフォーム) でサポートされるエラー・ログ分析ツールになりました。

SEA を含め、最新のハードウェア・サポートを利用するには、入手できる最新の WEBES をインストールしてください。最低でも、Alpha システムでは WEBES Version 4.2 以上、I64 システムでは WEBES Version 4.4 以上が必要です。

SEA のオペレーティング・システム要件と、サポートされるハードウェアについての詳細は、WEBES のインストレーション・ガイドを参照してください。このドキュメントは、他の WEBES ドキュメントとともに、次の Web サイトにあります。

<http://h18000.www1.hp.com/support/svctools/>

- Error Log Viewer (ELV)

ELV を使用すると、新しいハードウェア・プラットフォーム上で作成されたエラー・ログ・ファイルを素早く調べることができます。ELV は OpenVMS オペレーティング・システムに統合されており、DCL コマンドの ANALYZE/ERROR_LOG/ELV を入力することでアクセスできます。

ELV の詳細は、『OpenVMS システム管理ユーティリティ・リファレンス・マニュアル』の ELV の章を参照してください。

A.5 2004 年 12 月に DECwrite のサービス期間が終了

V8.2

DECwrite は、2004 年 12 月 31 日でサービス期間が終了し、Software Product Library から削除されました。

A.6 Error Log Report Formatter (ERF) のサポート終了

V8.2

Error Log Report Formatter (ERF) はサポートされなくなりました。OpenVMS Version 7.2 より前のシステムで作成されたログを、この製品で処理する必要がある場合は、次の Freeware Web サイトから ERF ドキュメントにアクセスできます。

<http://www.hp.com/go/openvms/freeware/>

ERF は Error Log Viewer (ELV) に置き換わりました。詳細については、オンラインヘルプで ANALYZE/ERROR_LOG/ELV を参照するか、『OpenVMS システム管理ユーティリティ・リファレンス・マニュアル』を参照してください。

ERF を使用する前に、ELV の CONVERT コマンドまたは DECevent の一部である Binary Error Log Translation ユーティリティを使用してエラー・ログ・ファイルを変換する必要があります。DECevent と DIAGNOSE コマンドはサポートされなくなりましたが、これらのツールを必要とするユーザは、次の Freeware Web サイトから、ソフトウェアと関連ドキュメントをダウンロードすることができます。

<http://www.hp.com/go/openvms/freeware/>

DECevent ソフトウェアをインストールするまでは、DIAGNOSE コマンドを使用するとエラーが出力されます。

A.7 ISA_CONFIG.DAT のサポートの将来のリリースでの中止

V7.1

SYSSMANAGER:ISA_CONFIG.DAT ファイルを使用して ISA デバイスを構成する機能は、OpenVMS Alpha の将来のリリースではサポートされなくなります。このファイルを使用している場合は、コンソールから ISACFG ユーティリティと、デバイス・ドライバをロードするための新しいファイル・ベースの自動構成方式を使用するように変換しなければなりません (『Writing OpenVMS Alpha Device Drivers in C』を参照)。

A.8 POSIX 1003.4a Draft 4 インタフェースのサポート中止

V7.0

Compaq POSIX Threads Library (以前の DECthreads) の POSIX 1003.4a, Draft 4 ("d4") インタフェースは、将来のリリースでサポートされなくなる予定です。POSIX 1003.4a, Draft 4 インタフェースを使用して作成されたアプリケーションは、POSIX スレッド・ライブラリで提供される新しい POSIX 1003.1c 標準 ("pthread") インタフェースに移行する必要があります。このリリースでは、移行を支援するために、Draft 4 POSIX 1003.4a インタフェース用の互換モードが提供されます。この互換モードは、将来のリリースでは削除されます。

A.9 NetBeans Version 3.6 のサポートは OpenVMS Version 8.3 で終了

V8.3

OpenVMS Alpha Version 8.3 および OpenVMS I64 Version 8.3 は、NetBeans Version 3.6 for OpenVMS がサポートされる最後のリリースです。NetBeans Version 3.6 は、OpenVMS Version 8.3 のサポート期間が過ぎてもサポートされます。また、OpenVMS Alpha および OpenVMS I64 用の NetBeans Version 3.6 は、Java™ Platform, Standard Edition, Development Kit (JDK) v 1.4.2-x 上でのみサポートされる点に注意してください。

OpenVMS 上の GUI ベースの開発環境としては、Distributed NetBeans の使用を検討してください。これは、コスト効率が高く柔軟な開発環境ソリューションです。Distributed NetBeans についての詳細は、次の Web サイトを参照してください。

<http://www.hp.com/products/openvms/distributednetbeans/>

A.10 アーカイブ扱いのマニュアル

V8.3

製品の製造が中止され、オペレーティング・システムが進化するにつれて、OpenVMS マニュアルの一部はアーカイブ扱いになります。アーカイブ扱いのマニュアルはそれ以降保守されず、OpenVMS ドキュメンテーション・セットには含まれません。ただし、次の Web サイトに保管されています。

<http://www.hp.com/go/openvms/doc>

左のサイド・バーにある“Archived documents”をクリックすると、アーカイブ扱いのマニュアルを参照できます。

インターロックされたメモリ命令の使用 (Alpha のみ)

インターロックされたメモリ命令を使用するための厳密な規則は、『Alpha Architecture Reference Manual, Third Edition (AARM)』に説明されています。Alpha 21264 (EV6) プロセッサと将来のすべての Alpha プロセッサは、これらの規則で決められている必要条件に関して、以前のプロセッサよりさらに厳しくなっています。この結果、以前は規則に準拠していなくても正常に動作していたコードが、21264 以上のプロセッサを搭載したシステムでは正しく実行できないことがあります。このような規則に準拠していないコード・シーケンスが発生することは、非常にまれであると考えられています。21264 プロセッサは、OpenVMS Alpha Version 7.1-2 より前のバージョンではサポートされません。

規則に従っていないコードを実行すると、インタープロセッサ・ロックが使用されるときに、プロセッサ間の同期が失われる可能性があり、インターロックされたシーケンスが常にエラーになる場合は、無限ループになることがあります。BLISS コンパイラの以前のバージョン、MACRO-32 コンパイラと MACRO-64 アセンブラの一部のバージョンでコンパイルされたプログラムや、一部の HP C および C++ プログラムのコード・シーケンスで、このような動作が発生することがあります。

影響を受けるコード・シーケンスでは、LDx_L/STx_C 命令を、アセンブリ言語ソースで直接、またはコンパイラで生成されたコードで使用しています。インターロックされた命令を使用する可能性の高いアプリケーションは複雑であるか、マルチスレッドされたアプリケーションであるか、または高度に最適化された固有に作成したロックおよび同期化手法を使用しているデバイス・ドライバです。

B.1 必要なコード・チェック

OpenVMS では、21264 プロセッサで実行されるコードにこのようなシーケンスがないかどうか確認してください。プロセス間ロック、マルチスレッド、プロセッサ間通信を行うコードでは、特に注意する必要があります。

Alpha 実行可能プログラムを分析して、規則に準拠していないコード・シーケンスがあるかどうか調べるために、SRM_CHECK ツールが開発されました。このツールは、エラーが発生する可能性のあるシーケンスを検出し、エラーを報告し、問題のあるシーケンスのマシン・コードを表示します。

B.2 コード分析ツール (SRM_CHECK) の使用

SRM_CHECK ツールは、OpenVMS Alpha Version 7.3-2 Operating System CD の次の場所にあります。

```
SYS$SYSTEM:SRM_CHECK.EXE
```

SRM_CHECK ツールを実行するには、フォーリン・コマンドとして定義 (または DCL\$PATH 機能を使用) し、チェックするイメージの名前を指定して起動します。問題が検出されると、マシン・コードが表示され、イメージ情報の一部が印刷されます。次の例では、このツールを使用して myimage.exe というイメージを分析する方法を示しています。

```
$ define DCL$PATH []  
$ srm_check myimage.exe
```

このツールでは、ワイルドカード検索がサポートされます。ワイルドカード検索を開始するには、次のコマンド行を使用します。

```
$ srm_check [*...]* -log
```

チェックされたイメージのリストを作成するには、-log 修飾子を指定します。-output 修飾子を使用すれば、出力をデータ・ファイルに書き込むことができます。たとえば、次のコマンドは出力を CHECK.DAT という名前のファイルに書き込みます。

```
$ srm_check 'file' -output check.dat
```

このイメージの MAP ファイルを調べれば、ツールからの出力を使用して、このシーケンスを生成したモジュールを検索することができます。表示されるアドレスは、MAP ファイルから見つけることができるアドレスに直接対応しています。

次の例に、SYSTEM_SYNCHRONIZATION.EXE というイメージに対して、分析ツールを使用した結果でできる出力を示します。

```
** Potential Alpha Architecture Violation(s) found in file...  
** Found an unexpected ldq at 00003618  
0000360C AD970130 ldq_l R12, 0x130(R23)  
00003610 4596000A and R12, R22, R10  
00003614 F5400006 bne R10, 00003630  
00003618 A54B0000 ldq R10, (R11)  
Image Name: SYSTEM_SYNCHRONIZATION  
Image Ident: X-3  
Link Time: 5-NOV-1998 22:55:58.10  
Build Ident: X6P7-SSB-0000  
Header Size: 584  
Image Section: 0, vbn: 3, va: 0x0, flags: RESIDENT EXE (0x880)
```

system_synchronization.exe の MAP ファイルには、次の情報が格納されます。

```
EXEC$NONPAGED_CODE      00000000 0000B317 0000B318 (    45848.) 2 ** 5
SMPROUT      00000000 000047BB 000047BC (    18364.) 2 ** 5
SMPINITIAL    000047C0 000061E7 00001A28 (     6696.) 2 ** 5
```

アドレス 360C は SMPROUT モジュールにあり、0 ~ 47BB のアドレスが格納されています。モジュールから出力されたマシン・コードを確認することで、コードの位置を調べ、リスト行番号を使用して、対応するソース・コードを識別することができます。SMPROUT のベースが 0 以外の場合は、アドレス (この場合は 360C) からベースを減算して、リスト・ファイル内での相対アドレスを求める必要があります。

このツールは、可能性のある違反を出力の中で報告します。SRM_CHECK は通常、セクションの属性によってイメージ内のコード・セクションを識別することができますが、OpenVMS イメージの場合は、同じ属性を持つデータ・セクションが格納されることがあります。この結果、SRM_CHECK はデータをコードであるかのようにスキャンし、データ・ブロックを規則に準拠していないコード・シーケンスであると解釈することがあります。このような状況はあまり発生することがなく、MAP とリスト・ファイルを調べることで検出できます。

B.3 規則に準拠していないコードの特徴

SRM_CHECK ツールによって検出される規則に準拠しないコードは、次の 4 つに分類できます。これらの大部分は、新しいコンパイラで再コンパイルすることで修正できます。ソース・コードを変更しなければならないことがあります。そのような場合はまれです。コンパイラのバージョンの詳細については、第 B.5 節を参照してください。

- OpenVMS コンパイラの一部のバージョンでは、「ループ・ローテーション」と呼ばれる最適化中に規則に準拠しないコード・シーケンスが発生します。この問題が発生するのは、ASM 関数を使用して C/C++ ソースに埋め込まれているアセンブリ言語コード内で LDx_L/STx_C 命令を使用する C または C++ プログラムの場合か、MACRO-32 または MACRO-64 で作成されたアセンブリ言語の場合だけです。LDx_L 命令と STx_C 命令の間に分岐が発生していることもありました。

この問題は再コンパイルすることで対処できます。

- 非常に古い BLISS、MACRO-32、DEC Pascal、または DEC COBOL コンパイラでコンパイルされた一部のコードには、規則に準拠しないシーケンスが含まれていることがあります。これらのコンパイラの初期のバージョンには、コード・スケジューリングのバグがあり、load_locked の後にロードが誤ってスケジューリングされていました。

この問題は再コンパイルすることで対処できます。

- ごくまれに、空きレジスタの数が少なすぎる場合には、MACRO-32 コンパイラは BBSSI 命令または BBCCI 命令に対して、規則に準拠しないコード・シーケンスを生成することがあります。

この問題は再コンパイルすることで対処できます。

- MACRO-64 または MACRO-32 が誤ってコーディングされ、アセンブリ言語が ASM 関数を使用して C または C++ ソースに組み込まれ、誤ってコーディングされたために、エラーが発生することがあります。

この問題が発生する場合は、ソース・コードを変更しなければなりません。新しい MACRO-32 コンパイラは、規則に準拠しないコードに対して、コンパイル時にフラグを付けます。

SRM_CHECK ツールがイメージから違反を検出した場合は、適切なコンパイラを使用してイメージを再コンパイルしなければなりません (第 B.5 節を参照)。再コンパイルした後、イメージを再び分析する必要があります。再コンパイルの後も違反が発生する場合は、ソース・コードを調べ、コード・スケジューリング違反が発生する原因を追求しなければなりません。その後、必要に応じてソース・コードを変更します。

B.4 コーディングの必要条件

『Alpha Architecture Reference Manual』では、プロセッサ間のデータの不可分な更新を実行する方法を説明しています。特に『Third Edition』には、この問題に関するさらに詳しい情報が含まれています。また、インターロックされたメモリ・シーケンスの規則について詳しく説明されています。

次の 2 つの必要条件が満たされない場合は、規則に準拠しないコードが生成されます。

- インターロックされたシーケンスで、LDx_L (load locked) 命令と STx_C (store conditional) 命令の間にメモリ操作 (load または store) を指定できない。
- LDx_L 命令と STx_C 命令の間で分岐を実行できない。このような場合は、分岐を実行せずに、LDx_L から STx_C に「フォール・スルー」しなければならない。

ターゲットが LDx_L とそれに対応する STx_C の間にある分岐を実行すると、規則に準拠しないシーケンスが作成される。たとえば、次の例で "label" への分岐を実行すると、分岐命令自体がシーケンスの内部にあるか外部にあるかにかかわらず、規則に準拠しないコードが作成される。

```
LDx_L Rx, n(Ry)
...
label: ...
STx_C Rx, n(Ry)
```


したがって、SRM_CHECK ツールは次の条件を検索します。

- LDx_L と STx_C の間のメモリ操作 (LDx/STx)
- LDx_L と STx_C の間に宛先がある分岐
- 先行する LDx_L 命令のない STx_C 命令

これは通常、LDx_L から STx_C へ逆方向分岐が実行されることを示します。ただし、デバイス・メールボックス書き込みを実行するハードウェア・デバイス・ドライバは例外です。これらのドライバは、STx_C を使用してメールボックスに書き込みを実行します。この状況は初期の Alpha システムでのみ検出され、PCI ベースのシステムでは検出されません。

- LDx_L と STxC の間にある余分な命令

AARM では、LDx_L と STx_C の間の命令の数を 40 未満にすることを推奨しています。理論的には、40 より多くの命令があると、シーケンスが完了しないようにするためにハードウェア割り込みが発生します。しかし、実際にはこの状況が発生したという報告はありません。

次の例に、SRM_CHECK でフラグが付けられたコードを示します。

```
** Found an unexpected ldq at 0008291C
00082914 AC300000 ldq_l      R1, (R16)
00082918 2284FFEC lda        R20, 0xFFEC(R4)
0008291C A6A20038 ldq        R21, 0x38(R2)
```

この例では、LDQ 命令が LDQ_L の後、STQ_C の前に検出されています。LDQ は、再コンパイルまたはソース・コードの変更によって、このシーケンスの外部に移動しなければなりません (第 B.3 節を参照してください)。

```
** Backward branch from 000405B0 to a STx_C sequence at 0004059C
00040598 C3E00003 br         R31, 000405A8
0004059C 47F20400 bis         R31, R18, R0
000405A0 B8100000 stl_c      R0, (R16)
000405A4 F4000003 bne         R0, 000405B4
000405A8 A8300000 ldl_l      R1, (R16)
000405AC 40310DA0 cmple      R1, R17, R0
000405B0 F41FFFA bne         R0, 0004059C
```

この例では、LDL_L と STQ_C の間から分岐が検出されています。この場合、LDx_L と STx_C の間に、アーキテクチャで要求されている「フォール・スルー」パスがありません。

注意

LDx_L から STx_C へのこの逆向きの分岐は、「ループ・ローテーション」最適化によって発生する、規則に準拠しないコードの特徴です。

次の MACRO-32 ソース・コードは「フォール・スルー」パスがあるものの、ロック・シーケンス内に可能性のある分岐とメモリ参照があるために、規則に準拠しないコードを示しています。

```
getlck: evax_ldql  r0, lockdata(r8) ; Get the lock data
        movl      index, r2        ; and the current index.
        tstl      r0               ; If the lock is zero,
        beql      is_clear         ; skip ahead to store.
        movl      r3, r2           ; Else, set special index.
is_clear:
        incl      r0               ; Increment lock count
        evax_stqc  r0, lockdata(r8) ; and store it.
        tstl      r0               ; Did store succeed?
        beql      getlck           ; Retry if not.
```

このコードを修正するには、INDEX の値を読み込むためのメモリ・アクセスを最初に LDQ_L/STQ_C シーケンスの外部に移動しなければなりません。次に、ラベル IS_CLEAR への、LDQ_L と STQ_C の間の分岐を取り除かなければなりません。この場合、CMOVEQ 命令を使用して分岐を取り除くことができます。CMOVxx 命令はしばしば、単純な値の移動の周囲にある分岐を取り除くために役立っています。次の例に、修正されたコードを示します。

```
        movl      index, r2        ; Get the current index
getlck: evax_ldql  r0, lockdata(r8) ; and then the lock data.
        evax_cmoveq r0, r3, r2     ; If zero, use special index.
        incl      r0               ; Increment lock count
        evax_stqc  r0, lockdata(r8) ; and store it.
        tstl      r0               ; Did write succeed?
        beql      getlck           ; Retry if not.
```

B.5 コンパイラのバージョン

表 B-1 は、規則に準拠しないコード・シーケンスを生成する可能性のあるコンパイラのバージョンと、再コンパイルするときに使用する推奨最小バージョンについて説明します。

表 B-1 OpenVMS コンパイラのバージョン

旧いバージョン	推奨最小バージョン
BLISS V1.1	BLISS V1.3
DEC Ada V3.5	HP Ada V3.5A
DEC C V5.x	DEC C V6.0
DEC C++ V5.x	DEC C++ V6.0
DEC COBOL V2.4 , V2.5 , V2.6	DEC COBOL V2.8

(次ページに続く)

表 B-1 (続き) OpenVMS コンパイラのバージョン

古いバージョン	推奨最小バージョン
DEC Pascal V5.0-2	DEC Pascal V5.1-11
MACRO-32 V3.0	OpenVMS Version 7.1-2 については V3.1 OpenVMS Version 7.2 については V4.1
MACRO-64 V1.2	下記参照

MACRO-64 アセンブラの現在のバージョンでも、ループ回転の問題が発生することがあります。しかし、MACRO-64 ではデフォルトでコードの最適化が実行されないため、この問題が発生するのは、最適化が有効に設定されている場合だけです。SRM_CHECK が MACRO-64 コードから規則に準拠しないシーケンスを検出した場合は、最初に最適化せずに再コンパイルする必要があります。その後、再びテストしてもシーケンスにフラグが付けられる場合は、ソース・コード自体に修正の必要な非準拠シーケンスが含まれています。

21264 プロセッサのある Alpha コンピュータでは、『Alpha Architecture Reference Manual, Third Edition』に記載されている LDx_L 命令および STx_C 命令のインターロックされたメモリ・シーケンスの制限事項を厳密に守る必要があります。インターロックされたメモリ命令の使用がアーキテクチャのガイドラインに準拠するように、MACRO-32 for OpenVMS Alpha Version 3.1 コンパイラに、追加のチェックが盛り込まれました。

『Alpha Architecture Reference Manual, Third Edition』の 4.2.4 項に、インターロックされたメモリ・シーケンス内での命令の使用規則が説明されています。MACRO-32 for OpenVMS Alpha Version 3.1 コンパイラは、MACRO-32 ソース・コードから生成するコード内でこれらの規則に従います。ただし、このコンパイラは EVAX_LQxL および EVAX_STxC の組み込みを用意しているため、これらの命令を直接ソース・コードに記述することができます。

MACRO-32 for OpenVMS Alpha Version 4.1 コンパイラは、準拠していないコード・シーケンスを検出するために追加のコード・チェックを行い、警告メッセージを表示するようになりました。

B.6 ALONONPAGED_INLINE または LAL_REMOVE_FIRST によるコードの再コンパイル

OpenVMS Alpha の MACRO-32 コードのうち、SYSS\$LIBRARY:LIB.MLB マクロ・ライブラリから ALONONPAGED_INLINE マクロまたは LAL_REMOVE_FIRST マクロを起動するコードは、OpenVMS Version 7.2 以降で再コンパイルして、これらのマクロの正しいバージョンが取得されるようにしなければなりません。これらのマクロを変更すると、新しい Alpha 21264 (EV6) 以上のプロセッサで検出される可能性のある、同期化に関する問題を修正できます。

注意

EXE\$ALONONPAGED ルーチン (またはその変形) を呼び出すソース・モジュールは、再コンパイルする必要がありません。これらのモジュールは、ユーザに意識させることなく、このリリースに含まれているルーチンの正しいバージョンを使用します。

A

Ada コンパイラ	5-11
AlphaServer 2100	
SCSI コントローラの制限事項	6-3
コンソール表示	6-2
AlphaServer 4100	
FRU テーブルの制限事項	6-4
AlphaServer 8200 システム	
FRU テーブルの制限事項	6-4
AlphaServer 8400 システム	
FRU テーブルの制限事項	6-4
AlphaServer ES47/ES80/GS1280 システム	
MBM での時刻の設定	6-5
RAD のサポート	6-4
STOP/CPU およびシャットダウン動作	6-5
ソフト・パーティションでの INIT コンソール・	
コマンドの使用	6-4
ライセンス要件	6-5
AlphaServer GS シリーズ・システム	
デバイスの制限事項	6-6
複数 I/O ポートの制限事項	6-8
AlphaStation 200/400	
ISA_CONFIG.DAT の変更が必要	6-9
AlphaStation 255	
PCI 構成の制限事項	6-9
Analyze ユーティリティ (I64)	5-33
デバッグの選択的出力の修正	5-34
プロセス・リソースの解放の修正	5-34
ATI RADEON 7000 グラフィックス	6-10
ハードウェア・アクセラレーション 3D グラフィ	
ックスはサポートされていない	6-10
ATI RADEON 7500 グラフィックス	6-11 ~ 6-17
DECWSOPENGLSHR_RADEON.EXE のリネー	
ム	6-12
DECwindows サーバがハングアップ	6-13
OpenGL は IEEE 演算のみをサポート	6-13
高リフレッシュ・レートでの画像への影	
響	6-12
atof の問題の修正	5-19
AUDIT_SERVER	
起動失敗	3-13

B

Backup API	
ジャーナリング・イベント	5-11
BLISS	
HP BLISS コンパイラを参照	
BMC コンソールの制約事項 (I64 のみ)	6-1
BUGCHECKFATAL システム・パラメー	
タ	5-22
<builtins.h>の変更	5-17

C

C++ コンパイラ	
HP C++ コンパイラを参照	
CANCEL SELECTIVE 関数, LTDRIVER での使用	
の改善	5-42
CDSA	5-20
他社の署名処理	1-13
CLUE コマンド	
I64 に移植されていない	5-47
CMAP ファイル	
新規	2-2
COBOL RTL	
HP COBOL RTL を参照	
COM for OpenVMS	
アプリケーションの重負荷によるエラー	2-3
サポート	2-3
Command Definition ユーティリティ (I64)	
	5-34
I64 イメージのレコード属性の修正	5-34
Common Data Security Architecture	
CDSA を参照	
Compaq Open3D レイヤード・プロダクト	
V8.2 ではサポートされない	A-1
confstr	5-14
CPUSPINWAIT バグ・チェック	5-22
CRCTX ルーチンの機能の強化	6-28
CREATE/MAILBOX コマンド	
一時的な制約	3-2
C RTL	5-12 ~ 5-19
atof("NaN") の修正	5-19
<builtins.h>の変更	5-17
confstr	5-14
DECC\$*.OLBライブラリの凍結	5-20
DECC\$SHRP.EXE	5-17
<decc\$types.h>の変更	5-17

C RTL (続き)	
exec*のメモリ・リークの修正	5-14
exec1失敗後の exit 動作の修正	5-14
_FAST_TOUPPER マクロの追加	5-19
fcntl	5-18
fopen	5-15
fwrite	5-18
2 GB のmalloc	5-13
nanosleep	5-18
sigaction	5-13
struct time tの変更	5-17
TCP/IP ヘッダ・ファイルのアップデー	
ト	5-12
<time.h>の変更	5-15
toascii関数の追加	5-13
<wchar.h>の変更	5-17
ソケット転送	5-18
バックポート・ライブラリ	5-15
64 ビットの算術関数	5-13
ビルトイン fciの追加	5-19
ファイル・ポインタ・ロックのハングアッ	
プ	5-15
Ctrl/H キー・シーケンス	
DEL への再マッピング (I64 のみ)	6-2
C コンパイラ	
HP C コンパイラを参照	
C プログラム	
コンパイルと、大文字と小文字の区別	5-11
C ランタイム・ライブラリ	
C RTL を参照	
D	
data-reduced ELF オブジェクト・ライブラリ	
リンク対象	5-40
DCE RPC	
HP DCE RPC for OpenVMS を参照	
DCL コマンド	3-2
DCL コマンド・プロシージャ	
SET SHADOW と SHOW SHADOW の使	
用	4-25
DDB 構造体	
アップデート	5-5
DDT Intercept Establisher ルーチン	
デバイス構成	4-15
Debugger	
OpenVMS Debugger を参照	
DECamds	
V8.2 ではサポートされない	A-2
DECC\$*.OLB ライブラリの凍結	5-20
DECC\$SHRP.EXE	5-17
<decc\$types.h>の変更	5-17
DECdfs for OpenVMS	
Version 2.4 が必要	2-3
DECdtm	
Oracle 8i , 9i	4-4
DECevent	A-3
サポートされない	A-2
DECmigrate	
V8.2 Open Source Tools CD に無い	3-3
DECnet for OpenVMS	1-3
DECnet/OSI	
DECnet-Plus for OpenVMS を参照	
DECnet-Plus for OpenVMS	1-3
新しいバージョンが必要	1-17
DEC PL/I	2-3
DECram	
HP DECram を参照	
DECram	
DECRYPT コマンドとの競合	2-9
DECwindows Motif	
HP DECwindows Motif を参照	
DECwindows X11 ディスプレイ・サーバ	6-17
グラフィック・ボードのサポート	6-26
周辺デバイスの接続要件	1-10
DECwrite	
サポートの終了	A-3
Delta/XDelta デバッガ	5-22
レジスタ表示に関する考慮	5-22
DEVICE_NAMING システム・パラメータ	4-5
DIAGNOSE コマンド	
サポートされない	3-2, A-2
DIGITAL Modular Computing Components	
(DMCC)	
KZPDA コントローラと PBXGA グラフィック・	
カード	6-17
SRM コンソールの更新	6-18
Digital Personal Workstation	6-18
DSSI ディスク・デバイス	
マイクロコード・リビジョン・レベル	6-22
E	
ECP (Enterprise Capacity and Performance)	
	4-5
EDIT/FDL	
推奨バケット・サイズの変更	4-5
EFISCP ユーティリティ	
使用は推奨できない	4-6
EFI ドライバ	6-29
EFI の注意事項	4-6
ELV	
Error Log Viewer (ELV) ユーティリティを参照	
Enterprise Capacity and Performance (ECP)	
ECP を参照	
Error Log Report Formatter (ERF)	
サポート終了	A-3
Error Log Viewer (ELV) ユーティリティ	4-8
EV6 Alpha プロセッサ	B-1
exec*のメモリ・リークの修正	5-14
exec1失敗後の exit 動作の修正	5-14

F

_FAST_TOUPPER マクロの追加	5-19
Fast Path	
無効化, ES40 上の Galaxy	4-19
fcntl	5-18
Fibre Channel	6-30
互換キット	4-12
テープ・デバイスのマルチパス・フェールオーバーの制限事項	4-16
FMS キット	2-4
fopen	5-15
Fortran	
HP Fortran を参照	
Freeware	3-1
OpenVMS I64 でメニューが利用できない	3-2
fwrite	5-18

G

Galaxy	
定義	4-17
2 GB の malloc	5-13
Gigabit Ethernet コントローラ	
制限	1-5
Graphical Configuration Manager (GCM)	
Galaxy でのサポート	4-18
Graphical Configuration Utility (GCU)	4-18

H

HP BLISS コンパイラ	
規則に準拠していないコードを実行した結果	B-1
警告 (I64 のみ)	5-23
HP C++ コンパイラ	
規則に準拠していないコードを実行した結果	B-1
HP COBOL RTL	5-25
HP C コンパイラ	
規則に準拠していないコードを実行した結果	B-1
HP DCE RPC for OpenVMS	2-4 ~ 2-7
HP DECram	2-8
DECram と DECRYPT のコマンド競合	2-9
V8.2 では SIP として出荷	2-8
Version 2.5 (VAX のみ)	2-8
アップグレード前の削除 (Alpha のみ)	1-16
HP DECwindows Motif	
一時停止画面でロック解除できないことがある (Alpha のみ)	2-9
キーボードのサポートの制限 (I64 のみ)	1-10
サポートされているバージョン	1-5
スタートアップ・メッセージ	1-10
ユーザが作成したトランスポートのサポート	2-10

HP Fortran	
I64 用	5-25
HP Secure Web Browser	
ODS-2 でインストール・エラーになる (I64 のみ)	3-3
必要メモリ量の増加	3-3
HP Secure Web Server	
サポート	2-10
HP SSL	
暗号化と SSL のためのスタートアップ・コマンド	1-13
HSGmn	
障害	6-19
Hypersort ユーティリティ	5-29 ~ 5-31

I

IDE CD	6-18
IEEE 浮動小数点	
フィルタ (I64 のみ)	5-8
INIT コンソール・コマンド	
ES47/ES80/GS1280 ソフト・パーティションでの使用	6-4
Integrity サーバ	
ファームウェア	1-7
Intel®アセンブラ (I64 のみ)	5-32
Invocation context block	5-52
IPC コマンド	3-10
ISA_CONFIG.DAT	
将来のリリースでのサポートの中止	A-4

K

Kerberos	
Kerberos for OpenVMS	1-13
OpenVMS と Kerberos ACME エージェントのデフォルトの起動順序は必須	4-30
KPB 拡張	5-3

L

LANCP	
アップグレード後にデバイス・データベースを交換する (Alpha のみ)	1-17
LAN ドライバ	
二重モードの不一致エラー	3-12
LIB\$GET_CURR_INVO_CONTEXT	
ドキュメントの訂正	3-9
LIB\$GET_INVO_CONTEXT	
ドキュメントの訂正	3-9
LIB\$GET_INVO_HANDLE	
ドキュメントの訂正	3-9
LIB\$GET_PREV_INVO_CONTEXT	
ドキュメントの訂正	3-9
LIB\$GET_PREV_INVO_HANDLE	
ドキュメントの訂正	3-9
LIB\$GET_UIB_INFO	
ドキュメントの訂正	3-9

LIB\$I64_GET_FR 5-52
 LIB\$I64_GET_GR 5-52
 LIB\$I64_PUT_INVO_REGISTERS 5-52
 LIB\$I64_SET_FR 5-52
 LIB\$I64_SET_GR 5-52
 LIB\$PUT_INVO_REGISTERS
 ドキュメントの訂正 3-9
 LIB\$LOCK_IMAGE
 ヘルプからの漏れ 5-52
 LIBRARIAN
 Librarian ユーティリティを参照 3-7
 Librarian ユーティリティ 3-7, 5-32
 .STB ファイルの制限事項 (I64 のみ) 5-32
 エラー報告に関する問題 5-33
 Library ユーティリティ
 訂正情報
 ELF オブジェクト・ライブラリへのアクセ
 ス 3-7
 /REMOVE 3-7
 LIBRTL
 呼び出し標準ルーチン (I64 のみ) 5-52
 LINK_ORDER ELF セクション・ヘッダ・フラ
 グ 5-39
 LTDRIVER の制限事項 5-42

M

MACRO-32 コンパイラ
 規則に準拠していないコードを実行した結
 果 B-1
 コードの再コンパイル B-7
 MACRO-64 アセンブラ
 規則に準拠していないコードを実行した結
 果 B-1
 Mail ユーティリティ (MAIL)
 呼び出し可能メールがカーネル・スレッドで使用
 された場合の問題 5-43
 MMG_CTLFLAGS システム・パラメータ ... 4-22
 Monitor ユーティリティの変更 4-1
 MP コンソールの制約事項 (I64 のみ) 6-1
 MULTIPROCESSING システム・パラメー
 タ 5-22

N

nanosleep 5-18
 NetBeans
 Java Standard Edition, Development Kit v
 1.4.2-x が必要 2-12
 NetBeans V3.6 のサポートは OpenVMS V8.3 で
 終了 A-4

O

Open3D グラフィック
 コントローラ・ボードのサポート 6-26
 ライセンス方式の変更 6-19
 OpenVMS
 ENCRYPT コマンドと DECRYPT コマン
 ド 1-16
 OpenVMS Alpha 用 Linker 5-35 ~ 5-36
 RMS_RELATED_CONTEXT オプショ
 ン 5-35
 スタックのエレメント数は最大 25 に制
 限 5-36
 多数のファイルの処理時にハングアップす
 る 5-35
 ライブラリ・チェックの動作の変更 5-35
 OpenVMS Cluster システム 4-9 ~ 4-17
 CI と LAN との間の切り替えによる性能の低
 下 4-15
 SCSI マルチパス・フェイルオーバ 4-14
 互換キット 4-12
 パッチ・キット 4-11
 複合バージョン用の互換キット 4-12
 ローリング・アップグレード 1-12
 OpenVMS Debugger 5-43, 5-57
 Ada イベントのサポート 5-9
 Alpha
 以前のバージョンはサポートされな
 い 5-46
 Alpha および I64
 SET SCOPE コマンドの変更 5-46
 SHOW IMAGE コマンドの変更 5-46
 C++ 言語の問題 5-10
 Ctrl/C と STOP の制限 (Alpha のみ) 5-9
 EXAMINE/INSTRUCTION %PREVLOC コマン
 ドの修正 5-10
 Heap Analyzer の問題点 5-57
 I64
 Basic 言語での問題 5-45
 C++ 言語での問題 5-45
 COBOL 言語での問題 5-45
 Fortran 言語での問題 5-45
 Pascal 言語での問題 5-46
 全般的な状況と回避方法 5-44
 PC クライアントのドキュメントの訂正 ... 3-4
 SHOW MODULE コマンドの拡張 5-10
 SHOW SYMBOL/TYPE の動作の修正 5-9
 修正された問題点 5-43
 OpenVMS DELTA/XDELTA Debugger
 マニュアルの更新 3-5
 OpenVMS Galaxy 4-17 ~ 4-19
 および ES40
 Fast Path の無効化 4-19
 非圧縮ダンプの制限事項 4-19
 ライセンスの実行 6-6
 OpenVMS I64
 DVD からのブート 1-9

OpenVMS I64 用 Linker	5-36 ~ 5-41
data-reduced ELF オブジェクト・ライブラリ	5-40
/EXPORT_SYMBOL_VECTOR の削除	5-40
LINK_ORDER セクション・ヘッダ・フラグ	5-39
OpenVMS Alpha 用 Linker との違い	5-39
/PUBLISH_GLOBAL_SYMBOLS の削除	5-40
オブションでの長いシンボル名	5-41
作成されたコード・スタブ	5-41
初期化されたオーバーレイ・プログラム・セクション	5-40
デマングル化されたシンボル名	5-41
OpenVMS Management Station	4-19
『OpenVMS Performance Management』ドキュメントの訂正	4-22
OpenVMS Registry	
Version 2 フォーマットのデータベースの破壊	4-19
OpenVMS Utility Routines Manual	
更新	3-6
OpenVMS システム・ダンプ・アナライザ	
CLUE コマンドは I64 に移植されていない	5-47
OpenVMS の呼び出し標準規則	
ローテートするレジスタ	
OpenVMS 用 HP MACRO	5-27
Alpha システム上の	5-28
I64 システム上の	5-27
/OPTIMIZE=VAXREGS 修飾子は I64 ではサポートされない	5-29
/TIE 修飾子の Alpha と I64 でのデフォルト値	5-29
浮動小数点数のゼロ除算エラーが検出されない (I64 のみ)	5-29
OpenVMS 用 MACRO	
OpenVMS 用 HP MACRO を参照	

P

Pascal	
STARLET ライブラリの作成には V5.8A が必要 (Alpha のみ)	2-11
アップグレード後の再インストール (Alpha)	2-12
PCBST_TERMINAL	
サイズの拡張	5-5
PCI 構成の制限事項	6-9
PEdriver	
LAN 輻輳への対処	4-15
PGFLQUOTA の問題	5-33
PL/I	
I64 に含まれないライブラリ	5-47
RTL サポート	2-3
POOLCHECK システム・パラメータ	5-22

POSIX スレッド・ライブラリ	5-48 ~ 5-52
I64 での THREADCP コマンドの動作	5-49
POSIX 1003.4a Draft 4 インタフェースのサポート中止	A-4
デバッグ計測機能は動作しない	5-52
動的 CPU 構成	5-51
浮動小数点例外 (I64 のみ)	5-49
例外処理中のスタック・オーバーフロー (I64 のみ)	5-48
PowerStorm 300/350 PCI グラフィック・サポート	6-20
Open3D のライセンスはチェックされない	6-20
PTDSREAD ルーチン	
ドキュメントの明確化	3-6
PTDS疑似端末ドライバ	
情報の訂正	
/REMOVE	3-7

R

RF73 および RFnn ディスク , コントローラ・メモリ・エラー	6-21
RMS	
FAB	5-24
rx7620 サーバ	1-5
rx8620 サーバ	1-5
RZnn ディスク・ドライブ	6-23 ~ 6-25

S

SCSI コントローラ	
AlphaServer 2100 システムでの制限事項	6-3
SCSI デバイス・ドライバ	6-26
SCSI マルチパスの非互換性	4-14
SDA	
OpenVMS のシステム・ダンプ・アナライザを参照	
SET DEVICE/SWITCH コマンド	4-17
SET PASSWORD コマンド	4-8
SET SHADOW	
DCL コマンド・プロシージャでの使用	4-25
SHAD_MAX_UNIT	
メモリ消費量	1-18
SHADOW_MAX_UNIT	
デフォルト設定	1-18
SHOW SHADOW	
DCL コマンド・プロシージャでの使用	4-25
sigaction	5-13
Smart Array 5300	
Volume Shadowing の制限事項	4-27
SMGS	
ドキュメントの修正	5-53
Software Public Rollout Reports	2-1
SORT32 ユーティリティ	5-31, 5-54 ~ 5-55
SPLINVIPL バグ・チェック	5-7

SRM_CHECK ツール	B-2
struct time_t の変更	5-17
Superdome	
sx1000	6-25
Superdome サーバ	1-5
sx1000 Superdome	6-25
sx1000 チップセット	1-5
SYSTEM_CHECK システム・パラメータ	5-22
System Event Analyzer (SEA) ユーティリティ	
I64 でのサポート	2-12
System Event Log (SEL)	
Integrity サーバ上でのクリア	1-6

T

TCP/IP Services for OpenVMS	1-3
TCP/IP ヘッダ・ファイルのアップデート	5-12
Terminal Fallback Facility (TFF)	4-22
制限事項	4-23
TFF	
Terminal Fallback Facility を参照	
THREADCP コマンド	
I64 での動作	5-49
TIE キット	1-19
<time.h> の変更	5-15
toacii 関数の追加	5-13
TQE	
タイマ・キュー・エントリを参照	
Translated Image Environment	
TIE キットを参照	1-19

U

U160 SCSI のサポート	
rx7620, rx8620	1-6
UCB 構造体	
アップデート	5-5
USB	
DECwindows で複数のキーボードとマウスがサポートされない	6-17

V

VAX Cluster キャッシュ	
Virtual I/O キャッシュを参照	
VCC	
Virtual I/O キャッシュを参照	
VIOC	
Virtual I/O キャッシュを参照	
Virtual I/O キャッシュ	
I64 では利用不可	4-24
XFC に置き換え	4-24
Volume Shadowing for OpenVMS	4-24 ~ 4-29
EFI でシャドウ・システム・ディスクを操作する場合の注意事項	4-6
/MINICOPY を使用したディスマウントの問題	4-28

Volume Shadowing for OpenVMS (続き)

Smart Array 5300 (KZPDC) の制限事項	4-27
異種デバイス・シャドウイング (DDS)	
KZPDC の制限事項	4-27
注意事項	4-25
互換キット	4-12
デバイス名の必要条件	4-24
ビットマップのメモリ要件	3-11
マージ操作の性能	4-28
Volume Shadowing 用のビットマップ・メモリ要件	3-11

W

Watchpoint ユーティリティ	5-56
<wchar.h> の変更	5-17
WEBES	
I64 でのサポート	2-12

X

X.25 データ・リンクがサポートされていない (I64 のみ)	2-8
XA	4-4
XFC	
拡張ファイル・キャッシュを参照	

Z

ZLX グラフィック・ボードのサポート	6-26
---------------------	------

A

アーカイブされたマニュアル	A-5
アダプタについての注意事項	6-29
アップグレード	
パス	1-11

I

移行ソフトウェア	1-19
異種デバイス・シャドウイング (DDS)	
KZPDC の制限事項	4-27
Smart Array 5300 の制限事項	4-27
書き込みビットマップとの相互作用	4-25
インストール・エラー	
HP Secure Web Browser	3-3
インストールとアップグレードの情報	
ネットワーク・オプション	1-3
インターロックされたメモリ命令の使用	B-1

力

回線切り替え	
性能の低下	4-15
外部認証	4-8
I64 サポート	4-8

外部認証 (続き)

SET PASSWORD コマンド	4-8
パスワードの有効期限切れの通知	4-9
書き込みビットマップ	
異種デバイス・シャドウイング (DDS)	4-25
拡張 DDT ビット	
修正された問題点	5-42
拡張ファイル・キャッシュ (XFC)	4-24
関連製品	
Software Public Rollout Reports	2-1
現在のリリースでサポートされるバージョン	2-1

キ

規則に準拠していないコード	B-1, B-3
---------------	----------

ク

クラスタ	
OpenVMS Cluster システムを参照	
クラスタ互換キット	4-12
クラスタ互換性のためのパッチ・キット	4-11
グラフィック	
I64 システムでのサポート	6-10
グラフィック・ボードのサポート	6-26

ケ

現在のリリースでのアプリケーションのサポート	2-1
------------------------	-----

コ

コンパイラ	
規則に準拠していないコード	B-1, B-6

サ

削除キー	
再マッピングが必要 (I64 のみ)	6-2
サーバ	
rx7620	1-5
rx8620	1-5
Superdome	1-5

シ

シグナル配列	
I64 にない	4-32
システム・サービスの変更点	5-1
システム・ディスク	
古いシステムと互換性がない	1-4
システムのクラッシュ	
回復 (I64 のみ)	4-4
システムのハングアップ	
回復 (I64 のみ)	4-4

システム・パラメータ	4-20 ~ 4-22
BUGCHECKFATAL	5-22
DEVICE_NAMING	
デバイス・ユニット最大数の増加に使用	4-5
MMG_CTLFLAGS ドキュメントの誤り	4-22
MULTIPROCESSING	5-22
POOLCHECK	5-22
SYSTEM_CHECK	5-22
新しいパラメータ	4-20
廃止されたパラメータ	4-20
変更	4-21
修正キット	
入手方法	1-3
複合バージョンの OpenVMS Cluster システムで必要	4-12

ス

スレッド単位のセキュリティ	
デバイス・ドライバへの影響	5-6
特権付きコードへの影響	5-6

セ

セキュリティ	
ファイル属性	4-3
セクション	5-42
セル型システム	
複数の nPartitions	4-18

ソ

ソケット転送	5-18
ソフトウェアのサポート方針	1-1

タ

タイマ・キュー・エントリ (TQE)	5-56
--------------------	------

テ

デバイス構成	4-15
デバイス・ドライバ	
IPL の設定	6-28
MON バージョンの処理	6-27
SCSI	6-26
再コンパイルと再リンク	6-26 ~ 6-27
スレッド単位のセキュリティの影響	6-27
テープ・ロボット	
自動マルチパス・フェールオーバー	4-17
デュアル・コントローラ HSGnn	
障害	6-19

ト

動的 CPU 構成	
POSIX スレッド・ライブラリ	5-51
ドキュメントの訂正	3-4
IPC コマンドの使用法	3-10
\$PUTMSG システム・サービス	3-10
ドキュメントの変更と訂正	
『Guide to OpenVMS File Applications』	
.....	5-23
LIB\$ヘルプの漏れ	5-52
『OpenVMS Performance Management』	
.....	4-22
『OpenVMS RTL Screen Management (SMGS) Manual』	5-53
アーカイブされたマニュアル	A-5
特権データ構造体	
CPU の名前空間	5-4
KPB 拡張	5-3
PCBST_TERMINAL のサイズの拡張	5-5
UCB と DDB のアップデート	5-5
スレッド単位のセキュリティの影響	5-6
動的スピンロックのフォーク	5-4
64 ビットの論理ブロック番号	5-4
変更	5-2 ~ 5-6
トレースバック・ファシリティ	
API の問題の修正	4-31

ネ

ネットワーク	
アップデートの制限	3-11
ネットワーク・オプション	1-3

ハ

バックポート・ライブラリ	5-15
パッチ・キット	
複合バージョンの OpenVMS Cluster システムで必要	4-12
パーティション	
ソフト	4-17
ハード	4-17
ハード・パーティション	4-17

ヒ

64 ビットの sigaction の問題の修正	5-13
64 ビットの算術関数	5-13
ビルトイン __fci の追加	5-19

フ

ファイル属性	
推奨値の更新	4-3
ファイル・ポインタ・ロックのハングアップ	
ブ	5-15
ファームウェア	
Alpha サーバ用	1-11
Integrity サーバ用	1-7
複数の nPartitions	
セル型システム	4-18
浮動小数点型データ	
アプリケーションへの考慮	5-8
フリーウェア	A-1
プログラミング	
\$PUTMSG システム・サービスの訂正	3-10
プログラム全体の浮動小数点モード (I64 のみ)	5-56
プログラムの再コンパイル	
Alpha の場合	5-2

ホ

ポート・ドライバ \$QIO	
制限事項	5-42

マ

マイクロコード・リビジョン・レベル	
DSSI ディスク・デバイス	6-22
更新するためのコマンド	6-23
マルチパス・フェールオーバー	
Fibre Channel テープ・デバイスの制限事項	
.....	4-16
テープ・ロボット	4-17

ラ

ライセンス	4-12
ライセンスについて	6-6 ~ 6-8
ライブラリアン・ユーティリティ	
data-reduced ELF オブジェクト・ライブラリとのリンク (I64 での制限事項)	5-32
ランタイム・ライブラリ・ルーチン	3-9

リ

リタイア製品情報	A-1 ~ A-5
----------	-----------

レ

レイヤード・プロダクト	
インストールの失敗	1-19

□

ロケール	
新規	2-10
ローテートするレジスタ	5-52

HP OpenVMS V8.3 リリース・ノート【翻訳版】

2006 年 10 月 発行

日本ヒューレット・パカード株式会社

〒140-8641 東京都品川区東品川 2-2-24 天王洲セントラルタワー

電話 (03)5463-6600 (大代表)

BA322-90060

