

# OpenVMS Alpha オペレーティング・システム

---

## コネクティビティ開発者ガイド

– COM for OpenVMS , OpenVMS Registry , OpenVMS Events について –

AA-RGLUA-TE

1999 年 9 月

本書は、COM for OpenVMS , OpenVMS Registry , OpenVMS Events logging についての説明書です。また、OpenVMS と Windows NT の認証および相互運用についても説明しています。

改訂 / 更新情報:

本書は改訂マニュアルです。

ソフトウェア・バージョン:

OpenVMS Alpha Version 7.2-1  
Microsoft Windows NT 4.0 SP3

コンパックコンピュータ株式会社

---

1999年9月

本書の著作権はコンパックコンピュータ株式会社が保有しており、本書中の解説および図、表はコンパック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、コンパック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。

© Compaq Computer Corporation 1999.

All Rights Reserved.

Printed in Singapore.

本製品は Microsoft 社からライセンスされたソフトウェアを含みます。

Copyright © Microsoft Corporation, 1991-1998. All rights reserved.

本製品は Bristol Technology 社からライセンスされたソフトウェアを含みます。

Copyright © Bristol Technology, Inc, 1990-1998. All rights reserved.

以下は、米国 Compaq Computer Corporation の商標です。

Alpha, AlphaServer, AlphaStation, DEC, DIGITAL, OpenVMS, POLYCENTER, Tru64, VAX, VMS, COMPAQ, Compaq ロゴ, および DIGITAL ロゴ。

以下は、他社の商標です。

ActiveX, Microsoft, MS, MS-DOS, Win32, Windows, Windows NT, NT, Windows 95, and Windows 98 は Microsoft 社の商標です。

Adobe, Display POSTSCRIPT および POSTSCRIPT は Adobe Systems 社の商標です。

Apple, Mac, Macintosh, Power Macintosh は Apple Computer 社の商標です。

AT&T は American Telephone and Telegraph 社の商標です。

Motif, OSF/1, UNIX, "X" device, IT DialTone, X/Open, The Open Group は米国および他国の Open Group の商標です。

Unicode は Unicode 社の商標です。

Wind/U は Bristol Technology 社の商標です。

その他のすべての商標および登録商標は、それぞれの所有者が保有しています。

本書にある COM コードの例は、Dale Rogerson 著『*Inside COM*』(Microsoft Press, 1997)からの引用です。

原典：OpenVMS Connectivity Developer Guide

Copyright ©1999 Compaq Computer Corporation

本書は CD-ROM でも提供しています。

---

# 目次

まえがき	xiii
1 COM for OpenVMS に関するリリース・ノート	
1.1 リリース・ノート	1-1
1.1.1 COM Version 1.0 for OpenVMS から COM Version 1.1 for OpenVMS へのアップグレード	1-1
1.1.2 DCOM\$RPCSS プロセス・リソース全部の消費	1-1
1.1.3 IDispatch インタフェースを介して値を渡すときの浮動小数点の制限事項	1-1
1.1.4 COM for OpenVMS を実行するには、DECwindows Motif®が必要	1-2
1.1.5 MIDL -w スイッチ	1-2
1.1.6 MIDL コンパイラは wchar_t リテラルを char として扱う	1-2
1.1.7 プロセス内サーバのリモートからのアクティブ化	1-3
1.1.8 COM for OpenVMS のセキュリティ:COM Version 1.0 for OpenVMS と COM Version 1.1 for OpenVMS の相違点	1-4
1.1.9 DCOM\$CNFG ユーティリティとアプリケーションの無効化: 操作によって起こる副次作用	1-4
1.1.10 COM for OpenVMS でサポートされるスレッド・モデル	1-5
1.1.11 COM for OpenVMS Service Control Manager の起動とシャットダウンの変更	1-5
1.1.12 CoQueryProxyBlanket () と CoQueryClientBlanket () は有効な主名文字列を返さない	1-5
1.1.13 ローカル・サーバ・パスに対して論理名を使用する登録済みアプリケーション	1-6
1.1.14 SAFEARRAY の制約	1-6
1.1.15 COM Version 1.0 for OpenVMS と COM Version 1.1 for OpenVMS は同じクラスタ内でサポートされない	1-6
1.1.16 COM for Version 1.1 for OpenVMS 用に OpenVMS Registry の値の再登録が必要	1-7
1.1.17 拡張 NTLM を有効にした状態での SP4 はサポートされない	1-7
1.1.18 信頼されるドメインの認証をサポートするためのパッチ・キット	1-7
1.1.19 IGNORE_EXTAUTH をサポートするためのパッチ・キット	1-8
1.1.20 DCOM\$RPCSS は再起動時にストールする	1-8
1.1.21 2 つの OpenVMS システム間での COM for OpenVMS の使用	1-8
1.1.22 特定のエラー・メッセージ	1-9
1.1.22.1 RPC Cannot Support エラー (800706E4)	1-9
1.1.22.2 RPC Server Unavailable エラー (800706BA)	1-9
1.1.22.3 RPC WHO_ARE_YOU エラー (EE1282FA)	1-9
1.1.22.4 RPC PROTOCOL エラー (800706C0)	1-9

2	OpenVMS Registry に関するリリース・ノート	
2.1	リリース・ノート	2-1
2.1.1	キーの属性の変更時にキー変更が通知されない	2-1
2.1.2	データベース検索の制限	2-1
2.1.3	キー・アクセス・ポリシー	2-1
2.1.4	OpenVMS Registry の最大データ・サイズの制限	2-2
2.1.5	REGS_EXQUOTA エラー	2-2
2.1.6	OpenVMS Registry の最大データベース・サイズの制限	2-3

## 第1部 COM for OpenVMS

3	COM for OpenVMS の概要	
3.1	COM とは	3-1
3.1.1	参考文献	3-1
3.2	COM for OpenVMS の概要	3-2
3.2.1	COM for OpenVMS での OpenVMS Registry の使用	3-5
3.3	COM for OpenVMS の使用	3-5
3.3.1	新規アプリケーションの開発	3-6
3.3.2	既存のアプリケーションのカプセル化	3-6
3.4	OpenVMS MIDL コンパイラ	3-8
4	COM for OpenVMS キットのインストール	
4.1	COM Version 1.1 for OpenVMS キットの内容	4-1
4.2	インストールの要件	4-1
4.3	サポートされる COM for OpenVMS のインストール・オプション	4-2
4.4	OpenVMS スタンドアロン・システムでの COM for OpenVMS のインストール	4-2
4.5	OpenVMS スタンドアロン・システムでの COM Version 1.0 for OpenVMS からのアップグレード	4-4
4.6	OpenVMS Cluster での COM for OpenVMS のインストール	4-6
4.7	OpenVMS Cluster での COM for OpenVMS のアップグレード	4-9
4.8	OpenVMS の外部認証	4-11
4.8.1	外部認証での LOGINOUT.EXE の使用	4-12
4.8.2	DCE 統合ログインの制限	4-13
4.9	COM for OpenVMS 環境とは	4-13
4.9.1	COM for OpenVMS Service Control Manager (SCM)	4-15
4.9.2	OpenVMS Registry サーバ	4-15
4.9.3	Advanced Server for OpenVMS サーバ	4-15
4.9.4	ACME サーバ	4-16
4.9.5	RPC エンドポイント・マップ	4-17
4.9.6	RPC および SSPI/NTLM レイヤ	4-17
4.9.7	OpenVMS Events	4-17
4.10	COM for OpenVMS のインストール	4-18
4.11	COM for OpenVMS のインストールの後処理	4-19

4.12	COM for OpenVMS の起動 (COM for OpenVMS Service Control Manager) . . . . .	4-20
4.12.1	リブート後の COM for OpenVMS の自動起動 . . . . .	4-21
4.13	COM for OpenVMS (COM for OpenVMS Service Control Manager のシャットダウン) . . . . .	4-21
4.13.1	DCOM\$SHUTDOWN の確認要求を行わない設定 . . . . .	4-23
<b>5</b>	<b>アプリケーションの開発と運用のための COM for OpenVMS ユーティリティ</b>	
5.1	DCOM\$SETUP ユーティリティ . . . . .	5-1
5.2	DCOM\$SETUP の実行 . . . . .	5-2
5.2.1	DCOM\$RPCSS アカウントの作成と構成 . . . . .	5-4
5.2.2	COM サーバ (DCOM\$RPCSS Process) の起動と停止 . . . . .	5-7
5.2.3	アプリケーションの登録 . . . . .	5-7
5.3	DCOM\$CNFG の実行 . . . . .	5-10
5.3.1	DCOM\$CNFG の「Application List」サブメニュー . . . . .	5-11
5.3.2	「Registry Value Permissions」サブメニュー . . . . .	5-15
5.3.3	「Registry Key Permissions」サブメニュー . . . . .	5-18
5.3.4	「Application Identity」サブメニュー . . . . .	5-22
5.3.5	DCOM\$CNFG の「System-wide Default Properties」サブメニュー . . . . .	5-23
5.3.6	「System-wide Default Security」サブメニュー . . . . .	5-25
5.4	Windows NT ドメインを介した認証の構成 . . . . .	5-26
5.5	プロセス内サーバの登録: DCOM\$REGSVR32 ユーティリティ . . . . .	5-27
<b>6</b>	<b>COM for OpenVMS アプリケーションの開発</b>	
6.1	ステップ 1: 一意識別子を生成する . . . . .	6-1
6.2	ステップ 2: MIDL コンパイラを使用してアプリケーションを構築する . . . . .	6-2
6.2.1	MIDL コンパイラの実行 . . . . .	6-3
6.2.2	DCOM\$RUNSHRLIB による MIDL コンパイラの実行 . . . . .	6-3
6.2.3	必須 MIDL スイッチ . . . . .	6-4
6.2.4	必須インクルード・ディレクトリ . . . . .	6-5
6.2.5	必須ヘッダ・ファイル . . . . .	6-5
6.3	ステップ 3: COM アプリケーションをコンパイルする . . . . .	6-5
6.3.1	必要なマクロ定義 . . . . .	6-5
6.3.2	必要なインクルード・ディレクトリ . . . . .	6-6
6.3.3	必要なヘッダ・ファイル: VMS_DCOM.H . . . . .	6-6
6.3.4	必要な C++ 修飾子 . . . . .	6-6
6.3.5	必要な C 修飾子 . . . . .	6-7
6.4	ステップ 4: COM アプリケーションをリンクする . . . . .	6-7
6.4.1	クライアントとプロセス外コンポーネントのリンク . . . . .	6-7
6.4.2	プロセス内コンポーネント共用可能イメージのリンク . . . . .	6-8
6.4.2.1	シンボル・ベクタの作成 . . . . .	6-8
6.4.3	プロキシ/スタブ共用可能イメージのリンク . . . . .	6-9
6.4.3.1	シンボル・ベクタの作成 . . . . .	6-9
6.5	必要な OpenVMS Registry エントリ . . . . .	6-10
6.5.1	HKEY_CLASSES_ROOT\CLSID . . . . .	6-10
6.5.1.1	コンポーネント CLSID . . . . .	6-11
6.5.1.2	プロキシ/スタブ CLSID . . . . .	6-11
6.5.2	HKEY_CLASSES_ROOT\インタフェース . . . . .	6-12

6.6	OpenVMS および Windows のエラー・コードのテキストへの変換	6-12
	NTA\$VMSGetMessage	6-13

## 第 2 部 OpenVMS Registry

### 7 OpenVMS Registry の概要

7.1	レジストリとは	7-1
7.1.1	参考文献	7-1
7.2	OpenVMS Registry の概念と定義	7-2
7.2.1	キー, サブキー, 値	7-2
7.2.1.1	キーと値の揮発性	7-3
7.2.1.2	キーのライトスルーとライトバック	7-3
7.2.1.3	他のキーと値へのキーのリンク	7-3
7.2.1.4	OpenVMS Registry キーと値の名前の作成規則	7-4
7.2.2	クラス	7-4
7.2.3	ハイブ	7-4
7.3	OpenVMS Registry の構造	7-5
7.4	OpenVMS Registry への読み込みと書き込み	7-7
7.4.1	\$REGISTRY システム・サービス	7-8
7.4.2	REG\$CP サーバ管理ユーティリティ	7-8
7.5	OpenVMS Registry のセキュリティ	7-8
7.5.1	OpenVMS セキュリティ・モデル	7-9
7.5.1.1	AUTHORIZE ユーティリティによる OpenVMS Registry へのアクセス・ライトの付与	7-10
7.5.2	Windows NT のセキュリティ・モデル	7-11
7.6	OpenVMS Registry サーバ操作の制御	7-11
7.6.1	Maximum Reply Age/Age Checker Interval 設定の定義	7-12
7.6.2	Database Log Cleaner Interval/Initial Log File Size 設定の定義	7-12
7.6.3	Default File Quota/File Quota Interval 設定の定義	7-13
7.6.4	Scan Interval 設定の定義	7-13
7.6.5	Log Registry Value Error 設定の定義	7-13
7.6.6	Operator Communications Interval 設定の定義	7-14
7.6.7	Process Time Limit 設定の定義	7-14
7.6.8	Reply Log Cleaner Interval 設定の定義	7-14
7.6.9	Snapshot Interval/Snapshot Location/Snapshot Versions 設定の定義	7-15
7.6.10	Write Retry Interval 設定の定義	7-15

### 8 OpenVMS Registry システム管理

8.1	OpenVMS Registry のインストール	8-1
8.2	OpenVMS Registry の構成: REG\$CONFIG Configuration ユーティリティ	8-1
8.2.1	OpenVMS Registry の値の構成	8-3
8.3	OpenVMS Registry の起動	8-6
8.3.1	OpenVMS Registry の手動起動	8-6
8.4	OpenVMS Registry のシャットダウン	8-7

8.5	OpenVMS Registry サーバのコマンド .....	8-7
	SHOW SERVER REGISTRY_SERVER .....	8-8
	SET SERVER REGISTRY_SERVER .....	8-9
8.6	クラスタ内での OpenVMS Registry のフェールオーバ .....	8-10
	8.6.1 OpenVMS Registry サーバ・プロセスの優先順位の変更 .....	8-10
8.7	Windows NT システムからの OpenVMS Registry への接続 .....	8-11
8.8	OpenVMS Registry クォータ .....	8-12
8.9	OpenVMS Registry のセキュリティ .....	8-12
8.10	OpenVMS Registry データベースのバックアップと復元 .....	8-12
8.11	OpenVMS Alpha 複合バージョン・クラスタでの OpenVMS Registry の使用 .....	8-13
8.12	国際化と Unicode のサポート .....	8-13
<b>9</b>	<b>OpenVMS Registry サーバの管理</b>	
9.1	コマンド・ラインからの OpenVMS Registry サーバの管理 .....	9-1
9.2	OpenVMS Registry データベースのバックアップと復元 .....	9-3
	9.2.1 OpenVMS Registry データベースのスナップショットの作成 .....	9-4
	9.2.2 OpenVMS Registry データベースのスナップショットの復元 .....	9-4
9.3	OpenVMS Registry サーバ管理ユーティリティの構文 .....	9-5
	CREATE DATABASE .....	9-6
	CREATE KEY .....	9-8
	CREATE SNAPSHOT .....	9-10
	CREATE VALUE .....	9-11
	DELETE KEY .....	9-13
	DELETE VALUE .....	9-15
	EXPORT .....	9-16
	IMPORT .....	9-18
	LIST KEY .....	9-20
	LIST VALUE .....	9-23
	MODIFY KEY .....	9-25
	MODIFY VALUE .....	9-27
	MODIFY TREE .....	9-29
	SEARCH KEY .....	9-31
	SEARCH VALUE .....	9-32
	SHOW .....	9-33
	START MONITORING .....	9-34
	STOP .....	9-35
	ZERO COUNTERS .....	9-36

10	OpenVMS Registry システム・サービス	
	\$REGISTRY と \$REGISTRYW .....	OR-2

### 第3部 OpenVMS Events

11	OpenVMS Events	
11.1	イベントとは .....	11-1
11.1.1	参考文献 .....	11-1
11.2	OpenVMS Events の概要 .....	11-2
11.2.1	Windows NT イベント・ビューアによる OpenVMS Events の表示 .....	11-2
11.2.2	Advanced Server for OpenVMS イベント・ビューアによる OpenVMS Events の表示 .....	11-2
11.2.3	OpenVMS のみでのイベントのログ .....	11-3
	NTA\$EVENTW .....	11-5
11.3	独自のイベントの作成 .....	11-11
11.4	OpenVMS Events の問題への対処 .....	11-11

### 第4部 認証

12	認証	
12.1	認証とは .....	12-1
12.2	NTA\$LOGON の使用による Windows NT 資格情報の取得 .....	12-1
12.2.1	NTA\$LOGON の省略可能な修飾子 .....	12-3
12.2.2	Windows NT 資格情報を取得するための NTA\$LOGON の使用例 .....	12-5
12.3	ACM (Authentication and Credential Management) オーソリティ .....	12-6
12.3.1	OpenVMS での Windows NT 認証 .....	12-6
12.3.2	Windows NT ユーザの認証; OpenVMS プロセスへの Windows NT 資格情報の付与 .....	12-7
12.3.3	ACME_SERVER プロセスの管理 (ACME サーバ・コマンド) .....	12-8
12.3.4	MSV1_0 ACME エージェントの構成 .....	12-9

### 第5部 付録

A	MIDL コンパイラのオプション	
A.1	モード .....	A-1
A.2	入力 .....	A-1
A.3	出力ファイルの生成 .....	A-1
A.4	出力ファイル名 .....	A-2
A.5	C コンパイラとプリプロセッサのオプション .....	A-2
A.6	環境 .....	A-2
A.7	エラー・メッセージと警告メッセージ .....	A-3

A.8	最適化.....	A-3
A.9	その他.....	A-4
<b>B</b>	<b>トラブルシューティング</b>	
B.1	RPC のトラブルシューティング.....	B-1
B.2	ACME サーバのトラブルシューティング.....	B-3
B.3	DCOM\$RPCSS プロセスのトラブルシューティング.....	B-5
<b>C</b>	<b>クックブックの例: OpenVMS でのサンプル・アプリケーションの作成</b>	
C.1	COM の例 (サンプル 1).....	C-1
C.1.1	OpenVMS での手順.....	C-1
C.1.1.1	OpenVMS でのアプリケーションの構築.....	C-1
C.1.1.2	OpenVMS でのアプリケーションの登録.....	C-2
C.1.1.3	プロセス外サーバとしての OpenVMS でのアプリケーション の実行.....	C-3
C.1.1.4	OpenVMS でのアプリケーションの実行とリモート・サーバ の指定.....	C-3
C.1.1.5	プロセス内サーバとしての OpenVMS でのアプリケーション の実行.....	C-4
C.1.2	Windows NT の場合の手順.....	C-4
C.1.2.1	Windows NT でのアプリケーションの構築.....	C-4
C.1.2.2	Windows NT でのアプリケーションの登録.....	C-5
C.1.2.3	Windows NT でのアプリケーションの実行.....	C-5
C.2	オートメーションの例 (Dispatch_Sample1).....	C-5
C.2.1	OpenVMS の場合の手順.....	C-5
C.2.1.1	OpenVMS でのアプリケーションの構築.....	C-6
C.2.1.2	OpenVMS でのアプリケーションの登録.....	C-6
C.2.1.3	プロセス外サーバとしての OpenVMS でのアプリケーション の実行.....	C-6
C.2.1.4	OpenVMS でのアプリケーションの実行とリモート・サーバ の指定.....	C-7
C.2.1.5	プロセス内サーバとしての OpenVMS でのアプリケーション の実行.....	C-7
C.2.2	Windows NT の場合の手順.....	C-8
C.2.2.1	Windows NT でのアプリケーションの構築.....	C-8
C.2.2.2	Windows NT でのアプリケーションの登録.....	C-8
C.2.2.3	Windows NT でのアプリケーションの実行.....	C-9
<b>D</b>	<b>COM Version 1.0 for OpenVMS から COM Version 1.1 for OpenVMS への アップグレード</b>	
D.1	COM for OpenVMS の以前のバージョンからのアップグレード.....	D-1
D.1.1	既存の COM for OpenVMS アプリケーションの再構築.....	D-1
D.1.2	Windows NT システムの構成.....	D-1
D.1.3	OpenVMS システムの構成.....	D-2
D.2	Windows NT で以前に構成されたアプリケーション.....	D-3
D.2.1	COM Version 1.1 for OpenVMS では OpenVMS Registry に情報を再登 録しなければならない.....	D-4
D.2.2	OpenVMS Registry のアプリケーション・セキュリティ設定の変更.....	D-4
D.2.2.1	COM アプリケーションのレジストリ・キー.....	D-6

## E 相違点, API, インタフェースの一覧

E.1	COM for OpenVMS と Microsoft COM の相違点	E-1
E.1.1	サービス制御マネージャ (SCM)	E-1
E.1.2	サーバ・アプリケーションのスタック・サイズ	E-1
E.1.3	“char”データ型の使用	E-1
E.1.4	MIDL コンパイラのバージョン	E-2
E.1.5	アプリケーション構成のアクセス権を変更するための DCOM\$CNFG の使用	E-2
E.2	API	E-4
E.3	インタフェース	E-7

## F COM for OpenVMS によりインストールされたファイル一覧

F.1	COM for OpenVMS によりインストールされたファイル	F-1
-----	----------------------------------	-----

## G 用語集

## H 略語集

## 索引

## 例

4-1	COM for OpenVMS のインストールの例	4-19
5-1	OpenVMS での“Simple”アプリケーションの登録例	5-8
5-2	SSERVER.REG_NT の内容	5-9
5-3	SSERVER.REG_VMS の内容	5-9
5-4	例: HostMapDomains の設定	5-26
5-5	DCOM\$REGSVR32 ユーティリティを使用したコンポーネントの登録	5-29
5-6	DCOM\$REGSVR32 ユーティリティを使用したコンポーネントの登録の解除	5-29
7-1	ユーザにライトを付与するための AUTHORIZE の使用	7-10
8-1	優先順位の設定	8-10
8-2	優先順位の変更	8-11
11-1	OpenVMS イベント・ログの例	11-4
12-1	NTASLOGON セッションの例	12-3
12-2	Windows NT 資格情報の初めての取得	12-5
12-3	Windows NT 資格情報の置換	12-5
12-4	ファイルへのパスワードの保存	12-5



3-1	OpenVMS のインフラストラクチャと COM for OpenVMS .....	3-3
4-1	プロセスとレイヤの関係 .....	4-14
5-1	「DCOM\$SETUP OpenVMS COM Tools」メニュー .....	5-2
5-2	「DCOM\$CNFG Main」メニュー .....	5-10
5-3	「Applications List」サブメニュー .....	5-11
5-4	「Application Properties」サブメニュー .....	5-12
5-5	「Application Location」サブメニュー .....	5-13
5-6	「Application Security」サブメニュー .....	5-14
5-7	「Registry Value Permissions」サブメニュー .....	5-15
5-8	「Edit Registry Value Permissions」サブメニュー .....	5-16
5-9	「Add Registry Value Permissions」サブメニュー .....	5-17
5-10	「Registry Key Permissions」サブメニュー .....	5-18
5-11	「Edit Registry Key Permissions」サブメニュー .....	5-19
5-12	「Special Access Registry Key Permissions」サブメニュー .....	5-20
5-13	「Add Registry Key Permissions」サブメニュー .....	5-21
5-14	「Application Identity」サブメニュー .....	5-22
5-15	「System-wide Default Properties」サブメニュー .....	5-23
5-16	「Default Authentication Level」サブメニュー .....	5-24
5-17	「Default Impersonation Level」サブメニュー .....	5-24
5-18	「System-wide Default Security」サブメニュー .....	5-25
7-1	キー, サブキー, 値の関係 .....	7-3
10-1	item-list-3 の構造 .....	OR-3
10-2	item-list-64b の構造 .....	OR-4

## 表

1-1	セキュリティの相違点の要約 .....	1-4
4-1	プロセス名とサーバ名の関係 .....	4-14
5-1	DCOM\$REGSVR32 コマンド・ライン・オプション .....	5-28
9-1	OpenVMS Registry サーバ管理ユーティリティのコマンド .....	9-2
10-1	アイテム記述子フィールド .....	OR-4
10-2	記述子フィールド .....	OR-5
10-3	有効な機能コード .....	OR-10
10-4	アイテム・コードの要約 .....	OR-21
11-1	OpenVMS Events エラーへの対処法 .....	11-12
12-1	NTASLOGON ユーティリティのコマンド・ライン・パラメータ .....	12-2
12-2	MSV1_0 ACME エージェントの論理名 .....	12-9
B-1	RPC エラー .....	B-1



## 対象読者

本書は主に、OpenVMS インフラストラクチャを使用して、OpenVMS 環境と Windows NT 環境の間で簡単に移動できるアプリケーションの開発者を対象にしています。特に、次の開発者を対象にしています。

- COM for OpenVMS 開発者: 既存の OpenVMS アプリケーションやデータのカプセル化を行ったり、OpenVMS システム用に COM アプリケーションを新たに作成する開発者。
- OpenVMS Registry 開発者: OpenVMS システムのみの情報を格納するために OpenVMS Registry を使用する開発者、または OpenVMS と Windows NT の両方のレジストリ情報の共有リポジトリとして OpenVMS Registry を使用する開発者。

本書は COM やレジストリの入門書ではありません。本書では、読者はオブジェクト指向 (OO) の概念や COM 開発手法を十分理解しており、レジストリが Windows NT システムでどのように動作するかも理解しているものと想定しています。本書では、COM とレジストリに関するオンライン情報の参照先を示し、COM、OO 開発、レジストリに関する参考文献も示しています。

## 本書の構成

本書には、COM for OpenVMS アプリケーションの開発や OpenVMS Registry の使用に必要なすべての情報が網羅されています。本書の構成は次のとおりです。

- リリース・ノート  
COM for OpenVMS、OpenVMS Registry、OpenVMS Events に関するリリース・ノート。
- 第 1 部  
COM for OpenVMS に関する情報。COM for OpenVMS のインストール、構成、実行、COM for OpenVMS アプリケーションの開発方法などを示します。
- 第 2 部  
OpenVMS Registry に関する情報。OpenVMS Registry の概要と概念、OpenVMS Registry サーバの起動とシステム管理、OpenVMS Registry システム・サービス、OpenVMS Registry サーバ管理について説明します。

- 第 3 部  
OpenVMS Events に関する情報。
- 第 4 部  
OpenVMS システムと Windows NT システムの間のユーザとアプリケーションの認証に関する情報。
- 第 5 部  
参照情報。MIDL コンパイラに関する情報，COM for OpenVMS クックブックの例，COM for OpenVMS でサポートされる COM API，COM for OpenVMS の以前のバージョンからのアップグレードの方法，インストールされるファイルの一覧，用語集，略語の一覧を示します。
- 索引

## サンプル・コードに示されている Win32 API 呼び出し

本書で参照しているサンプル・コードは COM for OpenVMS キットに添付されています。これらのサンプル・コードに示されている Win32® API 呼び出しは，説明のためのサンプルとしてのみ提供されています。

COM for OpenVMS には，COM for OpenVMS ソフトウェアで必要とされる Win32 API だけが添付されています。これらの COM API は付録 E, 相違点，API，インタフェースの一覧に示されています。

付録 E には示されていないものの，本書のサンプルで使用されていて，COM for OpenVMS キットのサンプル・コードに含まれている Win32 API 呼び出しは，Compaq 以外のソフトウェア・ベンダから提供されている呼び出しです。付録 E に示されていない Win32 API を OpenVMS で使用する場合は，たとえば Bristol Technologies ([www.bristol.com](http://www.bristol.com)) などの独立系ソフトウェア・ベンダからインタフェースを購入する必要があります。

## 関連資料

Open Systems Software Group (OSSG) の製品とサービスの詳細については，Compaq OpenVMS の World Wide Web サイトを参照してください。アドレスは次のとおりです。

[www.compaq.com/openvms](http://www.compaq.com/openvms)

## 本書で使用する表記法

### 製品名について

本書では、「OpenVMS Alpha」は「OpenVMS Alpha オペレーティング・システム」を、「OpenVMS VAX」は「OpenVMS VAX オペレーティング・システム」を指します。また特に明記しない限り、「OpenVMS」は、「OpenVMS Alpha オペレーティング・システム」および「OpenVMS VAX オペレーティング・システム」の両方を指します。

VMScuser システムは OpenVMS Cluster システムのことを指します。特に指定のないかぎり、OpenVMS Cluster または クラスタは VMScuser と同意語です。

本書では、DECwindows および DECwindows Motif はすべて DECwindows Motif for OpenVMS ソフトウェアを意味します。

また、本書では次の表記法も使用しています。

表記法	意味
Ctrl/x	Ctrl/x という表記は、Ctrl キーを押しながら別のキーまたはポインティング・デバイス・ボタンを押すことを示します。
PF1 x	PF1 x という表記は、PF1 に定義されたキーを押してから、別のキーまたはポインティング・デバイス・ボタンを押すことを示します。
<span style="border: 1px solid black; padding: 2px;">Return</span>	例の中で、キー名が四角で囲まれている場合には、キーボード上でそのキーを押すことを示します。テキストの中では、キー名は四角で囲まれていません。 HTML 形式のドキュメントでは、キー名は四角ではなく、括弧で囲まれています。
...	例の中の水平方向の反復記号は、次のいずれかを示します。 <ul style="list-style-type: none"><li>• 文中のオプションの引数が省略されている。</li><li>• 前出の 1 つまたは複数の項目を繰り返すことができる。</li><li>• パラメータや値などの情報をさらに入力できる。</li></ul>
.	垂直方向の反復記号は、コードの例やコマンド形式の中の項目が省略されていることを示します。このように項目が省略されるのは、その項目が説明している内容にとって重要ではないからです。
( )	コマンドの形式の説明において、括弧は、複数のオプションを選択した場合に、選択したオプションを括弧で囲まなければならないことを示しています。
[ ]	コマンドの形式の説明において、大括弧で囲まれた要素は任意のオプションです。オプションをすべて選択しても、いずれか 1 つを選択しても、あるいは 1 つも選択しなくても構いません。ただし、OpenVMS ファイル指定のディレクトリ名の構文や、割り当て文の部分文字列指定の構文の中では、大括弧に囲まれた要素は省略できません。
[   ]	コマンド形式の説明では、括弧内の要素を分けている垂直棒線はオプションを 1 つまたは複数選択するか、または何も選択しないことを意味します。

表記法	意味
{ }	コマンドの形式の説明において、中括弧で囲まれた要素は必須オプションです。いずれか 1 つのオプションを指定しなければなりません。
太字	太字のテキストは、新しい用語、引数、属性、条件を示しています。
<i>italic text</i>	イタリック体のテキストは、重要な情報を示します。また、システム・メッセージ (たとえば内部エラー <i>number</i> )、コマンド・ライン (たとえば <i>PRODUCER=name</i> )、コマンド・パラメータ (たとえば <i>device-name</i> ) などの変数を示す場合にも使用されます。
UPPERCASE TEXT	英大文字のテキストは、コマンド、ルーチン名、ファイル名、ファイル保護コード名、システム特権の短縮形を示します。
Monospace type	モノスペース・タイプの文字は、コード例および会話型の画面表示を示します。 C プログラミング言語では、テキスト中のモノスペース・タイプの文字は、キーワード、別々にコンパイルされた外部関数およびファイルの名前、構文の要約、または例に示される変数または識別子への参照などを示します。
-	コマンド形式の記述の最後、コマンド・ライン、コード・ラインにおいて、ハイフンは、要求に対する引数とその後の行に続くことを示します。
数字	特に明記しない限り、本文中の数字はすべて 10 進数です。10 進数以外 (2 進数、8 進数、16 進数) は、その旨を明記してあります。

---

# COM for OpenVMS に関するリリース・ノート

---

## 1.1 リリース・ノート

ここで説明する情報は本リリースに適用されます。

### 1.1.1 COM Version 1.0 for OpenVMS から COM Version 1.1 for OpenVMS へのアップグレード

COM Version 1.0 for OpenVMS から COM Version 1.1 for OpenVMS にアップグレードする場合は、付録 D と第 4.3 節に示すアップグレードの手順に従ってください。

### 1.1.2 DCOM\$RPCSS プロセス・リソース全部の消費

COM for OpenVMS 実行時環境では、DCOM\$RPCSS プロセスが常に実行されている必要があります。

DCOM\$RPCSS が多くの COM for OpenVMS アプリケーション・サーバを作成または削除した後、DCOM\$RPCSS がリソースをすべて消費してしまうことがあることがテスト中にわかりました。この状況になると、DCOM\$RPCSS は自動的に再起動しようとしません。DCOM\$RPCSS がシステムで実行されていない場合や、ハングしているように見える場合は、次のコマンドを使用して DCOM\$RPCSS をいったん停止した後、再起動してください。

```
$ @SYS$STARTUP:DCOM$SHUTDOWN
$ @SYS$STARTUP:DCOM$STARTUP
```

SYS\$MANAGER:DCOM\$RPCSS.OUT ファイル記録されているエラー・メッセージと情報メッセージを確認できます。また、正常に動作していないアプリケーション・サーバ・プロセスがないかどうか確認し、必要に応じてプロセスを停止する必要があります。

### 1.1.3 IDispatch インタフェースを介して値を渡すときの浮動小数点の制限事項

複数の浮動小数点パラメータ (単精度または倍精度) を IDispatch インタフェースに値で渡すときに、問題が発生することがあります。

複数の浮動小数点パラメータを使用するコンポーネント・メソッドは、IDispatch インタフェースを介してアクセスした場合、予測できない動作で終了します。浮動小数

点パラメータを参照で渡した場合は、このエラーは発生しません。この制限は将来のリリースで解決される予定です。

#### 1.1.4 COM for OpenVMS を実行するには、DECwindows Motif ®が必要

COM for OpenVMS を実行するシステムには、DECwindows Motif for OpenVMS をインストールしなければなりません。DECwindows Motif がシステムにすでにインストールされている場合は、改めてインストールする必要はありません。DECwindows Motif がシステムにインストールされていない場合は、OpenVMS Version 7.2-1 の CD-ROM の [KITS.DWMOTIF125\_KIT] ディレクトリに格納されている DECwindows Motif のインストール・キットを利用してください。

---

##### 注意

---

COM for OpenVMS を使用するためにだけ、DECwindows Motif をインストールする場合は、DW-MOTIF ライセンスは必要ありません。

---

#### 1.1.5 MIDL -w スイッチ

MIDL コンパイラでは、-w または -warn を指定することで、コンパイラが生成する警告のレベルを下げるすることができます。OpenVMS 用の MIDL コンパイラでは、-w スイッチだけがサポートされます。

#### 1.1.6 MIDL コンパイラは wchar\_t リテラルを char として取り扱う

COM for OpenVMS のこのリリースで、DEC C バージョン 5.7 またはそれ以前のバージョンを使用する場合、IDL ファイル内のワイド文字のリテラル文字列は、誤って "char" データ型として取り扱われます (DEC C バージョン 6.0 以上を使用している場合は、このリリース・ノートは適用されません)。

たとえば、IDL ファイルに次の文字列リテラルが格納されているとしましょう。

```
const wchar_t * PROGRAM_ID    = L"Sample.Component";
```

Microsoft Windows NT で MIDL コンパイラを使用すると、次のマクロ定義が作成されます。

```
#define PROGRAM_ID    ( L"Sample.Component" )
```

しかし、COM for OpenVMS 用の MIDL コンパイラでは、デフォルトで次のマクロ定義が作成されます。

```
#define PROGRAM_ID    ( "Sample.Component" )
```

この問題を回避するには、次の操作を行います。

1. できる限り、DEC C プリプロセッサを使用しないようにします。

プリプロセッサを使用せずに MIDL コンパイラを実行するには、コマンド・ラインに `-nocpp` または `-no_cpp` スイッチを指定します。次の例を参照してください。

```
$ midl -Oicf -nocpp -idcom$library: server.idl
```

---

### 重要

---

IDL ソース・ファイルや、メインの IDL ソース・ファイルからインポートされた IDL ソース・ファイルに条件付きアセンブリ・スイッチ (たとえば、`#ifdef` . . . `#endif`) が含まれている場合は、この方法は使用しないでください。

---

2. すべての文字列定数を `wchar_t` タイプではなく、`char` タイプとして定義します。

この方法を使用すると、Microsoft Windows NT の MIDL コンパイラも OpenVMS の MIDL コンパイラも、ワイド文字ではない文字列定数を作成します。ソフトウェアでワイド文字の文字列リテラルが必要な場合は、値を使用する前に、ソフトウェアで ANSI 文字列をワイド文字の文字列に変換できます。

3. IDL ソース・ファイル内で、ワイド文字の文字列定数をマクロ定義に置き換えます。

たとえば、文字列リテラルを次のように定義する代わりに、

```
const wchar_t * PROGRAM_ID = L"Sample.Component";
```

次に示すように、IDL `cpp_quote()` の内部で `#define` を使用します。

```
cpp_quote("#define PROGRAM_ID L\"Sample.Component\"")
```

DEC C プリプロセッサを使用する場合でも、Microsoft Windows NT と OpenVMS 用の MIDL コンパイラで作成される出力ヘッダ・ファイルは次のようになります。

```
#define PROGRAM_ID L"Sample.Component"
```

## 1.1.7 プロセス内サーバのリモートからのアクティブ化

サーバ・コンポーネントがプロセス内サーバとしてだけ登録されている場合、そのコンポーネントを OpenVMS でリモートからアクティブ化することはできません。システムがプロセス内サーバをリモートからアクティブ化しようとする、リモート・クライアントは `REGDB_E_CLASSNOTREG (80040154)` エラーを受信します。サーバ・コンポーネントをリモートからアクティブ化するには、コンポーネントをプロセス外サーバとして登録し、`DCOM$RPCSS` プロセスがクライアントの代わりにコンポーネントを起動できるようにしなければなりません。

### 1.1.8 COM for OpenVMS のセキュリティ:COM Version 1.0 for OpenVMS と COM Version 1.1 for OpenVMS の相違点

次の表は、COM Version 1.0 for OpenVMS と COM Version 1.1 for OpenVMS の相違点を示しています。

表 1-1 セキュリティの相違点の要約

領域	COM Version 1.0 for OpenVMS	COM Version 1.1 for OpenVMS
クライアント要求	Windows NT では認証される。OpenVMS への要求では認証されない。	Windows NT と OpenVMS で認証される。
セキュリティ	Windows NT では、サーバはクライアントのアイデンティティで動作でき、OpenVMS ではあらかじめ指定された OpenVMS のアイデンティティで動作できる。	Windows NT と OpenVMS の両方で、サーバはクライアントのアイデンティティで動作できる。
セキュリティ	Windows NT ではメソッド単位のセキュリティが認められるが、OpenVMS ではプロセス単位のセキュリティだけが認められる。	Windows NT と OpenVMS の両方で、メソッド単位のセキュリティが認められる。
アウトバウンド COM 要求	Windows NT でのみ認証される。	Windows NT と OpenVMS で認証される。
レジストリ	<i>Windows NT</i> : NT の資格情報で制御される。 <i>OpenVMS</i> : 特権やライト識別子などの OpenVMS セキュリティ制御機能に依存する。	<i>Windows NT</i> : NT の資格情報で制御される。 <i>OpenVMS</i> : Windows NT の資格情報または OpenVMS セキュリティ制御によって制御される。
イベント・ログ	Windows NT のみ。	Windows NT と OpenVMS の両方。

### 1.1.9 DCOM\$CNFG ユーティリティとアプリケーションの無効化: 操作によって起こる副次作用

COM for OpenVMS DCOM\$CNFG ユーティリティには、開発者がアプリケーションのプロパティを変更するために使用できるオプションがいくつかあります (たとえば、アプリケーションを実行できるコンピュータのロケーションの変更など)。これらのオプションを選択すると、OpenVMS Registry エントリが変更されます。

OpenVMS Registry では、クラスタ内で 1 つのデータベースがサポートされるので、これらのオプションのいずれかを変更すると、COM for OpenVMS を稼動しているクラスタ内のすべてのノードに影響があります。

たとえば、「System-wide Default Properties」サブメニューのオプション 1 を使用して、COM for OpenVMS を無効化すると、クラスタ全体で COM for OpenVMS が無効化されます。同様に、「Application Location」サブメニューのオプション 1 を使用して、このコンピュータでアプリケーションを実行することを禁止すると、クラスタ内のどのコンピュータでもそのアプリケーションを実行できなくなります。

### 1.1.10 COM for OpenVMS でサポートされるスレッド・モデル

COM Version 1.1 for OpenVMS では、アプリケーション・サーバに対してマルチスレッド・アパートメント (MTA, フリー・スレッドとも呼ばれる) モデルのみがサポートされます。マルチスレッド・アパートメント・モデルでは、コンポーネントは複数のスレッドを持つことができます。しかし、コードがスレッド・セーフであることを確認する必要があります。

スレッド・モデルの初期化呼び出しは次のとおりです。

```
CoInitializeEx(  
    NULL,  
    COINIT_MULTITHREADED  
)
```

CoInitialize( ) はシングル・スレッド・アパートメント (STA) モデルを暗黙に示すので、サーバ・アプリケーションで CoInitializeEx( ) の代わりに使用することはできません。

### 1.1.11 COM for OpenVMS Service Control Manager の起動とシャットダウンの変更

COM for OpenVMS の以前のバージョンでは、独立プロセス DCOM\$RPCSS が実際に動作しているのか (つまり、初期化を完了したのか)、まだ初期化中であるのかを確認できないことがありました。

今回のリリースでは、DCOM\$RPCSS.EXE イメージが変更されており、状態に応じて異なるプロセス名が表示されるようになりました。DCOM\$RPCSS.EXE イメージが初期化を実行している間は、プロセス名は“DCOM\$STARTUP-\*\*”になります。DCOM\$RPCSS.EXE イメージが実行中の場合は、プロセス名は“DCOM\$RPCSS”に変化します。プロセスが異常終了して再起動しなければならない場合は、プロセス名は“DCOM\$STARTUP-\*\*”に戻ります。

DCOM\$STARTUP.COM と DCOM\$SHUTDOWN.COM はどちらも、いずれかのプロセス名を検索するように変更されています。

COM for OpenVMS の起動の詳細については、第 4.12 節を参照してください。  
COM for OpenVMS のシャットダウンの詳細については、第 4.13 節を参照してください。

### 1.1.12 CoQueryProxyBlanket ( ) と CoQueryClientBlanket ( ) は有効な主名文字列を返さない

CoQueryProxyBlanket( ) と CoQueryClientBlanket( ) API は有効な主名文字列を返しません。これらの API を使用して主名を取得することはできません。

### 1.1.13 ローカル・サーバ・パスに対して論理名を使用する登録済みアプリケーション

ローカル・サーバ・パスに対して論理名を使用する COM アプリケーションをすでに登録している場合は、ローカル・サーバ・パスの実際の名前を使用するようにアプリケーションを変更 (再登録) しなければなりません。

たとえば、REGISTER\_SIMPLE.COM コマンド・プロシージャを使用して、COM Version 1.0 for OpenVMS で“Simple”アプリケーションを登録した場合は、新しい REGISTER\_SIMPLE.COM コマンド・プロシージャを使用して“Simple”アプリケーションを再登録する必要があります。

COM Version 1.1 for OpenVMS の登録コマンド・ファイルは更新されています。

システムは次に示すように、COM アプリケーション・ローカル・サーバ・パスを値データとして OpenVMS Registry に格納します。

```
"HKEY_CLASSES_ROOT\CLSID\{GUID}\LOCALSERVER32"
```

ローカル・サーバ・パスを変更するには、次の REG\$CP コマンドを使用します。

```
$ MCR REG$CP CREATE VALUE  
HKEY_CLASSES_ROOT\CLSID\{GUID}\Localserver32 -  
/TYPE=SZ/DATA=device:[directory]IMAGENAME.EXE
```

GUID は COM アプリケーションの CLSID です。Localserver32 と CLSID の詳細については、第 6.5 節を参照してください。

### 1.1.14 SAFEARRAY の制約

COM for OpenVMS MIDL コンパイラは Microsoft の MIDL コンパイラ V3.00.44 を基礎にしているため、COM for OpenVMS では、SAFEARRAY の使用は IDL ファイル内の LIBRARY ブロックの内部でのみサポートされます。Microsoft の MIDL コンパイラ V3.00.44 にも同じ制約があります。

### 1.1.15 COM Version 1.0 for OpenVMS と COM Version 1.1 for OpenVMS は同じクラスタ内でサポートされない

クラスタ内のノードに COM Version 1.1 for OpenVMS をインストールして構成する場合、同じクラスタ内の他のノードで COM Version 1.0 for OpenVMS が実行されないように、OpenVMS Registry をクラスタ全体で変更しなければなりません。

### 1.1.16 COM for Version 1.1 for OpenVMS 用に OpenVMS Registry の値の再登録が必要

COM Version 1.1 for OpenVMS では、セキュリティの設定を含むように、OpenVMS Registry の値を再登録しなければなりません。DCOM\$SETUP コマンドを使用して「OpenVMS COM Tools」メニューを表示し、オプション 3 を選択します。

COM Version 1.1 for OpenVMS で OpenVMS Registry の値を再登録すると、再登録を確認するメッセージが表示されます。そのたびに YES と応答します。次の例を参照してください。

```
[ Starting to Populate the COM for OpenVMS Registry ]  
Populating the Registry for OpenVMS may take up to 15 minutes  
depending on your system.  
Enter Y[ES] to continue: YES  
  
The COM for OpenVMS Registry has already been loaded. This  
action will overwrite the current COM for OpenVMS values  
and data.  
Enter Y[ES] to continue: YES
```

### 1.1.17 拡張 NTLM を有効にした状態での SP4 はサポートされない

COM Version 1.1 for OpenVMS では、次の制限のもとに NT SP4 がサポートされません。COM Version 1.1 for OpenVMS では、拡張 NTLM を有効にした状態で SP4 はサポートされません。

COM Version 1.1 for OpenVMS を SP4 と組み合わせて使用する場合は、拡張 NTLM が無効に設定されていることを確認してください。

SP4 の拡張 NTLM を無効にした状態では、SP4 と COM for OpenVMS は協調動作するように見えますが、SP4 は COM for OpenVMS と組み合わせて使用した状態で完全にテストされておらず、公式にはサポートされません。

弊社で行った SP4 に関するテストでは、SP4 と拡張 NTLM を無効にした状態で、12 文字より長いパスワードを使用した場合、認証要求が失敗するという制限が確認されています。

### 1.1.18 信頼されるドメインの認証をサポートするためのパッチ・キット

COM for OpenVMS のこのリリースでは、信頼されるドメインの認証はサポートされません。弊社はこの機能を提供するためにパッチ・キットを準備しています。

キットが提供される時期については、COM for OpenVMS の Web サイトを参照してください。COM for OpenVMS の Web サイトのアドレスは次のとおりです。

<http://www.openvms.digital.com/openvms/products/dcom/>

#### 1.1.19 IGNORE\_EXTAUTH をサポートするためのパッチ・キット

COM for OpenVMS のこのリリースでは、SECURITY\_POLICY システム・パラメータの IGNORE\_EXTUATH フラグはサポートされません。弊社はこの機能を提供するためにパッチを準備しています。

パッチの詳細については、COM for OpenVMS の Web サイトを参照してください。COM for OpenVMS の Web サイトのアドレスは次のとおりです。

<http://www.openvms.digital.com/openvms/products/dcom/>

#### 1.1.20 DCOM\$RPCSS は再起動時にストールする

クラスタ内で DCOM\$RPCSS を稼動しているシステムがクラッシュし、再起動すると、DCOM\$RPCSS プロセスは起動時にハングすることがあります。この状況では、プロセス名は DCOM\$STARTUP-\*\* のままであり、SYS\$STARTUP:DCOM\$RPCSS.OUT ファイルに次のエラー・メッセージが記録されます。

```
%PPL-W-SYSERROR, system service error  
-SYSTEM-W-VALNOTVALID, value block is not valid
```

この状態から回復するには、ストールされたプロセスを停止し、次のコマンドを入力して COM for OpenVMS を再起動します。

```
$ @SYS$STARTUP:DCOM$STARTUP
```

#### 1.1.21 2 つの OpenVMS システム間での COM for OpenVMS の使用

2 つの OpenVMS システム間で COM アプリケーションを実行するには、2 つの OpenVMS システムがどちらも同じドメインに存在しなければなりません。OpenVMS システムが同じドメイン内にない場合は、メソッド呼び出し時に認証が失敗します。OpenVMS システムと Windows NT システムの間でアプリケーションを実行している場合は、この制限は適用されません。

この制限をなくすために、弊社はパッチ・キットを準備しています。

キットの詳細については、COM for OpenVMS の Web サイトを参照してください。COM for OpenVMS の Web サイトのアドレスは次のとおりです。

<http://www.openvms.digital.com/openvms/products/dcom/>

## 1.1.22 特定のエラー・メッセージ

ここでは、特定の COM for OpenVMS エラー・メッセージについて説明します。

### 1.1.22.1 RPC Cannot Support エラー (800706E4)

シングル・スレッド・アパートメント (STA) モデルを使用しようとする、一部の COM API で次の戻り状態コードが表示されることがあります。

```
(800706E4)
```

このモデルは COM Version 1.1 for OpenVMS ではサポートされません。詳細については、第 1.1.10 項を参照してください。

### 1.1.22.2 RPC Server Unavailable エラー (800706BA)

特定の状況で、Windows NT でクライアント・アプリケーションを実行し、OpenVMS でサーバ・アプリケーションを実行している場合、次のエラーが発生することがあります。

```
RPC server unavailable (800706BA)
```

このエラーに対処するには、サーバ・アプリケーションをいったん停止して再起動した後 (アプリケーションを手動で起動した場合)、クライアント・アプリケーションを再起動します。

### 1.1.22.3 RPC WHO\_ARE\_YOU エラー (EE1282FA)

特定の状況で、クライアントとサーバの間で通信するときに、次のエラーが発生することがあります。

```
rpc_s_who_are_you_failed (EE1282FA)
```

このメッセージが表示された場合は、操作を再実行してください。

弊社はこの問題を解決するために、パッチを準備しています。

### 1.1.22.4 RPC PROTOCOL エラー (800706C0)

特定の状況で、クライアントとサーバの間で通信するときに、次のエラーが発生することがあります。

```
rpc_s_protocol_error (800706C0)
```

このメッセージが表示された場合は、操作を再実行してください。

弊社はこの問題を解決するために、パッチを準備しています。



---

## OpenVMS Registry に関するリリース・ノート

---

### 2.1 リリース・ノート

このリリース・ノートは本リリースに適用されます。

#### 2.1.1 キーの属性の変更時にキー変更が通知されない

REG\$\_NOTIFYFILTER アイテム・コードに対して REG\$\_CHANGEATTRIBUTES という値を指定した場合、その OpenVMS Registry キーが変更されると、変更が通知されなければなりません。しかし、OpenVMS Registry キーの属性を変更した場合、通知を要求したプロセスに変更が通知されません。

この問題に対処するには、REG\$\_NOTIFYFILTER アイテム・コードに対して別の値を指定します。REG\$\_CHANGENAME または REG\$\_CHANGELASTSET を使用してください。

#### 2.1.2 データベース検索の制限

REG\$CP サーバ管理ユーティリティの SEARCH コマンドや、REG\$FC\_SEARCH\_TREE\_DATA、REG\$FC\_SEARCH\_TREE\_KEY、REG\$FC\_SEARCH\_TREE\_VALUE 機能コードを使用する \$REGISTRY システム・サービスの呼び出しでは、クライアント・ノードの通信バッファが取り扱うことができる量より多くのデータがクライアントに返されることがあります。これらの機能は 16 レベル未満のパスに制限されており、4 KB 未満のデータを返します。

この問題を回避するには、正確なパスを指定することで検索の深さを制限します。つまり、データベースが大きいときに、データベース全体を検索しないようにします。

この制限は将来のリリースで修正される予定です。

#### 2.1.3 キー・アクセス・ポリシー

ユーザが OpenVMS Registry のキーまたは値へのアクセスを要求した場合、OpenVMS Registry はキー・パスの最初のキーと最後のキーを確認することで、指定されたキー・パスが有効かどうか調べます。

## 2.1.4 OpenVMS Registry の最大データ・サイズの制限

OpenVMS Registry データベースに格納するために、OpenVMS Registry サーバに送信できるデータの 1 ブロックの最大サイズは 7880 バイト以下に制限されています。この制限を超えると、次のエラーが表示されます。

```
REG-F_NORESPONSE, registry server failed to respond within allotted time  
period
```

この制限は、OpenVMS Registry サーバと OpenVMS Registry クライアントの間の通信プロトコルで、転送サイズが 8 K バイトに制限されているために発生します。

この制限は将来のリリースで修正される予定です。

## 2.1.5 REG\$\_EXQUOTA エラー

OpenVMS Registry の File Qouta を OpenVMS Registry データベースの現在のサイズより小さい値に設定すると、すべての OpenVMS Registry 操作で次のエラーが表示されます。

```
REG-E-EXQUOTA, registry file quota or page file quota exceeded
```

---

### 注意

---

1 回の削除操作で、OpenVMS Registry データベース・ファイルのサイズがクォータ・リミット以内になる場合は、このエラー・メッセージは表示されません。

---

このエラーを回避するには、File Qouta の値を OpenVMS Registry データベース・ファイルの現在のサイズ以上の値に一時的に変更し、その後で OpenVMS Registry データベース・ファイルのサイズが適切な File Qouta の制限の範囲内になるように削除操作を実行します (クォータの変更の方法については、第 7.6.3 項を参照してください)。

2 つの OpenVMS Registry ファイル・クォータに対して適用されるおよそのバイト数を判断するには、SYS\$REGISTRY:REGISTRY\$LOCAL\_MACHINE.REG ファイルと SYS\$REGISTRY:REGISTRY\$USERS.REG ファイルのサイズに 512 を乗算します。この計算の結果は、各ファイルのクォータに適用されるおよそのバイト数になります。OpenVMS Registry のファイル・クォータの設定方法については、第 2.1.6 項を参照してください。

この制限は将来のリリースで修正される予定です。

## 2.1.6 OpenVMS Registry の最大データベース・サイズの制限

OpenVMS Registry データベースに格納できるデータの最大サイズは、約 1.7 MB に制限されています。この制限を超えると、次のエラーが表示されます。

```
REG-F-DBACCESS, cannot access registry database object
```

これは致命的なエラーであり、OpenVMS Registry データベースにこれ以降アクセスできなくなります。

REG-F-DBACCESSエラーが発生しないようにするには、OpenVMS Registry が 1.7 MB 以上になることがないように、クォータを設定してデータベースのサイズを制限します。

クォータを設定するには、次のいずれかの操作を行います。

- Default File Quota の値を変更します。

Default File Quota の値を設定するには、次のコマンドを使用します。

```
$ mcr reg$cp modify value/name="Default File Quota"/type=dword -  
_$_ /data=%D1700000 "hkey_local_machine\system\registry\File Quotas"
```

- OpenVMS Registry データベースの各ファイルに対して File Quota の値を設定または変更します。

各ファイルの File Quota の値を設定するには、次のコマンドを使用します。

---

### 注意

---

File Quota の値をすでに設定している場合は、次の例で CREATE コマンドの代わりに MODIFY コマンドを使用します。

---

```
$ mcr reg$cp create value/name=REGISTRY$LOCAL_MACHINE/type=dword -  
_$_ /data=%D1700000 "hkey_local_machine\system\registry\File Quotas"
```

```
$ mcr reg$cp create value/name=REGISTRY$USERS/type=dword -  
_$_ /data=%D1700000 "hkey_local_machine\system\registry\File Quotas"
```

/DATA 修飾子の値として、32,000 ~ 2,000,000 (16 進数の 0x7D00 ~ 0x1E8480) の値を指定しなければなりません。

32,000 未満の値を指定すると、その値は無視され、Default File Quotaの値として 10,000,000 が使用されます。この File Quota のデフォルト値は大きすぎます。

2,000,000 より大きい値を指定すると、OpenVMS Registry データベース・サイズがその値を超えたときに、REG-F-DBACCESSエラーが発生します。

この制限は将来のリリースで修正される予定です。



# 第1部

---

## COM for OpenVMS

この後の章では、COM for OpenVMS の概要、COM for OpenVMS および関連ソフトウェアのインストールおよびコンフィギュレーション方法、COM for OpenVMS を使用してのCOM アプリケーションの作成方法について説明します。



---

## COM for OpenVMS の概要

---

### 3.1 COM とは

COM (Component Object Model: コンポーネント・オブジェクト・モデル) は、Microsoft が開発したテクノロジーであり、開発者は COM を使用することで分散ネットワーク・オブジェクトを作成できます。COM は Microsoft が Windows 3.x 製品で最初に導入したものであり、その当時は OLE (Object Linking and Embedding: オブジェクトのリンクと埋め込み) という名前でした。COM は、新しいスタイルのマルチベンダ分散コンピューティングを採用し適合させるための、強力で広く利用できるメカニズムを提供し、その一方でソフトウェアへの新規投資を最低限に抑えることができます。

Digital Equipment Corporation (現在は Compaq Computer Corporation) と Microsoft は COM 仕様を共同開発しました。COM 仕様は、最初のリリースでは NetOLE (Network OLE) という名前であり、その後、DCOM (Distributed COM) という名前に変更され、現在はネットワーク機能を含んでいます。つまり、COM は分散ネットワーク・オブジェクトをサポートします。

COM はオブジェクト・ベースのプログラミング・モデルであり、ソフトウェアの相互運用性を促進するように設計されています。COM を使用すると、2 つ以上のアプリケーション (またはコンポーネント) を簡単に協調動作させることができます。オブジェクトが異なるベンダによって異なるプログラミング言語で個別に作成された場合でも、異なるオペレーティング・システムを稼動する異なるマシンで動作する場合でも、協調動作が可能です。協調動作機能をサポートするために、COM では、アプリケーションが相互にソフトウェア・オブジェクトとして接続できるようにするためのメカニズムを定義し、実装しています。

COM の実装は Windows NT、Windows 95™、Windows 98、OpenVMS、Compaq Tru64™ UNIX® で提供されており、他の UNIX プラットフォームでも提供されています。

#### 3.1.1 参考文献

COM および関連トピックの詳細については、次の参考文献を参照してください。

- COM に関する参考文献:

— 『*Inside COM/Microsoft's Component Object Model*』 Dale Rogerson 著、Microsoft Press、Redmond、WA、1997。ISBN: 1-57231-349-8。

## COM for OpenVMS の概要

### 3.1 COM とは

本書で使用している例は Dale Rogerson の書籍から抜粋したものであり，出版社の許可を受けて使用しています。

- 『*Essential COM*』 Don Box 著，Addison Wesley Longman，Reading，MA，1998. ISBN: 0-201-63446-5.
- 『*Effective COM*』 Don Box，Keith Brown，Tim Ewald，Chris Sells 共著，Addison Wesley Longman，Reading，MA，1998. ISBN: 0-201-37968-6.
- 『*DCOM Explained*』 Rosemary Rock-Evans 著，Digital Press，Woburn，MA，1998. ISBN: 1-55558-216-8.

DCOM と COM に関するわかりやすい入門書であり，さまざまなプラットフォームでの COM の実装が説明されています。

- 『*Understanding ActiveX and OLE*』 David Chappell 著，Microsoft Press，Redmond，WA，1996. ISBN: 1-57231-216-5.
- Web サイト:
  - *The Component Object Model Specification*，Microsoft COM に関する Web サイトからダウンロード可能:

[www.microsoft.com/com](http://www.microsoft.com/com)

---

## 3.2 COM for OpenVMS の概要

COM for OpenVMS は，Microsoft の Windows NT 4.0 Service Pack 3 (SP3) Component Object Model (COM) ソフトウェアを Compaq が OpenVMS Alpha オペレーティング・システム上で実装したものです。

COM for OpenVMS のサポートで，Compaq はレジストリ，イベント・ロガー，NTLM セキュリティ，Win32 API も含めて，Windows NT のインフラストラクチャを OpenVMS に移植しました。COM for OpenVMS は The Open Group の DCE (Distributed Computing Environment) RPC のレイヤードです。COM for OpenVMS では，ローカル・エリア・ネットワーク (LAN)，ワイド・エリア・ネットワーク (WAN)，インターネット上のさまざまなコンピュータのオブジェクト間での通信をサポートします。COM for OpenVMS は Windows NT と接続するための重要な機能を提供するので，Affinity for OpenVMS プログラムにとって重要です。

図 3-1 は OpenVMS のインフラストラクチャを示しています。

図 3-1 OpenVMS のインフラストラクチャと COM for OpenVMS

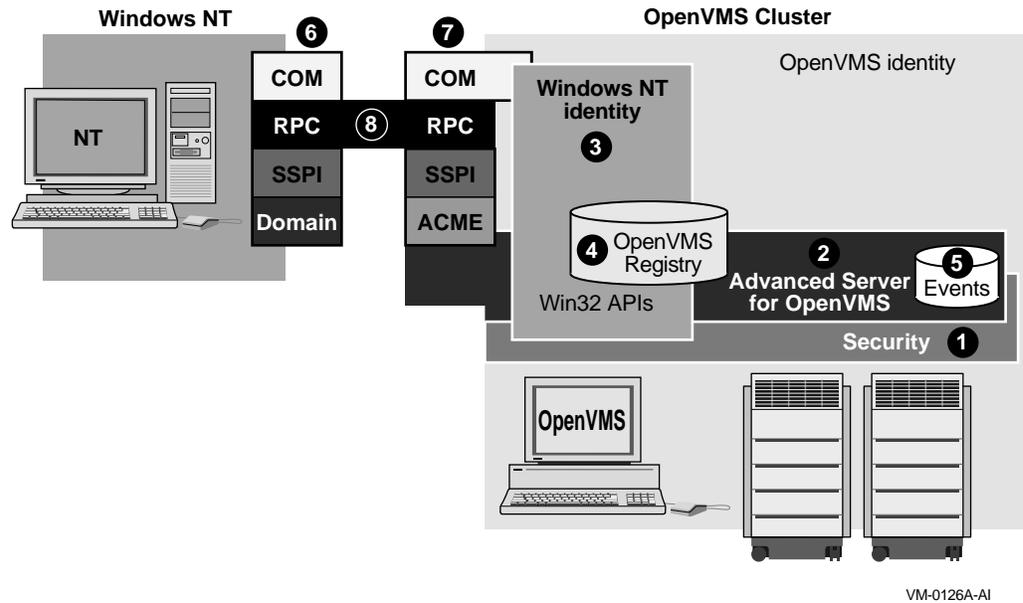


図 3-1 で， OpenVMS インフラストラクチャの重要な要素は次のとおりです。

#### Windows NT システム

図 3-1 の左側の小さいボックスは， Windows NT システムを表しています。

#### OpenVMS Cluster/OpenVMS アイデンティティ

図 3-1 の右側の大きいボックスは， OpenVMS システムを表しています。このボックスの内部および周囲には，番号の付いたいくつかのボックスがあります。ここではこれらの番号の付いたアイテムについて説明します。

#### 1 OpenVMS セキュリティ

これは，すべての OpenVMS システムで提供される標準の OpenVMS セキュリティ (ログイン，認証，ACL など) です。

#### 2 Advanced Server for OpenVMS

Advanced Server for OpenVMS は OpenVMS に対して Windows NT ユーザの認証機能を提供し， Windows NT ユーザに対して OpenVMS Registry とイベント・ビューアへの接続機能を提供します。

#### 3 Windows NT アイデンティティ/Win32 API

OpenVMS セキュリティ， MSV1\_0 ACME エージェント， Advanced Server for OpenVMS， OpenVMS Registry， イベント・ロガー， Win32 API (COM API) はすべて， OpenVMS システム内で Windows NT アイデンティティを作成することに貢献します。

#### 4 OpenVMS Registry

OpenVMS Registry は Windows NT システムのレジストリに類似しており、システム、ソフトウェア、ハードウェアの構成情報を OpenVMS に格納するための機能です。COM for OpenVMS では、OpenVMS Registry を使用して COM アプリケーションに関する情報を格納します。OpenVMS Registry の詳細については、本書の第 2 部を参照してください。

#### 5 イベント・ロガー

OpenVMS のイベント・ロガーは Windows NT システムのイベント・ロガーに類似しており、COM イベントに関する情報メッセージ、警告メッセージ、エラー・メッセージを記録します。OpenVMS Events の詳細については、第 11 章を参照してください。

#### 6 Windows NT COM スタック

Windows NT システムでは、COM 要求と応答は、COM、RPC、SSPI (セキュリティ)、ドメイン・レイヤを介してやり取りされます。

#### 7 OpenVMS COM スタック

OpenVMS システムでは、Windows NT の COM スタックをそのまま使用しますが、いくつかの機能が追加されています。OpenVMS システムでは、COM 要求と応答は、COM、RPC、SSPI (セキュリティ)、MSV1\_0 ACME エージェント、Advanced Server for OpenVMS レイヤを介してやり取りされます。MSV1\_0 ACME エージェント (図 3-1 では ACME) は ACM (Authentication and Credential and Management) オーソリティに対する拡張機能です。認証の詳細については、第 12 章を参照してください。

#### 8 RPC レイヤを介した接続

Windows NT システムと OpenVMS の間の COM 接続は、常に RPC レイヤを介して行われます。

開発者用に提供される COM for OpenVMS Developer's Kit には、アプリケーション開発用に MIDL (Microsoft Interface Definition Language) コンパイラと C スタイルのヘッダ・ファイルが含まれています。OpenVMS MIDL コンパイラの詳細については、第 3.4 節を参照してください。

OpenVMS では、Windows NT 資格情報を取得するための機能が提供されるようになりました。NTASLOGON を介して Windows NT 資格情報を取得する方法については、第 4.8 節と第 12 章を参照してください。

COM for OpenVMS では、COM for OpenVMS クライアント/サーバ・アプリケーションの導入のために、OpenVMS Alpha で無料の実行時環境が提供されるようになりました。

プロトコルの仕様とプログラミングのドキュメントを含めて、Microsoft の COM の詳細については、Microsoft COM の Web サイトを参照してください。アドレスは次のとおりです。

[www.microsoft.com/com](http://www.microsoft.com/com)

COM for OpenVMS の実装は、完全な Microsoft COM 実装のサブセットです。COM for OpenVMS API、サポートされるインタフェース、実装の相違点の一覧については、付録 E を参照してください。

COM に対する関心はますます高くなっていますが、それでも COM は依然として高度な技術です。経験の浅いユーザを対象としたものではなく、独立系ソフトウェア・ベンダ (ISV) や大規模な MIS (管理情報システム) ショップなど、高度な技能を持ったプログラマを対象にしています。

### 3.2.1 COM for OpenVMS での OpenVMS Registry の使用

COM for OpenVMS では、OpenVMS Registry が必要です。Windows NT システムのレジストリ・データベースと同様に、OpenVMS Registry には COM アプリケーションに関する情報、特に OpenVMS 上で実行されている COM アプリケーションに関する情報が格納されます。これらの COM for OpenVMS アプリケーションでは、OpenVMS Registry を使用して CLSID (クラス ID)、スタートアップ情報、セキュリティ設定などを OpenVMS Registry データベースに格納します。COM for OpenVMS では、OpenVMS で実装されている Win32 API を使用して、OpenVMS Registry からの読み込みと書き込みを行います。

COM for OpenVMS では、OpenVMS Registry データベースへのアクセスが必要です。COM for OpenVMS が OpenVMS Registry にアクセスできない場合は、COM for OpenVMS は起動しません。OpenVMS Registry の詳細については、第 7 章を参照してください。

---

## 3.3 COM for OpenVMS の使用

COM for OpenVMS を使用すると、次のことが可能です。

- 新しい COM for OpenVMS COM アプリケーションを開発できます。
- COM for OpenVMS と組み合わせて使用できるように、既存のアプリケーションをカプセル化できます。

ここでは、新規アプリケーションの開発とカプセル化について詳しく説明します。

### 3.3.1 新規アプリケーションの開発

次の場合には、COM for OpenVMS を使用して新規アプリケーションを開発できます。

- 2層クライアント/サーバ・コンピューティング・モデルで、OpenVMS サーバと Windows NT クライアントの間でデータを共有する場合。
- 3層コンピューティング環境の中央の層でデータを共有し、ビジネス・ロジックを配置する場合。

たとえば、Windows NT システムをクライアントとして使用すれば、そのグラフィカル・ユーザ・インタフェースを利用できます。ビジネス・ロジックは2番目の層のサーバでCOM オブジェクトの集合として作成できます。一方、3番目の層である大容量、高可用性の OpenVMS サーバでは、データベース・アクセス機能が提供されます。

- COM を使用して、1つ以上の OpenVMS システム間、または OpenVMS システムと Windows 以外のシステムの間でデータを共有する場合。

COM for OpenVMS には、次の利点があります。

- COM for OpenVMS は、C++ やオブジェクト指向プログラミングの技能を持つプログラマに対して、優れたプログラミング・モデルを提供します。
- COM for OpenVMS はマルチベンダの相互運用性を提供します。COM は現在、多くのプラットフォームで標準に提供されており、まだ提供されていないプラットフォームでも移植作業が進められています。
- COM for OpenVMS 実行時は、自動化されたデータのマーシャリング/アンマーシャリングを提供します。
- COM は、Microsoft Visual Basic®アプリケーションとの通信をサポートするために、OLE オートメーション・サービスを提供します。Visual Basic はクライアント/サーバ・コンピューティング用に非常に広く利用されているプログラミング環境です。
- COM はコンポーネントのバージョン・サポート機能を提供するので、既存の環境を妨害せずに、必要に応じてアプリケーションをアップグレードできます。

COM for OpenVMS アプリケーションの開発の例については、第6章と付録 C を参照してください。

### 3.3.2 既存のアプリケーションのカプセル化

手順型言語 (Fortran や COBOL など) で書かれ、キャラクタ端末インタフェースを使用するアプリケーションを使っている場合、COM “ラッパー”またはジャケットでこれらのアプリケーションを包み込むことにより、アプリケーションを新しいプラットフォームで実行することができるようになり、また、OpenVMS のクライアント/サーバ環境で実行することも可能になります。

従来のアプリケーションの中には、新しい技術に対応するように完全に書き直すと、非常に危険性の高いアプリケーションがあります。多くのアプリケーションは大規模で、複雑で、ドキュメントが不足しており、現在の保守担当者はこのようなアプリケーションを完全に理解できません。既存のアプリケーションをカプセル化すれば、作り直すより危険性が低くなり、また、書き直すための最初のステップとして利用することもできます。このようにしておけば、必要に応じて既存のアプリケーションの一部だけを書き直すことができ、その間もアプリケーションの以前のバージョンは安定した状態で利用できます。また、カプセル化では、開発者はコードを再利用でき、作業時間とリソースを節約できます。

一方、カプセル化には欠点もあります。保守作業が複雑になり、基礎になるコードを変更することができません。既存のアプリケーションが不安定だったり、保守が困難な場合は、カプセル化したアプリケーションがそれ以上のものになるわけではなく、ラッパーを追加したために、さらに状況が悪化する可能性もあります。

従来の手順型アプリケーションは複数の層でカプセル化することができます。ユーザ・インタフェース (UI)、データベース、データ操作ルーチンでカプセル化できます。

#### — ユーザ・インタフェース

ユーザ・インタフェース (UI) をカプセル化すると、UI を他のプラットフォームでサポートできるようになります (たとえば、Windows NT システムのグラフィカル・ユーザ・インタフェース [GUI] から)。

UI をカプセル化してユーザのデスクトップに移動しても、アプリケーションの他の部分は OpenVMS に残しておくことができます。バッチ処理プログラムは、このようなユーザ・インタフェースのカプセル化に適しています。画面管理を行うアプリケーション (たとえば、SMG や FMS) は、COM for OpenVMS を使用して従来のキャラクタ端末インタフェースをカプセル化することができ、ユーザは Windows NT スタイルのダイアログ・ボックスを通じてアクセスできるようになります。

#### — データベース

COM for OpenVMS を使用してデータベースをカプセル化すると、データベースは他のプラットフォームで稼働している分散アプリケーションの一部からアクセスできるようになります。この方法の場合、プログラマは安定した OpenVMS 上でデータベースを管理することができ、ユーザ・インタフェースとデータ・アクセス・ルーチンはリモートの (おそらく信頼性の低い) システムに置くことができるという利点があります。

#### — データベース操作ルーチン

データベース操作ルーチンをカプセル化すると、ルーチンは異種コンピューティング環境の他のどの COM コンポーネントからもアクセスできるようになります。

COM for OpenVMS を使用して OpenVMS アプリケーションをカプセル化するということは、カプセル化されるアプリケーションと会話する COM for OpenVMS サーバを作成することを意味します。COM for OpenVMS サーバは、アプリケーションが必要とする順序および形式で引数をアプリケーションに渡します。COM for OpenVMS サーバはアプリケーションからの出力を傍受して、それを表示装置、ユーザ・インタフェース、その他のルーチンに渡します。

---

## 3.4 OpenVMS MIDL コンパイラ

OpenVMS MIDL コンパイラは、次の点を除き、MIDL (Microsoft Interface Definition Language) コンパイラ V3.00.44 と同じです。

1. Microsoft MIDL では、複数の最適化レベルがサポートされます。一方、OpenVMS MIDL では、`-Oicf`だけがサポートされます。他の最適化レベルは使用できません。
2. `/cpp_cmd`スイッチと`/cpp_opt`スイッチは、OpenVMS MIDL では完全に機能しません。
3. Windows NT システムでは、Microsoft MIDL コマンド、スイッチ、修飾子で大文字と小文字が区別されます。OpenVMS MIDL コンパイラでは大文字と小文字が区別されません。OpenVMS MIDL コンパイラに渡されるすべてのコマンド、スイッチ、修飾子は小文字になります。このため、Microsoft MIDL のスイッチ `/I`と`/i`は OpenVMS では同じになります。
4. MIDL で生成されたファイルはプラットフォーム固有です。

MIDL は両方のプラットフォームで実行しなければなりません。どちらか一方のプラットフォーム (OpenVMS または Windows NT) で生成された MIDL 出力ファイルは、もう一方のプラットフォームにコピーしたり、使用することができません。

5. MIDL `-w`スイッチ

Microsoft MIDL コンパイラでは、`-w`または`-warn`を指定して、コンパイラで生成される警告のレベルを制限できます。OpenVMS MIDL コンパイラでは、`-w`スイッチだけがサポートされます。

---

## COM for OpenVMS キットのインストール

この章では、COM for OpenVMS キットの内容、必要なソフトウェア、インストールの前処理について説明します。また、COM for OpenVMS のインストールの方法とインストールの後処理についても説明します。

---

### 4.1 COM Version 1.1 for OpenVMS キットの内容

COM Version 1.1 for OpenVMS には、次のコンポーネントが含まれています。

- ソフトウェア
  - COM for OpenVMS 実行時ライブラリ
  - COM for OpenVMS MIDL コンパイラとヘッダ・ファイル
  - COM for OpenVMS 構成ユーティリティ
  - サンプル・アプリケーション
- ドキュメンテーション
  - 『OpenVMS Connectivity Developer Guide』 (PostScript 形式, HTML 形式, PDF 形式)  
(本書『OpenVMS コネクティビティ開発者ガイド』の英語版)

---

### 4.2 インストールの要件

次のソフトウェアが必要です。

- OpenVMS システム
  - OpenVMS バージョン 7.2-1 以上
  - DEC C バージョン 5.6 以上と DEC C++ バージョン 5.6 以上 (COM for OpenVMS アプリケーション開発の場合)
  - DIGITAL TCP/IP Services for OpenVMS Version 5.0 またはそれに相当するもの
  - Advanced Server for OpenVMS Version 7.2A 以上
  - COM for OpenVMS をインストールする前に、グローバル・ページ、グローバル・セクション、ディスク・ブロックに必要な空き領域があるかどうか確認してください。次の表は必要な空き領域を示しています。

## COM for OpenVMS キットのインストール

### 4.2 インストールの要件

ソフトウェア	グローバル・ページ	グローバル・セクション	ディスク・ブロック
COM for OpenVMS	11,000	27	57000
RPC 実行時	3,300	14	N/A

#### Advanced Server:

『日本語 Advanced Server for OpenVMS インストレーションおよび構成ガイド』を参照。

#### TCP/IP:

『DIGITAL TCP/IP Services for OpenVMS: Installation and Configuration』マニュアルを参照。

- Windows® NT™システム

- Windows NT 4.0 と Service Pack 3

- Microsoft® Visual C++ (Windows NT クライアント開発と、MIDL コンパイラに関する情報)。必要なコンパイラ・バージョンについては、Microsoft の Web サイトを参照。

- TCP/IP を有効に設定 (OpenVMS との接続のために必要)

---

## 4.3 サポートされる COM for OpenVMS のインストール・オプション

ここでは、COM Version 1.1 for OpenVMS のインストール・オプションとアップグレード・オプションについて説明します。

実行する操作	参照箇所
COM for OpenVMS を OpenVMS スタンドアロン・システムに初めてインストールする。	第 4.4 節を参照。
COM for OpenVMS を OpenVMS Cluster システムに初めてインストールする。	第 4.6 節を参照。
OpenVMS スタンドアロン・システムで COM Version 1.0 for OpenVMS から COM Version 1.1 for OpenVMS にアップグレードする。	第 4.5 節を参照。
OpenVMS Cluster システムで COM Version 1.0 for OpenVMS から COM Version 1.1 for OpenVMS にアップグレードする。	第 4.7 節を参照。

---

## 4.4 OpenVMS スタンドアロン・システムでの COM for OpenVMS のインストール

次の操作を行います。

1. OpenVMS Version 7.2-1 をインストールします。この手順については、『OpenVMS Alpha Version 7.x Upgrade and Installation Manual』を参照してください。

2. TCP/IP をインストールします。この手順については、『DIGITAL TCP/IP Services for OpenVMS: Installation and Configuration』マニュアルまたは TCP/IP サプライヤが提供するマニュアルを参照してください。
3. インストールしたシステムをシステム・ディスクからブートします。
4. COM Version 1.1 for OpenVMS をインストールします。この手順については、第 4.10 節を参照してください。
5. Advanced Server for OpenVMS をインストールします。この手順については、『日本語 Advanced Server for OpenVMS インストールおよび構成ガイド』を参照してください。
6. TCP/IP を構成し (スタートアップとリブートのための設定)、起動します。この手順については、『DIGITAL TCP/IP Services for OpenVMS: Installation and Configuration』マニュアルまたは TCP/IP サプライヤのマニュアルを参照してください。
7. 次の手順で OpenVMS Registry を構成します。
  - REG\$CONFIG を実行して、OpenVMS Registry を構成します。第 8.2 節を参照してください。
  - SYLOGICALS.COM ファイルを編集して、次のように SYS\$REGISTRY 論理名を定義します。

```
§ DEFINE/SYSTEM SYS$REGISTRY directory-specification
```
8. REG\$STARTUP.COM ファイルを実行して、OpenVMS Registry を起動します。
9. DCE を実行する場合は、ここで起動します。

---

**注意**

---

DCE を使用しなくても COM for OpenVMS を実行することはできますが、環境で DCE を使用する場合は、ここで起動してください。

---

この手順については、『DIGITAL DCE Installation and Configuration Guide』を参照してください。

OpenVMS 外部認証の詳細については、第 4.8 節を参照してください。

10. Advanced Server for OpenVMS を構成します。Advanced Server for OpenVMS の構成を終了するには、最後にリブートする必要があります。システム構成に応じて、0 ~ *n* 回リブートする必要があります。この手順については、『日本語 Advanced Server for OpenVMS インストールおよび構成ガイド』を参照してください。
11. Advanced Server for OpenVMS を起動します (リブート時にスタートアップするための設定)。この手順については、『日本語 Advanced Server for OpenVMS インストールおよび構成ガイド』を参照してください。

12. 次のコマンドを使用して、ACME サーバを起動します。

```
$ @SYS$STARTUP:NTA$STARTUP_NT_ACME
```

13. 次のコマンドを使用して、RPC を起動します。

```
$ @SYS$STARTUP:DCE$RPC_STARTUP.COM
```

14. COM for OpenVMS を構成します。この手順については、第 4.11 節と第 5.2 節を参照してください。

- OpenVMS Registry に情報を登録します。この手順については、第 5.2 節を参照してください。OpenVMS Registry データベースに情報を登録するには、オプション 3 を使用します。
- COM for OpenVMS Service Control Manager で必要な OpenVMS アカウントと Advanced Server for OpenVMS アカウントを作成します。詳細については、第 5.2 節を参照してください。アカウントを作成するには、オプション 8 を使用します。

15. SYLOGICALS.COM ファイルを編集し、次の行を追加します。

```
$ DEFINE DCOM$TO_BE_STARTED TRUE
```

16. COM for OpenVMS を起動します。この手順については、第 4.12 節を参照してください。

---

## 4.5 OpenVMS スタンドアロン・システムでの COM Version 1.0 for OpenVMS からのアップグレード

---

### 注意

---

アップグレードを開始する前に、Advanced Server for OpenVMS、OpenVMS Registry、その他のレイヤード製品の自動スタートアップを無効にして、COM for OpenVMS とその関連コンポーネントがアップグレードされるまで、これらの製品が起動されないようにしてください。

次の操作を行います。

- 次の製品が起動されないように、SYLOGICALS.COM ファイルを編集します。
  - OpenVMS Registry (DEFINE REG\$TO\_BE\_STARTED TRUE または DEFINE /SYSTEM REG\$TO\_BE\_STARTED TRUE という行を削除)
  - COM for OpenVMS (DEFINE DCOM\$TO\_BE\_STARTED TRUE という行をコメントに変更)
- 次の製品が起動されないように、SYS\$STARTUP:SYSTARTUP\_VMS.COM ファイルを編集します。
  - Advanced Server for OpenVMS (@SYS\$STARTUP:PWRK\$STARTUP.COM という行をコメントに変更)

COM for OpenVMS が実行中の場合は、最初に COM for OpenVMS をシャットダウンし、次に Advanced Server for OpenVMS をシャットダウンし (実行されている場合)、最後に OpenVMS Registry をシャットダウンします。

---

次の操作を行います。

1. OpenVMS Version 7.2-1 にアップグレードします。この手順については、『OpenVMS Alpha Version 7.x Upgrade and Installation Manual』を参照してください。
2. TCP/IP をアップグレードする場合は、ここでアップグレードしてください。この手順については、『DIGITAL TCP/IP Services for OpenVMS: Installation and Configuration』マニュアルまたは TCP/IP サプライヤのマニュアルを参照してください。
3. アップグレードしたシステムをシステム・ディスクからブートします。
4. COM for OpenVMS をアップグレードします。この手順については、第 4.10 節を参照してください。
5. Advanced Server for OpenVMS をインストールまたはアップグレードします。システム構成に応じて、0 ~  $n$  回リブートする必要があります。この手順については、『日本語 Advanced Server for OpenVMS インストレーションおよび構成ガイド』を参照してください。
6. リブート時に起動するように TCP/IP を有効に設定した場合を除き、TCP/IP を起動します。この手順については、『DIGITAL TCP/IP Services for OpenVMS: Installation and Configuration』マニュアルまたは TCP/IP サプライヤのマニュアルを参照してください。
7. リブート時に起動するように OpenVMS Registry を有効に設定した場合を除き、OpenVMS Registry を起動します。この手順については、第 8.2 節を参照してください。
8. DCE を実行する場合は、ここで DCE を起動します。

---

**注意**

---

DCE を使用しなくても COM for OpenVMS を実行することはできますが、環境で DCE を使用する場合は、ここで起動してください。

---

この手順については、『DIGITAL DCE Installation and Configuration Guide』を参照してください。

OpenVMS 外部認証の詳細については、第 4.8 節を参照してください。

9. Advanced Server for OpenVMS を構成します。Advanced Server for OpenVMS の構成を終了するには、リブートしなければなりません。システム構成に応じて 0 ~  $n$  回リブートする必要があります。この手順については、『日本語 Advanced

Server for OpenVMS インストレーションおよび構成ガイド』を参照してください。

10. Advanced Server for OpenVMS を起動します (リブート時にスタートアップするための設定)。この手順については、『日本語 Advanced Server for OpenVMS インストレーションおよび構成ガイド』を参照してください。

11. 次のコマンドを使用して、ACME サーバを起動します。

```
$ @SYS$STARTUP:NTA$STARTUP_NT_ACME
```

12. 次のコマンドを使用して、RPC を起動します。

```
$ @SYS$STARTUP:DCE$RPC_STARTUP.COM
```

13. COM Version 1.0 for OpenVMS から COM Version 1.1 for OpenVMS へのアップグレードの詳細については、付録 D を参照してください。

14. COM for OpenVMS を構成します。この手順については、第 4.11 節と第 5.2 節を参照してください。

- OpenVMS Registry に情報を登録します。この手順については、第 5.2 節を参照してください。OpenVMS Registry データベースに情報を登録するには、オプション 3 を使用します。
- COM for OpenVMS Service Control Manager で必要な OpenVMS アカウントと Advanced Server for OpenVMS アカウントを作成します。詳細については、第 5.2 節を参照してください。アカウントを作成するには、オプション 8 を使用します。

15. SYLOGICALS.COM ファイルに次の行を追加します。

```
$ DEFINE DCOM$TO_BE_STARTED TRUE
```

16. COM for OpenVMS を起動します。この手順については、第 4.12 節を参照してください。

---

## 4.6 OpenVMS Cluster での COM for OpenVMS のインストール

---

### 注意

---

ここで説明するクラスタ・インストール手順では、1 つのシステム・ディスクに COM for OpenVMS をインストールするものと想定しています。

---

次の操作を行います。

1. 必要に応じて、すべてのシステム・ディスクに OpenVMS Version 7.2-1 をインストールします。この手順については、『OpenVMS Alpha Version 7.x Upgrade and Installation Manual』を参照してください。

2. TCP/IP をインストールします。この手順については、『DIGITAL TCP/IP Services for OpenVMS: Installation and Configuration』マニュアルまたは TCP/IP サプライヤのマニュアルを参照してください。
3. インストールしたシステムをシステム・ディスクからブートします。
4. COM Version 1.1 for OpenVMS をインストールします。この手順については、第 4.10 節を参照してください。
5. クラスタ内のこのノードに Advanced Server for OpenVMS をインストールします。この手順については、『日本語 Advanced Server for OpenVMS インストールおよび構成ガイド』を参照してください。

---

#### 注意

---

クラスタ内の少なくとも 1 つの Alpha ノードに Advanced Server for OpenVMS をインストールしなければなりません。他のノードには Advanced Server for OpenVMS をインストールするか、または「External Authentication images (only)」を選択することができます。

---

6. TCP/IP を構成し (各ノードでリブート時にスタートアップするための設定)、TCP/IP を起動します。この手順については、『DIGITAL TCP/IP Services for OpenVMS: Installation and Configuration』マニュアルまたは TCP/IP サプライヤのマニュアルを参照してください。
7. OpenVMS Registry を構成します。
  - REG\$CONFIG を実行して、OpenVMS Registry を構成します。OpenVMS Registry はクラスタ全体で 1 回だけ構成しなければなりません。第 8.2 節を参照してください。
  - クラスタ内で OpenVMS Registry サーバを実行するすべての Alpha ノードで、SYS\$REGISTRY 論理名を DEFINE/SYSTEM に設定します。
  - クラスタ内のすべてのノードで、SYLOGICALS.COM ファイルを次のように編集します。
    - クラスタで共通の SYLOGICALS.COM ファイルを 1 つだけ使用し、各ノードの SYLOGICALS.COM ファイルからその共通ファイルを呼び出す場合は、変更は必要ありません。
    - OpenVMS Registry サーバを実行しないノードで、次の行を SYLOGICALS.COM ファイルに追加します。

```
$ DEFINE/SYSTEM REG$TO_BE_STARTED FALSE
```

Advanced Server for OpenVMS を使用するには、クラスタ内のノードで OpenVMS Registry を実行している必要があります。

8. DCE を構成します。

---

**注意**

---

DCE を使用しなくても COM for OpenVMS を実行することはできますが、環境で DCE を使用する場合は、ここで起動してください。

---

この手順については、『DIGITAL DCE Installation and Configuration Guide』を参照してください。

9. DCE を実行する場合は、ここで起動します。DCE を実行する各ノードで DCE を構成しなければなりません。

OpenVMS 外部認証の詳細については、第 4.8 節を参照してください。

10. Advanced Server for OpenVMS を構成し、起動します。この手順については、『日本語 Advanced Server for OpenVMS インストレーションおよび構成ガイド』を参照してください。

このノードで Advanced Server for OpenVMS を実行している場合は、リブート時に起動するように Advanced Server for OpenVMS を設定します (必要に応じて SYS\$STARTUP ファイルを編集します)。システム構成に応じて、0 ~ *n* 回リブートする必要があります。

このノードで Advanced Server for OpenVMS を実行していない場合は、SYLOGICALS.COM ファイルを編集し、PWRK\$ACME\_SERVER 論理名を定義します。この手順については、『日本語 Advanced Server for OpenVMS インストレーションおよび構成ガイド』を参照してください。PWRK\$ACME\_SERVER 論理名の詳細については、表 12-2 を参照してください。

11. ACME サーバを起動します。次のコマンドを使用します。

```
$ @SYS$STARTUP:NTA$STARTUP_NT_ACME
```

12. RPC を起動します。次のコマンドを使用します。

```
$ @SYS$STARTUP:DCE$RPC_STARTUP.COM
```

13. COM for OpenVMS を構成します。この手順については、第 4.11 節と第 5.2 節を参照してください。

- OpenVMS Registry に情報を登録します。この手順については、第 5.2 節を参照してください。OpenVMS Registry データベースに情報を登録するには、オプション 3 を使用します。クラスタで 1 回だけ、OpenVMS Registry に情報を登録する必要があります。
- COM for OpenVMS Service Control Manager で必要な OpenVMS アカウントと Advanced Server for OpenVMS アカウントを作成します。詳細については、第 5.2 節を参照してください。アカウントを作成するには、オプション 8 を使用します。これらのアカウントはクラスタ内で 1 回だけ作成する必要があります。

14. SYLOGICALS.COMファイルに次の行を追加します。

```
$ DEFINE DCOM$TO_BE_STARTED TRUE
```

15. COM for OpenVMS を起動します。この手順については、第 4.12 節を参照してください。

---

## 4.7 OpenVMS Cluster での COM for OpenVMS のアップグレード

---

### 注意

ここで説明するクラスタ・アップグレード手順では、1つのシステム・ディスクに COM for OpenVMS をインストールするものと想定しています。

---

---

### 注意

アップグレードを開始する前に、Advanced Server for OpenVMS とレイヤード製品の自動スタートアップを無効にして、COM for OpenVMS とその関連コンポーネントがアップグレードされるまで、これらの製品が起動されないようにしてください。

次の操作を行います。

- 次の製品が起動されないように、SYLOGICALS.COMファイルを編集します。
  - OpenVMS Registry ( DEFINE REG\$TO\_BE\_STARTED TRUEまたはDEFINE /SYSTEM REG\$TO\_BE\_STARTED TRUEという行を削除)
  - COM for OpenVMS ( DEFINE DCOM\$TO\_BE\_STARTED TRUEという行をコメントに変更)
- 次の製品が起動されないように、SYS\$STARTUP:SYSTARTUP\_VMS.COMファイルを編集します。
  - Advanced Server for OpenVMS ( @SYS\$STARTUP:PWRK\$STARTUP.COMという行をコメントに変更)

COM for OpenVMS が実行中の場合は、クラスタ内のすべてのノードで最初に COM for OpenVMS をシャットダウンし、次に Advanced Server for OpenVMS をシャットダウンし(実行されている場合)、最後に OpenVMS Registry をシャットダウンします。

---

次の操作を行います。

1. 必要なすべてのシステム・ディスクで OpenVMS Version 7.2-1 にアップグレードします。この手順については、『OpenVMS Alpha Version 7.x Upgrade and Installation Manual』を参照してください。
2. TCP/IP をアップグレードします。この手順については、『DIGITAL TCP/IP Services for OpenVMS: Installation and Configuration』マニュアルまたは TCP/IP サプライヤのマニュアルを参照してください。

3. アップグレードしたシステムをシステム・ディスクからブートします。
4. COM Version 1.1 for OpenVMS にアップグレードします。この手順については、第 4.10 節を参照してください。
5. クラスタ内のこのノードで Advanced Server for OpenVMS をアップグレードします。この手順については、『日本語 Advanced Server for OpenVMS インストールレーションおよび構成ガイド』を参照してください。

---

**注意**

---

クラスタ内の少なくとも 1 つの Alpha ノードに Advanced Server for OpenVMS をインストールしなければなりません。他のノードでは、Advanced Server for OpenVMS をインストールするか、または「External Authentication images (only)」を選択できます。

---

6. TCP/IP を構成します (各ノードでリブート時にスタートアップするための設定)。この手順については、『DIGITAL TCP/IP Services for OpenVMS: Installation and Configuration』マニュアルまたは TCP/IP サプライヤのマニュアルを参照してください。
7. 次の手順で OpenVMS Registry を構成します。
  - REG\$CONFIG を実行して、OpenVMS Registry を構成します。第 8.2 節を参照してください。OpenVMS Registry はクラスタで 1 回だけ構成する必要があります。
  - OpenVMS Registry サーバを実行するすべてのノードで SYLOGICALS.COM ファイルを編集して、SYS\$REGISTRY 論理名を定義します。次の例を参照してください。

```
$ DEFINE/SYSTEM SYS$REGISTRY  
cluster-visible-directory-specification
```

クラスタ内のすべてのノードで、SYLOGICALS.COM ファイルを次のように編集します。

- クラスタで共通の SYLOGICALS.COM ファイルを 1 つだけ使用し、各ノードの SYLOGICALS.COM ファイルからその共通ファイルを呼び出す場合は、変更は必要ありません。
- OpenVMS Registry サーバを実行しないノードで、次の行を SYLOGICALS.COM ファイルに追加します。

```
$ DEFINE/SYSTEM REG$TO_BE_STARTED FALSE
```

8. Advanced Server for OpenVMS を構成し、起動します。この手順については、『日本語 Advanced Server for OpenVMS インストールレーションおよび構成ガイド』を参照してください。

このノードで Advanced Server for OpenVMS を実行する場合は、リポート時に起動するように Advanced Server for OpenVMS を設定します (必要に応じて SYSSSTARTUP ファイルを編集します)。システム構成に応じて、0 ~  $n$  回リポートする必要があります。

このノードで Advanced Server for OpenVMS を実行していない場合は、SYLOGICALS.COM ファイルを編集して、PWRK\$ACME\_SERVER 論理名を定義します。この手順については、『日本語 Advanced Server for OpenVMS インストールेशनおよび構成ガイド』を参照してください。PWRK\$ACME\_SERVER 論理名の詳細については、表 12-2 を参照してください。

9. ACME サーバを起動します。次のコマンドを使用します。

```
$ @SYS$STARTUP:NTA$STARTUP_NT_ACME
```

10. RPC を起動します。次のコマンドを使用します。

```
$ @SYS$STARTUP:DCE$RPC_STARTUP.COM
```

11. COM Version 1.0 for OpenVMS から COM Version 1.1 for OpenVMS へのアップグレードの詳細については、付録 D を参照してください。

12. COM for OpenVMS を構成します。この手順については、第 4.11 節と第 5.2 節を参照してください。

- OpenVMS Registry に情報を登録します。この手順については、第 5.2 節を参照してください。OpenVMS Registry データベースに情報を登録するには、オプション 3 を使用します。クラスタで 1 回だけ、OpenVMS Registry に情報を登録する必要があります。
- COM for OpenVMS Service Control Manager で必要な OpenVMS アカウントと Advanced Server for OpenVMS アカウントを作成します。詳細については、第 5.2 節を参照してください。アカウントを作成するには、オプション 8 を使用します。これらのアカウントはクラスタ内で 1 回だけ作成する必要があります。

13. SYLOGICALS.COM ファイルに次の行を追加します。

```
$ DEFINE DCOM$TO_BE_STARTED TRUE
```

14. 特定のノードで COM for OpenVMS を起動します。この手順については、第 4.12 節を参照してください。

---

## 4.8 OpenVMS の外部認証

COM for OpenVMS では、COM オブジェクトへのアクセスを制御するために、NTLM セキュリティ資格情報を使用します。OpenVMS ユーザは NTA\$LOGON ユーティリティを使用して、Advanced Server for OpenVMS (PATHWORKS) 機能で定義されている Windows NT ユーザを認証し、必要な資格情報を取得できます。こ

のユーティリティは、ACM (Authentication and Credential Management) オペレーティビティに対するユーザ・インタフェースです。

OpenVMS での認証では、Windows NT のユーザ名を SYSUAF.DAT ファイルでローカルの OpenVMS アカウントにマップする必要があります。デフォルトでは、外部 (NTLM) 認証を行うには、Windows NT ユーザ名にマップされる OpenVMS アカウントに対して、EXTAUTH フラグをセットする必要があります。EXTAUTH フラグをセットしないと、OpenVMS は必要なマップを実行できません。フラグがセットされていない場合は、認証は失敗します。

たとえば、Advanced Server for OpenVMS (PATHWORKS) ユーザ・データベースで Windows NT ユーザ名 SMITH が定義されていて、ホスト・マップ・データベースで Windows NT ユーザ名 SMITH と、JSMITH という OpenVMS アカウントの間の対応関係が定義されている場合、OpenVMS システムの JSMITH の SYSUAF.DAT レコードで、EXTAUTH フラグが有効に設定されていなければなりません。EXTAUTH フラグが有効に設定されていない場合や、Windows NT ユーザ名のホスト・マップ・エントリが削除されている場合は、外部認証は失敗します。

SECURITY\_POLICY sysgen パラメータで IGNORE\_EXTAUTH フラグ (ビット 11, %X0800) をセットしておけば、外部認証のために使用するすべての OpenVMS アカウントで EXTAUTH フラグを有効にする必要がなくなります。IGNORE\_EXTAUTH フラグを有効にすると、OpenVMS ユーザは、EXTAUTH フラグが有効になっていない OpenVMS アカウントに Windows NT ユーザ名がマップされる場合でも、NTASLOGON ユーティリティを使用して資格情報を取得できます。

#### 4.8.1 外部認証での LOGINOUT.EXE の使用

EXTAUTH フラグはまた、ログオン・プロセス (つまりローカル・ログイン、ダイアルアップ・ログイン、リモート・ログイン、会話型ログイン、ネットワーク・ログイン) で OpenVMS ユーザを認証するために、外部認証を使用するように LOGINOUT.EXE に要求します。EXTAUTH フラグを有効にすると、LOGINOUT.EXE は SYSUAF.DAT レコードのパスワードではなく、外部認証を使用して OpenVMS ユーザ名とパスワードを確認します。

LOGINOUT 外部認証では、必ず SYSUAF アカウント・レコードで EXTAUTH フラグを有効にしておく必要があります。NTASLOGON や認証済み RPC と異なり、IGNORE\_EXTAUTH フラグを使用しても、この必要条件を無効にすることはできません。

## 4.8.2 DCE 統合ログインの制限

あるサイトの同じシステムで、外部認証と以前の LGI コールアウト機能の両方を使用することはできません。LGI コールアウト・イメージがインストールされている場合は、ログイン用の外部認証は無効になります。DCE 統合ログインでは LGI コールアウト・メカニズムを使用するため、OpenVMS では、DCE 統合ログインが有効に設定されている場合、Windows NT ベースの外部認証を使用してログインを行うことはできません。

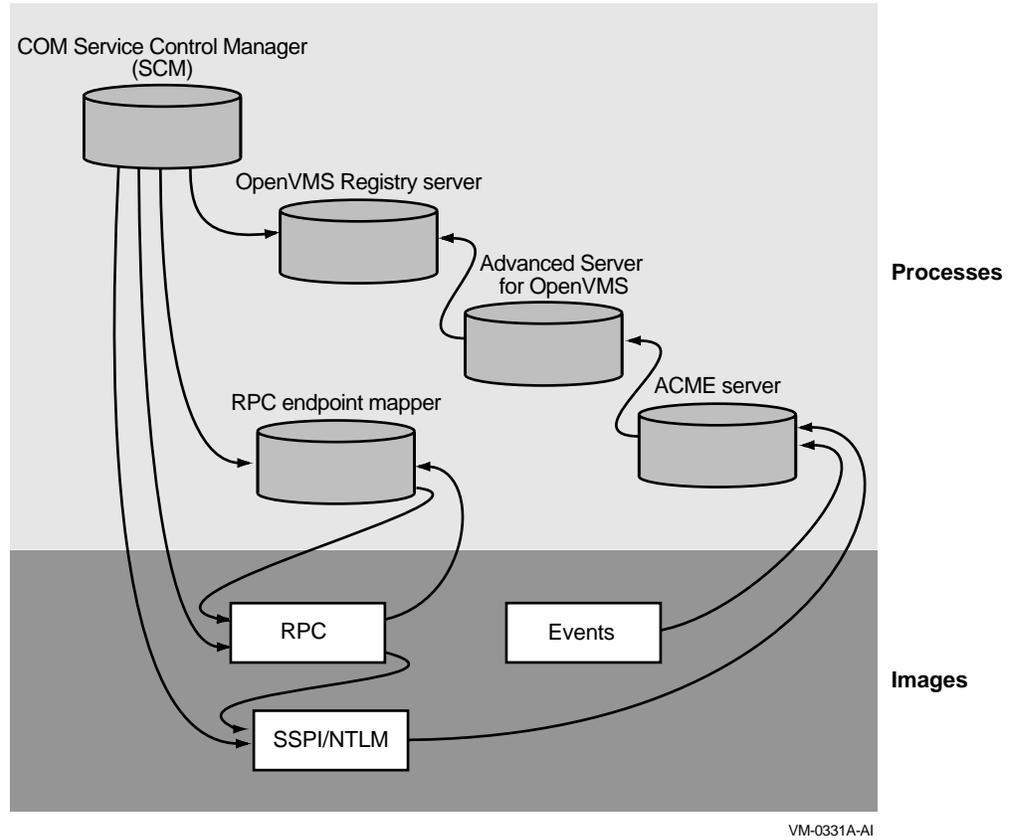
---

## 4.9 COM for OpenVMS 環境とは

COM for OpenVMS は相互に関係する多くのサーバ(プロセス)とオペレーティング・システム・イメージに依存します。ほとんどの場合、システムを再起動すると、サーバは自動的に起動します(自動的な起動を行うには、各コンポーネントをインストールし、構成しておくことが必要で、SYLOGICALS.COM ファイルも必要に応じて変更しなければなりません)。サーバの起動と構成の詳細については、第 4.3 節を参照してください。

図 4-1 は、プロセスとオペレーティング・システム・レイヤの関係と依存関係を示しています。

図 4-1 プロセスとレイヤの関係



VM-0331A-AI

表 4-1 は、プロセス名と、それに対応するサーバの関係を示しています。

表 4-1 プロセス名とサーバ名の関係

プロセス名	サーバ名	参照箇所
DCOM\$RPCSS	COM for OpenVMS Service Control Manager (SCM)	第 4.9.1 項
REGISTRY_SERVER	OpenVMS Registry サーバ	第 4.9.2 項
PWRKxxx	Advanced Server for OpenVMS サーバ (複数のプロセス)	第 4.9.3 項
ACME_SERVER	ACME サーバ	第 4.9.4 項
DCE\$RPCD	RPC エンドポイント・マップ	第 4.9.5 項

この節では、サーバとレイヤについて説明します。

#### 4.9.1 COM for OpenVMS Service Control Manager (SCM)

COM for OpenVMS Service Control Manager は COM for OpenVMS を有効にします。

プロセス名： DCOM\$RPCSS

必須コンポーネント： OpenVMS Registry , OpenVMS (RPC および SSPI/NTLM レイヤ)

このプロセスを必要とするコンポーネント： COM アプリケーション

構成： DCOM\$SETUP。第 5.2 節を参照。

起動： DCOM\$SETUP のオプション 4。第 5.2 節を参照。

シャットダウン・プロシージャ： DCOM\$SETUP のオプション 5。第 5.2 節を参照。

#### 4.9.2 OpenVMS Registry サーバ

OpenVMS Registry サーバは OpenVMS Registry データベースを管理します。

プロセス名： REGISTRY\_SERVER

必須コンポーネント： なし

このプロセスを必要とするコンポーネント： COM for OpenVMS , Advanced Server for OpenVMS

構成： REG\$CONFIG。第 8.2 節を参照。

起動： REG\$STARTUP。第 8.3.1 項を参照。

シャットダウン・プロシージャ： SET SERVER REGISTRY\_SERVER/EXIT。詳細については、第 8.4 節を参照。

#### 4.9.3 Advanced Server for OpenVMS サーバ

Advanced Server for OpenVMS サーバは Windows NT と OpenVMS の間の接続機能を提供します。

プロセス名：

NETBIOS  
PWRK\$ADMIN\_0  
PWRK\$KNBDAEMON

PWRK\$LICENSE\_R  
PWRK\$LMBROWSER  
PWRK\$LMDMN  
PWRK\$LMMCP  
PWRK\$LMSRV  
PWRK\$MASTER  
PWRK\$MONITOR  
PWRK\$NBDAEMON

ACME サーバを使用するには、PWRK\$LMSRV プロセスが必要です。

必須コンポーネント： OpenVMS Registry

このプロセスを必要とするコンポーネント： ACME サーバ

構成： PWRK\$CONFIG

起動： PWRK\$STARTUP

シャットダウン・プロシージャ： PWRK\$SHUTDOWN

詳細については、『DIGITAL PATHWORKS for OpenVMS (Advanced Server) Server Migration Guide』を参照してください。

#### 4.9.4 ACME サーバ

ACME サーバは資格情報の付与を制御します。

プロセス名： ACME\_SERVER

必須コンポーネント： Advanced Server for OpenVMS

このプロセスを必要とするコンポーネント： OpenVMS (RPC および SSPI/NTLM レイヤ) と OpenVMS Events

起動：

- SYLOGICALS.COMファイルに次の行が格納されている場合は、自動的に起動されません。NTA\$NT\_ACME\_TO\_BE\_STARTED YES
- 次のコマンドを入力して、ACME サーバを手動で起動することもできます。

```
$ @SYS$STARTUP:NTA$STARTUP_NT_ACME
```

シャットダウン・プロシージャ：

```
$ SET SERVER ACME {/EXIT | /ABORT}
```

詳細については、第 12.3.3 項を参照してください。

#### 4.9.5 RPC エンドポイント・マップ

RPC エンドポイント・マップは認証とセキュリティを制御します。

プロセス名： DCE\$RPCD

必須コンポーネント： RPC イメージ

このプロセスを必要とするコンポーネント： COM for OpenVMS Service Control Manager, RPC イメージ

起動： OpenVMS

シャットダウン・プロシージャ：次のコマンド・プロシージャを使用します。

```
$ @SYS$STARTUP:DCE$RPC_SHUTDOWN.COM
```

詳細については、『DIGITAL DCE for OpenVMS VAX and OpenVMS Alpha』マニュアルを参照してください。

#### 4.9.6 RPC および SSPI/NTLM レイヤ

RPC および SSPI/NTLM レイヤは、OpenVMS でリモート・プロシージャ・コールと Windows NT スタイルの認証機能を提供します。

プロセス名：なし (OpenVMS オペレーティング・システムの一部)

必須コンポーネント： OpenVMS, ACME サーバ

このプロセスを必要とするコンポーネント： COM for OpenVMS

起動： OpenVMS

シャットダウン・プロシージャ：なし

#### 4.9.7 OpenVMS Events

Events レイヤは OpenVMS で Windows NT スタイルのイベント・ログ機能を提供します。

プロセス名：なし (OpenVMS オペレーティング・システムの一部)

必須コンポーネント： ACME サーバ

このプロセスを必要とするコンポーネント： COM for OpenVMS

起動： OpenVMS

シャットダウン・プロシージャ：なし

詳細については、第 11 章を参照してください。

---

## 4.10 COM for OpenVMS のインストール

COM for OpenVMS インストール・キットには、POLYCENTER Software Installation ファイルが格納されています。キット名はDEC-AXPVMS-DCOM-V0101--1.PCSIです。COM for OpenVMS ファイルを OpenVMS Alpha バージョン 7.2-1 システムにインストールしなければなりません。キットをインストールする前に、第 4.2 節を参照して、インストールの要件を確認してください。

COM for OpenVMS をインストールするには、次のコマンドを使用して POLYCENTER Software Installation コーティリティを呼び出します。

```
$ PRODUCT INSTALL /SOURCE=device:[user] DCOM
```

*device:[user]*に対して、キットのデバイス名とディレクトリの場所をそれぞれ指定します。

---

### MIDL コンパイラ・ライセンス

---

COM for OpenVMS MIDL コンパイラにはライセンスが必要です。DCOM-MIDL PAK は COM for OpenVMS キットと別個に提供されます。

---

例 4-1 はインストールの例を示しています。

#### 例 4-1 COM for OpenVMS のインストールの例

```
$ product install dcom/source=disk:[directory]
The following product has been selected:
    DEC AXPVMS DCOM V1.1                Layered Product [Installed]
Do you want to continue? [YES]
Configuration phase starting ...
You will be asked to choose options, if any, for each selected product and
for
any products that may be installed to satisfy software dependency
requirements.
DEC AXPVMS DCOM V1.1
    Copyright Compaq Computer Corporation 1999. All rights reserved.
Do you want the defaults for all options? [YES]
    The following software is required to run COM for OpenVMS
    - OpenVMS Alpha V7.2-1 or later
    - Includes DCE RPC and OpenVMS Registry
    - TCP/IP Services for OpenVMS V5.0 or later (or equivalent product)
    - Advanced Server for OpenVMS V7.2A or later
    - In order to use the DCOM MIDL compiler you must install
      the DCOM-MIDL License PAK
    Do you want to continue? [YES]
Do you want to review the options? [NO]
Execution phase starting ...
The following product will be installed to destination:
    DEC AXPVMS DCOM V1.1                DISK$AXP_72PLUS:[VMS$COMMON.]
Portion done:
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
The following product has been installed:
    DEC AXPVMS DCOM V1.1                Layered Product
$
```

---

## 4.11 COM for OpenVMS のインストールの後処理

COM for OpenVMS キットをインストールした後、次の操作を行います。

1. OpenVMS Registry が実行されていることを確認します (第 8 章を参照してください)。
2. Advanced Server for OpenVMS が実行されていることを確認します (Advanced Server for OpenVMS プロセス名については、第 4.9.3 項を参照してください)。
3. ACME サーバが実行されていることを確認します (このプロセスの名前については、第 4.9.4 項を参照してください)。

4. RPC デーモンが実行されていることを確認します (プロセスの名前については、第 4.9.5 項を参照してください)。
5. DCOM\$SETUPユーティリティのオプション 3 を使用して、COM for OpenVMS の必要なキーと値を OpenVMS Registry に登録します (第 5.2 節を参照してください)。この操作は OpenVMS クラスタで 1 回だけ実行しなければなりません。
6. DCOM\$SETUPユーティリティのオプション 8 を使用して、DCOM\$RPCSS アカウントを構成します (第 5.2.1 項を参照してください)。この操作は OpenVMS クラスタで 1 回だけ実行しなければなりません。
7. DCOM\$SETUPユーティリティのオプション 4 を使用して、COM for OpenVMS を起動します (第 5.2 節を参照してください)。この操作は OpenVMS クラスタの各ノードで実行しなければなりません。
8. システムをリブートするときに COM for OpenVMS を自動起動するには、SYLOGICALS.COMファイルでDEFINE DCOM\$TO\_BE\_STARTEDという行を変更します (第 4.12.1 項を参照してください)。この操作は OpenVMS クラスタの各ノードで実行しなければなりません。

---

## 4.12 COM for OpenVMS の起動 (COM for OpenVMS Service Control Manager)

COM for OpenVMS を起動するには、次のコマンドを使用します。

```
$ @SYS$STARTUP:DCOM$STARTUP
```

また、DCOM\$SETUPを実行し、オプション 4 を選択することもできます (第 5.2 節を参照してください)。

COM for OpenVMS Service Control Manager は、初期化中、実行中、起動されていない状態のいずれかになります。COM for OpenVMS Service Control Manager の状態に応じて、次のいずれかのメッセージが表示されます。

- COM for OpenVMS Service Control Manager がこのノードで実行されている場合は、次のように、プロセスがすでにアクティブであることが報告されます。

```
DCOM Service Control Manager daemon (DCOM$RPCSS) is active [pid=xxxxxxx]
```

COM for OpenVMS Service Control Manager がこのノードで初期化中である場合は、次のように、プロセスがすでにアクティブであることが報告されます。

```
DCOM Service Control Manager daemon (DCOM$STARTUP-**) is active [pid=xxxxxxx]
```

- COM for OpenVMS Service Control Manager がこのノードで起動されていない場合は、次のように COM for OpenVMS が起動されます。

```
The OpenVMS Registry server is already started on this node.

*** DCOM system startup procedure ***

Starting DCOM Service Control Manager daemon ( "DCOM$STARTUP-***" ) . . .
After initialization, the daemon will use process name "DCOM$RPCSS" . . .

%RUN-S-PROC_ID, identification of created process is xxxxxxxx

*** DCOM startup successful ***

*** DCOM Startup Procedure Complete ***
```

#### 4.12.1 リブート後の COM for OpenVMS の自動起動

COM for OpenVMS の起動を制御するには、SYS\$MANAGER:SYLOGICALS.COM コマンド・ファイルを変更してください。

OpenVMS には変更された SYLOGICALS.TEMPLATE ファイルが含まれていて、このファイルには COM for OpenVMS と関連コンポーネント用の新しいスタートアップ・コマンドが登録されています。このテンプレート・ファイルの“Coordinated Startup”セクションを確認し、既存のスタートアップ・ファイルに適切な情報を追加してください。

システムのブート時に COM for OpenVMS が自動起動するように設定するには、次の行を SYLOGICALS.COM ファイルにコピーし、この行のコメントの設定を解除し、TRUE に設定されているかどうか確認します。

```
$ DEFINE DCOM$TO_BE_STARTED TRUE
```

システムのブート時に自動起動するように COM for OpenVMS を設定していない場合は、「DCOM\$SETUP OpenVMS COM Tools」メニューのオプション 4 を使用して、COM for OpenVMS を起動できます (第 5.2 節を参照)。

---

#### 4.13 COM for OpenVMS (COM for OpenVMS Service Control Manager のシャットダウン)

COM for OpenVMS をシャットダウンするには、次のコマンドを使用します。

```
$ @SYS$STARTUP:DCOM$SHUTDOWN
```

また、DCOM\$SETUP を実行し、オプション 5 を選択することもできます (第 5.2 節を参照してください)。

COM for OpenVMS Service Control Manager は停止、実行中、初期化中のいずれかの状態になります。COM for OpenVMS Service Control Manager の状態に応じて、次のいずれかのメッセージが表示されます。

- COM for OpenVMS Service Control Manager がこのノードで停止されている場合は、シャットダウンするものがないことが報告されます。

```
*** DCOM system shutdown procedure ***
```

```
There is no active DCOM$RPCSS daemon on this system.
```

```
*** DCOM Shutdown Procedure Complete ***
```

- COM for OpenVMS Service Control Manager がこのノードで実行中の場合は、プロセスが次のようにシャットダウンされます。

```
*** DCOM system shutdown procedure ***
```

```
***** Warning *****
```

```
*** Stopping the DCOM Service Control Manager daemon (DCOM$RPCSS)
```

```
*** Active DCOM applications will no longer be operational.
```

```
Do you want to proceed with this operation (YES/NO/?) [N]?
```

シャットダウン・プロシージャを続行するには、Yと入力します。

---

**注意**

---

この確認手順を行わないようにする方法については、第 4.13.1 項を参照してください。

---

次のメッセージが表示されます。

```
Terminating DCOM Service Control Manager daemon (DCOM$RPCSS) . . .
```

```
*** DCOM shutdown successful ***
```

```
*** DCOM Shutdown Procedure Complete ***
```

- COM for OpenVMS Service Control Manager がこのノードで初期化中の場合は、次のようにプロセスがシャットダウンされます。

```
*** DCOM system shutdown procedure ***
```

```
***** Warning *****
```

```
*** Stopping the DCOM Service Control Manager daemon (DCOM$RPCSS)
```

```
*** Active DCOM applications will no longer be operational.
```

```
Do you want to proceed with this operation (YES/NO/?) [N]?
```

シャットダウン・プロシージャを続行するには、Yと入力します。次のメッセージが表示されます。

```
Terminating DCOM Service Control Manager daemon (DCOM$STARTUP-**) . . .
```

```
*** DCOM shutdown successful ***
```

```
*** DCOM Shutdown Procedure Complete ***
```

## 4.13.1 DCOM\$SHUTDOWN の確認要求を行わない設定

NOCONFIRM パラメータを指定すると、DCOM\$SHUTDOWN コマンドの確認要求が表示されないようにすることができます。次のコマンドを使用します。

```
$ @SYS$STARTUP:DCOM$SHUTDOWN NOCONFIRM
```

次のシャットダウン・メッセージが表示されますが、シャットダウンを続行するかどうかの確認は行われません。

```
*** DCOM system shutdown procedure ***
Terminating DCOM Service Control Manager daemon (DCOM$RPCSS) . . .
*** DCOM shutdown successful ***
*** DCOM Shutdown Procedure Complete ***
```



---

## アプリケーションの開発と運用のための COM for OpenVMS ユーティリティ

この章では、COM アプリケーションを開発し、運用するために、OpenVMS システム (および必要に応じて Windows NT システム) を構成する方法について説明します。次の COM for OpenVMS ユーティリティについて説明します。

- DCOM\$SETUP ユーティリティ。このユーティリティは、システム管理者が COM for OpenVMS システム環境を構成するのに役立ちます。
- DCOM\$CNFG ユーティリティ。このユーティリティは、アプリケーション開発者が COM アプリケーションを構成し、確認するのに役立ちます。
- DCOM\$REGSVR32 ユーティリティ。このユーティリティを使用すると、アプリケーション開発者はプロセス内サーバ・アプリケーションの登録と登録の解除を行うことができます。

この章では、協調動作するように OpenVMS システムと Windows NT システムを構成する方法についても説明します。

---

### 操作を開始する前に

---

OpenVMS システムで COM for OpenVMS の構成を行う前に、必要なコンポーネントをインストールおよび構成し、COM for OpenVMS をインストールしておかなければなりません。これらの手順については、第 4 章を参照してください。

---

---

## 5.1 DCOM\$SETUP ユーティリティ

DCOM\$SETUP は、システム管理者が COM for OpenVMS システム環境を構成するのに役立つツール群です。

DCOM\$SETUP の表記法と要件

- Yes/No という質問に対しては、次のいずれかを入力できます。
  - YES または NO
  - Y または N
  - (デフォルト値を使用する場合)
- 一部の DCOM\$SETUP オプションでは、システム管理者特権と OpenVMS Registry へのアクセス権が必要です。

---

## 5.2 DCOM\$SETUP の実行

DCOM\$SETUPを実行するには、OpenVMS システム・プロンプトに対して@SYS\$STARTUP:DCOM\$SETUPと入力します。

「DCOM\$SETUP OpenVMS COM Tools」メニューが表示されます。

図 5-1 「DCOM\$SETUP OpenVMS COM Tools」メニュー

```
-----  
                                OpenVMS COM Tools  
                                1) DCOMCNFG, COM Configuration Properties  
                                2) GUIDGEN, Globally Unique Identifier Generator  
                                3) Populate the Registry database for COM  
                                4) Start the COM server  
                                5) Stop the COM server  
                                6) Register a COM application  
                                7) Create the DCOM$GUEST account and directory  
                                8) Configure the DCOM$RPCSS accounts  
  
                                H) Help  
                                E) Exit  
  
Please enter your choice:  
-----
```

オプションを選択するには、オプション番号を入力します。オプションは次のとおりです。

- 1) DCOMCNFG, COM Configuration Properties

COM for OpenVMS アプリケーションのプロパティの設定と表示に使用します。詳細については、第 5.3 節を参照してください。

- 2) GUIDGEN, Globally Unique Identifier Generator

GUID (globally unique identifiers: グローバル一意識別子) をさまざまな形式 (たとえば OpenVMS Registry または Windows NT Registry 形式) で生成します。GUID は各アプリケーションに固有の識別子を割り当てます。

DCOM\$SETUPのこのバージョンでは、GUID は OpenVMS Registry および Windows NT Registry 形式でのみ生成されます。他の形式については、第 6.1 節を参照してください。

- 3) Populate the Registry database for COM

OpenVMS Registry データベースをセットアップします。COM for OpenVMS では、特定のキーと値を OpenVMS Registry データベースに追加する必要があります。この操作を行うには、OpenVMS Registry への書き込みアクセス権と、Windows NT の Administrator 権限の両方が必要です。

- 4) Start the COM server

COM for OpenVMS Server Control Manager サーバ (DCOM\$RPCSS) を起動します。DCOM\$SETUPはSYS\$STARTUP:DCOM\$STARTUP プロシージャを呼び出して、サーバを起動します。詳細については、第 5.2.2 項を参照してください。

- 5) Stop the COM server

COM for OpenVMS Service Control Manager サーバ (DCOM\$RPCSS) をシャットダウンします。DCOM\$SETUPはSYS\$STARTUP:DCOM\$SHUTDOWN プロシージャを呼び出して、サーバを停止します。詳細については、第 5.2.2 項を参照してください。

- 6) Register a COM application

COM for OpenVMS サーバ・アプリケーションを登録します。次の種類のサーバを登録できます。

— プロセス内サーバ

プロセス内サーバを登録する場合は、サーバの場所が質問されます。

— ローカル・サーバまたはプロセス外サーバ

ローカル・サーバまたはプロセス外サーバを登録する場合は、次の情報が求められます。

+ フル・パス情報 (サーバの場所)

これは必須の値です。次の構文を使用してください。

*device:[directory]file-name.ext*

+ アプリケーション・タイトル

これは省略可能な値です。タイトルを指定しないと、デフォルト・タイトルが使用されます。

+ GUID

これは必須の値です。サーバに GUID が割り当てられていない場合は、システムが自動的に生成します。GUID と LocalServer32 の詳細については、第 6.5.1 項を参照してください。

登録が完了すると、次のファイルが生成されます。

1. Windows NT Registry ファイル (*server-name.REG\_NT*)。このファイルは、Windows NT システム上のアプリケーションを登録するために使用します。
2. OpenVMS コマンド・プロシージャ (*server-name.REG\_VMS*)。OpenVMS システム上のサーバを登録するために使用します。

これらのファイルを他のシステムで使用する場合は、サーバの現在の場所を示すように path 文を変更しなければなりません。詳細については、第 5.2.3 項を参照してください。

- 7) Create the DCOM\$GUEST account and directory  
NTLM 認証なしで COM for OpenVMS を使用するには、DCOM\$GUEST アカウントをあらかじめ作成しておかなければなりません。
- 8) Configure the DCOM\$RPCSS accounts  
DCOM\$RPCSS Advanced Server for OpenVMS ユーザ・アカウントと SYSUAF アカウントを構成し、作成します。COM for OpenVMS Service Control Manager (DCOM\$RPCSS) では、認証のためにこれらのアカウントが必要です。詳細については、第 5.2.1 項を参照してください。
- H) Help  
各メニュー・オプションのヘルプを表示します。
- E) Exit  
メニューを終了します。

### 5.2.1 DCOM\$RPCSS アカウントの作成と構成

これらの機能を表示するには、「DCOM\$RPCSS OpenVMS COM Tools」メニューからオプション 8 を選択します。次の情報が表示されます。

```
-----  
Configure the COM for OpenVMS Service Control Manager (DCOM$RPCSS) accounts
```

- 1) Create the DCOM\$RPCSS account in both the SYSUAF database and the Advanced Server for OpenVMS SAM database. The password you specify for the new DCOM\$RPCSS user is stored in a protected file.
  - 2) Update the DCOM\$RPCSS user password in the COM for OpenVMS Service Control Manager password file.
- E) Exit

```
Please enter your choice:  
-----
```

次のいずれかを入力します。

- 1) Create the DCOM\$RPCSS account . . .

このオプションを選択すると、DCOM\$RPCSS アカウントが SYSUAF データベースと Advanced Server for OpenVMS SAM データベースの両方に作成されます。

DCOM\$RPCSS ユーザ用に指定したパスワードは、保護されるファイルに格納され、COM for OpenVMS Service Control Manager はそのファイルを使用して、NTLM ネットワークにログインし、Windows NT アイデンティティを取得します。

---

#### 注意

このアカウントは、有効期限の切れないパスワードを使用して Advanced Server for OpenVMS データベースに作成されます。この動作を変更するには、(つまり、Advanced Server for OpenVMS ユーザ・ポリシーに従ってパ

パスワードの有効期限が切れるように、アカウントを変更するには、次の操作を行います。

1. Advanced Server for OpenVMS ADMIN ユーティリティを実行します。
2. Administrator アカウントにログインします。
3. 次の ADMIN コマンドを実行します。

```
ADMIN> MODIFY USER DCOM$RPCSS/FLAG=NODISPWDEXP
```

Advanced Server for OpenVMS のユーザ・ポリシーでパスワードの有効期限 (maximum password age) を判断するには、次の ADMIN コマンドを入力します。

```
ADMIN> SHOW ACCOUNT POLICY
```

DCOM\$RPCSS アカウントの Advanced Server for OpenVMS パスワードを変更する場合は、COM for OpenVMS Service Control Manager パスワード・ファイルでパスワードを更新しなければなりません (オプション 2 の「Update the DCOM\$RPCSS user password」を参照)。

---

次の操作を行います。

1. 1 と入力します。

次のメッセージが表示されます。

```
To create a new account, you must be logged on to an existing  
Advanced Server for OpenVMS account that is capable of adding  
new users.
```

```
Enter Y[ES] to log on to this account:
```

このアカウントを作成するには、PATHWORKS 管理者グループに属していなければなりません。

2. Y と入力します。

ログオンするように求められます。入力したパスワードは表示されません。

```
Enter username: JOSEPHM  
Password:  
Confirm password:
```

新しいパスワードの入力が求められます。その後、確認のために同じパスワードをもう一度入力するように求められます。入力したパスワードは表示されません。

```
Enter the new DCOM$RPCSS password.
```

```
Enter password:  
Confirm password:
```

## アプリケーションの開発と運用のための COM for OpenVMS ユーティリティ 5.2 DCOM\$SETUP の実行

このパスワードは、SYSUAF アカウント (DCOM\$RPCSS) と PATHWORKS ユーザ・アカウント (DCOM\$RPCSS) の両方で使用されます。このパスワードは COM for OpenVMS Service Control Manager パスワード・ファイルに格納されます。

次のアカウント作成情報が表示されます。

```
%PWRK-S-USERADD, user "DCOM$RPCSS" added to domain "DCOM1_DOMAIN"
Username: DCOM$RPCSS                Owner: COM
Account:                             UIC: [37776,1] ([DCOM$RPCSS])
CLI: DCL                             Tables: DCLTABLES
Default: SYS$SYSDEVICE:[DCOM$RPCSS]
LGICMD:
Flags: ExtAuth
Primary days: Mon Tue Wed Thu Fri
Secondary days:                      Sat Sun
No access restrictions
Expiration: (none) Pwdminimum: 6 Login Fails: 0
Pwdlifetime: (none) Pwdchange: (pre-expired)
Last Login: (none) (interactive), (none)
(non-interactive)
Maxjobs: 0 Fillm: 100 Byt1m: 64000
Maxacctjobs: 0 Shrfillm: 0 Pbyt1m: 0
Maxdetach: 0 BI01m: 150 JTquota: 4096
Prclm: 8 DI01m: 150 WSdef: 1024
Prio: 4 AST1m: 250 WSquo: 4000
Queprio: 4 TQE1m: 10 WSextent: 8000
CPU: (none) Enqlm: 2000 Pgflquo: 130000
Authorized Privileges:
NETMBX TMPMBX
Default Privileges:
NETMBX TMPMBX
%PWRK-S-HOSTMAPADD, user "DCOM$RPCSS" mapped to host user "DCOM$RPCSS"
Press RETURN to continue:
```

- 2) Update the DCOM\$RPCSS user password . . .

**Advanced Server for OpenVMS SAM データベースで DCOM\$RPCSS ユーザ・パスワードを変更する場合は、COM for OpenVMS Service Control Manager パスワード・ファイルのパスワードも更新しなければなりません。**

次の操作を行います。

1. 2と入力します。

次のメッセージが表示されます。

```
Enter the new DCOM$RPCSS password.
```

```
Enter password:
```

```
Confirm password:
```

2. 新しいパスワードを入力し、確認のために同じパスワードをもう一度入力します。

- E) Exit  
メニューを終了します。

## 5.2.2 COM サーバ (DCOM\$RPCSS Process) の起動と停止

COM for OpenVMS では、COM サーバ・プロセス (DCOM\$RPCSS) が常に動作していることが必要です。OpenVMS 上の DCOM\$RPCSS プロセスは、RPCSS プロセスが Microsoft Windows NT 上で提供する COM 実行時環境の機能と同じ機能を提供します。次の機能が提供されます。

- システム上で動作するサーバ・オブジェクトの一覧を作成および管理します。
- レジストリに定義されているアプリケーションのキャッシュを作成および管理します。このキャッシュは COM のパフォーマンスを向上します。
- 現在動作していないサーバ・オブジェクトへの接続をクライアントが要求したときに、サーバを独立プロセスとして起動します。
- リモートの Windows NT システム上の RPCSS プロセス、または OpenVMS システム上の DCOM\$RPCSS プロセスと通信して、リモート・サーバ・オブジェクトの検索や起動を行います。

DCOM\$RPCSS を起動するには、DCOM\$SETUP のオプション 4 (“Start”) を使用するか (第 5.2 節を参照)、SYS\$STARTUP:DCOM\$STARTUP から COM for OpenVMS スタートアップ・プロシージャを直接呼び出します。

システムで DCOM\$RPCSS を停止するには、DCOM\$SETUP のオプション 5 (“Stop”) を使用するか (第 5.2 節を参照)、SYS\$STARTUP:DCOM\$SHUTDOWN から COM for OpenVMS シャットダウン・プロシージャを直接呼び出します。

## 5.2.3 アプリケーションの登録

次の例では、COM for OpenVMS キットに添付されている COM for OpenVMS “Simple” アプリケーションを登録する方法について説明します。Windows NT システムでアプリケーションを使用できる場合は、作成される Windows NT ファイルを使用して、サーバを Windows NT システムに登録することができます。

Windows NT システムで “Simple” アプリケーションを作成する場合は、DCOM\$EXAMPLES:[SIMPLE] の README-SIMPLE.TXT ファイルの指示に従ってください。

---

### 注意

---

アプリケーションを登録するには、その前にアプリケーションを作成してコンパイルしておかなければなりません。詳細については、COM for OpenVMS キットに添付されている DCOM\$EXAMPLES:[SIMPLE] の例を参照してください。

---

## アプリケーションの開発と運用のための COM for OpenVMS ユーティリティ 5.2 DCOM\$SETUP の実行

次の操作を行います。

1. DCOM\$SETUPメニューから6またはREGISTERと入力します。
2. 質問に対して次のように応答します。

---

### 注意

---

“Simple”アプリケーションにはあらかじめ、CLSID が割り当てられています。

---

#### 例 5-1 OpenVMS での“Simple”アプリケーションの登録例

```
Enter server type (1. In-Proc 2. Out-Proc): 2 
Enter Local Path (device:[directory]filename.ext): USER:[SMITH]SSERVER.EXE

Enter Application Name (<RETURN> to assign default): COM Simple Server

Does the server have a CLSID {GUID} (Yes/No) [N]: Y 
Enter the CLSID (i.e. {xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}):
{5e9ddec7-5767-11cf-beab-00aa006c3606} 
Verify Application Information:
    Application Name: COM SIMPLE SERVER
    Local Path: USER:[SMITH]SSERVER.EXE
    Application ID: {5E9DDEC7-5767-11CF-BEAB-00AA006C3606}
Is the information correct (Yes/No) [Y]: 
Register application (Yes/No)? [Y]: 
SETUP-I-NEWFILES, The following files have been created:
    USER:[SMITH]SSERVER.REG_NT
    USER:[SMITH]SSERVER.REG_VMS
SETUP-I-SRVIN, Server has been registered
Press RETURN to continue: 
```

Windows NT システムで“Simple”アプリケーションを登録するには、次の操作を行います。

1. DCOM\$EXAMPLES:[SIMPLE]ディレクトリのすべてのファイルを Windows NT システムにコピーします。
2. SSERVER.REG\_NTをSSERVER.REGという名前に変更します。
3. ローカル・サーバ・パスを指すようにファイルを変更します。  
たとえば、DEVICE:\SSERVERをC:\SSERVERに変更します。
4. Install.batプログラムを実行して、必要なキーを Windows NT レジストリに追加します。

例 5-2 はSSERVER.REG\_NTの内容を示しています。

例 5-2 SSERVER.REG\_NT の内容

```
REGEDIT
HKEY_CLASSES_ROOT\CLSID\{5E9DDEC7-5767-11CF-BEAB-00AA006C3606}\ = DCOM
server application SSERVER
HKEY_CLASSES_ROOT\CLSID\{5E9DDEC7-5767-11CF-BEAB-00AA006C3606}\LaunchPermiss
ion = Y
HKEY_CLASSES_ROOT\CLSID\{5E9DDEC7-5767-11CF-BEAB-00AA006C3606}\LocalServer32
= DEVICE:\SSERVER
```

OpenVMS システムで“Simple”アプリケーションを再登録するには、システム・プロンプトに対して次のコマンドを入力します。

```
$ @SSERVER.REG_VMS
```

例 5-3 はSSERVER.REG\_VMSコマンド・プロシージャの内容を示しています。

例 5-3 SSERVER.REG\_VMS の内容

```
$ Set noon
$ regcp := $regcp
$ crekey := $regcp create key
$ creval := $regcp create value
$ modval := $regcp modify value
$ lisval := $regcp list value
$ crekey HKEY_CLASSES_ROOT\CLSID\{5E9DDEC7-5767-11CF-BEAB-00AA006C3606}
$ creval HKEY_CLASSES_ROOT\CLSID\{5E9DDEC7-5767-11CF-BEAB-00AA006C3606} -
    /data="DCOM server application SSERVER" /type=sz
$ creval
HKEY_CLASSES_ROOT\CLSID\{5E9DDEC7-5767-11CF-BEAB-00AA006C3606}/name="AppID"
-
    /data="{5E9DDEC7-5767-11CF-BEAB-00AA006C3606}" /type=sz
$ crekey
HKEY_CLASSES_ROOT\CLSID\{5E9DDEC7-5767-11CF-BEAB-00AA006C3606}\LaunchPermiss
ion
$ creval
HKEY_CLASSES_ROOT\CLSID\{5E9DDEC7-5767-11CF-BEAB-00AA006C3606}\LaunchPermiss
ion -
    /data="Y" /type=sz
$ crekey
HKEY_CLASSES_ROOT\CLSID\{5E9DDEC7-5767-11CF-BEAB-00AA006C3606}\LocalServer32
$ creval
HKEY_CLASSES_ROOT\CLSID\{5E9DDEC7-5767-11CF-BEAB-00AA006C3606}\LocalServer32
-
    /data="USER::[SMITH]SSERVER.EXE" /type=sz
$
```

---

## 5.3 DCOM\$CNFG の実行

DCOM\$CNFGは、COM 開発者が OpenVMS で COM for OpenVMS アプリケーションを構成し、管理するのに役立つユーティリティです。COM for OpenVMS アプリケーションに関する情報を問い合わせたり、プロパティを操作するには、DCOM\$CNFGユーティリティを使用します。

DCOM\$CNFGユーティリティを使用するには、DCOM\$SETUPメニューからオプション 1 を選択します。

---

### 注意

---

DCOM\$CNFGユーティリティを実行する前に、次のことを確認してください。

- アプリケーションのプロパティを表示するには、OpenVMS Registry への読み込みアクセス権が必要であり、アプリケーションのプロパティを変更するには、書き込みアクセス権が必要です。
  - ACME サーバが現在のシステムで動作していることを確認してください。アプリケーション・セキュリティのプロパティを表示または変更するには、ACME サーバが動作していなければなりません。詳細については、表 4-1 を参照してください。
  - アプリケーションのアイデンティティを変更する場合は、その前に Windows NT セキュリティ資格情報を取得してください。詳細については、第 12.2 節を参照してください。
- 

「DCOM\$CNFG Main」メニューが表示されます。

図 5-2 「DCOM\$CNFG Main」メニュー

```
-----  
DCOM$CNFG Main  
  
1 - Applications List  
2 - System-wide Default Properties  
3 - System-wide Default Security  
  
(E to Exit)  
(H for Help)  
  
Enter <CTRL-Z> or 'E' to return to the previous menu at any time  
  
Please enter your choice:  
-----
```

オプションは次のとおりです。

- 1 - Applications List  
このマシンに登録されているすべてのアプリケーションを一覧表示します。このオプションの詳細については、第 5.3.1 項を参照してください。

- 2 - System-wide Default Properties  
システム単位のマシン・プロパティを設定できます。このオプションの詳細については、第 5.3.5 項を参照してください。
- 3 - System-wide Default Security  
システム単位のセキュリティ・パラメータを設定できます。このオプションの詳細については、第 5.3.6 項を参照してください。

### 5.3.1 DCOM\$CNFG の「Application List」サブメニュー

このサブメニューを「DCOM\$CNFG Main」メニューから表示するには、オプション 1 を選択します。

「Applications List」サブメニューが表示されます。

図 5-3 「Applications List」サブメニュー

```
-----  
                          Applications List  
  
Index   Name  
  1     Inside COM, Chapter 11 Example  
  2     application 2  
  3     application 3  
  .     ...  
  .     ...  
  .     ...  
  
(E to Exit to previous menu)  
(H for Help)  
  
Please enter Index number to select an Application:  
-----
```

番号を入力してアプリケーションを選択します。選択したアプリケーションのプロパティを表示または設定できます。

このオプションを選択すると、「Application Properties」サブメニューが表示されます。

#### 注意

アプリケーションのプロパティ（「Location」、「Security」、「Identity」）（図 5-4 を参照）は、OpenVMS Registry で各アプリケーションに関連付けられている特殊キーに格納されます。DCOM\$CNFG ユーティリティを使用してこの特殊キーを作成するまで、アプリケーションのプロパティを変更することはできません。DCOM\$CNFG ユーティリティは、新たに登録されたアプリケーションを検出したときに、この特殊キーを作成します。しかし、DCOM\$CNFG ユーティリティがこのキーを作成するのは、Administrator グループのメンバであるアカウントの Windows NT セキュリティ資格情報を取得したユーザが DCOM\$CNFG ユーティリティを実行した場合だけです。

## アプリケーションの開発と運用のための COM for OpenVMS ユーティリティ 5.3 DCOM\$CNFG の実行

Windows NT 資格情報の取得の詳細については、第 12.2 節を参照してください。

アプリケーションのプロパティを管理するには、次の操作を行います。

1. アプリケーションを登録します。
  2. 次のいずれかの操作を行います。
    - Administrator グループのメンバであるアカウントの Windows NT セキュリティ資格情報を取得し、DCOM\$CNFG を実行します。
    - 適切な資格情報を保有するシステム管理者に DCOM\$CNFG を実行してもらいます。
  3. 自分のアカウントから DCOM\$CNFG を実行して、プロパティを管理します。
- 

図 5-4 「Application Properties」サブメニュー

```
-----  
Application Properties  
-----  
General Properties of this DCOM Application  
Application name: Inside COM, Chapter 11 Example  
Application id: {0C092C2C-882C-11CF-A6BB-0080C7B2D682}  
Application type: local server  
Local path: DISK1:[SMITH.DISPATCH_SAMPLE1]CMPNT.EXE  
Type Library: {D3011EE1-B997-11CF-A6BB-0080C7B2D682}  
version: 1.0 DISK1:[SMITH.DISPATCH_SAMPLE1]Server.tlb  
  
1 - Location Machine to run application  
2 - Security Security permissions for application  
3 - Identity User account to use to run application  
  
(E to Exit to previous menu)  
(H for Help)  
  
Please enter Application Property you wish to change:  
-----
```

タイプ・ライブラリ・ファイルが見つからない場合や、タイプ・ライブラリにアクセスできない場合は、タイプ・ライブラリ・ファイル名の横にエラー・メッセージが表示されます。

オプションは次のとおりです。

- 1 - Location: Machine to run application  
このオプションを選択すると、COM アプリケーションが実行されるマシンを設定または変更できます。

「Application Location」サブメニューが表示されます。

図 5-5 「Application Location」サブメニュー

```
-----  
Application Location  
  
The following settings allow DCOM to locate the correct computer  
for this application. If more than one machine is selected then  
DCOM uses the first available one. Client applications may override  
these selections.  
  
Application name: Inside COM, Chapter 11 Example  
  
1 - Run application on this computer (Yes/No)  
   Current value: Yes  
  
2 - Run application on another computer  
   Current value: Currently Disabled  
  
(E to Exit to previous menu)  
(H for Help)  
  
Please enter your choice:  
-----
```

オプションは次のとおりです。

— 1 - Run application on this computer

アプリケーションをローカル・コンピュータで実行するかどうかを指定します。現在の値を変更するときは、オプションを選択します。

— 2 - Run application on another computer

このオプションを選択すると、アプリケーションは指定のコンピュータで実行されます。オプションを選択し、次のいずれかを入力します。

+ 現在の値を変更するときは有効なシステム名。

+ 値を無効にするときはハイフン (-)。ハイフンを入力すると、このフィールドは“Currently Disabled”に設定されます。

• 2 - Security: Security permissions for application

このオプションを選択すると、次のセキュリティ・プロパティを設定できます。

— Access permission: ユーザまたはグループがこのアプリケーションにアクセスすることを許可または拒否します。

— Launch permission: ユーザまたはグループがこのアプリケーションを実行することを許可または拒否します。

— Configuration permission: アプリケーションに関する情報が格納されている OpenVMS Registry の領域への読み込みアクセス権、書き込みアクセス権、特殊アクセス権を持つユーザまたはグループを識別します。

別の設定を指定しない限り、システム単位のデフォルト・セキュリティ値が使用されます。

「Application Security」サブメニューが表示されます。

図 5-6 「Application Security」サブメニュー

```
-----  
Application Security  
Application name: Inside COM, Chapter 11 Example  
Current Access permissions: Custom  
Current Launch permissions: Custom  
Current Configuration permissions: Default  
  
1 - Use Default Access permission  
2 - Edit Custom Access permission  
3 - Use Default Launch permission  
4 - Edit Custom Launch permission  
5 - Use Default Configuration permission  
6 - Edit Custom Configuration permission  
  
(E to Exit to previous menu)  
(H for Help)  
  
Please enter your choice:  
-----
```

オプションは次のとおりです。

- 1 - Use Default Access permission

デフォルトのアクセス許可の値を使用します。

- 2 - Edit Custom Access permission

「Registry Value Permissions」サブメニューを表示します。このサブメニューでは、このアプリケーションのアクセス許可の値を表示、追加、変更、削除することができます。これらのサブメニューについては、第 5.3.2 項を参照してください。

あらかじめ他の値を設定した場合を除き、ACL エディタはシステム単位のデフォルト値を使用して起動されます。

- 3 - Use Default Launch permission

システム単位のデフォルトの起動許可 (Launch permission) の値を使用します。

- 4 - Edit Custom Launch permission

「Registry Value Permissions」サブメニューを表示します。このサブメニューでは、このアプリケーションの起動許可の値を表示、追加、変更、削除することができます。これらのサブメニューについては、第 5.3.2 項を参照してください。

あらかじめ他の値を設定した場合を除き、ACL エディタはシステム単位のデフォルト値を使用して起動されます。

- 5 - Use Default Configuration permission

システム単位の構成許可 (Configuration permission) の値を使用します。

- 6 - Edit Custom Configuration permission

「Registry Key Permissions」サブメニューを表示します。このサブメニューでは、このアプリケーションの特殊アクセス・セキュリティ許可を表示、追加、変更、削除、構成することができます。これらのサブメニューについては、第 5.3.3 項を参照してください。

### 5.3.2 「Registry Value Permissions」サブメニュー

このサブメニューを表示するには、次の操作を行います。

1. DCOM\$CNFG メニューからオプション 1 を選択します。
2. 「Applications List」サブメニューから適切なアプリケーションを選択します。
3. 「Application Properties」サブメニューからオプション 2 を選択します。
4. 「Application Security」サブメニューからオプション 2 または 4 を選択します。

図 5-7 「Registry Value Permissions」サブメニュー

```
-----  
Registry Value Permissions  
  
Application name: Inside COM, Chapter 11 Example  
Registry Value: LaunchPermission  
Owner: Administrator  
  
Index   Name                               Type of Access  
  1     OPENVMS_DCOM\USER1                 Deny  
  2     BUILTIN\Administrators            Allow  
  3     Everyone                          Allow  
  4     NT AUTHORITY\SYSTEM               Allow  
  5     OPENVMS_DCOM\USER2                 Allow  
  
(Index Number to Delete or Modify Access)  
(A to Add to list)  
  
(E to Exit to previous menu)  
(H for Help)  
  
Please enter your choice:  
-----
```

オプションは次のとおりです。

- Index Number...

アクセス・タイプを変更または削除するには、対応するインデックス番号を入力します。「Edit Registry Value Permissions」サブメニューが表示されます。

図 5-8 を参照してください。

- A to Add to List

このオプションを選択すると、「Add Registry Value Permissions」サブメニューが表示されます。このサブメニューでは、新しいエントリを OpenVMS Registry の値のアクセス制御リストに追加できます。図 5-9 を参照してください。

図 5-8 「Edit Registry Value Permissions」サブメニュー

```
-----  
                          Edit Registry Value Permissions  
Application name: Inside COM, Chapter 11 Example  
Registry Value: AccessPermission  
Owner: Administrator  
  
Name: OPENVMS_DCOM\USER1  
Type of Access: Deny  
  
1 - Delete entry from list  
2 - Change Access  
  
(E to Exit to previous menu)  
(H for Help)  
  
Please enter your choice:  
-----
```

オプションは次のとおりです。

- 1 - Delete entry from list

アクセス制御リストからエントリを削除します。すべてのエントリを削除すると、選択した値へのアクセス許可と起動許可がすべての人に対して拒否されず。

- 2 - Change Access

アクセス・タイプを「Allow to Deny」と「Deny to Allow」の間で切り換えま

図 5-9 「Add Registry Value Permissions」サブメニュー

```
-----  
Add Registry Value Permissions  
  
Application name: Inside COM, Chapter 11 Example  
Registry Value: LaunchPermission  
Owner: ROLLO  
  
1 - Add Specific User or Group  
2 - Add Everyone  
3 - Add NT AUTHORITY\System  
4 - Add BUILTIN\Administrators  
  
(E to Exit to previous menu)  
(H for Help)  
  
Please enter your choice:  
-----
```

オプションは次のとおりです。

— 1 - Add Specific User or Group

ユーザ/グループ名とアクセスの種類が要求されます。ユーザ名として *domain\username* を指定するか、アカウントがプライマリ・ドメインに存在する場合は、*username* を指定します。

— 2 - Add Everyone

アプリケーションに対するアクセス/起動 (Access/Launch) 許可を Everyone に対して許可または拒否します。

— 3 - Add NT AUTHORITY\System

アプリケーションに対するアクセス/起動 (Access/Launch) 許可を System に対して許可または拒否します。

— 4 - Add BUILTIN\Administrators

アプリケーションに対するアクセス/起動 (Access/Launch) 許可を Administrator に対して許可または拒否します。

ユーザが 2 つ以上のグループに属している場合は、Deny access の方が Allow access より優先します。

• 3 - Identity: User account to use to run application

このオプションを選択すると、指定したユーザ・アカウントのセキュリティ・コンテキストを使用して、アプリケーション・サーバを実行できます。

「Application Identity」サブメニューが表示されます。第 5.3.4 項を参照してください。

### 5.3.3 「Registry Key Permissions」サブメニュー

このサブメニューを表示するには、次の操作を行います。

1. DCOM\$CNFG メニューからオプション 1 を選択します。
2. 「Applications List」サブメニューから適切なアプリケーションを選択します。
3. 「Application Properties」サブメニューからオプション 2 を選択します。
4. 「Application Security」サブメニューからオプション 6 を選択します。

図 5-10 「Registry Key Permissions」サブメニュー

```
-----  
Registry Key Permissions  
  
Application name: Inside COM, Chapter 11 Example  
Registry Key: Inside COM, Chapter 11 Example  
Owner: Administrator  
  
Index   Name                               Type of Access  
-----  
1       BUILTIN\Administrators             Full Control  
2       NT AUTHORITY\SYSTEM                 Full Control  
3       CREATOR OWNER                       Full Control  
4       Everyone                           Special Access  
5       OPENVMS_DCOM\USER1                 Read  
  
(Index Number to Delete or Modify Access)  
(A to Add to list)  
  
(E to Exit to previous menu)  
(H for Help)  
  
Please enter your choice:  
-----
```

オプションは次のとおりです。

- Index Number...

アクセス・タイプを変更または削除するには、対応するインデックス番号を入力します。「Edit Registry Key Permissions」サブメニューが表示されます。図 5-11 を参照してください。

- A to Add to List

このオプションを選択すると、「Add Registry Key Permissions」サブメニューが表示されます。このサブメニューでは、新しいエントリを OpenVMS Registry キーのアクセス制御リストに追加できます。図 5-13 を参照してください。

図 5-11 「Edit Registry Key Permissions」サブメニュー

```
-----  
                        Edit Registry Key Permissions  
  
Application name: Inside COM, Chapter 11 Example  
Registry Key: Inside COM, Chapter 11 Example  
Owner: Administrator  
  
Name: BUILTIN\Administrators  
Type of Access: Full Control  
  
1 - Delete entry from list  
2 - Allow Full Control  
3 - Allow Read Access  
4 - Set/View Special Access  
  
(E to Exit to previous menu)  
(H for Help)  
  
Please enter your choice:  
-----
```

オプションは次のとおりです。

- 1 - Delete entry from list  
セキュリティ・アクセス許可リストからエントリを削除します。すべてのエントリを削除すると、誰もキーにアクセスできなくなり、所有者だけが許可を変更できます。
- 2 - Allow Full Control  
ユーザに対してキーへのアクセス、キーの変更、キーの所有権の取得を許可します。
- 3 - Allow Read Access  
ユーザに対してキーの表示を許可しますが、キーの保存や変更は禁止します。
- 4 - Set/View Special Access  
「Special Access Registry Key Permissions」サブメニューを表示します。このサブメニューでは、選択したユーザまたはグループに対してカスタマイズしたアクセス許可を設定できます。図 5-12 を参照してください。

図 5-12 「Special Access Registry Key Permissions」サブメニュー

```
-----  
Special Access Registry Key Permissions  
  
Application name: Inside COM, Chapter 11 Example  
Registry Key: Inside COM, Chapter 11 Example  
  
Name: Everyone  
  
Type of Access                                Current Value  
0 - Query Value                              Yes  
1 - Set Value                                 Yes  
2 - Create Subkey                             Yes  
3 - Enumerate Subkeys                         Yes  
4 - Notify                                    Yes  
5 - Create Link                               No  
6 - Delete                                    Yes  
7 - Write DACL                               No  
8 - Write Owner                              No  
9 - Read Control                             Yes  
  
(E to Exit to previous menu)  
(H for Help)  
  
Please enter your choice:  
-----
```

オプションは次のとおりです。

- 0 - Query Value  
ユーザに対して、キーの値の表示を許可します。
- 1 - Set Value  
ユーザがキーの 1 つ以上の値を設定することを許可します。
- 2 - Create Subkey  
ユーザがキーのサブキーを作成することを許可します。
- 3 - Enumerate Subkeys  
ユーザがキーのサブキーを指定することを許可します。
- 4 - Notify  
ユーザがキーからの通知イベントを監査することを許可します。
- 5 - Create Link  
ユーザがキーにシンボリック・リンクを作成することを許可します。
- 6 - Delete  
ユーザがキーを削除することを許可します。
- 7 - Write DACL  
ユーザがキーにアクセスして、随意 ACL をキーに書き込むことを許可します。

- 8 - Write Owner  
ユーザがキーにアクセスして、キーの所有権を取得することを許可します。
- 9 - Read Control  
ユーザがキーのセキュリティ情報にアクセスすることを許可します。

図 5-13 「Add Registry Key Permissions」サブメニュー

```
-----  
                          Add Registry Key Permissions  
  
Application name: Inside COM, Chapter 11 Example  
Registry Key: Inside COM, Chapter 11 Example  
Owner: Administrator  
  
1 - Add Specific User or Group  
2 - Add Everyone  
3 - Add NT AUTHORITY\System  
4 - Add BUILTIN\Administrators  
  
(E to Exit to previous menu)  
(H for Help)  
  
Please enter your choice:  
-----
```

オプションは次のとおりです。

- 1 - Add Specific User or Group  
ユーザ/グループ名とアクセスの種類が要求されます。ユーザ名として domain\username を指定するか、アカウントがプライマリ・ドメインに存在する場合は、username を指定します。
- 2 - Add Everyone  
Everyone に対して、アプリケーションへのフル・コントロールまたは読み込みアクセスを許可します。
- 3 - Add NT AUTHORITY\System  
System に対して、アプリケーションへのフル・コントロールまたは読み込みアクセスを許可します。
- 4 - Add BUILTIN\Administrators  
Administrator に対して、アプリケーションへのフル・コントロールまたは読み込みアクセスを許可します。

### 5.3.4 「Application Identity」サブメニュー

このサブメニューを表示するには、次の操作を行います。

1. DCOM\$CNFG メニューからオプション 1 を選択します。
2. 「Applications List」サブメニューから適切なアプリケーションを選択します。
3. 「Application Properties」サブメニューからオプション 3 を選択します。

「Application Identity」サブメニューが表示されます。

図 5-14 「Application Identity」サブメニュー

```
-----  
Application Identity  
Which user account do you want to use to run this application?  
Application name: Inside COM, Chapter 11 Example  
Current Identity: NTLM Account OPENVMS_DCOM\USER2  
  
1 - Launching User  
2 - NTLM Account  
3 - OpenVMS Username  
4 - OpenVMS DCOM Guest Account  
  
(E to Exit to previous menu)  
(H for Help)  
  
Please enter account you wish to use:  
-----
```

オプションは次のとおりです。

- 1 - Launching User

アプリケーションを起動したユーザのセキュリティ・コンテキストを使用して、アプリケーションを実行することを指定します。NTLM セキュリティを使用できる場合は、これがデフォルトです。

- 2 - NTLM Account

指定された NTLM アカウントのセキュリティ・コンテキストを使用して、アプリケーションを実行することを指定します。有効なユーザ/グループ名を指定すると、パスワードが求められます。入力したパスワードと、(NTA\$LOGONを通じて) ログオン時に使用したパスワードが一致するかどうか確認されます。パスワードが一致しない場合は、操作を続行して、この新しいパスワードを OpenVMS Registry に書き込むことができ、ログオン・パスワードと一致するパスワードを再入力することもできます。

#### 注意

新しいパスワードを入力した場合、新しいパスワードと他のパスワードの同期はとられません。パスワードの同期は手動でとらなければなりません。

パスワードが有効かどうか確認するには、IMPERSONATE 特権が必要です。

パスワードをデータベースに書き込むには、OpenVMS Registry に対してシステム単位の書き込みアクセス権 (SYSPRV または REG\$UPDATE) が必要です。

---

- 3 - OpenVMS Username

指定された OpenVMS アカウントのセキュリティ・コンテキストを使用して、アプリケーションが実行されることを指定します。このオプションを選択できるのは、認証されない COM for OpenVMS を使用している場合だけです。

- 4 - OpenVMS DCOM Guest Account

OpenVMS DCOM Guest アカウントのセキュリティ・コンテキストを使用して、アプリケーションが実行されることを指定します。このオプションを選択できるのは、認証されない COM for OpenVMS を使用している場合だけです。認証されない COM for OpenVMS を使用する場合は、このオプションがデフォルトです。

### 5.3.5 DCOM\$CNFG の「System-wide Default Properties」サブメニュー

このサブメニューを表示するには、「DCOM\$CNFG Main」メニューからオプション 2 を選択します。

「System-wide Default Properties」サブメニューが表示されます。

図 5-15 「System-wide Default Properties」サブメニュー

```
-----  
System-wide Default Properties  
1 - Enable Distributed COM on this computer (Yes/No)  
    Current value: Yes  
2 - Default Authentication Level  
3 - Default Impersonation Level  
  
(E to Exit to previous menu)  
(H for Help)  
Please enter your choice:  
-----
```

オプションは次のとおりです。

- 1 - Enable Distributed COM on this computer (Yes/No)

このコンピュータで COM を有効または無効にします。

- 2 - Default Authentication Level

アプリケーション間の通信でパケット・レベルのセキュリティを設定します。このシステム単位のデフォルトは、このコンピュータにインストールされているすべてのアプリケーションに適用されます。

図 5-16 「Default Authentication Level」サブメニュー

```
-----  
                        Default Authentication Level  
  
The Authentication Level specifies security at the packet level.  
  
Current value: Connect  
  
    1 - Default  
    2 - None  
    3 - Connect  
    4 - Call  
    5 - Packet  
    6 - Packet Integrity  
  
    (E to Exit to previous menu)  
    (H for Help)  
  
Please enter your choice:  
-----
```

適切な番号を入力して、認証レベルを選択します。インストールされている場合は、「Default Authentication Level」のシステム・デフォルトはConnectです。

- 3 - Default Impersonation Level

アプリケーションを誰が呼び出しているかをアプリケーションで判断できるかどうかと、アプリケーションがクライアントのアイデンティティを使用して操作を実行できるかどうかを指定します。

図 5-17 「Default Impersonation Level」サブメニュー

```
-----  
                        Default Impersonation Level  
  
The Impersonation Level specifies whether applications can determine  
who is calling them, and whether the application can perform  
operations using the client's identity.  
  
Current value: Identify  
  
    1 - Anonymous  
    2 - Identify  
    3 - Impersonate  
  
    (E to Exit to previous menu)  
    (H for Help)  
  
Please enter your choice:  
-----
```

適切な番号を入力して、偽装レベルを選択します。インストールされている場合は、「Default Impersonation Level」のシステム・デフォルトはIdentifyです。

### 5.3.6 「System-wide Default Security」サブメニュー

このサブメニューを表示するには、「DCOM\$CNFG Main」メニューからオプション 3 を選択します。

「System-wide Default Security」サブメニューが表示されます。

図 5-18 「System-wide Default Security」サブメニュー

```
-----  
System-wide Default Security  
1 - Access Permissions Default  
2 - Launch Permissions Default  
3 - Configuration Permissions Default  
  
(E to Exit to previous menu)  
(H for Help)  
  
Please enter your choice:  
-----
```

オプションは次のとおりです。

- 1 - Access Permissions Default:

「Registry Value Permissions」サブメニューを表示します。このサブメニューでは、すべてのアプリケーションのシステム単位のデフォルトに対して、アクセス許可の値の表示、追加、変更、削除を行うことができます。

- 2 - Launch Permissions Default:

「Registry Value Permissions」サブメニューを表示します。このサブメニューでは、すべてのアプリケーションのシステム単位のデフォルトに対して、起動許可の値の表示、追加、変更、削除を行うことができます。新しい設定を有効にするには、COM for OpenVMS Service Control Manager を再起動しなければなりません。

- 3 - Configuration Permissions Default

このリリースではまだ実装されていません。

システムを初めてインストールした場合、デフォルト設定により、Administrator と System アカウントにだけアプリケーションの起動許可とアクセス許可が与えられます。これらのデフォルト設定は変更しないでください。通常、個々のアプリケーションの起動許可やアクセス許可を変更することで、Everyone、さまざまな Groups、あるいは特定のユーザにアクセス許可を与えたり、拒否します。マシン単位でデフォルトのセキュリティ設定を調整すると、すべてのアプリケーションに影響するので、可能な限りこの手法を使用してください。

---

## 5.4 Windows NT ドメインを介した認証の構成

COM アプリケーションをあるドメインのシステムで実行し、別のドメインのシステムでアプリケーションの認証を受けることができます。

Windows NT ドメイン間で認証を構成するには、次の操作を行います。

1. ドメイン間の信頼関係を設定します。

詳細については、『*Advanced Server for OpenVMS Server Administrator's Guide*』の“Setting Up External Authentication by a Trusted Domain”を参照してください。

2. Advanced Server for OpenVMS ドメインで HostMapDomains パラメータを設定します。

詳細については、『*Advanced Server for OpenVMS Server Administrator's Guide*』の“Setting Up External Authentication by a Trusted Domain”を参照してください。

3. Windows NT ユーザ・アカウントとローカルの OpenVMS ユーザ・アカウントの間のアカウント・ホストマップ・エントリを設定します。

詳細については、第 12.3.2 項を参照してください。

例 5-4 は、HostMapDomains ファイルを設定する例を示しています。この例には、DOM\_JOE と DOM\_JANE の 2 つのドメインがあります。ドメイン DOM\_JANE は Advanced Server for OpenVMS を稼動しています。ドメイン DOM\_JOE は Windows NT ドメインです。例 5-4 のコマンドは、DOM\_JANE を DOM\_JOE に通知します。

### 例 5-4 例: HostMapDomains の設定

```
SYSJANE$ show sym regutl
  REGUTL == "$SYS$SYSTEM:PWRK$REGUTL.EXE"
SYSJANE$ regutl
REGUTL> SET PARAM /CREATE VMSSERVER HOSTMAPDOMAINS DOM_JOE
REGUTL> SHOW VALUE * HOSTMAPDOMAINS
Key: SYSTEM\CurrentControlSet\Services\AdvancedServer\UserServiceParameters
Value: HostmapDomains
Type: String
Current Data: DOM_JOE
```

## 5.5 プロセス内サーバの登録: DCOM\$REGSVR32 ユーティリティ

すべての COM コンポーネントは、プロセス外サーバとして実装されている場合も、プロセス内サーバとして実装されている場合も、使用する前に OpenVMS Registry に登録しなければなりません。

プロセス外サーバは実行可能プログラム (.EXEファイル) として実装され、通常はそのファイル内に格納されているコンポーネントの登録や登録の解除に使用されるコードを格納しています。プロセス外サーバは、実行可能ファイルを実行することができ、必要なレジストリ・キーを自動的に作成できるという点で、プロセス内サーバより有利です。

プロセス内サーバは通常、Windows NT ではダイナミック・リンク・ライブラリ (.DLLファイル) として実装され、OpenVMS では共用可能イメージとして実装されます。これらのファイルにも、ファイル内のコンポーネントの登録と登録の解除を自動的に行うためのコードが格納されています。しかし、これらのプロセス内サーバにはメイン・エントリ・ポイントが含まれていないため、実行可能イメージと同じ方法で実行することはできません。この結果、.DLLに格納されているコンポーネントを手動で登録したり、コマンド・プロシージャを作成して登録を行うようにしなければなりません。

Microsoft から提供されるREGSVR32ユーティリティを使用すると、DLL に含まれているコンポーネントを登録することができます。REGSVR32は次のコマンド・ライン引数を使用します。

- DLL 名
- コンポーネントの登録と登録の解除を切り換えるスイッチ

DLL のコンポーネントを登録する場合は、REGSVR32は指定された DLL から DllRegisterServerシンボルを検索し、検索したシンボルを呼び出します。DLL の登録を解除する場合は、REGSVR32はDllUnregisterServerを呼び出します。つまり、自動的に登録するすべてのプロセス内コンポーネントには、それぞれのエクスポート・ファイルにこれらの2つのエントリ・ポイントを格納しておかなければなりません。

OpenVMS システムで共用可能イメージに格納されているコンポーネントの登録を簡単に行うことができるように、Compaq は DCOM\$REGSVR32 ユーティリティを作成しました。DCOM\$REGSVR32 ユーティリティは、Microsoft REGSVR32ユーティリティと同じ処理を行います。登録するコンポーネントを格納した共用可能イメージには、それぞれのシンボル・ベクタにDllRegisterServerおよびDllUnregisterServerユニバーサル・シンボルを格納しなければなりません。DCOM\$REGSVR32 ユーティリティとREGSVR32ユーティリティはどちらも同じコマンド・ライン構文を使用します。

COM for OpenVMS をインストールするときに、DCOM\$REGSVR32.EXEファイルはSYS\$SYSTEMディレクトリに格納されます。

DCOM\$REGSVR32 ユーティリティを使用する前に、ユーティリティがフォーリン・コマンド・ラインを受け付けることができるようにシンボルを定義しなければなりません。次の例を参照してください。

```
$ regsvr32 ::= $DCOM$REGSVR32
```

また、次の方法で DCOM\$REGSVR32 ユーティリティをアクティブ化することもできます。

```
$ MCR DCOM$REGSVR32
```

いずれかの方法を使用してユーティリティをアクティブ化し、共用可能イメージに格納されているコンポーネントの登録または登録の解除を行います。

DCOM\$REGSVR32 のヘルプを表示するには、次のように入力します。

```
$ regsvr32 -?
```

表 5-1 は、DCOM\$REGSVR32 のコマンド・ライン・オプションを示しています。

表 5-1 DCOM\$REGSVR32 コマンド・ライン・オプション

スイッチ	使い方
-?, /?	ヘルプ・ファイル (この表) を表示する。
<i>shareable-image-name</i>	指定された共用可能イメージ名を登録する。
-u or /u <i>image-name</i>	指定された共用可能イメージ名の登録を解除する。

**注意**

DCOM\$REGSVR32 ユーティリティでは、共用可能イメージ名に完全なディレクトリ指定が必要です。

例 5-5 は、DCOM\$REGSVR32 ユーティリティを使用してプロセス内コンポーネント (共用可能イメージに含まれているコンポーネント) を登録する方法を示しています。

例 5-5 DCOM\$REGSVR32 ユーティリティを使用したコンポーネントの登録

```
$ regsvr32 USER$DISK:[SEYMOUR.DISPATCH_SAMPLE1]CMPNT$SHR.EXE
Class factory:          Create self.
DllRegisterServer:     Registering Server DLL
Creating key CLSID\{0C092C2C-882C-11CF-A6BB-0080C7B2D682}
Creating key CLSID\{0C092C2C-882C-11CF-A6BB-0080C7B2D682}\InProcServer32
Creating key CLSID\{0C092C2C-882C-11CF-A6BB-0080C7B2D682}\ProgID
Creating key
CLSID\{0C092C2C-882C-11CF-A6BB-0080C7B2D682}\VersionIndependentProgID
Creating key CLSID\{0C092C2C-882C-11CF-A6BB-0080C7B2D682}\TypeLib
Creating key InsideCOM.Chap11
Creating key InsideCOM.Chap11\CLSID
Creating key InsideCOM.Chap11\CurVer

Creating key InsideCOM.Chap11.1
Creating key InsideCOM.Chap11.1\CLSID
Class factory:          Destroy self.
```

例 5-6 は、DCOM\$REGSVR32 ユーティリティを使用してプロセス内コンポーネント (共用可能イメージに含まれているコンポーネント) の登録を解除する方法を示しています。

例 5-6 DCOM\$REGSVR32 ユーティリティを使用したコンポーネントの登録の解除

```
$ regsvr32 /u USER$DISK:[SEYMOUR.DISPATCH_SAMPLE1]CMPNT$SHR.EXE
Class factory:          Create self.
DllUnregisterServer:   Unregistering Server DLL
Deleting key InProcServer32
Deleting key ProgID
Deleting key VersionIndependentProgID
Deleting key TypeLib
Deleting key LocalServer32
Deleting key CLSID\{0C092C2C-882C-11CF-A6BB-0080C7B2D682}
Deleting key CLSID
Deleting key CurVer
Deleting key InsideCOM.Chap11
Deleting key CLSID
Deleting key InsideCOM.Chap11.1
Class factory:          Destroy self.
```



---

## COM for OpenVMS アプリケーションの開発

この章では、OpenVMS 用の COM アプリケーションの開発方法について説明します。

---

### 注意

---

この章で使用するサンプルの COM アプリケーションは、COM for OpenVMS キットの次のディレクトリにあります。

```
DCOM$EXAMPLES:[SAMPLE1]
DCOM$EXAMPLES:[SIMPLE]
DCOM$EXAMPLES:[DISPATCH_SAMPLE1]
```

SAMPLE1とDISPATCH\_SAMPLE1は、Microsoft Press から出版されている Dale Rogerson 著『*Inside COM*』という書籍からの抜粋です。この書籍は COM アプリケーションを開発する際に非常に役立つ参考文献です。

---

ここでは、COM for OpenVMS アプリケーションの作成方法について説明します。

---

### 注意

---

COM for OpenVMS アプリケーションを開発する場合、プロセスの仮想メモリが必要です。COM for OpenVMS アプリケーションを作成する前に、少なくとも 100,000 ページレットのページ・ファイル・クォータがあることを確認してください。これは DEC C++ コンパイラの要件です。

---

---

## 6.1 ステップ 1: 一意識別子を生成する

16 バイトのグローバル一意識別子 (GUID) を生成するには、DCOM\$GUIDGENユーティリティを使用します。このユーティリティは OpenVMS スタイルと UNIX スタイルの両方をサポートします。次の例を参照してください。

- OpenVMS スタイル

次のコマンドを入力します。

```
$ SET COMMAND DCOM$LIBRARY:DCOM$GUIDGEN.CLD
$ DCOM$GUIDGEN [/FORMAT=value] [/COUNT=value]
[/OUTPUT=value]
```

## COM for OpenVMS アプリケーションの開発

### 6.1 ステップ 1: 一意識別子を生成する

- UNIX スタイル

次のコマンドを入力します。

```
$ mcr dcom$guidgen [-cdghirs?] [-on]
```

次の表は GUID 形式オプションを示しています。

OpenVMS 修飾子 (value)	UNIX スイッチ	使い方
IDL	-i	GUID を IDL インタフェース・テンプレートに出力する。
STRUCT	-s	GUID を初期化された C 構造体として出力する。
IMPLEMENT_OLECREATE	-c	GUID を IMPLEMENT_OLECREATE(...) 形式で出力する。
DEFINE_GUID	-d	GUID を DEFINE_GUID(...) 形式で出力する。
GUID_STRUCT	-g	GUID を初期化された静的 const GUID 構造体として出力する。
REGISTRY_GUID	-r	GUID をレジストリ形式で出力する。

#### 注意

この表の最後の 4 つのオプションは、Windows NT の Guidgen ユーティリティの 4 つのオプションと同じです。

次の表は、DCOM\$GUIDGEN ユーティリティでサポートされる追加オプションを示しています。

OpenVMS 修飾子	UNIX スイッチ	使い方
/OUTPUT= <i>filename</i>	-o <i>filename</i>	出力を指定のファイルにリダイレクトする。
/COUNT= <i>number</i>	-n <i>number</i>	生成する GUID の数を指定する。
<i>not available</i>	-h, -?	コマンド・オプションの要約を表示する。

同じ GUID に対して複数の形式を指定できます。

## 6.2 ステップ 2: MIDL コンパイラを使用してアプリケーションを構築する

この後の節では、MIDL コンパイラを使用してアプリケーションを構築する方法について説明します。

## 6.2.1 MIDL コンパイラの実行

MIDL コンパイラは次のイメージで構成されています。

- SYS\$SYSTEM:DCOM\$MIDL.EXE

DCL コマンド・ラインに指定された引数 (パラメータ) を取り出す実行可能イメージ。

- SYS\$SHARE:DCOM\$MIDL\_SHR.EXE

DCOM\$MIDL.EXE の実際の作業を行う共用可能イメージ・ライブラリ。

MIDL を実行するには、最初に DCL シンボルを定義しなければなりません。次の例を参照してください。

```
$ midl ::= $dcom$midl
$ midl -?
$ midl -0icf -idcom$library: example.idl
```

midl -? コマンドは、有効なコマンド・ライン引数の一覧を表示します。これらの引数の一覧については、付録 A を参照してください。

## 6.2.2 DCOM\$RUNSHRLIB による MIDL コンパイラの実行

DCOM\$MIDL.EXE ユーティリティは、DCL フォーリン・コマンド・ライン・バッファから引数を取得します。DCL フォーリン・コマンドでは最大 255 文字を使用できません。

DCOM\$MIDL.EXE が受け付けることができる引数の数によっては、複雑な MIDL コマンド (たとえば引用符が必要な大文字と小文字が混在した引数を含むコマンドなど) を指定する場合は、この最大文字数を超えることがあります。

この問題を回避するには、SYS\$SYSTEM:DCOM\$RUNSHRLIB.EXE ユーティリティを使用します。次の操作を行います。

1. DCL コマンド DCOM\$RUNSHRLIB を定義します。

DCOM\$RUNSHRLIB.EXE を使用しなければならないプロセスは、最初に OpenVMS DCL Command Definition ユーティリティを使用して、DCL コマンド DCOM\$RUNSHRLIB を定義する必要があります。次の例を参照してください。

```
$ SET COMMAND DCOM$LIBRARY:DCOM$RUNSHRLIB.CLD
```

DCOM\$LIBRARY:DCOM\$RUNSHRLIB.CLD は DCOM\$RUNSHRLIB DCL コマンドを定義します。次の表はコマンドのパラメータを示しています。

COM for OpenVMS アプリケーションの開発  
 6.2 ステップ 2: MIDL コンパイラを使用してアプリケーションを構築する

引数	値	必須/省略可能
P1	共用可能イメージ・ライブラリの名前。論理名, SYS\$SHARE:内のイメージの名前, 完全なファイル指定のいずれかを指定できます。	必須
P2	argc/argv ベクタで C または C++ main()ルーチンとして呼び出されるルーチンの名前。	必須
P3	引用符で囲んだ修飾子の一覧。	省略可能

2. DCOM\$RUNSHRLIB.EXEを使用してコマンド・ラインを解析し, DCOM\$MIDL\_SHR共用可能イメージ・ライブラリ内のDCOM\$MIDL\_MAIN関数を呼び出すために, DCL シンボルmidlを定義します。次の例を参照してください。

```
$ midl ::= DCOM$RUNSHRLIB DCOM$MIDL_SHR DCOM$MIDL_MAIN
```

新しい DCL コマンド MIDL は, 1 つの引用符で囲まれた文字列内の複数のコマンド・ライン引数を受け付けます。コマンドが長すぎる場合は, 引用符で囲んだ複数の文字列を指定し, カンマで各文字列を区切ります。

たとえば, 次の複雑な MIDL コマンドは, そのままでは使用できません。

```
$ midl ::= $dcom$midl
$ midl -Zp8 -Oicf -Os -oldnames -char unsigned -
-error allocation -error bounds_check -error stub_data -
-ms_ext -c_ext -out [.OBJ] -
-I[INC] -I[PROJECT_WIDE_INC] -I[COMMON_INC] -IDCOM$LIBRARY: -
-DRMS_DB "-DOpenVMS_Definitions" "-DPermanentProcess" -
-header [.obj]example.h -client none -server none example.idl
%DCL-W-TKNOVF, command element is too long - shorten
```

次のように, DCOM\$RUNSHRLIBを使用すると, このコマンドを正しく指定できます。

```
$ set command dcom$library:dcom$runshrlib.cld
$ midl ::= DCOM$RUNSHRLIB DCOM$MIDL_SHR DCOM$MIDL_MAIN
$ midl "-Zp8 -Oicf -Os -oldnames -char unsigned",-
"-error allocation -error bounds_check -error stub_data",-
"-ms_ext -c_ext -out [.OBJ]",-
"-I[INC] -I[PROJECT_WIDE_INC] -I[COMMON_INC] -IDCOM$LIBRARY:",-
"-DRMS_DB -DOpenVMS_Definitions -DPermanentProcess",-
"-header [.obj]example.h -client none -server none example.idl"
```

### 6.2.3 必須 MIDL スイッチ

OpenVMS で MIDL を実行する場合, -Oicf MIDL コマンド・ラインスイッチを指定しなければなりません。

## 6.2.4 必須インクルード・ディレクトリ

MIDL コンポーネントは通常、UNKNWN.IDLをインポートします。これには IUnknownと IClassFactoryのコンポーネント定義が含まれています。UNKNWN.IDLと他の COM 関連の IDL およびヘッダ・ファイルは、DCOM\$LIBRARYにあります。コンポーネントの IDL ファイルを作成するには、次のスイッチを使用します。

```
-IDCOM$LIBRARY:
```

## 6.2.5 必須ヘッダ・ファイル

VMS\_DCOM.Hヘッダ・ファイルには、Bristol の Wind/U® Win32 環境で COM for OpenVMS アプリケーションを正しくコンパイルするためのマクロ定義が含まれています。このヘッダ・ファイルは、自分で作成するファイルのうち、COM API や Win32 API に依存するすべてのソース・ファイルとヘッダ・ファイルに含まなければなりません。MIDL で作成されるファイルは Win32 環境に依存するので、Compaq はすべての出力ファイルにVMS\_DCOM.Hをインクルードするように、OpenVMS 用の MIDL コンパイラを変更しました。

---

## 6.3 ステップ 3: COM アプリケーションをコンパイルする

ここでは、COM for OpenVMS アプリケーションのコンパイルの方法について説明します。

---

### 注意

---

OpenVMS Registry に対する Read アクセス権を必要とする COM アプリケーションを開発する場合は、REG\$LOOKUP 識別子があることを確認しなければなりません。コンパイル時または実行時に OpenVMS Registry に対して Read アクセス権と Write アクセス権を必要とするアプリケーションの場合、REG\$UPDATE 識別子または SYSPRV 特権が必要です。OpenVMS Registry に関連する特権の詳細については、第 7.5.1 項を参照してください。

---

### 6.3.1 必要なマクロ定義

VMS\_DCOM.H ファイルでは、Wind/U Win32 環境で使用される複数のマクロを定義しています。このヘッダ・ファイルを指定する include 文は、作成するすべてのソース・ファイル (コードまたはヘッダ・ファイル) のコメント以外の最初の行として指定しなければなりません。しかし、MIDL で生成されるファイルの場合、この規則が必ずしも保証されません。したがって、すべての C コマンドと CXX コマンドに次の/DEFINE 修飾子を必ず指定してください。

```
/DEFINE=(UNICODE=1,_WINDU_SOURCE=0X041000,_WIN32_DCOM)
```

## COM for OpenVMS アプリケーションの開発

### 6.3 ステップ 3: COM アプリケーションをコンパイルする

UNICODEマクロは、コードをコンパイルするときに、Win32 APIとデータ構造のワイド文字バージョンが有効に設定されているかどうか確認します(このマクロはVMS\_DCOM.Hにも定義されています)。このマクロを省略すると、Wind/U Win32環境でコンパイル・エラーが発生することがあります。

他の2つのマクロ定義はWind/U ヘッダ・ファイルで認識され、構造とCOM APIの正しい定義を行うのに必要です。

#### 6.3.2 必要なインクルード・ディレクトリ

COM for OpenVMS アプリケーションでは通常、DCOM\$LIBRARYに格納されているヘッダ・ファイルが必要です。

次の修飾子をCおよびCXX コマンド・ラインに指定してください。

```
/INCLUDE=DCOM$LIBRARY
```

/INCLUDE修飾子がコマンド・ラインにすでに指定されている場合は、DCOM\$LIBRARYをインクルードするようにコマンドを変更してください。

#### 6.3.3 必要なヘッダ・ファイル: VMS\_DCOM.H

VMS\_DCOM.Hヘッダ・ファイルでは、Wind/U ヘッダ・ファイルで 사용되는複数のマクロが定義されています。

このヘッダ・ファイルは、ソース・ファイル(ヘッダ・ファイルとインプリメンテーション・ファイルの両方)にコメント以外の最初の行として指定しなければなりません。

#### 6.3.4 必要なC++ 修飾子

COM for OpenVMS アプリケーションを作成する場合は、次のC++ 修飾子を指定しなければなりません。

- /EXCEPTIONS=CLEANUP

C++ 例外を有効にするために、C++ コマンドに/EXCEPTIONS=CLEANUP修飾子を指定します。

- /STANDARD=CFRONT

C++ コンパイラでは多くのコンパイル標準規格がサポートされます。可能な限り/STANDARD=CFRONTを使用してください。

/STANDARD=CFRONTは、AT&T® cfront 実装に定義されている言語表記規則に従うように、コンパイラに要求します。Wind/U ヘッダ・ファイルはOpenVMS プラ

ットフォームの他に、複数の UNIX プラットフォームでも使用されるため、これらのヘッダ・ファイルに対してこのスイッチを使用すると便利です。

### 6.3.5 必要な C 修飾子

COM for OpenVMS アプリケーションを開発するときに指定しなければならない DEC C 固有の修飾子はありません。

---

## 6.4 ステップ 4: COM アプリケーションをリンクする

COM for OpenVMS アプリケーションを作成するには、クライアント・イメージとコンポーネント・イメージの両方を作成しなければなりません。コンポーネントは、プロセス内コンポーネントまたはプロセス外コンポーネントとして実装できるため、共用可能イメージと実行可能イメージのどちらか一方、または両方を作成する必要があります。新しいインタフェースを作成する場合は、プロキシ/スタブ共用可能イメージも作成しなければなりません。プロキシ/スタブ共用可能イメージは、リモート・メソッド呼び出しの準備として、そのインタフェースのパラメータをパッケージするインタフェース固有のオブジェクトを提供します。プロキシは送信側のアドレス空間で動作し、受信側のアドレス空間の対応するスタブと通信します。

ここでは、クライアント、コンポーネント、プロキシ/スタブ・イメージをリンクするときに必要な手順について説明します。

### 6.4.1 クライアントとプロセス外コンポーネントのリンク

クライアント・イメージやコンポーネント実行可能イメージをリンクするために、修飾子を指定する必要はありませんが、次のように 2 つのイメージをリンクしなければなりません。

- Wind/U 共用可能イメージ (Win32 への依存関係を満たすため)
- DCOM OLE32 共用可能イメージ (COM API への参照を満たすため)

リンク時に発生する依存関係は次のとおりです。

- DCOM\$WIN32:WINDU.OPT
- DCOM\$LIBRARY:DCOM.OPT

1 つ以上の C++ モジュールがある場合は、標準の OpenVMS リンカの代わりに C++ リンカ (CXXLINK) を使用して、C++ リポジトリの場所を指定できるようにします (/CXX\_REPOSITORY 修飾子)。次の例を参照してください。

## COM for OpenVMS アプリケーションの開発

### 6.4 ステップ 4: COM アプリケーションをリンクする

```
$ CXXLINK/your-specific-linker-qualifiers  
list-of-object-modules, -  
_$_$ DCOM$WIN32:WINDU.OPT/OPTIONS, DCOM$LIBRARY:DCOM.OPT/OPTIONS -  
_$_$ application.OPT/OPTIONS  
/REPOSITORY=[.CXX_REPOSITORY]
```

オプション・ファイルは次の方法でもインクルードできます。

- コマンド・ラインではなく、オプション・ファイルにオブジェクト・モジュールの一覧を指定する方法。
- DCOM\$LIBRARY:DCOM.OPTを使用する方法。

#### 6.4.2 プロセス内コンポーネント共用可能イメージのリンク

プロセス内コンポーネント共用可能イメージの依存関係の一覧は、クライアント・イメージやコンポーネント実行可能イメージとは少し異なります。リンク時に発生する特定の依存関係は次のとおりです。

- DCOM\$WIN32:WINDU.OPT
- DCOM\$LIBRARY:DCOM.OPT

##### 6.4.2.1 シンボル・ベクタの作成

プロセス内コンポーネント共用可能イメージをリンクするには、COM for OpenVMS が共用可能イメージ内で呼び出すエントリ・ポイントのシンボル・ベクタを作成しなければなりません。Win32 実行時環境では、DllMainエントリ・ポイントに対して標準命名規則が適用されます。次の要素を含まなければなりません。

- `_Windu_接頭辞`
- 実際のエントリ・ポイント名
- イメージ名の形式に応じて、イメージ名またはイメージ名の一部を含む接尾辞。

イメージ名が\$SHRで終わる場合 (たとえば、CMPNT\$SHR) の場合、接尾辞はドル記号 (\$) も含むイメージ名です。

イメージ名が\$SHR以外で終了する場合 (たとえば、CMPNT\_SHARE), 接尾辞は完全なイメージ名です。

たとえば、CMPNT\$SHRという名前のコンポーネント共用可能イメージは、次のオプション・ファイルを使用してシンボル・ベクタを定義します。

```
!  
! The list of symbols exported by CMPNT$SHR.EXE.  
!  
SYMBOL_VECTOR=(-  
    _WindU_DllMain_CMPNT$/DllMain = PROCEDURE,-  
    DllGetClassObject              = PROCEDURE,-  
    DllCanUnloadNow                = PROCEDURE,-  
    DllRegisterServer              = PROCEDURE,-  
    DllUnregisterServer            = PROCEDURE)
```

CMPNT\_SHAREという名前のコンポーネント共用可能イメージは、次のオプション・ファイルを使用してシンボル・ベクタを定義します。

```
!  
! The list of symbols exported by CMPNT_SHARE.EXE.  
!  
SYMBOL_VECTOR=(-  
    _WindU_DllMain_CMPNT_SHARE/DllMain = PROCEDURE,-  
    DllGetClassObject                   = PROCEDURE,-  
    DllCanUnloadNow                     = PROCEDURE,-  
    DllRegisterServer                   = PROCEDURE,-  
    DllUnregisterServer                  = PROCEDURE)
```

### 6.4.3 プロキシ/スタブ共用可能イメージのリンク

プロキシ/スタブ共用可能イメージの依存関係の一覧は、クライアントおよびコンポーネント実行可能イメージの依存関係とは少し異なります。リンク時に発生する特定の依存関係は次のとおりです。

- DCOM\$WIN32:WINDU.OPT
- SYS\$LIBRARY:DCOM\$RPCRT4\_SHR.EXE

#### 6.4.3.1 シンボル・ベクタの作成

プロキシ/スタブ共用可能イメージのリンクでは、COM for OpenVMS が共用可能イメージ内で呼び出すエントリ・ポイントのシンボル・ベクタを作成しなければならないので、もう少し複雑になります。Win32 実行時環境では、DllMainエントリ・ポイントに対して標準の命名規則を適用しています。次の要素を含まなければならない。

- `_Windu_`接頭辞
- 実際のエントリ・ポイント名
- イメージ名の形式に応じて、イメージ名またはイメージ名の一部を含む接尾辞  
イメージ名が\$SHRで終わる場合 (たとえば、PROXY\$SHR)、接尾辞はドル記号 (\$) を含むイメージ名です。  
イメージ名が\$SHR以外で終わる場合 (たとえば、PROXY\_SHARE)、接尾辞は完全なイメージ名です。

たとえば、PROXY\$SHRという名前のプロキシ/スタブ共用可能イメージは、次のオプション・ファイルを使用してシンボル・ベクタを定義します。

## COM for OpenVMS アプリケーションの開発

### 6.4 ステップ 4: COM アプリケーションをリンクする

```
!  
! RPC Shareable Image  
!  
SYS$LIBRARY:DCOM$RPCRT4_SHR.EXE/SHARE  
!  
!  
! The list of symbols exported by PROXY$SHR.EXE.  
!  
SYMBOL_VECTOR=(-  
    _Windu_DllMain_PROXY$/DllMain    = PROCEDURE,-  
    DllGetClassObject                 = PROCEDURE,-  
    DllCanUnloadNow                   = PROCEDURE,-  
    GetProxyDllInfo                   = PROCEDURE,-  
    DllRegisterServer                 = PROCEDURE,-  
    DllUnregisterServer                = PROCEDURE)
```

PROXY\_SHAREという名前のプロキシ/スタブ共用可能イメージは、次のオプション・ファイルを使用してシンボル・ベクタを定義します。

```
!  
! RPC Shareable Image  
!  
SYS$LIBRARY:DCOM$RPCRT4_SHR.EXE/SHARE  
!  
!  
! The list of symbols exported by PROXY_SHARE.EXE.  
!  
SYMBOL_VECTOR=(-  
    _Windu_DllMain_PROXY_SHARE/DllMain = PROCEDURE,-  
    DllGetClassObject                 = PROCEDURE,-  
    DllCanUnloadNow                   = PROCEDURE,-  
    GetProxyDllInfo                   = PROCEDURE,-  
    DllRegisterServer                 = PROCEDURE,-  
    DllUnregisterServer                = PROCEDURE)
```

---

## 6.5 必要な OpenVMS Registry エントリ

この後の節では、必要な OpenVMS Registry エントリについて説明します。

### 6.5.1 HKEY\_CLASSES\_ROOT\CLSID

CLSID サブキーには、システムでサポートされるコンポーネントのすべての CLSID が含まれます。コンポーネントの CLSID はここに登録しなければなりません。登録する各 CLSID には、次の要素を含まなければなりません。

- 名前なし値。最後が 0 の文字列であり、コンポーネントを記述するデータ値です。
- AppID という名前付き値。最後が 0 の文字列あり、コンポーネントの CLSID を表すデータ値です。

### 6.5.1.1 コンポーネント CLSID

クラス識別子 (CLSIDs) は、OLE クラス・オブジェクトに関連付けられたグローバル一意識別子 (GUID) です。COM for OpenVMS サーバ・アプリケーションは通常、CLSIDs を OpenVMS Registry に登録することで、クライアントが OLE クラス・オブジェクトに関連付けられた実行可能コードを検索して、ロードできるようにしています。

コンポーネントの CLSID は `HKEY_CLASSES_ROOT\CLSID` というサブキーに登録します。

コンポーネント CLSID の登録には、次のサブキーを含まなければなりません。

- LocalServer32

このキーの値は最後が 0 の文字列であり、プロセス外サーバ実行可能イメージの場所を表すデータ値です。

- ProgID

このキーの値は最後が 0 の文字列であり、CLSIDs のプログラム ID を表すデータ値です。通常、*program.component.version* という形式です。

- VersionIndependentProgID

このキーの値は最後が 0 の文字列であり、CLSIDs のプログラム ID (バージョン番号を除く ID) を表すデータ値です。通常、*program.component* という形式です。

- InProcServer32

このキーの値は最後が 0 の文字列であり、プロセス内サーバ共用可能イメージの場所を表すデータ値です。

- Type Libraries

タイプ・ライブラリは、IDispatch インタフェースを実装するときに重要です。タイプ・ライブラリは、OLE オートメーション RegisterTypeLib 実行時ルーチン呼び出すときに、それ自体に登録します。TypeLib サブキーをコンポーネントの CLSID の下に追加しなければなりません。TypeLib サブキーには、タイプ・ライブラリの GUID が含まれます。たとえば、次のキーには LIBID を含まなければなりません。

```
HKEY_CLASSES_ROOT\CLSID\{GUID}\TYPELIB {value=LIBID}
```

### 6.5.1.2 プロキシ/スタブ CLSID

プロキシ/スタブ共用可能イメージは、インタフェースのパラメータをパッケージするためにインタフェース固有のオブジェクトを提供します。プロキシ/スタブ共用可能イメージにはオブジェクトが含まれるので、CLSIDs が必要であり、OpenVMS Registry に登録しなければなりません。プロキシの CLSID は、コンポーネントの CLSID と同じ方法で OpenVMS Registry に登録する必要があります。

プロキシの CLSID は `HKEY_CLASSES_ROOT\CLSID` というサブキーに登録しなければなりません。

プロキシ/スタブ CLSID の登録には、次のサブキーを含まなければなりません。

- InProcServer32

InProcServer32の値は最後が0の文字列であり、プロキシ/スタブ共用可能イメージの場所を表すデータ値です。プロキシ/スタブ CLSID とそのサブキーを使用して、COM はプロキシ/スタブ共用可能イメージを検索することができます。

## 6.5.2 HKEY\_CLASSES\_ROOT\インタフェース

Interfaceサブキーには、システムに登録されているすべてのインタフェースが含まれます。コンポーネントのインタフェース ID (IID) をこのサブキーに登録しなければなりません。

登録される各インタフェースには、少なくとも次のいずれかのサブキーが含まれます。

- NumMethods

NumMethodsの値は最後が0の文字列であり、これはインタフェースに含まれるメソッドの数を表すデータ値です。

- ProxyStubClsid32

ProxyStubClsid32の値は最後が0の文字列であり、これはプロキシ/スタブ共用可能イメージの CLSID を表すデータ値です。この CLSID は、第 6.5.1.2 項で説明したものと同一でなければなりません。

---

## 6.6 OpenVMS および Windows のエラー・コードのテキストへの変換

COM コンポーネントを開発し、テストする場合、OpenVMS システムと Windows NT システムから判読できないエラー・コードが返されることがあります。これらのコードをもっとわかりやすくするのに役立つように、Compaq はエラー・コードを表示可能なテキストに変換するための NTASVMSGetMessage ルーチンを提供しています。

このルーチンを実装するには、NTA\_MESSAGE.H ファイルを DCOM\$LIBRARY: ディレクトリにインクルードし、DCOM\$LIBRARY:NTA\_GETMSG.OBJ オブジェクト・モジュールとリンクしなければなりません。

この節では、NTASVMSGetMessage ルーチンについて説明します。

---

## NTA\$VMSGetMessage

NTA\$VMSGetMessage ルーチンはエラー・コードを表示可能なテキストに変換します。入力エラー・コードは次のいずれかでなければなりません。

- OpenVMS エラー・コード
- Windows HRESULT
- Windows Win32 エラー・コード
- “ユーザ定義の” Windows NT 状態コード・セット

---

### フォーマット

*Return=NTA\$VMSGetMessage (status, text, flag, [count])*

---

### 引数

status

OpenVMS 使用法: error\_code  
タイプ: longword (unsigned)  
アクセス: read only  
メカニズム: by value

この状態フィールドは次のいずれかでなければなりません。

---

入力エラー・コード	例
OpenVMS エラー・コード	0x074AA6BA
Windows HRESULT	0x80070031
Windows Win32 エラー・コード	0x00000031
Windows NT ユーザ定義ビットが セットされた状態コード	0xE74AA6BA

---

セキュリティ API が Windows NT 状態コードを返す場合は、状態フィールドの形式は OpenVMS 状態コードと Windows NT 状態コントロール・ビット・セットの OR です。次の例を参照してください。

---

入力エラー・コード	結果
OpenVMS エラー・コード	0x074AA6BA
Windows NT 状態コード	0xE74AA6BA

---

#### text

OpenVMS 使用法: error\_text  
タイプ: character string  
アクセス: write  
メカニズム: by reference

この引数は最後が NULL の文字列であり、SYSS\$GETMSG システム・サービスから返されたテキストが含まれます。返される最大サイズ (SYSS\$GETMSG システム・サービスで定義されているサイズ) は 256 バイトです。メモリが上書きされないようにするために、呼び出し側は少なくとも 257 バイトのバッファ・アドレスを提供しなければなりません。

#### flag

OpenVMS 使用法: flag  
タイプ: longword (unsigned)  
アクセス: read only  
メカニズム: by value

エラー・コードの変換を制御します。次の値は NTA\_MESSAGE.H に定義されています。

NTAWIN\$_UNKNOWN	不明のエラー・コード
NTAWIN\$_VMS	OpenVMS エラー・コード
NTAWIN\$_NT	Windows HRESULT エラー・コード
NTAWIN\$_WINDOWS	Windows Win32 エラー・コード
NTAWIN\$_USER	Windows NT 状態コード

NTAWIN\$\_UNKNOWN という値を指定した場合は、できるだけ正しいテキストを作成できるように、最適な見積もりが行われます。ルーチンはテキストを次のように解析します。

1. まず、Windows HRESULT であるかどうか確認します (上位ニブル= 0x8)。  
Windows HRESULT でない場合は、次のステップに進みます。
2. 次に、Windows NT のユーザ定義状態コードであるかどうか確認します (上位ニブル= 0xE)。この条件が満たされない場合は、次のステップに進みます。
3. これは OpenVMS エラー・コードであると判断されます。  
OpenVMS エラー・コードと Windows Win32 エラー・コードの区別はできません。

#### count

OpenVMS 使用法: FAO count  
タイプ: longword (unsigned)  
アクセス: write  
メカニズム: by reference

この引数は、戻りメッセージに任意に返される FAO 引数の数です。現在、すべての NTAWIN メッセージで ASCII 置換引数 (!AS) だけが使用されています。呼び出し側

は SYSSFAO で置換する前に、すべての数値データを ASCII に変換しなければなりません。

---

## 説明

このルーチンは OpenVMS SYSSGETMSG システム・サービスを使用します。メッセージは SYSSMESSAGE:NTAWINMSG.EXE イメージと SYSSMESSAGE:NTARPCMSG.EXE イメージに格納されます。

このルーチンを呼び出すには、NTA\_MESSAGE.H ファイルを DCOM\$LIBRARY:ディレクトリにインクルードし、SYS\$LIBRARY:DCOM\$WIN32\_SHR 共用可能イメージとリンクしなければなりません。

---

## 返される条件値

SYSSGETMSG システム・サービスからの状態

SYSSGETMSG システム・サービスの詳細については、『OpenVMS System Services Reference Manual』を参照してください。



# 第2部

---

## OpenVMS Registry

この後の章では、OpenVMS Registry データベースの概要、構成、OpenVMS Registry のインタフェースである\$REGISTRY システム・サービスおよび\$REGISTRYW システム・サービスについて説明します。



---

## OpenVMS Registry の概要

---

### 7.1 レジストリとは

Windows NT Registry は、ハードウェアとソフトウェア (オペレーティング・システムとアプリケーションの両方) に関する構成情報を格納した、システム全体で有効な 1 つの階層型データベースです。Windows NT Registry は、Windows 3.x の .ini ファイルに代わるものであり、アプリケーションと構成に関する情報を 1 つの場所に格納できます。

OpenVMS と Windows NT が協調動作できるようにするために、Compaq は OpenVMS でもレジストリを提供しました。Windows NT Registry と同様に、OpenVMS Registry は OpenVMS Registry データベースと OpenVMS Registry サーバの 2 つのコンポーネントで構成されています。OpenVMS Registry データベースはシステム単位またはクラスタ単位の階層型データベースであり、構成情報が格納されます。この情報はキーと関連値という形式で、データベース構造として格納されます。OpenVMS Registry サーバは、OpenVMS Registry データベースの作成とバックアップ、キーと値の作成、表示、変更、削除など、OpenVMS Registry のすべての操作を制御します。

OpenVMS Registry にはインタフェース (COM API とシステム・サービス) が含まれていて、アプリケーションは OpenVMS Registry サーバを制御し、OpenVMS Registry データベースに読み書きできます。また、OpenVMS Registry にはサーバ管理ユーティリティも含まれているので、システム管理者は OpenVMS の DCL コマンド・ラインから OpenVMS Registry 情報を表示し、更新できます。

OpenVMS Registry は Windows NT Registry と互換性があります。RegEdt32 などの NT クライアント・アプリケーションは、OpenVMS Registry に接続して、これを編集することができます。

#### 7.1.1 参考文献

Windows NT Registry および関連項目の詳細については、次の参考文献を参照してください。

- Windows NT レジストリに関する書籍:
  - 『Windows NT Server 4.0 Unleashed』 Jason Garms 著, SAMS Publishing, Indianapolis, IN, 1998。ISBN: 0-672-30933-5

## 7.2 OpenVMS Registry の概念と定義

OpenVMS Registry は Windows NT Registry と同様に、複数の分岐を含む階層型データベースです。

ここでは、OpenVMS Registry データベースの要素と操作について説明します。

### 7.2.1 キー、サブキー、値

キーとは、OpenVMS Registry データベースの基本的な要素の 1 つです。キーには、コンピュータ、システム、ユーザ固有の情報が格納されます。キーは OpenVMS Registry データベースのヘッダ・フィールドです。キーは階層構造 (ツリー構造) に並べることができます。

OpenVMS Registry には、次の 2 つのメイン (ルート) キーがあります。

- HKEY\_USERS には、各ユーザの情報が格納されます。
- HKEY\_LOCAL\_MACHINE には、ハードウェア、ソフトウェア、セキュリティ、一般的なシステム構成情報が格納されます。

HKEY\_CLASSES\_ROOT というキーは、HKEY\_LOCAL\_MACHINE の CLASSES サブキーを指し示します。これらのルート・キーについては、第 7.3 節で詳しく説明します。

サブキーとは、別のキーの子です。各キーには 0 個以上のサブキーを含むことができます。サブキーを使用すると、関連するキーを階層構造またはツリー構造の別のキーの下にまとめることができます。

値エントリ (または値) とは、名前付きのデータ要素です。これはレジストリ・データベース内のレコード・フィールドです。各キーには 0 個以上の値が関連付けられます。値には値名、値タイプ、一連のフラグ、関連データ (値のタイプによって定義されるもの) があります。OpenVMS Registry では次の値タイプをサポートしています。

- 最後が Null の文字列
- 最後が Null の文字列の配列 (最後が Null)
- 環境変数を含む、最後が Null の文字列 (論理名またはシンボル)
- 32 ビット・データ・アイテム
- 64 ビット・データ・アイテム
- 未加工のバイナリ

図 7-1 はキー、サブキー、値の関係を示しています。

図 7-1 キー、サブキー、値の関係

```
Key1=Value1
Key2
|
+-Subkey1=Value1
|
+-Subkey2=Value1,Value2
:
.
```

#### 7.2.1.1 キーと値の揮発性

OpenVMS Registry のキーと値は、不揮発性または揮発性として定義できます。不揮発性キーは OpenVMS Registry ファイルに保存されます。揮発性キーはテンポラリ・ファイルにキャッシュされます。

Windows NT システムでは、揮発性キーと値はシステムの再起動時に削除されます。

OpenVMS では、揮発性キーと値は、クラスタ内のすべてのノードがリブートされる時に自動的に削除されます。OpenVMS では、サーバ・フェールオーバーのときに揮発性キーが削除されないようになっていますが、クラスタのリブート時には削除されず (スタンドアロン・システムでは、揮発性キーと値はシステムのリブート時に削除されます)。

#### 7.2.1.2 キーのライトスルーとライトバック

キーを作成するときに、OpenVMS Registry がキーの変更情報をいつ書き込まなければならぬかを指定できます。書き込みオプションは次のとおりです。

- ライトスルー：変更はディスクに直ちに書き込まれます。
- ライトバック：変更はキャッシュに格納され、後で書き込まれます。

Cache Action属性を使用すると、キーの書き込み属性を指定できます。キーの作成時にキャッシュ・アクション属性を指定しなかった場合は、キーは親の属性を継承します。

SYS\$REGISTRYインタフェースを使用する場合は、キャッシュ・アクション属性の値とは無関係に、直ちに書き込みが行われるように (ライトスルー)、処理中の要求に対してREG\$M\_NOW機能コード修飾子を使用できます。

#### 7.2.1.3 他のキーと値へのキーのリンク

OpenVMS Registry キーは他の OpenVMS Registry キーにリンクすることができ、同じデータに対して複数のパスを提供できます。同様に、OpenVMS Registry の値は他の OpenVMS Registry 値にリンクできます。これらのキーと値のリンク、つまりシンボリック・リンクは、ファイルのリンクによく似ています。シンボリック・リンクは名前の参照です。

たとえば、Key AをKey Bにリンクできます。Key Aとその値を検索すると、Key Bの値が返されます。

また、シンボリック・リンクを連鎖させることもできます。つまり、Key AがKey Bを指し、Key BがKey Cを指すように設定できます。この結果、Key AはKey Cも指すようになります。リンクを指定するには、\$REGISTRY システム・サービスを使用するか、OpenVMS Registry サーバ管理コマンド・ライン・インタフェースを使用します。

#### 7.2.1.4 OpenVMS Registry キーと値の名前の作成規則

キーと値の名前には次の規則が適用されます。

- 1つのキーにサブキーと値を含むことができます。
- キー名では Unicode (4 バイト) 文字を使用できます。ただし、バックスラッシュ (\) 文字と Null 文字は使用できません。少なくとも 1 文字を指定しなければなりません。
- 値の名前は Unicode (4 バイト) 文字で作成できます。
- キー文字列は名前 (たとえば disk) またはパス (たとえば、Hardware\cosmos\disk) のいずれかです。
- 値文字列は名前のみです。
- キーを定義するときに、パスを指定したのに、1つ以上のパス・サブキーが見つからない場合は、これらのサブキーが自動的に作成されます。作成されるキーは親の属性を継承します。
- OpenVMS Registry データベースでは、キーと値の名前で大文字と小文字の区別が保存されます。\$REGISTRY システム・サービスを呼び出すときに、REG\$M\_CASESENSITIVE機能コード修飾子を指定しないと、名前の比較で大文字と小文字は区別されません。
- 純粋なバイナリ・データの場合、値の最大サイズは 1 MB (Windows NT との互換性を維持するため) です。

### 7.2.2 クラス

Class属性を使用すると、各キーに追加情報を格納できます。たとえば、Class text stringを指定すると、指定したキーで使用できるデータ・タイプを格納できます。

### 7.2.3 ハイブ

ハイブとは、OpenVMS Registry に格納されている関連キー、サブキー、値の集まりです。

Windows NT システムでは、ハイブは関連するLOGファイルと共に、

%SystemRoot%\system32\configディレクトリの1つのファイルに格納されます。Windows NTでは、ユーザはディスクの指定されたファイルにハイブを保存することができます、これらのファイルを後でロードすることができます。

OpenVMSシステムでは、OpenVMS Registry データベース全体がREGISTRY\$LOCAL\_MACHINE.REGとREGISTRY\$USERS.REGという2つのハイブで構成されています。OpenVMSでは、ハイブのロードとアンロードはサポートされません。

---

## 7.3 OpenVMS Registry の構造

Windows NT アプリケーションが OpenVMS Registry データベースとインタフェースをとることができるようにするために、OpenVMS Registry データベースには Windows NT Registry であらかじめ定義されているキーとサブキーの一部が含まれています。

OpenVMS Registry には次の定義済み標準キーが含まれています。

- HKEY\_CLASSES\_ROOT

Windows NT システムでは、このキーはドキュメントのクラスの定義とこれらのクラスに関連するプロパティのために予約されています。

OpenVMS システムでは、デフォルト設定でこのキーにはサブキーや値は含まれていません。

このエントリ・ポイントはHKEY\_LOCAL\_MACHINE\SOFTWARE\Classesサブキーにマップされます。

- HKEY\_USERS

Windows NT システムでは、このエントリ・ポイントの下のエントリはローカル・システムのユーザのデフォルト・ユーザ構成と、現在のユーザ構成を定義します。

OpenVMS システムでは、デフォルトでこのキーにはサブキーや値は含まれていません。

- HKEY\_LOCAL\_MACHINE

このエントリ・ポイントの下のエントリは、システム構成情報用に予約されています。

Windows NT システムでは、この領域にはバス・タイプ、システム・メモリ、インストールされているハードウェアとソフトウェアに関する情報が格納されません。

OpenVMS システムでは、このキーには次の定義済みサブキーが含まれていません。

— Hardware

Windows NT システムでは、ブート時に収集した情報からこのキーの揮発性サブキーが作成されます。

OpenVMS システムでは、このキーにはデフォルトでサブキーや値は含まれていません。

— Security

Windows NT システムでは、これらのキーにはローカル・コンピュータのすべてのセキュリティ情報が格納されます。これらのキーの情報はシステムが所有し、必要に応じて保護されます。

OpenVMS システムでは、デフォルトでこのキーにはサブキーや値は含まれていません。

— Software

Windows NT システムでは、このキーにはローカル・システムのソフトウェアに関する情報が格納されます。格納される情報はユーザ単位の構成とは無関係な情報です。

OpenVMS システムでは、このキーには次の定義済みサブキーが含まれます。

- \* Classes
- \* Compaq Computer Corporation
- \* Microsoft

— System

Windows NT システムでは、このキーには装置とサービスに関する情報が格納されます。

OpenVMS システムでは、このキーには次の定義済みサブキーが含まれます。

- \* CurrentControlSet

Windows NT システムでは、このキーには Control , Enum , Hardware Profiles , Services に関する情報が格納されます。

OpenVMS システムでは、このキーは Advanced Server for OpenVMS で使用するために予約されています。

- \* Registry

このサブキーは Windows NT システムにはありません。 OpenVMS システムでは、このキーにはサブキーと値という形式で、 OpenVMS Registry サーバの構成パラメータが格納されます。定義済みサブキーは次のとおりです。

+ File Quotas

OpenVMS Registry データベースを作成する時点では、このサブキーは空です。システム管理者は、OpenVMS Registry ファイルの名前を持つ値を作成することで、各 OpenVMS Registry データベース・ファイルに対してクォータを割り当てることができます。ファイルに対して値が指定されていないときは、OpenVMS Registry サーバはDefault File Quota設定のデフォルト値を使用します。

たとえば、システム管理者はREG\$CPを使用して、1 MBのクォータをOpenVMS Registry REGISTRY\$LOCAL\_MACHINE.REGファイルに割り当てることができます。次のコマンドを使用します。

```
$ MCR REG$CP
REG> CREATE VALUE/NAME=REGISTRY$LOCAL_MACHINE/TYPE=DWORD/ -
_REG> DATA=%D1000000
"hkey_local_machine\system\registry\File
Quotas"
```

+ File Monitor

このサブキーは使用されません。

+ Priority

このサブキーは、OpenVMS Registry データベースを作成する時点では空です。システム管理者は、OpenVMS Registry サーバが存在するクラスタ内のシステムのノード名を持つ値を作成することで、指定のノードでOpenVMS Registry サーバの優先順位を変更できます。

たとえば、システム管理者はREG\$CPサーバ管理ユーティリティを使用して、COSMOSというノードに優先順位 100 を割り当てることができます。次のコマンドを使用します。

```
$ MCR REG$CP
REG> CREATE VALUE/NAME=COSMOS/TYPE=DWORD/DATA=%D100 -
_REG>
"hkey_local_machine\system\registry\Priori
ty"
```

---

## 7.4 OpenVMS Registry への読み込みと書き込み

OpenVMS Registry に対する読み込みと書き込みは、次の方法で行います。

- OpenVMS で提供される COM API を介して COM for OpenVMS を使用する方法。この方法では、アプリケーション・プログラムは OpenVMS Registry キーと値の入力、変更、削除を行うことができます。

- \$REGISTRY および\$REGISTRYW システム・サービスと OpenVMS Registry サーバ管理ユーティリティのコマンドを使用する方法。この方法では、アプリケーション・プログラマは OpenVMS Registry キーと値の入力、変更、削除を行うことができます。詳細については、第 7.4.1 項を参照してください。
- Windows NT から Windows NT Registry API を使用するか、RegEdt32 (Windows NT レジストリ・エディタ) を使用する方法。この方法では、Windows NT ユーザは OpenVMS Registry キーと値の表示と編集を行うことができます。

#### 7.4.1 \$REGISTRY システム・サービス

OpenVMS Registry には、OpenVMS Registry サーバへのインタフェースを提供する 2 つの OpenVMS システム・サービスがあります。OpenVMS Registry システム・サービスを使用すると、OpenVMS Registry データベース内のキー、サブキー、値の照会、更新、作成を行うことができます。

\$REGISTRY システム・サービスと\$REGISTRYW システム・サービスの詳細については、第 10 章を参照してください。

#### 7.4.2 REG\$CP サーバ管理ユーティリティ

REG\$CPサーバ管理ユーティリティを使用すると、OpenVMS DCL プロンプトから OpenVMS Registry 情報を表示し、更新できます。また、このユーティリティでは、必要なシステム特権があれば、OpenVMS Registry データベース全体をファイルにバックアップしたり、ファイルから復元することができます。

REG\$CPサーバ管理ユーティリティの詳細については、第 9 章を参照してください。

---

### 7.5 OpenVMS Registry のセキュリティ

OpenVMS Registry では、OpenVMS と Windows NT の両方のセキュリティ・モデルが実装されています。

OpenVMS Registry データベースにアクセスするには、呼び出し側のプロセスは実行する操作にとって適切な OpenVMS Registry ライト識別子を持っているか (たとえば、読み込み操作の場合はREG\$LOOKUP、書き込み操作の場合はREG\$UPDATE、統計操作の場合はREG\$PERFORMANCE)、呼び出し側プロセスがSYSPRV 特権を持っていない限りなりません。

ここでは、このセキュリティ・モデルについて説明します。

## 7.5.1 OpenVMS セキュリティ・モデル

ユーザが OpenVMS Registry へのアクセスを要求すると、OpenVMS システムはユーザの Windows NT 資格情報をチェックし、次の手順でアクセスを許可します。

### 1. ユーザが Windows NT 資格情報を持っているか。

- ユーザが Windows NT システムから OpenVMS に接続したので Windows NT 資格情報を持っている場合は、OpenVMS はユーザが提供した資格情報をもとに、OpenVMS Registry へのアクセスを許可します。
- ユーザが COM Version 1.1 for OpenVMS クライアントであるので Windows NT 資格情報を持っている場合は、OpenVMS はユーザが提供した資格情報をもとに、OpenVMS Registry へのアクセスを許可します。
- ユーザがシングル・サインオンを介して OpenVMS へログオンしたので Windows NT 資格情報を持っている場合は、OpenVMS はユーザが提供した資格情報をもとに、OpenVMS Registry へのアクセスを許可します。

ユーザが提供した資格情報をもとに、OpenVMS Registry へのアクセスが許可されない場合は、ステップ 2 に進みます。

- ユーザが Windows NT 資格情報を持っていない場合は、次のステップに進みます。

### 2. ユーザが OpenVMS SYSPRV 特権を持っているか。

- SYSPRV 特権を持っている場合は、OpenVMS は OpenVMS Registry への完全なアクセスを許可します。
- ユーザが SYSPRV 特権を持っていない場合は、次のステップに進みます。

### 3. ユーザが REG\$UPDATE、REG\$LOOKUP、REG\$PERFORMANCE のいずれかのライト識別子を持っているか。

- ユーザが REG\$UPDATE、REG\$LOOKUP、REG\$PERFORMANCE のいずれかのライト識別子を持っている場合は、OpenVMS は提供されたライト識別子を使用して OpenVMS Registry へのアクセスを許可します。ユーザは次の方法で OpenVMS Registry データベースにアクセスできます。
  - REG\$UPDATE: 保守要求を除き、OpenVMS Registry に完全にアクセスできます。
  - REG\$LOOKUP: OpenVMS Registry への読み込みアクセスだけを実行できます。
  - REG\$PERFORMANCE: OpenVMS Registry サーバが収集したパフォーマンス・データへのアクセスを許可します。
- ユーザが REG\$UPDATE、REG\$LOOKUP、REG\$PERFORMANCE のいずれかのライト識別子を持っていない場合は、次のステップに進みます。

4. ユーザが Windows NT 資格情報を持っていない場合は、OpenVMS は OpenVMS ユーザに対して Windows NT の Everyone グループ・アクセスを許可します。この場合、OpenVMS ユーザが OpenVMS Registry キーにアクセスできるかどうかは、キーの所有者がキーまたはサブキーを作成したときに、Everyone に対してどのようなアクセス許可を定義したかに応じて異なります。これらのアクセス許可をもとに、OpenVMS ユーザは次のいずれかの操作を行うことができます。
  - キーとそのサブキーを読み込むことができます。
  - キーとそのサブキーを表示できません。

#### 7.5.1.1 AUTHORIZE ユーティリティによる OpenVMS Registry へのアクセス・ライトの付与

OpenVMS Authorize ユーティリティ (AUTHORIZE) を使用して、SYSPRV 特権と REG\$UPDATE、REG\$LOOKUP、REG\$PERFORMANCE 識別子をユーザ・プロセスに追加することができます。

---

#### 重要

---

OpenVMS Registry ライトを与えると、Windows NT のセキュリティ・アクセス・チェックが無効になります。

---

ライト識別子はアプリケーション固有であるため、AUTHORIZE コマンドを使用してライト識別子を作成することはできません。システムでこれらのライトを作成するには、REG\$CP サーバ管理ユーティリティを使用します。REG\$CP サーバ管理ユーティリティを実行すると、デフォルトでこれらのライトが作成されます。REG\$CP は適切な特権が与えられたアカウントから実行しなければなりません。REG\$CP の実行の詳細については、第 9 章を参照してください。

次の例では、SET RIGHTS\_LIST コマンドを使用して、すべてのユーザが OpenVMS Registry データベース内のキーとデータを表示することを許可する方法を示しています。このコマンドは REG\$LOOKUP 識別子をシステム・ライト・リストに追加します。

```
$ SET RIGHTS_LIST/ENABLE/SYSTEM REG$LOOKUP
```

例 7-1 では、AUTHORIZE を使用して OpenVMS Registry ライトを特定のユーザに与えたり、与えたライトを削除する方法を示しています。

例 7-1 ユーザにライトを付与するための AUTHORIZE の使用

(次ページに続く)

#### 例 7-1 (続き) ユーザにライトを付与するための AUTHORIZE の使用

```
$ SET DEF SYSS$SYSTEM
$ RUN AUTHORIZE

UAF> GRANT/IDENTIFIER REG$LOOKUP SMITH 1
UAF> GRANT/IDENTIFIER/ATTRIBUTES=DYNAMIC REG$UPDATE SMITH 2
UAF> REVOKE/IDENTIFIER REG$UPDATE SMITH 3
UAF> GRANT/IDENTIFIER REG$PERFORMANCE SYSTEM 4
```

- 1 この AUTHORIZE コマンドは、REG\$LOOKUP 識別子を Smith というユーザに与え、Smith が OpenVMS Registry データベース内のキーとデータを表示することを許可します。
- 2 この AUTHORIZE コマンドは、REG\$UPDATE 識別子を Smith というユーザに与え、Smith が OpenVMS Registry データベース内のキーとデータを変更することを許可します。動的属性により、Smith は SET RIGHT/ENABLE コマンドと SET RIGHT/DISABLE コマンドを使用して、プロセス・ライト・リストから REG\$UPDATE 識別子を削除したり、復元することができます。
- 3 この AUTHORIZE コマンドは、Smith というユーザに与えた REG\$UPDATE 識別子を削除します。
- 4 この AUTHORIZE コマンドは、REG\$PERFORMANCE 識別子をシステム管理者アカウントに与え、システム管理者が OpenVMS Registry パフォーマンス・データの監視を有効および無効に設定できるようにします。

## 7.5.2 Windows NT のセキュリティ・モデル

Windows NT ユーザは Advanced Server for OpenVMS を介してのみ、OpenVMS Registry にアクセスできます。OpenVMS はユーザの Windows NT 資格情報をもとに、Windows NT ユーザに対して OpenVMS Registry へのアクセスを許可します。

---

## 7.6 OpenVMS Registry サーバ操作の制御

OpenVMS Registry サーバ操作には、ファイル・クォータの制御、サーバ優先順位、エラー回復処理、データベース・バックアップの頻度、OpenVMS Registry サーバのチューニングなどがあります。

ここでは、OpenVMS Registry サーバ操作について説明し、各設定の最小値、最大値、デフォルト値を示します。これらの設定の変更方法については、第 9 章を参照してください。

### 7.6.1 Maximum Reply Age/Age Checker Interval 設定の定義

OpenVMS Registry サーバは、進行中の作業を追跡し、REG\$\_DUPLREQUESTエラーを返すことで、重複する要求に対処します。OpenVMS Registry サーバは、すでに完了した作業に対して重複する要求が受信された場合のために、完了した要求もしばらく残しておきます。この場合、OpenVMS Registry サーバは応答を再作成します。指定の時間が経過した後、要求は破棄されます。Maximum Reply Age設定は、これらの要求の保持時間を指定します。Age Checker Interval設定は、OpenVMS Registry サーバがこの時間を超える要求をどの程度の頻度でチェックするかを指定します。

デフォルト設定では、サーバは5秒ごとに完了した古い要求がないかどうかチェックします。デフォルト設定では、サーバは5秒より古い完了済み要求を破棄します。

設定名	デフォルト値	最小値	最大値
Maximum Reply Age	5	1	60
Age Checker Interval	5	1	60

### 7.6.2 Database Log Cleaner Interval/Initial Log File Size 設定の定義

OpenVMS Registry では、2 フェーズ・コミット・プロセスを使用して変更を OpenVMS Registry データベースに書き込みます。OpenVMS Registry は最初に変更をログ・ファイルに書き込み、その後、ログ・ファイルを OpenVMS Registry データベースに適用します。Database Log Cleaner Interval設定は、OpenVMS Registry がログ・ファイルを OpenVMS Registry データベースに適用する頻度を指定します。OpenVMS Registry がログ・ファイルを適用した後、OpenVMS Registry はInitial Log File Size設定のサイズをもとに、新しいログ・ファイルを作成します。

Database Log Cleaner Interval設定は、データベースに書き込みを行っても、ログ・ファイルを拡張する必要が発生しないように短い時間に設定しなければなりません。また、ログ・ファイルをデータベースに適用する操作を実行している間、データベースへの書き込みができなくなるので、ログを適用するのに必要な時間をかなり短くするために、ログ・ファイル・サイズは小さい値に設定しておかなければなりません。

デフォルト設定では、ログ・ファイルは5秒ごとに適用されます。デフォルト設定では、OpenVMS Registry ログ・ファイルの作成時のサイズは32 ブロック (16 KB) です。

設定名	デフォルト値	最小値	最大値
Database Log Cleaner Interval	5	1	30
Initial Log File Size	32	16	256

### 7.6.3 Default File Quota/File Quota Interval 設定の定義

OpenVMS Registry サーバはファイル・クォータを適用することで、OpenVMS Registry データベース・ファイルのサイズを制限します。ファイル・クォータは、OpenVMS Registry データベースを構成する各ファイルに対して割り当てることができます。ファイル・クォータを割り当てなかった場合は、OpenVMS Registry はDefault File Quota設定を使用します。

OpenVMS Registry サーバはOpenVMS Registry データベース・ファイルのサイズを定期的に再計算して、クォータを超えているかどうか判断します。File Quota Interval設定は、OpenVMS Registry がこの計算を行う頻度を指定します。

デフォルト設定では、Default File Quota設定は 10 MB です。デフォルト設定では、サーバは 30 秒ごとにファイル・クォータを再計算します。

設定名	デフォルト値	最小値	最大値
Default File Quota	0x10000000	0x7d00	0x3ffffff
File Quota Interval	30	10	60

### 7.6.4 Scan Interval 設定の定義

OpenVMS Cluster では、OpenVMS Registry サーバを複数のノードで実行できます。しかし、一度に 1 つの OpenVMS Registry サーバだけがアクティブになることができます。クラスタ内の他の OpenVMS Registry サーバに対する OpenVMS Registry サーバの優先順位によって、どの OpenVMS Registry サーバがアクティブになるかが決定されます。クラスタ構成が変化する場合は、システム管理者は 1 つ以上の OpenVMS Registry サーバの優先順位を調整できます。システム管理者が優先順位を変更した後、クラスタ内の OpenVMS Registry サーバは、最高の優先順位が割り当てられているサーバを判断し、必要に応じてその状態を自動的に変更します。Scan Interval設定は、OpenVMS Registry サーバが優先順位の変更をチェックする頻度を指定します。

デフォルト設定では、サーバは優先順位の変更を 120 秒ごとにチェックします。

設定名	デフォルト値	最小値	最大値
Scan Interval	120	60	300

### 7.6.5 Log Registry Value Error 設定の定義

OpenVMS Registry サーバ・パラメータの値のいずれかが有効な範囲でない場合は、OpenVMS Registry サーバはエラーをログに記録します。OpenVMS Registry が out-of-range エラーを検出すると、OpenVMS Registry サーバはそのパラメータに対

してデフォルト値を使用します。Log Registry Value Error設定は、エラーをログに記録するかどうかを指定するブール値です。

デフォルト設定では、OpenVMS Registry サーバは out-of-range エラーをログに記録しません。

設定名	デフォルト値	最小値	最大値
Log Registry Value Error	0	0	1

### 7.6.6 Operator Communications Interval 設定の定義

I/O エラーが発生した場合、OpenVMS Registry サーバは OPCOM を使用してオペレータ・コンソールにメッセージを表示できます。Operator Communications Interval設定は、I/O エラーが発生した後、そのエラーが続くかどうか判断するために、OpenVMS Registry サーバが待機する時間を指定します。エラーが続く場合は、OpenVMS Registry はメッセージをオペレータ・コンソールに表示します。

デフォルト設定では、OpenVMS Registry サーバは、エラーが 60 秒以上続くときに、メッセージをオペレータ・コンソールに表示します。

設定名	デフォルト値	最小値	最大値
Age Checker Interval	5	1	60
Operator Communication Interval	60	30	120

### 7.6.7 Process Time Limit 設定の定義

OpenVMS Registry サーバは、要求の処理時間が非常に長い場合、メッセージをサーバ・ログ・ファイルに書き込みます。Process Time Limit 設定は、要求の処理時間が長いかどうかを判断するための基準になる値です。

デフォルト設定では、各要求に対して 180 秒の処理時間が認められ、その時間が経過すると、OpenVMS Registry はメッセージをログに記録します。

設定名	デフォルト値	最小値	最大値
Process Time Limit	180	60	600

### 7.6.8 Reply Log Cleaner Interval 設定の定義

OpenVMS Registry サーバは最近の応答のログを管理します。フェールオーバーの場合は、このログを使用して進行中の作業を再実行します。指定された時間が経過

した後、サーバはこれらの応答を破棄します。Reply Log Cleaner Interval設定は、OpenVMS Registry がこれらの応答を破棄する頻度を指定します。

デフォルト設定では、OpenVMS Registry サーバは 5 秒ごとに応答を破棄します。

設定名	デフォルト値	最小値	最大値
Reply Log Cleaner Interval	10	5	60

### 7.6.9 Snapshot Interval/Snapshot Location/Snapshot Versions 設定の定義

OpenVMS Registry サーバは OpenVMS Registry データベースのバックアップ・コピーを管理します。Snapshot Interval設定は、OpenVMS Registry サーバがバックアップ・コピーを作成する頻度を指定します。Snapshot Location設定は、OpenVMS Registry がコピーを格納する場所を指定します。Snapshot Versions設定は、OpenVMS Registry が保存するバックアップ・コピーの数を指定します。

デフォルト設定では、OpenVMS Registry データベースは 1 日に 1 回ずつバックアップにコピーされます。デフォルト設定では、OpenVMS Registry データベースは、SYS\$REGISTRY論理名の定義によって指定される場所にコピーされます。デフォルト設定では、OpenVMS Registry は OpenVMS Registry データベースの 5 つのバージョンを保存します。

設定名	デフォルト値	最小値	最大値
Snapshot Interval	86400	3600	604800
Snapshot Location	SYS\$REGISTRY	—	—
Snapshot Versions	5	1	10

### 7.6.10 Write Retry Interval 設定の定義

OpenVMS Registry データベースに書き込むときに、OpenVMS Registry がエラーを検出すると、OpenVMS Registry サーバはWrite Retry Interval設定によって指定される間隔で書き込みを再実行します。

デフォルト設定では、OpenVMS Registry サーバは OpenVMS Registry データベースへの書き込みが失敗した場合、5 秒ごとに書き込みを再実行します。

設定名	デフォルト値	最小値	最大値
Writer Retry Interval	5	1	30



---

## OpenVMS Registry システム管理

---

### 8.1 OpenVMS Registry のインストール

OpenVMS Registry サーバは、OpenVMS バージョン 7.2-1 システム・インストールの一部としてインストールされます。

OpenVMS Registry を使用する前に、OpenVMS Registry サーバを構成し、OpenVMS Registry データベースに情報を登録しなければなりません。OpenVMS Registry サーバの構成の詳細については、第 8.2 節を参照してください。OpenVMS Registry データベースへの情報の格納については、第 5.2 節を参照してください。

第 8.3 節で説明するスタートアップ・プロセスを使用して、OpenVMS Registry サーバを初めて起動するときに、OpenVMS システムは OpenVMS Registry データベースを作成します。

OpenVMS Registry には複数の方法でアクセスできます。OpenVMS Registry にアクセスする方法に応じて、次の製品をインストールしなければなりません。

- COM API を使用して OpenVMS Registry にアクセスする場合は、COM for OpenVMS をインストールしなければなりません。詳細については、第 4 章を参照してください。
- Windows NT アプリケーション RegEdt32 を使用して OpenVMS Registry にアクセスする場合は、最初に Advanced Server for OpenVMS をインストールし、構成し、起動しなければなりません。詳細については、Advanced Server for OpenVMS のマニュアルを参照してください。

OpenVMS Registry サーバ管理ユーティリティや OpenVMS Registry システム・サービスを使用して、OpenVMS Registry にアクセスすることもできます。これらは OpenVMS バージョン 7.2-1 では OpenVMS Registry の一部としてインストールされます。

---

### 8.2 OpenVMS Registry の構成：REG\$CONFIG Configuration ユーティリティ

OpenVMS Registry Configuration ユーティリティ (REG\$CONFIG) は、OpenVMS Registry サーバの状態と OpenVMS Registry データベースの場所に関する情報を提供します。このユーティリティを使用すると、OpenVMS Registry 論理名とパスを変更できます。

OpenVMS Registry Configuration ユーティリティを起動するには、次のコマンドを入力します。

```
$ @SYS$MANAGER:REG$CONFIG
```

次のメニューが表示されます。

```
-----  
OpenVMS Registry Configuration Utility  
~~~~~  
1 - Configure OpenVMS Registry logical names and directory paths  
2 - Display OpenVMS Registry logical names and directory paths  
3 - Check the state of the OpenVMS Registry server  
4 - Start the OpenVMS Registry server on this node  
H - Help about this utility  
[E] - Exit  
Please enter your choice :  
-----
```

オプションを選択するには、オプション番号を入力します。オプションは次のとおりです。

- 1 - Configure OpenVMS Registry logical names and directory paths  
OpenVMS Registry サーバのスタートアップ値を構成し、OpenVMS Registry データベースの場所を指定できます。  
この手順については、第 8.2.1 項を参照してください。
- 2 - Display OpenVMS Registry logical names and directory paths  
このノードの OpenVMS Registry サーバ論理名 (スタートアップ値) と OpenVMS Registry データベースの場所の現在の値を表示します。
- 3 - Check the state of the OpenVMS Registry server  
OpenVMS Registry サーバの現在の状態を表示します。次のいずれかが表示されます。  
  
The OpenVMS Registry server is started in the cluster.  
The OpenVMS Registry server is started on this node.  
The OpenVMS Registry server is not started.
- 4 - Start the OpenVMS Registry server on this node  
現在のノードで OpenVMS Registry サーバを起動します。次のメッセージが表示されます。  
  
The OpenVMS Registry server has successfully started.

- H - Help about this utility

OpenVMS Registry Configuration ユーティリティのオプションに関するオンライン・ヘルプを表示します。

- [E] - Exit

OpenVMS Registry Configuration ユーティリティを終了します。

---

ヒント: Q (Quit) はいつでも入力可能

---

プロンプトに対してQと入力すると、「OpenVMS Registry Configuration Utility」メニューにいつでも戻ることができます。

論理名を構成しているときに Quit を使用した場合は、確認メッセージを受け取った値だけが更新されます。

---

### 8.2.1 OpenVMS Registry の値の構成

次の質問が表示されます。

1. スタンドアロンまたはクラスタ情報の入力が必要とされます。次のメッセージが表示されます。

```
Is this system now a node in a cluster or will this system
become part of a cluster? (Y/N/Q):
```

2. REG\$TO\_BE\_STARTED論理名に関する現在の情報が表示され、値を変更するかどうか質問されます。

```
- REG$TO_BE_STARTED -
```

```
[current value of REG$TO_BE_STARTED]
```

```
NOTE: Setting this logical to TRUE starts the OpenVMS Registry
server automatically when the system boots. Setting this logical
to FALSE prevents the OpenVMS Registry server from starting
when the system boots and prevents other products from starting
the OpenVMS Registry server. If the OpenVMS Registry Server is not
started at boot time, but other products that require an OpenVMS
Registry server are able to start the OpenVMS Registry server, you
do not need to assign a value to this logical.
```

```
Do you want to change this value? (Y/N/Q) [Y]:
```

Yを選択すると、新しい値の入力が求められます。

```
Enter the new value (TRUE/FALSE/NOVAL/Q):
```

次のいずれかの値を入力します。

動作	値
リポート時に OpenVMS Registry サーバを起動する。他の製品がサーバを起動することを許可する。	TRUE
リポート時に OpenVMS Registry サーバを起動しない。他の製品がサーバを起動することを許可しない。	FALSE
リポート時に OpenVMS Registry サーバを起動しない。他の製品がサーバを起動することを許可する(論理名の割り当てを解除する)。	NOVAL
この手順を中止し、「OpenVMS Registry Configuration Utility」メニューに戻る。	Q

In which logical name table do you want the logical defined?  
(SYSTEM/SYSCLUSTER/CLUSTER/Q) :

次のいずれかを入力します。

動作	値
REG\$TO_BE_STARTED論理名を LNMS\$SYSTEM 論理名テーブルに追加する。このテーブルには、システム内のすべてのプロセスで共用される名前が格納される。	SYSTEM
REG\$TO_BE_STARTED論理名を LNMS\$SYSCLUSTER 論理名テーブルに追加する。このテーブルには OpenVMS Cluster 内のすべてのプロセスで共用される名前が格納される。	SYSCLUSTER
REG\$TO_BE_STARTED論理名を LNMS\$CLUSTER 論理名テーブルに追加する。このテーブルはすべてのクラスタ単位の論理名テーブルの親テーブルである。	CLUSTER
この手順を中断し、「OpenVMS Registry Configuration Utility」メニューに戻る。	Q

新しい値または更新した値を入力した後、システムから変更が確認され、SYLOGICALS.COM ファイルに追加しなければならない行が表示されます。

```
The logical REG$TO_BE_STARTED has been temporarily defined.
Before you reboot the system you must edit your SYLOGICALS.COM
to include the line:
```

```
DEFINE/TABLE=table-name REG$TO_BE_STARTED
value
```

Press [Enter] to continue.

3. SYS\$REGISTRY論理名に関する現在の情報が表示され、値を変更するかどうか質問されます。

```
- SYS$REGISTRY logical -
current value of SYS$REGISTRY
```

## 8.2 OpenVMS Registry の構成 : REG\$CONFIG Configuration ユーティリティ

Note: When the OpenVMS Registry server is started, the system creates an OpenVMS Registry database at this location. If an OpenVMS Registry database already exists on your system, you must redefine the SYS\$REGISTRY logical to point to the existing OpenVMS Registry database location.

Do you wish to change this value? (Y/N/Q) [Y]:

Yを選択すると、新しい値が求められます。

Enter the new value for SYS\$REGISTRY ("yourvalue"/NOVAL/Q):

次のいずれかを入力します。

動作	値
OpenVMS Registry データベースの新しい場所または変更した場所を定義する。	DKA0:[SYS\$REGISTRY]などの有効なディレクトリ指定
論理名の割り当てを解除する。	NOVAL
この手順を中断し、「OpenVMS Registry Configuration Utility」メニューに戻る。	Q

4. 更新した値が表示され、値を確認するように求められます。

You have entered: value  
Is this correct? (Y/N/Q) [Y]:

5. 新しい論理名または更新した論理名を格納する論理名テーブル名の入力求められます。

In which logical name table do you want the logical defined?  
(SYSTEM/SYSCLUSTER/CLUSTER/Q):

次のいずれかを入力します。

動作	値
SYS\$REGISTRY論理名を LNM\$SYSTEM 論理名テーブルに追加する。このテーブルには、システム内のすべてのプロセスで共用される名前が格納される。	SYSTEM
SYS\$REGISTRY論理名を LNM\$SYSCLUSTER 論理名テーブルに追加する。このテーブルには、OpenVMS Cluster 内のすべてのプロセスで共用される名前が格納される。	SYSCLUSTER
SYS\$REGISTRY論理名を LNM\$CLUSTER 論理名テーブルに追加する。このテーブルはすべてのクラスタ単位の論理名テーブルの親テーブルである。	CLUSTER
この手順を中断し、「OpenVMS Registry Configuration Utility」メニューに戻る。	Q

新しい値または更新した値を入力した後、変更が確認され、SYLOGICALS.COM ファイルに追加しなければならない行が表示されます。

The logical SYS\$REGISTRY has been temporarily defined.  
Before you reboot the system you must edit your SYLOGICALS.COM file  
to include the line:

```
DEFINE/TABLE=table-name SYS$REGISTRY  
dir-spec
```

Press [Enter] to continue.

#### 6. OpenVMS Registry データベースの場所に関する情報が表示されます。

- SYS\$REGISTRY directory -

[directory status]

ディレクトリが存在しない場合、ディレクトリを作成するかどうか質問されま  
す。

!!Caution!! When the OpenVMS Registry server starts, the system  
creates an OpenVMS Registry database at this location. If you  
already have an OpenVMS Registry database on your system, you must  
redefine the SYS\$REGISTRY logical to point to that location.

Do you wish to create the directory? (Y/N/Q) [Y]:

Yと入力すると、ディレクトリの作成が確認されます。

The SYS\$REGISTRY directory has now been created.

Press [Enter] to return to the menu.

---

## 8.3 OpenVMS Registry の起動

OpenVMS Registry を起動する方法は制御できます。

- システムのリポート時に OpenVMS Registry を自動的に起動できます。
- OpenVMS Registry の稼動が必要な製品が OpenVMS Registry を起動するように設定できます。
- OpenVMS Registry を手動で起動できます。
- OpenVMS Registry が起動されないように設定できます。

OpenVMS Registry の起動方法を制御するには、第 8.2 節で説明した OpenVMS  
Registry Configuration ユーティリティを使用します。

### 8.3.1 OpenVMS Registry の手動起動

条件によっては、OpenVMS Registry サーバを手動で起動しなければならないこと  
があります。

このような場合は、SYS\$STARTUP:REG\$STARTUP.COM コマンド・プロシージャを使用してください。次のコマンド・プロシージャは、サーバ・プロセス・クォータが必要な最小値に設定されていることを確認します。

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

また、次のコマンドを使用して、OpenVMS Registry を手動で起動することもできます。

```
$ SET SERVER REGISTRY_SERVER/START
```

---

## 8.4 OpenVMS Registry のシャットダウン

OpenVMS Registry サーバは、システム・シャットダウンの一部として自動的にシャットダウンされます。

OpenVMS Registry を手動でシャットダウンする場合は、次のコマンドを使用します。

```
$ SET SERVER REGISTRY_SERVER/EXIT
```

---

## 8.5 OpenVMS Registry サーバのコマンド

OpenVMS Registry サーバのコマンドを使用すると、OpenVMS Registry サーバの状態を表示 (SHOW) したり、変更 (SET) することができます。この節では、OpenVMS Registry サーバのコマンドについて説明します。

---

## SHOW SERVER REGISTRY\_SERVER

指定されたノードの OpenVMS Registry の現在の状態を表示します。

このコマンドを使用するには、SYSPRV 特権が必要です。

---

### フォーマット

```
SHOW SERVER REGISTRY_SERVER  
[/MASTER | /CLUSTER | /NODE=(node,...)]  
[/PAGE]
```

---

### 修飾子

**/MASTER**

クラスタ内の現在の OpenVMS Registry マスタ・サーバのノードおよびプロセス ID (PID) を表示します。このコマンドはクラスタ内の OpenVMS Registry サーバと通信しません。この修飾子を使用するには、SYSPRV 特権の他に、SYSLCK 特権も必要です。

**/CLUSTER**

クラスタ内の各 OpenVMS Registry サーバから出力を返します。最初に OpenVMS Registry マスタ・サーバ情報が表示されます。

**/NODE=(node,...)]**

指定されたノードのサーバに関する OpenVMS Registry サーバ情報を返します。情報はノード名を入力した順に表示されます。指定するノード名は現在のクラスタ内のノードでなければなりません。

**/PAGE**

スクロール可能なページ形式で、返された出力を表示します。

---

# SET SERVER REGISTRY\_SERVER

OpenVMS Registry の状態を変更します。

このコマンドを使用するには、SYSPRV 特権が必要です。

---

## フォーマット

```
SET SERVER REGISTRY_SERVER  
[/MASTER | /CLUSTER | /NODE=(node,...)]  
[/START | /RESTART | /EXIT | /ABORT ]  
[[/NO]LOG ]
```

---

## 修飾子

**/MASTER**

指定されたコマンドを OpenVMS Registry マスタ・サーバに対してだけ実行します。SYSPRV 特権の他に SYSLCK 特権も必要です。

**/CLUSTER**

SET コマンドをクラスタ内の各 OpenVMS Registry サーバに対して実行し、最後に OpenVMS Registry マスタ・サーバを設定します。

**/NODE=(node,...)**

SET コマンドを指定されたノードの OpenVMS Registry サーバに対して実行します。コマンドはノード名の入力順に実行されます。ノード名は現在のクラスタ内のノードでなければなりません。

**/START[=(node,...)]**

クラスタ内の指定された 1 つ以上のノードで OpenVMS Registry サーバを起動します。

**/EXIT[=(node,...)]**

クラスタ内の 1 つ以上の指定されたノードで OpenVMS Registry サーバを停止します。

**/ABORT[=(node,...)]**

クラスタ内の 1 つ以上の指定されたノードで OpenVMS Registry サーバを中断します。

**[/NO]LOG**

新しい OpenVMS Registry ログ・ファイルを SYSS\$REGISTRY に作成します。NOLOG はデフォルトです。

---

## 8.6 クラスタ内での OpenVMS Registry のフェールオーバ

クラスタ内で複数の OpenVMS Registry サーバを稼動すれば、OpenVMS Registry の可用性と信頼性を向上できます。ただし、各ノードで稼動できるのは 1 つだけです。OpenVMS Registry サーバをいくつ稼動する場合でも、OpenVMS Registry データベースは 1 つだけ存在します。

クラスタ内で複数の OpenVMS Registry サーバを稼動する場合は、OpenVMS Registry サーバ・プロセスが 1 つだけアクティブになり、OpenVMS Registry データベースに書き込みを行います。他の OpenVMS Registry サーバ・プロセスはスタンバイ状態になります。

デフォルト設定では、クラスタ内で最初にアクティブになる OpenVMS Registry サーバ・プロセスは、そのプロセスが存在しなくなるか、OpenVMS Registry サーバ・プロセス間の優先順位が変更されるまでアクティブです。

### 8.6.1 OpenVMS Registry サーバ・プロセスの優先順位の変更

OpenVMS Registry サーバ・プロセスの優先順位は、OpenVMS Registry サーバ・プロセスを実行するクラスタ内の各ノードの優先順位を作成および変更することで変更できます。値が大きくなるほど、優先順位は高くなります。

例 8-1 は、優先順位の割り当てを示しています。ここでは、NODENAME1 がクラスタ内でアクティブな OpenVMS Registry サーバ・プロセスになるように、優先順位が割り当てられています。

#### 例 8-1 優先順位の設定

```
$ mcr reg$cp
REG> CREATE VALUE HKEY_LOCAL_MACHINE\SYSTEM\REGISTRY\PRIORITY -
_REG> /NAME=NODENAME1/DATA=15/TYPE=DWORD
REG> CREATE VALUE HKEY_LOCAL_MACHINE\SYSTEM\REGISTRY\PRIORITY -
_REG> /NAME=NODENAME2/DATA=10/TYPE=DWORD
REG> CREATE VALUE HKEY_LOCAL_MACHINE\SYSTEM\REGISTRY\PRIORITY -
_REG> /NAME=NODENAME3/DATA=5/TYPE=DWORD
```

例 8-1 で NODENAME1 がシャットダウンされると、OpenVMS Registry データベースの制御は NODENAME2 のサーバ・プロセスに渡されます。

例 8-2 は、システム管理者が NODENAME3 の優先順位を 20 に上げる手順を示しています。

#### 例 8-2 優先順位の変更

```
$ mcr reg$cp  
REG> MODIFY VALUE HKEY_LOCAL_MACHINE\SYSTEM\REGISTRY\PRIORITY -  
_REG> /NAME=NODENAME3/DATA=20/TYPE=DWORD
```

例 8-2 では、NODENAME1の OpenVMS Registry サーバ・プロセスはスタンバイ・モードになり、NODENAME3の OpenVMS Registry サーバ・プロセスがアクティブになります。

---

## 8.7 Windows NT システムからの OpenVMS Registry への接続

Windows NT システムから OpenVMS Registry に接続するには、次の操作を行わなければなりません。

- OpenVMS システム:
  - Advanced Server for OpenVMS をインストールします。
  - Advanced Server for OpenVMS を構成します。
- OWindows NT システム:
  - 必要なハードウェアをインストールし、構成します。
  - Windows NT Server または Workstation ソフトウェアをインストールし、構成します。

Windows システムから OpenVMS Registry データベースにアクセスする場合は、Windows NT システムで与えられているすべての権限が与えられます。たとえば、Windows NT システムに Administrator としてログオンしている場合は、OpenVMS Registry のすべてのキーと値の読み込みと書き込みが可能です。OpenVMS Registry キーへのアクセスは、Windows NT ユーザ・プロファイル (username と Group メンバシップ) をもとに行われます。Advanced Server for OpenVMS を介して OpenVMS Registry に接続します。キー、値、セキュリティ設定の表示と変更には、Windows Regedt32 アプリケーションを使用します。

---

### 警告

---

OpenVMS Registry データベースのキーと値を変更する場合は、十分注意してください。OpenVMS Registry データベースが破損すると、OpenVMS システムまたはクラスタのすべてのアプリケーションとユーザに影響します。

---

## 8.8 OpenVMS Registry クォータ

クォータは、OpenVMS Registry データベースのサイズを制限します。すべての OpenVMS Registry ファイルのルート・キー・データ・ファイルにクォータが割り当てられます。デフォルト設定では、これらのルート・キーはUSERSキー (REGISTRY\$USERS.REG) とLOCAL\_MACHINEキー (REGISTRY\$LOCAL\_MACHINE.REG) です。

クォータはファイルに格納される情報のサイズを制限しますが、他のファイルに格納される情報のサイズはクォータに含まれません。これらのファイルがサブツリーの一部の場合でも、そのファイルのサイズは含まれません。

デフォルト・クォータと各ファイルのクォータは、OpenVMS Registry の HKEY\_LOCAL\_MACHINE\SYSTEM\Registryキーに格納されます。これらのキーの詳細については、第 7.3 節を参照してください。

---

## 8.9 OpenVMS Registry のセキュリティ

ユーザは次の方法で OpenVMS Registry に直接にアクセス (読み込みと変更) できません。

- Windows NT システムから (Advanced Server for OpenVMS を介した接続によって) アクセスする方法
- OpenVMS Registry システム・サービスを使用する方法 (\$REGISTRY[W])
- OpenVMS Registry サーバ管理ユーティリティ (REG\$CP) を使用する方法

各ユーザが必要とするシステム特権とライト識別子の詳細については、第 7.5.1 項を参照してください。必要なシステム特権とライト識別子を与える方法については、第 7.5.1.1 項を参照してください。

キーのセキュリティ属性は Windows NT システムからだけ変更できます。OpenVMS システムからキーのセキュリティ属性を変更することはできません。OpenVMS では、Windows NT セキュリティ属性の作成や管理は行われません。

---

## 8.10 OpenVMS Registry データベースのバックアップと復元

OpenVMS Registry にはサーバ管理ユーティリティが含まれており、このユーティリティを使用すると、必要なシステム特権が与えられている場合、OpenVMS DCL プロンプトから OpenVMS Registry データベース全体をファイルにバックアップしたり、ファイルから復元することができます。

OpenVMS Registry データベースのバックアップと復元の詳細については、第 9.2 節と、REG\$CPサーバ管理ユーティリティの CREATE SNAPSHOT コマンドと EXPORT コマンドを参照してください。

---

## 8.11 OpenVMS Alpha 複合バージョン・クラスタでの OpenVMS Registry の使用

OpenVMS Registry サーバは OpenVMS Alpha 複合バージョン・クラスタ内で実行できます。つまり、OpenVMS Registry は、OpenVMS バージョン 7.2-1 以外の OpenVMS バージョンを含むクラスタ内で実行できます。ただし、OpenVMS Registry サーバは、OpenVMS バージョン 7.2-1 を稼動しているノードで実行しなければなりません。

---

## 8.12 国際化と Unicode のサポート

Windows NT と統合するために、OpenVMS Registry は Unicode に準拠しています。Unicode の詳細については、『OpenVMS バージョン 7.2 新機能説明書』を参照してください。



---

## OpenVMS Registry サーバの管理

---

### 9.1 コマンド・ラインからの OpenVMS Registry サーバの管理

OpenVMS Registry にはサーバ管理ユーティリティがあり、OpenVMS DCL プロンプトから OpenVMS Registry 情報の更新と表示を行うことができます。また、必要なシステム特権があれば、このユーティリティを使用して、OpenVMS Registry データベース全体をファイルにバックアップしたり、ファイルから復元することができます。

OpenVMS Registry データベースのバックアップと復元の詳細については、第 9.2 節と、この章のコマンド参照の節の CREATE SNAPSHOT、EXPORT、IMPORT コマンドを参照してください。

OpenVMS Registry サーバ管理ユーティリティを起動するには、次のいずれかのコマンドを入力します。

```
$ RUN SYS$SYSTEM:REG$CP
```

または

```
$ MCR REG$CP
```

---

#### 注意

---

OpenVMS Registry データベースにアクセスするには、クラスタまたはスタンドアロン・システムで OpenVMS Registry サーバをあらかじめ稼動しておかなければなりません。

---

表 9-1 は、OpenVMS Registry サーバ管理ユーティリティのコマンドを示しています。

## OpenVMS Registry サーバの管理

### 9.1 コマンド・ラインからの OpenVMS Registry サーバの管理

表 9-1 OpenVMS Registry サーバ管理ユーティリティのコマンド

コマンド	識別子	動作
CREATE DATABASE	SYSPRV	新しい OpenVMS Registry データベース・ファイルを作成する。
CREATE KEY	REG\$UPDATE	1 つ以上のキーを OpenVMS Registry データベースに作成する。
CREATE SNAPSHOT	SYSPRV	OpenVMS Registry データベース・ファイルの即時バックアップを作成する。
CREATE VALUE	REG\$UPDATE	キーのデータ・コンポーネントを指定する。
DELETE KEY	REG\$UPDATE	OpenVMS Registry データベースから 1 つ以上のキーを削除する。
DELETE VALUE	REG\$UPDATE	指定されたキーから 1 つ以上の値を削除する。
EXPORT	REG\$LOOKUP	OpenVMS Registry をテキスト形式でエクスポートする。
IMPORT	REG\$UPDATE	レジストリ・データベースのテキスト形式バージョンを OpenVMS Registry 形式にインポートする。
LIST KEY	REG\$LOOKUP	指定されたキーのすべてのサブキー情報を表示する。
LIST VALUE	REG\$LOOKUP	指定されたキーのすべての値を表示する。
MODIFY KEY	REG\$UPDATE	指定されたキーの情報を変更する。
MODIFY VALUE	REG\$UPDATE	指定された値の情報を変更する。
MODIFY TREE	REG\$UPDATE	指定されたキーとそのサブキーの情報を変更する。
SEARCH KEY	REG\$LOOKUP	指定されたキーと一致するすべてのキーのパス名を表示する。
SEARCH VALUE	REG\$LOOKUP	指定された値名と一致するすべてのキーのパス名を表示する。
SHOW COUNTERS	REG\$PERFORMANCE	カウンタ情報を表示する。
SHOW FILE	REG\$PERFORMANCE	OpenVMS Registry データベース・ファイルの統計情報を表示する。
SHOW INTERNAL	REG\$PERFORMANCE	内部値 (共用ライブラリで使用) を表示する。
START MONITOR	REG\$PERFORMANCE	監視機能を有効にする。
STOP MONITOR	REG\$PERFORMANCE	監視機能を無効にする。
ZERO COUNTERS	REG\$PERFORMANCE	監視カウンタをリセットする。

---

注意

---

SYSPRV 特権が与えられているユーザは、表 9-1 に示したすべてのコマンドを実行できます。ユーザに SYSPRV 特権が与えられていない場合のみ、OpenVMS Registry 識別子を指定しなければなりません。

ユーザに REG\$UPDATE 識別子を与えると、ユーザは表 9-1 に示したコマンドの他に、次のコマンドも実行できるようになります。

```
LIST KEY  
LIST VALUE  
SEARCH KEY  
SEARCH VALUE
```

---

---

## 9.2 OpenVMS Registry データベースのバックアップと復元

REG\$CP サーバ管理ユーティリティには、OpenVMS Registry データベースのバックアップと復元を行うための 2 つのコマンドがあります。

- EXPORT コマンドを使用すると、OpenVMS Registry のキーと値を OpenVMS または Windows NT 形式で必要に応じてバックアップできます。

このコマンドを使用すると、OpenVMS Registry データベースの一部または全部をエクスポートできます。対応する IMPORT コマンドを使用すると、OpenVMS Registry または Windows NT レジストリのキーと値を復元またはインポートできます。

詳細については、この章のコマンド参照の節の EXPORT コマンドと IMPORT コマンドを参照してください。

- CREATE SNAPSHOT コマンドを使用すると、指定したスケジュールで OpenVMS Registry データベース・ファイルを自動的にバックアップできます。

デフォルト設定では、REGISTRY\_SERVER プロセスは 24 時間ごとに OpenVMS Registry データベースのスナップショットを作成します。この間隔は OpenVMS Registry サーバ操作で Snapshot Interval 設定を変更することにより変更できます (この操作の詳細については、第 7.6 節を参照してください)。

次の例では、OpenVMS Registry データベースの自動スナップショットの間隔を、デフォルトの 24 時間ごとに 1 回から毎時 1 回に変更する方法を示しています。

```
$ MCR REG$CP  
REG> MODIFY VALUE  
HKEY_LOCAL_MACHINE\SYSTEM\REGISTRY -  
_REG> /NAME="Snapshot Interval"/DATA=3600/TYPE=DWORD
```

詳細については、この章の CREATE SNAPSHOT コマンドを参照してください。

### 9.2.1 OpenVMS Registry データベースのスナップショットの作成

OpenVMS Registry データベースのスナップショットを作成するには、次の操作を行います。

1. REGISTRY\_SERVER プロセスがクラスタ内で動作していることを確認します。
2. SYSPRV 特権が与えられているアカウントから次のコマンドを入力します。

```
$ MCR REG$CP  
REG> CREATE SNAPSHOT
```

この操作で作成されるスナップショットは、次の 2 つのファイルで構成されま  
す。これらのファイルは指定したディレクトリに格納されます。

```
REGISTRY$LOCAL_MACHINE.RSS  
REGISTRY$USERS.RSS
```

### 9.2.2 OpenVMS Registry データベースのスナップショットの復元

OpenVMS Registry データベースのスナップショットを復元するには、次の操作を行います。

1. クラスタ内のすべてのノードで REGISTRY\_SERVER プロセスをシャットダウンします  
( OpenVMS Registry のシャットダウンの詳細については、第 8.4 節を参照してく  
ださい)。

2. OpenVMS Registry スナップショット・ファイルが SYS\$REGISTRY ディレクトリに  
あることを確認します。

OpenVMS Registry スナップショット・ファイルが SYS\$REGISTRY ディレクトリに  
ない場合は、OpenVMS Registry スナップショット・ファイルを SYS\$REGISTRY ディ  
レクトリにコピーします。

3. 次のように、OpenVMS Registry スナップショット・ファイルの名前を変更しま  
す。

```
$ RENAME REGISTRY$LOCAL_MACHINE.RSS REGISTRY$LOCAL_MACHINE.REG  
$ RENAME REGISTRY$USERS.RSS REGISTRY$USERS.REG
```

4. REGISTRY\_SERVER プロセスを再起動します ( OpenVMS Registry を手動で起動する  
方法については、第 8.3.1 項を参照してください)。

---

**重要**

---

前回スナップショットを作成してからこの復元処理を実行するまでの間に、  
OpenVMS Registry データベースに書き込まれた情報は失われます。

---

---

## 9.3 OpenVMS Registry サーバ管理ユーティリティの構文

この後のコマンドの説明では、OpenVMS Registry の各コマンドをアルファベット順に説明します。

---

### 注意

---

ここで説明するどのコマンドでも、key-nameパラメータは、キーの完全なパスを指定する文字列であり、次のいずれかのエントリ・ポイントから始まります。

```
HKEY_LOCAL_MACHINE  
HKEY_USERS  
HKEY_CLASSES_ROOT
```

また、REG\$\_HKEY\_LOCAL\_MACHINE、REG\$\_HKEY\_USERS、REG\$\_HKEY\_CLASSES\_ROOTという文字列も指定できます。

どのサーバ管理コマンドでも、リンクは追従されていません(リンクの詳細については、第7.2.1.3項を参照してください)。

キーと値の名前で大文字と小文字を区別するには、キーと値を引用符で囲みます(たとえば、"value")。

---

---

## CREATE DATABASE

基本的な OpenVMS Registry データベース・ファイルを、SYS\$REGISTRY論理名で指定される場所に作成します。このコマンドは空のデータベースを作成し、定義済みキーをロードします。

このコマンドを入力したときに、データベース・ファイルがすでに存在する場合は、既存のファイルに上書きされません。ファイルがすでに存在することを示す警告が表示されます。新しい OpenVMS Registry データベースを作成する場合は、最初にデータベース・ファイルの以前のすべてのバージョンを削除しなければなりません。OpenVMS Registry データベース・ファイルを削除すると、OpenVMS Registry に格納されているすべてのキー、サブキー、値が失われます。

このコマンドを使用するには、SYSPRV 特権が必要です。

次の表は OpenVMS Registry データベースのファイルを示しています。

ファイル	説明
REGISTRY\$ROOT.DAT	データベースのルート
REGISTRY\$USERS.REG	HKEY_USERSツリー
REGISTRY\$LOCAL_MACHINE.REG	HKEY_LOCAL_MACHINEツリー
REGISTRY\$MASTER.RLG	マスタ・コミット・ログ・ファイル
REGISTRY\$REPLY.RLG	OpenVMS Registry データベースに対する変更要求を追跡するログ・ファイル

---

### フォーマット

CREATE DATABASE

---

### パラメータ

なし

---

### 修飾子

なし

例

```
REG> CREATE DATABASE
```

データベース・ファイルが消失したり，削除されたときに，基本の OpenVMS Registry データベース・ファイルを再作成します。

---

## CREATE KEY

1 つ以上のキーを OpenVMS Registry データベースに作成します。

このコマンドを使用するには、SYSPRV 特権またはREG\$UPDATE ライト識別子が必要です。

---

### フォーマット

```
CREATE KEY key-name [...]
```

---

### パラメータ

*key-name*[,...]

作成するキーの名前を指定します。キーをカンマで区切って指定すると、複数のキーを作成できます。

---

### 修飾子

/WRITEBEHIND

/NOWRITEBEHIND (デフォルト)

キー情報をディスクに直ちに書き込まなければならないことを指定します。

/NOWRITEBEHIND はライトスルー操作を行うことを指定します。

/VOLATILE=*level*

/NONVOLATILE (デフォルト)

新しいキーが揮発性であるかどうかを指定します。OpenVMS Registry をスタンドアロン OpenVMS システムで実行している場合は、システムのリポート時に揮発性キーは失われます。OpenVMS Registry を OpenVMS クラスタで実行している場合は、クラスタ内のすべてのノードがリポートされるときに、揮発性キーは失われます。

*level*の値は次のとおりです。

- NONE (/NONVOLATILE と同じ)
- CLUSTER

/CACHE\_ACTION=*value*

新しいキーのキャッシュ属性を指定します。*value*は WRITEBEHIND (デフォルト) または WRITETHRU (ディスクに直ちに書き込む) のいずれかです。

/CLASS\_NAME=*string*

キーのクラス名を指定します。

```
/SECPOLICY=policy
```

キーのセキュリティ・ポリシーを定義します。現在使用できるポリシーはNT\_40だけです。

```
/LINK=(TYPE=value, NAME=key-name)
```

キーを別のキーへのリンクとして定義します。リンク値は次のいずれかでなければなりません。

- SYMBOLICLINK
- NONE

リンクを削除するには、次のように入力します。

```
/LINK=(TYPE=NONE, NAME=" ")
```

---

## 例

```
REG> CREATE KEY/CACHE_ACTION=WRITEBEHIND HKEY_USERS\GUEST, HKEY_USERS\SYSTEM
```

GUESTキーとSYSTEMキーをHKEY\_USERSエントリ・ポイントの下に作成します。キーはライトバック属性で作成されます。

---

## CREATE SNAPSHOT

OpenVMS Registry データベースのスナップショットを作成します。つまり、キャッシュされているすべての OpenVMS Registry キーまたは値がディスクに書き込まれ、OpenVMS Registry データベース・ファイルのコピーが作成されます。

このコマンドを使用するには、SYSPRV 特権が必要です。

---

### フォーマット

CREATE SNAPSHOT

---

### パラメータ

なし

---

### 修飾子

/DESTINATION=file-spec

スナップショット・ファイルを書き込むかどうかを制御します。デフォルト設定では、スナップショットはSYS\$REGISTRY論理名によって指定される場所に作成されます。

/DESTINATION 修飾子に有効なディレクトリを指定しなかった場合は、スナップショット・ファイルは OpenVMS Registry サーバを起動したディレクトリに作成されます。

/VERSIONS=number

保存するスナップショット・ファイルのバージョン数を指定します。

---

### 例

```
REG> CREATE SNAPSHOT/DESTINATION=SYS$REGISTRY/VERSION=3
```

OpenVMS Registry データベースのスナップショットをSYS\$REGISTRYディレクトリに作成します。OpenVMS Registry データベースのスナップショット・ファイルのバージョン数が3より多くなる場合は、最も古いバージョンが削除されます (purge/keep=3コマンドと同じ)。

---

## CREATE VALUE

指定されたキーのデータ・コンポーネントを指定します。値が存在しない場合は、値が作成されます。

このコマンドを使用するには、SYSPRV 特権またはREG\$UPDATE ライト識別子が必要です。

---

### フォーマット

```
CREATE VALUE key-name
```

---

### パラメータ

*key-name*

値を設定するキーの名前を指定します。

---

### 修飾子

/FLAGS=*flag*

データ・フラグ値を指定します。これはアプリケーションに依存する 64 ビット・フラグであり、10 進数の場合は 0x、16 進数の場合は %X を先頭に付けて指定します。

/WRITEBEHIND

/NOWRITEBEHIND (デフォルト)

値をディスクに直ちに書き込まなければならないことを指定します。

/NOWRITEBEHIND はライトスルー操作を指定します。

/DATA=*value*

値は次のいずれかです。

- 文字列 (たとえば、/DATA=COSMOS)
- カンマで区切り、かっこで囲んだ文字列配列 (たとえば、/DATA=(COSMOS,Noidea)
- 8 進数 (%O)、10 進数、16 進数 (%X) のいずれかの形式のロングワード (たとえば、/DATA=%X1A0FCBまたは/DATA=1234)

/NAME=*string*

新しい値の名前を指定します。

## OpenVMS Registry サーバの管理 CREATE VALUE

/TYPE\_CODE=type

新しい値のタイプを指定します。タイプ値は次のいずれかでなければなりません。

- SZ: 最後が Null の Unicode 文字列
- EXPAND\_SZ: Unicode 文字列
- MULTI\_SZ: SZ 文字列を連結した配列
- DWORD: 32 ビット数値

/LINK=(TYPE=value, NAME=key-name)

キーを別のキーへのリンクとして定義します。リンク値は次のいずれかでなければなりません。

- SYMBOLICLINK
- NONE

リンクを削除するには、次のように入力します。

```
/LINK=(TYPE=NONE,NAME=" ")
```

---

### 例

```
REG> CREATE VALUE/DATA=COSMOS/TYPE=SZ/NAME=COMPUTERNAME HKEY_LOCAL_MACHINE\NODE
```

HKEY\_LOCAL\_MACHINE\NODEキーの値COMPUTERNAMEを作成し、そのタイプをSZに設定し、データ値をCOSMOSに設定します。

---

## DELETE KEY

指定されたキーを OpenVMS Registry データベースから削除します。キーにサブキーがある場合、そのキーは削除されません。

---

### 重要

キーを削除すると、シンボリック・リンクは無効になります。これは、シンボリック・リンクがある場合でも、指定したキーが削除されるからです。

---

### 注意

OpenVMS Registry データベースの定義済みキーは予約キーであり、削除することはできません。このようなキーとしては、`HKEY_USER`、`HKEY_LOCAL_MACHINE`、`HKEY_CLASSES_ROOT`があります。全キーの一覧については、第 7.3 節を参照してください。

このコマンドを使用するには、`SYSPRV` 特権または `REG$UPDATE` ライト識別子が必要です。

---

## フォーマット

```
DELETE KEY key-path key-name
```

---

## パラメータ

`key-path`

キー・パスを指定します。

`key-name`

削除するキーの名前を指定します。

---

## 修飾子

`/WRITEBEHIND`

`/NOWRITEBEHIND` (デフォルト)

キー情報をディスクに直ちに書き込まなければならないことを指定します。

`/NOWRITEBEHIND` はライトスルー操作を指定します。

例

```
REG> DELETE KEY HKEY_USERS\NODE GUEST
```

GUESTというキーを OpenVMS Registry データベースから削除します。

---

## DELETE VALUE

指定されたキーから値を削除します。

---

### 重要

---

値を削除すると、シンボリック・リンクは無効になります。これは、シンボリック・リンクがある場合でも、指定された値が削除されるからです。

---

このコマンドを使用するには、SYSPRV 特権またはREG\$UPDATE ライト識別子が必要です。

---

## フォーマット

```
DELETE VALUE key-name value-name
```

---

## パラメータ

*key-name*

値を削除するキーの名前を指定します。

*value-name*

削除する値を指定します。

---

## 修飾子

/WRITEBEHIND

/NOWRITEBEHIND (デフォルト)

情報をディスクに直ちに書き込まなければならないことを指定します。

/NOWRITEBEHIND はライトスルー操作を指定します。

---

## 例

```
REG> DELETE VALUE HKEY_USERS\GUEST PASSWORD
```

PASSWORD という値を GUEST キーから削除します。

---

## EXPORT

OpenVMS Registry データベースの内容はテキスト形式でエクスポートできます。データベース全体または特定のキーとサブキーだけをエクスポートできます。

エクスポートされるファイルの形式は、Windows NT 互換形式または OpenVMS 形式のいずれかとして指定できます。IMPORT コマンドは Windows NT 4.0 Regedit 形式および OpenVMS Registry 形式の両方をサポートします。

このコマンドを使用するには、REG\$LOOKUP ライト識別子が必要です。REG\$LOOKUP ライト識別子が割り当てられていない場合、REG\$LOOKUP ライト識別子を必要とするキーをエクスポートするには、SYS\$PRV 特権が必要です。

---

### フォーマット

```
EXPORT [DATABASE | KEY [key-name [[/NO]SUBKEYS]]] [/LOG] [/OUTPUT=file-name]
      [/FORMAT=[NT | OPENVMS]]
```

---

### パラメータ

DATABASE

OpenVMS Registry データベース全体をエクスポートします。

KEY [key-name [[/NO]SUBKEYS]]

OpenVMS Registry の特定のキーをエクスポートします。そのキーのサブキーもエクスポートできます。NOSUBKEYS がデフォルトです。

---

### 修飾子

/LOG

エクスポートの進行状況を画面に表示します。

/OUTPUT=file-name

エクスポートされたファイルの名前を指定します。デフォルトの出力ファイル名はREGISTRY.TXTです。

/FORMAT=[NT | OPENVMS]

データベースをファイルに書き込むときの形式を指定します。OPENVMS がデフォルトです。

例

```
REG> EXPORT DATABASE/LOG/OUTPUT=TUES_VERSION.TXT/FORMAT=NT
```

この例のEXPORTコマンドは、OpenVMS Registry データベース全体を Windows NT 4.0 Regedit 形式でTUES\_VERSION.TXTファイルに書き込み、エクスポートの進行状況を画面に表示します。

---

## IMPORT

(EXPORT コマンドで作成された) テキスト形式のファイルを OpenVMS Registry データベースにインポートします。

また、(「レジストリ」メニューの「Export Registry File...」オプションを選択することで) Windows NT 4.0 Regedit によってエクスポートされた Windows NT データも OpenVMS Registry データベースにインポートできます。

---

### Windows NT バイナリ値の変換

---

Windows NT バイナリ値 (構成データなど) は、OpenVMS でバイナリ値がサポートされない場合でも、OpenVMS Registry データベースにインポートできます。サポートされないバイナリ値をインポートし、変換した場合は、そのことを示すメッセージが表示されます。

---

このコマンドを使用するには、REG\$UPDATE ライト識別子が必要です。REG\$UPDATE ライト識別子が割り当てられていない場合、REG\$LOOKUP または REG\$UPDATE ライト識別子を必要とするキーをインポートするには、SYSPRV 特権が必要です。

次の表は、ライト識別子と特権がキーのインポートとエクスポートにどのような影響を与えるかを示しています。

与えられている特権と識別子:	Windows NT からエクスポートできる要素:	OpenVMS Registry にインポートできる要素:
特権なし ライト識別子なし	HKEY_LOCAL_MACHINE\SECURITYを除き、Advanced Server for OpenVMS で作成されたすべてのキー	なし
REG\$LOOKUP	Advanced Server for OpenVMS で作成されたすべてのキー	なし
REG\$UPDATE	Advanced Server for OpenVMS で作成されたすべてのキー	Advanced Server for OpenVMS で作成されたすべてのキー
SYSPRV	Advanced Server for OpenVMS で作成されたすべてのキー	Advanced Server for OpenVMS で作成されたすべてのキー

---

## フォーマット

```
IMPORT [/LOG] [/INPUT=file-name]
```

---

## パラメータ

なし

---

## 修飾子

/LOG

インポートの進行状況を画面に表示します。

/INPUT=file-name

インポートするファイルの名前を指定します。デフォルトの入力ファイル名はREGISTRY.TXTです。

---

## 例

```
REG> IMPORT/LOG/INPUT=TUES_VERSION.TXT
```

この例のIMPORTコマンドは、TUES\_VERSION.TXTファイルをインポートし、インポートの進行状況を画面に表示します。

---

## LIST KEY

指定されたキーの属性を表示します。

---

### 注意

シンボリック・リンクはサポートされていません。

---

このコマンドを使用するには、SYSPRV 特権またはREG\$LOOKUPライト識別子が必要です。

---

## フォーマット

LIST KEY *key-name*

---

## パラメータ

*key-name*

一覧表示するキーの名前を指定します。

---

## 修飾子

/FULL

表示可能なすべての情報、つまり/LAST\_WRITE、/CACHE\_ACTION、/INFORMATION、/LINK\_PATH、/CLASS\_NAME 修飾子で表示される情報を表示します。

/CACHE\_ACTION=*value*

サブキーのキャッシュ属性を指定します。*value*は WRITEBEHIND (デフォルト) または WRITETHRU (ディスクに直ちに書き込む) のいずれかです。

/CLASS\_NAME

サブキーのクラス名を表示します。

/INFORMATION

指定されたキーに関する情報 (サブキー番号、値番号、サブキー名の最大サイズなどを表示します)。

/LAST\_WRITE

サブキーの最終更新日時を表示します。

/LINK\_PATH

サブキーのリンク先のキー・パスを表示します。

/OUTPUT=file-spec

コマンドの出力先を指定します。ファイル名を指定しなかった場合は、デフォルト・ファイル名REGISTRY.LISが使用されます。

---

例

```

REG> LIST KEY/FULL HKEY_USERS\GUEST

Key name:           HKEY_USERS\GUEST
Security policy:    REG$K_POLICY_NT_40
Volatile:           REG$K_NONE
Cache:              REG$K_WRITEBEHIND
Class:              System Authorization
Link Type:          REG$K_NONE
Last written:       7-AUG-1998 12:42:08.55

Key information:
  Number of subkeys:      2      Number of values:      0
  Max size of subkey name: 40      Max size of class name: 40
  Max size of value name: 0      Max size of value data: 0

Subkey(s):

  Key name:           QUOTAS
  Security policy:    REG$K_POLICY_NT_40
  Volatile:           REG$K_NONE
  Cache:              REG$K_WRITEBEHIND
  Class:              Disk quota
  Link Type:          REG$K_NONE
  Last written:       7-AUG-1998 12:41:19.21

  Key information:
  Number of subkeys:      0      Number of values:
0
  Max size of subkey name: 0      Max size of class name:
0
  Max size of value name: 0      Max size of value data:
0

  Key name:           IDENTIFIER
  Security policy:    REG$K_POLICY_NT_40
  Volatile:           REG$K_NONE
  Cache:              REG$K_WRITETHRU
  Class:              Disk quota
  Link Type:          REG$K_SYMBOLICLINK
  Link Path:          HKEY_LOCAL_MACHINE\SOFTWARE\IDENTIFIER\GUEST
  Last written:       7-AUG-1998 12:42:08.55

```

OpenVMS Registry サーバの管理  
LIST KEY

```
Key information:
  Number of subkeys:      0      Number of values:
0
  Max size of subkey name: 0      Max size of class name:
0
  Max size of value name: 0      Max size of value data:
0
```

この例のLIST KEY/FULLコマンドは、GUESTキーの属性と、GUESTのサブキーの名前と属性を表示します。

---

注意

Max sizes情報には、文字数ではなくバイト数が表示されます (各文字は4バイト長です)。

---

---

## LIST VALUE

指定されたキーのすべての値と値の属性を表示します。

---

### 注意

---

シンボリック・リンクはサポートされていません。

---

このコマンドを使用するには、SYSPRV 特権またはREG\$LOOKUPライト識別子が必要です。

---

## フォーマット

LIST VALUE *key-name*

---

## パラメータ

*key-name*

列挙するキーの名前を指定します。

---

## 修飾子

/FULL

表示可能なすべての情報、つまり/TYPE\_CODE、/LINK\_PATH、/DATA\_FLAGS、/VALUE\_DATA 修飾子で表示される情報を表示します。

/TYPE\_CODE

値のタイプ・コードを表示します。

/FLAGS

値のデータ・フラグの ASCII 表現を 16 進数で表示します。

/LINK\_PATH

サブキーのリンク先のキー・パスを表示します。

/DATA

値の ASCII 表現を 16 進数で表示します。

/OUTPUT=file-spec

コマンドの出力先を指定します。ファイル名を指定しなかった場合は、デフォルト・ファイル名REGISTRY.LISが使用されます。

例

```
REG> LIST VALUE/TYPE_CODE/DATA HKEY_LOCAL_MACHINE\SOFTWARE\FORTRAN
Key name:          HKEY_LOCAL_MACHINE\SOFTWARE\FORTRAN
Security policy:   REG$K_POLICY_NT_40
Volatile:          REG$K_NONE
Last written:      11-AUG-1998 16:27:55.81

Value(s):
Value name:  Version
Volatile:    REG$K_NONE
Type:        REG$K_SZ
Data:        5.3-50

Value name:  Date Installed
Volatile:    REG$K_NONE
Type:        REG$K_SZ
Data:        04-Jan-1998
```

この例のLIST VALUE/TYPE\_CODE/DATAコマンドは、FORTRANキーとその値の名前、タイプ、データを表示します。

---

## MODIFY KEY

指定されたキーの属性を変更します。

---

### 重要

---

キーを変更すると、シンボリック・リンクは無効になります。これは、シンボリック・リンクによって示されるキーではなく、指定されたキーが変更されるからです。

---

このコマンドを使用するには、SYSPRV 特権またはREG\$UPDATEライト識別子が必要です。

---

### フォーマット

MODIFY KEY *key-name*

---

### パラメータ

*key-name*

変更するキーの名前を指定します。

---

### 修飾子

/CACHE\_ACTION=*value*

新しいキーのキャッシュ属性を指定します。 *value* は WRITEBEHIND (デフォルト) または WRITETHRU (ディスクに直ちに書き込む) のいずれかです。

/CLASS\_NAME=*string*

キーの新しいクラス名を指定します。

/WRITEBEHIND

/NOWRITEBEHIND (デフォルト)

キー情報をディスクに直ちに書き込まなければならないことを指定します。

/NOWRITEBEHIND はライトスルー操作を指定します。

/NEW\_NAME=*new-key-name*

キーの新しい名前を指定します。

/SECPOLICY=*policy*

キーのセキュリティ・ポリシーを定義します。現在指定できるポリシーはNT\_40だけです。

OpenVMS Registry サーバの管理  
MODIFY KEY

/LINK=(TYPE=value, NAME=key-name)

キーを別のキーへのリンクとして定義します。リンク値は次のいずれかでなければなりません。

- SYMBOLICLINK
- NONE

リンクを削除するには、次のように入力します。

/LINK=(TYPE=NONE, NAME=" ")

---

例

```
REG> MODIFY KEY/CACHE_ACTION=WRITEBEHIND  
HKEY_USERS\GUEST
```

GUESTキーのキャッシュ属性を変更します。

---

## MODIFY VALUE

指定された値のデータ・コンポーネントを指定します。このコマンドは既存の値を変更します。

---

### 重要

---

値を変更すると、シンボリック・リンクは無効になります。これは、シンボリック・リンクによって示される値ではなく、指定された値が変更されるからです。

---

このコマンドを使用するには、SYSPRV 特権またはREG\$UPDATEライト識別子が必要です。

---

## フォーマット

```
MODIFY VALUE /NAME=string key-name
```

---

## パラメータ

*key-name*

値を設定するキーの名前を指定します。

---

## 修飾子

/FLAGS=*flag*

データ・フラグの値を指定します。これはアプリケーションに依存する 64 ビットのフラグであり、0x を先頭に付けた 10 進数、または%X を先頭に付けた 16 進数として指定します。

/WRITEBEHIND

/NOWRITEBEHIND (デフォルト)

値情報をディスクに直ちに書き込まなければならないことを指定します。

/NOWRITEBEHIND はライトスルー操作を指定します。

/DATA=*value*

値のデータを指定します。値は次のいずれかです。

- 文字列 (たとえば、/DATA=COSMOS)
- カンマで区切り、括弧で囲んだ文字列配列 (たとえば、/DATA=(COSMOS,Noidea)

## OpenVMS Registry サーバの管理 MODIFY VALUE

- 8進数 (%O) , 10進数 , 16進数 (%X) のいずれかの形式のロングワード (たとえば, /DATA=%X1A0FCBまたは/DATA=1234)

/NAME=string

値の名前を指定します。

/TYPE\_CODE=type

新しい値のタイプを指定します。タイプは次のいずれかでなければなりません。

- SZ: 最後が Null の Unicode 文字列
- EXPAND\_SZ: Unicode 文字列
- MULTI\_SZ: SZ 文字列を連結した配列
- DWORD: 32 ビット数値

/LINK=(TYPE=value, NAME=key-name)

キーを別のキーへのリンクとして定義します。リンク値は次のいずれかでなければなりません。

- SYMBOLICLINK
- NONE

リンクを削除するには、次のように入力します。

/LINK=(TYPE=NONE,NAME=" ")

---

### 例

```
REG> MODIFY VALUE/DATA=COSMOS/TYPE=SZ/NAME=COMPUTERNAME HKEY_LOCAL_MACHINE\NODE
```

キーHKEY\_LOCAL\_MACHINE\NODEのCOMPUTERNAMEという値を作成し、タイプ・コードをSZに、データ値をCOSMOSに設定します。

---

## MODIFY TREE

指定されたキーとそのサブキーの情報を変更します。

---

### 重要

---

ツリーを変更すると、シンボリック・リンクは無効になります。これは、シンボリック・リンクによって示されるキーではなく、指定されたキーとサブキーが変更されるからです。

---

このコマンドを使用するには、SYSPRV 特権またはREG\$UPDATE ライト識別子が必要です。

---

## フォーマット

```
MODIFY TREE key-name
```

---

## パラメータ

*key-name*

変更するキーの名前を指定します。

---

## 修飾子

`/CACHE_ACTION=value`

キーとそのサブキーのキャッシュ属性を指定します。 *value* は WRITEBEHIND (デフォルト) または WRITETHRU (ディスクに直ちに書き込む) のいずれかです。

`/CLASS_NAME=string`

指定されたキーとそのすべてのサブキーの新しいクラス名を指定します。

`/WRITEBEHIND`

`/NOWRITEBEHIND` (デフォルト)

キー情報をディスクに直ちに書き込まなければならないことを指定します。

`/NOWRITEBEHIND` はライトスルー操作を指定します。

`/SECPOLICY=policy`

キーのセキュリティ・ポリシーを定義します。現在指定できるポリシーは NT\_40 だけです。

例

```
REG> MODIFY TREE/CACHE_ACTION=WRITEBEHIND HKEY_USERS\GUEST
```

GUESTキーとそのすべてのサブキーのキャッシュ属性を変更します。

---

## SEARCH KEY

指定されたキーと一致するすべてのキーのパス名を表示します。

このコマンドを使用するには、SYSPRV 特権またはREG\$LOOKUPライト識別子が必要です。

---

### フォーマット

SEARCH KEY *key-search*

---

### パラメータ

*key-search*

検索するキーの名前を指定します。

---

### 修飾子

/OUTPUT=*file-spec*

コマンドの出力先を指定します。ファイル名を指定しなかった場合は、デフォルト・ファイル名REGISTRY.LISが使用されます。

---

### 例

```
REG> SEARCH KEY HKEY_LOCAL_MACHINE\...\NODE
HARDWARE\CLUSTER\NODE
HARDWARE\LOCAL\NODE
NODE
```

HKEY\_LOCAL\_MACHINE\...\NODEと一致するすべてのキー・パスが表示されます。繰り返し記号 (...) ワイルドカードは、HKEY\_LOCAL\_MACHINEエントリ・ポイントとNODEサブキーの間にサブキーがいくつあってもかまわないことを指定します。検索で大文字と小文字は区別されません。

---

## SEARCH VALUE

指定された値名と一致するすべての値のパス名を表示します。

このコマンドを使用するには、SYSPRV 特権またはREG\$LOOKUPライト識別子が必要です。

---

### フォーマット

```
SEARCH VALUE key-name value-name
```

---

### パラメータ

*key-name*

検索するキー・パスの名前を指定します。

*value-name*

検索する値の名前を指定します。

---

### 修飾子

*/OUTPUT=file-spec*

コマンドの出力先を指定します。ファイル名を指定しなかった場合は、デフォルトのファイル名REGISTRY.LISが使用されます。

---

### 例

```
REG> SEARCH VALUE HKEY_LOCAL_MACHINE\... *AM%  
HARDWARE\CLUSTER\Name  
HARDWARE\CLUSTER\NODE\Name  
HARDWARE\LOCAL\NODE\Name  
NODE\COMPUTERNAME
```

HKEY\_LOCAL\_MACHINE\... \*am%と一致するすべての値の名前が表示されます。繰り返し記号 (...) ワイルドカードは、HKEY\_LOCAL\_MACHINE エントリ・ポイントと \*am% 値名の間サブキーがいくつあってもかまわないことを指定します。検索で大文字と小文字は区別されません。

---

## SHOW

OpenVMS Registry サーバの内部統計情報とその他の情報を表示します。

- SHOW COUNTERS

OpenVMS Registry サーバからの監視情報を表示します。

- SHOW FILE

OpenVMS Registry サーバにロードされたファイルの状態情報を表示します。

このコマンドを使用するには、SYSPRV 特権またはREG\$PERFORMANCEライト識別子が必要です。

---

### フォーマット

SHOW COUNTERS/FILE *[name]*

SHOW FILE *[name]*

---

### パラメータ

name

ファイルを識別します (/FILE 修飾子と組み合わせて使用しなければなりません)。

---

### 修飾子

/FILE

指定されたファイルまたはすべてのファイルのカウンタを表示します。

/PERFORMANCE

パフォーマンス・カウンタを表示します。

/OUTPUT=file-spec

コマンドの出力先を指定します。ファイル名を指定しなかった場合は、デフォルトのファイル名REGISTRY.LISが使用されます。

---

### 例

```
REG> SHOW COUNTERS/FILE
```

OpenVMS Registry サーバからの監視情報を表示します。

---

## START MONITORING

OpenVMS Registry サーバ内で監視コンポーネントを起動します。

このコマンドを使用するには、SYSPRV 特権またはREG\$PERFORMANCE ライト識別子が必要です。

---

### フォーマット

```
START MONITORING/FILE [name]  
START MONITORING/PERFORMANCE
```

---

### パラメータ

name  
ファイルを指定します (/FILE 修飾子と組み合わせて使用しなければなりません)。

---

### 修飾子

```
/FILE  
指定されたファイルまたはすべてのファイルのカウンタ情報の収集を開始します。  
  
/PERFORMANCE  
パフォーマンス・カウンタの情報収集を開始します。
```

---

### 例

```
REG> START MONITORING/PERFORMANCE
```

OpenVMS Registry の監視コンポーネントを有効にします。

---

# STOP

OpenVMS Registry サーバ内で監視コンポーネントを停止します。

このコマンドは OpenVMS Registry サーバ内で監視コンポーネントを停止するときに使用します。

このコマンドを使用するには、SYSPRV 特権またはREG\$PERFORMANCE ライト識別子が必要です。

---

## フォーマット

```
STOP MONITORING/FILE [name]
STOP MONITORING/PERFORMANCE
```

---

## パラメータ

name  
ファイルを指定します (/FILE 修飾子と組み合わせて使用しなければなりません)。

---

## 修飾子

/FILE  
指定されたファイルまたはすべてのファイルのカウンタ情報の収集を停止します。

/PERFORMANCE  
パフォーマンス・カウンタ情報の収集を停止します。

---

## 例

```
REG> STOP MONITORING/PERFORMANCE
```

OpenVMS Registry の監視コンポーネントを無効にします。

---

## ZERO COUNTERS

OpenVMS Registry サーバのカウンタを初期化します。

このコマンドを使用するには、SYSPRV 特権またはREG\$PERFORMANCEライト識別子が必要です。

---

### フォーマット

ZERO COUNTERS/FILE *[name]*  
ZERO COUNTERS/PERFORMANCE

---

### パラメータ

name  
ファイルを指定します (/FILE 修飾子と組み合わせて使用しなければなりません)。

---

### 修飾子

/FILE  
指定されたファイルまたはすべてのファイルのファイル・カウンタを初期化します。  
  
/PERFORMANCE  
すべてのパフォーマンス・カウンタを初期化します。

---

### 例

```
REG> ZERO COUNTERS/PERFORMANCE
```

パフォーマンス・カウンタをリセットします。

---

## OpenVMS Registry システム・サービス

この章では、OpenVMS Registry の\$REGISTRY システム・サービスと\$REGISTRYW システム・サービスについて説明します。

---

## \$REGISTRY と \$REGISTRYW

### OpenVMS Registry データベースへのインタフェース

\$REGISTRY システム・サービスと \$REGISTRYW システム・サービスは、OpenVMS Registry データベース・サーバへのインタフェースです。\$REGISTRY サービスを使用すると、OpenVMS Registry データベースに登録されているキー、サブキー、値の表示、更新、設定を行うことができます。

\$REGISTRY サービスでは、同期操作と非同期操作の両方がサポートされます。非同期完了の場合は、Registry (\$REGISTRY) システム・サービスを使用します。同期完了の場合は、Registry and Wait (\$REGISTRYW) システム・サービスを使用します。\$REGISTRYW システム・サービスは \$REGISTRY システム・サービスとほとんど同じ機能を実行しますが、要求された操作が完了した後で呼び出しプロセスに制御が返されます。システム・サービスの終了の詳細については、Synchronize (\$SYNCH) システム・サービスを参照してください。

このシステム・サービスは 64 ビット互換です。

---

### フォーマット

`SY$REGISTRY [efn], func, 0, itmlst, [iosb or ios_a_64], [astadr or astadr_64], [astprm or astprm_64]`

`SY$REGISTRYW [efn], func, 0, itmlst, [iosb or ios_a_64], [astadr or astadr_64], [astprm or astprm_64]`

---

### 引数

efn

OpenVMS usage: ef\_number  
type: longword (unsigned)  
access: read only  
mechanism: by value

\$REGISTRY で使用されるイベント・フラグの数。イベント・フラグを指定しなかった場合、デフォルトは `EFN$C_ENF` になります。イベント・フラグは \$REGISTRY で最初にクリアされ、操作が完了したときにセットされます。

func

OpenVMS usage: function\_code  
type: longword (unsigned)  
access: read only  
mechanism: by value

\$REGISTRY が実行する動作を指定する機能コード。func引数は、この機能コードを格納したロングワードです。機能コードには機能修飾子を含むことができます。機能修飾子の詳細については、この章の機能修飾子を参照してください。

\$REGISTRY の呼び出しには機能コードを 1 つだけ指定できます。どの機能コードを呼び出す場合も、itmlst引数に追加情報を指定しなければなりません。

itmlst

OpenVMS usage: item\_list3 or item\_list\_64b  
type: longword (unsigned)  
access: read only  
mechanism: by reference

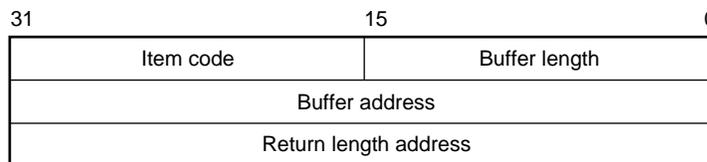
func引数によって指定される機能を実行するために、システムが使用する情報を示すアイテム・リスト。itmlst引数はアイテム・リストのアドレスです。アイテム・リストは 1 組以上のアイテム記述子で構成されます。各記述子は item-list-3 または item-list-64b 形式です。

一部の機能コードでは、1 回の呼び出しに複数の操作を指定できます。この場合、各アイテム・コード・セットの間に REG\$\_SEPARATOR アイテム・コードを指定しなければなりません。REG\$\_SEPARATOR アイテム・コードで区切られた各要求にアイテム・コードを指定する順序は任意です。

アイテム・コードは入力パラメータまたは出力パラメータとして指定できます。入力パラメータは機能の変更、コンテキストの設定、返す情報の指定を行います。出力パラメータは要求された情報を返します。item-list-3 リストの場合は、リストの最後にロングワードの 0 を指定しなければなりません。item-list-64b リストの場合は、リストの最後にクォドワードの 0 を指定しなければなりません。

図 10-1 は item-list-3 記述子の構造を示しています。

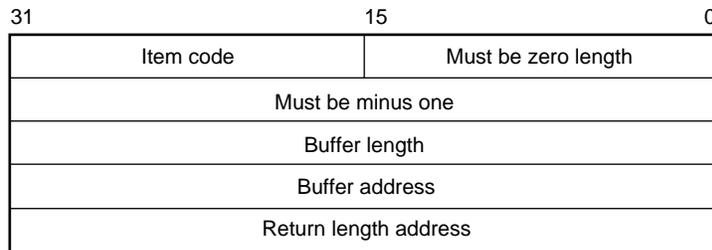
図 10-1 item-list-3 の構造



VM-0224A-AI

図 10-2 は item-list-64b 記述子の構造を示しています。

図 10-2 item-list-64b の構造



VM-0225A-AI

表 10-1 はアイテム記述子フィールドの定義を示しています。

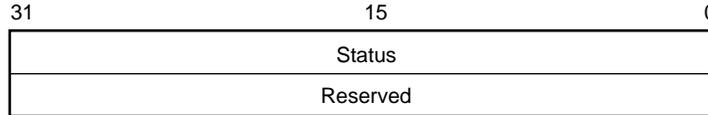
表 10-1 アイテム記述子フィールド

記述子フィールド	定義
Buffer length	バッファの長さを指定するワード。バッファは、\$REGISTRY で使用される情報を渡すか、または \$REGISTRY から情報を受取る場所である。必要なバッファの長さは、指定するアイテム・コードに応じて異なる。各アイテム・コード記述に必要な長さを指定する。
Item code	バッファ内に現在格納されている情報のタイプを示すシンボリック・コード、またはバッファに返されたシンボリック・コードを格納したワード。buffer address フィールドはバッファの場所を示す。
Buffer address	情報の受け渡しで使用されるバッファのアドレスが格納されているロングワード。
Return length address	\$REGISTRY から返された情報の実際の長さ (バイト数) を指定するワードのアドレスが格納されているロングワード。情報は buffer address フィールドで示されるバッファに格納される。このフィールドは出力アイテム・リスト・エントリにだけ適用され、入力エントリの場合は 0 でなければならない。return length address が 0 の場合は無視される。

iosb or ios\_a\_64

OpenVMS usage: status\_block  
type: buffer  
access: write only  
mechanism: by reference

\$REGISTRY 操作の最終終了状態と情報が返される状態ブロック。1つの機能コードに対して複数の操作が要求された場合は、iosbに返される値はSS\$\_NORMALまたはSS\$\_REGERRORになります。各操作の戻り状態はREG\$\_RETURNSTATUSアイテム・コード (指定されている場合) に返されます。iosb引数は\$REGISTRYの状態ブロックのアドレスです。



VM-0226A-AI

iosb引数を指定した場合は、\$REGISTRY の実行が開始されるときに、クオードワードの I/O 状態ブロックがクリアされます。

iosb引数は省略してもかまいませんが、次の理由により、なるべく指定してください。

- サービスの終了を通知するためにイベント・フラグを使用する場合は、I/O 状態ブロックの条件値を判定することで、イベント・フラグがサービス正常終了以外のイベントによってセットされなかったことを確認できます。
- サービスの終了の同期をとるために\$SYNCHシステム・サービスを使用する場合は、I/O 状態ブロックは\$SYNCHの必要な引数になります。

R0 に返される条件値と、I/O 状態ブロックに返される条件値は、\$REGISTRY サービスへの呼び出しに関して異なる情報を提供します。R0 に返される条件値は、サービスの呼び出し自体が正常終了したのか、異常終了したのかに関する情報を示します。I/O 状態ブロックに返される条件値は、サービス操作が正常終了したのか、異常終了したのかに関する情報を示します。

\$REGISTRY への呼び出しが正常終了したのか、異常終了したのかを正確に判断するには、最初に R0 に返された条件値を確認しなければなりません。R0 に正常終了を示す値が格納されている場合は、I/O 状態ブロックの条件値を確認しなければなりません。

表 10-2 はアイテム記述子フィールドの定義を示しています。

表 10-2 記述子フィールド

記述子フィールド	定義
Status	\$REGISTRY サービスの最終状態を指定するロングワード。1つの機能コードに対して複数の操作を要求した場合は、SS\$_NORMALまたはSS\$_REGERRORがiosbに返される。このフィールドは操作を開始するときに、0に設定される。
Reserved	予約されているロングワード。

astadr or astadr\_64

OpenVMS usage: ast\_procedure  
 type: procedure value  
 access: call without stack unwinding  
 mechanism: by reference

\$REGISTRY が完了したときに実行される AST サービス・ルーチン。astadr 引数はこのルーチンのアドレスです。astadr を指定した場合、AST ルーチンは \$REGISTRY サービスの呼び出し側プロセスと同じアクセス・モードで実行されます。

\$REGISTRY サービスの呼び出しが正常終了しなかった場合 (つまり、エラーが直ちに戻された場合) は、AST ルーチンは実行されません。

astprm or astprm\_64

OpenVMS usage: user\_arg  
type: longword (unsigned)  
access: read only  
mechanism: by value

astadr 引数によって指定された AST サービス・ルーチンに渡される AST パラメータ。astprm 引数はこのロングワード・パラメータを指定します。

---

## 説明

\$REGISTRY サービスは、レジストリ・キー、キーの値、キーの属性の作成、削除、変更を行う機能を提供します。

\$REGISTRY サービスでは、キーに対するハンドルを格納するためにプロセス P1 空間を使用します。\$REGISTRY サービスは IPL 0 で呼び出さなければならず、AST 要求を配布するためにシステム・ダイナミック・メモリを必要とします。

関連サービス

\$REGISTRYW

---

## 返される状態値

SS\$_ACCVIO	引数の 1 つを読み込みまたは書き込みできません。
SS\$_BADPARAM	機能コードまたはアイテム・リスト・コードの 1 つが不正です。
SS\$_EXASTLM	AST リミットを超えました。
SS\$_EXBYTLM	バイト・カウント・クォータを超えました。
SS\$_ILLEFC	イベント・フラグ番号が不正です。
SS\$_INSFARG	指定された引数の数が不足しています。
SS\$_INSFMEM	ダイナミック・メモリが不足しています。
SS\$_NORMAL	正常終了。
SS\$_TOO_MANY_ARGS	引数の数が多すぎます。
REG\$_ACCESSDENIED	キーに対して要求されたアクセスが拒否されました。
REG\$_BADFILEVER	ファイル・バージョン番号が不正です。

REG\$_BUFFEROVF	バッファ・オーバフロー。
REG\$_CANTCLEANVOLSEG	レジストリの揮発性セグメントを消去できません。
REG\$_CANTCONVCS	コード・セット変換エラー。
REG\$_CANTOPENOUTFILE	指定された出力ファイルを開くことができません。
REG\$_DBACCESS	レジストリ・データベース・オブジェクトにアクセスできません。
REG\$_DBALREADYLOADED	データベースはすでにロードされています。
REG\$_DBCREATE	レジストリ・データベースを作成できません。
REG\$_DBCSMISMATCH	データベース・チェックサムの不一致: 格納値= !XL 計算値= !XL
REG\$_DBFIND	レジストリ・データベースを見つけることができません。
REG\$_DBFULL	レジストリ・データベースが満杯です。
REG\$_DBLOAD	レジストリ・データベースをロードできません。
REG\$_DBNOTYETLOADED	データベースはまだロードされていません。
REG\$_DBVERMISMATCH	データベース・バージョンの不一致: 現在のバージョン= !UW.!UW データベース= !UW.!UW
REG\$_DELROOTKEY	ルート・キーが削除されました。
REG\$_DOUBLEDEALLOC	構造はすでにフリー・リストに登録されています。
REG\$_DTMUTEXERROR	DECthreads ミューテックスロック/アンロック・エラー。
REG\$_DTMUTEXINIT	DECthreads ミューテックス初期化エラー !UL
REG\$_DTMUTEXLOCK	DECthreads ミューテックス・ロック・エラー !UL
REG\$_DTMUTEXLOCKED	DECthreads ミューテックスは別のスレッドによってすでに保有されています。
REG\$_DTMUTEXUNLOCK	DECthreads ミューテックス・アンロック・エラー !UL
REG\$_DTRWLOCKINIT	DECthreads 読み込み/書き込みロック初期化エラー !UL
REG\$_DTRWLOCKLOCK	DECthreads 読み込み/書き込みロック・エラー !UL
REG\$_DTRWLOCKUNLOCK	DECthreads 読み込み/書き込みアンロック・エラー !UL
REG\$_DUPLREQUEST	作業中のハッシュ・テーブル挿入で重複する要求が検出されました。
REG\$_EXQUOTA	レジストリ・ファイル・クォータまたはページ・ファイル・クォータを超えました。
REG\$_FILECREATE	!AZ!AZ の作成でエラーが発生しました。
REG\$_FILENAMEINVAL	ファイル名が不正です。
REG\$_FILEOPEN	!AZ!AZ を開くときにエラーが発生しました。
REG\$_FILEREADEOF	ファイルの末尾をこえて読み込もうとしました。 FTE !XL
REG\$_FNAMMISMATCH	物理/論理ファイル名の不一致; FTE=!AZ LTE=!AZ
REG\$_FSOCORRUPT	ファイルにはすでに破損フラグが付いています。 FSO: !XL !XL
REG\$_FSOFILEINDEX	FSO: !XL !XL のファイル・インデックスが不正です。
REG\$_FSOOFFSET	FSO: !XL !XL のオフセットが不正です。
REG\$_FSOSEGNUMBER	FSO: !XL !XL のセグメント番号が不正です。
REG\$_FSOSEGREADERR	FSO: !XL !XL でセグメントの読み込みエラーが発生しました。

OpenVMS Registry システム・サービス  
\$REGISTRY と \$REGISTRYW

REG\$_FTEALLOC	ファイル・テーブル・エントリ !XL を !AZ に対して割り当てるときにエラーが発生しました。
REG\$_FTEALREADYEXIST	ファイル !AZ!AZ を作成できません。ファイルはすでに存在します。
REG\$_FTEALREADYOPEN	ファイルはすでに開かれています。
REG\$_FTEDUPNAME	ファイル・テーブル・エントリの割り当てエラー; ファイル名が重複します。
REG\$_FTEINSUFFINFO	指定されたファイル・テーブル・エントリは割り当てられていません。
REG\$_FTEINUSE	ファイル・テーブル・エントリの割り当てエラー, エントリは使用中です。
REG\$_FTENOTEXIST	指定されたファイル・テーブル・エントリは存在しません。
REG\$_FTENOTOPEN	指定されたファイルは開かれていません。
REG\$_FTMISMATCH	物理ファイル・インデックスの不一致; LTE = !XL, FTE = !XL
REG\$_HASLINK	キーに別のキーへのリンクがあります。
REG\$_HAVESUBKEYS	サブキーのあるキーを削除することはできません。
REG\$_INTERNERR	レジストリの内部エラー。
REG\$_INVCACHEACTION	キャッシュ・アクション・パラメータが不正です。
REG\$_INVCREDENTIALS	NT 資格情報が不正です。
REG\$_INVDATA	データ値が不正です。
REG\$_INVDATATYPE	データ・タイプ・パラメータが不正です。
REG\$_INVFUNCCODE	機能コードが不正です。
REG\$_INVKEYFLAGS	キー・フラグが不正です。
REG\$_INVKEYID	キーが存在しないか、または不正なキー ID が指定されました。
REG\$_INVKEYNAME	キー名が不正です。
REG\$_INVLINK	リンクまたはリンク・タイプが不正です。
REG\$_INVLINKPATH	リンク・パスが不正です。
REG\$_INVLOG	ログ・ファイルが不正です。
REG\$_INVLOGREC	ログ・レコードが不正です。
REG\$_INVPARAM	パラメータが不正です。
REG\$_INVPATH	キー・パスが不正です。
REG\$_INVSECDESCRIPTOR	セキュリティ記述子が不正です。
REG\$_INVSECPOLICY	セキュリティ・ポリシー・パラメータが不正です。
REG\$_INVSEGNUM	セグメント番号が不正です。
REG\$_INVVOLROOTKEY	揮発性ルート・キーを使用して新しいファイルを作成することはできません。
REG\$_IOREADERR	ブロック !UL で長さ !UL のディスク読み込みエラーが発生しました。
REG\$_IOWRITERR	ブロック !UL で長さ !UL のディスク書き込みエラーが発生しました。
REG\$_IPCCONACC	IPC 接続受諾エラー: !XL
REG\$_IPCCONREJ	IPC 接続拒否エラー: !XL
REG\$_IPCDCLAST	IPC は同期完了のために AST を宣言できません。 !XL

REG\$_IPCOPEASS	IPC 関連付けオープン・エラー: !XL
REG\$_IPLTOOHIGH	IPL 0 より上の呼び出しプロセスはこのサービス呼び出すことができません。
REG\$_KEYCHANGED	キーまたはサブキーが変更されています。
REG\$_KEYEXIST	キーはすでに存在します。
REG\$_KEYLOCKED	キーは別のスレッドによってロックされています。
REG\$_KEYNAMEEXIST	キー名はすでに存在します。
REG\$_LOGFILETABFULL	論理ファイル・テーブルが満杯です。
REG\$_LTENOTEXIST	指定された論理ファイル・テーブル・エントリは存在しません。
REG\$_MOREDATA	提供されたバッファが小さすぎるため、要求されたデータを格納できません。
REG\$_NOBLOCKFOUND	レジストリ・データベースに使用できるブロックが残っていません。
REG\$_NOKEY	指定されたキーは存在しません。
REG\$_NOMEMORY	メモリ不足。
REG\$_NOMOREITEMS	指定されたキーに対するアイテムはこれ以上ありません。
REG\$_NOMORESEG	使用できるセグメントはこれ以上ありません。
REG\$_NOMORESUBSTRING	部分文字列を見つけることができません。
REG\$_NOPATHFOUND	パスを見つけることができません。
REG\$_NORESPONSE	OpenVMS Registry サーバは使用できない状態です。
REG\$_NOSUCHFILE	このようなファイルは存在しません。
REG\$_NOTROOTKEY	ルート・キー・インデックスが不正です。
REG\$_NOTSUPPORTED	機能コード、アイテム・コード、アイテム値のいずれかがサポートされません。
REG\$_NOVALUE	指定された値は存在しません。
REG\$_OBJWITHLINK	削除されたキーまたは値を指すリンクがあります。
REG\$_REQRECEIVED	キー変更要求の通知を受け取りました。
REG\$_RESERVED	予約キーまたは値を削除/変更することはできません。
REG\$_ROOTINSFILE	ルート・ファイル内のファイル・リストが不足しています。
REG\$_RUIDMISMATCH	ルート・キー UID の不一致; LTE = !@XQ !@XQ; ルート・キー = !@XQ !@XQ
REG\$_SECVIO	このキーが最後に開かれたときに指定されたセキュリティ・アクセス方式に違反します。
REG\$_SEGREADERR	ファイル !AZ のセグメント !UL の読み込みでエラーが発生しました。
REG\$_STRINGTOOLONG	入力文字列が長すぎます。
REG\$_STRINGTRUNC	出力バッファが小さすぎるため、変換された文字列を格納できません。
REG\$_SVRVERMISMATCH	バージョンの不一致: サーバ= V!UL.!UL データベース= V!UL.!UL
REG\$_SVRSHUTDOWN	サーバのシャットダウン中です。
REG\$_TOOMANYOPENKEY	開かれているキーの数が多すぎます。一部のキーを閉じてください。

REG\$_UNKTHREQ	スレッド要求コードが不明です。
REG\$_VALUEEXIST	値はすでに存在します。
REG\$_VOLMISMATCH	揮発性キーに対して不揮発性サブキーを作成することはできません。

## 機能コード

表 10-3 は、機能コードと各コードの簡単な説明、その機能を実行するのに必要な OpenVMS Registry ライト識別子を示しています。各アイテム・コードの詳細については、この章のアイテム・コードを参照してください。

有効な NT アクセス・トークンを提供せず、SYSPRV 特権もない場合は、OpenVMS Registry ライト識別子が必要です。REG\$UPDATE 識別子が与えられている場合は、表 10-3 に示されているすべての機能を実行できます。

表 10-3 有効な機能コード

機能コード	識別子	説明
REG\$FC_CLOSE_KEY	REG\$LOOKUP	開かれているキーまたはサブキーを閉じる。
REG\$FC_CREATE_KEY	REG\$UPDATE	サブキーを作成して開く。
REG\$FC_DELETE_KEY	REG\$UPDATE	サブキーをキーから削除する。
REG\$FC_DELETE_VALUE	REG\$UPDATE	値をキーから削除する。
REG\$FC_ENUM_KEY	REG\$LOOKUP	キーのサブキーを一覧表示する。
REG\$FC_ENUM_VALUE	REG\$LOOKUP	キーの値を一覧表示する。
REG\$FC_FLUSH_KEY	REG\$UPDATE	キーのすべての情報をディスクにバックアップする。
REG\$FC_MODIFY_KEY	REG\$UPDATE	キーを変更する。
REG\$FC_MODIFY_TREE_KEY	REG\$UPDATE	キーとそのすべてのサブキーを変更する。
REG\$FC_NOTIFY_CHANGE_KEY_VALUE	REG\$UPDATE	キーまたは値が変更されたときに、そのことを通知する。
REG\$FC_OPEN_KEY	REG\$LOOKUP	キーまたはサブキーを開く。
REG\$FC_QUERY_KEY	REG\$LOOKUP	キーに関する情報を取り込む。
REG\$FC_QUERY_VALUE	REG\$LOOKUP	値に関する情報を取り込む。
REG\$FC_SEARCH_TREE_DATA	REG\$LOOKUP	キーとそのサブキーの値データを検索する。
REG\$FC_SEARCH_TREE_KEY	REG\$LOOKUP	キーとそのサブキーの名前を検索する。
REG\$FC_SEARCH_TREE_VALUE	REG\$LOOKUP	キーとそのサブキーの値を検索する。
REG\$FC_SET_VALUE	REG\$UPDATE	値名に関連付けられているデータを変更する。

### REG\$FC\_CLOSE\_KEY

この要求は指定されたキーの開かれているリソースを解放します。REG\$\_KEYID が定義済みキーを示す場合は、システムはこの要求を無視し、正常終了を返します。

次のいずれかのアイテム・コードを指定します。

アイテム・コード	必須	パラメータの種類
REG\$_KEYID	必須	入力
REG\$_RETURNSTATUS	省略可能	出力

#### REG\$FC\_CREATE\_KEY

キーが存在しない場合、この要求はREG\$\_KEYIDによって指定されるキーの下に新しいサブキーを作成します。キーが存在する場合、そのキーは変更されません。

REG\$\_KEYRESULTアイテム・コードを指定した場合は、指定したサブキーが開かれます。

結果はREG\$\_DISPOSITIONアイテム・コード・バッファに返されます。

この機能コードを使用すると、複数の要求を\$REGISTRY サービスに対する1つの呼び出しにまとめて指定できます。この複数要求機能を使用するには、REG\$\_SEPARATORアイテム・コードを使用して、現在の要求のアイテム・コード・セットの最後を示し、その後に別の要求が続くことを指定しなければなりません。

キーの値を設定するには、REG\$FC\_SET\_VALUE機能コードを指定して\$REGISTRY サービスを呼び出します。

次のアイテム・コードを指定します。

アイテム・コード	必須	パラメータの種類
REG\$_CACHEACTION	省略可能	入力
REG\$_CLASSNAME	省略可能	入力 (Unicode 文字列を示すポインタ。Unicode 文字は 4 バイトの長さである。)
REG\$_DISPOSITION	省略可能	出力
REG\$_KEYFLAGS	省略可能	入力
REG\$_KEYID	必須	入力
REG\$_KEYRESULT	省略可能	出力
REG\$_LINKPATH	省略可能	入力 (Unicode 文字列を示すポインタ。Unicode 文字は 4 バイトの長さである。)
REG\$_LINKTYPE	省略可能	入力
REG\$_RETURNSTATUS	省略可能	出力
REG\$_SECACCESS	省略可能	入力
REG\$_SECURITYPOLICY	省略可能	入力
REG\$_SEPARATOR	省略可能	なし
REG\$_SUBKEYNAME	必須	入力 (Unicode 文字列を示すポインタ。Unicode 文字は 4 バイトの長さである。)
REG\$_VOLATILE	省略可能	入力

REG\$\_LINKPATH アイテム・コードを指定する場合は、OpenVMS Registry にすでに定義されているキー・パスを指すコードを指定しなければなりません。この条件を満たさないと、REG\$\_INVALIDPATH エラーが返されます。

---

**注意**

---

REG\$\_CACHEACTION アイテム・コードを指定しないと、親キーと同じキャッシュ・アクション値を使用して新しいキーが作成されます。REG\$\_VOLATILE アイテム・コードと REG\$\_SECURITYPOLICY アイテム・コードに同じ規則が適用されます。

---

#### REG\$FC\_DELETE\_KEY

この要求は、指定されたサブキーとその値を OpenVMS Registry データベースから削除します。指定されたキーにサブキーがある場合、そのキーは削除されません。その場合は、サブキーを最初に削除しなければなりません。

この機能コードを使用すると、複数の要求を \$REGISTRY サービスに対する 1 つの呼び出しにまとめることができます。この複数要求機能を使用するには、REG\$\_SEPARATOR アイテム・コードを使用して、現在の要求のアイテム・コード・セットの最後を示し、その後に別の要求が続くことを指定しなければなりません。

次のアイテム・コードを指定します。

アイテム・コード	必須	パラメータの種類
REG\$_KEYID	必須	入力
REG\$_KEYPATH	省略可能	入力 (Unicode 文字列を示すポインタ。Unicode 文字は 4 バイトの長さである。)
REG\$_RETURNSTATUS	省略可能	出力
REG\$_SEPARATOR	省略可能	なし
REG\$_SUBKEYNAME	必須	入力 (Unicode 文字列を示すポインタ。Unicode 文字は 4 バイトの長さである。)

#### REG\$FC\_DELETE\_VALUE

この要求は、指定された値をキーから削除します。

この機能コードを使用すると、複数の要求を \$REGISTRY サービスに対する 1 つの呼び出しにまとめることができます。この複数要求機能を使用するには、REG\$\_SEPARATOR アイテム・コードを使用して、現在の要求のアイテム・コード・セットの最後を示し、その後に別の要求が続くことを指定しなければなりません。

次のアイテム・コードを指定します。

アイテム・コード	必須	パラメータの種類
REG\$_KEYID	必須	入力
REG\$_KEYPATH	省略可能	入力 (Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)
REG\$_RETURNSTATUS	省略可能	出力
REG\$_SEPARATOR	省略可能	なし
REG\$_VALUENAME	必須	入力

#### REG\$FC\_ENUM\_KEY

この要求は、キーの 1 つのサブキーに関する情報を取得します。サブキーは REG\$\_SUBKEYINDEX アイテム・コードに指定します。キーのすべてのサブキーを列挙するには、REG\$FC\_ENUM\_KEY 機能コードを使用して \$REGISTRY サービスを繰り返し呼び出さなければなりません。REG\$\_SUBKEYINDEX を最初は 0 に設定して、要求から REG\$\_NOMOREITEMS エラーが返されるまで、カウントに 1 を加算します。

次のアイテム・コードを指定します。

アイテム・コード	必須	パラメータの種類
REG\$_CACHEACTION	省略可能	出力
REG\$_CLASSNAME	省略可能	出力 (Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)
REG\$_KEYFLAGS	省略可能	出力
REG\$_KEYID	必須	入力
REG\$_KEYPATH	省略可能	入力 (Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)
REG\$_LASTWRITE	省略可能	出力
REG\$_LINKCOUNT	省略可能	出力
REG\$_LINKPATH	省略可能	出力 (Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)
REG\$_LINKTYPE	省略可能	出力
REG\$_RETURNSTATUS	省略可能	出力
REG\$_SECURITYPOLICY	省略可能	出力
REG\$_SUBKEYINDEX	必須	入力
REG\$_SUBKEYNAME	省略可能	出力 (Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)
REG\$_VOLATILE	省略可能	出力

#### REG\$FC\_ENUM\_VALUE

この要求は、指定されたキー識別子の値に関する情報を取得します。取得する値は REG\$\_VALUEINDEX アイテム・コードに指定します。

キーのすべての値を列挙するには、アプリケーションで REG\$FC\_ENUM\_VALUE 機能コードを使用して \$REGISTRY サービスを繰り返し呼び出さなければなりません。REG\$\_VALUEINDEX の値を最初は 0 に設定して、要求から REG\$\_NOMOREITEMS エラーが返されるまで、カウントに 1 を加算します。

次のアイテム・コードを指定します。

アイテム・コード	必須	パラメータの種類
REG\$_DATAFLAGS	省略可能	出力
REG\$_DATATYPE	省略可能	出力
REG\$_KEYID	必須	入力
REG\$_KEYPATH	省略可能	入力 (Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)
REG\$_RETURNSTATUS	省略可能	出力
REG\$_VALUEDATA	省略可能	出力
REG\$_VALUEINDEX	必須	入力
REG\$_VALUENAME	省略可能	出力
REG\$_VOLATILE	省略可能	出力

#### REG\$FC\_FLUSH\_KEY

この要求は、指定されたキーに関するすべての情報をディスクに書き込みます。この要求では、操作の完了後に制御が戻され、キーのすべての属性が OpenVMS Registry データベースに書き込まれます。

次のアイテム・コードを指定します。

アイテム・コード	必須	パラメータの種類
REG\$_KEYID	必須	入力
REG\$_KEYPATH	省略可能	入力 (Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)
REG\$_RETURNSTATUS	省略可能	出力

#### REG\$FC\_MODIFY\_KEY

この要求は指定されたキーの属性を変更します。

次のアイテム・コードを指定します。

アイテム・コード	必須	パラメータの種類
REG\$_CACHEACTION	省略可能	入力
REG\$_CLASSNAME	省略可能	入力 (Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)
REG\$_KEYFLAGS	省略可能	入力
REG\$_KEYID	必須	入力
REG\$_KEYPATH	省略可能	入力 (Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)
REG\$_LINKPATH	省略可能	入力 (Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)
REG\$_LINKTYPE	省略可能	入力
REG\$_NEWNAME	省略可能	入力

アイテム・コード	必須	パラメータの種類
REG\$_RETURNSTATUS	省略可能	出力
REG\$_SECURITYPOLICY	省略可能	入力

指定されたキーからリンクを削除するには、アドレスを 0 に設定して REG\$\_LINKPATH アイテム・コードを入力します。値またはサブキー（またはその両方）があるキーにリンクを追加することはできません。

#### REG\$FC\_MODIFY\_TREE\_KEY

この要求は、指定されたキーとそのすべてのサブキーの属性を変更します。リンクは追跡されず、変更されません。

次のアイテム・コードを指定します。

アイテム・コード	必須	パラメータの種類
REG\$_CACHEACTION	省略可能	入力
REG\$_CLASSNAME	省略可能	入力 (Unicode 文字列を示すポインタ。Unicode 文字は 4 バイトの長さである。)
REG\$_KEYID	必須	入力
REG\$_KEYPATH	省略可能	入力 (Unicode 文字列を示すポインタ。Unicode 文字は 4 バイトの長さである。)
REG\$_RETURNSTATUS	省略可能	出力
REG\$_SECURITYPOLICY	省略可能	入力

#### REG\$FC\_NOTIFY\_CHANGE\_KEY\_VALUE

この要求は、指定されたキーまたはそのサブキーが変更されたときに、呼び出しプロセスにそのことを通知します。つまり、要求された機能は制御を戻す前に、指定された条件が発生するのを待ちます。

次のアイテム・コードを指定します。

アイテム・コード	必須	パラメータの種類
REG\$_FLAGSUBKEY	必須	入力
REG\$_KEYID	必須	入力
REG\$_KEYPATH	省略可能	入力 (Unicode 文字列を示すポインタ。Unicode 文字は 4 バイトの長さである。)
REG\$_NOTIFYFILTER	必須	入力
REG\$_RETURNSTATUS	省略可能	出力

#### REG\$FC\_OPEN\_KEY

この要求は指定されたキーを開きます。サブキーを指定しなかった場合は、REG\$\_KEYID に指定されているキーが開かれます。REG\$\_KEYID に定義済みキー以外のキーが指定された場合は、そのキーを再び開きます (キーを複製します)。

次のアイテム・コードを指定します。

アイテム・コード	必須	パラメータの種類
REG\$_KEYID	必須	入力
REG\$_KEYRESULT	必須	出力
REG\$_KEYPATH	省略可能	入力 (Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)
REG\$_RETURNSTATUS	省略可能	出力
REG\$_SECACCESS	必須	入力
REG\$_SUBKEYNAME	省略可能	入力 (Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)

#### REG\$FC\_QUERY\_KEY

この要求は、指定されたキーに関する属性を取得します。

次のアイテム・コードを指定します。

アイテム・コード	必須	パラメータの種類
REG\$_CACHEACTION	省略可能	出力
REG\$_CLASSNAME	省略可能	出力 (Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)
REG\$_CLASSNAMEMAX	省略可能	出力
REG\$_KEYFLAGS	省略可能	出力
REG\$_KEYID	必須	入力
REG\$_KEYPATH	省略可能	入力 (Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)
REG\$_LASTWRITE	省略可能	出力
REG\$_LINKCOUNT	省略可能	出力
REG\$_LINKPATH	省略可能	出力 (Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)
REG\$_LINKTYPE	省略可能	出力
REG\$_RETURNSTATUS	省略可能	出力
REG\$_SECURITYPOLICY	省略可能	出力
REG\$_SUBKEYNAMEMAX	省略可能	出力
REG\$_SUBKEYSNUMBER	必須	出力
REG\$_VALUEDATAMAX	省略可能	出力
REG\$_VALUENAMEMAX	省略可能	出力
REG\$_VALUENUMBER	省略可能	出力
REG\$_VOLATILE	省略可能	出力

#### REG\$FC\_QUERY\_VALUE

この要求は、指定された値名のタイプ、データ・フラグ、データを取得します。

この機能コードを使用すると、複数の要求を \$REGISTRY サービスに対する 1 つの呼び出しにまとめることができます。この複数要求機能を使用するには、REG\$\_SEPARATOR アイテム・コードを使用して、現在の要求のアイテム・コー

ド・セットの最後を示し、その後に別の要求が続くことを指定しなければなりません。

次のアイテム・コードを指定します。

アイテム・コード	必須	パラメータの種類
REG\$_DATAFLAGS	省略可能	出力
REG\$_DATATYPE	省略可能	出力
REG\$_KEYPATH	省略可能	入力 (Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)
REG\$_KEYID	必須	入力
REG\$_LINKCOUNT	省略可能	出力
REG\$_LINKPATH	省略可能	出力 (Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)
REG\$_LINKTYPE	省略可能	出力
REG\$_RETURNSTATUS	省略可能	出力
REG\$_SEPARATOR	省略可能	なし
REG\$_VALUEDATA	省略可能	出力
REG\$_VALUENAME	必須	入力
REG\$_VOLATILE	省略可能	出力

#### REG\$FC\_SEARCH\_TREE\_DATA

この要求は、指定されたキーとそのすべての子孫をスキャンし、指定されたデータ情報と一致するものを検索します。指定できるデータ情報は REG\$\_DATAFLAGS アイテム・コード、または REG\$\_DATATYPE アイテム・コードと REG\$\_VALUEDATA アイテム・コードの組み合わせ、またはこの 3 つのアイテム・コードの組み合わせです。

REG\$\_FLAGOPCODE アイテム・コードは、REG\$\_DATAFLAGS アイテム・コードをデータベースとどのように照合するかを指定します (REG\$\_FLAGOPCODE アイテム・コードの詳細については、アイテム・コードの説明を参照してください)。

一致するものが見つかり、そのたびに指定されたキーに対応するパス名が REG\$\_PATHBUFFER アイテム・コードに追加されます。値のパス名は Unicode の Null 文字を使用して区切ります。

アプリケーションで提供されたバッファが十分な大きさでないために、検索されたすべての値パス名を格納できない場合は、SS\$\_BUFFEROVF エラー・メッセージが iosb 引数に返され、操作を正常終了するのに必要な長さが REG\$\_REQLENGTH アイテム (指定されている場合) に返されます。

REG\$\_KEYPATH アイテム・コードで 0 個以上のサブキーを照合するには、繰り返し記号 (...) ワイルドカードを使用します (たとえば、Hardware\...\disks と指定すると、Hardware サブキーから始まり、disk サブキーで終了し、その間に 0 個以上のサブキーのあるすべてのパスが検索されます)。サブキー全体またはサブキーの一部だけを照合するには、REG\$\_KEYPATH アイテム・コードにアスタリスク (\*) ワイルドカードを

指定します。キー名の中の 1 文字だけを照合するには、パーセント (%) ワイルドカードを REG\$ \_KEYPATH アイテム・コードに指定します。

次のアイテム・コードを指定します。

アイテム・コード	必須	パラメータの種類
REG\$ _DATAFLAGS	省略可能	入力
REG\$ _DATATYPE	省略可能	入力
REG\$ _FLAGOPCODE	省略可能	入力
REG\$ _KEYPATH	省略可能	入力 (Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)
REG\$ _KEYID	必須	入力
REG\$ _PATHBUFFER	必須	出力
REG\$ _REQLENGTH	省略可能	出力
REG\$ _RETURNSTATUS	省略可能	出力
REG\$ _VALUEDATA	省略可能	入力

#### REG\$FC\_SEARCH\_TREE\_KEY

この要求は、指定されたキーとそのすべての子孫をスキャンし、指定されたキー・パスを検索します。

この機能コードの場合、指定できるキー・パスは Unicode 文字列であり、繰り返し記号 (...), アスタリスク (\*), パーセント (%) ワイルドカード文字を含むことができますが、先頭にバックスラッシュ文字 (\) を指定することはできません。

1 つ以上のサブキーと一致することを指定するには、繰り返し記号 (...) ワイルドカードを REG\$ \_KEYPATH アイテム・コードに指定します (たとえば、Hardware\...\disks と指定すると、Hardware サブキーから始まり、disk サブキーで終了し、その間に 0 個以上のサブキーがあるすべてのパスが検索されます)。サブキー全体またはサブキーの一部と一致することを指定するには、アスタリスク (\*) ワイルドカードを REG\$ \_KEYPATH アイテム・コードに指定します。キー名の 1 文字と一致することを指定するには、パーセント (%) ワイルドカードを REG\$ \_KEYPATH アイテム・コードに指定します。

次の例は正しいキー・パスを示しています。

```
hardware\system*\disk%
```

一致するものが見つかるたびに、指定されたキー識別子を基準にした相対パス名が REG\$ \_PATHBUFFER アイテム・コードに追加されます。サブキーのパス名は Unicode の Null 文字 (4 バイト) で区切られます。

アプリケーションが提供したバッファが十分な大きさでないために、検索したサブキーのすべてのパス名を格納できない場合は、SS\$ \_BUFFEROVF エラー・メッセージが iosb 引数に返され、操作を正常終了するのに必要な長さが REG\$ \_REQLENGTH アイテム (指定されている場合) に返されます。

次のアイテム・コードを指定します。

アイテム・コード	必須	パラメータの種類
REG\$_KEYID	必須	入力
REG\$_KEYPATH	省略可能	入力 (Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)
REG\$_PATHBUFFER	必須	出力
REG\$_REQLENGTH	省略可能	出力
REG\$_RETURNSTATUS	省略可能	出力

#### REG\$FC\_SEARCH\_TREE\_VALUE

この要求は、指定されたキーとそのすべての子孫をスキャンし、指定された値名を検索します。

この機能コードの場合、指定できるキー名は Unicode 文字列であり、繰り返し記号 (...), アスタリスク (\*), パーセント (%) ワイルドカード文字を含むことができますが、先頭にバックスラッシュ文字 (\) を指定することはできません。

1 つ以上のサブキーと一致することを指定するには、繰り返し記号 (...) ワイルドカードを REG\$\_KEYPATH アイテム・コードに指定します (たとえば, Hardware\...\disks と指定すると, Hardware サブキーから始まり, disk サブキーで終了し, その間に 0 個以上のサブキーがあるすべてのパスが検索されます)。サブキー全体またはサブキーの一部と一致することを指定するには、アスタリスク (\*) ワイルドカードを REG\$\_KEYPATH アイテム・コードに指定します。キー名の 1 文字と一致することを指定するには、パーセント (%) ワイルドカードを REG\$\_KEYPATH アイテム・コードに指定します。

次の例は正しいキー・パスを示しています。

```
hardware\system\...
```

この機能コードの場合、指定できる名前は Unicode 文字列であり、アスタリスク (\*) およびパーセント (%) ワイルドカード文字を含むことができます。

一致するものが見つかるたびに、指定されたキー識別子を基準にした相対パス名が REG\$\_PATHBUFFER アイテム・コードに追加されます。サブキーのパス名は Unicode の Null 文字 (4 バイト) で区切られます。

アプリケーションが提供したバッファが十分な大きさでないために、検索したサブキーのすべてのパス名を格納できない場合は、SS\$\_BUFFEROVF エラー・メッセージが iosb 引数に返され、操作を正常終了するのに必要な長さが REG\$\_REQLENGTH アイテム (指定されている場合) に返されます。

次のアイテム・コードを指定します。

アイテム・コード	必須	パラメータの種類
REG\$_KEYPATH	省略可能	入力 (Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)
REG\$_KEYID	必須	入力
REG\$_PATHBUFFER	必須	出力
REG\$_REQLENGTH	省略可能	出力
REG\$_RETURNSTATUS	省略可能	出力
REG\$_VALUENAME	必須	入力

#### REG\$FC\_SET\_VALUE

この要求は、指定されたキーの値とタイプ情報を設定します。

この機能コードを使用すると、複数の要求を \$REGISTRY サービスに対する 1 つの呼び出しにまとめることができます。この複数要求機能を使用するには、REG\$\_SEPARATOR アイテム・コードを使用して、現在の要求のアイテム・コード・セットの最後を示し、その後に別の要求が続くことを指定しなければなりません。

値がリンクに設定される場合は、REG\$\_M\_IGNORE\_LINKS 機能コード修飾子を指定した場合を除き、リンクが有効であるかどうか確認されます。

次のアイテム・コードを指定します。

アイテム・コード	必須	パラメータの種類
REG\$_DATAFLAGS	省略可能	入力
REG\$_DATATYPE	省略可能	入力
REG\$_KEYID	必須	入力
REG\$_KEYPATH	省略可能	入力 (Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)
REG\$_LINKPATH	省略可能	入力 (Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)
REG\$_LINKTYPE	省略可能	入力
REG\$_RETURNSTATUS	省略可能	出力
REG\$_SEPARATOR	省略可能	なし
REG\$_VALUEDATA	省略可能	入力
REG\$_VALUENAME	省略可能	入力

## アイテム・コード

表 10-4 は、itmlst 引数にアイテム記述子として指定できるアイテム・コードを示しています。この表にはアイテム・コード、入力と出力の区別、データ・タイプが示されています。各アイテム・コードの詳細については、この表の後の説明を参照してください。

表 10-4 アイテム・コードの要約

アイテム・コード	入力/出力	データ・タイプ
REG\$_CACHEACTION	入力, 出力	ロングワード
REG\$_CLASSNAME	入力, 出力	(Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)
REG\$_CLASSNAMEMAX	出力	ロングワード
REG\$_DATAFLAGS	入力, 出力	クォドワード
REG\$_DATATYPE	入力, 出力	ロングワード
REG\$_DISPOSITION	出力	ロングワード
REG\$_FILELOAD	入力	Unicode 文字列
REG\$_FLAGOPCODE	入力	ロングワード
REG\$_FLAGSUBKEY	入力	ロングワード
REG\$_KEYPATH	入力	(Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)
REG\$_KEYFLAGS	入力, 出力	ロングワード
REG\$_KEYID	入力, 出力	ロングワード
REG\$_KEYRESULT	出力	ロングワード
REG\$_LASTWRITE	出力	クォドワード
REG\$_LINKCOUNT	出力	ロングワード
REG\$_LINKPATH	入力, 出力	(Unicode 文字列を示すポインタ。 Unicode 文字は 4 バイトの長さである。)
REG\$_LINKTYPE	入力, 出力	ロングワード
REG\$_NEWNAME	入力	Unicode 文字列
REG\$_NOTIFYFILTER	入力	ロングワード
REG\$_PATHBUFFER	出力	バッファ
REG\$_REQLENGTH	出力	ロングワード
REG\$_RETURNSTATUS	出力	ロングワード
REG\$_SECACCESS	入力	ロングワード
REG\$_SECURITYPOLICY	入力, 出力	ロングワード
REG\$_SEPARATOR	なし	なし
REG\$_SUBKEYINDEX	入力	ロングワード

(次ページに続く)

表 10-4 (続き) アイテム・コードの要約

アイテム・コード	入力/出力	データ・タイプ
REG\$_SUBKEYNAME	入力, 出力	(Unicode 文字列を示すポインタ。Unicode 文字は 4 バイトの長さである。)
REG\$_SUBKEYNAMEMAX	出力	ロングワード
REG\$_SUBKEYSNUMBER	出力	ロングワード
REG\$_VALUEDATA	入力, 出力	バッファ
REG\$_VALUEDATAMAX	出力	ロングワード
REG\$_VALUEINDEX	入力	ロングワード
REG\$_VALUENAME	入力, 出力	Unicode 文字列
REG\$_VALUENAMEMAX	出力	ロングワード
REG\$_VALUENUMBER	出力	ロングワード
REG\$_VOLATILE	入力, 出力	ロングワード

#### REG\$\_CACHEACTION

REG\$\_CACHEACTION アイテム・コードは入力アイテム・コードです。このアイテム・コードは、指定されたオブジェクトに関する情報をディスクに直ちに書き込まなければならないかどうかを指定するロングワード・フラグです。次のいずれかの値です。

キャッシュ値	説明
REG\$_K_WRITEBEHIND	指定されたオブジェクトに関する情報をディスクに後で書き込む (デフォルト)。
REG\$_K_WRITETHRU	指定されたオブジェクトに関する情報をディスクに直ちに書き込む。

#### 注意

このアイテム・コードを指定しなかった場合は、値またはキーはその値を親オブジェクトから継承します。デフォルトでは、エントリ・ポイント (REG\$\_HKEY\_CLASSES\_ROOT, REG\$\_HKEY\_LOCAL\_MACHINE, REG\$\_HKEY\_USERS) は、REG\$\_K\_WRITEBEHIND に等しい値に設定されます。

#### REG\$\_CLASSNAME

REG\$\_CLASSNAME アイテム・コードは、機能コードに応じて、入力アイテム・コードまたは出力アイテム・コードになります。クラス名はキーの情報フィールドです。たとえば、オブジェクトのタイプはクラス名です。このアイテム・コードは Unicode 文字列で構成できます。Unicode 文字は 4 バイト長です。

#### REG\$\_CLASSNAMEMAX

REG\$\_CLASSNAMEMAX アイテム・コードは出力アイテム・コードです。サブキーのクラス名を指定する最長の文字列の長さ (バイト数) が格納されます。

### REG\$\_DATAFLAGS

REG\$\_DATAFLAGS アイテム・コードは、機能コードに応じて、入力アイテム・コードまたは出力アイテム・コードになります。これは 64 ビットのアプリケーションに依存する値データ・フラグです。

### REG\$\_DATATYPE

REG\$\_DATATYPE アイテム・コードは、機能コードに応じて、入力アイテム・コードまたは出力アイテム・コードになります。これは、値データとして格納される情報のタイプを指定するか、または指定された値データ・コンポーネントの情報のタイプを格納します。値は次のいずれかです。

タイプ・コード	説明
REG\$_K_BINARY	バイナリ・データ
REG\$_K_DWORD	32 ビット数値
REG\$_K_EXPAND_SZ	Unicode 文字列
REG\$_K_MULTI_SZ	REG\$_K_SZ 文字列を連結した配列
REG\$_K_NONE	定義されている値タイプなし (デフォルト)
REG\$_K_QWORD	64 ビット数値
REG\$_K_SZ	最後が Null の Unicode 文字列

#### REG\$\_K\_EXPAND\_SZ と REG\$\_K\_SZ の相違点

文字列は通常、人間が判読できる形式の一連の文字です。OpenVMS Registry の多くの値エントリは、string (REG\$\_K\_SZ) または expandable string (REG\$\_K\_EXPAND\_SZ) 形式を使用して書き込まれます。拡張可能文字列は通常、人間が判読可能なテキストですが、文字列がアプリケーションから呼び出されるときに置換される変数が含まれることもあります。

たとえば、Windows NT システムでは、値エントリ %SystemRoot%\System32\Bootok.exe の %SystemRoot% は変数の拡張可能な部分です。この部分は、Windows NT システム・ファイルが格納されているディレクトリの実地の場所に置き換えられます。

### REG\$\_DISPOSITION

REG\$\_DISPOSITION アイテム・コードは出力アイテム・コードです。これはロングワードであり、値は次のいずれかです。

処置の値	説明
REG\$_K_CREATENEWKEY	キーが存在しなかったので作成された。
REG\$_K_OPENEXISTINGKEY	キーが存在し、開かれた。

### REG\$\_FLAGOPCODE

REG\$\_FLAGOPCODE アイテム・コードは入力アイテム・コードです。これは、REG\$\_DATAFLAGS 入力アイテム・コードを OpenVMS Registry データベース

内のデータ・フラグ・フィールドとどのような方法で照合するかを示すロングワード・フラグです。値は次のいずれかです。

オペレータ・コード・オプション	説明
REG\$K_ANY	OpenVMS Registry データベース内のデータ・フィールドには、REG\$_DATAFLAGS 入力アイテム・コードのフラグが少なくとも 1 つ含まれていなければならない。
REG\$K_EXACTMATCH	REG\$_DATAFLAGS 入力アイテム・コードは、OpenVMS Registry データベース内の data flags フィールドと正確に一致しなければならない。
REG\$K_EXCLUDE	OpenVMS Registry データベース内の data flags フィールドには、REG\$_DATAFLAGS 入力アイテム・コード内のフラグが含まれてはならない。
REG\$K_INCLUDE	OpenVMS Registry データベース内の data flags フィールドには、少なくとも REG\$_DATAFLAGS 入力アイテム・コード内のフラグが含まれなければならない。
REG\$K_NOTANY	OpenVMS Registry データベース内のデータ・フィールドには、REG\$_DATAFLAGS 入力アイテム・コード内のどのフラグも含まれてはならない。

#### REG\$\_FLAGSUBKEY

REG\$\_FLAGSUBKEY アイテム・コードは入力アイテム・コードです。これはロングワードの論理値 (Boolean) フィールドであり、次のことを指定します。

- 1 の場合は、指定されたキーとそのサブキーの変更点を報告します。
- 0 の場合は、指定されたキーだけの変更点を報告します。

#### REG\$\_KEYID

REG\$\_KEYID アイテム・コードは入力アイテム・コードです。これはキー識別子を含むロングワードです。

#### REG\$\_KEYRESULT

REG\$\_KEYRESULT アイテム・コードは出力アイテム・コードです。これはキー識別子を受け取るロングワードです。キー識別子は、REG\$\_KEYID アイテム・コードを使用して、他のレジストリ呼び出しに渡すことができます。

#### REG\$\_KEYPATH

REG\$\_KEYPATH アイテム・コードは入力アイテム・コードです。これはキー・パスを指定する Unicode 文字列です。Unicode 文字は 4 バイト長です。

#### REG\$\_LASTWRITE

REG\$\_LASTWRITE アイテム・コードは出力アイテム・コードです。これは絶対時間を表すクォドワードであり、指定されたキーが最後に書き込まれた時刻 (値の変更を含む) を受け取ります。

#### REG\$\_LINKCOUNT

REG\$\_LINKCOUNT アイテム・コードは出力アイテム・コードです。これはアイテムを参照するシンボリック・リンクの数を示すロングワードです。

#### REG\$\_LINKPATH

REG\$\_LINKPATH アイテム・コードは、機能コードに応じて、入力アイテム・コードまたは出力アイテム・コードになります。これは指定されたキーのリンク先のキー・パスを指定する Unicode 文字列です。Unicode 文字は 4 バイト長です。

#### REG\$\_LINKTYPE

REG\$\_LINKTYPE アイテム・コードは、機能コードに応じて、入力アイテム・コードまたは出力アイテム・コードになります。これはリンク・タイプを示すロングワードです。

リンク・タイプ	説明
REG\$_NONE	リンクなし (デフォルト)
REG\$_SYMBOLICLINK	シンボリック (論理) リンク

#### REG\$\_NEWNAME

REG\$\_NEWNAME アイテム・コードは、キーの新しい名前を指定する Unicode 文字列です。

#### REG\$\_NOTIFYFILTER

REG\$\_NOTIFYFILTER アイテム・コードは入力アイテム・コードです。これは、指定されたキーとそのサブキーおよび値に対して、どの変更を報告するかを指定するロングワード・マスクです。次の値を任意の組み合わせで指定します。

値	説明
REG\$_CHANGEATTRIBUTES	指定されたキーまたはそのサブキーの属性の変更。
REG\$_CHANGELASTSET	指定されたキーまたはそのサブキーの最終書き込み時刻の変更。
REG\$_CHANGENAME	作成と削除も含めて、指定されたキーまたはそのサブキーのキー名の変更。

#### 注意

指定されたキーのサブキーに対する変更が報告されるのは、REG\$\_FLAGSUBKEY アイテム・コードが 1 に設定されている場合だけです。

#### REG\$\_PATHBUFFER

REG\$\_PATHBUFFER アイテム・コードは出力アイテム・コードです。これは、Unicode の Null 文字 (4 バイト長) で区切られたキー・パスまたは値パスの集合を受け取るバッファです (アイテム記述子の 3 番目のロングワードには、バッファに書き込まれたバイト数が格納されます)。

#### REG\$\_REQLENGTH

REG\$\_REQLENGTH アイテム・コードは出力アイテム・コードです。操作を正常終了するのに必要なバッファ・サイズ (バイト数) を受け取るロングワードです。

#### REG\$\_RETURNSTATUS

REG\$\_RETURNSTATUS アイテム・コードは出力アイテム・コードです。指定された操作の最終終了状態を受け取るログワードです。詳細については、この章の返される条件値を参照してください。

#### REG\$\_SECACCESS

REG\$\_SECACCESS アイテム・コードは入力アイテム・コードです。新しいキーにとって適切なセキュリティ・アクセスを指定するログワード・マスクです。次の値を任意に組み合わせて指定できます。

セキュリティ・アクセス・マスク	説明
REG\$_ALLACCESS	次のアクセス値の組み合わせ。  REG\$_CREATELINK REG\$_CREATESUBKEY REG\$_ENUMSUBKEYS REG\$_NOTIFY REG\$_QUERYVALUE REG\$_SETVALUE
REG\$_CREATELINK	シンボリック・リンクの作成を許可する。
REG\$_CREATESUBKEY	サブキーの作成を許可する。
REG\$_ENUMSUBKEYS	サブキーの列挙を許可する。
REG\$_EXECUTE	読み込みアクセスを許可する。
REG\$_NOTIFY	変更の通知を許可する。
REG\$_QUERYVALUE	サブキーのデータの問い合わせを許可する。
REG\$_READ	次のアクセス値の組み合わせ。  REG\$_ENUMSUBKEYS REG\$_QUERYVALUE REG\$_NOTIFY
REG\$_SETVALUE	値とデータの設定を許可する。
REG\$_WRITE	次のアクセス値の組み合わせ。  REG\$_CREATESUBKEY REG\$_SETVALUE

#### REG\$\_SECURITYPOLICY

REG\$\_SECURITYPOLICY アイテム・コードは入力アイテム・コードです。キーに対して適用されるセキュリティ・ポリシーを指定するログワードです。値は次のとおりです。

ポリシーの設定	説明
REG\$_POLICY_NT_40	最初のキーと要求されたキーに対してアクセス権が必要である (デフォルト)。

#### REG\$\_SEPARATOR

REG\$\_SEPARATOR アイテム・コードは空アイテム・コードです。これは複数のアイテム・コードを区切るためのセパレータです。

この機能コードを使用すると、複数の要求を \$REGISTRY サービスに対する 1 つの呼び出しにまとめることができます。この複数要求機能を使用するには、REG\$\_SEPARATOR アイテム・コードを使用して、現在の要求のアイテム・コード・セットの最後を示し、その後には別の要求が続くことを指定しなければなりません。

#### REG\$\_SUBKEYINDEX

REG\$\_SUBKEYINDEX アイテム・コードは入力アイテム・コードです。検索するサブキーのインデックスを指定するロングワードです。

#### REG\$\_SUBKEYNAME

REG\$\_SUBKEYNAME アイテム・コードは入力アイテム・コードです。サブキーの名前を指定する Unicode 文字列です。Unicode 文字は 4 バイト長です。

#### REG\$\_SUBKEYNAMEMAX

REG\$\_SUBKEYNAMEMAX アイテム・コードは出力アイテム・コードです。指定されたキーの最長のサブキー名の長さ (文字数) を受け取るロングワードです。

#### REG\$\_SUBKEYSNUMBER

REG\$\_SUBKEYSNUMBER アイテム・コードは出力アイテム・コードです。指定されたキーに含まれているサブキーの数を受け取るロングワードです。

#### REG\$\_VALUEINDEX

REG\$\_VALUEINDEX アイテム・コードは入力アイテム・コードです。指定されたキー内で検索する値のインデックスを指定するロングワードです。値のインデックスは 0 から始まり、REG\$\_VALUENUMBER から返された数より 1 だけ少ない値まで可能です。

#### REG\$\_VALUEDATA

REG\$\_VALUEDATA アイテム・コードは、機能コードに応じて、入力アイテム・コードまたは出力アイテム・コードになります。OpenVMS Registry に書き込まれる値データ・コンポーネントが格納されるバッファであるか (入力)、OpenVMS Registry からデータ値コンポーネントを受け取るバッファになります (出力)。

#### REG\$\_VALUEDATAMAX

REG\$\_VALUEDATAMAX アイテム・コードは出力アイテム・コードです。指定されたキーの最長のデータ・コンポーネント値の長さ (バイト数) を受け取るロングワードです。

#### REG\$\_VALUENAME

REG\$\_VALUENAME アイテム・コードは、機能コードに応じて、入力アイテム・コードまたは出力アイテム・コードになります。値の名前を指定する Unicode 文字列です。

#### REG\$\_VALUENAMEMAX

REG\$\_VALUENAMEMAX アイテム・コードは出力アイテム・コードです。指定されたキーの最長の値名の長さ (文字数) を受け取るロングワードです。

#### REG\$\_VALUENUMBER

REG\$\_VALUENUMBER アイテム・コードは出力アイテム・コードです。指定されたキーに含まれている値の数を受け取るロングワードです。

#### REG\$\_VOLATILE

REG\$\_VOLATILE アイテム・コードはアイテムの揮発性を示します。出力として、オブジェクトの揮発性を返します。OpenVMS では、揮発性キーと揮発性値は、OpenVMS Registry サーバを稼動しているすべてのノードがリブートされたときに失われます (スタンドアロン・システムでは、揮発性キーと揮発性値は、システムのリブート時に失われます)。

揮発性タイプ	説明
REG\$K_CLUSTER	アイテムはクラスタのリブート時に削除される。
REG\$K_NONE	アイテムは揮発性ではない (デフォルト)。

---

## 機能修飾子

機能修飾子を使用すると、機能コード値の上位ビットを必要に応じて指定できます。これらの各ビットは機能の動作を変更できます。

たとえば、REG\$FC\_CREATE\_KEY という機能に対して REG\$M\_CASE\_SENSITIVE 機能修飾子を指定できます。このように機能と機能修飾子を組み合わせると、OpenVMS Registry に渡されるデータで大文字と小文字が区別されます。2 つの値は DEC C では、REG\$M\_CASE\_SENSITIVE | REG\$FC\_CREATE\_KEY として指定されます。

OpenVMS Registry 機能修飾子はヘッダ・ファイル REGDEF.H に定義されています。

#### REG\$M\_CASE\_SENSITIVE

キーと値を照合するときに、大文字と小文字を区別します。

#### REG\$M\_DISABLE\_WILDCARDS

この機能に対しては、ワイルドカード文字を通常の文字として取り扱います。

#### REG\$M\_IGNORE\_LINKS

操作がキーまたは値に関連付けられたシンボリック・リンクを追跡しないことを指定します。

デフォルト設定では、キーまたは値が別のキーまたは値にシンボリック・リンクで関連付けられている場合、システムはすべてのリンクを追跡し、機能コードに指定された操作をリンクされているキーまたは値に対して実行します。

REG\$M\_IGNORE\_LINKS 機能修飾子を指定すると、機能コードによって指定された操作は指定されたキーまたは値にだけ影響し、リンク先のキーや値には影響しません。

デフォルト設定では、キーまたは値にシンボリック・リンクが設定されている場合は、そのキーや値を削除できません。しかし、REG\$M\_IGNORE\_LINKS機能修飾子を指定した場合は、キーまたは値は削除されます。

#### REG\$M\_NOW

REG\$\_CACHEACTIONアイテム・コードの値とは無関係に、ディスクに直ちに書き込みます。



# 第3部

---

## OpenVMS Events

第3部では、OpenVMS Events についての関連情報を扱います。



## 11.1 イベントとは

Windows NT システムでは、イベントは、システムまたはアプリケーション内で発生する重要な出来事です。たとえば、サービスの開始や停止、ユーザのログオンやログオフ、リソースへのアクセスなどはイベントです。システムでイベントが検出されると、イベント・ログ・サービスがイベント (または監査エントリ) をレコード形式でシステム・ログ、セキュリティ・ログ、アプリケーション・ログのいずれかに書き込み、監査トレールを作成します。レコードには日時、ソース (発生源)、カテゴリ、イベント番号、ユーザ、コンピュータに関する情報が格納されます。Windows NT システムでは、これらのログやそこに記録されているイベントは、イベント・ビューアを使用して表示します。

COM Version 1.0 for OpenVMS では、OpenVMS はすべての COM for OpenVMS イベントを DCOM\$EVENTLOG.RPT テキスト・ファイルに書き込んでいました。COM Version 1.1 for OpenVMS では、OpenVMS は COM for OpenVMS イベントの Windows NT ログと Advanced Server for OpenVMS ログの両方をサポートするようになりました。COM for OpenVMS イベント (OpenVMS での COM サーバの起動など) をログに記録し、Windows NT システムまたは OpenVMS システムからこれらの OpenVMS イベントを確認できるようになりました。

OpenVMS Events の依存関係についてと、OpenVMS Events が OpenVMS インフラストラクチャの他の部分とどのように関係するかの詳細については、第 4.9 節を参照してください。

### 11.1.1 参考文献

イベントと関連トピックの詳細については、次の参考文献が役立ちます。

- イベント・ログに関する書籍:
  - 『*Windows NT Server 4.0 Unleashed*』 Jason Garms 著, SAMS Publishing, Indianapolis, IN, 1998。ISBN: 0-672-30933-5
  - 『*Win32 System Services: The Heart of Window 95 and Windows NT*』 Marshall Brain 著, Prentice Hall, Upper Saddle River, NJ, 1996。ISBN: 0-13-324732-5
- その他の情報源:
  - Microsoft *Win32 Software Development Kit*

特に、RegisterEventSource、ReportEvent、DeregisterEventSource機能に関する節と、System Services: *Event Logging*という節を参照してください。

---

## 11.2 OpenVMS Events の概要

システムは OpenVMS Events を Windows NT イベント・ログ、Advanced Server for OpenVMS イベント・ログ、OpenVMS システムのログ・ファイルに記録します。

OpenVMS Events を表示するには、次のいずれかを使用します。

- Windows NT イベント・ビューア (第 11.2.1 項を参照)
- Advanced Server for OpenVMS イベント・ビューア (第 11.2.2 項を参照)
- OpenVMS イベント・ログ・ファイル (第 11.2.3 項を参照)

### 11.2.1 Windows NT イベント・ビューアによる OpenVMS Events の表示

Windows NT イベント・ビューアを使用して OpenVMS Events を表示するには、次の操作を行います。

1. Windows NT イベント・ビューアを起動します。

「スタート」メニューから「プログラム」、「管理ツール」、「イベントビューア」を選択します。

2. 「イベントビューア」ウィンドウで「ログ」メニューの「コンピュータの選択...」をクリックし、リスト・ボックスから OpenVMS システムを選択します。
3. 「イベントビューア」ウィンドウで「ログ」メニューの「システム」をクリックして、システム・イベント・ログを表示します。システム・イベント・ログには、COM for OpenVMS イベントが格納されています。

COM for OpenVMS イベントだけを表示するには、次の操作を行います。

- 「イベントビューア」ウィンドウで「表示」メニューの「イベントのフィルタ...」をクリックします。  
「フィルタ」ウィンドウが表示されます。
- 「フィルタ」ウィンドウの「ソース:」リスト・ボックスから「DCOM」を選択します。

### 11.2.2 Advanced Server for OpenVMS イベント・ビューアによる OpenVMS Events の表示

COM for OpenVMS イベントを表示するには、次の操作を行います。

1. Advanced Server for OpenVMS が実行されていることを確認します。

2. 次の Advanced Server for OpenVMS ADMINISTRATOR コマンドを入力します。

```
$ ADMIN SHOW EVENTS/TYPE=SYSTEM/SOURCE=DCOM/FULL
```

ビューアは COM for OpenVMS イベントだけを表示し、COM for OpenVMS イベントに関連付けられた補足情報も表示します。

### 11.2.3 OpenVMS のみでのイベントのログ

場合によっては、COM for OpenVMS イベントを OpenVMS システムのみで書き込んだり、表示しなければならないことがあります。Windows NT ログの代わりに、Compaq は COM イベント情報を OpenVMS ファイルに書き込む別のイベント・ロガーを提供しています。このファイルは次の場所にあります。

```
SY$MANAGER:DCOM$EVENTLOG.RPT
```

COM サーバ (DCOM\$RPCSS) がエラーを検出すると、COM for OpenVMS はこのイベント・ログ・レポートを自動的に作成します。イベント・ロガーはファイルの最後に新しいイベントを追加します。記録されるイベントの形式は次のとおりです。

```
event type : ddd mmm dd hh:mm:ss yyyy
```

```
First event message
```

```
event type : ddd mmm dd hh:mm:ss yyyy
```

```
Second event message
```

```
.  
. .  
.
```

例 11-1 はイベント・ログの内容を示しています。

例 11-1 OpenVMS イベント・ログの例

```
$ Type SYS$MANAGER:DCOM$EVENTLOG.RPT  
  
1  
ERROR : Tue Sep 15 11:18:54 1998  
Unable to start a DCOM Server: {5E9DDEC7-5767-11CF-BEAB-00AA006C3606}  
Runas (null)/SMITH  
The Windows NT error: 1326  
Happened while starting: device:[account]SSERVER.EXE  
  
2  
ERROR : Tue Sep 15 19:14:45 1998  
The server {0C092C21-882C-11CF-A6BB-0080C7B2D682} did not register  
with DCOM within the required timeout.
```

- 1 システムは最初のエラー・イベントをTue Sep 15 11:18:54 1998に記録しました。COM サーバ (DCOM\$RPCSS) は、SMITHアカウントで稼動しているクライアントの代わりに、COM アプリケーションdevice:[account]SSERVER.EXEを起動できませんでした (クライアントは“access denied”などのエラーを受け取っている可能性があります)。この結果作成された Windows NT エラーは1326であり、これが“Logon failure: unknown user name or bad password”に変換されます。

このエラーを検出した場合は、OpenVMS Authorize ユーティリティ (AUTHORIZE) を使用して、ユーザ・アカウントが有効であるかどうか確認してください。

- 2 システムは2番目のエラー・イベントをTue Sep 15 19:14:45 1998に記録しました。COM サーバ (DCOM\$RPCSS) は、COM アプリケーション{0C092C21-882C-11CF-A6BB-0080C7B2D682}を起動することはできましたが、アプリケーションは正常に動作しませんでした。アプリケーションは指定された時間内にDCOM\$RPCSSに登録されませんでした (クライアントは“Server execution failed” CO\_E\_SERVER\_EXEC\_FAILUREなどのエラーを受け取っている可能性があります)。

このエラーを検出した場合は、対話方式でサーバ・アプリケーションを実行し、整合性を判断してください。

---

# NTA\$EVENTW

## OpenVMS Events へのインタフェース

アプリケーションがイベント・ログ・ファイルに情報を記録することを許可します。

NTA\$EVENTW ルーチンはすべての操作を同期的に完了します。

---

### フォーマット

NTA\$EVENTW [*nullarg*], *func*, *itmlst*, *evsb*

---

### 引数

*nullarg*

OpenVMS usage: reserved  
type: longword (unsigned)  
access: read only  
mechanism: by value

Compaq が使用するために予約されています。

*func*

OpenVMS usage: function\_code  
type: longword (unsigned)  
access: read only  
mechanism: by value

NTA\$EVENTW が実行する機能を指定する機能コード。 *func* 引数は機能コードを格納したロングワードです。 \$EVENTDEF マクロは各機能コードの名前を定義します。

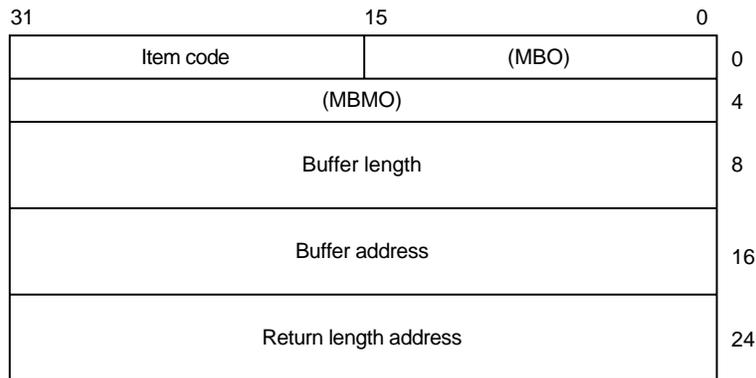
*itmlst*

OpenVMS usage: address of item list  
type: 64-bit address  
access: read only  
mechanism: by value

イベントのソース (発生源) またはイベントに関する情報を指定するアイテム・リスト。 *itmlst* 引数はアイテム記述子のリストの 64 ビット・アドレスであり、各記述子は情報アイテムを示します。 64 ビット形式のアイテム・リストはクオドワードの 0 で終了します。

次の図は、1 つのアイテム記述子の 64 ビット形式を示しています。

OpenVMS Events ルーチン  
NTA\$EVENTW



ZK-8782A-GE

evsb

OpenVMS usage: address of status block  
 type: 64-bit address  
 access: write only  
 mechanism: by reference

要求された操作の終了状態が格納されるイベント状態ブロック。

NTA\$EVENTW は要求を開始するときに、状態ブロックを 0 に設定します。要求の終了時に、EVT\$SL\_VMS\_STATUS フィールドには操作の主 (OpenVMS) 終了状態が格納されます。

エラーが発生した場合、EVT\$SL\_NT\_STATUS (0 以外の場合) は、エラー状態をさらに定義する補助エラー状態です。

---

機能コード

EVT\$FC\_REGISTER\_EVENT\_SOURCE  
 イベント・ログとの関連性を開きます。

アイテム・コード	必須	パラメータ	データ・タイプ
EVT\$SERVER_NAME	省略可能	入力	文字列 (4 バイトの Unicode)
EVT\$SOURCE	省略可能	入力	文字列 (4 バイトの Unicode)
EVT\$HANDLE	必須	出力	符号なしロングワード

- EVT\$SERVER\_NAME

この操作が実行されるサーバの UNC (汎用命名規則) 名。

UNC 名は \\server\share\path\file という形式です。このアイテムは 0 または未指定でなければなりません。これはクラスタ内で使用可能な Advanced Server for OpenVMS サーバで操作を実行します。

• EVTS\_SOURCE

イベントを記録するアプリケーションの名前。このフィールドは、説明テキストが格納されているアプリケーション・メッセージ・ファイルをアプリケーションのイベント・ログ・エントリと関連付けます。

ソースを指定する場合は、Eventlog\System キー、Eventlog\Security キー、Eventlog\Application レジストリ・キーのサブキーでなければなりません。たとえば、Myapp のソース名は、次のキーのレジストリ・エントリを示します。

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\
  Services\Eventlog\Application\Myapp)
```

Myapp レジストリ値 EventMessageFile は、このアプリケーションのイベントを変換するために使用されるメッセージ・ファイルとパスの名前を指定します。

ソースは指定しないか、または Null として指定できます。この場合、イベントはアプリケーション・ログ・ファイルに記録されますが、アプリケーションは関連イベントのメッセージ・ファイルを記録しません (この結果、置換テキストも記録されません)。

• EVTS\_HANDLE

アプリケーション・イベント・ログへのハンドルを返します。このハンドルは他の \$EVENT 機能への入力として必要です。

エラーが発生した場合、ハンドルとして 0 が返されます。このハンドルは CloseHandle API の責任の範囲外です。

EVT\$\_FC\_REPORT\_EVENT

イベント・ログ・エントリを生成します。

アイテム・コード	必須	パラメータ	データ・タイプ
EVT\$_HANDLE	必須	入力	符号なしロングワード
EVT\$_EVENT_TYPE	必須	入力	ワード・マスク
EVT\$_EVENT_CATEGORY	省略可能	入力	ワード
EVT\$_EVENT_ID	必須	入力	ロングワード
EVT\$_USER_SID	省略可能	入力	NT セキュリティ ID
EVT\$_NUMSTRINGS	省略可能	入力	ワード
EVT\$_DATASIZE	省略可能	入力	ロングワード
EVT\$_STRING_ARRAY	省略可能	入力	可変長記述子の配列 (4 バイトの Unicode)
EVT\$_RAW_DATA	省略可能	入力	バイナリ・データ

• EVTS\_HANDLE

以前の EVTS\_FC\_REGISTER\_EVENT\_SOURCE 呼び出しから返された値。

- EVTS\_EVENT\_TYPE

イベントの重大度を示します。タイプは次のいずれかです。

EVT\$\_SUCCESS  
EVT\$\_ERROR  
EVT\$\_WARNING  
EVT\$\_INFO  
EVT\$\_AUDIT\_SUCCESS  
EVT\$\_AUDIT\_FAILURE

重大度タイプは、WINNT.H に定義されている Windows NT の対応する重大度にマップされます。

- EVTS\_EVENT\_CATEGORY

1 ~ 65535 の範囲の整数値。EVTS\_EVENT\_CATEGORY は特定のソース固有の値です。

EVTS\_EVENT\_CATEGORY を使用すると、アプリケーションはメッセージ・ファイルをセクションに分割できます。各セクションにはイベント ID がインデックスとして付加されます。カテゴリを指定しなかった場合、デフォルトは 0 になります。

- EVTS\_EVENT\_ID

無制限の整数値。この値はアプリケーション・メッセージ・ファイルでカテゴリのインデックスとして使用されます。インデックスはこのイベント・メッセージに対して表示されるテキスト文字列の場所を示します。イベント ID は特定のソース固有の値です。

- EVTS\_USER\_SID

イベントをログに記録するスレッドの Windows NT セキュリティ ID (省略可能)。\$PERSONA システム・サービスを介して Windows NT 資格情報を取得したアプリケーションは、OpenProcessToken と GetTokenInformation Win32 API を呼び出すことにより、SID を取得できます。形式はこのサービスからは不明です。

- EVTS\_NUMSTRINGS

EVT\$\_STRING\_ARRAY アイテム・コードに指定された文字列の数。

- EVTS\_DATASIZE

EVT\$\_RAW\_DATA アイテム・コードによって示されたバッファの長さ (バイト数)。

- EVTS\_STRING\_ARRAY

文字列ポインタの配列。各エントリ・ポイントは最後が Null の文字列です。メッセージ・ファイル内の記述文字列には、%n の形式で文字列プレースホルダを格納できます。%1 は最初のプレースホルダを示します。この配列に指定された文字列は、システムがイベント・メッセージを表示するときに、これらのプレースホルダを置き換えます。

- EVT\$\_RAW\_DATA

イベント・メッセージにバイナリ・データを含むことを許可します。

たとえば、エラーが発生したコンポーネントからデータ構造をダンプできます。

EVT\$\_DEREGISTER\_EVENT\_SOURCE

イベント・ログとの関連付けを閉じます。

アイテム・コード	必須	パラメータ	データ・タイプ
EVT\$_HANDLE	必須	入力	符号なしロングワード

- EVT\$\_HANDLE

以前の EVT\$\_FC\_REGISTER\_EVENT\_SOURCE 呼び出しから返された値。

---

## アイテム・コード

アイテム・コード	パラメータ・タイプ	データ・タイプ
EVT\$_SERVER_NAME	入力	文字列
EVT\$_SOURCE	入力	文字列
EVT\$_HANDLE	入力/出力	符号なしロングワード
EVT\$_EVENT_TYPE	入力	ワード・マスク
EVT\$_EVENT_CATEGORY	入力	ワード
EVT\$_EVENT_ID	入力	ロングワード
EVT\$_USER_SID	入力	NT セキュリティ ID
EVT\$_NUMSTRINGS	入力	ワード
EVT\$_DATASIZE	入力	ロングワード
EVT\$_STRING_ARRAY	入力	文字列ポインタの配列
EVT\$_RAW_DATA	入力	バイナリ・データ

---

## 説明

NTA\$EVENTW ルーチンを使用すると、イベントのソースの登録と登録の解除、イベント・データの報告を行うことができます。このイベント・ログ機能を使用すると、アプリケーションの内部から情報を記録できます。イベント・ルーチンを使用して、アプリケーション内で進行状況を追跡したり、アプリケーションで検出された問題を識別できます。

NTA\$EVENTW ルーチンは同期的に終了します。つまり、要求が終了した後で制御は呼び出しプロセスに戻されます。

イベント・データを書き込むには、次の操作を行います。

1. イベント・ソースを登録します。

この操作は、イベント・データが書き込まれるイベント・ログを定義します。

2. イベントを報告します。

この操作を実行すると、情報が適切なイベント・ログに書き込まれます。

3. イベント・ソースの登録を解除します。

この操作は、イベント・ソースの登録の一部として取得されたリソースを解放します。

---

## 返される状態値

SS\$_NORMAL	サービスは正常終了しました。
SS\$_ACCVIO	引数の 1 つを読み込み/書き込みすることができません。
SS\$_BADPARAM	パラメータが不正です。
SS\$_NOPRIV	特権が不足しているため、指定されたイベント・ログにアクセスできません。
SS\$_TIMEOUT	要求がタイムアウトしました。
SS\$_UNREACHABLE	イベント・サービスを使用できません。
SS\$_REJECT	Windows NT LAN Manager サーバがエラーを検出しました。詳細については、Win32 の状態情報を参照してください。

---

## 11.3 独自のイベントの作成

デフォルト設定では、システムは COM for OpenVMS で生成された DCOM イベントをログに記録します。COM for OpenVMS イベントを記録する他に、ユーザが作成した COM アプリケーションの COM アプリケーション・イベントをログに記録することもできます。

COM for OpenVMS キットには、Win32 API を使用してアプリケーション・イベントを生成する方法を示すサンプル・コードが含まれています。このサンプル・コードは、Windows NT システムでそのまま使用できます。また、OpenVMS で COM for OpenVMS アプリケーションを作成する指示に従って、このサンプル・コードを正しく作成することもできます (必要なヘッダ・ファイルは DCOM\$LIBRARY から取得してください)。これらの手順については、第 6 章を参照してください。また、このサンプル・コードには、Wind/U を使用して例を作成するためのリンク手順も示されています。

---

## 11.4 OpenVMS Events の問題への対処

イベントを報告するときに発生したエラーは、追跡が困難なことがあります。これは、途中にあるいくつかのソフトウェア・レイヤをイベントが通過するからです。ここでは、OpenVMS Events が Windows NT ログに記録されるまでに、他のソフトウェア・レイヤをどのように通過するかについて説明します。

1. アプリケーションは Win32 イベント機能のいずれか (RegisterEventSource, ReportEvent, DeregisterEventSource) を呼び出します。
2. Win32 API は指定された引数を使用して、適切なアイテム・リストを作成し、NTASEVENTW ルーチン呼び出します。
3. NTASEVENTW ルーチンは、指定された情報 (機能コード、アイテム・リストなど) を確認し、SYSSACM システム・サービスに対して適切なアイテム・リストを作成します。

NTASEVENT がエラーを検出すると、R0 とイベント状態ブロックを使用して、そのエラーを Win32 API に返します。

4. SYSSACM システム・サービスは情報を確認し、その情報を NT ACME に渡します。

SYSSACM がエラーを検出すると、R0 と ACM 状態ブロックを使用して、そのエラーを NTASEVENTW に返します。

5. NT ACME は指定された情報を (IPC パイプを使用して) Advanced Server for OpenVMS のディスパッチャに渡します。

NT ACME がエラーを検出すると、ACM 状態ブロックを使用して、そのエラーを呼び出し側プロセスに返します。

6. Advanced Server for OpenVMS ディスパッチャは情報を確認し、要求された操作 (登録, レポート, 登録の解除) を実行するために適切なルーチン呼び出しをします。

Advanced Server for OpenVMS がエラーを検出すると、そのエラーを NT ACME に報告します。NT ACME はエラーを他の呼び出しプロセスに渡します。

イベント状態ブロックの内容を確認すると、エラーがどこで発生したのかを判断するのに役立ちます。表 11-1 は、実行しなければならないチェックを (重要な順に) 示しています。

表 11-1 OpenVMS Events エラーへの対処法

R0 状態	状態フィールド値	チェックするコンポーネント
エラー (ビット 0 がクリア)	EVT\$SL_NT_STATUS フィールドが 0 以外の値である。	エラーは Advanced Server for OpenVMS の内部で発生している可能性が最も高い。
エラー	EVT\$SL_VMS_STATUS フィールドが 0 以外の値で、EVT\$SL_NT_STATUS が 0 である。	エラーは SYSSACM システム・サービスまたは NT ACME の内部で発生している可能性が最も高い。
エラー	EVT\$SL_VMS_STATUS が 0 で、EVT\$SL_NT_STATUS が 0 である。	エラーは SYSSACM システム・サービスの内部で発生している可能性が最も高い。

#### 注意

Win32 API は通常、エラー状態を適切な NT エラー状態コードに変換し、GetLastError Win32 API を介して提供します (イベント API から返された状態は単に一般的なエラーを示します)。

# 第4部

---

## 認証

第4部では、プラットフォーム間でのユーザとアプリケーションの認証に関する情報について説明します。



---

## 12.1 認証とは

認証とは、システムへのアクセスを許可する前に、コンピュータ・システムがユーザの身元を確認する動作です。ユーザの認証が成功すると、システムはユーザの認可情報を資格情報の形式でユーザのプロセスにバインドします。システムはこれらの資格情報を使用して、システム・リソースへのアクセスを許可するのか、拒否するのかを判断します。

OpenVMS では、次に示すように、ネイティブ (SYSUAF ベース) と Windows NT 互換の認証および認可機能が提供されます。

- ネイティブ：認証は、SYSUAF.DAT ファイルに格納されているパスワード情報を使用して行われます。認可情報は UIC、特権、ライト識別子で構成されます。
- Windows NT：認証は、ドメイン・コントローラが管理する SAM データベースに格納されているパスワード情報を使用して行われます。認可情報は SAM データベースのユーザ・アカウント情報から取得したプライマリ SID、グループ SID、セッション・キー、特権で構成されます。

OpenVMS がユーザの認証 (ネイティブまたは Windows NT) に成功すると、ペルソナと呼ぶ構造を使用して、ユーザのネイティブな資格情報がプロセスに結びつけられます。Windows NT を認証のために使用した場合は、OpenVMS はユーザの Windows NT 資格情報もプロセスに結合します (ペルソナの拡張として)。

---

## 12.2 NTA\$LOGON の使用による Windows NT 資格情報の取得

NTA\$LOGON は NTLM 資格情報を取得するためのユーティリティです。OpenVMS Registry または COM for OpenVMS 機能にアクセスするために、Windows NT セキュリティを必要とするすべてのプロセスは、NTLM 資格情報を必要とします。

NTA\$LOGON に対してユーザ・アカウント名、パスワード、(必要な場合は) ドメイン名を提供しなければなりません。NTA\$LOGON は ACM (Authentication and Credential Management) オートリティを使用してドメイン・コントローラに連絡をとり、Windows NT アクセス・トークンを取得します。NTA\$LOGON は Windows NT 情報とユーザの OpenVMS 資格情報をマージします。

NTA\$LOGON の依存関係の詳細と、NTA\$LOGON が OpenVMS インフラストラクチャの他の部分とどのように関係するかについては、第 4.8 節と第 4.9 節 (特に ACME サーバと Advanced Server for OpenVMS server) を参照してください。

NTA\$LOGON ユーティリティを使用するには、次のいずれかを入力します。

- 次のコマンドを入力して NTA\$LOGON ユーティリティを実行します。

```
$ RUN SYS$SYSTEM:NTA$LOGON
```

ユーザ・アカウント名とパスワードが求められます。

- NTA\$LOGON を使用してコマンド・ラインを解析するために、DCL シンボルを定義します。次の例を参照してください。

```
$ NTLOGON ::= $NTA$LOGON
$ NTLOGON
```

コマンド・ラインにパラメータを指定できます。表 12-1 はコマンド・ライン・パラメータを示しています。パラメータを指定しなかった場合は、必須情報が求められます。

- NTA\$LOGON を使用してコマンド・ラインを解析するために、MCR コマンドを使用します。次の例を参照してください。

```
$ MCR NTA$LOGON
```

コマンド・ラインにパラメータを指定できます。表 12-1 はコマンド・ライン・パラメータを示しています。パラメータを指定しなかった場合は、必須情報が求められます。

表 12-1 は NTA\$LOGON ユーティリティのコマンド・ライン・パラメータを示しています。

表 12-1 NTA\$LOGON ユーティリティのコマンド・ライン・パラメータ

引数	値	必須/省略可能
P1	ユーザ・アカウント名。アカウント名は必須であるが、コマンド・ラインに指定しなかった場合は、NTA\$LOGON から入力求められる。	省略可能
P2	パスワード。パスワードは必須であるが、コマンド・ラインに指定しなかった場合は、NTA\$LOGON から入力求められる (入力したパスワードは画面に表示されない)。	省略可能

例 12-1 は、資格情報を取得するための典型的な NTA\$LOGON セッションを示しています。

## 例 12-1 NTA\$LOGON セッションの例

```
$ NTLOGON ::= $NTA$LOGON
$ NTLOGON joesmith
Password:
```

---

**注意**

---

Windows NT ドメイン名とユーザ・アカウント名では、大文字と小文字が区別されません。NTA\$LOGON はすべてのドメイン名とユーザ・アカウント名を大文字に変換します。コマンド・ラインにパスワードを指定すると、DCL はすべての文字を大文字に変換します。ただし、パスワードを引用符 (") で囲んだ場合は、大文字に変換されません。

---

## 12.2.1 NTA\$LOGON の省略可能な修飾子

NTA\$LOGON では次の修飾子を使用できますが、これらの修飾子は省略可能です。

- /DELETE

現在の Windows NT 資格情報を削除します。

/DELETE修飾子を/WRITE\_FILE修飾子と組み合わせて指定すると、指定したドメイン名とユーザ・アカウント名のパスワード・レコードがファイルから削除されます。

- /DOMAIN=*name*

ドメイン名を指定します。この修飾子は名前を大文字に変換します。この修飾子を指定しなかった場合は、デフォルトのログイン名が使用されます。

- /LIST

現在のプロセスに割り当てられているドメイン名とユーザ・アカウント名の一覧を表示します。

/LIST修飾子を/READ\_FILEまたは/WRITE\_FILE修飾子と組み合わせて使用した場合は、ファイルの内容の一覧が表示されます。

- /LOG

操作が完了したときに、そのことを示すメッセージを表示します。

- /OVERRIDE\_MAPPING

プロセスの OpenVMS ユーザ名が、ドメイン・コントローラで Windows NT ユーザ・アカウント名に対応付けられている OpenVMS ユーザ名と一致しない場合でも、指定された Windows NT ユーザ・アカウント名の Windows NT 資格情報を取得します。

この修飾子を使用するには、IMPERSONATE 特権が必要です。

- /READ\_FILE [=file]

この修飾子を指定すると、ユーザ入力装置からパスワードを読み込むのではなく、/WRITE\_FILE修飾子で作成されたバイナリ入力ファイルから指定されたドメイン名とユーザ・アカウント名が検索されます。/READ\_FILE修飾子では、NTA\$LOGON/WRITE\_FILE コマンドで作成されたバイナリ・ファイルだけがサポートされます。

一致するレコードが検索されると、NTA\$LOGON はそのパスワードを使用して Windows NT 資格情報を取得します。

ファイル指定を省略した場合は、次のデフォルトのファイル指定が使用されません。

```
DCE$COMMON:[000000]NTA$LOGON.DAT
```

- /TYPE={BATCH | DIALUP | LOCAL | NETWORK | REMOTE}

アクセスを許可または拒否するための規則を指定します。この修飾子を指定しなかった場合は、現在のプロセスのタイプがデフォルトとして使用されます。この修飾子は通常、独立プロセスに対して使用されます (独立プロセスにはデフォルト・タイプがありません)。

この修飾子を使用するには、IMPERSONATE 特権が必要です。

- /WRITE\_FILE [=file]

この修飾子を指定すると、指定されたドメイン名、ユーザ・アカウント名、パスワードが後で使用できるように出力ファイルに書き込まれます (/READ\_FILE修飾子を参照)。この場合、ユーザが指定したパスワードは使用されません。

ファイル指定を省略した場合は、次のデフォルトの場所とファイル名が使用されます。

```
DCE$COMMON:[000000]NTA$LOGON.DAT
```

---

### 重要

---

/READ\_FILE修飾子と/WRITE\_FILE修飾子は、OpenVMS Registry または COM for OpenVMS 機能にアクセスするために Windows NT 資格情報を取得するための方法が他にないサーバで使用することを目的としています。/READ\_FILE修飾子と/WRITE\_FILE修飾子を一般に使用することは望ましくありません。

パスワードをディスクのファイルに書き込んだ後、パスワード・ファイルが不正にアクセスされないようにするために、ファイルは厳重に保護してください。

---

## 12.2.2 Windows NT 資格情報を取得するための NTA\$LOGON の使用例

例 12-2 は、ユーザが NT 資格情報を初めて取得する方法を示しています。

### 例 12-2 Windows NT 資格情報の初めての取得

```
$ NTLOGON ::= $NTA$LOGON
$ NTLOGON/LIST
ERROR: NtOpenProcessToken() failure: -1073741700 0xc000007c
%SYSTEM-E-NOSUCHEXT, no such extension found

$ NTLOGON/LOG JOESMITH
[Persona #1 NT extension: Account= "JOESMITH" Domain= "NT_DOMAIN" ]
Password:
```

例 12-3 は、ユーザが Windows NT 資格情報を置換する方法を示しています。

### 例 12-3 Windows NT 資格情報の置換

```
$ NTLOGON/DELETE
$ NTLOGON/OVERRIDE_MAPPING/DOMAIN=OTHER_DOMAIN
Username: janebrown
Password:
```

例 12-4 は、ユーザがパスワードをディスクのファイルに保存する方法を示しています。パスワードは 2 回入力するように求められますが、入力しても画面に表示されません。

### 例 12-4 ファイルへのパスワードの保存

```
$ NTLOGON ::= $NTA$LOGON
$ NTLOGON/WRITE_FILE=DEV:[DIR]NTA$LOGON.DAT COM_SERVER
Password:
Confirm:
$ NTLOGON/READ_FILE=DEV:[DIR]NTA$LOGON.DAT/LIST
File DEV:[DIR]NTA$LOGON.DAT contains the following records:
02-MAR-1999 16:57:23.20 COM_SERVER
```

このファイルを作成した後、DCL コマンド・プロシージャに次のコマンドを追加できます。

```
$ NTLOGON ::= $NTA$LOGON
$ NTLOGON/READ_FILE=DEV:[DIR]NTA$LOGON.DAT COM_SERVER
```

---

## 12.3 ACM (Authentication and Credential Management) オーソリティ

ACM (Authentication and Credential Management) オーソリティは、OpenVMS と Windows NT に対してユーザを認証し、ユーザのセキュリティ・プロファイルを判断します。ACME\_SERVER プロセスはこれらの ACM サービスを提供します。ACME\_SERVER プロセスは ACME エージェントと呼ぶプラグイン・モジュールを使用します。ACME エージェントは、認証要求、問い合わせ要求、イベント要求への応答などの実際の作業を行います。

OpenVMS ACME エージェント (VMSS\$VMS\_ACMESHR.EXE) は、OpenVMS ネイティブ・サービスを提供します。MSV1\_0 ACME エージェント (PWRK\$MSV1\_0\_ACMESHR.EXE, Advanced Server for OpenVMS 製品のコンポーネント) は、Windows NT 接続サービスを提供します。

MSV1\_0 ACME エージェントは Windows NT 接続サービス要求を NTAS\$LOGON および SSPI/NTLM から、クラスタ内の 1 つ以上のシステムで稼動している Advanced Server for OpenVMS プロセスに転送します。PWRK\$ACME\_SERVER 論理名には、MSV1\_0 ACME が要求を転送する先のクラスタ・ノード名のリストを含むことができます。各名前はカンマで区切ります。Advanced Server for OpenVMS プロセスを 1 つ以上のクラスタ・ノードで実行し、ノード名を PWRK\$ACME\_SERVER 論理名に含むと、MSV1\_0 ACME エージェントは、接続が中断されたときに、要求を自動的にフェールオーバーできます。論理名が定義されていない場合は、デフォルトでローカル・マシン名に設定されます。

ACME\_SERVER プロセスは、RPC または COM for OpenVMS を実行するどのシステムにも存在しなければなりません。しかし、Advanced Server for OpenVMS プロセスは、クラスタ内の 1 つのノードにだけ存在すれば十分です。

### 12.3.1 OpenVMS での Windows NT 認証

ACME\_SERVER は要求された Windows NT ペルソナ拡張情報も含めて、完全な OpenVMS ペルソナを呼び出しプロセスに返すので、VMS ACME エージェントは次の規則を適用します。

- すべての Windows NT ユーザはローカルの OpenVMS ユーザ名に変換されなければなりません。

MSV1\_0 ACME は Advanced Server for OpenVMS HOSTMAP データベースを介して、このマップ機能を提供します。

- マップされた OpenVMS ユーザ名は、SYSUAF.DAT で有効な (無効にされていない) アカウントでなければなりません。そのアカウントのアクセス制限で、指定された日時にアクセスが許可されなければなりません。COM for OpenVMS と RPC では通常、認証時に NETWORK アクセス権を必要とします。

- マップされた OpenVMS ユーザ名は、EXTAUTH フラグがセットされたアカウントでなければなりません。EXTAUTH フラグがセットされていると、システム管理者は、マップのためにどの OpenVMS アカウントを使用できるかを細かく制御できます。SECURITY\_POLICY システム・パラメータで IGNORE\_EXTAUTH ビット (ビット番号 11 [10 進数]) を使用すると、このアカウント単位の機能を無効にできます。IGNORE\_EXTAUTH ビットを 1 にセットすると、OpenVMS はアカウントの EXTAUTH の設定とは無関係に、どのアカウントにもマップできるようになります。IGNORE\_EXTAUTH は ACME\_SERVER に対してだけ使用され、Logout では無視されます。

### 12.3.2 Windows NT ユーザの認証; OpenVMS プロセスへの Windows NT 資格情報の付与

OpenVMS で Windows NT ユーザを認証し、Windows NT ユーザの資格情報を OpenVMS プロセスに割り当てるには、次の操作を行います。

1. Advanced Server for OpenVMS データベースで HOSTMAP エントリを定義します。

HOSTMAP エントリは、Windows NT ユーザ・アカウントとローカルの OpenVMS ユーザ・アカウントの間の関連付けを定義します。OpenVMS が Windows NT ユーザを認証する場合、HOSTMAP エントリを使用して、OpenVMS ユーザ・アカウントを Windows NT ユーザ・アカウントにマップし、ローカルの OpenVMS ユーザ・プロファイルと Windows NT ユーザ・プロファイルを作成します。HOSTMAP エントリが存在しない場合は、OpenVMS は Windows NT ユーザ・アカウント名をローカルの OpenVMS ユーザ・アカウント名として使用します。

Advanced Server for OpenVMS の ADMINISTER ユーティリティを使用して、HOSTMAP 情報を定義します。たとえば、Windows NT の FRED というユーザをローカルの OpenVMS ユーザ・アカウント FREDERICK にマップするには、次のコマンドを入力します。

```
$ ADMINISTER ADD HOSTMAP FRED FREDERICK
```

FRED が信頼されるドメイン FINANCE に定義されている場合は、信頼されるドメイン名を HOSTMAP エントリに指定しなければなりません。次の例を参照してください。

```
$ ADMINISTER ADD HOSTMAP FINANCE\FRED FREDERICK
```

2. ホストマップを行うために OpenVMS ユーザ・アカウントを有効にします。

デフォルト設定では、OpenVMS は、ホストマップのために EXTAUTH フラグがセットされた OpenVMS ユーザ・アカウントだけを使用できます。このポリシーにより、OpenVMS システム管理者は Windows NT 認証でどの OpenVMS ユーザ・アカウントを使用できるかを制御できます。

( SECURITY\_POLICY システム・パラメータの IGNORE\_EXTAUTH ビット[ビット番号 11 (10 進数)]をセットすれば、このアカウント単位の制限をシステム全体で無効にできます。)

OpenVMS ユーザ・アカウントで EXTAUTH フラグをセットするには、AUTHORIZE ユーティリティを使用します。たとえば、OpenVMS ユーザ・アカウント FREDERICK の EXTAUTH フラグを有効にするには、次のコマンドを入力します。

```
$ AUTHORIZE MODIFY FREDERICK/FLAG=EXTAUTH
```

### 12.3.3 ACME\_SERVER プロセスの管理 (ACME サーバ・コマンド)

システムの起動時に ACME\_SERVER プロセスを起動し、MSV1\_0 ACME エージェントを構成するには、次のエントリを SYLOGICALS.COM に追加します。

```
$ DEFINE NTA$NT_ACME_TO_BE_STARTED YES
```

また、次のスタートアップ・コマンド・ファイルを使用して、ACME\_SERVER プロセスを手動で起動することもできます。

```
$ @SYS$STARTUP:NTA$STARTUP_NT_ACME
```

ACME\_SERVER をシャットダウンするには、次のコマンドを入力します。

```
$ SET SERVER ACME/EXIT
```

ACME エージェントで異常状態が発生したために、正常にサーバをシャットダウンできない場合は、/EXIT 修飾子の代わりに /ABORT 修飾子を使用して、ACME\_SERVER を強制終了します。

ACME\_SERVER のログ機能を有効にするには、次のコマンドを入力します。

```
$ SET SERVER ACME/LOG
```

このコマンドは ACME\$SERVER.LOG ファイルを SYSS\$MANAGER ディレクトリに作成します。このファイルは、発生する可能性のある問題を診断するのに役立ちます。

ACME\_SERVER 構成情報を表示するには、次のコマンドを入力します。

```
$ SHOW SERVER ACME[/FULL]
```

### 12.3.4 MSV1\_0 ACME エージェントの構成

表 12-2 は、MSV1\_0 ACME エージェントの特定の機能を制御するために使用できるシステム単位の論理名を示しています。

表 12-2 MSV1\_0 ACME エージェントの論理名

論理名	説明
PWRK\$ACME_SERVER	Advanced Server for OpenVMS プロセスで実行されていて、Windows NT 接続要求をサービスできるクラスタの SCS ノード名の一覧。各名前はカンマで区切る。ノード名を定義しなかった場合は、MSV1_0 ACME エージェントはローカル・システムの Advanced Server for OpenVMS プロセスに接続する。
PWRK\$ACME_RETRY_COUNT	Advanced Server for OpenVMS プロセスに接続するときに、MSV1_0 ACME エージェントが実行するリトライの最大回数。デフォルト値は 10 である。
PWRK\$ACME_RETRY_INTERVAL	リトライを実行する間隔であり、1/10 秒単位で指定する。デフォルトは 2.5 秒である。



# 第5部

---

## 付録

第5部では、COM for OpenVMS および OpenVMS Registry についての関連情報をまとめます。

各付録は、MIDL コンパイラ、トラブルシューティングについて、COM サンプル・コード、API およびインタフェースについて説明します。

また、COM 関連の書籍のクーポン券、用語集、略語（頭字語）も用意しています。



# A

## MIDL コンパイラのオプション

### A.1 モード

スイッチ	機能
/ms_ext	IDL 言語に対する Microsoft の拡張機能 (デフォルト)。
/c_ext	IDL ファイルで Microsoft C 拡張機能を可能にする (デフォルト)。
/osf	OSF モード - /ms_ext オプションと/c_ext オプションを無効にする。
/app_config	IDL ファイルで選択された ACF 属性を可能にする。
/mktyplib203	MKTYPLIB バージョン 2.03 互換性モード。

### A.2 入力

スイッチ	機能
/acf filename	属性構成ファイルを指定する。
/I directory-list	インクルード・パスに対して 1 つ以上のディレクトリを指定する。
/no_def_idir	カレント・ディレクトリと INCLUDE ディレクトリを無視する。

### A.3 出力ファイルの生成

スイッチ	機能
/client none	クライアント・ファイルを生成しない。
/client stub	クライアント・スタブ・ファイルだけを生成する。
/out directory	出力ファイルを生成するディレクトリを指定する。
/server none	サーバ・ファイルを生成しない。
/server stub	サーバ・スタブ・ファイルだけを生成する。
/syntax_check	構文の確認だけを行い、出力ファイルは生成しない。
/Zs	構文の確認だけを行い、出力ファイルは生成しない。
/old	以前の形式のタイプ・ライブラリを生成する。
/new	新しい形式のタイプ・ライブラリを生成する。

---

## A.4 出力ファイル名

---

スイッチ	機能
/cstub filename	クライアント・スタブ・ファイル名を指定する。
/dlldata filename	dlldata ファイル名を指定する。
/h filename	ヘッダ・ファイル名を指定する。
/header filename	ヘッダ・ファイル名を指定する。
/iid filename	インタフェース UUID ファイル名を指定する。
/proxy filename	プロキシ・ファイル名を指定する。
/sstub filename	サーバ・スタブ・ファイル名を指定する。
/tlb filename	タイプ・ライブラリ・ファイル名を指定する。

---

---

## A.5 C コンパイラとプリプロセッサのオプション

---

スイッチ	機能
/cpp_cmd cmd_line	C プリプロセッサの名前を指定する。
/cpp_opt options	C プリプロセッサの追加オプションを指定する。
/D name[=def]	#define の名前と省略可能な値を C プリプロセッサに渡す。
/no_cpp	C プリプロセッサ・オプションを無効にする。
/nocpp	C プリプロセッサ・オプションを無効にする。
/U name	以前の定義を削除する (未定義)。

---

---

## A.6 環境

---

スイッチ	機能
/char signed	C コンパイラのデフォルト char タイプは符号付きである。
/char unsigned	C コンパイラのデフォルト char タイプは符号なしである。
/char ascii7	char の値は 0 ~ 127 に制限される。
/dos	ターゲット環境は MS-DOS クライアントである。
/env dos	ターゲット環境は MS-DOS クライアントである。
/env mac	ターゲット環境は Apple Macintosh である。
/env powermac	ターゲット環境は Apple PowerMac である。
/env win16	ターゲット環境は Microsoft Windows 16 ビット (Win 3.x) である。
/env win32	ターゲット環境は Microsoft Windows 32 ビット (NT) である。
/mac	ターゲット環境は Apple Macintosh である。
/ms_union	カプセル化されていないユニオンに対して、Midl 1.0 非 DCE ワイヤ・レイアウトを使用する。
/oldnames	バージョン番号を名前に変換しない。
/powermac	ターゲット環境は Apple PowerMac である。
/rpcss	rpc_sm_enable_allocate を自動的にアクティブにする。

スイッチ	機能
/use_epv	entry-pt ベクタを介してサーバ側のアプリケーション呼び出しを生成する。
/no_default_epv	デフォルトのエントリ・ポイント・ベクタを生成しない。
/prefix client str	“str”接頭辞をクライアント側のエントリ・ポイントに追加する。
/prefix server str	“str”接頭辞をサーバ側のマネージャ・ルーチンに追加する。
/prefix switch str	“str”接頭辞をスイッチ・ルーチン・プロトタイプに追加する。
/prefix all str	“str”接頭辞をすべてのルーチンに追加する。
/win16	ターゲット環境は Microsoft Windows 16 ビット (Win 3.x) である。
/win32	ターゲット環境は Microsoft Windows 32 ビット (NT) である。

## A.7 エラー・メッセージと警告メッセージ

スイッチ	機能
/error none	すべてのエラー・チェック・オプションを無効にする。
/error allocation	out of memory エラーをチェックする。
/error bounds_check	サイズと伝送長の指定をチェックする。
/error enum	enum の値が有効な範囲であるかどうかをチェックする。
/error ref	ref ポインタが Null 以外の値であるかどうかをチェックする。
/error stub_data	サーバ側のスタブ・データが有効であるかどうか確認するために、追加チェックを行う。
/no_warn	コンパイラの警告メッセージを出力しない。

## A.8 最適化

スイッチ	機能
/align {1 2 4 8}	構造のパック・レベルを指定する。
/pack {1 2 4 8}	構造のパック・レベルを指定する。
/Zp{1 2 4 8}	構造のパック・レベルを指定する。
/Oi	完全に解釈されたスタブを生成する。
/Oic	NT 3.51 リリースの、標準インタフェースに対しては完全に解釈されたスタブを生成し、オブジェクト・インタフェースに対してはスタブレス・プロキシを生成する。
/Oicf	NT 4.0 リリースの、拡張機能を含む完全に解釈されたスタブを生成し、オブジェクト・インタフェースに対してはスタブレス・プロキシを生成する。
/Os	インライン・スタブを生成する。
/hookole	ローカル・オブジェクト・インタフェースに対して HookOle デバッグ情報を生成する。

---

## A.9 その他

---

スイッチ	機能
@response_file	応答ファイルから入力を受け付ける。
/?	MIDL コンパイラ・スイッチの一覧を表示する。
/confirm	MIDL ソースをコンパイルせずにオプションを表示する。
/help	MIDL コンパイラ・スイッチの一覧を表示する。
/nologo	バナー行を表示しない。
/o filename	出力を画面からファイルにリダイレクトする。
/W{0 1 2 3 4}	警告レベル 0 ~ 4 を指定する (デフォルト= 1)。
/WX	指定された/Wレベルの警告をエラーとして報告する。

---

---

## トラブルシューティング

---

### B.1 RPCのトラブルシューティング

同時に NTLM 認証を数多く行う場合は、次のエラーが発生する可能性があります。同時 NTLM 認証の数にはいくつかの要素が影響しますが、これらのエラーが発生する可能性が最も高いのは、ネットワークが混雑している場合か、RPC アプリケーション・サーバがタイムリーに要求に応答しない場合です。エラーは標準の RPC アプリケーション戻り値として返されます。

表 B-1 は、考えられる原因と対処方法を示しています。

表 B-1 RPC エラー

エラー	原因/対処方法
RPC_S_CONNECTION_REJECTED	<p>このエラーは、DECnet Phase IV または DECnet Phase V をトランスポートとして使用し、他の DECnet クライアントにサービスを提供するサーバの負荷が非常に高いときに、クライアント・アプリケーションで例外として発生します。</p> <p>クライアント RPC が新規作成されたソケットに対してタイム・バインドを実行し、ソケットがエラー 61 (connection refused) を返すと、このエラーが返されます。</p> <p>対処方法:</p> <ol style="list-style-type: none"><li>1. DECnet リソース・クォータの値を大きくします。</li><li>2. クライアント RPC プログラムを強化して例外をキャッチし、RPC を再実行するか、または別のサーバを選択するようにします。</li></ol>

---

(次ページに続く)

## トラブルシューティング

### B.1 RPC のトラブルシューティング

表 B-1 (続き) RPC エラー

エラー	原因/対処方法
RPC_S_CONNECTION_TIMED_OUT	<p>このエラーは、TCP または DECnet をトランスポートとして使用し、サーバの負荷が非常に高いときに、クライアント・アプリケーションで例外として発生します。</p> <p>クライアント RPC が新規作成されたソケットに対してタイム・バインドを実行し、サーバが接続を受け付けるか、または拒否するまでに非常に長い時間がかかる場合は、このエラーが返されません。</p> <p>対処方法:</p> <ol style="list-style-type: none"><li>1. ソケット・タイムアウトが発生するまでの待ち時間を長くするように、TCP または DECnet を構成します。</li><li>2. RPC クライアント・アプリケーションを強化して、このソケット・エラーを取得したときに、<code>rpc_mgmt_set_com_timeout()</code> を呼び出し、RPC 実行時ルーチンに対して再実行するように要求します。</li><li>3. 例外をキャッチし、RPC を再実行するか、または別のサーバを選択するように、クライアント RPC プログラムを変更します。</li></ol>
RPC_S_ASSOC_SHUTDOWN	<p>このエラーは、TCP または DECnet をトランスポートとして使用し、クライアントの負荷が非常に高いときに (通常、クライアントが RPC サーバでもある場合)、クライアント・アプリケーションで例外として発生します。</p> <p>RPC サーバが RPC_BIND パケットをクライアントから受信し、サーバが RPC_BIND_ACK パケットをクライアントに返送した後、サーバは 12 秒以内に REQUEST パケットを受信するものと解釈します。クライアントが 12 秒以内に REQUEST パケットを送信しなかった場合は、RPC サーバは関連付けを削除し、SHUTDOWN パケットをクライアントに送信します。クライアント RPC 実行時ルーチンは例外を RPC アプリケーションに通知します。</p> <p>このエラーは、クライアント RPC アプリケーションが RPC サーバとしても機能し、RPC サーバの負荷がすでに非常に高くなっているときに発生する可能性があります。</p> <p>対処方法:</p> <ol style="list-style-type: none"><li>1. 例外をキャッチし、RPC を再実行するか、または別のサーバを選択するように、クライアント RPC プログラムを実装します。</li></ol>

(次ページに続く)

表 B-1 (続き) RPC エラー

エラー	原因/対処方法
RPC_S_COMM_FAILURE	<p>このエラーは、DG (UDP) をトランスポートとして使用し、RPC サーバの負荷が非常に高いときに、クライアント・アプリケーションで例外として発生します。</p> <p>RPC クライアントはREQUESTパケットをサーバに送信します。クライアントが 30 秒以内にサーバからWORKINGパケット応答を受信しないと、クライアントはPINGパケットをサーバに送信して、サーバがまだアクティブであり、クライアントの要求を処理しているかどうかを判断します。RPC サーバの負荷が非常に高い場合は、クライアントでタイムアウトが発生する前に、サーバがWORKINGパケットをクライアントに返すことができないことがあります。</p> <p>対処方法:</p> <ol style="list-style-type: none"><li>1. RPC クライアント・アプリケーションは <code>rpc_mgmt_set_com_timeout()</code> を呼び出して、RPC 実行時ルーチンに対して、タイムアウトが発生する前に 30 秒以上待つように要求できます。</li><li>2. 例外をキャッチし、RPC を再実行するか、または別のサーバを選択するように、クライアント RPC プログラムを実装します。</li></ol>

## B.2 ACME サーバのトラブルシューティング

ACME サーバで発生した問題を解決するには、次の操作を行います。

1. ACME\_SERVER プロセスが稼動しているかどうかを確認し (SHOW SERVER ACME コマンドを使用)、MSV1\_0 ACME エージェントと Advanced Server for OpenVMS プロセスの間に接続が確立されているかどうかを確認します。
2. 接続が確立されていない場合は、PWRK\$ACME\_SERVER 論理名に、Advanced Server for OpenVMS プロセスを実行しているクラスタ内のシステムの SCS ノード名が含まれているかどうかを確認します。
3. PWRK\$ACME\_SERVER 論理名が正しく定義されている場合は、指定されたシステムで Advanced Server for OpenVMS プロセスが稼動しているかどうかを確認します (PWRK\$LMSRV プロセスを検索します)。
4. 認証が失敗した場合は、次のことを確認します。
  - ドメイン間認証 (EASTOSHKOSK\JOE) では、信頼関係が必要です。Advanced Server for OpenVMS の ADMINISTER ADD TRUST[/TRUSTED]または[/PERMITTED]コマンドを使用して、2 つのドメインの間に必要な信頼関係を確立します。
  - Windows NT のパスワードでは大文字と小文字が区別されます。大文字と小文字を正しく区別してパスワードを入力したかどうか確認してください。

## トラブルシューティング

### B.2 ACME サーバのトラブルシューティング

- Windows NT ユーザに OpenVMS hostmap アカウントが与えられていないか、または無効な OpenVMS アカウントにマップされます (デフォルトの UAF マップは PWRK\$DEFAULT であり、このアカウントでは DISUSER フラグがセットされています)。Advanced Server for OpenVMS ADMINISTER ADD HOSTMAP コマンドを使用して、Windows NT ユーザ名を有効な OpenVMS アカウントにマップします。
- Windows NT ユーザ・アカウントが不正であるか、有効期限が切れているか、無効であるか、不正なパスワードが割り当てられています。Advanced Server for OpenVMS ADMINISTER SHOW USER/FULL コマンドを使用して、完全なユーザ・アカウント情報を表示します。Advanced Server for OpenVMS の ADMINISTER SHOW ACCOUNT POLICY コマンドを使用して、ドメイン・ポリシー情報を表示します。
- OpenVMS アカウントで EXTAUTH フラグがセットされていません。AUTHORIZE で UAF ユーティリティの MODIFY *user-name* /FLAG=EXTAUTH コマンドを使用します (SECURITY\_POLICY システム・パラメータで IGNORE\_EXTAUTH ビット (ビット番号 11 [10 進数]) をセットすれば、この要件を無効にできます)。
- UAF レコード・フラグが DISUSER に設定されています。AUTHORIZE で、UAF ユーティリティの MODIFY *user-name*/FLAG=NODISUSER コマンドを使用します。
- UAF レコードのモダル制限により、“ログイン”が禁止されています (ローカル・ダイアルアップ、リモート・アクセス、ネットワーク・アクセス、バッチ・アクセスの制限を確認します)。AUTHORIZE で UAF ユーティリティの MOD *user-name*/LOCAL (または DIALUP, BATCH, NETWORK, REMOTE) キーワードを使用します。INTERACTIVE の場合は LOCAL, DIALUP, REMOTE アクセスの制限が設定されます。
- ログインの失敗の回数がシステムであらかじめ設定された回数 (SYSGEN の LGI\_BRK\_LIM パラメータによって設定された値) を超えたため、侵入サブシステムが侵入回避モードになりました。SHOW INTRUSION コマンドを使用して、侵入データベースを表示します。DELETE/INTRUSION *source* コマンドを使用して、エントリをデータベースから削除します。LGI\_BRK\_DISUSER がセットされている場合は、UAF レコードが DISUSER に設定されている可能性があります。OpenVMS の AUTHORIZE コマンドを使用して、フラグをリセットします。

---

## B.3 DCOM\$RPCSS プロセスのトラブルシューティング

OpenVMS システムで COM for OpenVMS アプリケーションを実行するには、DCOM\$RPCSS プロセスを稼動していなければなりません。DCOM\$STARTUP.COM コマンド・ファイルはこのプロセスを自動的に起動します。COM for OpenVMS アプリケーションの実行で問題が発生した場合は、このプロセスが稼動しているかどうかを確認してください。次のコマンドを使用します。

```
$ SHOW SYSTEM
```

プロセスが初期化中の場合は、プロセス名は DCOM\$STARTUP-\*\* になります。プロセスが正常な動作状態の場合は、プロセス名は DCOM\$RPCSS になります。

DCOM\$RPCSS プロセスからエラー・メッセージが記録されていないかどうか、SYSSMANAGER:DCOM\$RPCSS.OUT ログ・ファイルを確認してください。次のメッセージが記録されている可能性があります。

- %ACME-E-PWDEXPIRED, password has expired

DCOM\$RPCSS ログ・ファイルにこのエラーが記録されている場合は、次の操作を行います。

1. Advanced Server for OpenVMS ADMIN ユーティリティを実行し、DCOM\$RPCSS アカウントのパスワードを変更します。第 5.2.1 項を参照してください。
2. COM for OpenVMS Service Control Manager パスワード・ファイルを更新します。第 5.2.1 項を参照してください。



---

## クックブックの例: OpenVMS でのサンプル・アプリケーションの作成

---

### 注意

SAMPLE1とDISPATCH\_SAMPLE1は、Microsoft Press が出版している Dale Rogerson 著『*Inside COM*』から抜粋したものです。

---

---

### C.1 COMの例(サンプル1)

このサンプルでは、COM クライアントとサーバを実装します。コンポーネントはIXとIYという2つのインタフェースを提供します。また、クライアントはコンポーネントに対して3番目のインタフェースIZを問い合わせしていますが、これはコンポーネントが提供していないインタフェースです。

このサンプルでは、2つのOpenVMSシステム間の接続、2つのWindows NTシステム間の接続、OpenVMSシステムとWindows NTシステム間の接続を示しています。

---

### 注意

OpenVMSでアプリケーションを作成する前に、NTA\$LOGONを実行し、Windows NT資格情報を取得する必要があります。詳細については、第12.2節を参照してください。

---

#### C.1.1 OpenVMSでの手順

ここでは、OpenVMSシステムでアプリケーションを構築する方法について説明します。

##### C.1.1.1 OpenVMSでのアプリケーションの構築

ファイルをDCOM examples ディレクトリからローカル・ディレクトリにコピーします。次の例を参照してください。

```
$ set default mydisk:[mydirectory]
$ copy dcom$examples:[sample1]*.* []
```

## クックブックの例: OpenVMS でのサンプル・アプリケーションの作成 C.1 COM の例 (サンプル 1)

アプリケーションを構築するには、次のコマンド・プロシージャを実行します。

```
$ @build_sample1
```

MMS がある場合は、次に示すように、インクルードされた説明ファイルを使用できます。

```
$ MMS/DESCRIPTION=BUILD_SAMPLE1.MMS
```

BUILDファイルは、プロセス内サーバとプロセス外サーバの両方を構築し、登録します。

### C.1.1.2 OpenVMS でのアプリケーションの登録

ビルド・プロシージャは、CMPNT\$SHR.EXEとCMPNT.EXEの両方を自動的に登録します。コンポーネントを手動で登録するには、次の操作を行います。

- プロセス内サーバを登録するには、REGSVR32ユーティリティを使用します。次の例を参照してください。

```
$ regsvr32 := $DCOM$REGSVR32.EXE  
$ regsvr32 path-nameCMPNT$SHR.EXE
```

- プロセス内サーバの登録を解除するには、REGSVR32ユーティリティを使用します。次の例を参照してください。

```
$ regsvr32 /u path-nameCMPNT$SHR.EXE
```

- プロセス外サーバを登録するには、次の操作を行います。

```
$ cmpnt := $path-nameCMPNT.EXE  
$ cmpnt /regserver
```

- プロセス外サーバの登録を解除するには、次の操作を行います。

```
$ cmpnt /unregserver
```

- プロキシ・スタブを登録するには、REGSVR32ユーティリティを使用します。次の例を参照してください。

```
$ regsvr32 path-namePROXY$SHR.EXE
```

- プロキシ・スタブの登録を解除するには、REGSVR32ユーティリティを使用します。次の例を参照してください。

```
$ regsvr32 /u path-namePROXY$SHR.EXE
```

### C.1.1.3 プロセス外サーバとしての OpenVMS でのアプリケーションの実行

コンポーネントがプロセス外サーバであるサンプルを実行するには、CMPNT.EXEを実行します。コンポーネントからServer: Waitingというメッセージが表示されたら、クライアントを別のウィンドウまたは端末セッションで実行します。

- ウィンドウ (または端末セッション) 1:

```
$ run cmpnt
```

- ウィンドウ (または端末セッション) 2:

```
$ client := $path-nameCLIENT.EXE
For OutProc:
$ client
2
$
```

クライアントは次のメッセージを表示します。

```
To which server do you want to connect?
1) In-Process Server
2) Out-of-Process Server
:
```

2と入力して、プロセス外サーバを選択します。

### C.1.1.4 OpenVMS でのアプリケーションの実行とリモート・サーバの指定

リモート・マシン (またはサーバ・システム) として指定したシステムでCMPNT.EXEを実行します。リモート・システムはWindows NTシステムでもかまいません。コンポーネントからServer: Waitingメッセージが返された場合は、ローカル・マシン (またはクライアント・システム) として指定したシステムでクライアントを実行します。次の例を参照してください。

```
$ client := $path-nameCLIENT.EXE
$ client remote-system-name
2
$
```

クライアントは次のメッセージを表示します。

```
To which server do you want to connect?
1) In-Process Server
2) Out-of-Process Server
:
```

2と入力して、リモート・サーバ、つまりプロセス外サーバの実行を選択します。

クックブックの例: OpenVMS でのサンプル・アプリケーションの作成  
C.1 COM の例 (サンプル 1)

C.1.1.5 プロセス内サーバとしての OpenVMS でのアプリケーションの実行

コンポーネントがプロセス内サーバであるサンプルを実行するには、クライアントだけを実行します。次の例を参照してください。

```
For InProc:
$ client
1
$
```

クライアントは次のメッセージを表示します。

```
To which server do you want to connect?
1) In-Process Server
2) Out-of-Process Server
:
```

1と入力して、プロセス内サーバを選択します。

C.1.2 Windows NT の場合の手順

ここでは、Windows NT システムでアプリケーションを構築する方法について説明します。

C.1.2.1 Windows NT でのアプリケーションの構築

README-SAMPLE1.TXTファイルと次のファイルを COM examples ディレクトリから Windows NT システムにコピーします。

```
CLIENT.CXX
CMPNT.CXX
CMPNT.DEF
GUIDS.CXX
MAKE-ONE.
MAKEFILE.BAT
PROXY.DEF
REGISTRY.CXX
REGISTRY.H
SERVER.IDL
```

MAKEFILE.BATファイルを使用して、サンプルを構築します。次の例を参照してください。

```
> MAKEFILE
```

Makefile はプロセス内サーバとプロセス外サーバの両方を構築し、登録します。

### C.1.2.2 Windows NT でのアプリケーションの登録

ビルド・プロシージャmake-oneはCMPNT.DLL, PROXY.DLL, CMPNT.EXEを自動的に登録します。次の例を参照してください。

```
regsvr32 -s Cmpnt.dll
regsvr32 -s Proxy.dll
Cmpnt /RegServer
```

アプリケーションの登録を解除するには、次のコマンドを入力します。

```
regsvr32 -u Cmpnt.dll
regsvr32 -u Proxy.dll
Cmpnt /UnRegServer
```

### C.1.2.3 Windows NT でのアプリケーションの実行

CLIENTを実行します。OpenVMS でイン・プロセス・サーバとしてアプリケーションを実行する手順 (第 C.1.1.5 項), およびプロセス外サーバとしてアプリケーションを実行する手順 (第 C.1.1.3 項) と同じ手順を実行します。

リモート・マシンの名前 (UNC または DNS) を引数として使用して、リモート・マシンでオブジェクトのインスタンスを作成します。次の例を参照してください。

```
>Client hostname      ! point the client at the remote system
2                     ! means outproc invocation
>
```

---

## C.2 オートメーションの例 (Dispatch\_Sample1)

この例では、デュアル・インタフェースとしてオートメーション・コンポーネント・サーバを実装します。この例には、ディスク・インタフェースを介してデュアル・インタフェースに接続するDclientと、IUnknownインタフェース (v-table を使用) を介して接続する COM クライアント実装であるClientの 2 つのクライアントがあります。

このサンプルでは、2 つの OpenVMS システムの間の接続、2 つの Windows NT システムの間の接続、OpenVMS システムと Windows NT システムの間の接続を示します。

### C.2.1 OpenVMS の場合の手順

ここでは、OpenVMS システムでアプリケーションを構築する方法について説明します。

## クックブックの例: OpenVMS でのサンプル・アプリケーションの作成 C.2 オートメーションの例 (Dispatch\_Sample1)

### C.2.1.1 OpenVMS でのアプリケーションの構築

ファイルをDCOM `examples` ディレクトリからローカル・ディレクトリにコピーします。次の例を参照してください。

```
$ set default mydisk:[mydirectory]
$ copy dcom$examples:[dispatch_sample1]*.* []
```

アプリケーションを構築するには、次のコマンド・プロシージャを実行します。

```
$ @build_dispatch_sample1
```

MMS がある場合は、次の例に示すように、インクルードされた説明ファイルを使用できます。

```
$ MMS/DESCRIPTION=BUILD_DISPATCH_SAMPLE1.MMS
```

BUILDファイルはプロセス内サーバとプロセス外サーバの両方を構築し、登録します。

### C.2.1.2 OpenVMS でのアプリケーションの登録

ビルド・プロシージャはCMPNT\$SHR.EXEとCMPNT.EXEの両方を自動的に登録します。コンポーネントを手動で登録するには、次の操作を行います。

- プロセス内サーバを登録するには、REGSVR32ユーティリティを使用します。次の例を参照してください。

```
$ regsvr32 ::= $DCOM$REGSVR32.EXE
$ regsvr32 path-nameCMPNT$SHR.EXE
```

- プロセス内サーバの登録を解除するには、REGSVR32ユーティリティを使用します。次の例を参照してください。

```
$ regsvr32 /u path-nameCMPNT$SHR.EXE
```

- プロセス外サーバを登録するには、次の操作を行います。

```
$ cmpnt ::= $path-nameCMPNT.EXE
$ cmpnt /regserver
```

- プロセス外サーバの登録を解除するには、次の操作を行います。

```
$ cmpnt /unregserver
```

### C.2.1.3 プロセス外サーバとしての OpenVMS でのアプリケーションの実行

コンポーネントがプロセス外サーバであるサンプルを実行するには、CMPNT.EXEを実行します。

コンポーネントからServer: Waitingメッセージが表示されたら、別のウィンドウまたは端末セッションでクライアントを実行します。

- ウィンドウ (または端末セッション) 1:

```
$ run cmpnt
```

- ウィンドウ (または端末セッション) 2:

- ディスパッチ・クライアントの場合:

```
$ run dclient
```

- COM クライアントの場合:

```
$ run client
```

クライアントは次のメッセージを表示します。

```
To which server do you want to connect?  
1) In-Process Server  
2) Out-of-Process Server  
:
```

2と入力して、プロセス外サーバを選択します。

#### C.2.1.4 OpenVMS でのアプリケーションの実行とリモート・サーバの指定

リモート・マシン (またはサーバ・システム) として指定したシステムでCMPNT.EXEを実行します。リモート・システムは Windows NT システムでもかまいません。Server: Waitingメッセージがコンポーネントから表示されたら、ローカル・マシン (またはクライアント・システム) として指定したシステムでクライアントを実行します。次の例を参照してください。

COM クライアントを使用するには、次のように入力します。

```
$ client ::= $path-nameCLIENT.EXE  
$ client remote-system-name  
To which server do you want to connect?  
1) In-Process Server  
2) Out-of-Process Server  
:
```

2と入力して、リモート・サーバ実行、つまりサーバ外プロセスを選択します。

#### C.2.1.5 プロセス内サーバとしての OpenVMS でのアプリケーションの実行

コンポーネントがプロセス内サーバであるサンプルを実行するには、クライアントだけを実行します。次の例を参照してください。

- ディスパッチ・クライアントの場合:

```
$ run dclient
```

- COM クライアントの場合:

```
$ run client
```

クライアントは次のメッセージを表示します。

```
To which server do you want to connect?  
1) In-Process Server  
2) Out-of-Process Server  
:
```

1と入力して、プロセス内サーバを選択します。

## C.2.2 Windows NT の場合の手順

ここでは、Windows NT システムでアプリケーションを構築する方法について説明します。

### C.2.2.1 Windows NT でのアプリケーションの構築

README-DISPATCH-SAMPLE1.TXTファイルと次のファイルを COM examples ディレクトリから Windows NT システムにコピーします。

```
CLIENT.CXX  
DCLIENT.CXX  
CMPNT.CXX  
CMPNT.DEF  
MAKE-ONE.  
MAKEFILE.BAT  
REGISTRY.CXX  
REGISTRY.H  
SERVER.IDL
```

MAKEFILE.BATファイルを使用してサンプルを構築します。次の例を参照してください。

```
C:> MAKEFILE
```

Makefile はプロセス内サーバとプロセス外サーバの両方を構築し、登録します。

### C.2.2.2 Windows NT でのアプリケーションの登録

ビルド・プロシージャmake-oneはCMPNT.DLL、PROXY.DLL、CMPNT.EXEを自動的に登録します。次の例を参照してください。

```
regsvr32 -s Cmpnt.dll  
Cmpnt /RegServer
```

アプリケーションの登録を解除するには、次のコマンドを入力します。

```
regsvr32 -u Cmpnt.dll  
Cmpnt /UnRegServer
```

### C.2.2.3 Windows NT でのアプリケーションの実行

DCLIENTまたはCLIENTを実行します。OpenVMS でプロセス内サーバとしてアプリケーションを実行する手順 (第 C.2.1.5 項), およびプロセス外サーバとして実行する手順 (第 C.2.1.3 項) と同じ手順を実行します。

リモート・マシンの名前 (UNC または DNS) を引数として使用して, リモート・マシンでオブジェクトのインスタンスを作成します。



---

## COM Version 1.0 for OpenVMS から COM Version 1.1 for OpenVMS へのアップグレード

---

### D.1 COM for OpenVMS の以前のバージョンからのアップグレード

ここでは、COM for OpenVMS の以前のバージョンからアップグレードするときに必要な手順について説明します。

#### D.1.1 既存の COM for OpenVMS アプリケーションの再構築

COM for OpenVMS アプリケーションに次のいずれかの API に対する参照が含まれる場合は、その参照を含むモジュールを再コンパイルし、アプリケーションを再リンクしなければなりません。

```
LoadLibraryA  
LoadLibraryW  
LoadLibraryExW  
LoadLibraryExA  
GetModuleFileNameA  
GetModuleFileNameW  
GetModuleHandleW  
GetProcAddress  
FreeLibrary
```

COM Version 1.0 for OpenVMS に付属している一部のサンプル COM アプリケーションでは、REGISTRY モジュールと CMPNT モジュールにこれらの API に対する参照が含まれています。サンプルを構築した場合や、これらのサンプルをもとに独自の COM アプリケーションを構築した場合は、そのアプリケーションを再コンパイルし、再リンクしなければなりません。

#### D.1.2 Windows NT システムの構成

COM Version 1.0 for OpenVMS (非認証 COM) の場合、COM for OpenVMS のマニュアルには、非認証 COM for OpenVMS が Windows NT と相互運用できるように、Windows NT レジストリの特定の値を変更するように説明されていました。COM Version 1.1 for OpenVMS では、認証がサポートされるようになりました。この結果、COM Version 1.0 for OpenVMS で変更が必要だった Windows NT Registry の値をデフォルトの認証設定に戻す必要があります。Windows NT Registry の値を設定するには、次の操作を行います。

COM Version 1.0 for OpenVMS から COM Version 1.1 for OpenVMS へのアップグレード  
D.1 COM for OpenVMS の以前のバージョンからのアップグレード

1. Windows NT Registry エディタを起動します。
2. 次のレジストリ・キーを選択します。

HKEY\_LOCAL\_MACHINE\Software\Microsoft\Ole

3. 次の値名と値データを削除します。

値名	COM Version 1.0 for OpenVMS での推奨設定	デフォルト (認証) 値データ (COM Version 1.1 for OpenVMS の設定)	レジストリの種類
ActivationSecurity	N	削除	REG_SZ
PersonalClasses	N	削除	REG_SZ

4. Default Authentication LevelとDefault Impersonation Levelを確認し、必要に応じて変更します。次の操作を行います。

**注意**

これらの設定を表示し、更新するには、Windows NT の Administrator 特権が必要です。

- a. 「スタート」メニューの「ファイル名を指定して実行...」を選択します。
  - b. 「ファイル名を指定して実行」ダイアログ・ボックスで、dcomcnfgと入力します。  
「分散 COM の構成のプロパティ」シートが表示されます。
  - c. 「既定のプロパティ」タブをクリックします。
    - 「既定の認証レベル」リスト・ボックスに「接続」が表示されるはずで  
す。表示されない場合は、リスト・ボックスの矢印をクリックし、一覧から「接続」を選択します。
    - 「既定の偽装レベル」リスト・ボックスに「識別」が表示されるはずで  
す。表示されない場合は、リスト・ボックスの矢印をクリックし、「識別」を一覧から選択します。
5. これらの変更を有効にするには、Windows NT システムを再起動しなければなりません。

### D.1.3 OpenVMS システムの構成

OpenVMS システムで、OpenVMS Registry の特定の値を設定またはリセットしなければなりません。Windows NT Registry エディタを使用して OpenVMS Registry を変更することができ、REGSCP ユーティリティを使用することもできます。OpenVMS Registry の値を設定するには、次の操作を行います。

1. OpenVMS Registry の次のキーを選択します。

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Ole
```

2. ActivationSecurity, PersonalClasses, LegacyAuthenticationLevel, LegacyImpersonationLevel キーを削除します。キーを削除するには、次のコマンドを使用します。

```
$ MCR REG$CP  
REG> LIST VALUE HKEY_LOCAL_MACHINE\Software\Microsoft\Ole  
REG> DELETE VALUE HKEY_LOCAL_MACHINE\Software\Microsoft\Ole  
ActivationSecurity  
REG> DELETE VALUE HKEY_LOCAL_MACHINE\Software\Microsoft\Ole PersonalClasses  
REG> DELETE VALUE HKEY_LOCAL_MACHINE\Software\Microsoft\Ole  
LegacyAuthenticationLevel  
REG> DELETE VALUE HKEY_LOCAL_MACHINE\Software\Microsoft\Ole  
LegacyImpersonationLevel  
REG> LIST VALUE HKEY_LOCAL_MACHINE\Software\Microsoft\Ole  
REG> EXIT
```

---

## D.2 Windows NT で以前に構成されたアプリケーション

Windows NT で COM Version 1.0 for OpenVMS (非認証 COM for OpenVMS) と組み合わせて実行するようにアプリケーションを構成した場合は、COM Version 1.1 for OpenVMS (認証 COM for OpenVMS) を利用できるように、Windows NT の設定を再構成しなければなりません。

COM Version 1.0 for OpenVMS では、COM for OpenVMS のマニュアルに、コンポーネントを登録した後、認証されていないユーザがイメージをアクティブにすることができるように、そのコンポーネントでセキュリティ・プロパティを確認するように説明されていました。次の操作を行います。

1. Windows NT の「スタート」メニューの「ファイル名を指定して実行...」を選択します。
2. 「ファイル名を指定して実行」ダイアログ・ボックスで、dcomcnfgと入力します。

「分散 COM の構成のプロパティ」シートが表示されます。

3. 「アプリケーション」リストからオブジェクトの名前を選択し、「プロパティ...」ボタンをクリックします。

選択したオブジェクトのプロパティ・シートが表示されます。

4. プロパティ・シートで「セキュリティ」タブをクリックします。

- COM Version 1.0 for OpenVMS では、ユーザEveryoneに対してアクセスが許可されるように (Allow access), アクセス権を設定しなければなりません (レジストリ値AccessPermission)。

COM Version 1.1 for OpenVMS では、特定のユーザに対して独自のアクセス権 (レジストリ値AccessPermission) を設定できます。

「独自のアクセス権を使う」をクリックし、「編集」ボタンをクリックして、「レジストリ値のアクセス権」ダイアログ・ボックスを表示します。

- COM Version 1.0 for OpenVMS では、ユーザ Everyone がアプリケーション・サーバを起動できるように、起動アクセス権 (レジストリ値 LaunchPermission) を Allow launch に設定しなければなりませんでした。

COM Version 1.1 for OpenVMS では、独自の起動アクセス権 (レジストリ値 LaunchPermission) を設定して、Everyone を削除できます。

「独自のアクセス権を使う」をクリックし、「編集」ボタンをクリックして、「レジストリ値のアクセス権」ダイアログ・ボックスを表示します。

- COM Version 1.0 for OpenVMS では、ユーザ Everyone に対して少なくともレジストリ値への Read アクセスを許可するように、構成許可を設定しなければなりませんでした。

COM Version 1.1 for OpenVMS では、独自の構成アクセス権を設定して、Everyone を削除できます。

「独自のアクセス権を使う」をクリックし、「編集」ボタンをクリックして、「レジストリ値のアクセス権」ボックスを表示します。

COM Version 1.0 for OpenVMS では、セキュリティ・プロパティを設定した後、アプリケーションを実行するためにアカウントの ID を設定しなければなりませんでした。

COM Version 1.1 for OpenVMS では、アカウントの ID はオプション 1 または 2 に設定できます。

「識別」タブをクリックして、ユーザ・アカウントの選択を表示します。「対話ユーザー」オプションを選択します。

#### D.2.1 COM Version 1.1 for OpenVMS では OpenVMS Registry に情報を再登録しなければならない

COM Version 1.1 for OpenVMS では、セキュリティ設定を含むように、OpenVMS Registry に情報を再登録しなければなりません。DCOM\$SETUP コマンド・プロシージャを使用して「OpenVMS COM Tools」メニューを表示し、オプション 3 を選択します。

#### D.2.2 OpenVMS Registry のアプリケーション・セキュリティ設定の変更

OpenVMS V7.2 に添付されていた COM Version 1.0 for OpenVMS では、NTLM セキュリティがサポートされませんでした。その結果、COM Version 1.0 for OpenVMS COM アプリケーションを登録するために使用する OpenVMS アカウントが、アプリケーション登録の一部として作成された OpenVMS Registry キーの所有者になっていました。たとえば、COM Version 1.0 for OpenVMS を使用した場合、

SYSTEM アカウントにログインして、SAMPLE1 アプリケーションを登録すると、SAMPLE1 の OpenVMS Registry キーは SYSTEM によって所有されます。

OpenVMS V7.2-1 に添付されている COM Version 1.1 for OpenVMS では、NTLM セキュリティがサポートされるようになりました。OpenVMS Registry キーへのアクセスを制御するために、ネットワーク・アカウントが使用されるようになりました。このような変更が行われた結果、以前のセキュリティ設定では、特権が与えられていないユーザがアプリケーションのレジストリ・キーにアクセスできなくなることがあります。つまり、既存のアプリケーションを使用している非特権ユーザがアプリケーションの登録や登録の解除を行うことができなくなる可能性があります。

この登録のロックアウトを防止するには、アプリケーションのアクセス権を変更しなければなりません。アクセス権の変更は、Windows NT システムまたは OpenVMS システムから行うことができます。次のいずれかの操作を行います。

- Windows NT システムからアクセス権を変更する方法
  1. Windows NT システムで RegEdt32 を起動します
  2. 「レジストリ」メニューで「コンピュータの選択」を選択し、OpenVMS Registry が格納されている OpenVMS システムに接続します。
  3. 変更するアプリケーションに関連付けられているキーを選択します。
  4. 「セキュリティ」メニューで「アクセス権...」を選択し、ユーザにフル・コントロール・アクセス権を与えます。
  5. 最後の 2 つの手順は、アプリケーションに関連付けられている各レジストリ・キーに対して繰り返します。COM アプリケーション関連のレジストリ・キーについては、第 D.2.2.1 項を参照してください。
- OpenVMS システムからアクセス権を変更する方法
  1. 特権が与えられた OpenVMS アカウントにログインします。
  2. アプリケーションの登録を解除します。DCOM\$REGSVR32 ユーティリティを使用します。例 5-6 を参照してください。
  3. アプリケーションに関連付けられたすべてのレジストリ・キーを削除します。COM アプリケーション関連のレジストリ・キーについては、第 D.2.2.1 項を参照してください。
  4. 特権が与えられていないユーザ・アカウントにログインします。
  5. アプリケーションを登録します。DCOM\$REGSVR32 ユーティリティを使用するか(例 5-5 を参照)、「OpenVMS COM Tools」メニューからオプション 6 を選択します(第 5.2 節)を参照。

#### D.2.2.1 COM アプリケーションのレジストリ・キー

COM アプリケーションには複数のレジストリ・キーを関連付けることができます。アプリケーションに関連付けられたすべてのキーを変更しなければなりません。アプリケーションは通常、次のキーを登録します。

HKEY\_CLASSES\_ROOT\CLSID\{*guid*} とサブキー  
HKEY\_CLASSES\_ROOT\APPID\{*guid*}  
HKEY\_CLASSES\_ROOT\APPID\filename  
HKEY\_CLASSES\_ROOT\TYPELIB\{*typelib guid*}  
HKEY\_CLASSES\_ROOT\INTERFACES\{*interface guid(s)*} とサブキー  
HKEY\_CLASSES\_ROOT\name とサブキー  
HKEY\_CLASSES\_ROOT\version independent name とサブキー

---

#### 注意

---

HKEY\_CLASSES\_ROOTはHKEY\_LOCAL\_MACHINE\SOFTWARE\Classesのエイリアスです。 Regedt32 を使用して Windows NT から OpenVMS Registry に接続していて、HKEY\_CLASSES\_ROOTキーを変更する場合は、HKEY\_LOCAL\_MACHINE\SOFTWARE\Classesキーを変更してください。

---

---

## 相違点，API，インタフェースの一覧

この付録には，COM for OpenVMS と Microsoft COM の実装上の相違点，API の一覧，COM for OpenVMS の本リリースで提供されるインタフェースの一覧を示します。

---

### E.1 COM for OpenVMS と Microsoft COM の相違点

ここでは，COM for OpenVMS と Microsoft の COM の実装上の重要な相違点を示します。

#### E.1.1 サービス制御マネージャ (SCM)

OpenVMS では，Windows NT のサービス制御マネージャに相当する機能を提供していません。この結果，Server サービス (stop, start, pause, resume など) に依存するアプリケーションは，同様の機能を提供する OpenVMS 機能に依存します (このような機能が提供される場合)。

たとえば，OpenVMS サイト固有のスタートアップ・コマンド・プロシージャとシャットダウン・コマンド・プロシージャを使用して，システムのスタートアップ時にサービスを自動起動し，システムのシャットダウン時にサービスの自動的なシャットダウンを実装できます。RegisterServiceCtrlHandler や ChangeServiceConfig などのサービス API は，OpenVMS では提供されません。

#### E.1.2 サーバ・アプリケーションのスタック・サイズ

COM for OpenVMS では，サーバ・アプリケーションの機能はサーバ・スレッドのコンテキストで実行されます。この結果，サーバ機能のスタック空間は 48 KB に制限されます。ローカル変数や構造を格納するために追加空間が必要な場合は，ローカル変数または構造用に動的メモリを割り当てなければなりません。

#### E.1.3 “char”データ型の使用

OpenVMS と Windows NT は IDL ベースのデータ型 “char” を異なる方法で変換します。

OpenVMS では、このデータ型を MIDL\_CHAR として変換します。これは CHAR として定義されており、さらに“char”としても定義されています。OpenVMS コンパイラはデフォルト設定で、これを“unsigned char”に相当するものと解釈します。ほとんどの場合、これらは同じように使用できます。しかし、この 2 つは正確には同じではありません。C++ では、クラス・メンバ定義でこの 2 つを指定した場合、異なるデータ型として取り扱われます。

Windows NT では、このデータ型は直接“unsigned char”として変換されます。これは Visual C++ と矛盾します。Visual C++ では、“char”データ型は“signed char”と同じであるものとして取り扱われます。OpenVMS の場合と同様に、“char”はクラス・メンバ定義では“signed char”と同じではありません。

この矛盾を回避するには、次の 2 つの方法があります。

- IDL ファイルとすべてのメンバ定義で、“char”の代わりにデータ型“CHAR”を使用します。これは最も移植可能な解決方法です。この方法は他のシステム (たとえば UNIX) でも機能します。
- メソッド定義を条件付きでコンパイルして、OpenVMS が“char”として定義されているオブジェクト・メソッドを認識し、Windows NT が“unsigned char”として定義されているメソッドを認識するようにします。

#### E.1.4 MIDL コンパイラのバージョン

COM for OpenVMS に添付されている MIDL コンパイラは、Microsoft の MIDL コンパイラ V3.00.44 を基礎にしています。

#### E.1.5 アプリケーション構成のアクセス権を変更するための DCOM\$CNFG の使用

アプリケーションに関連付けられている一部のキーの OpenVMS Registry キー・アクセス権を変更するには、「Application Security」サブメニューのオプション 5 とオプション 6 を使用します。オプション 5 とオプション 6 は次のキーのセキュリティ設定に影響します。

```
HKEY_CLASSES_ROOT\APPID\{guid}
HKEY_CLASSES_ROOT\CLSID\{guid} とサブキー
```

Windows NT システムでは、HKEY\_CLASSES\_ROOT\CLSID\{guid} の下のサブキーのセキュリティ設定は、既存のセキュリティ設定が HKEY\_CLASSES\_ROOT\APPID\{guid} の元の設定と一致する場合にだけ変更されます。

OpenVMS システムでは、サブキーの設定は、HKEY\_CLASSES\_ROOT\APPID\{guid} の元の設定と既存の設定が一致しない場合でも変更されます。

オプション 5 とオプション 6 は, アプリケーションに関連付けられているすべてのキーの設定を変更するわけではありません。たとえば, オプション 5 とオプション 6 は次のキーには影響しません。

HKEY\_CLASSES\_ROOT\APPID\*file name*

HKEY\_CLASSES\_ROOT\TYPELIB\{*typelib guid*}

HKEY\_CLASSES\_ROOT\INTERFACES\{*interface guid(s)*} とサブキー

HKEY\_CLASSES\_ROOT\*name* とサブキー

HKEY\_CLASSES\_ROOT\*version independent name* とサブキー

これらのキーのセキュリティ設定を変更するには, 次の操作を行います。

1. Windows NT システムから RegEdt32 を起動します。
2. 「レジストリ」メニューの「コンピュータの選択」を選択し, OpenVMS Registry が格納されている OpenVMS システムに接続します。
3. 変更するアプリケーションに関連付けられているキーを選択します。
4. 「セキュリティ」メニューの「アクセス権...」を選択し, ユーザにフル・コントロール・アクセス権を与えます。
5. アプリケーションに関連付けられている各レジストリ・キーに対して, 最後の 2 つの手順を繰り返します (この節で前に説明したキーの一覧を参照)。

---

注意

---

HKEY\_CLASSES\_ROOT は HKEY\_LOCAL\_MACHINE\SOFTWARE\Classes のエイリアスです。 Regedt32 を使用して Windows NT から OpenVMS Registry に接続し, HKEY\_CLASSES\_ROOT キーを変更する場合は, HKEY\_LOCAL\_MACHINE\SOFTWARE\Classes キーを変更してください。

---

## E.2 API

セキュリティ・サポートが必要な API は , COM Version 1.0 for OpenVMS ではサポートされません。

本リリースでサポートされる API は次のとおりです。

BindMoniker  
BstrFromVector  
CLSIDFromProgID  
CLSIDFromString  
CoAddRefServerProcess  
CoCopyProxy  
CoCreateFreeThreadedMarshaler  
CoCreateGuid  
CoCreateInstance  
CoCreateInstanceEx  
CoDisconnectObject  
CoDosDateTimeToFileTime  
CoFileTimeNow  
CoFileTimeToDosDateTime  
CoFreeAllLibraries  
CoFreeLibrary  
CoFreeUnusedLibraries  
CoGetCallContext  
CoGetClassObject  
CoGetCurrentProcess  
CoGetInstanceFromFile  
CoGetInstanceFromIStorage  
CoGetInterfaceAndReleaseStream  
CoGetMalloc  
CoGetMarshalSizeMax  
CoGetPSClsid  
CoGetStandardMarshal  
CoGetTreatAsClass  
CoImpersonateClient  
CoInitialize  
CoInitializeEx  
CoInitializeSecurity  
CoIsHandlerConnected  
CoLoadLibrary  
CoLockObjectExternal  
CoMarshalInterface  
CoQueryAuthenticationServices  
CoQueryClientBlanket  
CoQueryProxyBlanket  
CoRegisterClassObject  
CoRegisterMallocSpy  
CoRegisterMessageFilter  
CoRegisterPSClsid  
CoReleaseMarshalData  
CoReleaseServerProcess  
CoResumeClassObjects

CoRevertToSelf  
CoRevokeClassObject  
CoRevokeMallocSpy  
CoSetProxyBlanket  
CoSuspendClassObjects  
CoTaskMemAlloc  
CoTaskMemFree  
CoTaskMemRealloc  
CoTreatAsClass  
CoUninitialize  
CoUnmarshalInterface  
CreateAntiMoniker  
CreateBindCtx  
CreateClassMoniker  
CreateDataAdviseHolder  
CreateDispTypeInfo  
CreateErrorInfo  
CreateGenericComposite  
CreateILockBytesOnHGlobal  
CreateItemMoniker  
CreatePointerMoniker  
CreateStdDispatch  
CreateStreamOnHGlobal  
CreateTypeLib  
DispGetIDsOfNames  
DispGetParam  
DispInvoke  
DllCanUnloadNow  
DllGetClassObject  
DllGetClassObject  
DllMain  
DllRegisterServer  
DllUnregisterServer  
DosDateTimeToVariantTime  
FreePropVariantArray  
GetActiveObject  
GetAltMonthNames  
GetClassFile  
GetConvertStg  
GetErrorInfo  
GetHGlobalFromILockBytes  
GetHGlobalFromStream  
GetRunningObjectTable  
IIDFromString  
IsEqualCLSID  
IsEqualGUID  
IsEqualIID  
IsValidIid  
IsValidInterface  
IsValidPtrIn  
IsValidPtrOut  
LHashValOfName  
LHashValOfNameSys  
LoadRegTypeLib  
LoadTypeLibEx

相違点 , API , インタフェースの一覧  
E.2 API

MkParseDisplayName  
MonikerCommonPrefixWith  
MonikerRelativePathTo  
ProgIDFromCLSID  
PropStgNameToFmtId  
PropVariantClear  
PropVariantCopy  
QueryPathOfRegTypeLib  
ReadClassStg  
ReadClassStm  
ReadFmtUserTypeStg  
RegisterActiveObject  
RegisterTypeLib  
ReleaseStgMedium  
RevokeActiveObject  
SafeArrayAccessData  
SafeArrayAllocData  
SafeArrayAllocDescriptor  
SafeArrayCopy  
SafeArrayCopyData  
SafeArrayCreate  
SafeArrayCreateVector  
SafeArrayDestroy  
SafeArrayDestroyData  
SafeArrayDestroyDescriptor  
SafeArrayGetDim  
SafeArrayGetElement  
SafeArrayGetElemsize  
SafeArrayGetLBound  
SafeArrayGetUBound  
SafeArrayLock  
SafeArrayPtrOfIndex  
SafeArrayPutElement  
SafeArrayRedim  
SafeArrayUnaccessData  
SafeArrayUnlock  
SetConvertStg  
SetErrorInfo  
StgCreateDocfile  
StgCreateDocfileOnILockBytes  
StgCreatePropSetStg  
StgCreatePropStg  
StgIsStorageFile  
StgIsStorageILockBytes  
StgOpenPropStg  
StgOpenStorage  
StgOpenStorageOnILockBytes  
StgSetTimes  
StringFromCLSID  
StringFromGUID2  
StringFromIID  
SysAllocString  
SysAllocStringByteLen  
SysAllocStringLen

SysFreeString  
SysReAllocString  
SysReAllocStringLen  
SysStringByteLen  
SysStringLen  
SystemTimeToVariantTime  
UnRegisterTypeLib  
VarDateFromUdate  
VarNumFromParseNum  
VarParseNumFromStr  
VarUdateFromDate  
VariantChangeType  
VariantChangeTypeEx  
VariantClear  
VariantCopy  
VariantCopyInd  
VariantInit  
VariantTimeToDosDateTime  
VariantTimeToSystemTime  
VectorFromBstr  
WriteClassStg  
WriteClassStm  
WriteFmtUserTypeStg

---

### E.3 インタフェース

本リリースでサポートされるインタフェースは、次のとおりです。

IAdviseSink  
IBindCtx  
IClassActivator  
IClassFactory  
IConnectionPoint  
IConnectionPointContainer  
ICreateErrorInfo  
ICreateTypeInfo  
ICreateTypeLib  
IDataAdviseHolder  
IDataObject  
IDispatch  
IEnumCallBack  
IEnumConnectionPoints  
IEnumConnections  
IEnumFORMATETC  
IEnumMoniker  
IEnumOLEVerb  
IEnumSTATDATA  
IEnumSTATPROPSETSTG  
IEnumSTATSTG  
IEnumString  
IEnumUnknown  
IEnumVariant  
IErrorInfo

相違点, API, インタフェースの一覧  
E.3 インタフェース

IExternalConnection  
ILockBytes  
IMalloc  
IMallocSpy  
IMarshal  
IMessageFilter  
IMoniker  
IMultiQI  
IParseDisplayName  
IPersist  
IPersistFile  
IPersistStorage  
IPersistStream  
IPropertySetStorage  
IPropertyStorage  
IRootStorage  
IRunnableObject  
IRunningObjectTable  
IStdMarshalInfo  
IStorage  
IStream  
ISupportErrorInfo  
ITypeComp  
ITypeInfo  
ITypeInfo2  
ITypeLib  
ITypeLib2  
IUnknown

---

# COM for OpenVMS によりインストールされたファイル 一覧

---

## F.1 COM for OpenVMS によりインストールされたファイル

下記のファイルは、COM for OpenVMS インストール・プロセスの一部としてインストールされたファイルです。

```
[000000]DEC-AXPVMS-DCOM-V0101--1.PCSI$TLB
[DCOM$LIBRARY]CDERR.H
[DCOM$LIBRARY]CGUID.H
[DCOM$LIBRARY]COGUID.H
[DCOM$LIBRARY]COMMDLG.H
[DCOM$LIBRARY]CONIO.H
[DCOM$LIBRARY]DCOM$GUIDGEN.CLD
[DCOM$LIBRARY]DCOM$REGDATA.REG
[DCOM$LIBRARY]DCOM$RUNSHRLIB.CLD
[DCOM$LIBRARY]DCOM.OPT
[DCOM$LIBRARY]DDE.H
[DCOM$LIBRARY]DDEML.H
[DCOM$LIBRARY]DLGS.H
[DCOM$LIBRARY]EXCPT.H
[DCOM$LIBRARY]IMM.H
[DCOM$LIBRARY]INITGUID.H
[DCOM$LIBRARY]LZEXPAND.H
[DCOM$LIBRARY]MCX.H
[DCOM$LIBRARY]MIDLES.H
[DCOM$LIBRARY]MIDL_STUB_TYPES.H
[DCOM$LIBRARY]MMSYSTEM.H
[DCOM$LIBRARY]NB30.H
[DCOM$LIBRARY]NTA_MESSAGE.H
[DCOM$LIBRARY]OAIDL.H
[DCOM$LIBRARY]OAIDL.IDL
[DCOM$LIBRARY]OBJBASE.H
[DCOM$LIBRARY]OBJIDL.H
[DCOM$LIBRARY]OBJIDL.IDL
[DCOM$LIBRARY]OCIDL.H
[DCOM$LIBRARY]OCIDL.IDL
[DCOM$LIBRARY]OLE2.H
[DCOM$LIBRARY]OLEAUTO.H
[DCOM$LIBRARY]OLECTL.H
[DCOM$LIBRARY]OLEIDL.H
[DCOM$LIBRARY]OLEIDL.IDL
[DCOM$LIBRARY]POPPACK.H
[DCOM$LIBRARY]PRSHOT.H
[DCOM$LIBRARY]PSHPACK1.H
```

COM for OpenVMS によりインストールされたファイル一覧  
F.1 COM for OpenVMS によりインストールされたファイル

[DCOM\$LIBRARY]PSHPACK2.H  
[DCOM\$LIBRARY]PSHPACK4.H  
[DCOM\$LIBRARY]PSHPACK8.H  
[DCOM\$LIBRARY]PTHREAD.H  
[DCOM\$LIBRARY]PTHREAD\_EXCEPTION.H  
[DCOM\$LIBRARY]RPC.H  
[DCOM\$LIBRARY]RPCDCE.H  
[DCOM\$LIBRARY]RPCDCEP.H  
[DCOM\$LIBRARY]RPCNDR.H  
[DCOM\$LIBRARY]RPCNSI.H  
[DCOM\$LIBRARY]RPCNSIP.H  
[DCOM\$LIBRARY]RPCNTERR.H  
[DCOM\$LIBRARY]RPCPROXY.H  
[DCOM\$LIBRARY]SERVPROV.H  
[DCOM\$LIBRARY]SERVPROV.IDL  
[DCOM\$LIBRARY]SHELLAPI.H  
[DCOM\$LIBRARY]STDOLE2.TLB  
[DCOM\$LIBRARY]STDOLE32.TLB  
[DCOM\$LIBRARY]TCHAR.H  
[DCOM\$LIBRARY]UNKNWN.H  
[DCOM\$LIBRARY]UNKNWN.IDL  
[DCOM\$LIBRARY]URLMON.H  
[DCOM\$LIBRARY]URLMON.IDL  
[DCOM\$LIBRARY]UUID.OLB  
[DCOM\$LIBRARY]VMS\_DCOM.H  
[DCOM\$LIBRARY]VMS\_IOCTL.H  
[DCOM\$LIBRARY]WCHAR.H  
[DCOM\$LIBRARY]WINBASE.H  
[DCOM\$LIBRARY]WINCON.H  
[DCOM\$LIBRARY]WINDEF.H  
[DCOM\$LIBRARY]WINDOWS.H  
[DCOM\$LIBRARY]WINDU\_PLATFORM.H  
[DCOM\$LIBRARY]WINDU\_STDLIB.H  
[DCOM\$LIBRARY]WINDU\_STRING.H  
[DCOM\$LIBRARY]WINDU\_VTBL.H  
[DCOM\$LIBRARY]WINERROR.H  
[DCOM\$LIBRARY]WINGDI.H  
[DCOM\$LIBRARY]WINNETWK.H  
[DCOM\$LIBRARY]WINNLS.H  
[DCOM\$LIBRARY]WINNT.H  
[DCOM\$LIBRARY]WINPERF.H  
[DCOM\$LIBRARY]WINREG.H  
[DCOM\$LIBRARY]WINSOCK.H  
[DCOM\$LIBRARY]WINSPOOL.H  
[DCOM\$LIBRARY]WINSVC.H  
[DCOM\$LIBRARY]WINUSER.H  
[DCOM\$LIBRARY]WINVER.H  
[DCOM\$LIBRARY]WYPES.H  
[DCOM\$LIBRARY]WYPES.IDL  
[DCOM\$LIBRARY]WUEXTEN.H  
[DCOM\$LIBRARY]WUUNALIGNED.H  
[DCOM\$LIBRARY]WUVERSION.H  
[DCOM\$WIN32.NLS]BIG5.NLS  
[DCOM\$WIN32.NLS]CTYPE.NLS

COM for OpenVMS によりインストールされたファイル一覧  
F.1 COM for OpenVMS によりインストールされたファイル

[DCOM\$WIN32.NLS]C\_037.NLS  
[DCOM\$WIN32.NLS]C\_10000.NLS  
[DCOM\$WIN32.NLS]C\_10001.NLS  
[DCOM\$WIN32.NLS]C\_10002.NLS  
[DCOM\$WIN32.NLS]C\_10003.NLS  
[DCOM\$WIN32.NLS]C\_10004.NLS  
[DCOM\$WIN32.NLS]C\_10005.NLS  
[DCOM\$WIN32.NLS]C\_10006.NLS  
[DCOM\$WIN32.NLS]C\_10007.NLS  
[DCOM\$WIN32.NLS]C\_10008.NLS  
[DCOM\$WIN32.NLS]C\_10010.NLS  
[DCOM\$WIN32.NLS]C\_10017.NLS  
[DCOM\$WIN32.NLS]C\_10029.NLS  
[DCOM\$WIN32.NLS]C\_10079.NLS  
[DCOM\$WIN32.NLS]C\_10081.NLS  
[DCOM\$WIN32.NLS]C\_10082.NLS  
[DCOM\$WIN32.NLS]C\_1026.NLS  
[DCOM\$WIN32.NLS]C\_1250.NLS  
[DCOM\$WIN32.NLS]C\_1251.NLS  
[DCOM\$WIN32.NLS]C\_1252.NLS  
[DCOM\$WIN32.NLS]C\_1253.NLS  
[DCOM\$WIN32.NLS]C\_1254.NLS  
[DCOM\$WIN32.NLS]C\_1255.NLS  
[DCOM\$WIN32.NLS]C\_1256.NLS  
[DCOM\$WIN32.NLS]C\_1257.NLS  
[DCOM\$WIN32.NLS]C\_1258.NLS  
[DCOM\$WIN32.NLS]C\_1361.NLS  
[DCOM\$WIN32.NLS]C\_20105.NLS  
[DCOM\$WIN32.NLS]C\_20261.NLS  
[DCOM\$WIN32.NLS]C\_20269.NLS  
[DCOM\$WIN32.NLS]C\_20273.NLS  
[DCOM\$WIN32.NLS]C\_20277.NLS  
[DCOM\$WIN32.NLS]C\_20278.NLS  
[DCOM\$WIN32.NLS]C\_20280.NLS  
[DCOM\$WIN32.NLS]C\_20284.NLS  
[DCOM\$WIN32.NLS]C\_20285.NLS  
[DCOM\$WIN32.NLS]C\_20290.NLS  
[DCOM\$WIN32.NLS]C\_20297.NLS  
[DCOM\$WIN32.NLS]C\_20420.NLS  
[DCOM\$WIN32.NLS]C\_20423.NLS  
[DCOM\$WIN32.NLS]C\_20833.NLS  
[DCOM\$WIN32.NLS]C\_20838.NLS  
[DCOM\$WIN32.NLS]C\_20866.NLS  
[DCOM\$WIN32.NLS]C\_20871.NLS  
[DCOM\$WIN32.NLS]C\_20880.NLS  
[DCOM\$WIN32.NLS]C\_20905.NLS  
[DCOM\$WIN32.NLS]C\_21025.NLS  
[DCOM\$WIN32.NLS]C\_21027.NLS  
[DCOM\$WIN32.NLS]C\_28592.NLS  
[DCOM\$WIN32.NLS]C\_28593.NLS  
[DCOM\$WIN32.NLS]C\_28594.NLS  
[DCOM\$WIN32.NLS]C\_28595.NLS  
[DCOM\$WIN32.NLS]C\_28596.NLS  
[DCOM\$WIN32.NLS]C\_28597.NLS

COM for OpenVMS によりインストールされたファイル一覧  
F.1 COM for OpenVMS によりインストールされたファイル

[DCOM\$WIN32.NLS]C\_28598.NLS  
[DCOM\$WIN32.NLS]C\_28599.NLS  
[DCOM\$WIN32.NLS]C\_29001.NLS  
[DCOM\$WIN32.NLS]C\_437.NLS  
[DCOM\$WIN32.NLS]C\_500.NLS  
[DCOM\$WIN32.NLS]C\_708.NLS  
[DCOM\$WIN32.NLS]C\_720.NLS  
[DCOM\$WIN32.NLS]C\_737.NLS  
[DCOM\$WIN32.NLS]C\_775.NLS  
[DCOM\$WIN32.NLS]C\_850.NLS  
[DCOM\$WIN32.NLS]C\_852.NLS  
[DCOM\$WIN32.NLS]C\_855.NLS  
[DCOM\$WIN32.NLS]C\_857.NLS  
[DCOM\$WIN32.NLS]C\_860.NLS  
[DCOM\$WIN32.NLS]C\_861.NLS  
[DCOM\$WIN32.NLS]C\_862.NLS  
[DCOM\$WIN32.NLS]C\_863.NLS  
[DCOM\$WIN32.NLS]C\_864.NLS  
[DCOM\$WIN32.NLS]C\_865.NLS  
[DCOM\$WIN32.NLS]C\_866.NLS  
[DCOM\$WIN32.NLS]C\_869.NLS  
[DCOM\$WIN32.NLS]C\_870.NLS  
[DCOM\$WIN32.NLS]C\_874.NLS  
[DCOM\$WIN32.NLS]C\_875.NLS  
[DCOM\$WIN32.NLS]C\_932.NLS  
[DCOM\$WIN32.NLS]C\_936.NLS  
[DCOM\$WIN32.NLS]C\_949.NLS  
[DCOM\$WIN32.NLS]C\_950.NLS  
[DCOM\$WIN32.NLS]KSC.NLS  
[DCOM\$WIN32.NLS]LOCALE.NLS  
[DCOM\$WIN32.NLS]L\_EXCEPT.NLS  
[DCOM\$WIN32.NLS]L\_INTL.NLS  
[DCOM\$WIN32.NLS]PRC.NLS  
[DCOM\$WIN32.NLS]PRCP.NLS  
[DCOM\$WIN32.NLS]SORTKEY.NLS  
[DCOM\$WIN32.NLS]SORTTBLS.NLS  
[DCOM\$WIN32.NLS]UNICODE.NLS  
[DCOM\$WIN32.NLS]XJIS.NLS  
[DCOM\$WIN32]WINDU\$GDISHR.EXE  
[DCOM\$WIN32]WINDU\$KERNELSHR.EXE  
[DCOM\$WIN32]WINDU\$PRNTSHR.EXE  
[DCOM\$WIN32]WINDU\$USERSHR.EXE  
[DCOM\$WIN32]WINDU.INI  
[DCOM\$WIN32]WINDU.OPT  
[SYS\$STARTUP]DCOM\$RPCSS.COM  
[SYS\$STARTUP]DCOM\$SHUTDOWN.COM  
[SYS\$STARTUP]DCOM\$STARTUP.COM  
[SYSEXE]DCOM\$CNFG.EXE  
[SYSEXE]DCOM\$COMREGEDT.EXE  
[SYSEXE]DCOM\$GUIDGEN.EXE  
[SYSEXE]DCOM\$MIDL.EXE  
[SYSEXE]DCOM\$REGSVR32.EXE  
[SYSEXE]DCOM\$RPCSS.EXE  
[SYSEXE]DCOM\$RUNSHRLIB.EXE

COM for OpenVMS によりインストールされたファイル一覧  
F.1 COM for OpenVMS によりインストールされたファイル

```
[SYSEXE]DCOM$SCLIENT.EXE
[SYSEXE]DCOM$SSERVER.EXE
[SYSEXE]DCOM$SSERVER_REG.COM
[SYSEXE]DCOM$TOOL.EXE
[SYSHLP.EXAMPLES.DCOM.DISPATCH_SAMPLE1]BUILD_DISPATCH_SAMPLE1.COM
[SYSHLP.EXAMPLES.DCOM.DISPATCH_SAMPLE1]BUILD_DISPATCH_SAMPLE1.MMS
[SYSHLP.EXAMPLES.DCOM.DISPATCH_SAMPLE1]CLIENT.CXX
[SYSHLP.EXAMPLES.DCOM.DISPATCH_SAMPLE1]CMPNT$SHR.OPT
[SYSHLP.EXAMPLES.DCOM.DISPATCH_SAMPLE1]CMPNT.CXX
[SYSHLP.EXAMPLES.DCOM.DISPATCH_SAMPLE1]CMPNT.DEF
[SYSHLP.EXAMPLES.DCOM.DISPATCH_SAMPLE1]DCLIENT.CXX
[SYSHLP.EXAMPLES.DCOM.DISPATCH_SAMPLE1]MAKE-ONE.
[SYSHLP.EXAMPLES.DCOM.DISPATCH_SAMPLE1]MAKEFILE.BAT
[SYSHLP.EXAMPLES.DCOM.DISPATCH_SAMPLE1]README-DISPATCH-SAMPLE1.TXT
[SYSHLP.EXAMPLES.DCOM.DISPATCH_SAMPLE1]REGISTRY.CXX
[SYSHLP.EXAMPLES.DCOM.DISPATCH_SAMPLE1]REGISTRY.H
[SYSHLP.EXAMPLES.DCOM.DISPATCH_SAMPLE1]SERVER.IDL
[SYSHLP.EXAMPLES.DCOM.EVENTS]BUILD_EVENTS_SAMPLE.COM
[SYSHLP.EXAMPLES.DCOM.EVENTS]EVENTS_SAMPLE.C
[SYSHLP.EXAMPLES.DCOM.EVENTS]EVENTS_SAMPLE.H
[SYSHLP.EXAMPLES.DCOM.EVENTS]NTA_WIN32.C
[SYSHLP.EXAMPLES.DCOM.SAMPLE1]BUILD_SAMPLE1.COM
[SYSHLP.EXAMPLES.DCOM.SAMPLE1]BUILD_SAMPLE1.MMS
[SYSHLP.EXAMPLES.DCOM.SAMPLE1]CLIENT.CXX
[SYSHLP.EXAMPLES.DCOM.SAMPLE1]CMPNT$SHR.OPT
[SYSHLP.EXAMPLES.DCOM.SAMPLE1]CMPNT.CXX
[SYSHLP.EXAMPLES.DCOM.SAMPLE1]CMPNT.DEF
[SYSHLP.EXAMPLES.DCOM.SAMPLE1]GUIDS.CXX
[SYSHLP.EXAMPLES.DCOM.SAMPLE1]MAKE-ONE.
[SYSHLP.EXAMPLES.DCOM.SAMPLE1]MAKEFILE.BAT
[SYSHLP.EXAMPLES.DCOM.SAMPLE1]PROXY$SHR.OPT
[SYSHLP.EXAMPLES.DCOM.SAMPLE1]PROXY.DEF
[SYSHLP.EXAMPLES.DCOM.SAMPLE1]README-SAMPLE1.TXT
[SYSHLP.EXAMPLES.DCOM.SAMPLE1]REGISTRY.CXX
[SYSHLP.EXAMPLES.DCOM.SAMPLE1]REGISTRY.H
[SYSHLP.EXAMPLES.DCOM.SAMPLE1]SERVER.IDL
[SYSHLP.EXAMPLES.DCOM.SIMPLE]BUILD_SIMPLE.COM
[SYSHLP.EXAMPLES.DCOM.SIMPLE]INSTALL.BAT
[SYSHLP.EXAMPLES.DCOM.SIMPLE]MAKEFILE.
[SYSHLP.EXAMPLES.DCOM.SIMPLE]README-SIMPLE.TXT
[SYSHLP.EXAMPLES.DCOM.SIMPLE]REGISTER_SIMPLE.COM
[SYSHLP.EXAMPLES.DCOM.SIMPLE]SCLIENT.CPP
[SYSHLP.EXAMPLES.DCOM.SIMPLE]SSERVER.CPP
[SYSHLP.EXAMPLES.DCOM.SIMPLE]SSERVER.REG
[SYSHLP.EXAMPLES.DCOM.WRAPPER]BUILD_WRAPPER.COM
[SYSHLP.EXAMPLES.DCOM.WRAPPER]MAKE-ONE.
[SYSHLP.EXAMPLES.DCOM.WRAPPER]MAKEFILE.BAT
[SYSHLP.EXAMPLES.DCOM.WRAPPER]README.TXT
[SYSHLP.EXAMPLES.DCOM.WRAPPER]REGISTRY.CXX
[SYSHLP.EXAMPLES.DCOM.WRAPPER]REGISTRY.H
[SYSHLP.EXAMPLES.DCOM.WRAPPER]TEST.COM
[SYSHLP.EXAMPLES.DCOM.WRAPPER]WR$SHR.OPT
[SYSHLP.EXAMPLES.DCOM.WRAPPER]WRAPPER.CXX
[SYSHLP.EXAMPLES.DCOM.WRAPPER]WRAPPER.DEF
```

COM for OpenVMS によりインストールされたファイル一覧  
F.1 COM for OpenVMS によりインストールされたファイル

```
[SYSHLP.EXAMPLES.DCOM.WRAPPER]WRAPPER.IDL
[SYSHLP.EXAMPLES.DCOM.WRAPPER]WRAPPERCLIENT.CXX
[SYSHLP]OVMS_CONNECT_DEV_GDE_0799.PDF
[SYSHLP]OVMS_CONNECT_DEV_GDE_0799.PS
[SYSHLP]OVMS_CONNECT_DEV_GDE_0799.HTML
[SYSHLP]OVMS_CONNECT_DEV_GDE_0799_001.HTML
[SYSHLP]OVMS_CONNECT_DEV_GDE_0799_002.HTML
[SYSHLP]OVMS_CONNECT_DEV_GDE_0799_003.HTML
[SYSHLP]OVMS_CONNECT_DEV_GDE_0799_004.HTML
[SYSHLP]OVMS_CONNECT_DEV_GDE_0799_005.HTML
[SYSHLP]OVMS_CONNECT_DEV_GDE_0799_006.HTML
[SYSHLP]OVMS_CONNECT_DEV_GDE_0799_007.HTML
[SYSHLP]OVMS_CONNECT_DEV_GDE_0799_008.HTML
[SYSHLP]OVMS_CONNECT_DEV_GDE_0799_CONTENTS.HTML
[SYSHLP]OVMS_CONNECT_DEV_GDE_0799_CONTENTS_001.HTML
[SYSHLP]OVMS_CONNECT_DEV_GDE_0799_INDEX.HTML
[SYSHLP]VM-0126A.GIF
[SYSHLP]VM-0224A.GIF
[SYSHLP]VM-0225A.GIF
[SYSHLP]VM-0226A.GIF
[SYSHLP]VM-0227A.GIF
[SYSHLP]VM-0228A.GIF
[SYSHLP]VM-0283A.GIF
[SYSHLP]VM-0331A.GIF
[SYSHLP]VM-8782A.GIF
[SYSHLP]ZK-8782A.GIF
[SYSLIB]DCOM$MIDL_SHR.EXE
[SYSLIB]DCOM$NT_WRAPPERS_SHR.EXE
[SYSLIB]DCOM$OLE32_SHR.EXE
[SYSLIB]DCOM$OLEAUT32_SHR.EXE
[SYSLIB]DCOM$RPCRT4_SHR.EXE
[SYSLIB]DCOM$WIN32_SHR.EXE
[SYSMGR]DCOM$CREATE_ACCOUNT.COM
[SYSMGR]DCOM$REGISTRY_KEYS.COM
[SYSMGR]DCOM$SETUP.COM
[SYSMSG]DCOM$GUIDGEN_MSG.EXE
[SYSMSG]NTARPCMSG.EXE
[SYSMSG]NTAWINMSG.EXE
[000000]DEC-AXPVMS-DCOM-V0101--1.PCSI$DESCRIPTION
```

### 値 (レジストリ値)

レジストリの要素であり、レジストリ・キーまたはサブキーのエントリまたは値です。

### カプセル化

既存のアプリケーションのコードの大部分とその機能は変更せずに、キー・エントリ・ポイントに新しいコードや更新したコード (通常は別のプログラミング言語を使用) を含むことにより、既存のアプリケーションを更新したり、既存のアプリケーションを今後も使用できるようにするための処理。

たとえば、Windows クライアントから情報を収集し、データがキャラクタ・セル・インタフェースから入力されたかのように、データの書式を変換し、そのデータを既存のキャラクタ・セル・アプリケーションに提供することで、Windows グラフィカル・インタフェースをキャラクタ・セル・アプリケーションに追加できます。

### キー (レジストリ・キー)

レジストリの要素であり、コンピュータ、システム、ユーザ固有の情報が格納されます。

### クラス (レジストリ・クラス)

レジストリ要素の属性であり、レジストリ・キーまたはサブキーと共に追加説明情報を格納できます。

### サブキー (レジストリ・サブキー)

レジストリの要素であり、レジストリ・キーの子です。レジストリ・キーは 0 個以上のサブキーを持つことができます。

### ハイブ

レジストリに格納されているキー、サブキー、値エントリの独立した集合。

### プロセス外サーバ

要求を出しているクライアントと異なるシステムに存在するアプリケーション。Windows NT システムでは、プロセス外サーバは通常、.EXE ファイルとして実装されます。

### プロセス内サーバ

要求を出しているクライアントと同じシステムに存在するアプリケーション。Windows NT システムでは、プロセス内サーバは通常、DLL として実装されます。OpenVMS システムでは、プロセス内サーバは通常、共用可能イメージとして実装されます。

ラッパー

カプセル化を参照。

レジストリ

システム・ハードウェアとソフトウェアに関する構成情報を格納した1つ以上のファイルで構成される階層型データベース。

---

ACM	Authentication and Credential Management Authority
ACME	Authentication and Credential Management Extension
API	Application Program Interface
COM	Component Object Model
CLSID	Class ID
DCOM	Distributed Component Object Model
DLL	Dynamic Link Library
FMS	Forms Management System
GUI	Graphical User Interface
GUID	Globally Unique Identifier
MIDL	Microsoft Interface Definition Language
OO	Object oriented
RPC	Remote Procedure Call
SAM	Security Account Manager
SID	Security Identifier

## 略語集

SMG

Screen Management Facility

SSPI

Security Support Provider Interface

UI

User Interface

UIC

User Identification Code

## A

ACM (Authentication and Credential Management)	
オーソリティ	12-6
ACM オーソリティ	4-11
Advanced Server for OpenVMS イベント・ビューア	11-2
Advanced Server for OpenVMS からの COM for OpenVMS イベントの表示	11-2

## C

COM	
Microsoft Web サイト	3-5
定義	3-1
COM for OpenVMS	
C++ 修飾子	6-6
CLSID の登録	6-11
COM for OpenVMS サーバ・アプリケーションの登録	5-3
COM for OpenVMS サーバの起動	5-3
COM for OpenVMS サーバの停止	5-3
COM アプリケーションの構築	6-2
COM アプリケーションのコンパイル	6-5
COM アプリケーションのリンク	6-7
COM の情報を OpenVMS Registry データベースに登録する	5-2
C 修飾子	6-7
DCOM\$CNFG	5-1
DCOM\$REGSVR32	5-1
DCOM\$RUNSHRLIB	6-3
DCOM\$SETUP	5-1
GUIDGEN, Globally Unique Identifier Generator	5-2
GUID 形式オプション	6-2
HKEY_CLASSES_ROOT\CLSID サブキー	6-10, 6-11
HKEY_CLASSES_ROOT\Interface サブキー	6-12
InProcServer32 サブキー	6-11, 6-12
LocalServer32 サブキー	6-11
MIDL コンパイラ	6-2
NumMethods サブキー	6-12
OpenVMS Registry エントリ	6-10
OpenVMS Registry の使用	3-5
ProgID サブキー	6-11
ProxyStubClsid32 サブキー	6-12
Typelib サブキー	6-11

## COM for OpenVMS (続き)

VersionIndependentProgID サブキー	6-11
VMS_DCOM	6-6
アプリケーション開発の例	6-1
アプリケーションの作成	6-1
一意識別子 (GUID) の生成	6-1
インストールされたファイル	F-1
既存のアプリケーションのカプセル化	3-6
構成のためのユーティリティ	5-1
コンポーネント CLSID	6-11
サポートされる COM API	E-4
サポートされる COM インタフェース	E-7
使用	3-5
新規アプリケーションの開発	3-6
セキュリティの実装方法の相違点の要約	1-4
タイプ・ライブラリ	6-11
定義	3-2
プロキシ/スタブ CLSID	6-11
ヘッダ・ファイル	6-6
マクロ定義	6-5
COM for OpenVMS Developer's Kit	3-4
COM for OpenVMS サーバの起動	5-7
COM for OpenVMS サーバの停止	5-7
COM for OpenVMS 実行時	3-4
COM for OpenVMS によりインストールされたファイル一覧	F-1
COM for OpenVMS の SYLOGICALS ファイルの変更	4-21
COM for OpenVMS の構成のためのユーティリティ	5-1
COM for OpenVMS のシャットダウン	4-21
NOCONFIRM パラメータ	4-23
COM for OpenVMS の使用	3-5
COM イベントの作成	11-11

## D

DCOM\$CNFG	
Add Registry Key Permissions サブメニュー	5-21
Add Registry Value Permissions サブメニュー	5-16
Application Identity サブメニュー	5-22
Application List サブメニュー	5-11
Application Location サブメニュー	5-13
Application Properties サブメニュー	5-12
Application Security サブメニュー	5-14
Default Authentication Level サブメニュー	5-23

DCOMSCNFG (続き)	
Default Impersonation Level サブメニュー	
—	5-24
Edit Registry Key Permissions サブメニュー	
—	5-18
Edit Registry Value Permissions サブメニュー	
—	5-16
Registry Key Permissions サブメニュー	
—	5-18
Registry Value Permissions サブメニュー	
—	5-15
Special Access Registry Key Permissions サブメニュー	
—	5-19
System-wide Default Properties サブメニュー	
—	5-23
System-wide Default Security サブメニュー	
—	5-25
実行	5-10
定義	5-10
メニュー	5-10
DCOMSCNFG オプション	
Run application on another computer	5-13
システム単位のデフォルト・プロパティの表示	5-11
DCOMSCNFG のオプション	
Default authentication level	5-23
Default impersonation level	5-24
Enable Distributed COM on this computer	5-23
Launching user	5-22
Location: Machine to run application	5-12
NTLM アカウント	5-22
OpenVMS DCOM Guest Account	5-23
OpenVMS ユーザ名	5-23
Run application on this computer	5-13
Security permissions for application	5-13
User account to use to run application	5-17
システム単位のデフォルト・セキュリティの表示	5-11
マシン上のすべての COM アプリケーションの一覧表示	5-10
DCOM\$REGSVR32	
activation	5-28
コマンド・ライン・オプション	5-28
定義	5-27
場所	5-27
例	5-28
DCOM\$REGSVR32 ユーティリティ	5-27
DCOM\$RPCSS プロセス	5-7
DCOM\$RPCSS プロセスの起動	5-7
DCOM\$RPCSS プロセスの停止	5-7
DCOM\$SETUP	
オプション	5-2
実行	5-2
定義	5-1
表記法	5-1
メニュー	5-2
要件	5-1

DCOM\$TO_BE_STARTED 論理名	4-21
-------------------------	------

## E

EXTAUTH フラグ	4-12
-------------	------

## H

HKEY_CLASSES_ROOT	
定義	7-5
HKEY_LOCAL_MACHINE	
定義	7-5
HKEY_USERS	
定義	7-5

## I

IGNORE_EXTAUTH フラグ	4-12
--------------------	------

## L

LOGINOUT.EXE	4-12
--------------	------

## M

Microsoft MIDL コンパイラ	3-8
MIDL コンパイラ	6-3
DCOM\$RUNSHRLIB	6-3
イメージ	6-3
インクルード・ディレクトリ	6-5
実行	6-3
スイッチ	6-4
定義	6-3
ヘッダ・ファイル	6-5

## N

NTASLOGON	3-4, 4-11, 12-1
NTLM セキュリティ資格情報	4-11
NTLM 認証	4-12

## O

OpenVMS Events	
表示	11-2
ロギング	11-2
OpenVMS Events の問題への対処	11-11
OpenVMS MIDL コンパイラ	3-8
OpenVMS Registry	
COM for OpenVMS での使用	3-5
OpenVMS Alpha 複合バージョン・クラスタ内での実行	8-13
Unicode のサポート	8-13
Windows NT システムへの接続	8-11
アクセス・ライトの付与	7-10
インストール	8-1
起動	8-6
クォータ	8-12
クラスタ内でのフェールオーバー	8-10

OpenVMS Registry (続き)	
構成のためのユーティリティ	8-1
サーバ操作の制御	7-11
シャットダウン	8-7
セキュリティ	8-12
セキュリティ・モデル	7-8
定義	7-1
バックアップ	8-12
復元	8-12
読み込みと書き込み	7-7
OpenVMS Registry Configuration ユーティリティ	
オプション	8-2
メニュー	8-2
OpenVMS Registry サーバ操作	
Age Checker Interval	7-12
Database Log Cleaner Interval	7-12
Default File Quota	7-13
File Quota Interval	7-13
Initial Log File Size	7-12
Log Registry Value Error	7-13
Maximum Reply Age	7-12
Operator Communications Interval	7-14
Process Time Limit	7-14
Reply Log Cleaner Interval	7-14
Scan Interval	7-13
Snapshot Interval	7-15
Snapshot Location	7-15
Snapshot Versions	7-15
Write Retry Interval	7-15
OpenVMS Registry サーバ操作の制御	7-11
OpenVMS Registry サーバのクラスタ・フェールオ	
ーバ	8-10
OpenVMS Registry サーバのコマンド	8-7
OpenVMS Registry データベースへのアクセ	
ス	7-8
OpenVMS Registry の概念と定義	7-2
OpenVMS Registry の起動	8-6
手動	8-6
OpenVMS Registry のバックアップ	8-12
OpenVMS Registry の復元	8-12
OpenVMS Registry へのアクセス・ライト	7-10
OpenVMS Registry を構成するためのユーティリ	
ティ	8-1
OpenVMS イベント・ログ・ファイル	11-2
OpenVMS イベント・ログ・ファイルの COM for	
OpenVMS イベントの表示	11-3
OpenVMS および Windows エラー・コードの変	
換	6-12
OpenVMS セキュリティ・モデル	7-9
OpenVMS と Windows NT の相違点	E-1
OpenVMS と Windows NT の相違点:	
char データ型	E-1
MIDL コンパイラのバージョン	E-2
アプリケーション構成のアクセス権の変	
更	E-2
サーバ・アプリケーションのスタック・サイ	
ズ	E-1
サービス制御マネージャ	E-1

OpenVMS のインフラストラクチャ	3-3
---------------------	-----

## R

REGSCP サーバ管理ユーティリティ	7-8
SREGISTRYW システム・サービス	7-8
SREGISTRY システム・サービス	7-8
Release note: COM for OpenVMS	
RPC PROTOCOL エラー (800706C0)	1-9

## S

SECURITY_POLICY sysgen パラメータ	4-12
SET SERVER REGISTRY_SERVER	8-9
SHOW SERVER REGISTRY_SERVER	8-8
"Simple"アプリケーションの例	
NT での登録	5-8
OpenVMS での登録	5-8
Simple アプリケーションの例	
OpenVMS での再登録	5-9
作成	5-7
登録	5-7

## U

Unicode	8-13
---------	------

## W

Windows NT Registry	
定義	7-1
Windows NT イベント・ビューア	11-2
Windows NT からの COM for OpenVMS イベントの	
表示	11-2
Windows NT 資格情報	
チェック	7-9
付与	7-9
Windows NT 資格情報のチェック	7-9
Windows NT システムへの接続	8-11
Windows NT のセキュリティ・モデル	7-11

## A

値	7-2
値エントリ	7-2
アップグレードに関する注意: COM for OpenVMS	
アプリケーション・セキュリティ設定の変	
更	D-4
既存のアプリケーションの再構築	D-1
相互運用のための OpenVMS と Windows NT の	
構成	D-1
アプリケーションの登録	
例	5-7

## イ

イベント	11-1
イベント・ビューア	11-1
イベント・ログ・サービス	11-1
イベント・ログへの独自の COM イベントの書き込み	11-11
インフラストラクチャ	3-3

## カ

外部認証	4-12
カプセル化	3-6

## キ

キー	7-2
キーと値の揮発性	7-3
キーと値の作成	7-4
キーのライトスルー	7-3
キーのライトバック	7-3
キーのリンク	7-3

## ク

クラス	
定義	7-4

## コ

コンポーネントの登録の解除	5-29
---------------	------

## サ

作成	
プロキシ/スタブ共用可能イメージ	6-9
サブキー	7-2
サポートされる COM API	E-4
サポートされる COM API の一覧	E-4
サポートされる COM インタフェース	E-7
サポートされる COM インタフェース一覧	E-7

## シ

資格情報	12-1
資格情報の付与	7-9
シンボル・ベクタ	6-8

## ソ

相互運用性	
HostMapDomains の使用による信頼されるドメイン間の認証の構成	5-26
OpenVMS と Windows NT の構成	D-1

## ト

トラブルシューティング	
ACME サーバ	B-3
DCOMSRPCSS プロセス	B-5
RPC	B-1

## ニ

認証	12-1
----	------

## ハ

ハイブ	
定義	7-4

## フ

プロキシ/スタブ共用可能イメージ	6-9
------------------	-----

## ヘ

ベルソナ	12-1
------	------

## リ

リリース・ノート: COM for OpenVMS	
COM for OpenVMS でサポートされるスレッド・モデル	1-5
COM Version 1.1 for OpenVMS 用に OpenVMS Registry の値の再登録が必要	1-7
DCOMSCNFG ユーティリティとアプリケーションの無効化	1-4
DCOMSRPCSS は再起動時にストールする	1-8
DCOMSRPCSS プロセス・リソース全部の消費	1-1
DECwindows Motif が必要	1-2
IDispatch インタフェース	1-1
IGNORE_EXTAUTH のサポート	1-8
MIDL -w スイッチ	1-2
MIDL コンパイラは wchar_t リテラルを char として取り扱う	1-2
RPC Cannot Support エラー (800706E4)	
	1-9
RPC Server Unavailable エラー (800706BA)	
	1-9
RPC WHO_ARE_YOU エラー (EE1282FA)	
	1-9
SAFEARRAY の制約	1-6
Service Control Manager の起動とシャットダウン	1-5
SP4	1-7
アップグレードの手順	1-1
クラスタ内では COM for OpenVMS の 1 つのバージョンのみ	1-6

リリース・ノート: COM for OpenVMS (続き)	
信頼されるドメインの認証のサポート . . . . .	1-7
2つのOpenVMSシステム間でのCOM for OpenVMSの使用 . . . . .	1-8
浮動小数点 . . . . .	1-1
プロセス内サーバのリモートからのアクティブ 化 . . . . .	1-3
ローカル・サーバ・パス名に対して論理名を使用 する登録済みアプリケーション . . . . .	1-6
リリース・ノート: OpenVMS Registry	
REGS_EXQUOTA エラー . . . . .	2-2
Search コマンドの機能の制限 . . . . .	2-1
キー・アクセス・ポリシー . . . . .	2-1

キーの変更時に通知されない . . . . .	2-1
最大データ・サイズ . . . . .	2-2
最大データベース・サイズ . . . . .	2-3
リンク	
シンボル・ベクタの作成 . . . . .	6-8
プロキシ/スタブ共用可能イメージ . . . . .	6-9
プロセス外コンポーネント . . . . .	6-7
プロセス内コンポーネント . . . . .	6-8

## レ

レジストリ値 . . . . .	7-2
------------------	-----



OpenVMS Alpha オペレーティング・システム  
コネクティビティ開発者ガイド

---

1999年9月 発行

コンパックコンピュータ株式会社

〒140-8641 東京都品川区東品川 2-2-24 天王洲セントラルタワー

電話 (03)5463-6600 (大代表)

---

AA-RGLUA-TE

