

HP OpenVMS

V8.4 リリース・ノート【翻訳版】

部品番号: BA322-90102

2010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:

新規マニュアルです。

ソフトウェア・バージョン:

OpenVMS Integrity Version 8.4

OpenVMS Alpha Version 8.4

© Copyright 2010 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 8.4 Release Notes
© 2010 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 Intel Itanium 9300 ベース・サーバのサポートに関する情報	1-4
1.5 HP DECprint Supervisor のインストールに関する制限事項	1-4
1.6 ネットワーク・オプション	1-4
1.7 古いバージョンの OpenVMS と互換性のないディスク	1-5
1.8 HP DECwindows Motif for OpenVMS	1-5
1.9 アップグレード・パス	1-6
1.10 OpenVMS Integrity ユーザ向けの注意事項	1-8
1.10.1 ストレージ・コントローラ	1-8
1.10.2 rx7620 および rx8620 での U160 SCSI のサポート	1-9
1.10.3 Integrity サーバでの System Event Log (SEL) のクリア	1-9
1.10.4 Integrity サーバ用のファームウェア	1-10
1.10.5 インストール DVD からのブート	1-12
1.10.6 USB または vMedia デバイスからのブート	1-13
1.10.7 Small Memory Configurations Error Message	1-13
1.10.8 HP DECwindows Motif に関する注意事項	1-13
1.10.8.1 キーボードのサポート	1-13
1.11 OpenVMS Alpha 向けの注意事項	1-14
1.11.1 OpenVMS Alpha Version 8.4 のファームウェア	1-14
1.12 Kerberos for OpenVMS	1-14
1.13 SYSTARTUP_VMS.COM の修正	1-17
1.14 Encryption for OpenVMS	1-17
1.15 HP DECram V3.nのアップグレード	1-18
1.16 LANCP デバイス・データベースの変換	1-18
1.17 DECnet-Plus の新しいバージョンが必要	1-18
1.18 アップグレード前に TIE キットの削除が必要	1-19
1.19 代替デバイスや代替ディレクトリへのレイヤード・プロダクトのインストールの失敗	1-20

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
2.1.1	HP BASIC for OpenVMS	2-1
2.2	HP TCP/IP Services for OpenVMS	2-1
2.3	NetBeans Version 5.5.1 で必要となる最新の JDK	2-2
2.4	DFS マウントされたディスクへのアクセスの問題	2-2
2.5	HP DCE for OpenVMS の制限事項 (Integrity のみ)	2-3
2.6	XML-C 製品の ZIP ファイル	2-3
2.7	追加された CMAP ファイル	2-3
2.8	COBOL: I/O 実行時診断と RMS 特殊レジスタの変更	2-4
2.9	COM for HP OpenVMS (Alpha のみ)	2-4
2.9.1	COM for OpenVMS のサポート	2-4
2.9.2	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	2-4
2.10	サポートされるバージョンの DECdfs	2-4
2.11	DECforms Web Connector Version 3.0 (Alpha のみ)	2-5
2.12	DEC PL/I: OpenVMS での RTL のサポート	2-5
2.13	FMS キット	2-6
2.14	HP DECram	2-6
2.14.1	DECram は OpenVMS Version 8.2 以降に含まれる	2-6
2.14.2	DECRYPT DCL コマンドとの競合	2-7
2.15	HP DECwindows Motif for OpenVMS	2-7
2.15.1	新しいロケールの追加	2-7
2.15.2	ユーザが作成したトランスポートはサポートされない	2-8
2.16	HP Secure Web Server のサポート	2-8
2.17	HP Pascal for OpenVMS Alpha Systems	2-8
2.17.1	STARLET ライブラリの作成には HP Pascal V5.8A (以降) が必要 (Alpha のみ)	2-9
2.17.2	アップグレード後の HP Pascal のインストール (Alpha のみ)	2-9
2.18	I64 システムでの WEBES および SEA のサポート	2-9
3	一般ユーザ向けの注意事項	
3.1	AEST で変換したイメージの問題	3-1
3.2	SYS\$GETTIM_PREC システム・サービス宣言	3-1
3.3	F\$GETSYI("RAD_CPUS") の問題	3-1
3.4	HP Code Signing Service for OpenVMS	3-2
3.5	SHOW FORWARD/USER でユーザ名の表示が切り捨てられる問題	3-2
3.6	シンボリック・リンクの実装の変更	3-2
3.6.1	論理名	3-2
3.6.2	DIRECTORY コマンドで Audit アラームが発生する問題の修正	3-3
3.7	SHOW SYSTEM/STATE=MUTEX がプロセスを表示しない問題	3-3

3.8	HP Secure Web Browser V1.1-12 のインストール時の警告メッセージ	3-4
3.9	Ctrl/P の動作の問題	3-4
3.10	シリアル・ポートの名前	3-5
3.11	古いファームウェアでは VMS V8.3-1H1 がシステム・イベント・ログに書き込む メッセージを変換できない	3-7
3.12	CRTL 内の TZ 関数	3-9
3.13	InfoServer コーティリティと FDDI	3-9
3.14	DCL コマンドの SET PASSWORD の新しい修飾子	3-9
3.15	OpenVMS Freeware	3-10
3.16	DCL コマンド	3-11
3.16.1	OpenVMS グラフィック・コンソールでの SHUTDOWN.COM(Integrity のみ)	3-11
3.16.2	MOUNT コマンドの制限事項	3-11
3.16.3	SHOW LICENSE/CHARGE_TABLE で OpenVMS ゲスト・システムの ソケット数を認識しない問題	3-11
3.16.4	DIAGNOSE コマンドはサポートされない	3-12
3.17	Open Source Tools CD での DECmigrate の提供について	3-12
3.18	HP Secure Web Browser	3-12
3.18.1	必要メモリ量の増加	3-12
3.18.2	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 (Integrity のみ)	3-13
3.19	ドキュメントの訂正	3-13
3.19.1	『HP OpenVMS Linker Utility Manual』の訂正	3-13
3.19.1.1	HP C++ Examples	3-13
3.19.2	『HP PCSI Utility Online help and Manual』: \$PRODUCT REGISTER VOLUME の構文エラーの訂正	3-14
3.19.3	『iCAP Release Notes』: GiCAP の機能は現在使用できない	3-14
3.19.4	『POLYCENTER Software Installation Utility Developer's Guide』: PRODUCT コマンドの訂正	3-14
3.19.5	『HP OpenVMS RTL Library (LIB\$) Manual』の訂正	3-15
3.19.6	ドキュメントの訂正: LCKMGR_CPUID システム・パラメータ	3-15
3.19.7	MMG_CTLFLAGS: ドキュメントの訂正	3-15
3.19.8	『HP OpenVMS System Analysis Tools Manual』	3-15
3.19.9	『HP OpenVMS Programming Concepts Manual』	3-15
3.19.9.1	システム・ダンプの保存	3-16
3.19.10	『HP OpenVMS DELTA/XDELTA Debugger Manual』の更新	3-16
3.19.11	『HP OpenVMS Version 8.2 新機能説明書』: Librarian コーティリテ イの訂正	3-16
3.19.11.1	/REMOVE 修飾子の訂正	3-16
3.19.11.2	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	3-17
3.19.12	『OpenVMS RTL Library (LIB\$) Manual』の訂正	3-18
3.19.12.1	『OpenVMS RTL Library (LIB\$) Manual』: LIB\$CVT_DX_DX の丸め規則の明確化	3-18
3.19.13	『OpenVMS RTL Library (LIB\$) Manual』: プラットフォームの制限 の明確化	3-18
3.19.14	『OpenVMS システム管理者マニュアル』: IPC コマンドの制限	3-19
3.20	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	3-20

3.21	同期データ・リンクの非サポート	3-20
3.22	LAN ドライバから報告される二重モード不一致エラー	3-20
4	システム管理に関する注意事項	
4.1	SYS\$TIMEZONE_RULE 論理名がハイフン (-) をカレット (^) に置換	4-1
4.2	Virtual オプション付きでのライセンス処理	4-1
4.3	iSCSI デモ・キット	4-2
4.4	Integrity VM における OpenVMS ゲスト OS	4-2
4.4.1	シャットダウン動作の変更	4-2
4.4.2	OpenVMS ゲストによる外付け I/O デバイスのサポートに関する注意事項	4-3
4.4.3	ネットワークおよびストレージ・インタフェースのサポート	4-3
4.4.4	HP-UX ゲストと OpenVMS ゲストでの同じ仮想スイッチの使用に関する制限事項	4-3
4.4.5	vNICs が構成されていない場合の OpenVMS ゲストの既知の問題	4-3
4.5	HP Availability Manager に関する注意事項	4-4
4.6	HP SIM による OpenVMS のプロビジョニング	4-4
4.6.1	OpenVMS ゲストのプロビジョニングに関する制限	4-5
4.6.2	システム・ファームウェアに関する注意事項	4-5
4.6.3	複数のサーバのプロビジョニング	4-5
4.6.4	HP SIM Central Management Server からのプロビジョニング	4-5
4.6.5	InfoServer の名前の長さ	4-5
4.6.6	OpenVMS InfoServer と Integrity サーバが同一 LAN 上に存在しなければならぬケース	4-6
4.6.7	EFI ファーム・ウェア	4-6
4.6.8	Management Processor	4-6
4.6.9	プロビジョニングによる OpenVMS TCP/IP の構成に関する既知の問題	4-6
4.6.10	OpenVMS TCP/IP のプロビジョニングに関する制限事項	4-6
4.6.11	AutoBoot タイムアウト値の設定	4-7
4.7	Insight ソフトウェアによる OpenVMS の管理	4-7
4.8	性能の強化	4-7
4.8.1	書き込みビットマップの拡張	4-7
4.8.1.1	WBM_MSG_INT パラメータの修正	4-8
4.8.1.2	WBM_MSG_UPPER および WBM_MSG_LOWER パラメータの修正	4-8
4.8.1.3	非同期 SetBit メッセージ	4-8
4.8.1.4	シークエンシャル I/O に対する SetBit メッセージの削減	4-9
4.8.2	例外処理性能の改善 (Integrity のみ)	4-9
4.8.3	イメージの起動 (Integrity のみ)	4-9
4.8.4	グローバル・セクションの作成と削除	4-10
4.8.5	Dedicated CPU Lock Manager	4-10
4.8.6	Ctrl/T アライメント・フォルト	4-10
4.9	ACPI ブート時のエラー・メッセージおよび警告メッセージ	4-10
4.10	Accounting ユーティリティでの長いデバイス名のサポート	4-11
4.11	新しいシステム・パラメータ PAGED_LAL_SIZE	4-11
4.11.1	ページング・プールのルックアサイド・リスト	4-11

4.12	2 TB ディスク・ボリュームのサポートに関する制限事項	4-12
4.13	SAS テープ・ドライブの構成	4-12
4.14	外付け SAS ディスク・デバイスのネーミング	4-13
4.15	外部認証	4-13
4.15.1	外部認証とパスワードのポリシー	4-13
4.15.2	OpenVMS Integrity 外部認証サポート	4-14
4.15.3	DECterm 端末セッションでの SET PASSWORD の動作	4-14
4.15.4	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	4-14
4.15.5	ACME_SERVER プロセスにおける制限事項 (Integrity サーバのみ)	4-15
4.16	IPB (Itanium Primary Bootstrap) が有効なダンプ・デバイスの検出に失敗する問題	4-15
4.17	SHUTDOWN.COM の変更	4-15
4.18	OpenVMS Cluster システム	4-16
4.18.1	Cluster over IP (IP クラスタ・インターコネクト)	4-16
4.18.1.1	必要なソフトウェア	4-16
4.18.1.2	Integrity サーバのサテライトノードとブートサーバは同一 LAN 内に存在	4-16
4.18.1.3	Alpha サテライト・ノードではディスク・サーバとの LAN チャンネルを必要とする	4-16
4.18.1.4	IPv6 のサポート	4-17
4.18.1.5	DHCP (Dynamic Host Configuration Protocol) あるいはセカンダリ・アドレスのサポート	4-17
4.18.1.6	複数の IP インタフェース構成	4-17
4.18.1.7	ifconfig コマンドの使用	4-17
4.18.1.8	複数のゲートウェイ構成	4-17
4.18.1.9	ブロック転送 XMIT の変更	4-17
4.18.1.10	ダウンライン・ロードのための LANCP	4-18
4.18.1.11	デュプレックスの不一致	4-18
4.18.1.12	共有システム・ディスクのアップグレード	4-18
4.18.1.13	拡張された CLUSTER_CONFIG_LAN コマンド・プロシージャ	4-18
4.18.2	Integrity VM での OpenVMS Cluster のサポート	4-19
4.18.2.1	OpenVMS ゲストのクラスタ・インターコネクト	4-19
4.18.2.2	Integrity VM クラスタ環境での MSCP のサポート	4-19
4.18.2.3	オンライン・マイグレーションのサポート	4-19
4.18.3	混成プラットフォームのサポート	4-19
4.18.4	ポート割り当てクラスを使用したサテライト・システム	4-20
4.19	6 メンバ・シャドウセットの混成クラスタにおける互換性	4-20
4.20	6 メンバ・シャドウセットの下位互換性	4-21
4.21	WBEM Services および WBEM Providers に関する注意事項	4-21
4.21.1	OpenPegasus 2.9 ベースの WBEM Services for OpenVMS	4-22
4.21.2	OpenVMS ゲストに対する WBEM Providers のサポート	4-22
4.21.3	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	4-22
4.21.4	コマンド行オプションは引用符で囲む	4-22
4.22	Monitor ユーティリティの変更	4-22
4.22.1	Integrity VM におけるゲスト・オペレーティング・システム	4-23
4.22.2	MONITOR データのバージョン間での互換性	4-24
4.22.3	記録ファイルからのデータのプレイバック	4-24

4.23	新しいパラメータ	4-25
4.24	SYS\$LDDRIVER の制限事項	4-25
4.25	システム・パラメータ CPU_POWER_MGMT のデフォルト値の変更	4-26
4.26	予約メモリ機能を使用するサテライト・システムのブート	4-26
4.27	SCACP エラー・カウンタが再送エラーをレポートすることがある	4-27
4.28	Virtual Connect	4-27
4.28.1	フェールオーバーと RECNXINTERVAL	4-27
4.29	メディア使用前の INITIALIZE/ERASE	4-27
4.30	Performance Data Collector for OpenVMS (TDC)	4-28
4.31	システムのハングアップまたはクラッシュからの回復 (Integrity のみ)	4-28
4.32	Oracle 8iおよび 9iでの DECdtm/XA (Alpha のみ)	4-29
4.33	デバイス・ユニットの最大数の増加	4-29
4.34	EDIT/FDL: 推奨バケット・サイズの変更	4-29
4.35	EFI\$CP ユーティリティ: 使用は推奨できない	4-30
4.36	Error Log Viewer (ELV) ユーティリティ: TRANSLATE/PAGE コマンド	4-30
4.37	クラスタ互換性パッチ・キット	4-30
4.37.1	クラスタの互換性のために必要なパッチ・キット	4-31
4.37.2	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	4-33
4.37.3	DDT Intercept Establisher ルーチンとデバイス構成通知結果	4-34
4.37.4	CI と LAN との間のお線切り替えによるクラスタの性能の低下	4-34
4.37.5	マルチパス・テープ・フェールオーバーの制限事項	4-35
4.37.6	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	4-36
4.38	OpenVMS Galaxy (Alpha のみ)	4-36
4.38.1	Galaxy の定義	4-36
4.39	セル型システムでの複数の nPartition	4-37
4.39.1	ES40 上の Galaxy: 非圧縮ダンプの制限事項	4-37
4.39.2	ES40 上の Galaxy: Fast Path の無効化	4-38
4.40	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	4-38
4.41	システム・パラメータ	4-38
4.41.1	新しいシステム・パラメータ	4-38
4.41.2	廃止されたシステム・パラメータ	4-39
4.41.3	システム・パラメータの変更	4-40
4.42	Terminal Fallback Facility (TFF)	4-40
4.43	User Environment Test Package (UETP) (Integrity のみ)	4-42
4.44	推奨するキャッシュ方式	4-42
4.45	OpenVMS の Analyze ユーティリティ	4-42
4.45.1	フォーマットされたシンボル・ベクタがデータ・セグメントに正しく表 示される	4-43
4.45.2	トランスファー・アレイがデータ・セグメント内にフォーマットされる ようになった	4-43

4.45.3	システム・バージョン・アレイがダイナミック・セグメント内にフォーマットされるようになった	4-44
4.45.4	/SEGMENT 修飾子の機能拡張	4-44
4.45.5	セクション・エスケープ機能のサポート	4-44
4.46	OpenVMS の INSTALL コーティリティ (S2 空間に常駐イメージをインストールする)	4-45
5	プログラミングに関する注意事項	
5.1	lib\$routines.h ファイルにおけるプロトタイプ宣言の修正	5-1
5.2	シンボリック・デバッガ	5-1
5.3	C ランタイム・ライブラリ	5-1
5.4	プロセス/アプリケーションがハングする	5-4
5.5	POSIX スレッドを使用しているプログラムでの AST 実行要求の明確化	5-4
5.6	ディレクトリ・ファイルの RMS \$PARSE 検証	5-5
5.7	IOLOCK8 を使わない FibreChannel ポート・ドライバ	5-5
5.8	C++ コンパイラ	5-6
5.9	DCE IDL C++ アプリケーションのビルド	5-7
5.10	特権プログラムの再コンパイルが必要 (Alpha のみ)	5-7
5.11	特権データ構造体の変更	5-7
5.11.1	KPB 拡張	5-8
5.11.2	CPU の名前空間	5-9
5.11.3	64 ビットの論理ブロック番号 (LBN)	5-9
5.11.4	動的スピンロックのフォーク	5-9
5.11.5	UCB と DDB のアップデート	5-10
5.11.6	PCBST_TERMINAL のサイズの拡張	5-10
5.11.7	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	5-11
5.12	浮動小数点型データを使用するアプリケーション	5-13
5.12.1	IEEE 浮動小数点フィルタ (Integrity のみ)	5-13
5.12.2	Ada イベントのサポート (Integrity のみ)	5-14
5.12.3	C++ 言語の問題 (Integrity のみ)	5-14
5.13	Ada コンパイラ (Integrity のみ)	5-14
5.14	Backup API: ジャーナリング・コールバック・イベントの制限事項	5-14
5.15	C プログラム: CASE_LOOKUP=SENSITIVE を設定したコンパイル	5-15
5.16	C ランタイム・ライブラリ	5-15
5.16.1	C RTL TCP/IP ヘッダ・ファイルのアップデート	5-15
5.16.2	バックポート・ライブラリが提供されなくなった	5-16
5.16.3	ヘッダ・ファイル<time.h>の変更	5-16
5.16.4	ヘッダ・ファイル<time.h>での非 ANSI の*_r 関数の参照	5-17
5.16.5	ヘッダ・ファイル<builtins.h>の__CMP_SWAP*と_Interlocked*が C++ から参照可能	5-17
5.16.6	Integrity システムへのビルトイン__fci の追加	5-17
5.16.7	DECC\$.OLB オブジェクト・ライブラリに新しいエントリがない	5-18
5.17	呼び出し標準規則とローテートするレジスタ (Integrity のみ)	5-18

5.18	Common Data Security Architecture (CDSA) に関する考慮	5-18
5.18.1	Secure Delivery	5-19
5.18.2	インストールと初期化に関する注意事項	5-19
5.19	デバッグ・モード: CPUSPINWAIT バグ・チェックの回避	5-20
5.20	Delta/XDelta デバッガ	5-21
5.20.1	XDelta のレジスタ表示に関する考慮 (Integrity のみ)	5-21
5.21	ファイル・アプリケーション: 『Guide to OpenVMS File Applications』 の訂正	5-21
5.22	RMS 構造体についての HP BLISS コンパイラの警告 (Integrity のみ)	5-22
5.23	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	5-23
5.24	HP COBOL ランタイム・ライブラリ (RTL)	5-24
5.24.1	COBOL CALL 文の性能改善	5-24
5.25	HP Fortran for Integrity Servers	5-24
5.26	HP MACRO for OpenVMS	5-25
5.26.1	Macro-32 コンパイラの拡張	5-25
5.26.2	OpenVMS Integrity 用 HP MACRO	5-28
5.26.3	OpenVMS Alpha システム用の HP MACRO	5-29
5.26.4	/OPTIMIZE=VAXREGS 修飾子は Integrity サーバではサポートされない	5-29
5.26.5	浮動小数点数のゼロ除算エラーが検出されない (Integrity のみ)	5-30
5.27	Hypersort ユーティリティ	5-30
5.27.1	弊社への問題の報告	5-30
5.27.2	ラージ・ファイルの制限事項	5-30
5.27.3	Hypersort と VFC ファイルの制限事項	5-31
5.27.4	/FORMAT=RECORD_SIZE の制限事項	5-31
5.27.5	Hypersort と検索リスト, および論理名の使用	5-31
5.27.6	作業ファイルの空き領域不足	5-31
5.27.7	入力アスタリスク (*) の制限事項	5-32
5.27.8	最適化されたワーキング・セット・エクステンとページ・ファイル・クォータの設定	5-32
5.28	Intel®アセンブラ (Integrity のみ)	5-32
5.29	Librarian ユーティリティ	5-32
5.29.1	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 (Integrity のみ)	5-32
5.29.2	Integrity サーバ・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 (Integrity のみ)	5-33
5.29.3	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
5.30	OpenVMS Alpha 用 Linker ユーティリティ	5-34
5.30.1	SYMBOL_VECTOR リンカ・オプションの制限事項	5-34
5.30.2	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	5-34
5.30.3	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
5.30.4	スタックのエレメント数は最大 25 に制限	5-36
5.31	OpenVMS Integrity 用 Linker ユーティリティ	5-36
5.31.1	SYMBOL_VECTOR リンカ・オプションの制限事項	5-36

5.31.2	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
5.31.3	/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
5.31.4	最大セクション数	5-38
5.31.5	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	5-39
5.31.6	demangler 情報を検索するとアクセス違反になる	5-39
5.31.7	NOGLOSYM エラー・メッセージに対する誤った二次メッセージ	5-39
5.31.8	未定義シンボルについての誤った情報	5-39
5.31.9	誤った UNMAPFIL エラー	5-40
5.31.10	共用可能イメージ・マップ内の識別子の最大長の変更	5-40
5.31.11	共用可能イメージに対するリンケージ・タイプ・チェック	5-40
5.31.12	プログラム・セクションの ABS 属性が無視される	5-40
5.31.13	コマンド行に FP_MODE リテラルを指定していないとリンクはアクセス違反となる	5-41
5.31.14	OpenVMS Integrity 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	5-41
5.31.15	Integrity リンカと Alpha リンカの違い	5-41
5.31.16	LINK_ORDER セクション・ヘッダ・フラグはサポートされていない	5-41
5.31.17	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	5-42
5.31.18	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	5-43
5.31.19	リンクの修飾子/EXPORT_SYMBOL_VECTOR と/PUBLISH_GLOBAL_SYMBOLS の削除	5-43
5.31.20	オプションでの長いシンボル名のサポート	5-43
5.31.21	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	5-44
5.31.22	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	5-44
5.32	LTDRIVER: CANCEL SELECTIVE の制限事項	5-44
5.33	Mail コーティリティ: 呼び出し可能メールのスレッドの制限事項	5-45
5.34	OpenVMS のシステム・ダンプ・アナライザ (SDA)	5-45
5.34.1	CLUE コマンドは OpenVMS Integrity に移植されていない	5-45
5.35	OpenVMS Integrity Version 8.2 に含まれない PL/I ライブラリ	5-45
5.36	POSIX スレッド・ライブラリ	5-46
5.36.1	プロセス共有オブジェクトのサポート	5-46
5.36.2	pthread_mutex_lock の新しい戻り状態値	5-46
5.36.3	新しい API pthread_mutex_tryforcedlock_np のサポート	5-47
5.36.4	例外処理中のスタック・オーバフロー (Integrity のみ)	5-48
5.36.5	Integrity サーバでの THREADCP コマンドの動作	5-49
5.36.6	浮動小数点のコンパイルと例外 (Integrity のみ)	5-49
5.36.7	C 言語コンパイル・ヘッダ・ファイルの変更	5-49
5.36.8	新しい優先順位調整アルゴリズム	5-50
5.36.9	プロセス・ダンプ	5-51
5.36.10	動的 CPU 構成の変更	5-51
5.36.11	デバッガ計測機能は動作しない	5-52
5.37	RTL ライブラリ (LIB\$)	5-52
5.37.1	RTL ライブラリ (LIB\$) のヘルプ	5-52
5.37.2	RTL Library (LIB\$): 呼び出し標準ルーチン (Integrity のみ)	5-52
5.38	Screen Management (SMG\$) のドキュメント	5-53

5.39	SORT32 ユーティリティ	5-54
5.39.1	DFS サービス・ディスクでの CONVERT の問題	5-54
5.39.2	一時作業ファイルが削除されないことがある	5-54
5.39.3	複合条件のある SORT/SPECIFICATION: 要件	5-55
5.39.4	可変長レコードでの性能の問題	5-55
5.39.5	作業ファイル・ディレクトリの制約事項	5-55
5.40	タイマ・キュー・エントリ (TQE)	5-55
5.41	Watchpoint ユーティリティ (Integrity のみ)	5-56
5.42	プログラム全体の浮動小数点モード (Integrity のみ)	5-56
6	ハードウェアに関する注意事項	
6.1	USB デバイスのサポート	6-2
6.2	MP コンソールと BMC コンソールの制約事項 (Integrity のみ)	6-2
6.2.1	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
6.2.2	削除キーへの Ctrl/H の再マッピング	6-3
6.3	AlphaServer 2100	6-3
6.3.1	コンソール表示	6-3
6.3.2	SCSI コントローラの制限事項	6-4
6.4	AlphaServer 8200/8400: FRU テーブル・エラー	6-5
6.5	AlphaServer ES47/ES80/GS1280 システム	6-5
6.5.1	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	6-5
6.5.2	RAD のサポート	6-5
6.5.3	ライセンス要件	6-6
6.5.4	STOP/CPU およびシャットダウン動作	6-6
6.5.5	MBM での時刻の設定	6-6
6.6	AlphaServer GS シリーズ・システム	6-6
6.6.1	AlphaServer GS80/160/320 システム: デバイスの制限事項	6-7
6.6.2	OpenVMS Galaxy ライセンスの実行	6-7
6.6.3	ライセンスのインストール	6-7
6.6.4	AlphaServer GS60/GS60E/GS140 複数 I/O ポート・モジュール構成の制限事項	6-9
6.7	AlphaStation 200/400: ISA_CONFIG.DAT の変更が必要	6-10
6.8	AlphaStation 255: PCI 構成の制限事項	6-10
6.9	ATI RADEON 7000 グラフィック (Integrity のみ)	6-11
6.9.1	Integrity でのグラフィック・サポート	6-11
6.9.2	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	6-11
6.10	ATI RADEON 7500 グラフィック	6-11
6.10.1	リソースの要件	6-12
6.10.2	DECWSOPENGLSHR_RADEON.EXE を DECWSMESA3DSHR_RADEON.EXE にリネーム	6-13
6.10.3	サポートの制限事項	6-13
6.10.4	高リフレッシュ・レートでの画像への影響	6-13
6.10.5	OpenGL は IEEE 演算のみをサポート	6-14
6.10.6	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	6-14

6.10.7	モニタは初期化時に接続されていなければならない	6-14
6.10.8	ブート・リセットの推奨 (Alpha のみ)	6-15
6.10.9	オーバレイ・プレーンはサポートされない	6-15
6.10.10	単一カラーマップのみサポート	6-15
6.10.11	すべてのウィンドウで同じビット深度	6-15
6.10.12	読み取り/書き込みカラー・マップのピクセル深度	6-15
6.10.13	backing store と save unders は 3D ウィンドウをサポートして いない	6-16
6.10.14	スレッドの制限事項	6-16
6.10.15	シングル・バッファ・ビジュアルがサポートされていない	6-16
6.10.16	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
6.10.17	タイマ・メカニズム	6-17
6.11	DECwindows X11 ディスプレイ・サーバ (Alpha のみ)	6-18
6.11.1	S3 マルチヘッド・グラフィック	6-18
6.12	DIGITAL Modular Computing Components (DMCC)	6-18
6.12.1	Alpha 5/366 および 5/433 PICMG SBC の制限事項	6-18
6.12.2	SRM コンソールの更新	6-18
6.13	Digital Personal Workstation: OpenVMS V7.3-1 およびそれ以降のブート	6-19
6.14	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
6.15	Open3D グラフィックのライセンス方式の変更	6-20
6.16	OpenVMS 用の PowerStorm 300/350 PCI グラフィック・サポート	6-21
6.17	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
6.18	RZnn ディスク・ドライブの考慮事項	6-24
6.18.1	RZ25M と RZ26N ディスク・ドライブ: 推奨事項	6-24
6.18.2	RZ26N および RZ28M ディスク: 推奨ファームウェア・サポート	6-25
6.18.3	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	6-25
6.18.3.1	ファームウェア・リビジョン・レベル 442 の必要条件	6-25
6.18.3.2	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	6-26
6.19	sx1000 Integrity Superdome	6-26
6.20	ZLX グラフィック・ボードのサポート	6-26
6.21	OpenVMS デバイス・ドライバの再コンパイルと再リンク	6-27
6.21.1	Alpha および VAX の SCSI デバイス・ドライバ	6-27
6.21.2	OpenVMS Alpha デバイス・ドライバ	6-27
6.22	MON バージョンのデバイス・ドライバの処理	6-28
6.23	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	6-28
6.24	OpenVMS Alpha ドライバのデバイス IPL の設定	6-28
6.25	CRCTX ルーチンの機能の強化	6-29
6.26	アダプタについての注意事項	6-30
6.26.1	Fibre Channel の EFI ドライバとファームウェアの要件	6-30
6.26.2	複数の Fibre Channel ブート・エントリを使用したブート	6-30

A	インターロックされたメモリ命令の使用 (Alpha のみ)	
A.1	必要なコード・チェック	A-1
A.2	コード分析ツール (SRM_CHECK) の使用	A-2
A.3	規則に準拠していないコードの特徴	A-3
A.4	コーディングの必要条件	A-4
A.5	コンパイラのバージョン	A-6
A.6	ALONONPAGED_INLINE または LAL_REMOVE_FIRST によるコードの再コンパイル	A-7

索引

表

1-1	サポートされている DECwindows Motif のバージョン	1-6
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	1-10
4-1	クラスタ互換性のために必要なパッチ・キット	4-31
4-2	クラスタの互換性のために必要なパッチ・キット	4-32
4-3	Galaxy の定義	4-36
4-4	TFF 文字フォールバック・テーブル	4-41
5-1	Macro-32 の新しいビルトイン	5-26
6-1	デバイス記述ブロックの変更点	6-10
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	6-24
6-4	リビジョン・レベル 442 ファームウェアの互換性	6-26
A-1	OpenVMS コンパイラのバージョン	A-6

対象読者

本書は、HP OpenVMS Alpha または HP OpenVMS I64 Integrity サーバ Version 8.4 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 8.4 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、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、V8.3 に

対する制限事項で、Version 8.4 で修正されたものには、V8.4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 8.4 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 Integrity サーバ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 8.4 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 V8.4 リリース・ノート[翻訳版]』 (本書)
- 『HP OpenVMS V8.4 新機能説明書』
- 『HP OpenVMS V8.4 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。

注意

日本語 OpenVMS のインストールに関しては、以下のドキュメントを参照してください。

- 『日本語 OpenVMS V8.4 リリース・ノート』
 - 『日本語 OpenVMS V8.4 インストレーション・ガイド』
-

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

Integrity, OpenVMS Alpha および 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: 7.3-2 から 8.2 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: 8.2 から 8.3 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: 8.2-1)。

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 DECnet (フェーズ IV)
- HP DECnet-Plus for OpenVMS
- HP OpenVMS Cluster Client Software
- HP OpenVMS Cluster Software 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 Intel Itanium 9300 ベース・サーバのサポートに関する情報

V8.4

新しい Integrity サーバは今後提供される予定のアップデート・キットによってサポートが可能になりますが、OpenVMS Version 8.4 にはこれを可能にするための機能が含まれています。

1.5 HP DECprint Supervisor のインストールに関する制限事項

V8.4

HP DECprint Supervisor (DCPS) のインストールの際、現在のディレクトリが CD あるいは DVD メディア上のディレクトリになっていると、次のようなエラー・メッセージでインストールが失敗します。

```
%DCL-W-INVRANGE, field specification is out of bounds - check sign and size  
%SYSTEM-F-IVLOGNAM, invalid logical name
```

この問題を回避するたくめに、作業ディレクトリをシステム・ディスクに設定してからキットをインストールしてください。

```
$ SET DEF SYS$SYSTEM  
$ PROD INST /SOURCE=VM141$DKA1:[000000.DCPS_I64027.KIT] *
```

この問題は、HP DECprint Supervisor (DCPS) の将来のリリースで修正される予定です。

1.6 ネットワーク・オプション

V8.4

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 8.4 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 8.4 for OpenVMS または HP DECnet フェーズ IV Version 8.4 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。

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。これらのマニュアルは OpenVMS のドキュメント Web サイトで参照することができます。

OpenVMS Version 8.4 へアップグレードした場合は、TCP/IP Services for OpenVMS Version 5.7 を使用する必要があります。

1.7 古いバージョンの OpenVMS と互換性のないディスク

V8.3

OpenVMS のインストール・プロシージャは、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 とのディスク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場合があります。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。

1.8 HP DECwindows Motif for OpenVMS

V8.4

次の表に、HP OpenVMS 8.4 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 Integrity V8.4	DECwindows Motif for OpenVMS Integrity V1.7
OpenVMS Alpha V8.4	DECwindows Motif for OpenVMS Alpha V1.7

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。

1.9 アップグレード・パス

V8.4

OpenVMS Version 8.4 へ直接アップグレード可能なアップグレード・パスは以下のとおりです。

OpenVMS Alpha の場合:

- Version 7.3-2 から V8.4 へ
- Version 8.2 から V8.4 へ
- Version 8.3 から V8.4 へ

OpenVMS Integrity の場合:

- Version 8.2-1 から V8.4 へ
- Version 8.3 から V8.4 へ
- Version 8.3-1H1 から V8.4 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 8.4 にアップグレードする必要があります。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する必要があります。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された新しいオペレーティング・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 (混成バージョン・クラスタ)。ただし、この場合システム・ディスクが複数必要になります。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 8.4 を含む混成バージョン・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 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、それらのシステムがサポートされているバージョンにアップグレードされるまで、ローリング・アップグレードは実行できません。

これらのバージョンとの混成クラスタをサポートするには、1つあるいは複数の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 4.37.1 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 つまでです。保証サポートが提供されるのは、OpenVMS Integrity Version 8.4 との組み合わせです。詳細は、『HP OpenVMS V8.4 インストール・ガイド[翻訳版]』を参照してください。

OpenVMS オペレーティング・システムで保証サポートが提供される組み合わせ、移行サポートが提供される組み合わせ、あるいは OpenVMS Alpha Version 8.4 のインストール/アップグレード手順、OpenVMS Integrity Version 8.4 のインストール手

順などの詳細は、『HP OpenVMS V8.4 インストレーション・ガイド[翻訳版]』を参照してください。

1.10 OpenVMS Integrity ユーザ向けの注意事項

以下の注意事項は、OpenVMS Integrity システムに適用されます。

1.10.1 ストレージ・コントローラ

V8.3

rx7620 サーバ、rx8620 サーバ、HP Integrity Superdome サーバなどの HP Integrity サーバに対しては、HP sx1000 チップセットで CPU、メモリ、および I/O サブシステムを提供します。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・チップは、I/O デバイスやオフ・セル・メモリへのパスも提供します。

これらのサーバでは、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 Integrity Version 8.3 には、2 つの主要なストレージ・インターコネクがあります。

- SCSI ストレージ・タイプは、直接接続の外部ストレージ A7173A U320 SCSI アダプタの他、特定の Integrity Server のコア I/O に使用される U320 です。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。

SAS ベースのストレージは、OpenVMS Integrity Version 8.3-1H1 以降でサポートされています。

制限事項

以前のバージョンの OpenVMS をご使用の場合は、下記の重要な考慮事項にご注意ください。

- U160 アダプタ (A6829A) は、OpenVMS Integrity Version 8.3 以降では公式にはサポートしていません。また、2005 年にこのアダプタのサポート期間は終了しています。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 アダプタは OpenVMS Integrity Version 8.3 以降ではサポートされません。Version 8.4 上で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。

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

V8.3

rx7620 や rx8620 などの HP Integrity サーバには、コア I/O としてシステムに U160 (SCSI) が内蔵されています。これらの SCSI コントローラから内部 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外部 SCSI ラックに接続することはサポートされていません。

1.10.3 Integrity サーバでの 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 Integrity サーバは自動的に 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。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
```

clearlogs SEL 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 V8.4 インストール・ガイド[翻訳版]』を参照してください。

1.10.4 Integrity サーバ用のファームウェア

V8.3

OpenVMS Integrity Version 8.4 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。

エントリークラスの Integrity サーバに対しては、最新のシステム・ファームウェアを使用することをお勧めします。エントリークラスの Integrity サーバにおけるシステム・ファームウェアのアップデートについては、『HP OpenVMS V8.4 インストールレーション・ガイド[翻訳版]』を参照してください (rx7620, rx8620, あるいは Superdome をご使用の場合は、HP のカスタマーサポートにお問い合わせください)。

表 1-2 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	4.27	4.01	E.03.30	適用外
rx1620	4.27	4.01	E.03.30	適用外
rx2600	2.31	1.53	E.03.32	適用外
rx2620	4.29	4.04	E.03.32	適用外
rx4640	4.29	4.06	E.03.32	1.10
rx2660*	4.11	5.24	F.02.23	適用外
rx3600*	4.11	5.25	F.02.23	1.23
rx6600*	4.11	5.25	F.02.23	1.23

*ご使用の rx2660, rx3600 あるいは rx6600 で Intel Itanium 9100 プロセッサを装備している場合、表 1-2 に示すバージョンよりも、少なくとも 1 バージョン高いファームウェアが必要になります。

セル・ベースのサーバで MP ファームウェアのリビジョン・レベルを確認するには、MP コマンド・メニューにアクセスして sysrev コマンドを実行する必要があります。sysrev コマンドは、MP を持つすべての HP Integrity サーバで利用することができます。セル・ベースの Integrity サーバでは、EFI の info fw コマンドは MP (Management Processor) ファームウェアのリビジョン・レベルを表示しません。

MP を持たないエントリークラスの Integrity サーバでファームウェア・バージョンを確認するには、次の例のように、EFI (Extensible Firmware Interface) プロンプトで info fw コマンドを実行します。以下に例を示します。

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
1.10 OpenVMS Integrity 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

1.10 OpenVMS Integrity 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

EFI へのアクセス方法および使用方法については、『HP OpenVMS V8.4 インストール・ガイド[翻訳版]』を参照してください。詳細は、サーバに付属のハードウェアのドキュメントを参照してください。

エントリークラスの Integrity サーバでファームウェアをアップグレードする方法については、『HP OpenVMS V8.4 インストール・ガイド[翻訳版]』を参照してください。rx7620, rx8620, あるいは Superdome でファームウェアをアップデートする場合は、HP のカスタマーサポートにご相談ください。

1.10.5 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 Integrity サーバで、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-MemmgmtInit 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。

1.10.6 USB または vMedia デバイスからのブート

V8.4

OpenVMS Version 8.4 では%SYSTEM-I-MOUNTVER メッセージと Universal Serial Bus Configuration Manager メッセージが追加されており、USB あるいは vMedia デバイスを使用して Integrity サーバをブートした場合にこれらのメッセージが表示されます。

このメッセージは無視してかまいません。

1.10.7 Small Memory Configurations Error Message

V8.4

OpenVMS Integrity システムをネットワーク経由でブートする際、あるいは Integrity VM のゲスト OS として OpenVMS をブートする際、メイン・メモリからメモリ・ディスクの割り当てを行います。

OpenVMS Version 8.4 では、このメモリ・ディスクのデフォルト・サイズは 256 MB ですが、比較的メモリ・サイズが小さな古いシステムではこれだけのサイズを割り当てることができず、次のようなエラー・メッセージが表示されます。

```
ERROR: Unable to allocate aligned memory
%VMS_LOADER-I-Cannot allocate 256Meg for memory disk. Falling back to 64Meg.
%VMS_LOADER-I-Memorydisk allocated at:0x0000000010000000
```

メッセージが表示された後、OpenVMS は代替手段として 64 MB だけを割り当て、最初のブート時にいくつかの新しいドライバを除外します。代替手段がとられると、それを示すフォールバック・メッセージが表示されます。他のエラー・メッセージを伴わないでフォールバック・メッセージが表示された場合は、最初に表示されたエラー・メッセージは無視できます。

1.10.8 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 Integrity ユーザを対象としています。

1.10.8.1 キーボードのサポート

V8.2

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

1.11 OpenVMS Alpha 向けの注意事項

以下の注意事項は、OpenVMS Alpha ユーザに適用されます。

1.11.1 OpenVMS Alpha Version 8.4 のファームウェア

V8.4

OpenVMS Alpha Version 8.4 は、Alpha Systems Firmware CD Version 7.3 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 8.4 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。

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・ノートを参照してください。V7.3 および最新のファームウェアに関する情報は次のサイトから入手することができます。

<http://h18002.www1.hp.com/alphaserver/firmware/>

1.12 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。

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
1.12 Kerberos for OpenVMS

\$ @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
- [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:

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

1.12 Kerberos for OpenVMS

```
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 Services 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。

1.13 SYSTARTUP_VMS.COM の修正

V8.4

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。暗号化および SSL 製品がインストールされている以前のバージョンの OpenVMS システムからのアップグレードを行う場合は、SYS\$MANAGER: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。

1.14 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 1.15 節を参照してください。

1.15 HP DECram V3.nのアップグレード

V8.2

OpenVMS Alpha Version 8.2 および OpenVMS Integrity 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 V8.4 インストール・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 2.14 節も参照してください。

1.16 LANCP デバイス・データベースの変換

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
```

1.17 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であるため、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 SYSSOUTPUT 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。

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。

1.18 アップグレード前に TIE キットの削除が必要

V8.2-1

Translated Image Environment (TIE) は OpenVMS Integrity 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 Integrity Version 8.2 または Version 8.2-1 に 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 Integrity 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 Integrity Version 8.2-1 以降のシステムにはインストールしないでください。

1.19 代替デバイスや代替ディレクトリへのレイヤード・プロダクトのインストールの失敗

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

2.1.1 HP BASIC for OpenVMS

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テッド・レイヤード・プロダクト CD で入手できます。

2.2 HP TCP/IP Services for OpenVMS

V8.4

OpenVMS Version 8.4 へアップグレードした場合、TCP/IP Services のバージョンは HP TCP/IP Services for OpenVMS Version 5.7 を使用する必要があります。詳細は『HP TCP/IP Services for OpenVMS Version 5.7 Release Notes』を参照してください。

2.3 NetBeans Version 5.5.1 で必要となる最新の JDK

V8.4

OpenVMS Alpha および OpenVMS Integrity 用の NetBeans Version 5.5.1 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。

注意

NetBeans Version 5.5.1 は JDK Version 6.0-1 for Integrity をサポートしません。

2.4 DFS マウントされたディスクへのアクセスの問題

V8.4

DFS 経由で ODS-5 ディスクへアクセスする場合、クライアントからのパス指定に MFD を含むとアクセスが失敗し、次の例に示すようなエラー・メッセージが表示されます。パス指定に MFD を含まない場合は、アクセスは成功します。

次の例は、アクセス・ポイントが DKA100:[USERS] でクライアント・ディスクが DFSC1001 の場合の例です。

```
$ DIR DFSC1001:[000000] ! fails with the error message
%DIRECT-E-OPENIN, error opening DFSC1001:[000000]*.*;*
as input -RMS-E-DNF, directory not found
-SYSTEM-W-NOSUCHFILE, no such file

$ TYPE DFSC1001:[000000]file.dat ! fails with the error message  1
%TYPE-W-SEARCHFAIL, error searching for DFSC1001:[000000]file.dat
-RMS-E-DNF, directory not found
-SYSTEM-W-NOSUCHFILE, no such file

$ DIR DFSC1001:[SUBDIR] ! works as expected  2
```

- 1 file.dat は USERS.DIR の下にあるファイルです。
- 2 SUBDIR は USERS.DIR の下にあるサブディレクトリです。

この問題は、DECdfs V2.4B で解決されます。

2.5 HP DCE for OpenVMS の制限事項 (Integrity のみ)

V8.4

OpenVMS Version 8.4 では、InstallメニューからInstall or Upgrade Layered Productsを選択して HP DCE for OpenVMS Version 3.2 をインストールしようとすると、次のようなエラー・メッセージを表示して処理が失敗します。

```
%PCSI-E-ERROWNER, error in owner specification 'DCE$SERVER'  
-SYSTEM-E-NORIGHTSDB, rights database file not found %PCSI-E-OPFAILED,  
operation failed
```

回避策

DCE をインストールするには、OE DVD インストレーション・ディスクを OpenVMS システムにマウントし、DCE_I64032 ディレクトリにある DCE\$INSTALL.COM プロシージャを実行します。

2.6 XML-C 製品の ZIP ファイル

V8.3-1H1

OpenVMS for Integrity サーバの XML-C 製品は、ZIP ユーティリティでアーカイブされた形態で自己解凍形式の実行可能ファイルが提供されます。XML-C のインストール・マニュアルでは、この実行可能ファイルを使って製品をインストールする方法が説明されています。インストールを開始する前に、ZIP ファイルを解凍して実行可能ファイルを取り出してください。

2.7 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
```

2.8 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 RMSS_FNM が発生しましたが、OpenVMS Alpha Version 7.2 以降のバージョンでは RMSS_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はありません。

2.9 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。

2.9.1 COM for OpenVMS のサポート

V8.4

現在 COM Version 1.4 for OpenVMS は、OpenVMS Alpha Version 7.3-2 以降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/>

2.9.2 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。

2.10 サポートされるバージョンの DECdfs

V8.4

OpenVMS Version 8.4 以降では、DECdfs Version 2.4B 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。

V8.3

OpenVMS Version 8.3 および 8.3-1H1 では、DECdfs Version 2.4A 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。

2.11 DECforms Web Connector Version 3.0 (Alpha のみ)

V7.3-1

DECforms がすでにインストールされている場合、DECforms Web Connector V3.0 を OpenVMS Version 7.3-1 以降で実行するには、次のタスクを実行します。

1. 次の行を削除するか、コメント・アウトします。

```
$ @SYS$COMMON:[JAVA$122.COM] JAVA$122_SETUP.COM
```

上記の行は、FORMS\$INSTALL_AREA ディレクトリの次のコマンド・プロシージャに含まれています。

- FORMS_SMGR_STARTUP.COM
 - FORMS_WEB\$STARTUP.COM
 - FORMS_WEB_CONFIG.COM
2. すべてのプロセスに対して、Java™環境がシステム全体に設定されているか確認します。システムの SYLOGIN.COM ファイルに Java 環境設定を追加することをお勧めします。
 3. ブラウザ・クライアントで Sun Java Plugin Version 1.2.2 が使用されているか確認してください(『SPD』および管理ガイドを参照)。

2.12 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、SYSS\$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 が存在する場合は、それをコピーしておきアップグレード後に復元するか、あるいは後で PL/I キットを再インストールしてください。

第 5.35 節の、関連する注意を参照してください。

2.13 FMS キット

V8.3

次の FMS キットのいずれか (またはより新しいバージョン) を OpenVMS Alpha および OpenVMS Integrity 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

OpenVMS V8.2 以降のシステム (Alpha および Integrity) では FMS V2.5 がサポートされます。

2.14 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 1.15 節を参照してください。

2.14.1 DECram は OpenVMS Version 8.2 以降に含まれる

V8.2

OpenVMS Alpha および OpenVMS Integrity 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、SYSS\$COMMON:[SYSS\$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.SYSS\$LDR]) 内に残っている SYSS\$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時、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  
No older versions of DECram are supported on OpenVMS Version 8.2.  
DECram Version 2.5 will continue to be supported on VAX system only.
```

2.14.2 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"
```

2.15 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。

2.15.1 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)
```

2.15.2 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。

2.16 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 Integrity) 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。

2.17 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。

2.17.1 STARLET ライブラリの作成には HP Pascal 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" とこたえたと、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テッド・レイヤード・プロダクト CD で入手できます。

2.17.2 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。

2.18 I64 システムでの WEBES および SEA のサポート

V8.3

OpenVMS の関連製品に関する注意事項
2.18 I64 システムでの WEBES および SEA のサポート

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/>

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 V8.4 新機能説明書』を参照してください。

3.1 AEST で変換したイメージの問題

V8.4

OpenVMS Version 8.4 では、AEST V3.0 で変換したイメージのいくつかが正しく機能しません。将来のリリースで別途 TIE ECO キットを提供する予定です。

3.2 SYS\$GETTIM_PREC システム・サービス宣言

V8.4

FORTRAN ライブラリ FORSYSDEF.TLB には、SYS\$GETTIM_PREC システム・サービス宣言は存在しません。他のプログラミング言語とは違い、FORSYSDEF.TLB は FORTRAN コンパイラと共に提供され、サポートする最も古いバージョンのオペレーティング・システムに対してビルドされています。将来のリリースでは、サポートする最も古いバージョンのオペレーティング・システムに対してビルドされた通常のライブラリと共に、SYS\$GETTIM_PREC システム・サービス宣言を含んだ OpenVMS Version 8.4 ベースの FORSYSDEF ライブラリを提供する予定です。

3.3 F\$GETSYI("RAD_CPUS") の問題

V8.4

セル・ローカル・メモリ (CLM) とインターリーブ・メモリ (ILM) が構成され、64 CPU を装備したセル・ベースの Integrity サーバでは、F\$GETSYI("RAD_CPUS") の出力には、ベース RAD のメンバとして 64 番目の CPU が含まれません。

この問題は将来のリリースで修正される予定です。

3.4 HP Code Signing Service for OpenVMS

V8.4

OpenVMS Version 8.4 以降では、PCSI および VMSINSTAL 形式のキットは HP Code Signing Service (HPCSS) を使用して署名されています。署名は別ファイルとして提供されます。たとえば、DXDA053.Aには署名ファイルとしてDXDA053.A_HPCが添付されます。OpenVMS V8.4 以降では、オペレーティング・システムと共に提供される HPBINARYCHECKER を使用してキットの正当性確認が行なわれます。詳細は『HP OpenVMS V8.4 新機能説明書』を参照してください。

3.5 SHOW FORWARD/USER でユーザ名の表示が切り捨てられる問題

V8.4

SHOW FORWARD/USER を使用したワールドカード検索では、次の例のようにユーザ名の表示が 31 文字になるよう切り捨てが発生します。

```
MAIL> set forward/user=1234567890123456789012345678901234567890123456789
01234567890123456789012345678901234567890 system
MAIL> sho forward/user=12345678901234567890123456*
Username                               Forwarding address
1234567890123456789012345678901 SYSTEM
```

この問題は将来のリリースで修正されます。

3.6 シンボリック・リンクの実装の変更

V8.4

OpenVMS Version 8.3 で導入されたシンボリック・リンクが (Symlink) は、OpenVMS Version 8.4 で実装が改善されています。

3.6.1 論理名

V8.4

新しい symlink の実装により、論理名をターゲットパス名の最初の要素として使用できます。

以下に例を示します。

```
$ CREATE /SYMLINK="/SYS$HELP/CC_RELEASE_NOTES.PS" RELNOTES.PS
$ DIR /SIZE /NOSYMLINK RELNOTES.PS

Directory SYS$SYSROOT:[SYSMGR]

RELNOTES.PS;1          209

Total of 1 file, 209 blocks.
```

3.6.2 DIRECTORY コマンドで Audit アラームが発生する問題の修正

V8.4

symlink の以前の実装では、SDIR のようなコマンドでディレクトリをリストしようとした際に、指定先のファイルに対するアクセス許可をユーザが持っていない場合、Audit アラームが発生していました。

DIRECTORY コマンドは、ファイルが symlink であるかどうか判断するためにファイル・ヘッダを読み取ります(すなわち、ファイル・アクセスを行います)。このファイル・アクセスにより、ファイルへのアクセス許可を持たないユーザが DIRECTORY コマンドを実行した場合、Audit アラームが発生していました。

この問題は、OpenVMS Version 8.4 の新しい symlink の実装、すなわち、ファイルが symlink の場合にディレクトリ・エントリのフラグ DIR\$V_SPECIAL (以前の値は DIR\$V_NEXTREC) を 1 に設定する、という方法で修正されています。

注意

OpenVMS Version 8.3 で作成したシンボリック・リンクの互換性

OpenVMS Version 8.4 で作成した Symlink は OpenVMS Version 8.3 でも動作します。ただし、OpenVMS Version 8.3 で作成した Symlink は OpenVMS Version 8.4 では動作しません。

OpenVMS Version 8.3 で作成したシンボリック・リンクを OpenVMS Version 8.4 で動作するフォーマットに変換するためには、/REPAIR 修飾子を指定して ANALYZE/DISK_STRUCTURE (VERIFY) ユーティリティを実行します。

3.7 SHOW SYSTEM/STATE=MUTEX がプロセスを表示しない問題

V8.4

\$ SHOW SYSTEM/STATE=MUTEX コマンドは MUTEX 状態のプロセスを表示しません。

しかしながら、次のコマンドを実行することにより MUTEX 状態のプロセスを表示することができます。

```
$ SHOW SYSTEM
```

3.8 HP Secure Web Browser V1.1-12 のインストール時の警告メッセージ

V8.4

SeaMonkey Version 1.0 は Mozilla Version 1.8b1 のコードをベースにビルドされています。すでに SWB Version 1.7-13 がインストールされている OpenVMS サーバに HP Secure Web Browser Version 1.1-12 をインストールした場合、次のような警告メッセージが表示されます。

```
%PCSI-W-VERLOW, you have selected a lower version of an
installed product
-PCSI-W-VERINS, the installation of product HP I64VMS CSWB V1.1-12
-PCSI-W-VERREM, will remove current product HP I64VMS CSWB V1.7-13
Do you want to continue? [YES]
```

これは、PCSI は新たにインストールする製品のバージョンはインストール済みの同じ製品のバージョンよりも常に新しいことを想定していますが、最新の SWB のバージョン番号がシステムにインストール済みのバージョンよりも低いために発生します。

この警告メッセージは無視してかまいません。

3.9 Ctrl/P の動作の問題

恒久的な制限事項

特定の Integrity サーバの構成では、コンソールで Ctrl/P を押しても OpenVMS から IPC (Interrupt Priority C) メニューが表示されません。Ctrl/P を使用する予定がある場合は、この機能が動作するかどうかテストしてください。

必要に応じて、以下の手順を実行することで Ctrl/P の機能を回復させることができます。

1. SDA を実行して動作中のシステムを分析します。

```
$ ANALYZE/SYSTEM
```

2. CLUE CONFIG コマンドを使用して、システム上のアダプタを表示します。

```
SDA> CLUE CONFIG/ADAPTER
```

3. 表示の中で "Console Serial Line Driver"アダプタ (SRA:) を探します。

System Adapter Configuration:

```
-----  
TR Adapter      ADP              Hose Bus   BusArrayEntry  Node GSIN  iVec  SCB  
-----  
...  
5 ACPI_IA64_I  FFFFFFFF.8832E0C0  0  00 IA64_BUS  
6 PCI          FFFFFFFF.88342A80  9  00 PCI  
                FFFFFFFF.88342E58  0  0018 00DF 15F0  
                FFFFFFFF.88342F68  8  0018 00DF 15F0  
...  
Port Slot Device Name/HW-Id  
-----
```

```
SRA: 0 Console Serial Line Driver  
EWA: 1 A6865A (Fast Ethernet)
```

4. SRA: と同じ GSIN (Global System Interrupt Number) を共有しているコントローラを特定します。この例では EWA: が該当します。
5. SDA を終了し、次のコマンドを入力します (EWA は正しいコントローラで置き換えてください)。

```
$ SET DEVICE EWA0/PREFERRED_CPUS='F$GETSYI("PRIMARY_CPUID")'
```

この手順が完了すると、Ctrl/P が正しく機能するようになります。システムがリブートしたときに正しく動作するように、SYSSMANAGER:SYLOGICALS.COM を編集して、SET DEVICE コマンドを追加することをお勧めします。I/O アダプタを追加または削除すると、ふたたび Ctrl/P が動作しなくなる可能性があります。その場合は、上記の手順を再度実行してください。

システムをブートした時に XDELTA または System Code Debugger がロードされている場合、Ctrl/P は影響を受けません。Ctrl/P を入力すると、次の例のように XDELTA のプロンプトが表示されます。

```
Console Brk at 807CF3D2 on CPU 0  
807CF3D2!          cmp4.lt      p0, p6 = 3F, r4 (New IPL = 3)
```

3.10 シリアル・ポートの名前

V8.3-1H1

OpenVMS での列挙は、OpenVMS の汎用デバイス・タイプ命名規則に従って、デバイスに英字と番号を割り当てる作業です。シリアル・ポートの場合には、この列挙は TTA0, TT0, ... のようになり、汎用シリアル・ポート・デバイスの場合には、EFI Boot Manager または EFI シェル・プロンプトでシステムのプライマリ・コンソールとして選択したシリアル・ポート・デバイスに対して、OPA0 となります。

OpenVMS V8.2 では、Alpha ベース・システムの OpenVMS で確立された規則と前例に従って、システム・シリアル・ポートを列挙してきました。OpenVMS V8.3 では、これらの規則が破棄され、エンド・ユーザから見て一貫性のないポート命名規則が使われるようになりました。特に、V8.2 から V8.3 へ移行する場合に顕著です。

OpenVMS V8.3-1H1 では、HP Integrity の OpenVMS V8.2 で確立された一貫性のあるシリアル・ポート命名規則に戻りました。シリアル・ポートの名前を不必要に変更しないためと、OpenVMS Alpha システムのポリシーに合わせるためです。シリアル・ポートは複数の機能をサービスする等の理由から、シリアル・ポートの名前は変わる可能性があります。

プライマリ・コンソールとして選択したシリアル・ポートは、必ず OPA0 になります。グラフィック・コンソールをプライマリとして選択した場合には、キーボードとグラフィックス・ディスプレイで OPA0 が構成され、シリアル・ポートには TTA0、TTB0、... のような名前が割り当てられます。

Integrated Lights Out (iLO) Management Processor (MP) のシリアル・ポートは、プライマリ・コンソールとして選択しない限り、シリアル・ポートとして接続されず、オペレーティング・システムからは表示されません。これは汎用的な使用には適していません。汎用シリアル・ポートに必要なデータ・レートのサポートができないためです。これは大部分のシステムで、オプションのコンポーネントです。システムに付属しているオプションのリスト、または次の Web サイトにあるシステム・マニュアルを確認してください。

<http://docs.hp.com>

プライマリ・コンソールとして選択できるシリアル・ポートには、iLO と Baseboard Management Console (BMC) の 2 つがあります。どちらをプライマリ・コンソールとして選択しても、OpenVMS では OPA0 と表示されます。そしてシステムに他のシリアル・ポートがある場合には、TTA0 または TTB0 と表示されます。次の表に、略語とその定義を示します。

略語	定義
MP	iLO MP のシリアル・ポート。このコンポーネントはシステムによってはオプションです。
BMC	BMC のシリアル・ポート。このコンポーネントはすべてのシステムが備えているわけではありません。
AP	補助ポート。16550 互換の補助シリアル・ポート。すべてのシステムが備えているわけではありません。
VGA	グラフィック・コンソール。iLO-MP のオプションのコンポーネントです。システムに VGA オプションが付属していない場合には、PCI スロットのいずれかにグラフィックス・オプションをインストールすることで、この機能が利用できます。
NA	該当なし。
NC	OpenVMS ではシリアル・ポートとして構成されません。

略語	定義
NS	サポートされていません。

次の表に、この節でふれた背面パネルの図の参照先を示します。

プラットフォーム	背面パネルの図
rx1600 , rx1620	http://docs.hp.com/en/AB430-96004/ch03s03.html#i1021437
rx2600 , rx2620	http://docs.hp.com/en/AD117-9003A-ed2/AD117-9003A-ed2.pdf
rx4640	http://docs.hp.com/en/A9950-96009/A9950-96009.pdf
rx3600 , rx6600	http://docs.hp.com/en/5991-8053-ed9/5991-8053-ed9.pdf
rx2660	http://docs.hp.com/en/5991-8053-ed9/5991-8053-ed9.pdf
rx8620	http://docs.hp.com/en/A7026-96037-en/A7026-96037-en.pdf
bl860c	http://docs.hp.com/en/5991-8053-ed9/5991-8053-ed9.pdf
bl870c	http://docs.hp.com/en/5991-8053-ed9/index.html

次の表に、上記の HP Integrity プラットフォームでのシリアル・ポート名を示します。プライマリ・コンソールとして選択したデバイスには、必ず OPA0 という名前が付きます。

プラットフォーム	プライマリ・コンソール・ポート	MP	BMC	AP	VGA
rx1600	MP (オプション)	OPA0	TTA0	NA	OPA0
rx1620	BMC	NC	OPA0		
	VGA (オプション)	NC	TTA0		
rx2600	MP (オプション)	OPA0	TTB0	TTA0	OPA0
rx2620	BMC	NC	OPA0	TTA0	
	VGA (オプション)	NC	TTB0	TTA0	
rx4640	MP	OPA0	NA	NA	OPA0
	VGA (オプション)	NC			
rx3600	MP (オプション)	OPA0	TTA0	OPA0	
rx6600	BMC	NC	OPA0		
	VGA (オプション)	NC	TTA0		
rx2660	MP	OPA0	NA	TTA0	OPA0
	VGA (オプション)	NC		TTA0	
rx8620	MP	OPA0	NA	NA	OPA0
	VGA	NC			
BL860c	MP	OPA0	NA	NA	OPA0
	VGA	NC			
BL870c	MP	OPA0	NA	NA	OPA0
	VGA	NC			

3.11 古いファームウェアではVMS V8.3-1H1 がシステム・イベント・ログに書き込むメッセージを変換できない

V8.3-1H1

一般ユーザ向けの注意事項

3.11 古いファームウェアでは VMS V8.3-1H1 がシステム・イベント・ログに書き込むメッセージを変換できない

インストール時に、V8.3-1H1 は新しいメッセージのシステム・イベント・ログへの書き込みを開始します。システム・イベント・ログを表示するには、(大部分のシステムで)メイン MP メニューからシステム・イベント・ログの表示を選択します (SL: Show Event Logs)。

古いファームウェアでは、メッセージが、正しい "OS_OPENVMS_BUGCHECK" と "OS_OPENVMS_SHUTDOWN" の代わりに、"IPMI Type-E0 Event" と変換されます。

以下に、古いファームウェアが動作しているシステムでの OS_OPENVMS_BUGCHECK メッセージ (警報レベル*5 - クリティカル) の例を示します。

```
291      0 *5 0xB4801C9700E01B50 000000000019000C IPMI Type-E0 Event
                                         30 Jul 2007 14:03:41
```

以下に、古いファームウェアが動作しているシステムでの OS_OPENVMS_SHUTDOWN メッセージ (警報レベル 2 - 情報) の例を示します。

```
296      0 2 0x54801C9900E01BD0 00000000001A000C IPMI Type-E0 Event
                                         30 Jul 2007 14:22:06
```

新しいファームウェアでは、"IPMI Type-E0 Event" の代わりに、"OS_OPENVMS_BUGCHECK" または "OS_OPENVMS_SHUTDOWN" が使われます。

3 番目のメッセージ "OS_BOOT_COMPLETE" は、新しいファームウェアが動作しているシステムでは、異なる警報レベルになります。これは、OpenVMS により、「情報」または「レベル 2」に変更されました。

```
301  OS  0 2 0x548016E100E01B80 0000000000000001 OS_BOOT_COMPLETE
                                         23 Aug 2007 14:25:44
```

新しいファームウェアでは、"T - View Mode Configuration Text" を選択すると、以下のメッセージが表示されます。

```
MP:SL (+,-,CR,D, F, L, J, H, K, T, A, U, ? for Help, Q or Ctrl-B to Quit) >t .
```

```
.  
.
```

```
Log Entry 301: 23 Aug 2007 14:25:44
Alert Level 2: Informational
Keyword: OS_BOOT_COMPLETE
OS Boot Complete
Logged by: O/S Kernel (Generic) 0
Data: Major change in system state - Boot Complete 0x548016E100E01B80 0000000000000001
```

3.12 CRTL 内の TZ 関数

V8.3-1H1

論理名 TZ や DCL シンボル TZ は、特定の C プログラムの時刻関連の関数で使われるタイムゾーンを定義するために、C ランタイム・ライブラリ (C RTL) で使われます。(TZ の完全な説明、その使用法、具体的な関数については、C ランタイム・ライブラリのマニュアルの tzset() 関数の説明を参照してください。)

論理名 TZ や DCL シンボル TZ は、OpenVMS Version 7.3 から、C ランタイム・ライブラリで使われています。ただし、Version 8.3 で一部変更されました。

Version 8.3 より前では、TZ にタイムゾーンとして無効な値を定義すると、タイムゾーンには省略時の値であるローカルタイム (すなわち、システムの現在のタイムゾーン) が設定されました。OpenVMS 8.3 リリースからは、TZ に無効なタイムゾーンを定義すると、C ランタイム・ライブラリの時刻関数は、UTC 時刻を使用するようになりました。

論理名 TZ や DCL シンボル TZ にタイムゾーンとして無効な文字列を定義すると、C プログラムの実行時に予期しない動作が発生する可能性があることに注意してください。

3.13 InfoServer ユーティリティと FDDI

V8.3-1H1

OpenVMS の InfoServer ユーティリティを使って FDDI ネットワーク・アダプタ経由でクライアントをブートする機能はサポートされていません。

3.14 DCL コマンドの SET PASSWORD の新しい修飾子

V8.3-1H1

DCL コマンドの SET PASSWORD は、/PROMPT 修飾子で、2 つの値 /PROMPT=FIXED と /PROMPT=VARIABLE を受け付けるようになりました。DCL コマンド・プロシージャで SET PASSWORD コマンドを使う場合には、/PROMPT=VARIABLE 修飾子は使わないでください。使った場合には動作は正常に行われますが、障害ステータスが出た場合は、表示されるだけで DCL には返されません。

3.15 OpenVMS Freeware

V8.4

OpenVMS Freeware コレクションは、OpenVMS ユーザにさまざまなパブリック・ドメイン・ソフトウェア、オープンソース・ソフトウェア、フリーウェア・パッケージ、および HP が開発したソフトウェアおよびツールを提供します。

以前のバージョンの OpenVMS では、フリーウェア・コレクションは CD で提供していましたが、OpenVMS Version 8.4 以降は下記の URL の OpenVMS Freeware Web サイトで提供します。

<http://h71000.www7.hp.com/openvms/freeware/index.html>

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:[FREEWARE]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 [FREEWARE] 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]FREWARE.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。

3.16 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。

3.16.1 OpenVMS グラフィック・コンソールでの SHUTDOWN.COM(Integrity のみ)

恒久的な制限事項

グラフィック・コンソール付きの OpenVMS Integrity サーバ・システムでは、SYSSYSTEM:SHUTDOWN.COM によるシステムの停止は期待どおりに動作しません。"SYSTEM SHUTDOWN COMPLETE"メッセージが表示された後システムは停止せず、コンソール・キーボードでキーがタイプされるのを待ちます。しかし、まるでリブートが要求されたかのようにリセットし続けます。ブートオプション・リストの最初のオプションが有効なブート・デバイスの場合は、OpenVMS システムがリブートされます。

3.16.2 MOUNT コマンドの制限事項

V8.4

異なるバージョンのメンバ・システムで構成される OpenVMS クラスタでは、OpenVMS Version 8.4 システムからの/CLUSTER および/CACHE=[NO]DATA によるボリュームのマウントが、以前のバージョンの OpenVMS で MOUNT-F-BADPARAM のエラー状態で失敗します (%MOUNT-W-RMTMNTFAIL)。

ボリュームをマウントして状態で XFC を有効/無効にする方法については、『HP OpenVMS V8.4 新機能説明書』を参照してください。

3.16.3 SHOW LICENSE/CHARGE_TABLE で OpenVMS ゲスト・システムのソケット数を認識しない問題

V8.4

SHOW LICENSE/CHARGE_TABLE あるいは SHOW LICENSE/UNIT_REQUIREMENTS コマンドは、次の例のように Integrity VM 上の OpenVMS ゲストに対してプロセッサ・ソケット数をゼロと表示します。この問題は、OpenVMS ゲストからホスト・システムのソケット数を認識するための適切なインタフェースが欠けているために発生するものです。

一般ユーザ向けの注意事項

3.16 DCL コマンド

```
$ SHOW LICENSE/CHARGE_TABLE
OpenVMS I64/LMF Charge Information for node PERFORM
This is an _HP_VMM(1.42GHz/6.0MB), with 8 cores active
This platform supports up to 0 processor socket(s)
Type: PPL, Units Required: 8 (I64 Per Processor)
Type: PCL, Units Required: 8 (I64 Per Core)
```

この制限事項は将来のリリースで修正されます。

3.16.4 DIAGNOSE コマンドはサポートされない

V8.2

DIAGNOSE コマンドは、OpenVMS Version 8.2 ではサポートされません。

3.17 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。このソフトウェア・キットは OpenVMS Version 7.3-2 用のメディアに含まれていました。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>

3.18 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。

3.18.1 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 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。

3.18.2 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 (Integrity のみ)

V8.2

OpenVMS Integrity 用 HP SWB 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。

3.19 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。

3.19.1 『HP OpenVMS Linker Utility Manual』の訂正

V8.4

3.19.1.1 HP C++ Examples

2.6.2 項の 7 番目のリスト項目に次のパラグラフを追加すべきです。

Note that on Integrity servers, you can use either the CXXLINK command or invoke the OpenVMS Linker to combine your object modules into one executable image. On OpenVMS Alpha, you must use the CXXLINK utility to link the object modules into one executable image. On Integrity server systems, the only benefit of using CXXLINK is that CXXLINK reports non-mangled names of undefined multiply-defined. It does this by intercepting Linker diagnostics and converting mangled names reported by the Linker to their original names, using the information in the demangler database.

3.19.2 『HP PCSI Utility Online help and Manual』：\$PRODUCT REGISTER VOLUME の構文エラーの訂正

V8.4

HP PCSI ユーティリティのオンライン・ヘルプは\$PRODUCT REGISTER VOLUME コマンドの構文を次のように間違っで定義しています。

```
$PRODUCT REGISTER VOLUME old-volume-label device-name
```

正しい構文は次のようになります。

```
$PRODUCT REGISTER VOLUME old-logvolnam device-name
```

3.19.3 『iCAP Release Notes』：GiCAP の機能は現在使用できない

V8.3-1H1

SYSSMANAGER:ICAP\$CONFIG.COM を実行している際に、"Enter (Y)es to configure this system with GiCAP support (N):"プロンプトに対して、"Y"と答えると、以下のメッセージが表示されます。

```
HP OpenVMS Industry Standard 64
Global Instant Capacity on Demand (GiCAP) configuration utility
*** GiCAP functionality is not currently available ***
*** GiCAP will be enabled at a later date via an ECO kit ***
```

また、iCAP (Instant Capacity) のリリース・ノートにも、OpenVMS Integrity Version 8.3-1H1 での GiCAP サポートについての説明があります。この機能は現在は提供されていませんが、将来のアップデート・キットで提供される予定です。詳細は、OpenVMS の Web サイトを参照してください。

3.19.4 『POLYCENTER Software Installation Utility Developer's Guide』：PRODUCT コマンドの訂正

V8.4

パラメータ・セクションの *producer* の正しい説明は次のとおりです。

ソフトウェア製品の法的所有権者を示します。このパラメータでは、文字列をダブル・クォートで囲っても囲わなくても構いません。

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

V8.3

V8.2 版の『HP OpenVMS RTL Library (LIB\$) Manual』では、LIB\$SET_SYMBOL の文字列値が間違っていて記述されています。正しい値は以下のとおりです。

使用前に、文字列値から後続のブランクが取り除かれることはありません。文字列値の最大値は 4096 文字です。整数値は使用できません。LIB\$SET_SYMBOL は、整数の CLI シンボルではなく文字列の CLI シンボルを設定するためのものです。

3.19.6 ドキュメントの訂正: LCKMGR_CPUID システム・パラメータ

V8.3

『OpenVMS Performance Management』では、システム・パラメータ LCKMGR_CPUID へのいくつかの参照が LOCKMGR_CPU のように記述されています。後者のパラメータへの参照は正しくなく、これらはこのドキュメントの次回の改訂時に修正されます。

3.19.7 MMG_CTLFLAGS: ドキュメントの訂正

V8.2

『OpenVMS Performance Management』では、MMG_CTLFLAGS システム・パラメータのビット 1 の説明に間違いがあります。正しい説明は以下のとおりです。

"Reclamation enabled by out swapping processes that have been idle for longer than LONGWAIT seconds. This occurs when the size of the free list drops below the value of FREEGOAL."

3.19.8 『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。

3.19.9 『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。

3.19.9.1 システム・ダンプの保存

V8.3

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 がいないかぎり、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。

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

V8.3

HP DELTA デバッガが OpenVMS Integrity 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 Integrity システムで DELTA を利用するための情報が追加されました。

3.19.11 『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 Integrity Librarian ユーティリティに関する訂正情報です。

3.19.11.1 /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。

Integrity 版 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。

3.19.11.2 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・ポイントは 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 ;
```

一般ユーザ向けの注意事項
3.19 ドキュメントの訂正

```
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。

3.19.12 『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。

3.19.12.1 『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、スケール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。

3.19.13 『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 (LIBS) Manual』では、以下のルーチンが Alpha と Integrity サーバ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 (LIBS) 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。

3.19.14 『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。

3.20 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・プロセスが完了しない可能性があります。

3.21 同期データ・リンクの非サポート

OpenVMS は、Integrity サーバでは同期データ・リンク・ハードウェアをサポートしません。

3.22 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 回のみです。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 (SYS\$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。

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 V8.4 新機能説明書』を参照してください。

4.1 SYS\$TIMEZONE_RULE 論理名がハイフン (-) をカレット (^) に置換

V8.4

OpenVMS Version 8.3-1H1 以降、ハイフン文字 (-) をカレット文字 (^) に置換するように SYS\$TIMEZONE_RULE 論理名が修正されています。この変更は、DTSS をサポートするために TDF で行なわれています。DTSS は一般的に使用される UNIX の GMT-X タイムゾーン・ルールを処理することができず、タイムゾーン名と同一タイムゾーン・ルール文字列をサポートしていません。

たとえば、GMT-1 タイムゾーン・ルールは、SYS\$TIMEZONE_RULE 文字列 GMT-1 を生成します。しかし、ルール・ファイル名とルール文字列がともに GMT-1 であるため DTSS は適切に機能しませんでした。

この変更をサポートするために、CRTL および DTSS コンポーネントも変更されます。

たとえば、この変更が行なわれる前のタイムゾーン論理名は以下のとおりでした。

```
"SYS$TIMEZONE_RULE" = "CET-1CEST-2,M3.5.0/02,M10.4.0/03"
```

変更の結果、タイムゾーン論理名は以下のようになります。

```
"SYS$TIMEZONE_RULE" = "CET^1CEST^2,M3.5.0/02,M10.4.0/03"
```

4.2 Virtual オプション付きでのライセンス処理

V8.4

OpenVMS Version 8.4 では、Virtual オプション付きのライセンス・ロードはゲスト・クラスタ・メンバに限定されています。非ゲストの OpenVMS Version 8.4 システムでは、Virtual オプション付きでライセンスをロードできません。

V8.4 よりも古いバージョンの OpenVMS システムがインストールされている OpenVMS クラスタ・メンバでは、Virtual オプション付きでライセンスがロードされます。Virtual オプション付きでロードしてもゲスト・システムにおける動作には悪影響を与えませんが、必要に応じて、ロード処理が行なわれないように/INCLUDE あるいは/EXCLUDE を使用することをお勧めします。

Integrity VM における OpenVMS ゲストのライセンスについての詳細は、『HP OpenVMS License Management Utility Manual』を参照してください。

4.3 iSCSI デモ・キット

V8.4

OpenVMS Version 8.4 では、iSCSI デモ・キットはサポートされません。このキットは OpenVMS Version 8.4 では使用しないことをお勧めします。

4.4 Integrity VM における OpenVMS ゲスト OS

OpenVMS Version 8.4 は HP Virtualization をサポートするため、HP Integrity Virtual Machines (Integrity VM) 上にゲスト・オペレーティング・システムとしてインストールすることができます。Integrity VM 固有の制限事項については、Integrity VM 製品のドキュメントを参照してください。

この項では、Integrity VM 上の OpenVMS ゲストに関する既知の問題と制限事項について説明します。

4.4.1 シャットダウン動作の変更

V8.4

SYSS\$SYSTEM:SHUTDOWN.COM をリポート指定無しで実行した場合、システムは POWER_OFF オプションを使用します。ゲスト・ノードがクラスタ環境にある場合、POWER_OFF オプションとともに REMOVE_NODE オプションを使用してクォーラムが調整されます。

このオプションが使用される結果として、仮想マシンがシャットダウンされます。このため仮想コンソールで MP コマンド `pc -on` を使用するか、あるいはホストで次のコマンドを実行して、仮想マシンを再起動する必要があります。

```
# hpvmstart -P <<OpenVMS guest name>>
```

4.4.2 OpenVMS ゲストによる外付け I/O デバイスのサポートに関する注意事項

V8.4

OpenVMS ゲストは、CD/DVD ドライブ、メディア・チェンジャー、テープ・デバイスなどの付属装置はサポートしません。テープ・デバイスを使用したい場合は、クラスタ内のシステムに物理的に接続されたデバイスに TMSCP を利用してアクセスします。

4.4.3 ネットワークおよびストレージ・インタフェースのサポート

V8.4

OpenVMS ゲストは AVIO (Accelerated Virtual) I/O インタフェースのみをサポートします。

Integrity VM のコマンドをゲストに対して使用して VIO デバイスを構成してもこれらのデバイスはスタートアップ時に特に明確なエラーは示しませんが、VIO デバイスは OpenVMS が稼動するゲストのサポート構成の範囲からは外れます。

4.4.4 HP-UX ゲストと OpenVMS ゲストでの同じ仮想スイッチの使用に関する制限事項

V8.4

HP-UX ゲストと OpenVMS ゲストで同じ仮想スイッチを使用するように構成すると、これらのゲスト間のネットワーク通信に失敗します。この問題は OpenVMS の将来のリリースで解決されます。

この問題の回避方法は、HP-UX ゲストと OpenVMS ゲストで異なる仮想スイッチを使用するように構成することです。

4.4.5 vNICs が構成されていない場合の OpenVMS ゲストの既知の問題

V8.4

vNIC (Virtual Network Interface Card) が構成されていない OpenVMS ゲストで DECnet スタートアップの後で TCP/IP が起動された場合、クラッシュが発生します。vNIC が少なくとも 1 つ構成された状態で OpenVMS ゲストを使用することをお勧めします。

vNIC が構成されていない場合に、OpenVMS ゲスト上で DECnet あるいは TCP/IP を個々に動作させることは可能です。

4.5 HP Availability Manager に関する注意事項

V8.4

この節では、HP Availability Manager Version 3.1 の既知の問題について説明します。

- SYSS\$STARTUP:AMDS\$STARTUP STOP コマンドで Data Collector を停止し、SYSS\$STARTUP:AMDS\$STARTUP START コマンドで再起動した場合、OpenVMS Alpha および OpenVMS Integrity サーバでは、Data Analyzer のイベント・ウィンドウには管理しているノードに関してイベントが何もポストされません。

回避方法:

次のいずれかのコマンドを実行します。

- 次のコマンドで Data Collector を再起動します。

```
$ @SYSS$STARTUP:AMDS$STARTUP RESTART
```

- 次のコマンドで Availability Manager Server を再起動します。

```
$ AVAIL/SERVER
```

- 次のコマンドで Availability Manager Analyzer を再起動します。

```
$ AVAIL/ANALYZER
```

- Availability Manager Analyzer がすべてのリモート・ノードに対して "Path Lost" PATHLST イベントを報告した後、しばらく経過すると、データの表示が行われなくなります。

回避方法:

この問題を解決するには、次のようなコマンドを入力して、SYSGEN パラメータの LAN_FLAGS ビットを 16 に変更して通常の動作に戻します。

```
$ MC SYSGEN SET LAN_FLAGS 16
```

4.6 HP SIM による OpenVMS のプロビジョニング

HP SIM Version 4.0 による OpenVMS のプロビジョニングには、以下の制限事項が適用されます。

4.6.1 OpenVMS ゲストのプロビジョニングに関する制限

V8.4

Integrity VM における OpenVMS ゲスト・オペレーティング・システムのプロビジョニングはサポートされません。

4.6.2 システム・ファームウェアに関する注意事項

V8.4

BL860c および BL870c サーバのシステム・ファームウェアのバージョンは 4.21 以上でなければなりません。 rx3600 および rx6600 サーバのシステム・ファームウェアのバージョンは 4.11 以上でなければなりません。

4.6.3 複数のサーバのプロビジョニング

V8.4

- InfoServer による HP SIM プロビジョニングでは、同時に最大 8 台のサーバのプロビジョニングをサポートします。
- vMedia による HP SIM プロビジョニングでは、一度に 1 台のサーバしかプロビジョニングできません。

4.6.4 HP SIM Central Management Server からのプロビジョニング

V8.4

OpenVMS は、HP SIM Central Management Station、すなわち Microsoft Windows がインストールされた HP ProLiant サーバからプロビジョニングできます。

4.6.5 InfoServer の名前の長さ

V8.3-1H1

プロビジョニングが正しく行なわれるためには InfoServer の名前は 12 文字よりも短くなければなりません。これは一時的な制限事項です。

4.6.6 OpenVMS InfoServer と Integrity サーバが同一 LAN 上に存在しなければならないケース

V8.3-1H1

サーバ・ブレードのプロビジョニングでは、OpenVMS InfoServer と Integrity サーバは同一 LAN 上に存在する必要があります。

4.6.7 EFI ファームウェア

V8.3-1H1

BladeSystem の EFI ファームウェアは V5.0 以上でなければなりません。

4.6.8 Management Processor

V8.4

Management Processor は Advanced iLO2 ファームウェアを実行していなければなりません。

4.6.9 プロビジョニングによる OpenVMS TCP/IP の構成に関する既知の問題

V8.4

ターゲット・サーバで TCP/IP のサーバ・コンポーネント BIND , LPD , LBROKER および SMTP を有効にするように選択している場合、プロビジョニングによって OpenVMS TCP/IP を構成する際にはこれらを起動しないでください。

プロビジョニングによる TCP/IP の構成の後、手動でこれらのサービスを構成および再起動することでこの問題に対処してください。

4.6.10 OpenVMS TCP/IP のプロビジョニングに関する制限事項

V8.4

プロビジョニングによる OpenVMS TCP/IP の構成には以下のような制限事項があります。

- TCP/IP の構成では IPv4 アドレスのみをサポートしています。IPv6 アドレスは現在のところサポートしていません。
- エイリアスあるいはセカンダリ IP アドレスの構成はサポートしていません。
- ターゲット・サーバにおける DHCP サーバ・コンポーネントの構成はサポートしていません。

- プロビジョニングでは、ネットワーク・インタフェースの構成は各ターゲット・サーバで1つだけ可能です。
- HP TCP/IP Services for OpenVMS のオプション・コンポーネントの構成はサポートしていません。
- プロビジョニングでは、論理 LAN デバイスの設定および LAN フェールオーバーの構成はサポートしていません。

4.6.11 AutoBoot タイムアウト値の設定

V8.4

OpenVMS のプロビジョニングを行う場合、各ターゲット・サーバの AutoBoot Timeout 値は 5 秒以上に設定されていなければなりません。このパラメータの設定は、EFI Boot Manager メニューから実行できます。(Boot Configuration -> AutoBoot Configuration -> Set AutoBoot Timeout)

4.7 Insight ソフトウェアによる OpenVMS の管理

V8.4

Insight ソフトウェアについての詳細は以下の Web サイトを参照してください。

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

4.8 性能の強化

V8.4

OpenVMS V8.4 では、以下のような性能の強化が行なわれています。

4.8.1 書き込みビットマップの拡張

V8.4

書き込みビットマップ (WBM) は、ミニマージおよびミニコピー操作の際に OpenVMS Volume Shadowing が使用する機能です。ディスク上のどのブロックに書き込みが行なわれたかを、クラスタ内の他のノードに伝えます。本リリースでは以下のような変更が行なわれています。

4.8.1.1 WBM_MSG_INT パラメータの修正

V8.4

WBM_MSG_INT パラメータは、バッファード・モードのときに SetBit メッセージをどれだけ遅らせることができるか、その時間を示します。この時間間隔で SetBit バッファが SetBit メッセージでいっぱいになっていなければ、メッセージが送信されません。このパラメータ値の単位はミリ秒ですが、このタイマーが使用する換算係数が 1 桁間違っていました。以前は、WBM_MSG_INT の値が 10 の場合、バッファード・モードで 100 ミリ秒の遅れという結果になっていました。本リリースでは、値 10 が正しく 10 ミリ秒の遅れを示すように修正されています。

4.8.1.2 WBM_MSG_UPPER および WBM_MSG_LOWER パラメータの修正

V8.4

WBM_MSG_UPPER は、シングル・メッセージ・モードで動作している場合に、バッファード・メッセージ・モードに切り換えるかどうかを判断するためのしきい値です。ある 100 ミリ秒の間に WBM_MSG_UPPER を越える SetBit 操作が行なわれた場合、メッセージング・モードはバッファード・モードに切り替わります。WBM_MSG_UPPER のデフォルト値は 80 です。

WBM_MSG_LOWER は、バッファード・メッセージ・モードで動作している場合に、シングル・メッセージ・モードに切り換えるかどうかを判断するためのしきい値です。ある 100 ミリ秒の間に WBM_MSG_LOWER を下回る SetBit 操作が行なわれた場合、メッセージング・モードはシングル・モードに切り替わります。デフォルト値は 20 です。

4.8.1.3 非同期 SetBit メッセージ

V8.4

1 つのシャドウセットに対して複数のマスタ・ビットマップ・ノードを持つことが可能ですが、これまでは、SetBit メッセージは複数のマスタ・ビットマップ・ノードへ同期的に送信されていました。最初のリモート・マスタ・ビットマップ・ノードから SetBit メッセージに対する応答を受け取ると、そのメッセージは次のマスタ・ビットマップ・ノードへ送信されていました。そして、すべてのリモート・マスタ・ビットマップ・ノードからの応答を受け取ると、I/O が再開されていました。

本バージョンでは、SetBit メッセージはすべてのマスタ・ビットマップ・ノードへ非同期的に送信されます。全てのマスタ・ビットマップ・ノードからの応答を受け取ると、I/O 操作が再開されます。これにより、書き込みビットマップ・コードによる I/O 操作の待機時間が短縮されます。

4.8.1.4 シーケンシャル I/O に対する SetBit メッセージの削減

V8.4

ディスクに対するシーケンシャルな書き込みが発生すると、リモート・ビットマップのシーケンシャルなビットを設定する Setbit メッセージの送信が行なわれます。WBM のコードは、ビットマップのどの場所に多くの前のビットが設定されているかを認識するようになりました。これにより、シーケンシャルな書き込みが続く場合により少ない Setbit メッセージで済むように WBM は追加ビットを設定します。シーケンシャルな I/O が続くと想定すると、Setbit メッセージが 10 分の 1 ほどに低減され、シーケンシャル書き込みの I/O レートが改善されます。

4.8.2 例外処理性能の改善 (Integrity のみ)

V8.4

OpenVMS Integrity サーバ・システムの例外処理に対していくつかの性能改善が行なわれています。この変更により例外処理のオーバーヘッドが軽減されますが、例外処理のすべてのケースで性能改善が見られるわけではありません。

OpenVMS Version 8.4 は、デコードされたアンwind・データをキャッシュします。このキャッシュは、ユーザ呼び出しが可能な呼び出し標準ルーチンで例外処理の際に使用されます。これらの呼び出し標準ルーチンは、C++ の try/throw/catch コンストラクトや C 言語の setjmp/longjmp コンストラクトなどのプログラミング言語コンストラクトを実装するために RTL でも使用されています。

予想外のエラーの場合、VMS システム・パラメータ KTK_D3 を使用して、キャッシュを一時的に無効にすることができます。デフォルト値の 0 でキャッシュは有効になります。値 1 でキャッシュが無効になります。この特別なパラメータ KTK_D3 は、HP が提供するデバッグ/テスト・イメージで使用されます。この種のテスト・イメージをシステムでご使用の場合は、デフォルト値の 0 にリセットするようにしてください。

4.8.3 イメージの起動 (Integrity のみ)

V8.4

イメージを起動し実行中、ページング I/O はそのイメージのページをメモリに保管します。Integrity サーバ・システムでは、実行されるコードがページに含まれている場合、それらのページで I キャッシュ・フラッシュを実行する必要があります。このため、実際には実行されない多くのページで I キャッシュ・フラッシュが発生することになります。実行されないページでの I キャッシュ・フラッシュを避けるため、本リリースでは、そのページで最初にインストラクションが実行される時に I キャッシュ・フラッシュが実行されるように変更されています。この変更により、実行

されないページでの I キャッシュ・フラッシュは行なわれず、全体的な性能の向上が期待できます。

4.8.4 グローバル・セクションの作成と削除

V8.4

種々のタイプのグローバル・セクションの作成および削除の領域でオペレーティングシステムの性能が強化されています。この変更による利点として、大規模な SMP システムにおける MP Synch の削減が期待できます。

4.8.5 Dedicated CPU Lock Manager

V8.4

Dedicated CPU Lock Manager は、16 個以上の CPU を持ちロッキング・レートが非常に高いシステムにおいて使用される機能です。より高いレートでロッキング操作が実行されるよう Dedicated CPU Lock Manager が改善されています。

4.8.6 Ctrl/T アライメント・フォルト

V8.4

ターミナルで Ctrl/T 操作を行うと、多数のアライメント・フォルトが発生しました。この問題は OpenVMS Version 8.4 では解決されています。

4.9 ACPI ブート時のエラー・メッセージおよび警告メッセージ

V8.4

セルベースのシステム (rx8640 あるいは rx7640 など) では、OpenVMS のブート時に次のようなメッセージが表示される場合があります。

```
ACPI Error (utmutex-0430): Mutex [1] is not acquired, cannot release [20071219]
```

一部のシステム (最新のプロセッサを搭載した rx2660 など) では、電源管理機能が有効になっていると OpenVMS のブート時に次のようなメッセージが表示される場合があります。

```
ACPI Warning (nseval-0250): Excess arguments - method [_OST] needs 3, found 7 [20080701]
```

これらのメッセージは無視してかまいません。将来のリリースで修正される予定です。

4.10 Accounting ユーティリティでの長いデバイス名のサポート

V8.4

長いデバイス名を処理できるように Accounting ユーティリティが修正されています。たとえば、TNA10000: や MBA1000: など、ユニット番号が 9999 よりも大きな端末 (TNA)、ユニット番号が 999 よりも大きな MBA デバイス、あるいはその他のデバイス名など、7 文字以上のデバイス名を表示できるようになりました。

以前は、デバイス名が 7 文字を越える場合は、任意の不定な文字が表示されていました。accounting.dat ファイルへの新しいレコードの書き込みには新しいアカウンティング・レコード・バージョン (version4) が使用され、Accounting ユーティリティでこれらの新しいレコードの読み取りおよび表示が可能です。

4.11 新しいシステム・パラメータ PAGED_LAL_SIZE

PAGED_LAL_SIZE は、ページング動的プールのルックアサイド・リストで使用する最大サイズをバイト数で設定します。

4.11.1 ページング・プールのルックアサイド・リスト

V8.4

ページング動的プールでルックアサイド・リストを使用することが可能になり、これによりシステム性能の向上が期待できる場合があります。この機能は SYSGEN パラメータ PAGED_LAL_SIZE で制御され、デフォルトではこの機能は無効 (パラメータ値 0) になっています。

可変ページング・プール・フリーリストで断片化が発生した場合、このルックアサイド・リストの使用を有効にすることにより効果をもたらすことがあります。SYSGEN パラメータ PAGED_LAL_SIZE は、ルックアサイド・リストで使用する最大サイズをバイト数で設定します。この最大サイズよりも大きなパケットは、可変ページング・プール・フリーリストから割り当てられます。512 バイトの値を使用すると、論理名の作成と削除を集中的に実行するようなシステムで効果があることが確認されています。

このパラメータを動的に変更して、必要に応じて機能を有効にしたり、調整したり、あるいは無効にすることができます。機能を有効にした後、値を小さくすると、もはやアクティブには使用されていないパケットがページング・プール・ルックアサイド・リストに存在する場合があります。これは、DCL の "Over-limit Lookaside Blocks" や SDA の SHOW MEMORY/POOL/FULL コマンドで確認できます。これらのパケットは以前使用されていましたが、今は新しい PAGED_LAL_SIZE よりも大きくなっています。これらのパケットは、この SYSGEN パラメータの値が増加するか、あるいは

はページング・プールの不足が発生してルックアサイド・リストからパケットが再要求されると再び使用されます。

ルックアサイド・リスト上のパケットがページ・プールのほとんどあるいはすべてを消費するような状況を防ぐために、ページ・プールのルックアサイド・リストはページ・ダイナミック・プールの最後の4分の1のパケットには使用されません。ページ・プール・メモリの不足が発生した場合、ルックアサイド・リスト上のパケットも再要求されます。

デフォルト値0でこの機能を無効にした場合、ページ・プールの動作は以前のバージョンのOpenVMSと同じで、ページ・プール変数フリーリストからパケットの割り当ておよび割り当て解除が行なわれます。

4.12 2 TB ディスク・ボリュームのサポートに関する制限事項

V8.4

OpenVMS Version 8.4 は最大 2 TB のディスク・ボリュームをサポートしますが、以下のような制限事項があります。

- V8.4 より古いバージョンの OpenVMS では 1 TB よりも大きなボリュームはサポートしておらず、1 TB よりも大きなボリュームのマウントもサポートしていません。間違って古いバージョンの OpenVMS にマウントしてしまうのを防ぐために、MOUNT の最新のパッチでは、そのようなシステムで 1 TB を超えるサイズのボリュームのマウントを明示的に許可していません。
- F\$GETDVI()レキシカル関数の項目 MAXBLOCK, FREEBLOCKS, EXPSIZE, および VOLSIZE はターゲット・ディスクのサイズに依存した情報を返します。OpenVMS Version 8.4 では、ターゲット・ディスクのサイズが 1 TB を超える場合、これらの F\$GETDVI()項目は負の値を返すこととなります。これは、DCL が 32-bit 符号付き整数の計算および比較を行うためです。F\$GETDVI()とこれらのタイプのアイテムコードを使用するコマンド・プロシージャは、1 TB よりも大きなボリュームを扱えるように修正する必要があります。DCL 整数表現の範囲を越える数値の処理については、『OpenVMS DCL ディクショナリ』を参照してください。

4.13 SAS テープ・ドライブの構成

V8.4

SAS テープ・ドライブの命名および構成は、Fibre Channel テープ・ドライブの構成に使用するのと同じコマンドで実行する必要があります。詳細は、『OpenVMS Cluster 構成ガイド』の Fibre Channel テープ・サポートの項を参照してください。

4.14 外付け SAS ディスク・デバイスのネーミング

V8.4

Smart Array コントローラ以外で制御される外付けの SAS ドライブは、`3DGA<UDID>`として構成できます。<UDID>には LUN に対するユニークなデバイス ID を指定します。SAS デバイスと Fibre Channel デバイスを区別するために、外付け SAS ディスク・デバイス名にはアロケーション・クラス値 3 を使用するのに対し、Fibre Channel ディスク・デバイス名にはアロケーション・クラス値 1 を使用する点に注意してください。

4.15 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。

注意

外部認証ユーザに対する注意事項:

外部認証をサポートする SYS\$ACM 対応の LOGINOUT.EXE と SETPO.EXE (SET PASSWORD) イメージを使用している場合、OpenVMS Version 8.4 へのアップグレードを行うと SYS\$ACM 対応のイメージがリストアされます。

ACMELOGIN キットのインストールについては SYS\$HELP:ACME_DEV_README.TXT を参照してください。

4.15.1 外部認証とパスワードのポリシー

V8.4

ユーザ認証には SYSUAF.DAT 以外のソースに対する外部認証を使用し、カスタマイズしたパスワード処理のパスワード・ポリシーを使用している場合、Password Policy 共有イメージをインストールし、LOAD_PWD_POLICY システム・パラメータを有効にした後に、ACME Server を再起動する必要があります。

次のコマンドを使用して ACME を再起動してください。

```
$ SET SERVER ACME_SERVER /RESTART
```

4.15.2 OpenVMS Integrity 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 Integrity 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。

OpenVMS Integrity 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 OpenVMS Integrity メンバ・ノードへ OpenVMS Integrity 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。

4.15.3 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ため、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
$
```

4.15.4 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。

4.15.5 ACME_SERVER プロセスにおける制限事項 (Integrity サーバのみ)

本リリースでは、Integrity サーバでは 1 つのスレッドのみがアクティブなので、Integrity サーバでは SET SERVER ACME/CONFIG=THREAD_MAX コマンドは無視されます。

注意

Integrity サーバではスレッドの数を増やさないでください。スレッドの数を増やすと、ACME_SERVER プロセスがクラッシュし、ログインに失敗ことがあります。

4.16 IPB (Itanium Primary Bootstrap) が有効なダンプ・デバイスの検出に失敗する問題

V8.4

ダンプ・デバイス (DOSD) が別の接続済みの HBA に構成された PCI バス用の HP PCIe コンボ・カードである AD221 など、ブリッジされたデバイスを接続すると、ダンプ・デバイスの PCI バス・ナンバリングのリナンバリングが行なわれ、有効なダンプ・デバイスを見つけるのが困難になります。

回避方法

新しい I/O カードに接続した後、ブート/ダンプ・オプションを確認してください。その後、DUMP_DEV とブート・デバイス・リストをリフレッシュしてください。

4.17 SHUTDOWN.COM の変更

V8.4

SHUTDOWN.COM が修正されており、キュー・システムのシャットダウン・プロシージャ SYSHUTDOWN_0010.COM が存在すればこれを実行します。テンプレートには 3 つのサンプル・ルーチンが含まれており、キュー・システムをより速くシャットダウンおよび再起動あるいはフェールオーバーすることができます。

4.18 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。

4.18.1 Cluster over IP (IP クラスタ・インターコネクト)

HP OpenVMS Version 8.4 では、Cluster over IP 機能が拡張されています。この機能は、業界標準の Internet プロトコルを使用して単一の LAN あるいは VLAN セグメントを超えたクラスタ環境を構築するための機能を提供します。また、OpenVMS クラスタの耐災害性能を向上させるものでもあります。

この項では、Cluster over IP に関する既知の問題および制限事項について説明します。

4.18.1.1 必要なソフトウェア

V8.4

Cluster over IP は、OpenVMS Alpha Version 8.4 および OpenVMS Integrity Version 8.4 上でのみ利用できます。Cluster over IP は、HP TCP/IP Services for OpenVMS の V5.7 も必要とします。

4.18.1.2 Integrity サーバのサテライトノードとブートサーバは同一 LAN 内に存在

V8.4

Cluster over IP を正常に初期化しクラスタに正しく参加するには、Integrity サーバのサテライト・ノードは、そのサテライト・ノードのブート・サーバと同じ LAN に存在しなければなりません。

サテライトのブート時に Cluster over IP を開始するには、Integrity サーバのサテライト・ノードとそのサテライト・ノードのブート・サーバ間で LAN クラスタ通信を持つこととも必要になります。

4.18.1.3 Alpha サテライト・ノードではディスク・サーバとの LAN チャネルを必要とする

V8.4

IP のみの環境では、Alpha サテライト・ブートは処理に失敗します。すなわち、Alpha サテライトをブートする際、ブート・サーバを含むすべてのノードがクラスタ通信の IP チャネルのみを使用している場合、次のメッセージを表示してサテライト・ブートに失敗します。

```
cluster-W-PROTOCOL_TIMEOUT, NISCA protocol timeout %VMScluster-I-REINIT_WAIT,  
Waiting for access to the system disk server
```

4.18.1.4 IPv6 のサポート

V8.4

Cluster over IP では、クラスタ通信インタフェースに対して IPv6 型のアドレスはサポートしていません。

4.18.1.5 DHCP (Dynamic Host Configuration Protocol) あるいはセカンダリ・アドレスのサポート

V8.4

Cluster over IP では、クラスタ通信に使用するアドレスはそのインタフェースの静的なプライマリ・アドレスでなければなりません。さらに、クラスタ通信に使用する IP アドレスとインタフェースは、フェールセーフ構成で使用されないようにしなければなりません。

4.18.1.6 複数の IP インタフェース構成

V8.4

同じデフォルト・ゲートウェイで複数の IP インタフェースを構成すると、いずれかのインタフェースにおける通信の損失により CLUEXITS でクラスタ通信の中断が発生する可能性があります。

4.18.1.7 ifconfig コマンドの使用

V8.4

クラスタ通信に使用するインタフェースが ifconfig により再起動されると、他のノードに対するクラスタ通信の損失が発生し、ノードの CLUEXIT も発生します。

4.18.1.8 複数のゲートウェイ構成

V8.4

Cluster over IP の構成情報は、ブート時の最初にロードされる構成ファイルに保管されています。この構成情報には、TCP/IP が使用するデフォルト経路あるいはゲートウェイも含まれます。現在のところ構成ファイルにはデフォルト経路のみを入れることができ、ノードの起動時に使用されます。

4.18.1.9 ブロック転送 XMIT の変更

V8.4

PEdriver は、LAN インタフェース (BUS) と同様にクラスタ通信に使用する各 IP インタフェースをエミュレートします。IP バスは下記に示すように Xchain_Disabled 属性を持ちます。これは、TCP/IP によって転送されるブロック転送パケットが、PEdriver から TCP/IP バッファへコピーされることを意味しています。

システム管理に関する注意事項

4.18 OpenVMS Cluster システム

```
$ mc scacp show ip
```

```
NODEG PEA0 Device Summary 16-FEB-2009 12:29:15.92:
```

Device	Type	Errors + Events	Status	Mgt Priority	Buffer Size	MgtMax BufSiz	Line Speed	Total Pkts (S+R)	Current IP Address
IE0		184	Run Online XChain_Disabled	0	1394	0	N/A	1419711	15.146.235.222

4.18.1.10 ダウンライン・ロードのための LANCP

V8.4

Cluster over IP の構成に関連する変更と Cluster over IP を有効にするための設定は CLUSTER_CONFIG_LAN.COM でのみ可能なので、Cluster over IP は、Alpha におけるダウンライン・ロードには DECnet の代わりに LANCP を必要とします。この制限事項は将来のリリースで解決される予定です。

4.18.1.11 デュプレックスの不一致

V8.4

ホスト・デュプレックスにおけるデュプレックス・モードの不一致、あるいは半二重から全二重へのデュプレックス・モードの変更は、IP がクラスタ通信に使用されたときに CLUEXIT が発生する原因となります。CLUEXIT の発生を防ぐために、デュプレックスの不一致について確認することをお勧めします。

4.18.1.12 共有システム・ディスクのアップグレード

V8.4

共有システム・ディスク構成では、古いバージョンから OpenVMS Version 8.4 へのアップグレードの際に、アップグレードを実行しているノードで Cluster over IP が有効になります。他のノードについては、アップグレードの後に CLUSTER_CONFIG_LAN コマンド・プロシージャを実行して Cluster over IP を有効にしてください。

たとえば、システム PIPER および MARLIN が共有システム・ディスクにそれぞれルート SYS0 および SYS1 を持っているとします。ノード PIPER でアップグレードを実行した場合、PIPER では Cluster over IP が有効になります。MARLIN で Cluster over IP を有効にするには、CLUSTER_CONFIG_LAN コマンド・プロシージャを実行してください。

この制限事項は、将来のリリースで取り除かれます。

4.18.1.13 拡張された CLUSTER_CONFIG_LAN コマンド・プロシージャ

V8.4

Cluster over IP を構成するために CLUSTER_CONFIG_LAN コマンド・プロシージャが拡張されています。このコマンド・プロシージャを使用して Cluster over IP を有効にし、クラスタ通信に IP を使用することができます。

このコマンド・プロシージャを使用してクラスタにスタンドアロン・ノードを追加すると、次のようなメッセージが表示されます。

```
"IA64 node, using LAN for cluster communications. PEDRIVER will be loaded.  
No other cluster interconnects are supported for IA64 nodes."
```

Integrity サーバ・ノード上の構成プロシージャによってこのようなメッセージが表示されるにもかかわらず、LAN あるいは IP のどちらかあるいは両方をクラスタ通信に使用することができます。ノードの属性がクラスタ・メンバに変更されると、LAN はデフォルトで有効になります。IP は、CLUSTER_CONFIG_LAN コマンド・プロシージャによりオプションで有効にすることができます。PEdriver は、LAN 通信および IP 通信の両方に対してロードされます。

この CLUSTER_CONFIG_LAN コマンド・プロシージャのメッセージは、将来のリリースで修正されます。

4.18.2 Integrity VM での OpenVMS Cluster のサポート

V8.4

OpenVMS Integrity Version 8.4 は、Integrity VM 上でゲスト OS としてサポートされます。OpenVMS ゲストはクラスタ環境で構成可能です。

4.18.2.1 OpenVMS ゲストのクラスタ・インターコネクト

V8.4

OpenVMS ゲストは、LAN あるいは Cluster over IP (IPCI) の両方でクラスタ内の他のノードと通信できます。

4.18.2.2 Integrity VM クラスタ環境での MSCP のサポート

V8.4

MSCP は、OpenVMS ゲスト・システムで構成されるクラスタ環境で共有ストレージ機能を提供するのに使用されます。

4.18.2.3 オンライン・マイグレーションのサポート

V8.4

クラスタの一部となっている OpenVMS ゲストのオンライン・マイグレーションはサポートされません。

4.18.3 混成プラットフォームのサポート

V8.2

- Integrity サーバを含む運用クラスタに 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 Integrity サーバのどちらかをクラスタから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 ソフトウェアの 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については、『HP OpenVMS インストール・ガイド[翻訳版]』を参照してください。

4.18.4 ポート割り当てクラスを使用したサテライト・システム

V8.2

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 Integrity サーバ・サテライト・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 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。

4.19 6メンバ・シャドウセットの混成クラスタにおける互換性

V8.4

OpenVMS Version 8.4 は、Volume Shadowing の機能として拡張メンバーシップ機能をサポートします。この機能により、シャドウセットで 3 メンバよりも多く、最大 6 メンバを持つことができます。この機能は、シャドウセットに 4 番目のメンバが追加されたときに有効になります。混成バージョンの OpenVMS クラスタにおいて、以下のようないくつかの重要なポイントがあります。

- 拡張メンバーシップ・シャドウイング機能を使用するには、そのシャドウセットをマウントするすべてのシステムで OpenVMS Version 8.4 が稼動していなければなりません。

- 拡張メンバーシップ・シャドウイング機能を使用して OpenVMS Version 8.4 システムにシャドウセットをマウントしようとした時に、そのシャドウセットがすでにクラスタ内の以前のバージョンの OpenVMS システムにマウントされている場合、V8.4 システムへのマウントは失敗します。
- 拡張メンバーシップ・シャドウイング機能を使用して OpenVMS Version 8.4 システムにマウントされているシャドウセットを拡張メンバーシップ・シャドウイング機能をサポートしていない以前のバージョンの OpenVMS システムにマウントしようとする、そのマウント処理は失敗します。
- 一旦シャドウセットで拡張メンバーシップ・シャドウイング機能が有効になると、たとえその後メンバ数が 4 メンバよりも少なくなってもその特性は維持されます。その特性は、そのシャドウセットのマウントがクラスタワイドで外されるまで残ります。
- このシャドウセット機能は OpenVMS VAX ではサポートされません。この機能を有効にせずに OpenVMS Alpha あるいは OpenVMS Integrity システムにシャドウセットをマウントしている場合は、このシャドウセットを OpenVMS VAX システムにマウントすることができます。この互換性は仮想ユニット特性により維持されます。

4.20 6メンバ・シャドウセットの下位互換性

V8.4

ストレージ制御ブロック (SCB) の新しい領域に拡張メンバーシップ・シャドウイング機能のサポートに必要な拡張メンバーシップ配列を保管します。このため、コマンド行でメンバが指定されている場合 (すなわち最大 3 メンバ)、あるいはメンバが Index 0, 1, あるいは 2 (古い) スロットにある場合のみ、以前のバージョンの OpenVMS システムへの 6 メンバ・シャドウセットのマウントが可能です。

以前のバージョンの OpenVMS では、シャドウセットの再構築に使用される \$MOUNT/INCLUDE 修飾子は既存のメンバ・リストのみを見つけることができ、SCB の新しいメンバ領域は見つけることができません。このため、SCB の新しい拡張メンバ領域のメンバはマウントしません。

4.21 WBEM Services および WBEM Providers に関する注意事項

ここでは、WBEM における既知の問題および制限事項について説明します。

4.21.1 OpenPegasus 2.9 ベースの WBEM Services for OpenVMS

WBEM Services for OpenVMS Version 2.9 は、The Open Group の Pegasus オープン・ソース・プロジェクトの OpenPegasus 2.9 コード・ストリームをベースにしています。

4.21.2 OpenVMS ゲストに対する WBEM Providers のサポート

V8.4

ゲスト OS は仮想マシン上で実行されるので、OpenVMS ゲスト上で実行中の WBEM Providers は、CPU、メモリ、エンクロージャ、シャーシ、ファン、電源、および管理プロセッサなどの WBEM インスタンス・データおよびイベントをサポートしません。これらは、VM Host OS 上で実行中の WBEM Providers によってサポートされます。

4.21.3 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していません)。

4.21.4 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:

```
$ cimmoofl "-E" "--xml"
```

誤:

```
$ cimmoof -E -xml
```

4.22 Monitor ユーティリティの変更

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。

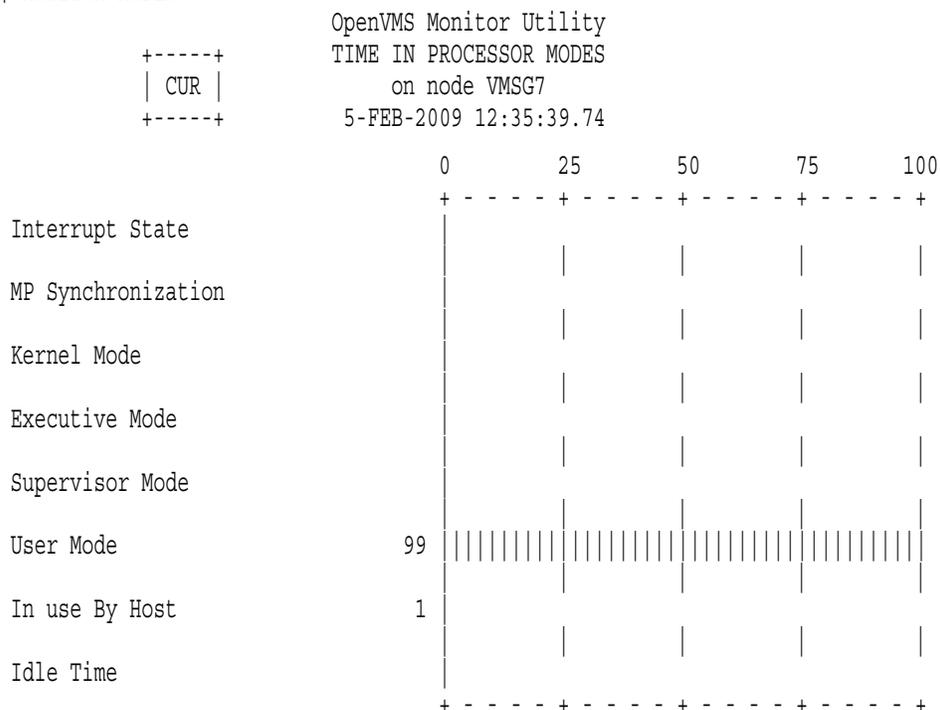
4.22.1 Integrity VM におけるゲスト・オペレーティング・システム

V8.4

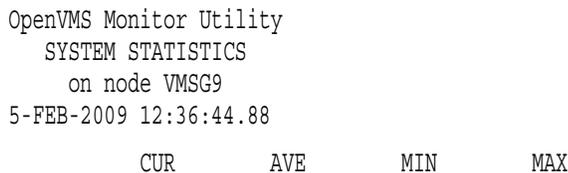
OpenVMS Integrity Version 8.4 は、Integrity Virtual Machines (Integrity VM) 上でゲスト・オペレーティング・システムとして動作することが可能です。OpenVMS を Integrity VM システム上のゲストとして実行している場合、Monitor ユーティリティはこのゲストが使用した総 CPU 時間を示します。また、Monitor ユーティリティは、Integrity VM によってゲストに割り当てられた総 CPU 時間も示します。

これらの情報は、MONITOR MODES および MONITOR SYSTEM /ALL コマンドにより表示できます。システムがゲストとして実行中の場合は、上記のコマンドを実行すると "Compatibility Mode" の代わりに "In use by Host" と表示されます。このフィールドは、他のゲストあるいは Integrity VM によって使用されており現在のゲストには利用できない総 CPU 時間として解釈されます。表示のスケールは、ホストに物理的に搭載されている CPU の数ではなく、そのゲストで実際に構成されている vCPU (Virtual CPU) の数をベースにします。

\$ MONITOR MODES



\$ MONITOR SYSTEM/ALL



システム管理に関する注意事項

4.22 Monitor ユーティリティの変更

Interrupt State	0.00	0.12	0.00	0.33
MP Synchronization	0.00	0.00	0.00	0.00
Kernel Mode	0.00	0.06	0.00	0.50
Executive Mode	0.00	0.00	0.00	0.00
Supervisor Mode	0.00	0.00	0.00	0.00
User Mode	98.33	98.03	96.50	98.50
In use By Host	1.66	1.77	1.33	3.33
Idle Time	0.00	0.00	0.00	0.00
Process Count	25.00	24.72	24.00	25.00
Page Fault Rate	0.00	10.96	0.00	47.50
Page Read I/O Rate	0.00	0.96	0.00	3.16
Free List Size	46851.00	46945.54	46850.00	47105.00
Modified List Size	317.00	316.90	316.00	317.00
Direct I/O Rate	0.00	1.37	0.00	5.50
Buffered I/O Rate	1.00	2.68	0.66	9.83

注意

ゲストでMONITOR MODESおよびMONITOR SYSTEM /ALLコマンドを実行した時に表示されるデータは、このゲストが仮想 CPU で消費した時間です。

4.22.2 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。

4.22.3 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
OpenVMS Version 8.3, 8.3-1H1, 8.4	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。

4.23 新しいパラメータ

V8.3-1H1

本リリースでは、新しい GH_RES_CODE_S2 パラメータをサポートします。このパラメータは、常駐 64 ビット S2 スペースの常駐イメージ・コード粒度ヒント・リージョンのサイズをページ単位で指定するのに使用します。

/SEGMENT=CODE=P2 修飾子を指定してリンクしたイメージだけが、このリージョンにコードを置くことができます。詳細については、『OpenVMS Linker Utility Manual』および『OpenVMS システム管理者マニュアル』の INSTALL ユーティリティの説明を参照してください。

GH_RES_CODE には、AUTOGEN および FEEDBACK 属性があります。

4.24 SYS\$LDDRIVER の制限事項

V8.2-1H1

SYS\$LDDRIVER.EXE は、フリーウェアの擬似デバイス・ドライバです。これを使うと、OpenVMS オペレーティング・システムで仮想ディスクが作成できます。OpenVMS V7.3-1 以降では、このドライバは SYS\$COMMON:[SYS\$LDR]内に置か

れ、CDRECORD または COPY/RECORDABLE_MEDIA を使った CD や DVD のマスタリングのためのソース仮想ディスクの作成をサポートしています。このフリーウェアのドライバの使用はこの形式でのみサポートされています。これ以外での使用は、下記のように規定された、フリーウェアの使用上の制限事項に該当します。

OpenVMS Freeware はそのままの状態を保証なしで提供されます。配布や再配布についての制限はありません。弊社では、このソフトウェアについてのサービス、ソフトウェアの修正、および正常な動作に関する保証は行いません。

4.25 システム・パラメータ CPU_POWER_MGMT のデフォルト値の変更

V8.3-1H1

システム・パラメータ CPU_POWER_MGMT のデフォルト値が 1 から 0 (すなわち on から off) に変更されています。この変更により、IDLE からの中断あるいは CPU の終了の遅延が改善されています。この値は、OpenVMS の将来のバージョンで再度変更される可能性があります。

4.26 予約メモリ機能を使用するサテライト・システムのブート

V8.3-1H1

Integrity サテライト・システムで SYSMAN の予約メモリ機能を使用するためには、SYS\$SYSTEM:VMS\$RESERVED_MEMORY.DATA ファイルのプロテクションが全ユーザから READ+EXECUTE アクセス可能な状態でなければなりません。この設定が行なわれていないと、サテライト・システムのブート時に次のような警告メッセージが表示されます。

```
%VMS_LOADER-W-Warning: Unable to load file SYS$SYSTEM:VMS$RESERVED_MEMORY.DATA
```

サテライトに対するメモリ・リザベーションを追加するために SYSMAN を実行した後、SYS\$MANAGER:CLUSTER_CONFIG_LAN.COM を実行して、VMS\$RESERVED_MEMORY.DATA ファイルに正しいプロテクションを設定してください。プロテクションを設定するには、クラスタ構成プロシージャのメイン・メニューから次のメニューを選択してください。

3. CHANGE a cluster member's characteristics.

「CHANGE」メニューから次のメニュー項目を選択します。

13. Reset an IA64 satellite node's boot environment file protections.

What is the satellite name (leave blank to use a specific device and root)?

メモリ・リザベーションを追加したシステムのサテライト名、あるいはサテライト・ブート・デバイスと root を入力します。将来のリリースでは SYSMAN が修正されこの状況は発生しなくなります。

4.27 SCACP エラー・カウンタが再送エラーをレポートすることがある

V8.3-1H1

システムの PEA0: デバイスが多数のエラーを表示する場合には、これらのエラーは再送エラーであり、実際のエラーではありません。実際のエラーを確認するには、SCACP コーティリティを使って PEA0 チャネルで多数の再送が行われているか確認し、LANCP コーティリティを使って、PEdriver が使っている LAN デバイスで実際デバイス・エラーがあるか確認する必要があります。再送は行われているけれども、デバイス・エラーは発生していない場合には、PEA0: デバイス・エラーは再送エラーであり、実際のエラーではない可能性があります。

4.28 Virtual Connect

ここでは Virtual Connect に関する注意事項について説明します。

4.28.1 フェールオーバーと RECNXINTERVAL

V8.3-1H1

特に規模の大きいクラスタ環境の場合、Virtual Connect Manager のフェールオーバー時間に余裕を持たせるために、RECNXINTERVAL に省略時の値より 20 は大きな値を設定することが必要となります。

4.29 メディア使用前の INITIALIZE/ERASE

V8.3-1H1

ストレージ・メディアを初めて使用する前に、ストレージ・メディアに対して DCL コマンド INITIALIZE/ERASE を実行することをお勧めします。このようにすることで、別のオペレーティング・システムや診断ツールでそれまで使っていた古いデータを削除することができます。

このような古いデータがあった場合には、コンソール・コマンドの出力では、次の例に示すように、3つの疑問符(???)で示されます。

システム管理に関する注意事項

4.29 メディア使用前の INITIALIZE/ERASE

```
Shell> ls fs1:\
Directory of: fs1:\
00/00/07 19:16p    1,788,984,016 ???
00/00/80 12:00a         0 ???
    2 File(s)  1,788,984,016 bytes
    0 Dir(s)
```

この問題は、将来のリリースで修正される予定です。

4.30 Performance Data Collector for OpenVMS (TDC)

V8.4

OpenVMS Version 8.4 をインストールすると、TDC_RT Version 2.3-20 がインストールされます。TDC Version 2.3-20 は、Multinet および TCPWare 環境での評価は行なわれていません。

4.31 システムのハングアップまたはクラッシュからの回復 (Integrity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。

4.32 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。

4.33 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、UCB\$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。

4.34 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。

4.35 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。

4.36 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。

4.37 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。	

4.37.1 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、同じ名前での、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 4.18.3 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 4.18.3 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。

4.37.2 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 (SYSSMKDRIVER.EXE)、または SCSI 汎用クラス・ドライバ (SYSSGKDRIVER) 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。

4.37.3 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。

4.37.4 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。

4.37.5 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、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。

4.37.6 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。

4.38 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。

4.38.1 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。

(次ページに続く)

表 4-3 (続き) Galaxy の定義

Galaxy が使われる文脈	説明
システム・パラメータ	メモリ共有を設定します。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有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有したくない場合があります)。このパラメータでは、ノードで共有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。メモリの共有のための設定は、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、メモリは共有されません (デフォルト)。
ソフト・パーティション	CPU の移動、API の使用、メモリの共有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。

4.39 セル型システムでの複数の nPartition

V8.2-1

HP Integrity rx7620、HP Integrity rx8620、HP Integrity Superdome などのサーバに複数の nPartition 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 nPartition 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。

4.39.1 ES40 上の Galaxy: 非圧縮ダンプの制限事項

恒久的制限事項

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。

4.39.2 ES40 上の Galaxy: Fast Path の無効化

恒久的制限事項

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 のリポ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 をリポ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。

4.40 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ポートした場合、リポ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。

4.41 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。

4.41.1 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。

4.41.2 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。

4.41.3 システム・パラメータの変更

V8.4

OpenVMS Version 8.4 では、以下のシステム・パラメータが変更されました。

- AUTO_DLIGHT_SAV - now dynamic
- MULTITHREAD - now dynamic

V8.3

OpenVMS Version 8.3 で以下のシステム・パラメータが変更されています。

- 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_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。

4.42 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 (SYSSFBDriver.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 SYSSMANAGER 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。

```
$ @SYSSMANAGER: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 コーティリティ (TFU) を起動します。

```
$ RUN SYSS$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 SYSSFBDRIVER.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 (SYSSFBDRIVER.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。

4.43 User Environment Test Package (UETP) (Integrity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。

4.44 推奨するキャッシュ方式

恒久的制限事項

Virtual I/O キャッシュ (VIOC) (別名 VAX Cluster キャッシュ (VCC)) は、OpenVMS Integrity では利用できません。OpenVMS Integrity 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 Integrity 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。

4.45 OpenVMS の Analyze ユーティリティ

ここでは、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。

4.45.1 フォーマットされたシンボル・ベクタがデータ・セグメントに正しく表示される

これまでは、シンボル・ベクタ要約情報には、シンボル・ベクタが存在するセグメントが表示されず、ダイナミック・セグメント内のシンボル・ベクタだけがフォーマットされていました。

この問題は、OpenVMS V8.3-1H1 で修正されました。シンボル・ベクタが存在するデータ・セグメントでは、フォーマットされたシンボル・ベクタが表示されるようになりました。フォーマットされたシンボル・ベクタはデータに埋め込まれ、データのダンプ内に表示されます。

同一データを2度フォーマットिंगするのを避けるために、ダイナミック・セグメントではシンボル・ベクタは表示されなくなりました。シンボル・ベクタを簡単にフォーマットできるようにするために、/SEGMENT 修飾子に対して SYMBOL_VECTOR キーワードが使用できるようになりました。このキーワードを指定すると、結果の出力にはフォーマットされたシンボル・ベクタだけが表示されます。周りのデータは表示されません。すべてのデータをフォーマットして表示するには、番号を使ってセグメントを選択してください。

以前のコマンド/SEGMENT=DYNAMIC と同等のシンボル・ベクタの出力が必要な場合には、/SEGMENT=(DYNAMIC,SYMBOL_VECTOR) 修飾子を使ってください。

要約情報には、シンボル・ベクタが存在するデータ・セグメントの名前が表示されません。

4.45.2 トランスファー・アレイがデータ・セグメント内にフォーマットされるようになった

これまでは、セグメント番号または ALL キーワードのいずれかを使って、トランスファー・アレイが存在するデータ・セグメントを選択しても、トランスファー・アレイはフォーマットされませんでした。トランスファー・アレイについての情報は、要約にのみ表示されていました。

この問題は、OpenVMS Version 8.3-1H1 で修正されました。

フォーマットされたトランスファー・アレイがデータ・セグメント内に表示されるようになりました。

4.45.3 システム・バージョン・アレイがダイナミック・セグメント内にフォーマットされるようになった

システム・バージョン・データはダイナミック・セグメント内にあります。これまで、セグメント番号、ALL または DYNAMIC のキーワードのいずれかを使って、ダイナミック・セグメントを選択しても、システム・バージョン・アレイは表示されませんでした。システム・バージョン・アレイについての情報は、要約にのみ表示されていました。

この問題は、OpenVMS Version 8.3-1H1 で修正されました。

フォーマットされたシステム・バージョン・アレイがダイナミック・セグメント内に表示されるようになりました。

4.45.4 /SEGMENT 修飾子の機能拡張

ダイナミック・セグメント用の /SEGMENT 修飾子が、機能拡張されました。Analyze は、/SEGMENT=DYNAMIC 修飾子でキーワードを受け付けるようになり、カスタマイズされた情報が表示できるようになりました。選択可能な情報に対するキーワードは、以下のとおりです。

- ALL - (省略時の指定) ダイナミック・セグメントのすべての部分をフォーマット
- TAGS - タグ・アレイをフォーマット
- IMAGE_STRINGS - 指定したイメージの文字列をフォーマット
- RELOCATIONS - イメージの再配置情報をフォーマット
- FIXUPS - イメージのフィックスアップをフォーマット
- SYSTEM_VERSION_ARRAY - システム・バージョン・アレイをフォーマット

省略時の指定である /SEGMENT=ALL は、すべてのイメージ情報をフォーマットします。

TAGS キーワードを使ったフォーマットには必要なイメージの名前が含まれるため、名前を表示するために IMAGE_STRINGS を指定する必要はないことに注意してください。

4.45.5 セクション・エスケープ機能のサポート

OpenVMS V8.3 では、Analyze ユーティリティは、65,280 を超えるセクションを持つオブジェクト・モジュールを解析すると、完了しませんでした。セクション・ヘッダ・テーブルの表示で、ループに陥っていました。

この問題は、OpenVMS V8.3-1H1 で修正されました。

4.46 OpenVMS の INSTALL コーティリティ (S2 空間に常駐イメージをインストールする)

INSTALL コーティリティは、常駐イメージのコード・セグメントを 64 ビット S2 アドレス空間へインストールする機能をサポートするようになりました。ただしすべてのコードが、完全な 64 ビット・アドレス空間 (P2 または S2) で実行できるわけではありません。たとえば、例外を処理するためには、コードは 64 ビット PC に対応していなければなりません。また、一部のコンパイラでは、64 ビット・アドレス空間用のコードを生成するときには、/POINTER_SIZE=64 コマンド修飾子を指定する必要があります。

対応していないコードを S2 空間にマッピングするのを避けるために、INSTALL コーティリティは、省略時の指定では、コード・セグメントを S0/S1 空間にマッピングします。INSTALL コーティリティが常駐イメージのコード・セグメントを S2 空間にマッピングするのは、以下の条件が満たされた場合だけです。

- 開発者がイメージをリンクする際に/SEGMENT_ATTRIBUTE=CODE=P2 修飾子を使うことにより、コードが 64 ビット対応であることを明示的に指定した場合。
- S2 空間の常駐コード領域に、コード・セグメントをマッピングできるだけの大きさの割り当て済みの空間がある場合。領域のサイズは、システム・パラメータ GH_RES_CODE_S2 (ページ数) で決まります。省略時の値は 0 です。つまり、デフォルトでは、64 ビット対応の常駐イメージであっても、コードは S0/S1 空間にマッピングされます。

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 lib\$routines.h ファイルにおけるプロトタイプ宣言の修正

V8.4

lib\$routines.h で宣言されているプロトタイプ lib\$stat_vm_64 は、定義と一致するように修正されています。

```
#ifdef __NEW_STARLET
unsigned int lib$stat_vm_64(
    __int64 *code,
    unsigned __int64 *value_argument);
#else /* __OLD_STARLET */
unsigned int lib$stat_vm_64(__unknown_params);
#endif /* #ifdef __NEW_STARLET */
```

5.2 シンボリック・デバッグ

V8.4

OpenVMS Alpha Version 8.4 では、ブレークポイントを設定した際、デバッグは FORTRAN 関数と、別のコンパイル・ユニットで同じ名前宣言された変数とを見分けることができません。

5.3 C ランタイム・ライブラリ

OpenVMS Version 8.3-1H1 で修正された問題は、以下のとおりです。

- ランタイム・ライブラリには、解放したメモリにアクセスして、その結果を使ってポインタを進めるという不完全なコードがありました。マルチスレッド・コードでは、元のスレッドがポインタを進める前に別のスレッドがそのメモリを再利用する可能性がありました。この問題は、ポインタを解放する前にアクセスするようにアップデートすることによって解決されました。

- 新しいプロセス・ワイドの例外処理モード (`pure_unix`) が導入されました。このモードでは、非 C++ 例外 (OpenVMS コンディションとも呼ばれる) が C++ のキャッチオール・ハンドラで捕捉されることはありません。このモードは、`cxxl$set_condition(condition_behavior)` に `pure_unix` 引数を付けて呼び出すことによって要求できます。

```
cxxl$set_condition(pure_unix);  
  
condition_behavior enum declared in <cxx_exception.h> header  
has been extended to include pure_unix member.
```

次のサンプル・プログラムを例に、`pure_unix` モードの動作を説明します。このプログラムは、`ACCVIO` でクラッシュするように記述されています。`cxxl$set_condition()` の呼び出しをコメント・アウトした場合には、プログラムは "caught" を出力して終了します。

```
#include <stdio.h>  
#include <cxx_exception.h>  
  
void generateACCVIO() { *((int*)0) = 0; }  
  
int main() {  
    cxxl$set_condition(pure_unix);  
    try {  
        generateACCVIO();  
    }  
    catch(...) {  
        puts("caught");  
    }  
}
```

注意

この新機能を使うためには、新バージョンの `cxx_exception.h` を使用する必要があります。これは V7.3 以降のコンパイラに付属している `CXXL$ANSI_DEF.TLB` ファイルに含まれています。

- ランタイム・ライブラリは、関数内で記憶域の存続期間が自動的に決まるオブジェクトが定義されていると、捕捉した例外によって関数が終了したときに、オブジェクトの破棄に失敗することがありました。この問題は修正されました。
- シグナルの捕捉を有効にしている場合、ランタイム・ライブラリを使うと、スレッド・キャンセル・シグナル (`CMAS_ALERTED`) とスレッド終了シグナル (`CMAS_EXIT_THREAD`) を、それぞれタイプ `CXXLSPTHREAD_CANCEL` (または `CX6LSPTHREAD_CANCEL`) と `CXXLSPTHREAD_EXIT` (または `CX6LSPTHREAD_EXIT`) へのポインタまたは参照を使って、キャッチ・ハンドラで捕捉できるようになりました。新しいタイプではこれらのシグナルは排他的に捕捉されます。

注意

この新機能を使うためには、新バージョンの `cxx_exception.h` を使用する必要があります。これは V7.3 以降のコンパイラに付属している `CXXL$ANSI_DEF.TLB` に含まれています。

- C++ RTL では、最近の CRTL の変更に合わせて、SIGTRAP の内部マッピングが `SS$_BREAK` から `SS$_TBIT` に変更されました。
- C++ RTL では、デストラクタがスタックをアンwindしているときに例外が発生すると、デストラクタが例外で終了していても、`std::terminate()` を呼び出していました。この問題は修正されました。
- C++ RTL は、C++ 例外の処理中に外部例外 (たとえば、非 C++ OpenVMS コンディション) が発生すると、`std::terminate()` を呼び出していました。この動作は、発生した OpenVMS コンディションによってスタックのアンwindが必要になる場合にのみ `std::terminate()` を呼び出すように修正されました。
- OpenVMS コンディションは C++ キャッチ・ハンドラで捕捉できるため、C++ RTL はそのコンディションを C++ 例外の表現に合った内部形式に変換します。この変換により、トレースバックの際に誤った情報が表示されることがありました。この問題は修正されました。

以下の問題は、今回のバージョンの C++ Library (V7.3 以降のコンパイラ) で修正されました。

- <http://issues.apache.org/jira/browse/STDCXX-397> で説明されているように、`<algorithm.cc>` ヘッダ内の `__introsort_loop()` 関数にはバグがありました。すなわち、特定の入力シーケンスで、`std::sort` の性能が低下することがありました。詳細は、<http://issues.apache.org/jira/browse/STDCXX-397> にある問題 STDCXX-397 に対する Apache の追跡情報を参照してください。

このバグは修正されました。ただし、特定の入力シーケンスでは、この修正により `std::sort` の動作が変わることがあります。つまり、同等の順位を持つエレメントがソート結果のシーケンス内で置かれる相対的位置が変わることがあります。この動作上の変化は、`std::stable_sort` とは異なり、`std::sort` では同等の順位を持つ特定のエレメントの相対的順序を保証していないため許容することはできませんが、`std::sort` の従来動作に依存しているアプリケーションが正常に動作しなくなる可能性があります。このような状況を避けるために、この修正は `__RW_FIX_APACHE_STDCXX_397` マクロで条件付けられており、このマクロを定義してプログラムをコンパイルした場合にのみ、この修正が有効になります。

- 標準 GNU モードでコンパイルすると、ライブラリは `_RWSTD_NO_IMPLICIT_INCLUSION` マクロを定義するようになりました。この変更により、ライブラリ・ヘッダに対応するテンプレート定義ファイルがインクルード (`#include`) されるようになりました。これは、標準 GNU モードでは暗黙的な `#include` が無効になっているために必要になります。

この変更を行う前は、以下のプログラムは、標準 GNU モードでコンパイルすると、未定義シンボルをリンクしていました。

```
#include <vector>

int main() {
    std::vector<int> v;
    v.push_back(0);
}
```

- C++ 規格の第 27.6.1.3 項[lib.istream.unformatted]によれば、std::basic_istream クラスの以下の get メンバ関数は、空行の場合のように、文字が何も含まれていない場合には、setstate(failbit) を呼び出すことになっています。I64 システムではこの関数は failbit を設定していましたが、Alpha システムでは設定していませんでした。

```
istream_type& get(char_type *s, streamsize n, char_type delim);
istream_type& get(char_type *s, streamsize n);
```

5.4 プロセス/アプリケーションがハングする

lib\$find_image_symbol ランタイム・ライブラリ・ルーチンの LIBRTL ドキュメントには、以下の制限事項が適用されます。

pthread (またはそれよりも古い CMA スレッド・インタフェース) を使っている共有可能イメージをアプリケーションで動的にアクティベートしている場合には、メイン・イメージに pthread\$rtl イメージをリンクする必要があります。

5.5 POSIX スレッドを使用しているプログラムでの AST 実行要求の明確化

V8.3-1H1

スレッド化されたプログラムで AST を利用することができます。『Guide to the POSIX Threads Library』の付録 B.12.5 では、一般的な使用上の注意が説明されています。しかし、その説明では、アップコールを無効にした(省略時の設定)プログラムでの AST 実行要求の動作が明確ではありません。

アップコールを無効にしたプログラムでは、ユーザ・モードの AST は、AST の実行が要求された時点で実行中のすべてのスレッドに割り込みます。したがって、AST サービス・ルーチンでは、実行するコンテキストに前提(スレッド ID、利用できるスタック領域など)を設けることはできません。

また、このガイドの付録 B.12.5 内の大部分の説明では、将来提供される OpenVMS のバージョンについて記載していることに注意してください。一般化された "per-thread" やスレッドをターゲットとする AST についての説明は、将来オペレーティン

グ・システムが拡張される可能性があることを示しています。ただし、現在までのすべての OpenVMS リリースでは、ユーザ・モードの AST は、総じてプロセスに対して要求されているものとして処理されています。

5.6 ディレクトリ・ファイルの RMS \$PARSE 検証

V8.3-1H1

OpenVMS V8.3 から、\$PARSE サービスは、ディレクトリ指定に該当するすべてのディレクトリを検証して、ディレクトリ特性がセットされていることを確認するようになりました。以前の OpenVMS バージョンでは、.DIR 拡張子を持つディレクトリでないファイルにアクセスを試みると、必ずしも \$PARSE サービスからではなく、\$OPEN サービスから SS\$_BADIRECTORY エラーが返されました。V8.3 から、エラーは必ず \$PARSE サービスから返されるようになりました。ただし、構文チェックだけの \$PARSE については除きます。

5.7 IOLOCK8 を使わない FibreChannel ポート・ドライバ

V8.3-1H1

多くの I/O サブシステム・コンポーネントは、IOLOCK8 スピンロックを使って CPU 間で動作の同期をとります。スピンロックを取得することにより、性能ボトルネックが発生します。V8.3-1H1 から、各 FibreChannel ポート・ドライバ (SYSSPGQDRIVER, SYSSPGADRIVER, SYSSFGEDRIVER) デバイスでは、IOLOCK8 の代わりに独自のポート固有のスピンロックを使って内部動作の同期をとるようになりました。これにより、大部分の構成で、各 CPU の IOLOCK8 スピンロックの待ち時間が大幅に減少し、同時に FibreChannel I/O の転送速度が向上しました。

新しいポート・ドライバのいずれかに接続するクラス・ドライバには、軽微な変更が必要なので、以前のままでは動作しない非 HP クラス・ドライバを利用していないか調べる必要があります。簡単に確認する方法は、SDA コマンド CLUE SCSI/SUMMARY の出力を見て、他社製のクラス・ドライバ・デバイスの名前が、"Device" 欄の FGx0 または PGx0 ポート・デバイスのデバイス階層に表示されていないか調べることです。

詳細は、次の SDA セッション例の後の注意を参照してください。

```
$ ANALYZE/SYSTEM
OpenVMS system analyzer
SDA> CLUE SCSI /SUMMARY
```

プログラミングに関する注意事項

5.7 IOLOCK8 を使わない FibreChannel ポート・ドライバ

SCSI Summary Configuration:

SPDT	Port	STDT	SCSI-Id	SCDT	SCSI-Lun	Device	UCB	Type	Rev
81624200	FGB0								
		8162CDC0	3						
				8162D240	0	GGA22	8162F380	HSV200	
				8162F180	1	DGA22801	8162FD40	HSV200	6100
				81632900	2	DGA22802	81632AC0	HSV200	6100
				816354C0	3	DGA22803	81635680	HSV200	6100
				81638080	4	DGA22804	81638240	HSV200	6100
		8162D400	4						
				8162DD80	0	GGA22	8163AC40	MRD200	
				8163B5C0	1	RJA22801	8163B780	RFD200	6100
				8163C840	2	RJA22802	8163CA00	RFD200	6100
				8163DAC0	3	RJA22803	8163DC80	RFD200	6100
				8163ED40	4	RJA22804	8163EF00	RFD200	6100

解説

この出力でのすべての DGA デバイスと GGA デバイスは、それぞれ変更された HP クラス・ドライバ SYS\$DKDRIVER および SYS\$GKDRIVER 経由でアクセスされているので、新しいポート・ドライバと一緒に使っても安全です。

タイプが MRD200 の物理デバイスは弊社が認定したデバイスではありませんが、IOLOCK8 問題は発生しません。このデバイスは GGAx ユニット経由でアクセスされるため、変更された HP 汎用クラス・ドライバ SYS\$GKDRIVER が使われるからです。

RJA デバイスは、変更された HP クラス・ドライバで制御されないため、新しいポート・ドライバでは動作しません。

5.8 C++ コンパイラ

V8.3-1H1

Integrity サーバ用の C++ Version 7.2 for OpenVMS では、マクロ `__INITIAL_POINTER_SIZE` に 0 が設定済みです。これは、C++ Version 7.1 Compiler では未定義のままでした。これは C++ Version 7.2 を C コンパイラと調和させ、`pointer_size` プラグマをサポートするための意図的な変更です。C++ Version 7.1 ではサポートしていません。

この変更によって、ポインタ・タイプを宣言しているシステム・ヘッダ・ファイルで選択される特定の宣言を使っていて、今まで正常にコンパイルされたコードで診断メッセージが表示されることがあることに注意してください。starlet ヘッダを使って、`__NEW_STARLET` を定義してコンパイルするアプリケーションで、このメッセージが表示されることがあります。

新しい宣言に合わせてアプリケーションのソース・コードを変更することができない場合には、CXX コマンド行にコマンド行修飾子/UNDEF=__INITIAL_POINTER_SIZE を追加して、C++ Version 7.2 コンパイラによって、上記のマクロが定義されないようにし、Version 7.1 コンパイラの場合に定義しているのと同じ宣言がシステム・ヘッダで提供されるようにしてください。

5.9 DCE IDL C++ アプリケーションのビルド

V8.3-1H1

CXX Version 7.2 以降で DCE IDL C++ アプリケーションをビルドすると、リンカから未定義シンボルの警告が出力されます。これは既知の問題です。HP サポート・サービスに連絡して、この警告が表示されないようにするために必要なパッチを要求してください。

5.10 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/SYSEXEXE 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。

5.11 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 Integrity 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/SYSEXEXE 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。

プログラミングに関する注意事項

5.11 特権データ構造体の変更

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

Integrity サーバでは、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。

5.11.1 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。Integrity 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 Integrity での変更により、Alpha と Integrity 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。

5.11.2 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、Integrity 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。

5.11.3 64 ビットの論理ブロック番号 (LBN)

V8.4

OpenVMS はこれまで 32 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 2 TB に制限されます。LBN の内部領域は 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。現在の 32 ビットの LBN は今後も維持され、64 ビット拡張後もそのまま使えます。

5.11.4 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きました。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 SPLSC_DYNAMIC が設定されている場合のみ参照されます。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。

5.11.5 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。

5.11.6 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。

5.11.7 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

恒久的な変更

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_STDSREQCOM, COM_STDSPOST, COM_STDSPOST_NOCNT, IOC_STDSPOST_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_STDSREFERENCE_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が、

この状態で NSA_STDSREFERENCE_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_STDSREFERENCE_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用します。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。

- EXESQUEUE_FORK
- EXE_STDSQUEUE_FORK
- EXESPRIMITIVE_FORK
- EXE_STDSPRIMITIVE_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。

5.12 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 Integrity 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 Integrity 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/>

5.12.1 IEEE 浮動小数点フィルタ (Integrity 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。

5.12.2 Ada イベントのサポート (Integrity 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 Integrity 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・ブレイ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。

5.12.3 C++ 言語の問題 (Integrity のみ)

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。

5.13 Ada コンパイラ (Integrity 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。

5.14 Backup API: ジャーナリング・コールバック・イベントの制限事項

恒久的な制限事項

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。

5.15 C プログラム: CASE_LOOKUP=SENSITIVE を設定したコンパイル

恒久的な制限事項

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。

5.16 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。

5.16.1 C RTL TCP/IP ヘッダ・ファイルのアップデート

V8.4

C RTL では、TCP/IP を呼び出すユーザ向けのヘッダ・ファイルを提供しています。これらのヘッダ・ファイルは、C RTL のヘッダ・ライブラリ (DECC\$RTLDEF.TLB) に含まれています。

TCP/IP の新機能に合わせて、以下の新しいヘッダ・ファイルが追加されています。

SCTP.H
SCTP_UIO.H

これらのヘッダ・ファイルにより、SCTP (Stream Control Transmission Protocol) サポートが提供されます。SCTP については、『HP TCP/IP Services for OpenVMS Version 5.7 Release Notes』を参照してください。

5.16.2 バックポート・ライブラリが提供されなくなった

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、提供されなくなりました。

5.16.3 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。

5.16.4 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。

5.16.5 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。

5.16.6 Integrity 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。

5.16.7 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 Integrity 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。

5.17 呼び出し標準規則とローテートするレジスタ (Integrity 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。

5.18 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。

5.18.1 Secure Delivery

V8.4

Version 8.4 以降，すべての OpenVMS キットは，PCSI キットおよび VMSINSTAL キットも含めて HP Code Signing Service (HPCSS) を使用して署名されます。署名および確認に CDSA は使用しません。新しい署名ファイル<full kit name>_HPC が作成され，キットと共に提供されます。この署名ファイルを使用してキットの確認が行なわれます。

新しい評価機能 HPBinarychecker がすべての OpenVMS システムにインストールされ，HP Code Signing Service を使用してキットの署名を確認します。詳細は，『HP OpenVMS V8.4 新機能説明書』を参照してください。

OpenVMS Version 8.4 よりも前のバージョンで署名されたキットは，V8.4 上の CDSA を使用して確認できます。

5.18.2 インストールと初期化に関する注意事項

V8.4

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，次のように 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
```

なお，このアップグレード・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。

また，システムをリポートする際にはアップグレード・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 の削除は行わないでください。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 PRODUCT REMOVE の使用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時に PCSI ユーティリティで使用するためのものです)。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，密接に関連付けられているので，OpenVMS から CDSA を削除しようとするすると正常に動作せず，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すると，次のようなメッセージが表示されます。

```
The following product has been selected:
  HP I64VMS CDSA V2.4-315                Layered Product

Do you want to continue? [YES]

%PCSI-E-HRDREF, product HP I64VMS CDSA V2.4-315 is referenced by HP I64VMS OPENV
MS V8.4

The two products listed above are tightly bound by a software dependency.
If you override the recommendation to terminate the operation, the
referenced product will be removed, but the referencing product will have
an unsatisfied software dependency and may no longer function correctly.
Please review the referencing product's documentation on requirements.

Answer YES to the following question to terminate the PRODUCT command.
However, if you are sure you want to remove the referenced product then
answer NO to continue the operation.
```

5.19 デバッグ・モード: CPUSPINWAIT バグ・チェックの回避

恒久的な条件

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。

5.20 Delta/XDelta デバッガ

ここでは、OpenVMS Alpha および Integrity システム上で動作する OpenVMS Delta および XDelta デバッガに関する注意事項について説明します。

5.20.1 XDelta のレジスタ表示に関する考慮 (Integrity のみ)

V8.2

OpenVMS Integrity 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 5.17 節を参照してください。

5.21 ファイル・アプリケーション: 『Guide to OpenVMS File Applications』の訂正

恒久的な訂正

『Guide to OpenVMS File Applications』が改訂される際には、下記の訂正が反映される予定です。

- 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。」

5.22 RMS 構造体についての HP BLISS コンパイラの警告 (Integrity のみ)

恒久的な条件

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, クォドワード・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場合があります。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。

5.23 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

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。

5.24 HP COBOL ランタイム・ライブラリ (RTL)

V8.4

OpenVMS Alpha および OpenVMS Integrity Version 8.4 では、HP COBOL RTL (DEC\$COBRTL) は V2.9-785 にアップデートされました。

5.24.1 COBOL CALL 文の性能改善

V8.4

COBOL CALL 文でサブルーチンを呼び出す際、OpenVMS Integrity では性能の低下が見られ、OpenVMS Alpha よりも長い時間がかかっていました。

ランタイム・ライブラリ (RTL) が修正され、COBOL CALL 文の性能が大きく改善されています。現在では OpenVMS Integrity での性能は OpenVMS Alpha での性能と同等になっています。

5.25 HP Fortran for Integrity Servers

V8.2

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 Integrity システム上で動作し、OpenVMS Integrity 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 Integrity 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 Integrity 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、Integrity サーバ 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
```

5.26 HP MACRO for OpenVMS

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 Integrity 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。

5.26.1 Macro-32 コンパイラの拡張

V8.4

Macro-32 コンパイラが拡張されており、OpenVMS Alpha および OpenVMS Integrity にインストラクション・レベルの新しいビルトインが含まれています。

表 5-1 に、追加された新しいビルトインを要約します。

表 5-1 Macro-32 の新しいビルトイン

ビルトイン	オペランド ¹	説明	サポート・プラットフォーム
EVAX_WH64	<AB>	Alpha 書き込みヒント 64 インストラクションを生成します。	Alpha
IA64_PADD1	<WQ,RQ,RQ>	最初の引数をデストネーションとし、次の2つの引数をソース・オペランドとして、シングルバイト形式で並列追加インストラクションを生成します。	Integrity
IA64_PADD1_SSS	<WQ,RQ,RQ>	最初の引数をデストネーションとし、次の2つの引数をソース・オペランドとして、シングルバイト形式で並列追加インストラクションを生成します。結果と両方のオペランドは signed として扱います。	Integrity
IA64_PADD1_UUS	<WQ,RQ,RQ>	最初の引数をデストネーションとし、次の2つの引数をソース・オペランドとして、シングルバイト形式で並列追加インストラクションを生成します。結果と最初のソース・オペランドは unsigned として扱います。2つめのソース・オペランドは signed として扱います。	Integrity
IA64_PADD1_UUU	<WQ,RQ,RQ>	最初の引数をデストネーションとし、次の2つの引数をソース・オペランドとして、シングルバイト形式で並列追加インストラクションを生成します。結果と両方のオペランドは unsigned として扱います。	Integrity
IA64_PADD2	<WQ,RQ,RQ>	最初の引数をデストネーションとし、次の2つの引数をソース・オペランドとして、2バイト形式で並列追加インストラクションを生成します。	Integrity
IA64_PADD2_SSS	<WQ,RQ,RQ>	最初の引数をデストネーションとし、次の2つの引数をソース・オペランドとして、2バイト形式で並列追加インストラクションを生成します。結果と両方のオペランドは signed として扱います。	Integrity
IA64_PADD2_UUS	<WQ,RQ,RQ>	最初の引数をデストネーションとし、次の2つの引数をソース・オペランドとして、2バイト形式で並列追加インストラクションを生成します。結果と最初のソース・オペランドは unsigned として扱い、2つめのソース・オペランドは signed として扱います。	Integrity
IA64_PADD2_UUU	<WQ,RQ,RQ>	最初の引数をデストネーションとし、次の2つの引数をソース・オペランドとして、2バイト形式で並列追加インストラクションを生成します。結果と両方のソース・オペランドは unsigned として扱います。	Integrity

¹ビルトインは、オペランド WQ - クオードワード書き込み、PQ - クオードワード読み取り、AB - バイトのアドレス、を必要とします。

(次ページに続く)

表 5-1 (続き) Macro-32 の新しいビルトイン

ビルトイン	オペランド ¹	説明	サポート・プラットフォーム
IA64_PADD4	<WQ,RQ,RQ>	最初の引数をデストネーションとし、次の2つの引数をソース・オペランドとして、4バイト形式で並列追加インストラクションを生成します。	Integrity
IA64_MUX1	<WQ,RQ,RQ>	最初の引数をデストネーションとし、2つめの引数をソース・オペランドとして、'mux1' インストラクションを生成します。 ² 3つめの引数は置換の実行を指定します。	Integrity
EVAX_S4ADDQ	<WQ,RQ,RQ>	Alpha 型クオワード (すなわち 8 バイト) の追加インストラクションを生成します。	Alpha および Integrity
IA64_LFETCH_NT1, IA64_LFETCH_NT2, IA64_LFETCH_NTA, IA64_LFETCH_EXCL_NT1, IA64_LFETCH_EXCL_NT2, IA64_LFETCH_EXCL_NTA	<RQ,RQ>	これらの拡張されたプリフェッチ・ビルトインは、デフォルトでないキャッシュ位置ヒント (すなわち ".nt1", ".nt2", および ".nta") を指定する手段を提供します。最初のオペランドはプリフェッチするアドレスで、2つめのオペランドは reg-base-update-form か imm-baseupdate-form のどちらかのためのものです。オペランドが文字ゼロの場合、no-baseupdate-形式が使用されます。	Integrity
IA64_LFETCH_FAULT_NT1, IA64_LFETCH_FAULT_NT2, IA64_LFETCH_FAULT_NTA, IA64_LFETCH_FAULT_EXCL_NT1, IA64_LFETCH_FAULT_EXCL_NT2, IA64_LFETCH_FAULT_EXCL_NTA		"FAULT"付きの LFETCH インストラクションを生成します。	Integrity

¹ビルトインは、オペランド WQ - クオワード書き込み、PQ - クオワード読み取り、AB - バイトのアドレス、を必要とします。

²置換を指定する文字は3つめの引数として使用される必要があります。文字に対する置換のマッピングは次のとおりです。

- BRCST 0
- MIX 8
- SHUF 9
- ALT 10
- REV 11

インストラクションの基本的な情報については、それぞれのハードウェア・アーキテクチャのマニュアルを参照してください。

5.26.2 OpenVMS Integrity 用 HP MACRO

V8.3

HP MACRO for OpenVMS Integrity 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。Integrity 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必要があります。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できます。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。

5.26.3 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
```

5.26.4 /OPTIMIZE=VAXREGS 修飾子は Integrity サーバ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、Integrity 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。Integrity 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。

5.26.5 浮動小数点数のゼロ除算エラーが検出されない (Integrity 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するようになります。

5.27 Hypersort ユーティリティ

ここでは、OpenVMS Alpha および OpenVMS Integrity 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 5.39 節を参照してください。

5.27.1 弊社への問題の報告

恒久的な条件

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。

5.27.2 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 5.27.8 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・エクステントを設定します。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。

5.27.3 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。

5.27.4 /FORMAT=RECORD_SIZE の制限事項

恒久的な制限事項

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。

5.27.5 Hypersort と検索リスト、および論理名の使用

恒久的な制限事項

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。

5.27.6 作業ファイルの空き領域不足

恒久的な制限事項

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。

5.27.7 入力アスタリスク (*) の制限事項

恒久的な制限事項

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。

5.27.8 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

恒久的な制限事項

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。

5.28 Intel®アセンブラ (Integrity のみ)

恒久的な制限事項

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、恒久的な要件となります。

5.29 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。

5.29.1 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 (Integrity 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。

5.29.2 Integrity サーバ・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 (Integrity 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 Integrity 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]TRACEMSG.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 Integrity 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。

Integrity サーバでは、.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。これは恒久的な状態です。

5.29.3 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

恒久的な制限事項

OpenVMS Alpha および Integrity 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。

5.30 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。Integrity Linker に関する注意事項は第 5.31 節を参照してください。

5.30.1 SYMBOL_VECTOR リンカ・オプションの制限事項

恒久的な制限事項

SYMBOL_VECTOR リンカ・オプションを使用して汎用宣言したい共有イメージのシンボル名あるいはプログラム・セクション名は、TPA\$_SYMBOL シンボル・タイプに属する文字で構成されていなければなりません。TPA\$_SYMBOL シンボル・タイプは下記の文字で構成されます。

A から Z の英字 (大文字および小文字)、ドル記号 (\$)、アンダースコア (_)、コード 192 よりも大きな DEC 多国語文字。

次の例では、TPA\$_SYMBOL シンボル・タイプに属さない文字 '~' がプロシージャ名に含まれています。このため、SYMBOL_VECTOR リンカ・オプションが次のようなエラー・メッセージを表示しています。

```
$ LINK/SHARE SOME_LIBRARY, SYS$INPUT/OPT
SYMBOL_VECTOR=(WRONG~NAME=PROCEDURE)
%ILINK-F-OPTSYNERR, syntax error in options file SYS$INPUT:.;
-ILINK-E-OPTLIN, options line in error
      SYMBOL_VECTOR=(WRONG'~'NAME=PROCEDURE)
```

5.30.2 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。

5.30.3 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。

5.30.4 スタックの要素数は最大 25 に制限

恒久的な制限事項

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。

5.31 OpenVMS Integrity 用 Linker ユーティリティ

ここでは、Integrity サーバ用 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 5.30 節を参照してください。

5.31.1 SYMBOL_VECTOR リンカ・オプションの制限事項

恒久的な制限事項

SYMBOL_VECTOR リンカ・オプションの制限事項については第 5.30.1 項を参照してください。

5.31.2 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

V8.3-1H1

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 修飾子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 修飾子(デフォルト)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、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>

The image file is not affected; it can be executed with the RUN command
without any problem:

```
$ RUN MINIREF
```

この誤りは OpenVMS V8.3-1H1 の Linker で修正されました。

5.31.3 /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

V8.3-1H1

トランスファー・アドレスが含まれている Integrity サーバ・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でした。

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
%ILINK-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 Integrity サーバ・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 コマンドまたは LIBRARY 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
```

または

```
$ 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を次に示します。

```
$ LINK/EXECUTABLE=MAIN SUBROUTINES.OBJ, LIB.OLB/LIBRARY
```

明示的に使用する例を次に示します。

```
$ LINK/EXECUTABLE=MAIN SUBROUTINES.OBJ, LIB.OLB/INCLUDE=MAIN
```

この問題は、以下のどちらかの形で現れます。

- 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。

この誤りは OpenVMS V8.3-1H1 のリンカで修正されました。

5.31.4 最大セクション数

V8.3-1H1

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、OpenVMS Integrity V8.3 の Linker では実装されていませんでした。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ていました。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ーフェースの数は 65280 未満でなければなりませんでした。

OpenVMS V8.3-1H1 I64 Linker では、この制限はなくなりました。入力ファイル、オブジェクト・モジュール、あるいは共用イメージは、65280 を超えるセクションを持つことができます。

5.31.5 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

V8.3-1H1

OpenVMS Integrity 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。

この誤りは OpenVMS V8.3-1H1 で修正されました。

5.31.6 demangler 情報を検索するとアクセス違反になる

V8.3-1H1

状況によっては、共用可能イメージ内に demangler 情報がないにもかかわらず、リンカでその情報の検索を試みると、リンカがアクセス違反で異常終了することがありました。

この問題は OpenVMS V8.3-1H1 で修正されました。

5.31.7 NOGLOSYM エラー・メッセージに対する誤った二次メッセージ

V8.3-1H1

NOGLOSYM エラー・メッセージに対して、OpenVMS Integrity V8.3 の Linker は誤った再配置タイプを表示し、一部の情報を 2 度表示していました。

この問題は OpenVMS V8.3-1H1 で修正されました。

5.31.8 未定義シンボルについての誤った情報

V8.3-1H1

未定義シンボルに対する USEUNDEF 操作によって、同一の参照についての情報が誤って 2 回表示されることがありました。この問題は、コンパイラが未定義シンボルへの参照について、再配置ペア (LTOFF22X/LDXMOV) を生成した場合に発生します。

この問題は OpenVMS V8.3-1H1 で修正されました。

5.31.9 誤った UNMAPFIL エラー

V8.3-1H1

ELF でない入力ファイルがあると、リンカは INVLDHDR エラー・メッセージを表示していました。そして INVLDHDR エラーの後には、誤った UNMAPFIL エラー・メッセージを表示していました。

この問題は OpenVMS V8.3-1H1 で修正されました。

5.31.10 共用可能イメージ・マップ内の識別子の最大長の変更

V8.3-1H1

リンカ・マップでは、リンカはオブジェクト・モジュールと共用可能イメージについて、最大で 14 文字の識別子をプリントしていました。本来は、オブジェクト・モジュールの識別子は無制限で、共用可能イメージの識別子は 15 文字です。

V8.3-1H1 Linker は、共用可能イメージの識別子として、最大で 15 文字プリントするようになりました。

5.31.11 共用可能イメージに対するリンケージ・タイプ・チェック

V8.3-1H1

これまで、リンカは共用可能イメージのシンボルに対して、タイプとリンケージのチェックを行っていませんでした。

OpenVMS V8.3-1H1 では、このチェックを行うようになりました。

5.31.12 プログラム・セクションの ABS 属性が無視される

V8.3-1H1

Integrity サーバでは、オフセットを定数に変換するためのラベルを持つプログラム・セクション (PSECT) には ABS 属性を設定できません。リンカにより次のようなエラー・メッセージが表示されます。

```
%ILINK-E-ILLABSPSC, absolute psect <psect-name> has non-zero length (not allowed)
```

OpenVMS 8.3-1H1 Linker は ABS プログラム・セクション属性を無視し、次のような情報メッセージを表示します。

```
%ILINK-I-PSCATTIGN, psect attribute ABS is not supported for OpenVMS ELF sections, attribute ignored
```

5.31.13 コマンド行に FP_MODE リテラルを指定していないとリンクはアクセス違反となる

V8.3-1H1

OpenVMS V8.3 では、コマンド行に FP_MODE リテラルを指定していないと、Integrity サーバ Linker はアクセス違反になっていました。

この問題は OpenVMS V8.3-1H1 で修正されました。

5.31.14 OpenVMS Integrity 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

Integrity サーバ上のオブジェクト・モジュールとイメージ・ファイルの形式である ELF (Executable and Linkable Format) に対する OpenVMS Integrity 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

5.31.15 Integrity リンカと Alpha リンカの違い

V8.3

OpenVMS Integrity 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 Integrity Linker の新機能についての説明もあります。

5.31.16 LINK_ORDER セクション・ヘッダ・フラグはサポートされていない

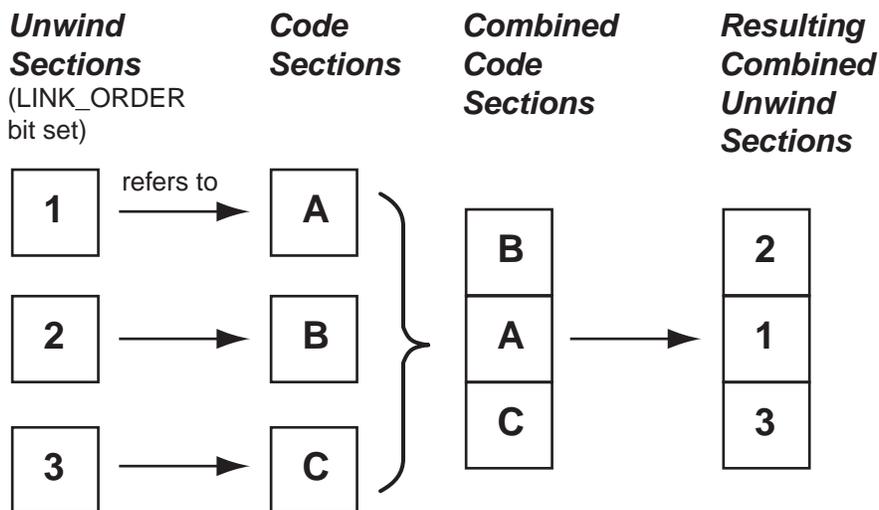
V8.2

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

5.31.17 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。これは恒久的な状態です。

5.31.18 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 Integrity 版 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 Integrity 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。

5.31.19 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 Integrity 版の 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 Integrity の将来のリリースの Integrity Linker で修正される予定です。

5.31.20 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。

5.31.21 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

Integrity 版 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。

5.31.22 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。

5.32 LTDRIVER: CANCEL SELECTIVE の制限事項

恒久的な制限事項

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。

5.33 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。

5.34 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。

5.34.1 CLUE コマンドは OpenVMS Integrity に移植されていない

V8.2

次の CLUE コマンドは、OpenVMS Integrity 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
```

5.35 OpenVMS Integrity 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 Integrity には含まれていません。PL/I コア・イメージのうち OpenVMS Integrity にはないものは以下のとおりです。

```
[SYSLIB]DPLI$RTL$SHR.EXE
```

```
[SYSMSG]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 Integrity 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 2.12 節の、関連する注意事項を参照してください。

5.36 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。

5.36.1 プロセス共有オブジェクトのサポート

V8.2

OpenVMS Alpha および OpenVMS Integrity 上の POSIX スレッド・ライブラリは、プロセス共有のミューテックスと条件変数をサポートします。

注意

ミューテックスと条件変数に対するプロセス共有の読み書きロックは Tru64 システムでのみサポートされます。

OpenVMS でサポートされる pthread ルーチンは以下のものです。

- pthread_condattr_getpshared
- pthread_condattr_setpshared
- pthread_mutexattr_getpshared
- pthread_mutexattr_setpshared

5.36.2 pthread_mutex_lock の新しい戻り状態値

V8.2

POSIX スレッド・ライブラリ pthread_mutex_lock をプロセス共有ミューテックスと使用すると新しいエラー状態 EABANDONED を返します。この状態は、ミューテックスを所有しながらプロセスが終了した場合のみ返されます。

5.36.3 新しいAPI pthread_mutex_tryforcedlock_np のサポート

V8.2

POSIX スレッド・ライブラリは、新しいAPI pthread_mutex_tryforcedlock_np をサポートします。pthread_mutex_tryforcedlock_np API は、解放されたプロセス共有ミューテックスの所有権を待ちます。

構文

```
pthread_mutex_tryforcedlock_np(mutex);
```

引数	データ型	アクセス
mutex	opaque pthread_mutex_t	read

C バインディング

```
#include <pthread.h>
int
pthread_mutex_tryforcedlock_np (
    pthread_mutex_t *mutex);
```

引数

mutex
Mutex to be locked.

説明

プロセス共有オブジェクトにより、ミューテックスや条件変数などのオブジェクトを複数のプロセス間で共有することができます。ミューテックスを所有しながらプロセスが終了した場合、そのミューテックスは、終了したプロセスが所有したままの状態になります。このため、他のスレッドあるいはプロセスがそのミューテックスを所有することはできません。

しかし、オプションのシステム・サービスとしてsys\$psshared_registerが用意されています。このシステム・サービスを使用すると、終了したプロセスが所有しているミューテックスを解放することができます。

この場合、pthread_mutex_lock の呼び出しで EABANDONED が返され、アプリケーションは pthread_mutex_tryforcedlock_np を呼び出して解放されたミューテックスの所有権を取得することができます。

戻り値

エラー状態が発生すると、このルーチンはエラーのタイプを示す整数値を戻します。次のような戻り値があります。

戻り値	説明
0	正常終了
[EPERM]	そのミュートックスは解放されておらず、他のスレッドが所有している。
[EINVAL]	そのミュートックスはプロセス共有ミュートックスではない。
[ENOSYS]	不正なシステム・サービスがコールされた。

5.36.4 例外処理中のスタック・オーバフロー (Integrity のみ)

V8.2

Integrity サーバでの例外処理には、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、Integrity サーバでの例外処理中に、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、Integrity サーバ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。

5.36.5 Integrity サーバでの THREADCP コマンドの動作

V8.2

OpenVMS Integrity 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、Integrity サーバでのスレッド・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。

5.36.6 浮動小数点のコンパイルと例外 (Integrity 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 Integrity 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 Integrity 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 Integrity 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。

5.36.7 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。

5.36.8 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。

5.36.9 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、非常に大きくなる場合があります。

5.36.10 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。

5.36.11 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。

5.37 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。

5.37.1 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。

5.37.2 RTL Library (LIB\$): 呼び出し標準ルーチン (Integrity 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。

5.38 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

キーパッド・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。

5.39 SORT32 ユーティリティ

ここでは、OpenVMS Alpha および OpenVMS Integrity 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 5.27.8 項と第 5.27.1 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 5.27 節を参照してください。

5.39.1 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。

5.39.2 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。SYSSCRATCH 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。

5.39.3 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")))
```

5.39.4 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。

5.39.5 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。

5.40 タイマ・キュー・エントリ (TQE)

恒久的な制限事項

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。

5.41 Watchpoint コーティリティ (Integrity のみ)

V8.2

Watchpoint コーティリティは、OpenVMS Integrity に移植されていません。弊社では、このコーティリティを今後のリリースで移植する予定です。

5.42 プログラム全体の浮動小数点モード (Integrity 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 Integrity 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 (リンクが決定したもの) 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>

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。

- USB デバイスのサポート (第 6.1 節)
- MP コンソールと BMC コンソール (第 6.2 節)
- AlphaServer 2100 (第 6.3 節)
- AlphaServer 8200/8400 (第 6.4 節)
- AlphaServer ES47/ES80/GS1280 システム (第 6.5 節)
- AlphaServer GS シリーズ (第 6.6 節)
- AlphaStation 200/400 (第 6.7 節)
- AlphaStation 255 (第 6.8 節)
- ATI RADEON 7000 グラフィック (第 6.9 節)
- ATI RADEON 7500 グラフィック (第 6.10 節)
- DECwindows X11 ディスプレイ・サーバ (第 6.11 節)
- DIGITAL モジュラ・コンピューティング・コンポーネント (第 6.12 節)
- Digital パーソナル・ワークステーション (第 6.13 節)
- デュアル・コントローラ HSGnn デバイス (第 6.14 節)
- Open3D グラフィック (第 6.15 節)
- PowerStorm 300/350 PCI グラフィック・コントローラ (第 6.16 節)
- RFnn DSSI ディスク・デバイス (第 6.17 節)
- RZnn ディスク・デバイス (第 6.18 節)
- sx1000 Integrity Superdome (第 6.19 節)
- ZLX グラフィック・ボード (第 6.20 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 USB デバイスのサポート

V8.4

HP では、HP が提供する USB デバイスを OpenVMS がインストールされた HP Integrity サーバ上でテストし、これをサポートしています。

場合によっては、他社製のデバイスも OpenVMS 上でテストされ、これをサポートしています。これらのデバイスについては、OpenVMS のエンジニアリング・チームで障害報告に対応し、ドライバの不具合を修正します。これらのデバイスについては、『HP 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』の付録に掲載しています。

USB の機能として、広範囲な種類のデバイスが 1 つの汎用デバイス・ドライバでサポートされます。OpenVMS では、未知の他社製デバイスの構成および使用を制限することはしていません。ただし、未知のデバイスの正しい動作を保証している訳ではありません。そのようなデバイスで問題が発生した場合、サポート窓口を通して報告していただくことは可能ですが、問題の修正や ECO パッチの提供を保証するものではありません。

他社製デバイスの正式なサポートが必要な場合は、HP の顧客担当チームあるいは代理店を通して要望を上げていただくことになります。

6.2 MP コンソールと BMC コンソールの制約事項 (Integrity 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。

6.2.1 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。

6.2.2 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

6.3 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。

6.3.1 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
```

ハードウェアに関する注意事項

6.3 AlphaServer 2100

```
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)
```

6.3.2 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.
```

6.4 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。

6.5 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 6.6.2 項の注意事項も、これらのシステムに関係します。

6.5.1 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。

6.5.2 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。

6.5.3 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。

6.5.4 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。

6.5.5 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。

6.6 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する注意事項について説明します。

6.6.1 AlphaServer GS80/160/320 システム: デバイスの制限事項

恒久的な制限事項

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。

6.6.2 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。

6.6.3 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。

6.6.4 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。

6.7 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、SYSS\$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

6.8 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・ポートと共有されます。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。

6.9 ATI RADEON 7000 グラフィック (Integrity のみ)

V8.2

ここでは、OpenVMS Integrity 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 6.10.1 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。

6.9.1 Integrity でのグラフィック・サポート

V8.2

OpenVMS Integrity は以下のグラフィック・オプションをサポートします。

- ATI Radeon 7500 PCI
- ATI Radeon 7000 PCI (内臓式および着脱可能式)
- ATI RN50 PCI

6.9.2 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。

6.10 ATI RADEON 7500 グラフィック

V7.3-2

ハードウェアに関する注意事項

6.10 ATI RADEON 7500 グラフィック

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 6.15 節を参照してください。

6.10.1 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 KB + 4*16 MB + 4*8 KB = 64 MB + 328 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)

6.10.2 DECW\$OPENGLSHR_RADEON.EXE を DECW\$MESA3DSHR_RADEON.EXE にリネーム

V8.2

共有ライブラリ SYSSLIBRARY: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 SYSSLIBRARY: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。

6.10.3 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。

6.10.4 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。

6.10.5 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。

6.10.6 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) に出力があると、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
```

6.10.7 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。

6.10.8 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。

6.10.9 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。

6.10.10 単一カラーマップのみサポート

RADEON 7500 グラフィック・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、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 6.10.16 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。

6.10.11 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・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。

6.10.12 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。

ハードウェアに関する注意事項

6.10 ATI RADEON 7500 グラフィック

```
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
```

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。

6.10.13 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。

6.10.14 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・プログラムで使用することができます。

6.10.15 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。

6.10.16 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。

6.10.17 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
```

DECW\$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
```

6.11 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。

6.11.1 S3 マルチヘッド・グラフィック

恒久的な制限事項

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。

6.12 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。

6.12.1 Alpha 5/366 および 5/433 PICMG SBC の制限事項

恒久的な制限事項

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。

6.12.2 SRM コンソールの更新

恒久的な制限事項

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。

<http://h18002.www1.hp.com/alphaserver/firmware/>

6.13 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。

6.14 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。

6.15 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 Integrity 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/>

6.16 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 6.15 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。

6.17 RFnn DSSI ディスク・デバイスとコントローラ・メモリのエラー

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
-----
_$22$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。

6.18 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。

6.18.1 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。

6.18.2 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。

6.18.3 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 6.18.3.1 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 6.18.3.2 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。

6.18.3.1 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

6.18.3.2 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.
```

6.19 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ースで修正される予定です。

6.20 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)

6.21 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 5.11 節の関連する注意事項も参照してください。

6.21.1 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 6.21.2 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。

6.21.2 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す。

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 6.21.1 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変に変更されました。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。

6.22 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。

6.23 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 5.11.7 項を参照してください。

6.24 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。

6.25 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 (CRCTXSV_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 (CRCTXSV_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。

IOC\$ALLOC_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。IOC\$ALLOC_CNT_RES は有効ビットを設定してから戻ります。

IOC\$DEALLOC_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 IOC\$DEALLOC_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。IOC\$DEALLOC_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 (CRCTX\$V_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。CRCTX\$V_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。

6.26 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。

6.26.1 Fibre Channel の EFI ドライバとファームウェアの要件

OpenVMS Integrity 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 fcd_update2.nsh または fcd_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。

6.26.2 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 Integrity 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。

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際に、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。

A.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。

A.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。

```
SYSS$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。

A.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が、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 A.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 LD_x_L/ST_x_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。LD_x_L 命令と ST_x_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 A.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。再コンパイルの後に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。

A.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 A.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 cmpl_e   R1, R17, R0
000405B0 F41FFFFA 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.
```

A.5 コンパイラのバージョン

表 A-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 A-1 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
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 A-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。

A.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

ACPI からの・エラーメッセージ	4-10
Ada コンパイラ	5-14
AlphaServer 2100	
SCSI コントローラの制限事項	6-4
コンソール表示	6-3
AlphaServer 4100	
FRU テーブルの制限事項	6-5
AlphaServer 8200 システム	
FRU テーブルの制限事項	6-5
AlphaServer 8400 システム	
FRU テーブルの制限事項	6-5
AlphaServer ES47/ES80/GS1280 システム	
MBM での時刻の設定	6-6
RAD のサポート	6-5
STOP/CPU およびシャットダウン動作	6-6
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	6-5
ライセンス要件	6-6
AlphaServer GS シリーズ・システム	
デバイスの制限事項	6-7
複数 I/O ポートの制限事項	6-9
AlphaStation 200/400	
ISA_CONFIG.DAT の変更が必要	6-10
AlphaStation 255	
PCI 構成の制限事項	6-10
API	
pthread_mutex_tryforcedlock_np	5-47
AST 実行要求	
POSIX	5-4
ATI RADEON 7000 グラフィック	6-11
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	6-11
ATI RADEON 7500 グラフィック	6-11~6-17
DECWSOPENGLSHR_RADEON.EXE のリネー ム	6-13
DECwindows サーバがハングアップ	6-14
OpenGL は IEEE 演算のみをサポート	6-14
高リフレッシュ・レートでの画像への影 響	6-13
Autoboot タイムアウト値の設定	4-7

B

Backup API	
ジャーナリング・イベント	5-14
BL860c および BL870c サーバ	4-5
BLISS	
HP BLISS コンパイラを参照	
BMC コンソールの制約事項 (Integrity の み)	6-2
BUGCHECKFATAL システム・パラメー タ	5-20
<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	5-44
CDSA	5-18
CLUE コマンド	
OpenVMS Integrity に移植されていな い	5-45
Cluster over IP	
CLUSTER_CONFIG_LAN	4-18
共有システム・ディスク	4-18
CMAP ファイル	
新規	2-3
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	2-4
サポート	2-4
Common Data Security Architecture	
CDSA を参照	
CPU_POWER_MGMT デフォルト	4-26
CPUSPINWAIT バグ・チェック	5-20
CRCTX ルーチンの機能の強化	6-29
C RTL	5-15
<builtins.h>の変更	5-17
DECC\$*.OLBライブラリの凍結	5-18
<time.h>の変更	5-16
バックポート・ライブラリ	5-16
ビルトイン _fciの追加	5-17
Ctrl/H キー・シーケンス	
DEL への再マッピング (Integrity のみ)	6-3

CTRL-P	3-4
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	5-15
C ランタイム・ライブラリ	
C RTL を参照	
訂正	5-1

D

data-reduced ELF オブジェクト・ライブラリ	
リンク対象	5-42
DCL コマンド	3-11
DDB 構造体	
アップデート	5-10
DDT Intercept Establisher ルーチン	
デバイス構成	4-34
DECCS*.OLB ライブラリの凍結	5-18
DECdfs for OpenVMS	
サポートされるバージョン	2-4
DECdtm	
Oracle 8i, 9i	4-29
DECforms Web Connector	
OpenVMS Version 7.3-1 以降で実行	2-5
DECmigrate	
V8.2 Open Source Tools CD に無い	3-12
DECnet for OpenVMS	1-4
DECnet/OSI	
DECnet-Plus for OpenVMS を参照	
DECnet-Plus for OpenVMS	1-4
新しいバージョンが必要	1-18
DEC PL/I	2-5
DECram	
HP DECram を参照	
DECRAM	
DECRYPT コマンドとの競合	2-7
DECwindows Motif	
HP DECwindows Motif を参照	
DECwindows X11 ディスプレイ・サーバ	6-18
グラフィック・ボードのサポート	6-26
Delta/XDelta デバッグ	5-21
レジスタ表示に関する考慮	5-21
DEVICE_NAMING システム・パラメータ	4-29
DFS マウント・ディスク	2-2
DIAGNOSE コマンド	
サポートされない	3-12
DIGITAL Modular Computing Components (DMCC)	
KZPDA コントローラと PBXGA グラフィック・カード	6-18
SRM コンソールの更新	6-18
Digital Personal Workstation	6-19
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

EDIT/FDL	
推奨バケット・サイズの変更	4-29
EFISCP ユーティリティ	
使用は推奨できない	4-30
EFI ドライバ	6-30
ELV	
Error Log Viewer (ELV) ユーティリティを参照	
Error Log Viewer (ELV) ユーティリティ	4-30
EV6 Alpha プロセッサ	A-1

F

F\$GETSYI("RAD_CPUS")	3-1
Fast Path	
無効化, ES40 上の Galaxy	4-38
Fibre Channel	6-30
互換キット	4-31
テープ・デバイスのマルチパス・フェールオーバの制限事項	4-35
FibreChannel	5-5
FMS キット	2-6
Fortran	
HP Fortran を参照	
Freeware	3-10

G

Galaxy	
定義	4-36
GiCAP	3-14
Gigabit Ethernet コントローラ	
制限	1-8

H

HP Availability Manager	
既知の問題	4-4
HP BASIC for OpenVMS	2-1
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	A-1
警告 (Integrity のみ)	5-22
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	A-1
HP COBOL RTL	5-24
HP Code Signing Service for OpenVMS	
HPCSS	3-2
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	A-1
HP DCE for OpenVMS の制限事項	2-3
HP DECprint Supervisor	
インストールの制限事項	1-4

HP DECram	2-6
DECGRAM と DECRYPT のコマンド競合	2-7
V8.2 では SIP として出荷	2-6
Version 2.5 (VAX のみ)	2-6
アップグレード前の削除 (Alpha のみ)	1-18
HP DECwindows Motif	
キーボードのサポートの制限 (I64 のみ)	1-13
サポートされているバージョン	1-5
ユーザが作成したトランスポートのサポ ート	2-8
HP Fortran	
Integrity 用	5-24
HP Secure Web Browser	
ODS-2 でインストール・エラーになる (Integrity のみ)	3-13
必要メモリ量の増加	3-12
HP Secure Web Server	
サポート	2-8
HP SSL	
暗号化と SSL のためのスタートアップ・コマン ド	1-17
HP-UX ゲストと OpenVMS ゲストに関する制限事 項	4-3
HSGnn	
障害	6-19
Hypersort ユーティリティ	5-30 ~ 5-32
<hr/>	
IDE CD	6-19
IEEE 浮動小数点	
フィルタ (Integrity のみ)	5-13
InfoServer の名前の長さ	4-5
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	6-5
Insight ソフトウェアによる OpenVMS の管 理	4-7
INSTALL ユーティリティ	
常駐イメージをインストールする	4-45
Integrity VM 上のゲスト OS	
シャットダウン動作	4-2
Integrity サーバ	
ファームウェア	1-10
Intel Itanium 9300 ベース・サーバのサポートに関 する情報	1-4
Intel®アセンブラ (Integrity のみ)	5-32
Invocation context block	5-52
IPC コマンド	3-19
iSCSI デモ・キット	
サポートされない	4-2

K

Kerberos	
Kerberos for OpenVMS	1-14
KPB 拡張	5-8

L

LAN	4-6
LANCP	
アップグレード後にデバイス・データベースを変 換する (Alpha のみ)	1-18
LAN ドライバ	
二重モードの不一致エラー	3-20
LIB\$GET_CURR_INVO_CONTEXT	
ドキュメントの訂正	3-18
LIB\$GET_INVO_CONTEXT	
ドキュメントの訂正	3-18
LIB\$GET_INVO_HANDLE	
ドキュメントの訂正	3-18
LIB\$GET_PREV_INVO_CONTEXT	
ドキュメントの訂正	3-18
LIB\$GET_PREV_INVO_HANDLE	
ドキュメントの訂正	3-18
LIB\$GET_UIB_INFO	
ドキュメントの訂正	3-18
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	3-18
lib\$ routines.h	5-1
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	3-16
Librarian ユーティリティ	3-16, 5-32
.STB ファイルの制限事項 (Integrity の み)	5-33
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	3-17
/REMOVE	3-16
LIBRTL	
呼び出し標準ルーチン (Integrity のみ)	5-52
Licenses	
virtual option	4-1
LINK_ORDER ELF セクション・ヘッダ・フラ グ	5-41
Linker for OpenVMS Alpha	
SYMBOL_VECTOR リンカ・オブショ ン	5-34

Linker for OpenVMS Integrity	
SYMBOL_VECTOR リンカ・オプション	5-36
Linker ユーティリティ	
ABS 属性が無視される	5-40
AFP_MODE リテラルを指定していないとリンカはアクセス違反となる	5-41
demangler 情報	5-39
SELECTIVE_SEARCH	5-37
誤った UNMAPFIL エラー	5-40
誤った二次メッセージ	5-39
共用可能イメージの作成日が正しくない	5-39
最大セクション数	5-38
識別子の最大長の変更	5-40
正しくないイメージ間デバッグ・フィックスアップ	5-36
未定義シンボルについての誤った情報	5-39
リンケージ・タイプ・チェック	5-40
LTDRIVER の制限事項	5-44

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	A-1
コードの再コンパイル	A-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	A-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	5-45
MMG_CTLFLAGS システム・パラメータ	3-15
Monitor ユーティリティの変更	4-22
MP コンソールの制約事項 (Integrity のみ)	6-2
MULTIPROCESSING システム・パラメータ	5-20

N

NetBeans	
Java Standard Edition, Development Kit v 1.4.2-7 以上が必要	2-2

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	6-20
OpenVMS	
ENCRYPT コマンドと DECRYPT コマンド	1-17
OpenVMS Alpha 用 Linker	5-34 ~ 5-36
RMS_RELATED_CONTEXT オプション	5-34
スタックの要素数は最大 25 に制限	5-36

OpenVMS Alpha 用 Linker (続き)	
多数のファイルの処理時にハングアップする	5-34
ライブラリ・チェックの動作の変更	5-35
OpenVMS Cluster システム	4-16 ~ 4-36
CI と LAN との間の切り替えによる性能の低下	4-34
SCSI マルチパス・フェイルオーバー	4-33
互換キット	4-31
バッチ・キット	4-30
複合バージョン用の互換キット	4-31
ローリング・アップグレード	1-7
OpenVMS Debugger	
Ada イベントのサポート	5-14
C++ 言語の問題	5-14
OpenVMS DELTA/XDELTA Debugger	
マニュアルの更新	3-16
OpenVMS Galaxy	4-36 ~ 4-38
および ES40	
Fast Path の無効化	4-38
非圧縮ダンプの制限事項	4-37
ライセンスの実行	6-7
OpenVMS Integrity	
DVD からのブート	1-12
OpenVMS Integrity 用 Linker	5-36 ~ 5-44
data-reduced ELF オブジェクト・ライブラリ	5-42
/EXPORT_SYMBOL_VECTOR の削除	5-43
LINK_ORDER セクション・ヘッダ・フラグ	5-41
OpenVMS Alpha 用 Linker との違い	5-41
/PUBLISH_GLOBAL_SYMBOLS の削除	5-43
オプションでの長いシンボル名	5-43
作成されたコード・スタブ	5-44
初期化されたオーバーレイ・プログラム・セクション	5-43
デマングル化されたシンボル名	5-44
OpenVMS Performance Management	
ドキュメントの訂正	3-15
OpenVMS Registry	
Version 2 フォーマットのデータベースの破壊	4-38
OpenVMS TCP/IP プロビジョニングの制限事項	4-6
OpenVMS システム・ダンプ・アナライザ	
CLUE コマンドは OpenVMS Integrity に移植されていない	5-45
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	5-25
Alpha システム上の	5-29
Integrity システム上の	5-28
/OPTIMIZE=VAXREGS 修飾子は OpenVMS Integrity ではサポートされない	5-29
浮動小数点数のゼロ除算エラーが検出されない (Integrity のみ)	5-30

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) 2-9
アップグレード後の再インストール (Alpha)
. 2-9
PCBST_TERMINAL
サイズの拡張 5-10
PCI 構成の制限事項 6-10
PEdriver
LAN 輻輳への対処 4-34
PGFLQUOTA の問題 5-33
PL/I
OpenVMS Integrity に含まれないライブラ
リ 5-45
RTL サポート 2-5
POOLCHECK システム・パラメータ 5-20
POSIX スレッド・ライブラリ 5-46 ~ 5-52
Integrity サーバでの THREADCP コマンドの動
作 5-49
pthread_mutex_lock 5-46
pthread_mutex_tryforcedlock_np 5-47
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 (Integrity のみ) 5-49
プロセス共有オブジェクトのサポート 5-46
例外処理中のスタック・オーバフロー (Integrity
のみ) 5-48
PowerStorm 300/350 PCI グラフィック・サポー
ト 6-21
Open3D のライセンスはチェックされな
い 6-21
pthread_mutex_lock
新しい戻り状態 5-46
pthread_mutex_tryforcedlock_np
API 5-47

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 5-23
rx7620 サーバ 1-8
rx8620 サーバ 1-8
RZnn ディスク・ドライブ 6-24 ~ 6-26

S

SAS テープ・ドライブの構成 4-12
SCSI コントローラ
AlphaServer 2100 システムでの制限事
項 6-4

SCSI デバイス・ドライバ 6-27
SCSI マルチパスの非互換性 4-33
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 4-36
SET PASSWORD コマンド 4-14
SHOW FORWARD/USER コマンド 3-2
SHOW LICENSE
/CHARGE_TABLE 3-11
/UNIT_REQUIREMENTS 3-11
SHUTDOWN.COM 4-15
small memory configurations 1-13
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 5-32, 5-54 ~ 5-55
SPLINVIPL バグ・チェック 5-12
SRM_CHECK ツール A-2
Superdome
sx1000 6-26
Superdome サーバ 1-8
sx1000 Superdome 6-26
sx1000 チップセット 1-8
SYSSGETTIM_PREC システム・サービス 3-1
SYSSSYSTEM:SHUTDOWN.COM コマン
ド 3-11
SYSGEN 4-26
SYSSTIMEZONE_RULE 論理名 4-1
SYSMAN 4-26
SYSTEM_CHECK システム・パラメータ 5-20
System Event Analyzer (SEA) ユーティリティ
I64 でのサポート 2-9
System Event Log (SEL)
Integrity サーバ上でのクリア 1-9

T

TCP/IP Services for OpenVMS 1-4
TCP/IP サーバ・コンポーネント
BIND, LPD, LBROKER, SMTP 4-6
Terminal Fallback Facility (TFF) 4-40
制限事項 4-41
TFF
Terminal Fallback Facility を参照
THREADCP コマンド
Integrity サーバでの動作 5-49
TIE キット 1-19
<time.h>の変更 5-16
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
TZ 関数 3-9, 4-27

U

U160 SCSI のサポート	
rx7620, rx8620	1-9
UCB 構造体	
アップデート	5-10
USB	
デバイス・サポート	6-2

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 connect	4-27
Virtual Connect	
フェールオーバー	4-27
Virtual I/O キャッシュ	
I64 では利用不可	4-42
XFC に置き換え	4-42
Volume Shadowing for OpenVMS	
互換キット	4-31

W

Watchpoint コーティリティ	5-56
WEBES	
I64 でのサポート	2-9

X

XA	4-29
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
---------------------	------

ア

アダプタについての注意事項	6-30
新しい戻り状態	
pthread_mutex_lock	5-46

イ

移行ソフトウェア	1-19
インストール・エラー	
HP Secure Web Browser	3-13
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	1-4
インターロックされたメモリ命令の使用	A-1

カ

回線切り替え	
性能の低下	4-34
外付け SAS ディスク・デバイス	4-13
外部認証	4-13
I64 サポート	4-14
SET PASSWORD コマンド	4-14
パスワードの有効期限切れの通知	4-14
書き込みビットマップ	4-7
書き込むメッセージ	3-7
拡張 DDT ビット	
修正された問題点	5-44
拡張ファイル・キャッシュ (XFC)	4-42
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	A-1, A-3
---------------	----------

ク

クラスタ	
OpenVMS Cluster システムを参照	
クラスタ互換キット	4-31
クラスタ互換性のためのパッチ・キット	4-30
グラフィック	
Integrity サーバのサポート	6-11
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	
ト	2-1
検証	5-5

コ

コンパイラ	
規則に準拠していないコード	A-1, A-6

サ

削除キー	
再マッピングが必要 (Integrity のみ)	6-3
サテライト・システム	4-26
サーバ	
rx7620	1-8
rx8620	1-8
Superdome	1-8

シ

- システム・ディスク
 - 古いシステムと互換性がない 1-5
- システムのクラッシュ
 - 回復 (I64 のみ) 4-28
- システムのハングアップ
 - 回復 (I64 のみ) 4-28
- システム・パラメータ 4-38 ~ 4-40
 - BUGCHECKFATAL 5-20
 - DEVICE_NAMING
 - デバイス・ユニット最大数の増加に使用 4-29
 - MMG_CTLFLAGS ドキュメントのエラ
 - 3-15
 - MULTIPROCESSING 5-20
 - POOLCHECK 5-20
 - SYSTEM_CHECK 5-20
 - 新しいパラメータ 4-38
 - 廃止されたパラメータ 4-39
 - 変更 4-40
- 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 4-31
- シリアル・ポートの名前 3-5
- シンボリック・デバッグ 5-1
- シンボリック・リンクの実装 3-2

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 5-11
 - 特権付きコードへの影響 5-11

セ

- 制限事項
 - SYSSLDDRIVER 4-25
- 性能 4-9
 - COBOL CALL 5-24
- 性能の改善
 - イメージの起動 4-9
- 性能の強化 4-7
 - Ctrl/T アライメント・フォルト 4-10
 - Dedicated CPU Lock Manager 4-10
 - グローバル・セクションの作成と削除 4-10
- セル型システム
 - 複数の nPartition 4-37

ソ

- 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) 5-55

テ

- ディスク・ボリュームのサポート
 - 制限事項 4-12
- デバイス構成 4-34
- デバイス・ドライバ
 - IPL の設定 6-28
 - MON バージョンの処理 6-28
 - SCSI 6-27
 - 再コンパイルと再リンク 6-27 ~ 6-28
 - スレッド単位のセキュリティの影響 6-28
- テープ・ロボット
 - 自動マルチパス・フェールオーバー 4-36
- 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 3-13
 - IPC コマンドの使用法 3-19
- ドキュメントの変更と訂正
 - Guide to OpenVMS File Applications 5-21
 - LIB\$ヘルプの漏れ 5-52
 - OpenVMS Performance Management 3-15
 - 『OpenVMS RTL Screen Management (SMGS) Manual』 5-53
- 特権データ構造体
 - CPU の名前空間 5-9
 - KPB 拡張 5-8
 - PCBST_TERMINAL のサイズの拡張 5-10
 - UCB と DDB のアップデート 5-10
 - スレッド単位のセキュリティの影響 5-11
 - 動的スピンロックのフォーク 5-9
 - 64 ビットの論理ブロック番号 5-9
 - 変更 5-7 ~ 5-11

ナ

- 長いデバイス名のサポート 4-11

ネ

- ネットワーク
 - アップデートの制限 3-20
- ネットワーク・オプション 1-4

八

バックポート・ライブラリ	5-16
パッチ・キット	
複合バージョンの OpenVMS Cluster システムで必要	4-31
パーティション	
ソフト	4-36
ハード	4-36
ハード・パーティション	4-36

ヒ

ビルトイン __fci の追加	5-17
-----------------	------

フ

ファームウェア	
Alpha サーバ用	1-14
EFI	4-6
Integrity サーバ用	1-10
複数の nPartition	
セル型システム	4-37
複数のサーバのプロビジョニング	4-5
浮動小数点型データ	
アプリケーションへの考慮	5-13
プログラム全体の浮動小数点モード (Integrity のみ)	5-56
プログラムの再コンパイル	
Alpha の場合	5-7
プロビジョニング	
OpenVMS	
HP SIM	4-4
システム・ファームウェア	4-5

へ

変換イメージ	
--------	--

AEST	3-1
------	-----

ホ

ポート・ドライバ \$QIO	
制限事項	5-44

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	
Fibre Channel テープ・デバイスの制限事項	4-35
テープ・ロボット	4-36

ラ

ライセンス	4-31
ライセンスについて	6-7~6-9
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 (OpenVMS Integrity での制限事項)	5-32
ランタイム・ライブラリ・ルーチン	3-18

レ

レイヤード・プロダクト	
インストールの失敗	1-20

ロ

ロケール	
新規	2-7
ローテートするレジスタ	5-52

HP OpenVMS V8.4 リリース・ノート【翻訳版】

2010年10月 発行

日本ヒューレット・パカード株式会社

〒102-0076 東京都千代田区五番町7番地

電話 (03)3512-5700 (大代表)

BA322-90102

