

HP OpenVMS Migration Software for Alpha to Integrity Servers

イメージ変換ガイド

2006年2月

本書では、Alpha Environment Software Translator (AEST) と他の OpenVMS Migration Software for Alpha to Integrity Servers ツールを使用して、OpenVMS Alpha アプリケーションを OpenVMS Industry Standard 64 システム向けに変換したり移植する方法を説明します。

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 Migration Software for Alpha to
 Integrity Servers Version 1.0
オペレーティング・システム: OpenVMS I64 Version 8.2

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

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

また、本書に記載されている事項は、予告なく変更されることがありますので、あらかじめご承知おきください。万一、本書の記述に誤りがあった場合でも、日本ヒューレット・パッカー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・パッカーは、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 またはその子会社の商標または登録商標です。

PostScript は、Adobe Systems Incorporated の登録商標です。

UNIX は、The Open Group の登録商標です。

Microsoft は、Microsoft Corporation の登録商標です。

原典: HP OpenVMS Migration Software for Alpha to Integrity Servers - Guide to Translating Images

目次

まえがき	vii
第 1 部 イメージ変換のユーザ・ガイド	
1 イメージ変換の概要	
1.1 OMSAI の概要	1-1
1.1.1 OMSAI の機能	1-2
1.1.2 移行計画での OMSAI の役割	1-2
1.2 イメージ変換のツールとサポート	1-3
1.2.1 Alpha Environment Software Translator ユーティリティ	1-3
1.2.1.1 Alpha イメージ情報ファイル	1-3
1.2.1.2 AEST の動作方法	1-4
1.2.1.3 コード分析	1-4
1.2.1.4 コード生成	1-5
1.2.2 Translated Image Environment	1-5
2 イメージの変換	
2.1 イメージを変換する前に	2-1
2.2 イメージを変換するための AEST の実行	2-1
2.2.1 AEST のリターン・ステータス	2-3
2.2.2 AEST の修飾子	2-3
2.2.3 AEST の出力ファイル	2-3
2.3 VESTed イメージの変換	2-4
3 変換済みイメージの実行	
3.1 変換済みイメージの実行	3-1
3.2 変換済みイメージへの参照の取り扱い	3-1
第 2 部 イメージ変換の開発者ガイド	
4 イメージの分析	
4.1 /AUDIT 修飾子の使用	4-1

5	イメージ情報ファイルの使用	
5.1	Alpha イメージ情報ファイル	5-1
5.2	AIIF ファイルの構文	5-1
6	OpenVMS Alpha の共用イメージの変換と置き換え	
6.1	相互運用の要件	6-1
6.1.1	/TIE 修飾子と/NONATIVE_ONLY 修飾子	6-1
6.1.2	上位互換性の維持	6-2
6.2	共用イメージ・バリエーションの構築手順	6-3
6.2.1	オリジナルの OpenVMS Alpha 共用イメージの構築	6-3
6.2.2	変換済み共用イメージの作成	6-4
6.2.2.1	変換済みメイン・イメージが、変換済み共用イメージを呼び出す	6-4
6.2.2.2	ネイティブのメイン・イメージが、変換済み共用イメージを呼び出す	6-5
6.2.3	置き換えを行うネイティブ共用イメージの構築	6-5

第3部 リファレンス情報

A コマンドの要約

B エラー・メッセージとステータス・メッセージ

B.1	AEST メッセージについて	B-1
B.2	AEST メッセージの説明	B-1

C 変換および性能上の制限事項

C.1	制限および性能上の問題の検出	C-1
C.2	変換できないイメージ	C-1
C.3	変換可能だが警告が出力されるイメージ	C-2
C.4	検出できない変換上の問題があるイメージ	C-2

例

2-1	イメージの変換	2-2
4-1	SIEVE.EXE の検査情報	4-1
A-1	要約のフォーマット	A-2

図

1-1	AEST の入力，処理，および出力	1-4
1-2	実行時のコンポーネント	1-6

表

2-1	AEST の出力ファイル	2-3
C-1	変換できないイメージ	C-1
C-2	変換可能だが警告が出力されるイメージ	C-2

対象読者

HP OpenVMS Migration Software for Alpha to Integrity Servers (OMSAI) を使用すると、OpenVMS Alpha イメージと、OMSAI によって Alpha 向けに変換された OpenVMS VAX イメージ (以降は VESTed イメージと呼びます) を同等の I64 イメージに変換できるため、OpenVMS Alpha アプリケーションと VESTed イメージを OpenVMS Industry Standard 64 (I64) システムに簡単に移行できます。OMSAI は、Alpha Environment Software Translator (AEST) ユーティリティと、変換作業を簡単にするように設計されたプログラムやコマンド・ファイルの集まりからなっています。『OpenVMS Migration Software for Alpha to Integrity Servers イメージ変換ガイド』(本書) では、AEST ユーティリティについて説明し、また OpenVMS Alpha アプリケーションを OpenVMS Industry Standard 64 システムへ移行する計画の一環としてこのユーティリティを使用する方法について説明しています。

本書は、次のユーザを対象としています。

- OpenVMS Industry Standard 64 システムへ移行する計画の一環として、OpenVMS Alpha アプリケーションのすべてまたは一部を変換するユーザ。
- OpenVMS Industry Standard 64 システム用の、変換済み共用イメージを開発するユーザ。

本書の構成

本書は、以下の 3 部構成になっています。

- 第 1 部: イメージ変換のユーザ・ガイド
第 1 部の情報は、すべてのユーザに関係します。
 - 第 1 章では、イメージ変換処理と、その処理をサポートするソフトウェアコンポーネントについて説明しています。
 - 第 2 章では、OpenVMS Alpha イメージを変換するユーティリティの使用方法について説明しています。
 - 第 3 章では、OpenVMS Industry Standard 64 システム上で、変換後のイメージを実行する方法について説明しています。
- 第 2 部: イメージ変換の開発者ガイド

第 2 部の情報は、変換済みイメージの性能を最適化する必要があるユーザ、ソース・コードにアクセスして編集を行い、変換を改善したり、OpenVMS Alpha システム上での再構築用にソース・ファイルを準備するユーザ、および変換済み共用イメージを用意するユーザに関係します。

- 第 4 章では、AEST の分析機能を使用して変換内容を強化したり、移行に影響のあるソースの問題点を見つける方法を説明しています。
- 第 5 章では、イメージ変換処理で AEST が作成し使用する、イメージ情報ファイルについて説明しています。
- 第 6 章では、OpenVMS I64 システム上のネイティブの共用イメージと相互運用する変換済み共用イメージの開発方法について説明しています。

- 第 3 部: リファレンス情報

第 3 部の情報は、すべてのユーザに関係します。

- 付録 A では、AEST のコマンド行と修飾子について、詳細を説明しています。
- 付録 B では、AEST のエラー・メッセージを、その説明と、推奨されるユーザ対処 (存在する場合) とともに、アルファベット順でリストしています。
- 付録 C では、変換の問題と、推奨する対処方法を説明しています。

第1部

イメージ変換のユーザ・ガイド

第1部には、次の情報が記載されています。

イメージ変換処理と、その処理をサポートするソフトウェア・コンポーネント	第1章
Alpha Environment Software Translator (AEST) ユーティリティによる OpenVMS Alpha イメージの変換	第2章
OpenVMS I64 システム上での変換済みイメージの実行	第3章

イメージ変換の概要

この章では、次のトピックについて説明します。

OMSAI の概要	第 1.1 節
OMSAI ツールとそのサポートについての説明	第 1.2 節

1.1 OMSAI の概要

HP OpenVMS Migration Software for Alpha to Integrity Servers (OMSAI) を使用すると、OpenVMS Alpha アプリケーションを OpenVMS Industry Standard 64 (I64) システムに簡単に移行できます。OMSAI コーティリティ、Alpha Environment Software Translator (AEST) は、OpenVMS Alpha の実行可能ファイルや共用イメージを、OpenVMS I64 システム上で動作する変換済みイメージに変換します。変換済みイメージを実行するときには、OpenVMS I64 システムは、OpenVMS Alpha システム上で実行しているかのような環境を用意し、透過的にイメージをサポートします。

イメージの変換と実行は、次の例のように簡単に行うことができます。

OpenVMS Alpha システム
\$ aest sieve 1

- 1 イメージの変換に使用する AEST コマンドです。このコマンドは、ファイル拡張子.EXE を前提としています。

OpenVMS I64 システム
\$ run sieve_av 1
Sieve of Eratosthenes 2
500 iterations
1899 primes found
time taken : 2 seconds

- 1 変換済みイメージの実行に使用する DCL コマンドです。OpenVMS Alpha システム上での SIEVE.EXE の実行にも、同じコマンドが使用されます。
- 2 変換済みイメージが表示した出力です。

OMSAI キットには、SYSSYSROOT:[SYSHLP.EXAMPLES.AEST]ディレクトリにインストールされる SIEVE.EXE イメージが含まれています。ご使用のシステムでこのコマンドを試してみてください。AEST は変換済みイメージを現在のディレクトリ内に作成し、入力イメージ・ファイル名に_AVを付加した名前を付けます。変換済みのバージョンの SIEVE.EXE は、SIEVE_AV.EXE という名前になります。

イメージ変換の概要

1.1 OMSAI の概要

AEST を使用すると、多数のイメージを簡単に変換できます。ただし、共用イメージの変換や、ユーザ作成または他社製の共用イメージにリンクされているイメージの変換を行う場合は、追加の手順が必要となることもあります。たとえば、変換に影響するような、Alpha のアーキテクチャや OpenVMS Alpha オペレーティング・システムに依存する部分がイメージに存在することもあります。場合によっては、AEST の修飾子を使用して、このような依存性に対処することもできます。この方法で対応できない場合は、ソース・ファイルが利用可能であればソース・ファイルを変更して再構築し、Alpha への依存性を無くさなければならないことがあります。

これ以降、この概要の節では、OMSAI の機能 (第 1.1.1 項) と、移行計画での OMSAI の役割 (第 1.1.2 項) について説明します。

1.1.1 OMSAI の機能

OMSAI には、以下の機能があります。

- 自動変換

OMSAI は、イメージを自動的に変換します。コードの分析と変換に、ユーザの操作を必要としません。

- イメージ分析

OMSAI はイメージを分析し、検出した情報を各種の形式 (メッセージおよびリスト) で報告します。検出した情報は、変換に役立つだけでなく、OpenVMS I64 システム上でオリジナルのソースを用意して、再コンパイルおよび再リンクするのも役立ちます。

AEST は、すべての OpenVMS Alpha イメージを変換できるわけではありません。いくつかの制限事項があります。たとえば、AEST は、Version 6.1 よりも前のバージョンでリンクされたイメージは変換できません。また、OpenVMS I64 システムで Alpha コードに対応するコードを正しく生成できないため、AEST は一部のコーディング方法をサポートしていません。サポートしていないコードを検出した場合、AEST はエラー・メッセージを出力します。報告される問題によっては、変換済みイメージが作成されないこともあります。

1.1.2 移行計画での OMSAI の役割

OpenVMS Alpha アプリケーションを OpenVMS I64 システムに移行するには、以下の方法を利用できます。

- アプリケーションのソース・ファイルから再構築する (推奨)。

イメージ変換の場合よりも I64 システムでの性能が良くなり、イメージが小さくなるため、ソース・ファイルの再コンパイルおよび再リンクによる再構築を行うことをお勧めします。ソース・ファイルと適切なコンパイラが利用可能な場合は、アプリケーションを再コンパイルして再リンクすることができます。

- アプリケーションの OpenVMS Alpha イメージを変換する。
アプリケーションのソースまたは適切なコンパイラが利用できない場合は、イメージを変換するしか方法がありません。
- ソース・ファイルからの再構築と、イメージの変換を組み合わせる。
一部のアプリケーションのソース・ファイルを再コンパイルし再リンクしてから、残りのアプリケーションのイメージを変換するのが、3 番目の移行方法です。この方法は、OpenVMS I64 システムが相互運用をサポートしているため可能です。つまり、OpenVMS I64 システムでは、ネイティブ・イメージと変換済みイメージが、互いに他方を呼び出したり、他方から呼び出されたりすることができます。
アプリケーションの再構築とイメージ変換を組み合わせる方法を使うと、移行方法を柔軟に計画することができます。

1.2 イメージ変換のツールとサポート

イメージ変換のツールとサポートには、次のものがあります。

- AEST ユーティリティ。主要な変換ツールです (第 1.2.1 項)。
- Translated Image Environment (TIE)。実行時に変換済みイメージをサポートする、OpenVMS I64 システムでのネイティブの共用イメージとその他のコンポーネントです (第 1.2.2 項)。

1.2.1 Alpha Environment Software Translator ユーティリティ

AEST ユーティリティは、実行可能イメージと共用イメージを変換します。このユーティリティは OpenVMS Alpha イメージ・ファイル (IMAGE.EXE) を入力として受け付け、このイメージ・ファイルを分析して Alpha コードを見つけてから、変換済みイメージ・ファイル (IMAGE_AV.EXE) を作成します。オリジナルと同じ機能を実行する変換済みイメージは、I64 コードと、オリジナルの OpenVMS Alpha イメージの両方からなる、OpenVMS I64 イメージです。ここでは、AEST がコードを見つけて分析するために使用するテキスト・ファイルについて説明してから、AEST の動作方法について簡単に説明します。

1.2.1.1 Alpha イメージ情報ファイル

Alpha イメージ情報ファイル (.AIIF ファイルともいう) は、イメージ変換時に使用される追加情報を AEST に提供するテキスト・ファイルです。Alpha イメージ情報ファイルには、AEST ユーティリティが外部参照 (プロシージャ呼び出しなど) を (多くの場合別の) 共用イメージ内の代替ルーチンにリダイレクトするために使用する情報が含まれています。AIIF ファイルは、共用イメージのエクスポートされたインタフェースのプロパティ、つまり、イメージのシンボル・ベクタに記載された正確な位置を示します。これらのファイルには、Alpha のエントリ・ポイントと I64 のエント

イメージ変換の概要

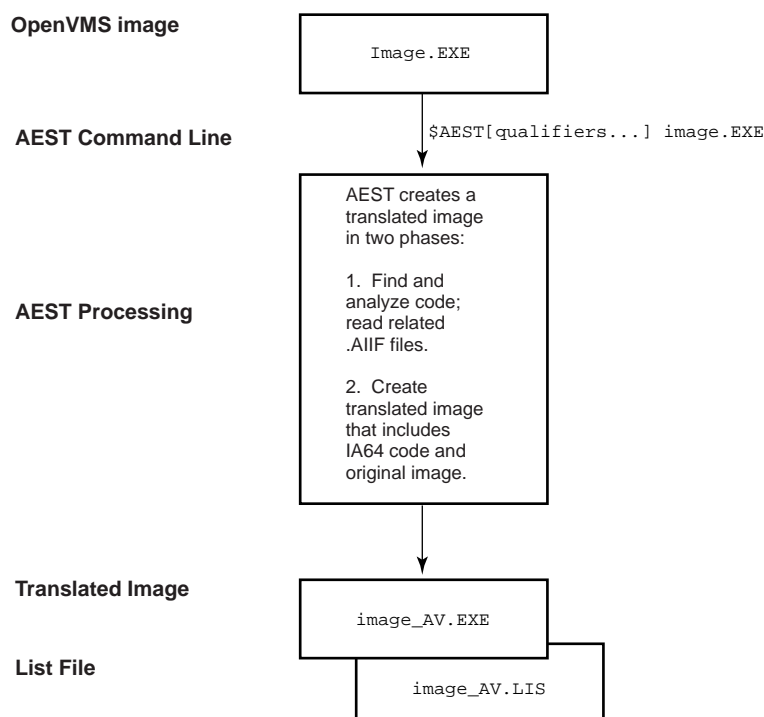
1.2 イメージ変換のツールとサポート

リ・ポイントの間のマッピングが含まれています。AESTがこの情報ファイルにアクセスする方法については、第5章で説明します。

1.2.1.2 AESTの動作方法

AESTは、OpenVMS Alpha イメージと同等のOpenVMS I64 イメージを生成するために、OpenVMS Alpha イメージを2つの主要なフェーズ(分析フェーズとコード生成フェーズ)で処理します。図1-1に、AESTの入力、処理、および出力を示します。

図 1-1 AESTの入力、処理、および出力



VM-1184A-AI

1.2.1.3 コード分析

分析フェーズでは、AESTは、入力イメージ・ファイルの分析を広範囲に渡って行って、エントリ・ポイントを見つけ、コードとデータを分離し、OpenVMS I64 環境では正しく再生成できない変則的なコードを検出します。分析時に見つからなかったコードは実行時にTIEが解釈しなければならないため、AESTは、可能な限り多くのコードを見つけようとします。

1.2.1.4 コード生成

変換の第 2 フェーズでは、変換済みイメージ (変換済みコードと、オリジナルの完全な OpenVMS Alpha イメージを含む I64 イメージ) が生成されます。変換済みコードは、オリジナルのイメージ内の、対応する Alpha コードと同じ機能を実行する、ネイティブの I64 コードです。変換済みイメージを OpenVMS I64 システム上で実行すると、オリジナルのイメージの動作が再現されます。

1.2.2 Translated Image Environment

Translated Image Environment (TIE) は、OpenVMS I64 システム上で変換済みイメージを実行するために必要なリソースを提供します。各種のコンポーネントが共同で動作し、変換済みイメージの実行をサポートします。

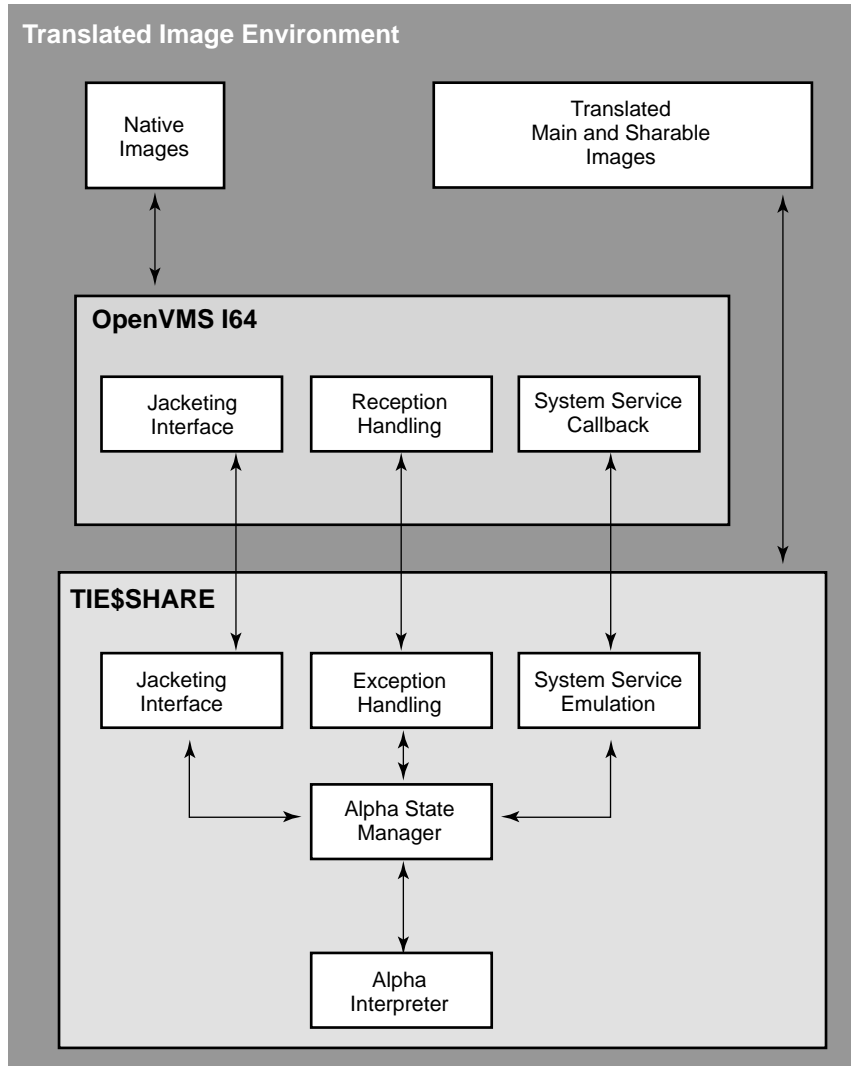
- 変換済みイメージ自身。オリジナルの OpenVMS Alpha イメージと、変換済みコードが含まれています。変換済みコードには、AEST が挿入した、TIE\$SHARE への呼び出しが含まれています。これらの呼び出しは、OpenVMS I64 システムにネイティブでない機能の処理を起動します。
- TIE\$SHARE。OpenVMS I64 の共用イメージです。TIE\$SHARE には、変換済みイメージが OpenVMS Alpha システム上で動作しているかのように変換済みイメージを実行させる機能が用意されています。TIE\$SHARE には、以下の機能があります。
 - オリジナルの Alpha コードと、変換済みコードとの関係を定義する、Alpha の状態情報とその他の情報を管理します。
 - 例外処理など、変換済みイメージが必要とする OpenVMS Alpha の機能を実現します。
 - AEST が変換しなかった Alpha コードを解釈実行します。
- 変換済みイメージが使用する、一部の OpenVMS Alpha 実行時ライブラリの変換済みのバージョン。
- I64 ネイティブのライブラリと変換済み Alpha イメージ間のジャケット・ライブラリ。変換済みイメージからの非標準の呼び出しを I64 ネイティブの実行時ライブラリに取り次ぎます。
- 例外処理の実行、AST の配信、変換済みイメージとネイティブ・イメージとの間の通信などを可能にするために、TIE\$SHARE と協調して動作する OpenVMS I64 オペレーティング・システムのその他の機能。

自動ジャケット機能 (第 6 章を参照) には、変換済みイメージとネイティブ・イメージの間の、大半の通信の相互運用メカニズムが用意されています。この機能には、Alpha の呼び出し規則と I64 の呼び出し規則の間のブリッジが用意されています。

図 1-2 に、実行時のコンポーネントの相互関係を示します。

イメージ変換の概要
1.2 イメージ変換のツールとサポート

図 1-2 実行時のコンポーネント



VM-1183A-AI

イメージの変換

この章では、次のトピックについて説明します。

イメージを変換する前に考慮しなければならない事項	第 2.1 節
イメージを変換するための AEST の実行	第 2.2 節

他の章にも、関連するトピックがあります。

共用イメージを変換する際に特別に考慮する事項	第 6 章
変換済みイメージの実行	第 3 章

2.1 イメージを変換する前に

すべてのイメージが変換できるわけではありません。変換できないイメージの例としては、特権コードを含むイメージや、特権モードで実行しなければならないイメージ、AEST でサポートされていない言語で作成されたコードを含むイメージがあります。イメージを変換できるかどうかを確認するには、イメージの変換を実際に行う前に、`/AUDIT` 修飾子を指定して AEST を実行してください。`/DUMP` 修飾子を追加すると、イメージの内部構造に関する追加情報が含まれているリスト・ファイルが生成されます。

`/AUDIT` 修飾子についての詳細は、付録 A の AEST についての説明を参照してください。

2.2 イメージを変換するための AEST の実行

次のコマンドは、OpenVMS Alpha の実行可能イメージや共用イメージを変換します。

```
AEST[/qualifier,...] image[.EXE]
```

`image` は、変換する OpenVMS Alpha イメージのファイル名です。省略時の拡張子は、`.EXE` です。

第 2.2.2 項では、AEST の修飾子について説明しています。変換が成功すると、AEST は変換済みイメージを現在のディレクトリに作成し、次のように、入力ファイル名に `_AV` を付加した名前を付けます。

イメージの変換

2.2 イメージを変換するための AEST の実行

image_AV.EXE

注意

ファイル名の長さは、39 文字を超えることはできません。この制限事項があるため、入力イメージ・ファイル名の長さが 36 文字を超えるときには、AEST は_AV という文字列を付加するために入力イメージ・ファイル名を短縮します。

AEST は、ERROR レベルまたは FATAL レベルのメッセージを出力するエラーを検出すると、変換済みイメージを作成しません。このような場合、変換が成功しなかった理由を示すメッセージが出力されます。

例 2-1 に、成功した、DHRYSTONE.EXE というイメージの変換を示します。

例 2-1 イメージの変換

```
$ directory/brief 1
Directory AST_00:[AEST.TEST]
DHRYSTONE.EXE;1
Total of 1 file.
$ aest dhrystone 2
$ directory/brief 3
Directory AST_00:[AEST.TEST]
DHRYSTONE.EXE;1      DHRYSTONE_AV.EXE;1
DHRYSTONE_AV.LIS;1
Total of 3 files.
$ type dhrystone_av.lis 4
AEST XA29 DEV_0.05A (May 7 2004) starting at May 14 42004 18:28:06 with command line: 5
AEST DHRYSTONE.EXE
%AEST-I-TRANSOK, Translation completed successfully
```

例 2-1 の説明は、次のとおりです。

- 1 DHRYSTONE.EXE という名前の OpenVMS Alpha イメージの、簡単なディレクトリ・リストです。
- 2 DHRYSTONE.EXE を変換する AEST コマンド行です。
- 3 オリジナルのイメージと、AEST が作成した新しい 2 つのファイルを示す、簡単なディレクトリ・リストです。
 - DHRYSTONE_AV.EXE— 変換済みイメージ
 - DHRYSTONE_AV.LIS— リスト・ファイル
- 4 DHRYSTONE_AV.LIS ファイルを表示するコマンドです。このリスト・ファイルには、最初に AEST のバージョン、変換の日時、実行したコマンド行、および変換したイメージのヘッダ情報が出力されます。

- 5 変換中に出力された、AEST メッセージの要約です。この要約は、メッセージを次のように分類します。
 - デフォルトで表示される、AEST の標準のメッセージ。
 - 変換中に AEST の進行状況を報告する詳細メッセージ。

第 2.2.2 項に説明されている AEST の修飾子を使用すると、AEST がイメージの変換と分析を行う方法を調整できます。

2.2.1 AEST のリターン・ステータス

AEST は実行を終えると、以下のいずれかのメッセージを終了ステータスとして DCL に返します。

- TRANSOK。変換が正常に終了しました。
- TRANSWARN。警告付きで変換が完了しました。出力されたイメージを使用する前に、警告メッセージを見直してください。
- TRANSERROR。変換が失敗しました。出力イメージは作成されませんでした。
- TRANSFATAL。変換できませんでした。

リターン・ステータスは、動作中に AEST が出力したすべてのメッセージのうち、最も高い重大度レベル (INFO, WARNING, ERROR, または FATAL) を示します。

2.2.2 AEST の修飾子

AEST コマンドには、処理をさまざまな方法で制御する修飾子があります。AEST コマンド行と各修飾子についての詳細は、付録 A を参照してください。

2.2.3 AEST の出力ファイル

AEST は、変換するイメージのタイプと、コマンド行で指定した修飾子に従って、いくつかのタイプの出力ファイルを生成します。表 2-1 に、出力ファイルの一覧を、省略時の名前、制御用の AEST 修飾子とともに示します。

表 2-1 AEST の出力ファイル

省略時のファイル名	修飾子	説明
image_AV.EXE	/EXECUTABLE[=filespec] 省略時の指定: /EXECUTABLE	変換済みイメージ。AEST は、入力イメージ・ファイル名が 36 文字を超えているときには、_AV を付加するために短縮します。

(次ページに続く)

イメージの変換

2.2 イメージを変換するための AEST の実行

表 2-1 (続き) AEST の出力ファイル

省略時のファイル名	修飾子	説明
image_AV.LIS	/LIST[= filespec] 省略時の指定: /LIST	リスト・ファイル。

2.3 VESTed イメージの変換

VESTed イメージ (オリジナルが OpenVMS VAX イメージで、以前に OMSVA Migration Software パッケージの VEST ユーティリティで OpenVMS Alpha イメージに変換したイメージ) の変換は、ネイティブの OpenVMS Alpha イメージの変換と似ています。VESTed アプリケーションの変換には、追加情報ファイルも、AEST ユーティリティのスイッチも必要ありません。

変換済みイメージの実行

この章では、次のトピックについて説明します。

変換済みイメージの実行

第 3.1 節

変換済みイメージへの参照の取り扱い

第 3.2 節

3.1 変換済みイメージの実行

変換済みイメージは通常、他の OpenVMS イメージと同じように、OpenVMS I64 オペレーティング・システムで実行できます。OpenVMS オペレーティング・システムには、必要なコンポーネントがすべてあります。変換済みイメージを実行するには、DCL の RUN コマンドを使用します。たとえば、サンプル・プログラム SIEVE_AV.EXE (第 1 章で変換) を実行するには、次のコマンドを入力します。

```
$ run sieve_av
Sieve of Eratosthenes
500 iterations
1899 primes found
time taken : 2 seconds
```

Translated Image Environment (TIE) (第 1.2.2 項を参照) は、変換済みイメージの実行中にエラーを検出すると、エラー・メッセージを出力します。

3.2 変換済みイメージへの参照の取り扱い

変換済みイメージの起動方法によっては、名前を image.EXE から image_AV.EXE に変更しなければならないことがあります。

RUN コマンドを使用する場合は、第 2.2 節のように、変換済みイメージ名だけを指定します。イメージ名がコマンド言語定義 (CLD) ファイル内で指定されている場合は、CLD ファイル内のイメージ名を変更するか、次の例のように、古い名前が新しい名前を指すようにする論理名を定義します。

```
$ DEFINE MYMAIN YOUR$DISK:[YOUR_DIR]MYMAIN_AV.EXE;
```

フォーリン・コマンド・シンボルを使用してイメージを起動している場合は、変換済みイメージ名を指定するようにシンボル定義を変更します。

変換済みイメージの実行

3.2 変換済みイメージへの参照の取り扱い

変換済みアプリケーションに、SYS\$SHARE に置かれていない共用イメージが含まれている場合、これらのイメージを正しく指す(つまり、正しい位置と変換済みイメージ名を反映する)論理名を定義しなければなりません。例を次に示します。

```
$ DEFINE MYMATH_AV YOUR$DISK:[YOUR_DIR]MYMATH_AV.EXE;
```

注意

弊社が用意した、SYS\$SHARE 内の共用イメージとリンクされている変換済みイメージを実行すると、OpenVMS I64 システムは正しいイメージを自動的に起動します。共用イメージの論理名を再定義する必要はありません。弊社が用意したものでない変換済み共用イメージを SYS\$SHARE 内に置く場合は、適切な論理名を明示的に定義しなければなりません。

場合によっては、一部の外部参照を、別の共用イメージに振り向けることもできます。この場合、独自の.AIIF ファイルを作成しなければなりません。.AIIF ファイルのフォーマットについては、第5章を参照してください。

第2部

イメージ変換の開発者ガイド

第2部には、次の情報が記載されています。

変換済みイメージの性能改善。OpenVMS Alpha オペレーティング・システム用にアプリケーションを再構築する方法に影響するソース・コードの問題の検出。	第4章
イメージ情報ファイルの使用	第5章
変換済みイメージの作成とネイティブの共用イメージの作成	第6章

この章では、/AUDIT 修飾子を使用して、アプリケーションの変換と再構築に影響するイメージの特性を知る方法について説明します。

4.1 /AUDIT 修飾子の使用

/AUDIT 修飾子は、イメージを分析し、アプリケーションの移行方法を決定する上で役立つ、簡単な評価の要約を表示するように AEST に指示します。AEST の分析中に出力されるエラー・メッセージをベースとするこの要約は、次の質問に対する回答となります。

- ソースが利用可能であれば、このイメージを OpenVMS Industry Standard 64 システム上で再コンパイルして再構築できますか?
- このイメージを変換できますか?
- 使用されているソース言語は何ですか?

/AUDIT 修飾子についての詳細は、付録 A を参照してください。この説明には、一連の AEST/AUDIT コマンドを実行して要約説明のファイルを作成してから、DCL コマンドを使用して、要約説明を抽出し編集する手順が含まれています。

例 4-1 に、SIEVE.EXE プログラムの検査でのリスト・ファイルを示します。

例 4-1 SIEVE.EXE の検査情報

```
$ AEST/AUDIT SIEVE.EXE
AEST (V1.0, Bld DEV_0.8/Feb 21 2005)

$ TYPE SIEVE_AV.LIS
AEST (V1.0, Bld DEV_0.8/Feb 21 2005)
Image "SIEVE", "V1.0", 24-FEB-2005 15:29:40.96
<SUM> Image name                               Tran Languages
<SUM> -----
<SUM> $1$DGA120:[VERIFICATION.BINTRAN.EXAMPLES]SIEVE.EXE   YES  C
$
```

イメージ情報ファイルの使用

この章では、Alpha イメージ情報ファイル (.AIIF) について説明します。

5.1 Alpha イメージ情報ファイル

Alpha イメージ情報ファイルは、イメージの変換中に使用する追加情報を AEST に提供するテキスト・ファイルです。AEST は、現在変換中の Alpha ファイルのイメージ参照の名前を変更するために、この .AIIF ファイル内の情報を使用します。このようなファイルは、変換済みイメージを異なる共用イメージに再リンクしたり、フィックスアップ情報を変更するために使用されます。

.AIIF ファイルには、イメージのシンボル・ベクタに記述されている正確な位置情報である、共用イメージのエクスポート済みインタフェースのプロパティが含まれています。これらのファイルには、Alpha のエントリ・ポイントと Itanium のエントリ・ポイントのマッピングが含まれています。

5.2 AIIF ファイルの構文

ファイル内のどこにでも、C スタイルの複数行のコメントを記述できます。

ファイル内の各行のフィールド間では、次のデリミタを使用できます。

- ''
- '\t'
- ','

.AIIF ファイルでは、次の 2 つのタイプの行を使用できます。

1. 1 つ目のタイプの行は、リンクされているライブラリの古いシンボル・ベクタ・インデックスを、新しいシンボル・ベクタ・インデックス (他のライブラリ内に存在する可能性が高い) に再割り当てするために必要です。この目的には、次の形式を使用します。

```
old_symv_idx new_symv_idx [, "new_name"]
```

イメージ情報ファイルの使用

5.2 AIIF ファイルの構文

引数の意味は、次のとおりです。

引数	説明	使用方法
old_symv_idx	古いシンボル・ベクタ・インデックス	(10 進数)
new_symv_idx	新しいシンボル・ベクタ・インデックス	(10 進数)
new_name	新しくリンクされたライブラリ (オプション)	(二重引用符 (" ") で囲まれた ASCII 文字列)

2. 2 つ目のタイプの行は、新しくリンクされたライブラリに新しい GSMATCH フィールドを割り当てるために必要です。この目的には、次の形式を使用します。

```
"name" match_ctl minor_id major_id
```

引数の意味は、次のとおりです。

引数	説明	使用方法
name	新しくリンクされたライブラリの名前	(二重引用符 (" ") で囲まれた ASCII 文字列)
match_ctl	マッチ処理	(10 進数)。イメージ・アクティベータが使用するマッチ・アルゴリズムを示します。次のいずれかの値を指定します。 0 - ALWAYS 1 - EQUAL 2 - LEQUAL
minor_id	マイナー ID	(10 進数)
major_id	メジャー ID	(10 進数)

OpenVMS Alpha の共用イメージの変換と置き換え

この章では、次のトピックについて説明します。

相互運用の要件	第 6.1 節
OpenVMS Alpha の共用イメージの変換と置き換えの手順	第 6.2 節

6.1 相互運用の要件

OpenVMS Industry Standard 64 システムでは、変換済みイメージとネイティブ・イメージが互いに他方を呼び出すことにより、相互運用が可能です。呼び出しは、Alpha (または VAX) 呼び出し規則と I64 呼び出し規則との間で必要な変換を実行する、システム・ジャケット・ルーチンを通して行われます。変換済みイメージを呼び出したり、変換済みイメージと通信するネイティブ・イメージを作成するときは、特定のリンク修飾子とコンパイル修飾子を使用する必要があります。さらに、OpenVMS Alpha イメージを置き換えるネイティブの共用イメージを作成するときは、そのイメージとの互換性を維持しなければなりません。

この章では、Alpha の共用イメージの例として MYMATH という C のテスト・プログラムを使用し、MYMATH を呼び出すメイン・イメージの例として MYMAIN という C プログラムを使用します。これらのプログラム例は、変換済みイメージを呼び出したり、変換済みイメージから呼び出されるネイティブ・イメージを作成する方法を説明する上で役立つ、以下の相互運用の要件に重点を置いています。

- コンパイラの修飾子/TIE と、リンカの修飾子/NONATIVE_ONLY を使用して、変換済みイメージと相互運用できるネイティブ・イメージを作成すること。
- Alpha の共用イメージとそれを置き換えるネイティブの共用イメージの間でシンボル・ベクタの上位互換性を確保すること。

6.1.1 /TIE 修飾子と/NONATIVE_ONLY 修飾子

/TIE 修飾子と/NONATIVE_ONLY 修飾子は、それぞれコンパイラとリンカに、OpenVMS I64 システムが変換済みイメージとの間でジャケット呼び出しができるようにするコードを含めるように指示します。/TIE 修飾子を指定すると、コンパイラは、変換済みイメージとネイティブ・イメージとの間のジャケット呼び出しを正しく行うために Translated Image Environment (TIE) が必要とする、プロシージャ・シグネチャ・ブロック (PSB) を作成します。LINK コマンドに/NONATIVE_ONLY 修飾子を指定すると、リンカは、コンパイラが作成した PSB 情報をイメージ内に含めます。これらの修飾子による相互運用は、省略時の設定ではないことに注意してくだ

さい。/TIE と/NONATIVE_ONLY はユーザが明示的に指定しなければなりません。ユーザがこれらの修飾子を指定しない限り、変換済みイメージはネイティブ・イメージと相互運用することはできません。

これらの修飾子についての詳細は、コンパイラの適切なドキュメントや、OpenVMS Linker Utility Manual を参照してください。

6.1.2 上位互換性の維持

同じ共用イメージの、次に続くバージョンでは、呼び出し元イメージを再リンクしなくてよいように、同じ呼び出しインタフェースを維持して、上位互換とする必要があります。OpenVMS Linker Utility Manual では、OpenVMS Alpha と OpenVMS I64 のシステム・イメージのシンボル・ベクタを使用して上位互換性を維持する方法について説明しています。以下のイメージを作成する場合にも、上位互換性を維持してください。

- 変換済み共用イメージ
- OpenVMS Alpha の共用イメージを置き換えるネイティブ・イメージ
- 変換済み共用イメージを置き換えるネイティブ・イメージ

同じイメージの次に続くバージョンを構築するときは一貫性のあるシンボル・ベクタを作成するのと同じように、同じ目的を達成するために、変換済みイメージとネイティブの置き換えイメージとで一貫性のあるシンボル・ベクタを作成します。これにより、OpenVMS I64 システムに移行するときに、上位互換性が確保されます。ネイティブの置き換えイメージをリンクするときに、オリジナル・イメージの転送ベクタに一致するように、シンボル・ベクタを構築します。ネイティブ・イメージに新しいルーチンが含まれている場合、オリジナルの順序を変えないように、これらのルーチンのシンボル・ベクタ・エントリを古いルーチンのエントリの後に置きます。オリジナルの順序を維持するシンボル・ベクタの作成には、どのようなメカニズムを使用してもかまいません。

シンボル・ベクタが同じエントリ順序を維持していない場合は、呼び出しイメージが中断する恐れがあります。AEST は、あるバージョンのイメージを変換するときに正しい順序のシンボル・ベクタを作成するかもしれませんが、それ以降のバージョンのイメージの変換では正しい順序のシンボル・ベクタを作成しないかもしれません。別の見方をすれば、ネイティブの置き換えイメージをオリジナルの順序と無関係に作成すると、ルーチンが予期したアドレスにないため、変換済みイメージがその置き換えイメージを呼び出せない可能性があります。すべてのルーチンをネイティブ・モードで再生成できないために、ジャケット・イメージを作成しなければならない場合は、新しいエントリ順序に合わせるために、ジャケット・イメージを再構築しなければならないことがあります。この処理は複雑なので、お勧めできません。この章で説明する手順を代わりに使用してください。

転送ベクタ, シンボル・ベクタ, および互換性についての詳細は, OpenVMS Linker Utility Manual と VAX MACRO and Instruction Set Reference Manual を参照してください。

6.2 共用イメージ・バリエーションの構築手順

この節では, 以下の作業の手順を説明します。

- オリジナルの OpenVMS Alpha 共用イメージの構築 (第 6.2.1 項)。
- 変換済み共用イメージの作成 (第 6.2.2 項)。
- ネイティブの置き換えイメージの構築 (第 6.2.3 項)。

OMSAI キットには, この節で使用するプログラム例すべてと, そのプログラムを構築し実行するためのコマンド・ファイルが含まれています。キットのインストール後, ソース・ファイルとコマンド・ファイルは, SYS\$EXAMPLES の AEST サブディレクトリに置かれています。コマンド・ファイルは, 次の順序で実行しなければなりません。

1. 最初に, OpenVMS Alpha システム上で BUILD_MYMATH_AXP.COM を実行します。HP C コンパイラが必要です。
2. 次に, OpenVMS I64 システム上で BUILD_MYMATH_IA64.COM を実行します。HP C コンパイラが必要です。

手順の説明は, すべて同じ形式になっています。

- 作成するイメージ・タイプの説明 (たとえば, 変換済み共用イメージを呼び出す変換済みメイン・プログラム)。
- 手順を示すコマンドまたはコード・シーケンス。
- 必要であれば, コマンドやコード・シーケンスが何を行っているかの説明。

注意

これらの手順は, 相互運用可能なイメージを作成する基本的なステップの簡単な例であり, すべてのケースを考慮しているわけではありません。この手順は, アプリケーションの要件に合わせて変更する必要があります。

6.2.1 オリジナルの OpenVMS Alpha 共用イメージの構築

最初に, OpenVMS Alpha の共用イメージを作成し, 構築します。

OpenVMS Alpha の共用イメージの変換と置き換え

6.2 共用イメージ・バリエーションの構築手順

```
SYMBOL_VECTOR=( myadd=PROCEDURE,-  
mysub=PROCEDURE,-  
mydiv=PROCEDURE,-  
mymul=PROCEDURE)  
GSMATCH=LEQUAL,2,0  
[EXIT]
```

MYMATH をコンパイルしてオブジェクト・ファイルに AXP_MYMATH という名前を付け、それをリンクして共用イメージを作成します。

次に、OpenVMS Alpha のメイン・イメージを作成します。

```
$ CC MYMAIN/OBJ=AXP_MYMAIN.OBJ  
$ LINK AXP_MYMAIN.OBJ,SY$INPUT/OPTIONS  
AXP_MYMATH/SHAREABLE  
[EXIT]
```

MYMAIN をコンパイルしてオブジェクト・ファイルに AXP_MYMAIN という名前を付け、それを共用イメージ AXP_MYMATH とリンクします。

次に、論理名を定義してメイン・イメージを実行します。

```
$ DEFINE AXP_MYMATH YOUR$DISK:[YOUR_DIR]AXP_MYMATH.EXE;  
$ RUN/NODEBUG AXP_MYMAIN
```

論理名 AXP_MYMATH を定義して、AXP_MYMATH.EXE の位置を指すようにします。

6.2.2 変換済み共用イメージの作成

この項では、OpenVMS Alpha システムで行う 2 つの手順について説明します。

- 変換済み共用イメージを作成し、そのイメージを呼び出す、変換済みメイン・イメージを作成する (第 6.2.2.1 項を参照)。
- 変換済み共用イメージを作成し、そのイメージを呼び出すネイティブ・メイン・イメージを作成する (第 6.2.2.2 項)。

6.2.2.1 変換済みメイン・イメージが、変換済み共用イメージを呼び出す

OpenVMS Alpha システム上の共用ライブラリを使用するメイン・イメージがあるとします。次のコマンドを実行して、両者を変換します。

```
$ AEST AXP_MYMAIN  
$ AEST AXP_MYMATH
```

変換済みイメージは、それぞれ AXP_MYMAIN_AV.EXE、AXP_MYMATH_AV.EXE という名前になります。OpenVMS I64 システムは省略時のディレクトリで AXP_MYMATH_AV.EXE を探するため、AXP_MYMAIN_AV を OpenVMS I64 システム上で単純に実行すると、失敗します。

イメージの変換後は、AXP_MYMATH_AV.EXE が存在する実際の場所を指す論理名を定義する (推奨) か、変換済みイメージを省略時のディレクトリのひとつにコピーします。

AXP_MYMATH_AV.EXE の位置の論理名を定義するには、次のコマンドを入力します。

```
$ DEFINE AXP_MYMATH_AV YOUR$DISK:[YOUR_DIR]AXP_MYMATH_AV.EXE
```

これで、変換済みイメージを実行できます。

```
$ RUN AXP_MYMAIN_AV
```

6.2.2.2 ネイティブのメイン・イメージが、変換済み共用イメージを呼び出す

この手順は、OpenVMS Alpha 共用イメージを変換する方法と、そのイメージを呼び出すネイティブのメイン・イメージを作成する方法を示しています。

OpenVMS Alpha の共用イメージを変換するには、次のコマンドを実行します。

```
$ AEST AXP_MYMATH
```

IA64_MYMAIN という名前で、メイン・イメージ MYMAIN のネイティブ・バージョンを作成します。ソースをコンパイルするときには /TIE 修飾子を指定し、オブジェクト・ファイルをリンクするときには /NONATIVE_ONLY 修飾子を指定します。IA64_MYMAIN とネイティブの共用イメージをリンクするときと同じ方法で、IA64_MYMAIN と AXP_MYMATH_AV をリンクします。例を次に示します。

```
$ CC/TIE MYMAIN/OBJ=IA64_MYMAIN  
$ LINK/NONATIVE_ONLY IA64_MYMAIN, SYS$INPUT:/OPTIONS  
AXP_MYMATH_AV.EXE/SHAREABLE  
[EXIT]
```

AXP_MYMATH_AV.EXE の位置を指すように、論理名 AXP_MYMATH_AV を定義します。IA64_MYMAIN を実行すると、AXP_MYMATH_AV が正しく呼び出されます。例を次に示します。

```
$ DEFINE AXP_MYMATH_AV YOUR$DISK:[YOUR_DIR]AXP_MYMATH_AV.EXE;  
$ RUN IA64_MYMAIN
```

6.2.3 置き換えを行うネイティブ共用イメージの構築

/TIE 修飾子を指定してネイティブ I64 共用イメージをコンパイルし、/NONATIVE_ONLY 修飾子を指定してリンクします。オリジナルの OpenVMS Alpha 共用イメージの順序に従ってエントリを順序付ける、リンカ・オプション・ファイルをインクルードします。例を次に示します。

OpenVMS Alpha の共用イメージの変換と置き換え

6.2 共用イメージ・バリエーションの構築手順

```
$ CC/TIE MYMATH/OBJ=IA64_MYMATH
$ LINK/SHAREABLE/NONATIVE_ONLY IA64_MYMATH, SYS$INPUT:/OPTIONS
SYMBOL_VECTOR=( myadd=PROCEDURE,-
mysub=PROCEDURE,-
mydiv=PROCEDURE,-
mymul=PROCEDURE)
GSMATCH=LEQUAL,2,0
[EXIT]
```

次に、論理名を定義して、メイン・イメージを実行します。

```
$ DEFINE AXP_MYMATH_AV YOUR$DISK:[YOUR_DIR]IA64_MYMATH.EXE;
$ RUN/NODEBUG AXP_MYMAIN_AV
```

第3部

リファレンス情報

第3部には、次の情報が含まれています。

AEST コマンド行と修飾子の詳細説明	付録 A
すべての AEST メッセージのアルファベット順のリスト (説明と推奨する処置を含む)	付録 B
変換上の問題の検出	付録 C

コマンドの要約

AEST

AEST ユーティリティは、OpenVMS Alpha の実行可能イメージと共用イメージを、機能が同等の OpenVMS I64 システム・イメージに変換します。AEST を使用すると、OpenVMS Alpha イメージを分析し、変換の可能性を評価することもできます。

形式

```
AEST[/qualifier,...] image[.EXE]
```

修飾子	省略時の値
/AIIF	なし
/AUDIT	なし
/DEBUG	/NODEBUG
/DUMP	/NODUMP
/EXECUTABLE	/EXECUTABLE
/INTERPRET	/NOINTERPRET
/LIST	/LIST
/VERBOSE	/NOVERBOSE

/AIIF

現在のイメージを変換するときに、指定された Alpha イメージ情報ファイル (AIIF) を使用するように AEST に指示します。通常、AEST は、Alpha イメージの名前にサフィックス_AV を付加して、変換済みイメージの名前とします。/AIIF 修飾子を使用すると、この動作を変更できます。

省略時の値	なし
形式	/[NO]AIIF=(filespec[,filespec...])
修飾子の値	filespec[,filespec...]

説明

Alpha イメージ情報ファイル (.AIIF ファイル) には、現在変換中の Alpha ファイル内のイメージ参照の名前を変更するために AEST が使用する情報が含まれています。このようなファイルは、変換済みイメージを異なる共用イメージに再リンクしたり、フィックスアップ情報を変更するために使用されます。

.AIIF ファイルには、共用イメージのエクスポート済みインタフェースのプロパティ (イメージのシンボル・ベクタに記述されている正確な位置) が記述されています。こ

コマンドの要約

これらのファイルには、Alpha のエントリ・ポイントと I64 のエントリ・ポイントのマッピングが含まれています。

/NOAIIF 修飾子は、指定された AIIF ファイルが存在しても使用しないように、AEST に指示します。

AEST は AIIF ファイルを以下の場所で、この順番で検索します。

1. AEST コマンド行の/AIIF 修飾子の値として指定されたディレクトリ。
2. 論理名で指定された場所。各 AIIF ファイルの論理名は、共用イメージ・リストと、拡張子 (.AIIF) から適切に作成されていなければなりません。たとえば、動的イメージ SYS\$PUBLIC_VECTORS に対して、適切な論理名は SYS\$PUBLIC_VECTORS.AIIF です。
3. 現在のディレクトリ。
4. SYS\$LIBRARY 論理名が指すディレクトリ。

/AUDIT

イメージの変換は行わずに、入力イメージを分析して、その移行特性の要約を表示するように AEST に指示します。特に指定しなくても、AEST はイメージを変換する前に検査を実行しますが、リスト・ファイルには詳細情報を出力しません。

形式	/AUDIT
修飾子の値	なし

説明

要約には、次の情報が表示されます。

- イメージが変換可能かどうか。
- ソース言語、つまりイメージの言語 (分かっている場合)。

検査の要約は、1 行の説明であり、そのイメージに対して選択する移行方法を決定する上で役立つ、イメージの移行特性を示します。要約情報は、リスト・ファイル内のすべての他のメッセージの後にあります。例 A-1 に、要約のフォーマットを示します。

例 A-1 要約のフォーマット

```
<SUM> Image name                               Tran Languages
<SUM> -----
<SUM> AST_00:[GROUP.TEST]DHRYSTONE.EXE         YES C
```

左側のイメージ名の欄には、入力イメージの完全なファイル指定が表示されます。

右側の 2 つの欄には、イメージの移行特性が表示されます。

Tran	イメージが変換可能かどうか YES または NO で示されます。
Languages	イメージのデバッグ・シンボル・テーブル (DST) で識別されるソース言語がリストされます。この言語は、分析対象のイメージにだけ関係し、このイメージが呼び出す共用イメージには関係しません。

一連の AEST/AUDIT コマンドを実行してから、OpenVMS コマンドを使用して要約情報を抽出することにより、要約説明のファイルを作成できます。例を次に示します。

```
$ AEST/AUDIT IMAGE1
$ AEST/AUDIT IMAGE2
.
.
.
$ AEST/AUDIT IMAGE n
$ SEARCH/OUTPUT=TEMP.1 *.LIS "<SUM>"
$ SEARCH/OUTPUT=TEMP.2 TEMP.1 "<SUM>"
$ SORT/NODUPLICATE TEMP.2 TEMP.3
$ PRINT TEMP.3
```

/AUDIT を使用すると、次の AEST 修飾子が強制的に使用されます。

```
/NOEXECUTABLE
```

例を次に示します。

```
$ AEST/AUDIT DHRystone
```

この例は、サンプル・プログラム DHRystone.EXE の検査要約を要求しています。リスト・ファイル DHRystone_AV.LIS には、以下のテキストが含まれています。

```
$ AEST/AUDIT DHRystone
AEST (V1.0, Bld DEV_0.8/Feb 21 2005)
$
$ TYPE DHRystone_AV.LIS
AEST (V1.0, Bld DEV_0.8/Feb 21 2005)
Image "DHRystone", "V1.0", 24-FEB-2005 15:29:42.22
<SUM> Image name                               Tran Languages
<SUM> -----
<SUM> $1$DGA120:[VERIFICATION.BINTRAN.EXAMPLES]DHRystone.EXE      YES  C
$
```

```
/DEBUG
```

AEST に、すべての詳細メッセージを生成するよう指示します。

省略時の値	入力イメージが /DEBUG 修飾子を指定してリンクされていれば /DEBUG、/NODEBUG でリンクされていれば /NODEBUG です。
フォーマット	/[NO]DEBUG

コマンドの要約

修飾子の値	なし
/DUMP	
省略時の値	/DUMP=NONE
フォーマット	/DUMP=(list)
修飾子の値	次の 1 つまたは組み合わせ: ALPHA[=list] , IA64[=list] , INTERNAL[=list]

説明

/DUMP は、ソース Alpha イメージについての情報、結果として得られた I64 イメージについての情報、および可能であれば一部の内部情報をリスト・ファイルに出力するように AEST に指示します。

/DUMP を指定すると、次の指定と同等に扱われます。

```
/DUMP=(ALPHA=ALL, IA64=ALL, INTERNAL=NONE)
```

```
/DUMP=( [...] ALPHA=[...])
```

省略時の値	ALL
フォーマット	/DUMP=(ALPHA[=keywords])
修飾子の値	NONE ALL— 省略時の値。コード以外の、下記の情報をすべて出力します。 NONE— ソース・イメージの情報を出力しません。 HEADERS— イメージ・ヘッダを出力します。 SECTIONS— イメージ・セクションのリストを出力します。 SHR_IMAGES— 変換済みイメージが指す共用イメージのリストを出力します。 FIXUPS— フィックスアップのリストを出力します。 RELOCATIONS— 再配置情報のリストを出力します。 SYMB_VECTOR— シンボル・ベクタが存在する場合、その内容を出力します。 SYMB_TABLE— シンボル・テーブルが存在する場合、その内容を出力します。 INSTRUCTIONS— Alpha コードの命令を出力します。 BLOCKS— Alpha コードの基本ブロックと命令を出力します。 ENTRY_POINTS— 検出したエントリ・ポイントのリストを出力します。 VERBOSE— リスト (特にセクション・リスト) に詳細を追加します。

```
/DUMP=( [...] IA64 [=...])
```

省略時の値	ALL
フォーマット	/DUMP=(IA64[=keywords])

修飾子の値

ALL— 省略時の値。 BUNDLES 以外の、下記の情報をすべて出力します。

NONE— 変換済みイメージ情報を出力しません。

HEADERS— 変換済みイメージ・ヘッダを (内部表現で) 出力します。

SEGMENTS— 変換済みイメージ・セグメントの簡単なリストを出力します。

SHR_IMAGES— 変換済みイメージが指す共用イメージのリストを出力します。

FIXUPS— フィックスアップのリストを出力します。

RELOCATIONS— 再配置情報のリストを出力します。

SYMB_VECTOR— シンボル・ベクタが存在する場合、その内容を出力します。

SYMB_TABLE— シンボル・テーブルが存在する場合、その内容を出力します。

BUNDLES— 生成された I64 コードを出力します。

BLOCKS— Alpha の基本ブロックと、対応する I64 バンドルを出力します。初期化コード (および生成されたその他の I64 コード) も出力されます。

TII— TIE 情報構造体を出力します。

TRANS_VECTOR— 転送ベクタを出力します。

GOT— グローバル・オフセット・テーブルの内容を出力します。

/DUMP=([...] INTERNAL [=...])

この修飾子は、障害の解決だけを目的としています。通常は、この修飾子での出力は必要ありません。

省略時の値

ALL

フォーマット

/DUMP=(INTERNAL[=keywords])

キーワード

ALL— 下記のすべての情報を出力します。

NONE— 出力を行いません。

INTFIX— 内部のフィックスアップ (変換済みイメージには現れません) を出力します。

INTSYM— 内部シンボル (変換済みイメージには現れません) を出力します。

FDS— トランスレータにより生成された関数記述子のリストを出力します。

/EXECUTABLE

変換済みイメージの作成を有効にし、オプションで、ファイル指定も含めます。

省略時の値

/EXECUTABLE

フォーマット

/EXECUTABLE [=filespec]

修飾子の値

filespec

説明

特に指定しなければ、コマンド行で相反する修飾子 (たとえば、/AUDIT 修飾子) を指定したり、AEST がイメージ変換を行えないようなエラー状態を検出しない限り、AEST は変換済みイメージを作成します。AEST は、ERROR メッセージまたは FATAL メッセージを出力した場合は、変換済みイメージを作成しません。

ファイルを指定しないと、AEST は変換済みイメージを現在のディレクトリに書き込み、入力イメージのファイル名に_AV を付加した名前を付けます。たとえば、OpenVMS Alpha イメージの名前が PROGRAM.EXE だとすると、変換済みイメージの省略時の名前は、PROGRAM_AV.EXE になります。

注意

ファイル名の長さは、39文字を超えることはできません。この制限事項があるため、入力イメージ・ファイル名の長さが36文字を超えるときには、AESTは_AVという文字列を付加するために入力イメージ・ファイル名を短縮します。

/INTERPRET

イメージのコードを変換すべきか、実行時に解釈すべきかを制御します。

省略時の値	/NOINTERPRET
フォーマット	/INTERPRET
修飾子の値	なし

説明

AESTは、実行時に解釈するAlphaコードが少なくなるように、できるだけ多くのAlphaコードを検出し、解析して変換しようとします。特に指定しなければ、AESTはEISDSV_EXE ISDフラグを使用して、“executable”とマークされているイメージ・セクション内で検出したすべてのコードを変換します。

/INTERPRETを指定した場合でも、AESTは“executable”がマークされたAlphaイメージ・セクションで検出されたすべてのコードを解析し、対応する構造体をI64イメージに出力しますが、対応するI64コードは生成しません。これにより、AESTの実行時間と、変換に必要な仮想メモリが少なくなります。

/INTERPRETを使用した場合の変換済みイメージの実行時の性能は、コードを変換した場合のイメージよりも低くなります。

/LIST

リスト・ファイルを要求します。このとき、ファイル名を指定することもできます。

省略時の値	/LIST
フォーマット	/LIST [=filespec]
修飾子の値	filespec リスト・ファイルのファイル指定を示します。

説明

ファイル指定を指定しないと、AESTはリスト・ファイルを現在のディレクトリに書き込み、入力イメージのファイル名に_AVを付加して、拡張子をLISにした名前を付けます。たとえば、入力イメージがSIEVE.EXEの場合、省略時のリスト・ファイルはSIEVE_AV.LISとなります。

/VERBOSE

AEST に、すべての詳細メッセージを出力するように指示します。

省略時の値	/NOVERBOSE
フォーマット	/[NO]VERBOSE
修飾子の値	なし

/VERBOSE オプションの使用例を、次に示します。

```
$ AEST/VERBOSE DHRYSTONE
AEST V1.0 DEV_0.8 (Feb 21 2005) starting at Feb 25 2005 15:38:05 with command line:
AEST/VERBOSE DHRYSTONE Reading Source image...
Done.
Analyzing Alpha image code...
Done.
Translating data structures...
Done.
Translating code...
Done.
Adding init code...
Done.
Building TIE information...
Done.
Processing external symbols...
Done.
Fixing internal structures...
Building SDS...
Fixing code...
Freeing code...
Building TIE segment...
Fixing data...
Done.
Preparing to write target image...
Done.
Writing target image...
Done.
%AEST-I-TRANSOK, Translation completed successfully
$
$ TYPE DHRYSTONE_AV.LIS
AEST V1.0 DEV_0.8 (Feb 21 2005) starting at Feb 25 2005 15:38:05 with command line:
AEST/VERBOSE DHRYSTONE %AEST-I-TRANSOK, Translation completed successfully
$
```

エラー・メッセージとステータス・メッセージ

この付録では、次のトピックについて説明します。

AEST メッセージについて	第 B.1 節
AEST メッセージのアルファベット順リスト	第 B.2 節

B.1 AEST メッセージについて

AEST エラー・メッセージとステータス・メッセージは、イメージ内のさまざまな状況を示します。たとえば、AEST は、変換を正しく行うことができないコードを見分けることができます。各メッセージは、以下のいずれかの重大度レベルに分類されます。

- INFO メッセージは、AEST の実行状況を表示します。
- WARNING メッセージは、調査が必要な、疑わしいコードが検出されたことを示します。WARNING メッセージにより AEST が変換済みイメージを作成しなくなることはありません。
- ERROR メッセージは、AEST が入力イメージを変換できない、コード上の問題があることを示します。
- FATAL メッセージは、変換が全くできない問題 (たとえば、入力ファイルが見つからないなど) があることを示します。

第 B.2 節に、個々のメッセージの説明と、必要な場合は推奨する対処方法について示します。AEST メッセージは、表示されるとき、またはログ・ファイルに書き込まれるときには、機能名 (AEST)、重大度レベルを示す英字 (I, W, E, または F)、メッセージ識別子、およびエラーまたはステータスの簡単な説明からなります。例を次に示します。

```
%AEST-E-HASSECNAME, Image contains secondary image name -- not translatable
```

B.2 AEST メッセージの説明

AIIF_COLLISION

説明: .AIIF ファイルに、矛盾した情報が含まれています。一部の新しくリンクされたイメージのマッチ制御フィールドが複数回定義されていて、定義が矛盾しています。

BADEXE

説明: 実行可能イメージのフォーマットが不正です。この問題は、正しい OpenVMS Alpha イメージでないファイルがユーザが変換しようとした場合に発生します。

BADVEST

説明: 変換する OpenVMS Alpha イメージが、サポートされていないバージョンの VEST (VEST 1.0 より以前) により生成されています。

HASSECNAME

説明: 変換可能イメージが CLI イメージです。CLI イメージの変換は許されていません。

INVALID_SYMBOL

説明: シンボルの仮想アドレスが範囲外です (変換済みセグメントが見つかりません)。

ISDINIT

説明: イメージに OpenVMS 初期化セクションが含まれているため、変換できません。

ISDRESIDENT

説明: イメージにメモリ常駐セクションが含まれているため、変換できません。

LNKEXEC

説明: 変換可能イメージが、特定の OpenVMS Alpha カーネル・バージョンにリンクされているため、変換できません。

LNKSYS

説明: 変換可能イメージが OpenVMS Alpha とリンクされており、参照シンボルが含まれています。このイメージは変換できません。

OPENIN

説明: 変換対象のイメージ・ファイルをオープンできません。

PRIVOPC

説明: 特権命令 (opcode) が OpenVMS Alpha イメージ内に検出されました。変換済みイメージは生成されましたが、特権 Alpha 命令の実行が要求されると、このイメージの実行は失敗します。

RTLNOTSUPP

説明: 変換対象の OpenVMS Alpha イメージが、OMSAI で正式にサポートされていない実行時ライブラリにリンクされていました。変換済みイメージは生成されましたが、このイメージの正しい動作は保証されません。

TRANSERROR

説明: 変換はエラーで終了しましたが、実行可能なイメージは作成されました。変更済みイメージは正しく動作しない可能性が高いため、変換中に出力されたすべてのエラー・メッセージをチェックしてください。

TRANSFATAL

説明: 致命的なエラーが発生したため変換が完了せず、変換済み実行可能イメージは生成されませんでした。このイメージの変換はできません。

TRANSOK

説明: 変換が正常に完了し、変換済みイメージが生成されました。

TRANSWARN

説明: 変換が警告付きで完了し、変換済み実行可能イメージが生成されました。変換済みイメージを使用する前に、警告メッセージを見直してください。

変換および性能上の制限事項

この付録では、次のトピックについて説明します。

変換や性能上の制限事項があるイメージを見つける方法	第 C.1 節
変換できないイメージ	第 C.2 節
変換されるが WARNING エラー・メッセージが出力されるイメージ	第 C.3 節
変換中には検出できない互換性の問題があるイメージ	第 C.4 節

C.1 制限および性能上の問題の検出

Alpha Environment Software Translator (AEST) コーティリティを使用すると、OpenVMS Alpha イメージに存在する変換上の制限事項の大半を検出できます。コード分析フェーズにおいて、AEST は検出した、問題のあるコードを示し説明するメッセージを出力します。重大度が FATAL または ERROR のメッセージがある場合、AEST は変換済みイメージを作成しません。最も重大なメッセージレベルが WARNING の場合、AEST は、OpenVMS I64 システム上で正常に実行できる変換済みイメージを作成します。ただし、各 WARNING メッセージを注意深く調べることをお勧めします。

この付録では、制限事項や性能上の問題に関連する AEST メッセージを個別に取り上げます。検出されたコーディング上の問題についての説明は、付録 B にあるメッセージの説明を参照してください。

C.2 変換できないイメージ

表 C-1 で説明しているイメージでは、FATAL メッセージまたは ERROR メッセージが出力されます。

表 C-1 変換できないイメージ

説明	メッセージ
バージョン 6.1 より前の OpenVMS Alpha でリンクされているイメージ	BADEXE

(次ページに続く)

表 C-1 (続き) 変換できないイメージ

説明	メッセージ
ユーザ作成のシステム・サービスや、その他の非ユーザ・モードのコードがあるイメージ	HASSECNAME LNKEXEC ISDINIT ISPROTECT
特定のバージョンの OpenVMS とリンクされているイメージ	LNKSYS
オリジナルの OpenVMS VAX イメージが、バージョン 5.5 より前の OS 上でリンクされている場合に、DECmigrate ツールによって作成された VESTed バイナリ・イメージ	BADVEST

C.3 変換可能だが警告が出力されるイメージ

表 C-2 で説明しているイメージでは、WARNING メッセージが出力されます。各 WARNING メッセージを注意深く調べることをお勧めします。指摘されたコードが原因で変換済みイメージを実行できないということは、ほとんどありません。場合によっては、変換済みイメージを OpenVMS I64 システム上で正しく実行できるようにする処置を取らなければならないこともあります。ソース・コードが利用できる場合は、イメージを変換する前に、ソース・コードに修正を加えてから、イメージの再コンパイルおよび再リンクを行うか、別のオプションを指定してイメージを再リンクしなければならないことがあります。

AEST は、表 C-2 に示す理由以外でも WARNING メッセージを出力することがあります。

表 C-2 変換可能だが警告が出力されるイメージ

説明	メッセージ
特権命令を含むイメージ	PRIVOPC
現在のバージョンの OMSAI でサポートされていない標準の OpenVMS 実行時ライブラリを参照するイメージ	RTLNOTSUPP

C.4 検出できない変換上の問題があるイメージ

問題の中には、OpenVMS Alpha イメージの変換中には検出できず、OpenVMS I64 システム上で変換済みアプリケーションを実行したときに発生するものがあります。このような問題は、以下のようなイメージで発生します。

- DECthreads/Pthreads 以外のユーザ・モードのスレッド・マネージャを使用するイメージ。
- 独自のスレッド管理を実装しているイメージ。

- システム・サービスを使用してカーネル・スレッド・メカニズムに直接アクセスしているイメージ。

