

JP1 Version 10

JP1/NETM/DM API リファレンス
(UNIX(R)用)

文法書

3021-3-182

■ 対象製品

P-9D42-2CA1 JP1/NETM/DM Client 10-10 (適用 OS : Solaris)

P-1M42-2CA1 JP1/NETM/DM Client 10-10 (適用 OS : AIX)

P-1J42-2CA1 JP1/NETM/DM Client 10-10 (適用 OS : HP-UX(IPF))

P-1B42-2B81 JP1/NETM/DM SubManager 08-50 (適用 OS : HP-UX(PA-RISC))

P-9U42-2B71 JP1/NETM/DM SubManager 07-00 (適用 OS : HP Tru64 UNIX)

これらのプログラムプロダクトのほかにもこのマニュアルをご利用になれる場合があります。詳細は「リリースノート」でご確認ください。

■ 輸出時の注意

本製品を輸出される場合には、外国為替及び外国貿易法の規制並びに米国輸出管理規則など外国の輸出関連法規をご確認の上、必要な手続きをお取りください。

なお、不明な場合は、弊社担当営業にお問い合わせください。

■ 商標類

AIX は、米国およびその他の国における International Business Machines Corporation の商標です。

HP Tru64 UNIX は、Hewlett-Packard Development Company, L.P.の商標です。

HP-UX は、Hewlett-Packard Development Company, L.P.のオペレーティングシステムの名称です。

Itanium は、アメリカ合衆国およびその他の国における Intel Corporation の商標です。

Microsoft は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Oracle と Java は、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。

UNIX は、The Open Group の米国ならびに他の国における登録商標です。

Windows は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows NT は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows Vista は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

その他記載の会社名、製品名は、それぞれの会社の商標もしくは登録商標です。

プログラムプロダクト「P-9D42-2CA1」には、Oracle Corporation またはその子会社、関連会社が著作権を有している部分が含まれています。

プログラムプロダクト「P-9D42-2CA1」には、UNIX System Laboratories, Inc.が著作権を有している部分が含まれていません。

■ 発行

2013 年 12 月 3021-3-182

■ 著作権

All Rights Reserved. Copyright (C) 2013, Hitachi, Ltd.

はじめに

このマニュアルは、UNIX 版 JP1/NETM/DM Client が提供する機能のうち、API の機能と使い方について説明したものです。

■ 対象読者

JP1/NETM/DM の各プログラムの機能および操作方法についての知識を持ち、JP1/NETM/DM Client の API を使用してシステムの自動運転をしようとしている管理者の方を対象としています。

■ マニュアルの構成

このマニュアルは、次に示す章から構成されています。なお、このマニュアルは各 OS に共通のマニュアルです。OS ごとに差異がある場合は、本文中でそのつど内容を書き分けています。

第 1 章 概要

JP1/NETM/DM Client が提供する関数の概要について説明しています。

第 2 章 配布管理の関数

JP1/NETM/DM Client の配布管理システムで使用する、ソフトウェアの配布管理のための関数について説明しています。

第 3 章 配布状況ファイルアクセス関数

中継システムから配布状況ファイルにアクセスするための関数について説明しています。

■ 関連マニュアル

このマニュアルの関連マニュアルを次に示します。必要に応じてお読みください。

- JP1 Version 10 JP1/NETM/DM Client(UNIX(R)用) (3021-3-181)
- JP1 Version 9 JP1/NETM/DM Client(UNIX(R)用) (3020-3-S85)
- JP1 Version 8 JP1/NETM/DM SubManager(UNIX(R)用) (3020-3-L42)
- JP1 Version 7i JP1/NETM/DM SubManager(UNIX(R)用) (3020-3-G36)

■ このマニュアルでの表記

このマニュアルでは、製品名称を次のように表記します。

このマニュアルでの表記		正式名称	
UNIX	AIX	AIX 5L V5.3	
		AIX 6.1	
	HP-UX	HP-UX(PA-RISC)	HP-UX 11i(PA-RISC)
			HP-UX 11i V2(PA-RISC)
		HP-UX(IPF)	HP-UX 11i V2(IPF)
			HP-UX 11i V3(IPF)
	Solaris	Solaris 9	
		Solaris 10	

このマニュアルでの表記		正式名称	
UNIX	Tru64 UNIX	HP Tru64 UNIX	
Windows	Windows 98	Microsoft(R) Windows(R) 98 Operating System	
	Windows Me	Microsoft(R) Windows(R) Millennium Edition Operating System	
	Windows NT	Windows 2000	Microsoft(R) Windows(R) 2000 Advanced Server Operating System
			Microsoft(R) Windows(R) 2000 Datacenter Server Operating System
			Microsoft(R) Windows(R) 2000 Professional Operating System
			Microsoft(R) Windows(R) 2000 Server Operating System
	Windows NT 4.0	Windows NT 4.0	Microsoft(R) Windows NT(R) Server Network Operating System Version4.0
			Microsoft(R) Windows NT(R) Workstation Operating System Version4.0
	Windows Server 2003	Windows Server 2003	Microsoft(R) Windows Server(R) 2003 R2, Datacenter Edition
			Microsoft(R) Windows Server(R) 2003 R2, Enterprise Edition
			Microsoft(R) Windows Server(R) 2003 R2, Standard Edition
			Microsoft(R) Windows Server(R) 2003, Datacenter Edition
			Microsoft(R) Windows Server(R) 2003, Enterprise Edition
			Microsoft(R) Windows Server(R) 2003, Standard Edition
	Windows Server 2003 (IPF)	Windows Server 2003 (IPF)	Microsoft(R) Windows Server(R) 2003, Datacenter Edition for Itanium-based Systems
Microsoft(R) Windows Server(R) 2003, Enterprise Edition for Itanium-based Systems			
Windows Server 2003 (x64)	Windows Server 2003 (x64)	Microsoft(R) Windows Server(R) 2003 R2, Datacenter x64 Edition	
		Microsoft(R) Windows Server(R) 2003 R2, Enterprise x64 Edition	
		Microsoft(R) Windows Server(R) 2003 R2, Standard x64 Edition	

このマニュアルでの表記		正式名称	
Windows	Windows NT	Windows Server 2003 (x64)	Microsoft(R) Windows Server(R) 2003, Datacenter x64 Edition
			Microsoft(R) Windows Server(R) 2003, Enterprise x64 Edition
			Microsoft(R) Windows Server(R) 2003, Standard x64 Edition
	Windows Server 2008		Microsoft(R) Windows Server(R) 2008 Enterprise
			Microsoft(R) Windows Server(R) 2008 Standard
	Windows Vista		Microsoft(R) Windows Vista(R) Business
			Microsoft(R) Windows Vista(R) Enterprise
			Microsoft(R) Windows Vista(R) Ultimate
	Windows XP		Microsoft(R) Windows(R) XP Home Edition Operating System
			Microsoft(R) Windows(R) XP Professional Operating System
Windows 95		Microsoft(R) Windows(R) 95 Operating System	
Windows CE		Microsoft(R) Windows(R) CE Operating System	

■ このマニュアルで使用する英略語

このマニュアルで使用する英略語の一覧を次に示します。

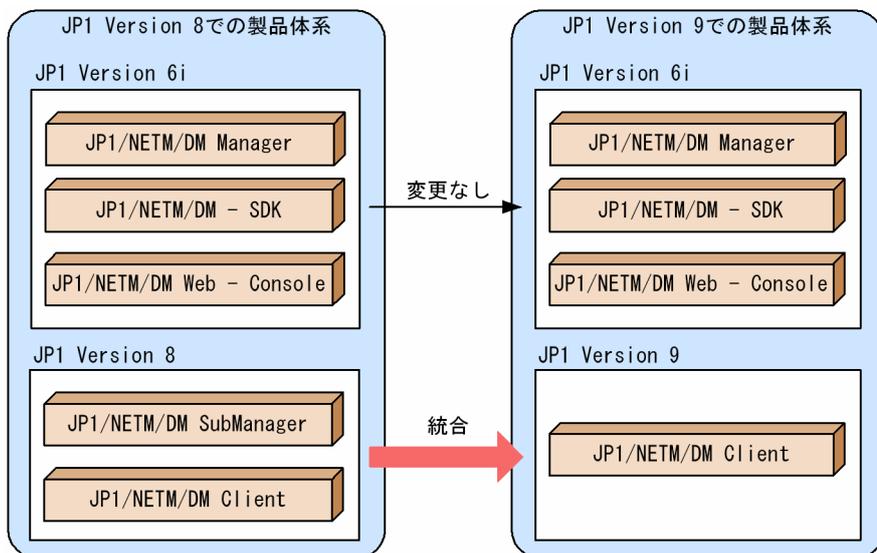
英略語	正式名称
API	Application Programming Interface
GUI	Graphical User Interface
IPF	Itanium(R) Processor Family
OS	Operating System
PC	Personal Computer
PP	Program Product
RDB	Relational Database
TCP/IP	Transmission Control Protocol/Internet Protocol
UAP	User Application Program
WS	Workstation

■ JP1 Version 9 での製品体系の変更について

JP1 Version 9 では、JP1/NETM/DM SubManager を廃止し、ソフトウェアの中継機能を JP1/NETM/DM Client に統合しました。

なお、JP1/NETM/DM Manager は、Version 6i のまま変更ありません。JP1/NETM/DM Client のバージョン 09-00 以降に対応する JP1/NETM/DM Manager の最新のバージョンは、06-75 です。

JP1 Version9 での製品体系の変更を次に示します。



■ このマニュアルの使用方法

本文中の「JP1/NETM/DM Client」という表記は、「Version 8 以前の UNIX 版 JP1/NETM/DM SubManager」に読み替えることができます。

HP-UX をご使用の方はディレクトリ名が異なります。また、HP-UX(IPF)をご使用の方は、API の共有ライブラリ、および次に示す関数の形式が異なります。

- _gaiDLVStart()
- _gaiDLVEnd()
- _gaiDLVRewind()
- _gaiDLVFind()
- _gaiDLVFindNext()
- _gaiDLVFindUnique()
- _gaiDLVDel()
- _gaiDLVDelUnique()

本文にそのつど記述していますので、ご注意ください。

■ このマニュアルで使用している記号

このマニュアルの説明で使用する記号の意味を次に示します。

(1)文法記述記号

関数の記述形式について記述する記号です。

記号	意味
{ } 波括弧	この記号で囲まれ、 (ストローク) で区切られている項目のうち、どれか一つを選んで指定することを示します。 (例) {A B C} は「A, B または C のどれか一つを指定する」ことを示します。

記号	意味
[] きっ甲	この記号で囲まれている項目を省略できることを示します。 (例) [A] は、A と指定するか、何も指定しないことを示します。
… 点線	この記号の直前に示す項目を、繰り返し複数個指定できることを示します。 (例) A, B, B, …は「A の後ろに B を必要個数だけ指定する」ことを示します。

(2)属性表示記号

関数の記述形式、関数に指定する値の範囲などについて記述する記号です。

記号	意味
~	この記号の前に示された項目が、この記号以降の< >, (())などで示す規則に従って記述されなければならないことを示します。
< >	項目を記述するときに従わなければならない構文要素記号を示します (内容については「(3)構文要素記号」を参照のこと)。
(())	指定できる値の範囲を示します。
« »	省略できる項目を省略したときに、プログラムが仮定する値を示します。

(3)構文要素記号

関数の記述形式、関数に指定する値の範囲などについて記述する記号です。

構文要素	意味
<英数字>	数字と英字。数字は、特に記載がなければ10進数を指します。

■ KB (キロバイト) などの単位表記について

1KB (キロバイト), 1MB (メガバイト), 1GB (ギガバイト), 1TB (テラバイト) はそれぞれ1,024バイト, 1,024²バイト, 1,024³バイト, 1,024⁴バイトです。

目次

1	概要	1
1.1	ライブラリ関数の種類	2
1.1.1	配布管理の関数	2
1.1.2	配布状況ファイルアクセス関数	3
1.2	ライブラリ関数の使用方法	5
1.2.1	使用言語	5
1.2.2	引数の設定方法	5
1.2.3	参照関数の使用方法	6
2	配布管理の関数	9
	使用するファイルおよび動作環境	10
	開始・終了の関数	11
	_gaiAPIStart() — API の使用開始	12
	_gaiAPIEnd() — API の使用終了	13
	操作の関数	14
	_gaiRSCCreateStore() — ソフトウェアの登録 (ソフトウェアを資源保管庫に登録する)	15
	_gaiRSCDelete() — パッケージの削除	19
	_gaiCMDCreate() — 指令の作成・登録	20
	_gaiCMDRun() — 指令の実行	23
	_gaiCMDDelete() — 指令の削除	24
	参照の関数	25
	_gaiRSCRewind() — 資源属性ファイルの参照 (ファイルの先頭レコードの参照)	26
	_gaiRSCFind() — 資源属性ファイルの参照 (先頭レコードの参照)	27
	_gaiRSCFindNext() — 資源属性ファイルの参照 (前回参照した次のレコードの参照)	28
	_gaiRSCFindUnique() — 資源属性ファイルの参照 (特定レコードの参照)	29
	_gaiRSCReferArea() — 資源属性ファイルの参照 (レコード中の特定エリアの参照)	30
	_gaiCMDRewind() — 実行管理ファイルの参照 (ファイルの先頭レコードの参照)	31
	_gaiCMDFind() — 実行管理ファイルの参照 (先頭レコードの参照)	32
	_gaiCMDFindNext() — 実行管理ファイルの参照 (前回参照した次のレコードの参照)	33
	_gaiCMDFindUnique() — 実行管理ファイルの参照 (特定レコードの参照)	34
	_gaiCMDReferArea() — 実行管理ファイルの参照 (レコード中の特定エリアの参照)	35
	API のリターンコードを参照する関数	36
	_gaiRTNGetMainCode() — 概要コードの参照	37
	_gaiRTNGetSubCode() — 詳細コードの参照	38

3	配布状況ファイルアクセス関数	39
	使用するファイル	40
	開始・終了の関数	41
	_gaiDLVStart() — 配布状況ファイルへのアクセスの開始	42
	_gaiDLVEnd() — 配布状況ファイルへのアクセスの終了	43
	レコードの参照の関数	44
	_gaiDLVRewind() — 配布状況ファイルの先頭レコードの参照	45
	_gaiDLVFind() — 配布状況ファイルの任意のレコードの参照	46
	_gaiDLVFindNext() — 配布状況ファイルの前回参照した次のレコードの参照	48
	_gaiDLVFindUnique() — 配布状況ファイルの特定レコードの参照	49
	_gaiDLVReferArea() — 配布状況ファイルのレコード中の特定エリアの参照	50
	レコードの削除の関数	51
	_gaiDLVDel() — 配布状況ファイルの任意のレコードの削除	52
	_gaiDLVDelUnique() — 配布状況ファイルの特定のレコードの削除	54
	API のリターンコードを参照する関数	55
	_gaiDLVGetMainCode() — 概要コードの参照	56
	_gaiDLVGetSubCode() — 詳細コードの参照	57

付録		59
付録 A	管理ファイルの参照エリアと識別子	60
	付録 A.1 資源属性ファイル	60
	付録 A.2 実行管理ファイル	63
	付録 A.3 配布状況ファイル	67
付録 B	リターンコード	70
	付録 B.1 配布管理の関数のリターンコード	70
	付録 B.2 配布状況ファイルアクセス関数のリターンコード	72
付録 C	各バージョンの変更内容	73

索引		75
----	--	----

1

概要

JP1/NETM/DM Client が提供するアプリケーションプログラミングインタフェース (API), およびその API を使用してアプリケーションプログラムを作成する方法の概要について説明します。

1.1 ライブラリ関数の種類

JPI/NETM/DM Client で使用できる関数には、次の 2 種類があります。

- 配布管理の関数
- 配布状況ファイルアクセス関数

1.1.1 配布管理の関数

配布管理の関数の一覧を表 1-1 に示します。

表 1-1 配布管理の関数の一覧

分類		機能	関数	
API の使用開始・終了		API の使用開始	_gaiAPIStart()	
		API の使用終了	_gaiAPIEnd()	
ソフトウェア・指令の操作	ソフトウェア操作	ソフトウェアの登録 (ソフトウェアを資源保管庫に登録する)	_gaiRSCCreateStore()	
		パッケージの削除	_gaiRSCDelete()	
	指令操作	指令の作成・登録	_gaiCMDCreate()	
		指令の実行	_gaiCMDRun()	
		指令の削除	_gaiCMDDelete()	
	管理ファイルの参照	資源属性ファイルの参照	ファイルの先頭レコードの参照	_gaiRSCRewind()
先頭レコードの参照			_gaiRSCFind()	
前回参照した次のレコードの参照			_gaiRSCFindNext()	
特定レコードの参照			_gaiRSCFindUnique()	
レコード中の特定エリアの参照			_gaiRSCReferArea()	
実行管理ファイルの参照		ファイルの先頭レコードの参照	_gaiCMDRewind()	
		先頭レコードの参照	_gaiCMDFind()	
		前回参照した次のレコードの参照	_gaiCMDFindNext()	
		特定レコードの参照	_gaiCMDFindUnique()	
		レコード中の特定エリアの参照	_gaiCMDReferArea()	
API のリターンコードの参照		概要コードの参照	_gaiRTNGetMainCode()	
		詳細コードの参照	_gaiRTNGetSubCode()	

- **API の使用開始・終了の関数**
API の開始・終了処理をする関数です。API の開始・終了時に必ず呼び出してください。
- **ソフトウェア・指令の操作の関数**
ソフトウェア、指令を操作する関数です。

- **管理ファイルの参照の関数**

管理ファイルを参照する関数です。参照の関数は、管理ファイルのレコードを一つずつ入力領域に読み込みます。

また、入力領域に読み込んだレコードの特定のエリアを参照できます。

参照の関数で参照できるファイルは、資源属性ファイルおよび実行管理ファイルです。

管理ファイルのレコードの参照方法には、ファイルの先頭レコードの参照、先頭レコードの参照、前回参照した次のレコードの参照、特定レコードの参照、およびレコード中の特定エリアの参照があります。

- **API のリターンコードを参照する関数**

実行結果を参照する関数です。実行結果を示すリターンコードとして概要コードと詳細コードがあります。

1.1.2 配布状況ファイルアクセス関数

配布状況ファイルアクセス関数の一覧を表 1-2 に示します。

表 1-2 配布状況ファイルアクセス関数の一覧

分類	機能	関数
アクセスの開始・終了	配布状況ファイルへのアクセスの開始	_gaiDLVStart()
	配布状況ファイルへのアクセスの終了	_gaiDLVEnd()
配布状況ファイルのレコードの参照	先頭レコードの参照	_gaiDLVRewind()
	任意のレコードの参照	_gaiDLVFind()
	前回参照した次のレコードの参照	_gaiDLVFindNext()
	特定レコードの参照	_gaiDLVFindUnique()
	レコード中の特定エリアの参照	_gaiDLVReferArea()
配布状況ファイルのレコードの削除	任意のレコードの削除	_gaiDLVDel()
	特定のレコードの削除	_gaiDLVDelUnique()
API のリターンコードの参照	概要コードの参照	_gaiDLVGetMainCode()
	詳細コードの参照	_gaiDLVGetSubCode()

- **アクセスの開始・終了の関数**

配布状況ファイルへのアクセスの開始・終了処理をする関数です。API の開始・終了時に必ず呼び出してください。

- **レコードの参照の関数**

配布状況ファイルを参照する関数です。参照の関数は、ファイルのレコードを一つずつ入力領域に読み込みます。参照方法には、ファイルの先頭レコードの参照、先頭レコードの参照、前回参照した次のレコードの参照、特定レコードの参照、およびレコード中の特定エリアの参照があります。

- **レコードの削除の関数**

配布状況ファイルを削除する関数です。

- **API のリターンコードを参照する関数**

1 概要

実行結果を参照する関数です。実行結果を示すリターンコードとして概要コードと詳細コードがあります。

1.2 ライブラリ関数の使用方法

JP1/NETM/DM Client が提供する API を使用して、アプリケーションプログラムを作成する方法について説明します。

1.2.1 使用言語

JP1/NETM/DM Client が提供する API を使用してアプリケーションプログラムを作成する場合、C 言語の仕様に従ってコーディングしてソースプログラムを作成します。API の呼び出しには、OS で標準的に提供されている関数と同様、引数を指定します。呼び出した関数が正常に実行したかどうかは、戻ってくる値（リターン値）で判断してください。

コンパイル方法の例を次に示します。

```
cc -o ファイル名 ファイル名.c -I/NETMRDSEX1/include -Y /NETMRDSEX1/lib/afapi.a
```

！ 注意事項

各ライブラリ関数の errno は保証されません。また、シグナルトラップは変更されません。

1.2.2 引数の設定方法

引数への値の設定方法を表 1-3 に示します。

表 1-3 引数に設定する値の条件

引数	値の設定方法
配布先名	<p>配布先名とは、パッケージの配布先である配布先システム名のことで、配布先名はホスト名で指定します。</p> <ul style="list-style-type: none"> • <code>_gaiCMDCreate()</code> の場合 [ホスト名] • <code>_gaiCMDCreate()</code> 以外の関数の場合 [ホスト名...ホスト名]
グループ id	<p>全角文字、および次の文字を除く半角文字を使用できます。 「! (感嘆符)」 「" (ダブルクォーテーション)」 「% (パーセント)」 「' (シングルクォーテーション)」 「* (アスタリスク)」 「. (ピリオド)」 「/ (スラント)」 「: (コロン)」 「< (山括弧)」 「> (山括弧)」 「? (疑問符)」 「@ (単価記号)」 「¥ (円記号)」 「 (ストローク)」</p> <p>文字数は半角文字で最大 32 文字です。</p> <p>異なるプラットフォームが混在するシステムでグループ id を定義する場合、プラットフォーム間で文字コードを一致させてください。文字コードが異なると配布できない場合があります。</p> <p>配布先システムから定義したグループ id については、各配布先システムのマニュアルを参照してください。</p>
完全資源名	<p>完全資源名は、資源保管庫のパッケージの名称で、システム内でユニークです。</p> <ul style="list-style-type: none"> • 資源保管庫内のパッケージの完全資源名 [DM 種別コード.グループ名.資源名. [[バージョン/リビジョン] [.世代番号]]]
DM 種別コード	<p>パッケージを登録したシステムの種別を指定します。</p> <p>C: UNIX システム D: Windows システム</p>

引数	値の設定方法
グループ名	資源保管庫内の資源グループを指定します。#@%を除く大文字英数字 2 文字で指定します。
資源名	パッケージング時に指定したパッケージの名称です。#@%を除く、1~44 文字の大文字英数字で指定します。
バージョン/リビジョン	パッケージのバージョン/リビジョンです。#@%を除く、1~8 けたの大文字英数字で指定します。/(スラント)も使用できます。
世代番号	パッケージの世代番号です。#@%を除く大文字英数字で指定します。
配布先指令番号	指令を管理するために JP1/NETM/DM が付ける char 型 16 けたの 16 進数です。rdsafind コマンドまたは実行管理ファイルを参照して番号を確認してください。

1.2.3 参照関数の使用方法

参照の関数は、参照する管理ファイルの情報を 1 レコードずつ入力領域に読み込みます。読み込んだレコードは識別子を設定すると、レコードの特定エリアを参照できます。

参照できる管理ファイルは、資源属性ファイルおよび実行管理ファイルです。これらのファイルは、関数を呼び出して次のことができます。

- ファイルの先頭レコードの参照
- 条件を指定したレコードの参照
- 次レコードの参照
- 特定のレコードの参照
- レコードの特定エリアの参照

参照の関数の使い方

関数中の XXX には、参照するファイルに応じて、RSC または CMD が入ります。

- ファイルのレコードを先頭から格納順で連続して参照する場合は、_gaiXXXRewind()に続けて _gaiXXXFindNext()を呼び出します。
- 条件を指定してレコードを参照する場合は、_gaiXXXFind()を呼び出します。
- _gaiXXXFind()に続けて_gaiXXXFindNext()を呼び出すと同一条件のレコードを連続して参照できます。ただし、一つのアプリケーションプログラムでは一つの条件だけしか設定できません。異なる条件で参照する場合は、別のアプリケーションプログラムで参照してください。
- 次レコードを参照する関数は、_gaiXXXRewind()または_gaiXXXFind()のあとで使用します。次レコードを参照する関数を先に呼び出すと要求順序不正のエラーになります。
- 特定のレコードを参照する場合は、_gaiXXXFindUnique()を呼び出します。
- 入力領域に読み込んだレコードの特定エリアを参照する場合は、_gaiXXXReferArea()を呼び出します。_gaiXXXReferArea()に参照するエリアの識別子を設定します。識別子は、「付録 A 管理ファイルの参照エリアと識別子」をファイルに応じて参照してください。

参照の関数の詳細については「2. 配布管理の関数」を参照してください。

(1) 資源属性ファイルの参照

資源属性ファイルでは、パッケージと配布の情報（資源名、バージョン/リビジョン、配布先、組み込み形式など）を参照できます。

資源属性ファイルは、ヘッダ部とエントリ部で構成されています。ヘッダ部には資源グループの情報が格納されています。エントリ部にはパッケージと配布の情報が格納されています。

資源属性ファイルのレコードは次の順でソートされています。

DM種別コード→グループ名→資源名→バージョン/リビジョン→世代番号

資源属性ファイルを参照する関数は、_gaiRSCRewind(), _gaiRSCFind(), _gaiRSCFindNext(), _gaiRSCFindUnique()です。

読み込んだレコードの特定エリアを参照する場合は_gaiRSCReferArea()を使用します。

資源属性ファイルで、参照できるエリアとその識別子については「付録 A.1 資源属性ファイル」を参照してください。

(2) 実行管理ファイルの参照

実行管理ファイルでは、指令の情報（指令種別、配布先名、組み込みの形式など）やパッケージの転送状況を参照できます。

実行管理ファイルのレコードは、次の順でソートされています。

配布先名→（グループid）→指令番号

(a) 使用する関数

実行管理ファイルを参照する関数は、_gaiCMDRewind(), _gaiCMDFind(), _gaiCMDFindNext(), _gaiCMDFindUnique()です。読み込んだレコードの特定エリアを参照する場合は_gaiCMDReferArea()を使用します。

(b) 実行できる指令

JPI/NETM/DM Client で実行できる指令は配布・組み込み指令だけです。

実行管理ファイルで参照できるエリアとその識別子については、「付録 A.2 実行管理ファイル」を参照してください。

2

配布管理の関数

JP1/NETM/DM Client が提供する配布管理の関数について説明します。

使用するファイルおよび動作環境

JP1/NETM/DM Client の API に関連するファイルについて説明します。

■ライブラリ関数に必要なファイル

JP1/NETM/DM Client は、運用コマンドで提供しているサービスをアプリケーションプログラムで実現できるように API を次のファイルで提供しています。

共用ライブラリ

/NETMRDSEX1/lib/afapi.sl … HP-UX(PA-RISC), Solaris の場合

/NETMRDSEX1/lib/afapi.o … AIX の場合

/NETMRDSEX1/lib/afapi.so … Tru64 UNIX の場合

/opt/NETMDMWEX1/lib/afapi.sl … HP-UX(IPF)で 32 ビットのアプリケーションを作成する場合

/opt/NETMDMWEX1/lib/afapi64.sl … HP-UX(IPF)で 64 ビットのアプリケーションを作成する場合

ヘッダファイル

/NETMRDSEX1/include/afapi.h

なお、OS が HP-UX の場合、ファイル名は/opt/NETMDMWEX1~ (以下同じ) となります。

■サンプルプログラムのファイル

JP1/NETM/DM Client の API を使ったサンプルプログラムは、次のファイル名で格納してあります。ライブラリ関数の標準的な使い方を知りたい場合に参照してください。

サンプルプログラム

/NETMRDSEX1/sample/rscdpsmp.c

なお、OS が HP-UX の場合ファイル名は/opt/NETMDMWEX1~ (以下同じ) となります。

■動作環境

API の動作環境を次に示します。

- API は、JP1/NETM/DM Client の配布管理機能が稼働しているときだけ使用できます。
- JP1/NETM/DM Client の配布管理機能を `rdsafse -p` コマンドで計画停止させると、API の処理が終了してから停止します。速やかに計画停止させるためには、プログラム中で API の終了処理をする必要があります。
- JP1/NETM/DM Client の配布管理機能が強制終了すると、その時点で API は使用できなくなります。

開始・終了の関数

`_gaiAPIStart()`は API の環境を準備する関数で、`_gaiAPIEnd()`は API を終了する関数です。API の開始・終了時には必ず呼び出します。

次に、開始・終了の関数の詳細を説明します。

_gaiAPIStart() — API の使用開始

形式

```
#include "NETMRDSEX1/include/afapi.h"
int _gaiAPIStart(int *cnt_inf, unsigned int *status, char *opt);
```

機能

API を使用するための準備をします。この関数が正常終了したあと、_gaiAPIEnd()を発行するまで、ライブラリ中の関数を繰り返し使用できます。API 使用中に JP1/NETM/DM Client の配布管理機能が停止した場合は、この関数を再度発行する必要があります。

プログラムで値を設定する引数

opt

将来の拡張に備えるための領域です。NULL 文字を指定します。

JP1/NETM/DM Client から値が返される引数

cnt_inf

API を使用するための制御情報が返されます。

status

処理結果の詳細情報が返されます。

リターン値

0 : 正常終了

-1 : 異常終了

_gaiAPIEnd() — API の使用終了

形式

```
#include "NETMRDSEX1/include/afapi.h"  
int _gaiAPIEnd(int *cnt_inf, unsigned int *status);
```

機能

API の終了処理をします。API の使用を終了する場合は必ずこの関数を発行してください。

プログラムで値を設定する引数

cnt_inf

_gaiAPIStart()の第 1 引数に返された値を指定します。

JP1/NETM/DM Client から値が返される引数

status

処理結果の詳細情報が返されます。

リターン値

0：正常終了

-1：異常終了

操作の関数

JPI/NETM/DM Client で使用できる配布管理の関数のうち、操作の関数には、ソフトウェア操作および指令操作の関数があります。

操作の関数は一つのアプリケーションプログラムから何度でも呼び出せます。

各操作の対象別の関数には、次のものがあります。

■ソフトウェア操作

ソフトウェア操作では次の操作ができます。

- ソフトウェアの登録（ソフトウェアを資源保管庫に登録する）：`_gaiRSCCreateStore()`
- パッケージの削除：`_gaiRSCDelete()`

■指令操作

指令操作では、パッケージの配布・組み込み指令を操作できます。

- 指令の作成・登録：`_gaiCMDCreate()`
- 指令の実行：`_gaiCMDRun()`
- 指令の削除：`_gaiCMDDelete()`

次に、各操作の関数の詳細を上記の順に説明します。

`_gaiRSCCreateStore()` — ソフトウェアの登録 (ソフトウェアを資源保管庫に登録する)

形式

```
#include "/NETMRDSEX1/include/afapi.h"
int _gaiRSCCreateStore(int *cnt_inf, unsigned int *status, char
    *passwd, char *rscname, char *rscpath
    [, DMGAI_RSC_CRT_INSDIR, char *insdir]
    [, DMGAI_RSC_CRT_INSMODE, char *insmode]
    [, DMGAI_RSC_CRT_INSOWN, char *insown]
    [, DMGAI_RSC_CRT_INSGROUP, char *insgroup]
    [, DMGAI_RSC_CRT_INSTIM, char *instim]
    [, DMGAI_RSC_CRT_INSDAY, char *insday]
    [, DMGAI_RSC_CRT_PREVER, char *prever]
    [, DMGAI_RSC_CRT_KEEPPDATE, char *keepdate]
    [, DMGAI_RSC_CRT_PRGNM, char *prgnm]
    [, DMGAI_RSC_CRT_APPRG, char *apprg]
    [, DMGAI_RSC_CRT_APWATCH, char *apwatch]
    [, DMGAI_RSC_CRT_OWNER, char *owner]
    [, DMGAI_RSC_CRT_RCVRY, char *rcvry]
    [, DMGAI_RSC_CRT_KEEPPBACKUP, char *keepbackup]
    [, DMGAI_RSC_CRT_LISTBACKUP, char *listbackup]
    [, DMGAI_RSC_CRT_SCRIPTF, char *scriptf]
    [, DMGAI_RSC_CRT_COMPRESS, char *compress]
    [, DMGAI_RSC_CRT_ENCODE, char *encode]
    [, DMGAI_RSC_CRT_INSTALM, char *instalM]
    , 0)
```

機能

資源保管庫にソフトウェアを登録します。

プログラムで値を設定する引数

`cnt_inf`

`_gaiAPIStart()`の第1引数で返された値を指定します。

`passwd`

資源登録パスワードを指定します。

JPI/NETM/DM Clientの配布管理機能の設定ファイル中の「資源アップロードパスワード (PasswordForUpload)」で指定した値を指定します。

`rscname`

登録するソフトウェアの完全資源名を指定します。

DM種別コードはC (UNIX)を設定してください。

`rscpath` ~((1~128文字))

ユーザ資源パスを指定します。

登録するソフトウェアのディレクトリ名またはファイル名を、絶対パスまたは相対パスで指定します。相対パスで指定した場合は、ソフトウェアの組み込み先ディレクトリを変更することができます。絶対パス指定の場合は変更できません。

このオペランドに指定したユーザ資源のユーザ名およびグループ名は/etc/passwdに定義されている必要があります。

`insdir` ~((1~64文字))

登録するソフトウェアの組み込み先ディレクトリ名を指定します。

組み込み先ディレクトリ名は相対パス指定で登録するとき指定できます。指定されたディレクトリが未定義、または省略時は/を仮定します。

insmode ~((700~777))<<755>>

組み込み先ディレクトリのファイルモードを 700 から 777 の範囲で指定します。

組み込み先ディレクトリに指定されたディレクトリが、組み込み先に存在しない場合に指定します。ファイルモードを指定する場合、ディレクトリの所有者とディレクトリのグループを必ず指定してください。省略時は 755 を仮定します。

insown ~((0~59999))<<0>>

組み込み先ディレクトリの所有者を 0 から 59999 の範囲で指定します。

組み込み先ディレクトリに指定されたディレクトリが、組み込み先に存在しない場合に指定します。指定された所有者が未定義、または省略時はroot (ユーザ番号 0) を仮定します。

insgroup ~((0~59999))<<3>>

組み込み先ディレクトリのグループを 0 から 59999 の範囲で指定します。

組み込み先ディレクトリに指定されたディレクトリが存在しない場合に指定します。ディレクトリのグループを指定する場合、必ずディレクトリの所有者を指定してください。指定されたグループ番号が未定義、または省略時はsys (グループ番号 3) を仮定します。

instim

組み込みタイミングを次の形式で指定します。

DMGAI_RSC_CRT_INSTIM_BOOT

システム起動時にパッケージを組み込む場合

DMGAI_RSC_CRT_INSTIM_EXEC

システム稼働中にパッケージを組み込む場合

DMGAI_RSC_CRT_INSTIM_SHUT

システム停止時にパッケージを組み込む場合

DMGAI_RSC_CRT_INSTIM_INSM

UAP の指示によって組み込む場合

insday

組み込み日時を *YYYYMMDDhhmm* の形式で指定します。

prever ~<英数字 (大文字) とスラント>((1~6 文字))<<000000>>

前提バージョン/リビジョン (1~6 文字) を英数字 (大文字) と/で指定します。

省略した場合は、前提バージョン/リビジョンのチェックはしません (無条件配布)。000000 を指定すると、無条件配布となります。

keepdate ~<<登録日の翌日>>

登録するソフトウェアが配布されたときに、資源保管庫に保管される期限 (パッケージの満了日付) を *YYYYMMDD* の形式で指定します。

パッケージの満了日付は、パッケージング日以降を指定します。パッケージの満了日付が正しくない場合、または省略時はパッケージング日の翌日を仮定します。

prgnm ~((1~50 文字))

プログラム名を指定します。省略時は完全資源名中の資源名を仮定します。

apprg ~((1~64 文字))

パッケージの組み込み後に起動するユーザ組み込み処理を絶対パス (1~64 文字) で指定します。

apwatch ~((0~999))<<0>> (単位：秒)

ユーザ組み込み処理の実行監視時間を 0 から 999 の範囲で秒単位で指定します。

0 を指定した場合または省略した場合は、ユーザ組み込み処理を監視しません。

owner ~((1~14 文字))

登録するソフトウェアの所有者を指定します。

省略した場合は、ユーザ資源のユーザ名を仮定します。

rcvry ~<<DMGAI_RSC_CRT_RCVRY_NO>>

パッケージの配布時に組み込みが失敗した場合、パッケージのリカバリを実行するかどうか（リカバリ機能を使用するかどうか）を次の形式で指定します。省略時はDMGAI_RSC_CRT_RCVRY_NO を仮定します。

DMGAI_RSC_CRT_RCVRY_YES

リカバリを実行する

DMGAI_RSC_CRT_RCVRY_NO

リカバリを実行しない

keepbackup ~((0~999))<<0>>

リカバリ機能を使用する場合、取得したバックアップの保存期間を 0 から 999 の範囲の日数で指定します。リカバリ機能を使用しない場合、この指定は無効になります。

バックアップは、パッケージ配布日からここで指定した日数後に削除されます。

省略時は 0（保存しない）を仮定します。リカバリを再試行しないパッケージであれば、0 を指定してください。

listbackup ~((1~128 文字))

バックアップリスト名を指定します。

リカバリ機能を使用する場合、取得するバックアップのファイル名およびディレクトリ名を記述したバックアップリストのファイル名を指定します。

リカバリ機能を使用しない場合、この指定は無効になります。

scriptf ~((1~128 文字))

インストールスクリプトファイル名を指定します。

compress ~<<DMGAI_RSC_CRT_COMPRESS_NO>>

圧縮オプションを次の形式で指定します。省略時はDMGAI_RSC_CRT_COMPRESS_NO を仮定します。

DMGAI_RSC_CRT_COMPRESS_YES

登録するソフトウェアを圧縮する

DMGAI_RSC_CRT_COMPRESS_NO

登録するソフトウェアを圧縮しない

encode ~<<DMGAI_RSC_CRT_ENCODE_NO>>

暗号オプションを次の形式で指定します。省略時はDMGAI_RSC_CRT_ENCODE_NO（暗号化しない）を仮定します。DMGAI_RSC_CRT_ENCODE_YES（暗号化する）を設定する場合は、JP1/NETM/DM Encryption Option が必要です。

DMGAI_RSC_CRT_ENCODE_YES

登録するソフトウェアを暗号化する

DMGAI_RSC_CRT_ENCODE_NO

登録するソフトウェアを暗号化しない

instalml ~<<DMGAI_RSC_CRT_INSTALM_BACK>>

インストールモードを次の形式で指定します。省略時はDMGAI_RSC_CRT_INSTALM_BACKを仮定します。

DMGAI_RSC_CRT_INSTALM_GUI

GUI インストールモード

DMGAI_RSC_CRT_INSTALM_BACK

バックグラウンドインストールモード

JP1/NETM/DM Client から値が返される引数

status

処理結果の詳細情報が返されます。

リターン値

0：正常終了

-1：異常終了

注意事項

引数の最後には必ず 0 を設定してください。

_gaiRSCDelete() — パッケージの削除

形式

```
#include "NETMRDSEX1/include/afapi.h"  
int _gaiRSCDelete(int *cnt_inf, unsigned int *status, char *key);
```

機能

資源保管庫内のパッケージを削除します。

プログラムで値を設定する引数

cnt_inf

_gaiAPIStart()の第1引数で返された値を指定します。

key

削除するパッケージを次の形式で指定します。

DM種別コード.グループ名.資源名.バージョン/リビジョン.世代番号

JP1/NETM/DM Client から値が返される引数

status

処理結果の詳細情報が返されます。

リターン値

0：正常終了

-1：異常終了

注意事項

削除したパッケージの配布指令が登録されていると、配布先のパッケージ取り込み時に矛盾が発生するので、配布指令が登録されているパッケージを削除しないでください。

`_gaiCMDCreate()` — 指令の作成・登録

形式

```
#include "/NETMRDSEX1/include/afapi.h"
int _gaiCMDCreate(int *cnt_inf, unsigned int *status, char *cmd_kind,
                 char *key, char *cmdno
                 {, DMGAI_CMD_CRT_DLVNM, char *addr
                   |, DMGAI_CMD_CRT_DLVID, char *gid}
                 [, DMGAI_CMD_CRT_TIMNG, char *timing]
                 [, DMGAI_CMD_CRT_INSDAY, char *insday]
                 [, DMGAI_CMD_CRT_INSDIR, char *insdir]
                 [, DMGAI_CMD_CRT_INSOWN, char *insown]
                 [, DMGAI_CMD_CRT_INSGROUP, char *insgroup]
                 [, DMGAI_CMD_CRT_INSMODE, char *insmode]
                 , 0)
```

機能

指令を実行管理ファイルに登録します。

プログラムで値を設定する引数

cnt_inf

`_gaiAPIStart()`の第1引数に返された値を指定します。

cmd_kind

指令種別を次のように指定します。

DMGAI_CMD_KIND_D

配布・組み込み指令

key

配布対象となるパッケージを次の形式で指定します。

DM種別コード.グループ名.資源名.バージョン/リビジョン.世代番号

addr

配布先をホスト名で指定します。

gid

配布先のグループidを指定します。

timing

パッケージの組み込みタイミングを次の形式で指定します。

DMGAI_TIMING_NORMAL

配布先に同一パッケージが存在する場合はパッケージを組み込まない。

DMGAI_TIMING_FORCE

配布先に同一パッケージが存在しても、すでにあるパッケージと置き換えて組み込む。

DMGAI_TIMING_BOOT

パッケージの組み込みを配布先システムの起動時に実行する。

DMGAI_TIMING_SHUT

パッケージの組み込みを配布先システムの停止時に実行する。

DMGAI_TIMING_EXEC

パッケージの組み込みを配布先システムの稼働中に実行する。

DMGAI_TIMING_INSM

UAP の指示によってパッケージを組み込む。

insday

組み込み日時を *YYYYMMDDhhmm* の形式で指定します。

insdir ~<英数字>((1~64 文字))

組み込み先ディレクトリ名を指定します。

パッケージング時に指定した組み込み先ディレクトリを変更する場合に指定します。

ソフトウェアをハードディスクから相対パス指定で登録した場合に指定できます。指定したディレクトリが配布先システムにない場合は、パッケージを組み込むときに作成して組み込まれます。ディレクトリ名は、配布先システムの種類によって次の形式で指定してください。

配布先システムが UNIX の場合

[/] で始まる名称 (絶対パス名)

配布先システムが Windows の場合

[¥] で始まる名称 (絶対パス名) またはドライブ名で始まる名称を ['] で囲んだ形式

このオプションは、UNIX マシンから登録したパッケージを配布する場合に指定できます。

insown ~((0~59999))<0>

組み込み先ディレクトリの所有者を 0 から 59999 の範囲で指定します。

パッケージング時に指定した組み込み先ディレクトリの所有者を変更する場合に指定します。指定した組み込み先ディレクトリが配布先システムにない場合の所有者を、ユーザ番号で指定します。指定した所有者が未定義の場合は、root (ユーザ番号 0) が仮定されます。

このオプションは、UNIX マシンから登録したパッケージを配布する場合に指定できます。

insgroup ~((0~59999))<3>

組み込み先ディレクトリのグループを 0 から 59999 の範囲で指定します。

パッケージング時に指定した組み込み先ディレクトリのグループを変更する場合に指定します。指定した組み込み先ディレクトリが配布先システムにない場合のグループを、グループ番号で指定します。指定したグループ番号が未定義の場合は、sys (グループ番号 3) が仮定されます。

このオプションは、UNIX マシンから登録したパッケージを配布する場合に指定できます。

insmode ~((700~777))

組み込み先ディレクトリのモードを 700 から 777 の範囲で指定します。

パッケージング時に指定した組み込み先ディレクトリのモードを変更する場合に指定します。指定した組み込み先ディレクトリが配布先システムにない場合のファイルモードを指定します。ファイルモードは、パーミッションビットだけを絶対値で指定してください。

このオプションは、UNIX マシンから登録したパッケージを配布する場合に指定できます。

JP1/NETM/DM Client から値が返される引数**status**

処理結果の詳細情報が返されます。

cmdno

指令番号が返されます。領域は、サイズに *DMGAI_CMDNO_SIZE* を指定して確保してください。

リターン値

0: 正常終了

2 配布管理の関数

-1：異常終了

注意事項

引数の最後には必ず 0 を設定してください。

`_gaiCMDRun()` — 指令の実行

形式

```
#include "/NETMRDSEX1/include/afapi.h"  
int _gaiCMDRun(int *cnt_inf, unsigned int *status);
```

機能

実行管理ファイルに登録されている指令を実行します。

プログラムで値を設定する引数

`cnt_inf`

`_gaiAPIStart()`の第1引数に返された値を指定します。

JP1/NETM/DM Client から値が返される引数

`status`

処理結果の詳細情報が返されます。

リターン値

0：正常終了

-1：異常終了

`_gaiCMDDelete()` — 指令の削除

形式

```
#include "/NETMRDSEX1/include/afapi.h"
int _gaiCMDDelete(int *cnt_inf, unsigned int *status,
                 char *del_kind, char *cmdno
                 {, DMGAI_CMD_DEL_DLVM, char *addr
                  |, DMGAI_CMD_DEL_DLVID, char *gid}
                 , 0)
```

機能

実行管理ファイルに登録されている指令を削除します。また、配布先の指定誤りなどによって配布先システムに滞留している、配布済みの指令を取り消します。

プログラムで値を設定する引数

`cnt_inf`

`_gaiAPIStart()`の第1引数に返された値を指定します。

`del_kind`

削除種別を次の形式で指定します。

`DMGAI_DELKIND_NORMAL`

配布先システムに滞留している指令がある場合に、それらを削除してから実行管理ファイル中の指令を削除します。

`DMGAI_DELKIND_FORCE`

指令の状態に関係なく、実行管理ファイルから削除します。

`cmdno` ~((16けた))

削除する指令の指令番号(16けた)を指定します。

`addr`

削除する指令の配布先をホスト名で指定します。

`gid` ~((2けた))

グループidを指定した指令を削除する場合にグループid(2けた)を指定します。

グループidを指定した指令を削除する場合は必ず指定します。

JP1/NETM/DM Client から値が返される引数

`status`

処理結果の詳細情報が返されます。

リターン値

0: 正常終了

-1: 異常終了

注意事項

引数の最後には必ず0を設定してください。

参照の関数

JP1/NETM/DM Client で使用できる配布管理の関数のうち、参照の関数には次のものがあります。

■資源属性ファイルの参照

資源属性ファイルを参照するには、次の関数を使用できます。

- ファイルの先頭レコードの参照：_gaiRSCRewind()
- 先頭レコードの参照：_gaiRSCFind()
- 前回参照した次のレコードの参照：_gaiRSCFindNext()
- 特定レコードの参照：_gaiRSCFindUnique()
- レコード中の特定エリアの参照：_gaiRSCReferArea()

資源属性ファイルで参照できるエリアとその識別子については、「付録 A.1 資源属性ファイル」を参照してください。

■実行管理ファイルの参照

実行管理ファイルを参照するには、次の関数を使用できます。

- ファイルの先頭レコードの参照：_gaiCMDRewind()
- 先頭レコードの参照：_gaiCMDFind()
- 前回参照した次のレコードの参照：_gaiCMDFindNext()
- 特定レコードの参照：_gaiCMDFindUnique()
- レコード中の特定エリアの参照：_gaiCMDReferArea()

実行管理ファイルで参照できるエリアとその識別子、および参照エリアと指令の組み合わせについては、「付録 A.2 実行管理ファイル」を参照してください。

次に、各参照の関数の詳細を上記の順に説明します。

`_gaiRSCRewind()` — 資源属性ファイルの参照 (ファイルの先頭レコードの参照)

形式

```
#include "/NETMRDSEX1/include/afapi.h"  
int _gaiRSCRewind(int *cnt_inf, unsigned int *status, char *getarea);
```

機能

資源属性ファイル中に「DM 種別コード→グループ名→資源名→バージョン/リビジョン→世代番号」の順に昇順にソートされている資源属性レコードの、先頭のレコードを入力領域に読み込みます。

この関数の終了後、`_gaiRSCFindNext()`を繰り返し発行することで、次のレコードを連続して読み込みます。なお、入力領域に読み込まれたレコードから特定の属性情報を取り出す場合は、`_gaiRSCReferArea()`を使用してください。

プログラムで値を設定する引数

`cnt_inf`

`_gaiAPIStart()`の第1引数に返された値を指定します。

`getarea`

レコード入力領域を指定します。入力領域は、サイズに`DMGAI_AFILE_SIZE`を指定して確保してください。

JP1/NETM/DM Client から値が返される引数

`status`

処理結果の詳細が返されます。

リターン値

0 : 正常終了

-1 : 異常終了

`_gaiRSCFind()` — 資源属性ファイルの参照 (先頭レコードの参照)

形式

```
#include "NETMRDSEX1/include/afapi.h"
int _gaiRSCFind (int *cnt_inf, unsigned int *status, char *key,
                char *getarea);
```

機能

資源属性ファイル中に「DM 種別コード→グループ名→資源名→バージョン/リビジョン→世代番号」の順に昇順にソートされている資源属性レコードの中から、条件に一致する最初のレコードを入力領域に読み込みます。

この関数の終了後、`_gaiRSCFindNext()`を繰り返し発行することで、同じ条件の次のレコードを連続して読み込むことができます。なお、入力領域に読み込まれたレコードから特定の属性情報を取り出す場合は、`_gaiRSCReferArea()`を使用してください。

プログラムで値を設定する引数

`cnt_inf`

`_gaiAPIStart()`の第1引数に返された値を指定します。

`key`

参照するパッケージの条件を次の形式で指定します。

資源属性ファイルのヘッダ部を参照する場合

DM種別コード [.グループ名]

資源属性ファイルのエントリ部を参照する場合

DM種別コード.グループ名.資源名 [.バージョン/リビジョン [.世代番号]]

`getarea`

レコード入力領域を指定します。入力領域は、サイズに`DMGAI_AFILE_SIZE`を指定して確保してください。

JP1/NETM/DM Client から値が返される引数

`status`

処理結果の詳細情報が返されます。

リターン値

0 : 正常終了

-1 : 異常終了

_gaiRSCFindNext() — 資源属性ファイルの参照 (前回参照した次のレコードの参照)

形式

```
#include "NETMRDSEX1/include/afapi.h"  
int _gaiRSCFindNext(int *cnt_inf, unsigned int *status, char *getarea);
```

機能

資源属性ファイル中の資源属性レコードの中から、前回参照したレコードの、次のレコードを入力領域に読み込みます。

この関数は、_gaiRSCRewind()または_gaiRSCFind()のあとに、繰り返し発行できます。

_gaiRSCFind()のあとに発行した場合は、そこで指定した条件に一致する次のレコードを読み込みます。なお、入力領域に読み込まれたレコードから特定の属性情報を取り出す場合は、_gaiRSCReferArea()を使用してください。

プログラムで値を設定する引数

cnt_inf

_gaiAPIStart()の第1引数に返された値を指定します。

getarea

レコード入力領域を指定します。入力領域は、サイズにDMGAI_AFIL_SIZEを指定して確保してください。

JP1/NETM/DM Client から値が返される引数

status

処理結果の詳細情報が返されます。

リターン値

0 : 正常終了

-1 : 異常終了

`_gaiRSCFindUnique()` — 資源属性ファイルの参照 (特定レコードの参照)

形式

```
#include "NETMRDSEX1/include/afapi.h"
int _gaiRSCFindUnique(int *cnt_inf, unsigned int *status, char *key,
                    char *getarea);
```

機能

資源属性ファイルの中から、指定された完全資源名に一致する資源属性レコードを、入力領域に読み込みます。入力領域に読み込まれたレコードから特定の属性情報を取り出す場合は、`_gaiRSCReferArea()`を使用してください。

プログラムで値を設定する引数

`cnt_inf`

`_gaiAPIStart()`の第1引数に返された値を指定します。

`key`

参照するパッケージを次の形式で指定します。

資源属性ファイルのヘッダ部を参照する場合

DM種別コード.グループ名

資源属性ファイルのエントリ部を参照する場合

DM種別コード.グループ名.資源名.バージョン/リビジョン.世代番号

`getarea`

レコード入力領域を指定します。入力領域は、サイズに`DMGAI_AFILE_SIZE`を指定して確保してください。

JP1/NETM/DM Client から値が返される引数

`status`

処理結果の詳細情報が返されます。

リターン値

0: 正常終了

-1: 異常終了

`_gaiRSCReferArea()` — 資源属性ファイルの参照 (レコード中の特定エリアの参照)

形式

```
#include "NETMRDSEX1/include/afapi.h"
int _gaiRSCReferArea(char *getarea, unsigned int *status,
                    unsigned int code, {char | int} *area
                    [, unsigned int code, {char | int} *area] ... ,0);
```

機能

資源情報ファイルを参照する関数 (`_gaiRSCRewind()`, `_gaiRSCFind()`, `_gaiRSCFindNext()`, `_gaiRSCFindUnique()`) で入力領域に読み込んだレコードの、特定の属性情報を参照します。属性情報については、「付録 A.1 資源属性ファイル」を参照してください。

プログラムで値を設定する引数

getarea

`_gaiRSCRewind()`, `_gaiRSCFind()`, `_gaiRSCFindNext()`, または `_gaiRSCFindUnique()` を使用して資源レコードを読み込んだ入力領域を指定します。

code

参照したい属性情報の識別子を指定します。code は、area と組にして繰り返し指定できます。

area

参照するデータを格納する領域を指定します。領域は、データのサイズに 1 バイトを加えた大きさで準備してください。また、領域の型は、参照するデータの型 (char 型または int 型) へのポインタとしてください。

JP1/NETM/DM Client から値が返される引数

status

処理結果の詳細情報が返されます。

リターン値

0 : 正常終了

-1 : 異常終了

注意事項

引数の最後には必ず 0 を設定してください。

_gaiCMDRewind() — 実行管理ファイルの参照 (ファイルの先頭レコードの参照)

形式

```
#include "/NETMRDSEX1/include/afapi.h"
int _gaiCMDRewind(int *cnt_inf, unsigned int *status, char *getarea);
```

機能

実行管理ファイル中に「配布先名→グループ id→配布先指令番号」の順で昇順にソートされているレコードの中から、先頭のレコードを入力領域に読み込みます。

この関数の終了後、_gaiCMDFindNext()を繰り返し発行することで、次のレコードを連続して読み込みます。なお、入力領域に読み込まれたレコードから特定の属性情報を取り出す場合は、_gaiCMDReferArea()を使用してください。

プログラムで値を設定する引数

cnt_inf

_gaiAPIStart()の第1引数に返された値を指定します。

getarea

レコード入力領域を指定します。入力領域は、サイズにDMGAI_EFILE_SIZEを指定して確保してください。

JP1/NETM/DM Client から値が返される引数

status

処理結果の詳細情報が返されます。

リターン値

0：正常終了

-1：異常終了

`_gaiCMDFind()` — 実行管理ファイルの参照 (先頭レコードの参照)

形式

```
#include "/NETMRDSEX1/include/afapi.h"
int _gaiCMDFind(int *cnt_inf, unsigned int *status, char *key,
               char *getarea);
```

機能

実行管理ファイル中に「配布先名→グループ id→配布先指令番号」の順で昇順にソートされているレコードの中から、条件に一致する最初のレコードを入力領域に読み込みます。

この関数の終了後、`_gaiCMDFindNext()`を繰り返し発行することで、同じ条件の次のレコードを連続して読み込みます。なお、入力領域に読み込まれたレコードから特定の属性情報を取り出す場合は、`_gaiCMDReferArea()`を使用してください。

プログラムで値を設定する引数

`cnt_inf`

`_gaiAPIStart()`の第1引数に返された値を指定します。

`status`

処理結果の詳細が返されます。

`key`

読み込む指令の条件を次のどちらかの形式で指定します。

- ホスト名 [.配布先指令番号]
- %グループid [.配布先指令番号]

`getarea`

レコード入力領域を指定します。入力領域は、サイズに`DMGAI_EFILE_SIZE`を指定して確保してください。

JP1/NETM/DM Client から値が返される引数

`status`

処理結果の詳細が返されます。

リターン値

0 : 正常終了

-1 : 異常終了

_gaiCMDFindNext() — 実行管理ファイルの参照 (前回参照した次のレコードの参照)

形式

```
#include "/NETMRDSEX1/include/afapi.h"  
int _gaiCMDFindNext(int *cnt_inf, unsigned int *status, char *getarea);
```

機能

実行管理ファイルの前回参照したレコードの、次のレコードを入力領域に読み込みます。

この関数は、_gaiCMDRewind()または_gaiCMDFind()のあとに、繰り返し発行できます。

_gaiCMDFind()のあとに発行した場合は、そこで指定した条件に一致する次のレコードを読み込みます。なお、入力領域に読み込まれたレコードから特定の属性情報を取り出す場合は、_gaiCMDReferArea()を使用してください。

プログラムで値を設定する引数

cnt_inf

_gaiAPIStart()の第1引数に返された値を指定します。

getarea

レコード入力領域を指定します。入力領域は、サイズにDMGAI_EFILE_SIZEを指定して確保してください。

JP1/NETM/DM Client から値が返される引数

status

処理結果の詳細情報が返されます。

リターン値

0：正常終了

-1：異常終了

`_gaiCMDFindUnique()` — 実行管理ファイルの参照 (特定レコードの参照)

形式

```
#include "NETMRDSEX1/include/afapi.h"
int _gaiCMDFindUnique(int *cnt_inf, unsigned int *status,
                     char *key, char *getarea);
```

機能

指定された配布先名および配布先指令番号に一致する指令を、入力領域に読み込みます。入力領域に読み込まれたレコードから特定の情報を取り出す場合は、`_gaiCMDReferArea()`を使用してください。

プログラムで値を設定する引数

`cnt_inf`

`_gaiAPIStart()`の第1引数に返された値を指定します。

`key`

読み込む指令の条件を次のどちらかの形式で指定します。

- ホスト名.配布先指令番号
- %グループid.配布先指令番号

`getarea`

レコード入力領域を指定します。入力領域は、サイズに`DMGAI_EFILE_SIZE`を指定して確保してください。

JP1/NETM/DM Client から値が返される引数

`status`

処理結果の詳細情報が返されます。

リターン値

0：正常終了

-1：異常終了

`_gaiCMDReferArea()` — 実行管理ファイルの参照 (レコード中の特定エリアの参照)

形式

```
#include "NETMRDSEX1/include/afapi.h"
int _gaiCMDReferArea(char *getarea, unsigned int *status,
                    unsigned int code, {char | int} *area
                    [, unsigned int code, {char | int} *area] ..., 0);
```

機能

実行管理ファイルを参照する関数 (`_gaiCMDRewind()`, `_gaiCMDFind()`, `_gaiCMDFindNext()`, `_gaiCMDFindUnique()`) で入力領域に読み込んだレコードの、特定の属性情報を参照します。参照できる属性情報については、「付録 A.2 実行管理ファイル」を参照してください。

プログラムで値を設定する引数

getarea

`_gaiCMDRewind()`, `_gaiCMDFind()`, `_gaiCMDFindNext()`, または `_gaiCMDFindUnique()` を使用してレコードを読み込んだ領域を指定します。

code

参照したい属性情報の識別子を指定します。code は、area と組にして繰り返し指定できます。

area

参照するデータを格納する領域を指定します。領域は、データのサイズに 1 バイトを加えた大きさで準備してください。また、領域の型は、参照するデータの型 (char 型または int 型) へのポインタとしてください。

JP1/NETM/DM Client から値が返される引数

status

処理結果の詳細情報が返されます。

リターン値

0 : 正常終了

-1 : 異常終了

注意事項

引数の最後には必ず 0 を設定してください。

API のリターンコードを参照する関数

JPI/NETM/DM Client で使用できる配布管理の関数のうち、API のリターンコードを参照する関数には次のものがあります。

- 概要コードの参照：_gaiRTNGetMainCode()
- 詳細コードの参照：_gaiRTNGetSubCode()

API の実行結果は、概要コードおよび詳細コードとして詳細情報格納領域に設定されます。

概要コードでエラー種別を参照できます。_gaiRTNGetMainCode()を呼び出して参照してください。また、詳細コードを参照すると、エラー内容の詳細な情報を参照できます。_gaiRTNGetSubCode()を呼び出して参照してください。

概要コードおよび詳細コードは、使用する関数でその内容が異なります。それぞれのコードの意味は「付録 B リターンコード」で確認してください。

次に、API のリターンコードを参照する関数についての詳細を上記の順に説明します。

_gaiRTNGetMainCode() — 概要コードの参照

形式

```
#include "/NETMRDSEX1/include/afapi.h"  
unsigned int _gaiRTNGetMainCode(unsigned int *status);
```

機能

各関数の実行結果の概要コードを詳細情報格納領域から取り出し、リターン値として返します。さらに詳細な情報（詳細コード）を知りたい場合は、_gaiRTNGetSubCode()を使用してください。

プログラムで値を設定する引数

status

各関数で返された詳細情報を指定します。

リターン値

「付録 B.1 配布管理の関数のリターンコード」の概要コードを参照してください。

_gaiRTNGetSubCode() — 詳細コードの参照

形式

```
#include "NETMRDSEX1/include/afapi.h"  
unsigned int _gaiRTNGetSubCode(unsigned int *status);
```

機能

各関数の実行結果の詳細コードを詳細情報格納領域から取り出し、リターン値として返します。

プログラムで値を設定する引数

status

各関数で返された詳細情報を指定します。

リターン値

「付録 B.1 配布管理の関数のリターンコード」の詳細コードを参照してください。

3

配布状況ファイルアクセス関数

JP1/NETM/DM Client が提供する配布状況ファイルアクセス関数について説明します。

使用するファイル

JP1/NETM/DM Client の API に関連するファイルについて説明します。

■ライブラリ関数に必要なファイル

ライブラリ関数は、共用ライブラリで提供してあります。このファイル名およびヘッダファイル名を次に示します。

共用ライブラリ

/NETMRDSEX1/lib/dmwapi.sl ... HP-UX, Solaris の場合
/NETMRDSEX1/lib/dmwapi.o ... AIX の場合
/NETMRDSEX1/lib/dmwapi.so ... Tru64 UNIX の場合
/opt/NETMDMWEX1/lib/dmwapi.sl ... HP-UX(IPF)で 32 ビットのアプリケーションを作成する場合
/opt/NETMDMWEX1/lib/dmwapi64.sl ... HP-UX(IPF)で 64 ビットのアプリケーションを作成する場合

ヘッダファイル

/NETMRDSEX1/include/dmwapi.h

なお、OS が HP-UX の場合、ファイル名は/opt/NETMDMWEX1/~（以下同じ）となります。

■サンプルプログラムのファイル

JP1/NETM/DM Client の API を使ったサンプルプログラムは、次に示すファイル名で格納してあります。ライブラリ関数の標準的な使い方を知りたい場合に参照してください。

サンプルプログラム

/NETMRDSEX1/sample/dlvdspmp.c

なお、OS が HP-UX の場合、ファイル名は/opt/NETMDMWEX1/~（以下同じ）となります。

開始・終了の関数

`_gaiDLVStart()`は配布状況ファイルにアクセスする API の環境を準備する関数です。

`_gaiDLVEnd()`は配布状況ファイルにアクセスする API を終了する関数です。

API の開始・終了時には必ず呼び出します。

次に、開始・終了の関数の詳細を説明します。

_gaiDLVStart() — 配布状況ファイルへのアクセスの開始

形式

```
#include "/NETMRDSEX1/include/dmwapi.h"  
int _gaiDLVStart (int *cnt_inf, unsigned int *status) ;
```

OS が HP-UX (IPF) の場合の形式を次に示します。

```
#include "/NETMRDSEX1/include/dmwapi.h"  
int _gaiDLVStart (void **cnt_inf, unsigned int *status) ;
```

機能

配布状況ファイルにアクセスする API の開始処理をします。

プログラムで値を設定する引数

cnt_inf

API を使うための情報格納エリアのアドレスを設定します。

JP1/NETM/DM Client から値が返される引数

status

処理結果の詳細情報を格納するエリアを設定します。

リターン値

0 : 正常終了

-1 : 異常終了

注意事項

- `_gaiDLVStart()` を呼び出したあとに配布状況の先頭レコードを参照する関数 (`_gaiDLVRewind()`) を呼び出すと、配布状況を参照できます。
- 処理結果の詳細情報を格納するエリアを参照するときは、処理結果を参照する関数 (`_gaiDLVGetMainCode()`, `_gaiDLVGetSubCode()`) を使ってください。

_gaiDLVEnd() — 配布状況ファイルへのアクセスの終了

形式

```
#include "/NETMRDSEX1/include/dmwapi.h"  
int _gaiDLVEnd (int *cnt_inf, unsigned int *status) ;
```

OS が HP-UX (IPF) の場合の形式を次に示します。

```
#include "/NETMRDSEX1/include/dmwapi.h"  
int _gaiDLVEnd (void **cnt_inf, unsigned int *status) ;
```

機能

配布状況ファイルにアクセスする API の終了処理をします。

プログラムで値を設定する引数

cnt_inf

配布状況ファイルへのアクセスの開始関数 (_gaiDLVStart()) の第 1 引数を設定します。

JP1/NETM/DM Client から値が返される引数

status

処理結果の詳細情報を格納するエリアを設定します。

リターン値

0 : 正常終了

-1 : 異常終了

注意事項

- 配布状況ファイルを読み込み終わったら、_gaiDLVEnd() を必ず呼び出してください。
- 処理結果の詳細情報を格納するエリアを参照するときは、処理結果を参照する関数 (_gaiDLVGetMainCode(), _gaiDLVGetSubCode()) を使います。

レコードの参照の関数

JP1/NETM/DM Client で使用できる配布状況ファイルアクセス関数のうち、レコードの参照の関数には次のものがあります。

- 配布状況ファイルの先頭レコードの参照：_gaiDLVRewind()
- 配布状況ファイルの任意のレコードの参照：_gaiDLVFind()
- 配布状況ファイルの前回参照した次のレコードの参照：_gaiDLVFindNext()
- 配布状況ファイルの特定レコードの参照：gaiDLVFindUnique()
- 配布状況ファイルのレコード中の特定エリアの参照：_gaiDLVReferArea()

次に、各参照の関数についての詳細を上記の順に説明します。

`_gaiDLVRewind()` — 配布状況ファイルの先頭レコードの参照

形式

```
#include "/NETMRDSEX1/include/dmwapi.h"
int _gaiDLVRewind (int *cnt_inf, char *deliv_inf, unsigned int *status) ;
```

OS が HP-UX (IPF) の場合の形式を次に示します。

```
#include "/NETMRDSEX1/include/dmwapi.h"
int _gaiDLVRewind (void **cnt_inf, char *deliv_inf, unsigned int *status) ;
```

機能

配布状況ファイルから、先頭の配布状況レコードを読み込みます。

プログラムで値を設定する引数

`cnt_inf`

配布状況ファイルへのアクセスの開始関数 (`_gaiDLVStart()`) の第 1 引数を設定します。

`deliv_inf`

配布状況の受信領域のポインタを設定します。受信領域はサイズに変数名 `DMWGAI_DLV_SIZE` (ヘッダファイルで定義済み) を指定して確保します。

JP1/NETM/DM Client から値が返される引数

`status`

処理結果の詳細情報を格納するエリアを設定します。

リターン値

0 : 正常終了

1 : 配布状況サーバが起動していない

-1 : 異常終了

注意事項

- 配布状況ファイル内のレコードは、配布先-資源名-配布先指令番号の組み合わせで、昇順に並んでいます。
- `_gaiDLVRewind()`は、配布状況ファイルへのアクセスの開始関数 (`_gaiDLVStart()`) のあとで呼んでください。`_gaiDLVStart()`を呼ばないで`_gaiDLVRewind()`を呼ぶと、エラーになります。
- `_gaiDLVRewind()`では、先頭の 1 レコード分の配布状況を読み込みます。複数レコード分の配布状況を読み込む場合には、配布状況ファイルの次レコードを参照する関数 (`_gaiDLVFindNext()`) を繰り返し呼んでください。
- 配布状況の受信領域に格納される配布状況の任意エリアの参照方法は、配布状況エリアを参照する関数 (`_gaiDLVReferArea()`) を使います。

`_gaiDLVFind()` — 配布状況ファイルの任意のレコードの参照

形式

```
#include "/NETMRDSEX1/include/dmwapi.h"
int _gaiDLVFind (int *cnt_inf, unsigned int *status, char *key_kind,
char *key, char *deliv_inf) ;
```

OS が HP-UX (IPF) の場合の形式を次に示します。

```
#include "/NETMRDSEX1/include/dmwapi.h"
int _gaiDLVFind (void **cnt_inf, unsigned int *status, char *key_kind,
char *key, char *deliv_inf) ;
```

機能

配布状況ファイルから、指定した任意のキーに該当する先頭のレコードを読み込みます。

プログラムで値を設定する引数

`cnt_inf`

配布状況ファイルへのアクセスの開始関数 (`_gaiDLVStart()`) の第 1 引数を設定します。

`key_kind`

キーの種別を次の形式で指定します (ヘッダファイルで定義済み)。

`DMWGAI_DLV_KEY_KIND_ADR`

配布先名でサーチします。

`DMWGAI_DLV_KEY_KIND_RSC`

完全資源名でサーチします。

`DMWGAI_DLV_KEY_KIND_STS`

配布状況のステータスでサーチします。

`key`

入力する配布状況の配布名、パッケージ名、または配布状況のステータスを次の形式で指定します。

入力するキーの種別が `DMWGAI_DLV_KEY_KIND_ADR` の場合

ホスト名 [.ホスト名.ホスト名…ホスト名]

入力するキーの種別が `DMWGAI_DLV_KEY_KIND_RSC` の場合

完全資源名 (「DM 種別コード.グループ名.資源名.バージョンリビジョン.世代番号」の形式で指定)

入力するキーの種別が `DMWGAI_DLV_KEY_KIND_STS` の場合

- `DMWGAI_DLV_KEEPING`
指令ファイル受信の完了 (KEEPING)
- `DMWGAI_DLV_KEEP`
指令保管の完了 (KEEP)
- `DMWGAI_DLV_KEEP_ERROR`
パッケージ保管の失敗 (KEEP_ERROR)
- `DMWGAI_DLV_CLT_START`

- クライアント起動の完了 (CLT_START)
- DMWGAI_DLV_CLT_START_ERROR
クライアント起動の失敗 (CLT_START_ERROR)
- DMWGAI_DLV_CLT_WAIT
クライアント起動待ち (CLT_WAIT)
- DMWGAI_DLV_INS_WAIT
組み込み待ち (INS_WAIT)
- DMWGAI_DLV_GEN_START
ゼネレータの起動 (GEN_START)
- DMWGAI_DLV_COMPLETE
実行完了 (COMPLETE)
- DMWGAI_DLV_ERROR
実行失敗 (ERROR)
- DMWGAI_DLV_SETUP_ERROR
セットアップエラー (SETUP_ERROR)

deliv_inf

配布状況の受信領域のポインタを設定します。受信領域はサイズに変数名 `DMWGAI_DLV_SIZE` (ヘッダファイルで定義済み) を指定して確保します。

JP1/NETM/DM Client から値が返される引数

status

処理結果の詳細情報を格納するエリアを設定します。

リターン値

0：正常終了

1：配布状況サーバが起動していない

-1：異常終了

注意事項

- 配布状況ファイル内のレコードは、配布先-資源名-配布先指令番号の組み合わせで、昇順に並んでいます。
- `_gaiDLVFind()` は、配布状況ファイルへのアクセスの開始関数 (`_gaiDLVStart()`) のあとで呼んでください。`_gaiDLVStart()` を呼ばないで `_gaiDLVFind()` を呼ぶと、エラーになります。
- `_gaiDLVFind()` では、入力した key に該当する先頭の 1 レコード分の配布状況を読み込みます。入力した key に該当する複数レコード分の配布状況を読み込む場合には、配布状況ファイルの次レコードを参照する関数 (`_gaiDLVFindNext()`) を繰り返し呼んでください。
- 配布状況の受信領域に格納される配布状況の任意エリアの参照方法は、配布状況エリアを参照する関数 (`_gaiDLVReferArea()`) を使います。
- 処理結果の詳細情報格納エリアを参照するときは、処理結果を参照する関数 (`_gaiDLVGetMainCode()`, `_gaiDLVGetSubCode()`) を使います。

`_gaiDLVFindNext()` — 配布状況ファイルの前回参照した次のレコードの参照

形式

```
#include "/NETMRDSEX1/include/dmwapi.h"
int _gaiDLVFindNext (int *cnt_inf, char *deliv_inf, unsigned int *status) ;
```

OS が HP-UX (IPF) の場合の形式を次に示します。

```
#include "/NETMRDSEX1/include/dmwapi.h"
int _gaiDLVFindNext (void **cnt_inf, char *deliv_inf, unsigned int *status) ;
```

機能

配布状況ファイルから、前回読み込んだレコードの次のレコードを読み込みます。

プログラムで値を設定する引数

`cnt_inf`

配布状況ファイルへのアクセスの開始関数 (`_gaiDLVStart()`) の第 1 引数を設定します。

`deliv_inf`

配布状況の受信領域のポインタを設定します。受信領域はサイズに変数名 `DMWGAI_DLV_SIZE` (ヘッダファイルで定義済み) を指定して確保します。

JP1/NETM/DM Client から値が返される引数

`status`

処理結果の詳細情報を格納するエリアを設定します。

リターン値

0 : 正常終了

-1 : 異常終了

注意事項

- 配布状況ファイル内のレコードは、配布先-資源名-配布先指令番号の組み合わせで、昇順に並んでいます。
- `_gaiDLVFindNext()` は、次に示す関数を呼んだあとで呼んでください。これらの関数を呼ばないで `_gaiDLVFindNext()` を呼んだ場合はエラーになります。
 - `_gaiDLVRewind()` (配布状況ファイルの先頭レコードを参照する関数)
 - `_gaiDLVFind()` (配布状況ファイルの任意のレコードを参照する関数)
- 複数レコード分の配布状況を読み込む場合には、`_gaiDLVFindNext()` を繰り返し呼んでください。
- 配布状況の受信領域に格納される配布状況の任意エリアの参照方法は、配布状況エリアを参照する関数 (`_gaiDLVReferArea()`) を使います。
- 処理結果の詳細情報格納エリアを参照するときは、処理結果を参照する関数 (`_gaiDLVGetMainCode()`, `_gaiDLVGetSubCode()`) を使います。

`_gaiDLVFindUnique()` — 配布状況ファイルの特定レコードの参照

形式

```
#include "/NETMRDSEX1/include/dmwapi.h"
int _gaiDLVFindUnique (int *cnt_inf, unsigned int *status, char *key, char *deliv_inf) ;
```

OS が HP-UX (IPF) の場合の形式を次に示します。

```
#include "/NETMRDSEX1/include/dmwapi.h"
int _gaiDLVFindUnique (void **cnt_inf, unsigned int *status, char *key, char *deliv_inf) ;
```

機能

配布状況ファイルから、指定した特定のキーに該当するレコードを読み込みます。

プログラムで値を設定する引数

`cnt_inf`

配布状況ファイルへのアクセスの開始関数 (`_gaiDLVStart()`) の第 1 引数を設定します。

`key`

入力する配布状況のキーを次の形式で指定します。

ホスト名 [.ホスト名...ホスト名] .完全資源名. 配布先指令番号

`deliv_inf`

配布状況の受信領域のポインタを設定します。受信領域はサイズに変数名 `DMWGAI_DLV_SIZE` (ヘッダファイルで定義済み) を指定して確保します。

JP1/NETM/DM Client から値が返される引数

`status`

処理結果の詳細情報を格納するエリアを設定します。

リターン値

0 : 正常終了

1 : 配布状況サーバが起動していない

-1 : 異常終了

注意事項

- 配布状況ファイル内のレコードは、配布先-資源名-配布先指令番号の組み合わせで、昇順に並んでいます。
- 配布状況の受信領域に格納される配布状況の任意エリアの参照方法は、配布状況エリアを参照する関数 (`_gaiDLVReferArea()`) を使います。
- 処理結果の詳細情報格納エリアを参照するときは、処理結果を参照する関数 (`_gaiDLVGetMainCode()`, `_gaiDLVGetSubCode()`) を使います。

`_gaiDLVReferArea()` — 配布状況ファイルのレコード中の特定エリアの参照

形式

```
#include "NETMRDSEX1/include/dmwapi.h"
int _gaiDLVReferArea (char *deliv_inf, unsigned int *status,
unsigned int code1, char *area1, ...
unsigned int coden, char *arean, 0) ;
```

機能

配布状況ファイルから、先頭の配布状況レコードを読み込みます。

プログラムで値を設定する引数

`deliv_inf`

配布状況の受信領域のポインタを設定します。受信領域はサイズに変数名 `DMWGAI_DLV_SIZE` (ヘッダファイルで定義済み) を指定して確保します。

`code`

参照するエリアの識別子を設定します。設定できる識別子については、リターン値を参照してください。`code` は、`area` と組にして複数設定できます。

`area`

参照エリアの内容を格納する領域のポインタを設定します。領域のサイズは、(内容が格納できる大きさ+1バイト) 分必要です。`area` は、`code` と組にして複数設定できます。

JP1/NETM/DM Client から値が返される引数

`status`

処理結果の詳細情報を格納するエリアを設定します。

リターン値

0: 正常終了

-1: 異常終了

参照エリアと識別子については、「付録 A.3 配布状況ファイル」を参照してください。

注意事項

- 引数の最後には必ず 0 を設定してください。
- 配布状況ファイル内のレコードは、配布先-資源名-配布先指令番号の組み合わせで、昇順に並んでいます。
- 処理結果の詳細情報格納エリアを参照するときは、処理結果を参照する関数 (`_gaiDLVGetMainCode()`, `_gaiDLVGetSubCode()`) を使います。

レコードの削除の関数

JP1/NETM/DM Client で使用できる配布状況ファイルアクセス関数のうち、レコードの削除の関数には次のものがあります。

- 配布状況ファイルの任意のレコードの削除：_gaiDLVDel()
- 配布状況ファイルの特定のレコードの削除：_gaiDLVDelUnique()

次に、各削除の関数についての詳細を上記の順に説明します。

`_gaiDLVDel()` — 配布状況ファイルの任意のレコードの削除

形式

```
#include "/NETMRDSEX1/include/dmwapi.h"
int _gaiDLVDel (int *cnt_inf, unsigned int *status, char *key_kind, char *key) ;
```

OS が HP-UX (IPF) の場合の形式を次に示します。

```
#include "/NETMRDSEX1/include/dmwapi.h"
int _gaiDLVDel (void **cnt_inf, unsigned int *status, char *key_kind, char *key) ;
```

機能

配布状況ファイルから、指定した任意のキーに該当するレコードを削除します。

プログラムで値を設定する引数

`cnt_inf`

配布状況ファイルへのアクセスの開始関数 (`_gaiDLVStart()`) の第 1 引数を設定します。

`key_kind`

削除するキーの種別を次の形式で指定します (ヘッダファイルで定義済み)。

`DMWGAI_DLV_KEY_KIND_ADR`

配布先名をキーにして削除します。

`DMWGAI_DLV_KEY_KIND_RSC`

完全資源名をキーにして削除します。

`DMWGAI_DLV_KEY_KIND_STS`

配布状況のステータスで削除します。

`key`

入力する配布状況の配布名、パッケージ名、または配布状況のステータスを次の形式で指定します。

入力するキーの種別が `DMWGAI_DLV_KEY_KIND_ADR` の場合

ホスト名 [.ホスト名.ホスト名. …ホスト名]

入力するキーの種別が `DMWGAI_DLV_KEY_KIND_RSC` の場合

完全資源名

入力するキーの種別が `DMWGAI_DLV_KEY_KIND_STS` の場合

- `DMWGAI_DLV_KEEPING`
指令ファイル受信の完了 (`KEEPING`)
- `DMWGAI_DLV_KEEP`
指令保管の完了 (`KEEP`)
- `DMWGAI_DLV_KEEP_ERROR`
パッケージ保管の失敗 (`KEEP_ERROR`)
- `DMWGAI_DLV_CLT_START`
クライアント起動の完了 (`CLT_START`)
- `DMWGAI_DLV_CLT_START_ERROR`

クライアント起動の失敗 (CLT_START_ERROR)

- DMWGAI_DLV_CLT_WAIT
クライアント起動待ち (CLT_WAIT)
- DMWGAI_DLV_INS_WAIT
組み込み待ち (INS_WAIT)
- DMWGAI_DLV_GEN_START
ゼネレータの起動 (GEN_START)
- DMWGAI_DLV_COMPLETE
実行完了 (COMPLETE)
- DMWGAI_DLV_ERROR
実行失敗 (ERROR)
- DMWGAI_DLV_SETUP_ERROR
セットアップエラー (SETUP_ERROR)

deliv_inf

配布状況の受信領域のポインタを設定します。受信領域はサイズに変数名 `DMWGAI_DLV_SIZE` (ヘッダファイルで定義済み) を指定して確保します。

JP1/NETM/DM Client から値が返される引数

status

処理結果の詳細情報を格納するエリアを設定します。

リターン値

0: 正常終了

1: 配布状況サーバが起動していない

-1: 異常終了

注意事項

- 配布状況ファイル内のレコードは、配布先-資源名-配布先指令番号の組み合わせで、昇順に並んでいます。
- 処理結果の詳細情報格納エリアを参照するときは、処理結果を参照する関数 (`_gaiDLVGetMainCode()`, `_gaiDLVGetSubCode()`) を使います。

`_gaiDLVDelUnique()` — 配布状況ファイルの特定のレコードの削除

形式

```
#include "/NETMRDSEX1/include/dmwapi.h"
int _gaiDLVDelUnique() (int *cnt_inf, unsigned int *status, char *key) ;
```

OS が HP-UX (IPF) の場合の形式を次に示します。

```
#include "/NETMRDSEX1/include/dmwapi.h"
int _gaiDLVDelUnique() (void **cnt_inf, unsigned int *status, char *key) ;
```

機能

配布状況ファイルから、指定した特定のキーに該当するレコードを削除します。

プログラムで値を設定する引数

`cnt_inf`

配布状況ファイルへのアクセスの開始関数 (`_gaiDLVStart()`) の第 1 引数を設定します。

`key`

入力する配布状況のキーを次の形式で指定します。

ホスト名 [.ホスト名. …ホスト名] .完全資源名. 配布先指令番号

完全資源名は、「DM 種別コード.グループ名.資源名.バージョンリビジョン.世代番号」の形式で指定します。

JP1/NETM/DM Client から値が返される引数

`status`

処理結果の詳細情報を格納するエリアを設定します。

リターン値

0 : 正常終了

1 : 配布状況サーバが起動していない

-1 : 異常終了

注意事項

- 配布状況ファイル内のレコードは、配布先-資源名-配布先指令番号の組み合わせで、昇順に並んでいます。
- 処理結果の詳細情報格納エリアを参照するときは、処理結果を参照する関数 (`_gaiDLVGetMainCode()`, `_gaiDLVGetSubCode()`) を使います。

API のリターンコードを参照する関数

JPI/NETM/DM Client で使用できる配布状況ファイルアクセス関数のうち、API のリターンコードを参照する関数には次のものがあります。

- 概要コードの参照：_gaiDLVGetMainCode()
- 詳細コードの参照：_gaiDLVGetSubCode()

API の実行結果は、概要コードおよび詳細コードとして詳細情報格納領域に設定されます。

概要コードでエラー種別を参照できます。_gaiDLVGetMainCode() を呼び出して参照してください。また、詳細コードを参照すると、エラー内容の詳細な情報を参照できます。_gaiDLVGetSubCode() を呼び出して参照してください。

概要コードおよび詳細コードは、使用する関数でその内容が異なります。それぞれのコードの意味は「付録 B リターンコード」で確認してください。

次に、API のリターンコードを参照する関数についての詳細を上記の順に説明します。

_gaiDLVGetMainCode() — 概要コードの参照

形式

```
#include "NETMRDSEX1/include/dmwapi.h"  
int _gaiDLVGetMainCode (unsigned int *status) ;
```

機能

各関数の詳細情報の格納エリアから、概要コードを参照します。

プログラムで値を設定する引数

status

各関数の処理結果の詳細情報を格納している領域を設定します。

リターン値

「付録 B.2 配布状況ファイルアクセス関数のリターンコード」の概要コードを参照してください。

_gaiDLVGetSubCode() — 詳細コードの参照

形式

```
#include "NETMRDSEX1/include/dmwapi.h"  
int _gaiDLVGetSubCode (unsigned int *status) ;
```

機能

各関数の詳細情報の格納エリアから、詳細コードを参照します。

プログラムで値を設定する引数

status

各関数の処理結果の詳細情報を格納している領域を設定します。

リターン値

「付録 B.2 配布状況ファイルアクセス関数のリターンコード」の詳細コードを参照してください。

付録

付録 A 管理ファイルの参照エリアと識別子

APIで参照できる管理ファイルには、次の三つがあります。

- 資源属性ファイル
- 実行管理ファイル
- 配布状況ファイル

APIのエリア参照関数を使用して参照できる管理ファイルのエリアとエリアの識別子を次に示します。

付録 A.1 資源属性ファイル

資源属性ファイルは、各資源グループごとの情報が格納されているヘッダ部（1レコード）と、登録ソフトウェアの情報が格納されているエントリ部（複数レコード）から構成されます。ヘッダ部の参照エリアと識別子を表 A-1 に、エントリ部の参照エリアと識別子を表 A-2 に示します。

表 A-1 資源属性ファイルの参照エリアと識別子（ヘッダ部）

参照できるエリア名	型	サイズ (バイト)	設定する識別子	エリアの説明
DM 種別コード	C	1	DMGAI_RSCH_DMKIND	ソフトウェアの DM 種別コードを示す C: UNIX D: Windows
グループ名	C	2	DMGAI_RSCH_GROUP	ソフトウェアのグループ名
グループ別資源数	X	4	DMGAI_RSCH_RSCNO	グループ内の登録ソフトウェア数

(凡例)

C: char 型

X: signed int 型

表 A-2 資源属性ファイルの参照エリアと識別子（エントリ部）

参照できるエリア名	型	サイズ (バイト)	設定する識別子	エリアの説明
DM 種別コード	C	1	DMGAI_RSCE_DMKIND	パッケージの DM 種別コードを示す C: UNIX D: Windows
グループ名	C	2	DMGAI_RSCE_GROUP	パッケージのグループ名
資源名	C	44	DMGAI_RSCE_RSCNM	資源名
バージョン/リビジョン	C	8	DMGAI_RSCE_VERREV	パッケージのバージョン/リビジョン
世代番号	C	4	DMGAI_RSCE_GENNO	パッケージの世代番号
登録フラグ	C	1	DMGAI_RSCE_CHKFLG	パッケージの登録状態を示す

参照できるエリア名	型	サイズ (バイト)	設定する識別子	エリアの説明
登録フラグ	C	1	DMGAI_RSCE_CHKFLG	Y: パッケージング完了 N: パッケージング未完了
仮想ファイル名	C	8	DMGAI_RSCE_VERNM	パッケージの仮想ファイル名
プログラム識別子	C	1	DMGAI_RSCE_PRGTYP	プログラムの種別を示す I: 他社ソフトウェア P: プログラムプロダクト 空白: ユーザプログラム, ユーザデータ*1
プログラム名	C	50	DMGAI_RSCE_PRGNM	プログラム名
容量 1	X	4	DMGAI_RSCE_RSCSIZE	パッケージの容量 (単位: バイト)
登録日	C	6	DMGAI_RSCE_ENTDAY	パッケージの登録日を <i>yymmdd</i> で示す <i>yy</i> : 西暦年下 2 けた, <i>mm</i> : 月, <i>dd</i> : 日
所有者名	C	14	DMGAI_RSCE_OWNER	パッケージの登録者, 作成者
組み込みタイミング	C	1	DMGAI_RSCE_INSTIM	パッケージの組み込みタイミングを示す B: システム起動時 E: システム稼働中 S: システム停止時 U: UAP による組み込み
稼働中インストールコード	C	1	DMGAI_RSCE_INSE	システム稼働中にインストールできるかどうかを示す <ul style="list-style-type: none">パッケージがプログラムプロダクトの場合 Y: インストールできる N: インストールできないパッケージがプログラムプロダクト以外の場合は空白
起動時インストールコード	C	1	DMGAI_RSCE_INSB	システム起動時にインストールできるかどうかを示す <ul style="list-style-type: none">パッケージがプログラムプロダクトの場合 Y: インストールできる N: インストールできないパッケージがプログラムプロダクト以外の場合は空白
ユーザ組み込み処理監視時間	X	4	DMGAI_RSCE_APWATCH	ユーザ組み込み処理, または PP インストール後処理の監視時間

参照できるエリア名	型	サイズ (バイト)	設定する識別子	エリアの説明
パッケージの満了日付	C	6	DMGAI_RSCE_KEEPPDATE	保管庫での保管期間を <i>yyymmdd</i> で示す <i>yy</i> : 西暦年下 2 けた, <i>mm</i> : 月, <i>dd</i> : 日
組み込み先ディレクトリ	C	64	DMGAI_RSCE_INSDIR	組み込み先のディレクトリ名
ディレクトリの所有者	X	4	DMGAI_RSCE_INSOWN	組み込み先のディレクトリの所有者 ((0~59999))
ディレクトリのファイルモード	X	4	DMGAI_RSCE_INSMODE	組み込み先のディレクトリのファイルモード ((700~777))
ディレクトリグループ	X	4	DMGAI_RSCE_INSGROUP	組み込み先のディレクトリグループ ((0~59999))
バックアップの保存期間	X	4	DMGAI_RSCE_KEEPPBACKUP	バックアップデータの保存期間 (日数)
ユーザコメント※2	C	64	DMGAI_RSCE_USRCOMMENT	ユーザコメント, またはユーザ固有の処理の指定を示す
インストール日時	C	12	DMGAI_RSCE_INSDAY	組み込み日時を <i>yyymmddhhmmss</i> で示す <i>yy</i> : 西暦年下 2 けた, <i>mm</i> : 月, <i>dd</i> : 日, <i>hh</i> : 時, <i>mm</i> : 分, <i>ss</i> : 秒
パッケージの形式	C	1	DMGAI_RSCE_RSCFORM	パッケージの形式を示す H : 圧縮暗号化してパッケージングしたパッケージ K : UNIX/Windows 共用資源 空白 : 圧縮暗号化しないでパッケージングしたパッケージ その他 : 上記以外のパッケージ
組み込み前提バージョン	C	6	DMGAI_RSCE_PREVER	組み込みの前提バージョン
インストールスクリプトファイルの有無	C	1	DMGAI_RSCE_SCRIPTF	インストールスクリプトファイルの有無を示す Y : 有り N : 無し
ゼネレータ区分	C	1	DMGAI_RSCE_GENCLS	ゼネレータ区分を示す C : 標準ゼネレータ Y : 組み込み支援機能による組み込み
リンケージファイル名	C	40	DMGAI_RSCE_LINKNAME	リンケージファイル名

参照できるエリア名	型	サイズ (バイト)	設定する識別子	エリアの説明
組み込み後起動 UAP 名	C	40	DMGAI_RSCE_UAPNAME	組み込み後起動 UAP 名
組み込みファイル名	C	14	DMGAI_RSCE_INSFIL	組み込みファイル名

(凡例)

C : char 型

X : signed int 型

注※1

Windows 用の差分パッケージを含みます。

注※2

ユーザコメント、ユーザ組み込み処理名、または PP インストール後処理名が格納されます。ユーザ組み込み前処理とユーザ組み込み後処理を両方使用した場合、ユーザ組み込み処理のエリアには、次の順序でユーザ組み込み処理の名称が格納されます。

ユーザ組み込み後処理名	NULL	ユーザ組み込み前処理名
-------------	------	-------------

付録 A.2 実行管理ファイル

実行管理ファイルを参照すると、指令に関する情報が取得できます。取得できる情報は、配布・組み込み指令だけです。

参照できる実行管理ファイルのエリアとその識別子を表 A-3 に示します。

表 A-3 配布指令の参照エリアと識別子

参照できるエリア名	型	サイズ (バイト)	設定する識別子	エリアの説明
指令種別	C	1	DMGAI_CMDE_CMDKIND	指令の種別を示す D : 配布・組み込み指令
組み込み種別	C	1	DMGAI_CMDE_TIMING	組み込み種別を示す F : 強制的に組み込む 空白 : 同一パッケージがある場合は組み込まない
配布先名	C	64*1	DMGAI_CMDE_DLVNM	パッケージの配布先名
グループ id 名	C	32	DMGAI_CMDE_DLVID	パッケージの配布先グループ id
配布先指令番号	C	16	DMGAI_CMDE_CMDNO	指令番号
ステータス	UC	7	DMGAI_CMDE_STATUS	指令のステータスを次の順で示す 1 バイト目 : ホストステータス 2 バイト目 : キャンセルステータス 3 バイト目 : エンドステータス 1 4 バイト目 : ユーザステータス 5 バイト目 : エンドステータス 2

参照できるエリア名	型	サイズ (バイト)	設定する識別子	エリアの説明
ステータス	UC	7	DMGAI_CMDE_STATUS	6バイト目：インタフェースプログラムリターンコード 7バイト目：リカバリステータス
DM 種別コード	C	1	DMGAI_CMDE_DMKIND	パッケージのDM 種別コードを示す C：UNIX D：Windows
グループ名	C	2	DMGAI_CMDE_GROUP	パッケージのグループ名
資源名	C	44	DMGAI_CMDE_RSCNM	資源名
バージョン/リビジョン	C	8	DMGAI_CMDE_VERREV	パッケージのバージョン/リビジョン
世代番号	C	4	DMGAI_CMDE_GENNO	パッケージの世代番号
プログラム名称	C	50	DMGAI_CMDE_PRGNM	パッケージのプログラム名称
仮想ファイル名	C	8	DMGAI_CMDE_VERNM	パッケージの仮想ファイル名
プログラム識別子	C	1	DMGAI_CMDE_PRGTYP	プログラムの種別を示す I：他社ソフトウェア P：プログラムプロダクト 空白：ユーザプログラム，ユーザデータ※2
容量 1	X	4	DMGAI_CMDE_RSCSIZE	パッケージの容量 (単位：バイト)
登録日	C	6	DMGAI_CMDE_ENTDAY	パッケージの登録日を <i>yyymmdd</i> で示す <i>yy</i> ：西暦年下 2 けた， <i>mm</i> ：月， <i>dd</i> ：日
所有者名	C	14	DMGAI_CMDE_OWNER	パッケージの登録者，作成者
組み込みタイミング	C	1	DMGAI_CMDE_INSTIM	パッケージの組み込みタイミングを示す B：システム起動時 E：システム稼働中 S：システム停止時 U：UAP による組み込み
稼働中インストールコード	C	1	DMGAI_CMDE_INSE	システム稼働中にインストールできるかどうかを示す <ul style="list-style-type: none"> • パッケージがプログラムプロダクトの場合 Y：インストールできる N：インストールできない • パッケージがプログラムプロダクト以外の場合は空白

参照できるエリア名	型	サイズ (バイト)	設定する識別子	エリアの説明
起動時インストールコード	C	1	DMGAI_CMDE_INSB	システム起動時にインストールできるかどうかを示す <ul style="list-style-type: none"> パッケージがプログラムプロダクトの場合 Y: インストールできる N: インストールできない パッケージがプログラムプロダクト以外の場合は空白
ユーザ組み込み処理監視時間	X	4	DMGAI_CMDE_APWATCH	ユーザ組み込み処理, または PP インストール後処理の監視時間
パッケージの満了日付	C	6	DMGAI_CMDE_KEEPPDATE	保管庫での保管期間を <i>yymmdd</i> で示す <i>yy</i> : 西暦年下 2 けた, <i>mm</i> : 月, <i>dd</i> : 日
組み込み先ディレクトリ	C	64	DMGAI_CMDE_INSDIR	組み込み先のディレクトリ名
ディレクトリの所有者	X	4	DMGAI_CMDE_INSOWN	組み込み先のディレクトリの所有者 ((0~59999))
ディレクトリのファイルモード	X	4	DMGAI_CMDE_INSMODE	組み込み先のディレクトリモード ((700~777))
ディレクトリのグループ	X	4	DMGAI_CMDE_INSGROUP	組み込み先のディレクトリグループ ((0~59999))
バックアップデータ保存期間	X	4	DMGAI_CMDE_KEEPPBACKUP	バックアップデータの保存期間 (日数)
ユーザコメント※ ³	C	64	DMGAI_CMDE_USRCOMMENT	ユーザコメント, またはユーザ固有の処理の指定を示す
インストール日時	C	12	DMGAI_CMDE_INSDAY	組み込み日時を <i>yymmddhhmmss</i> で示す <i>yy</i> : 西暦年下 2 けた, <i>mm</i> : 月, <i>dd</i> : 日, <i>hh</i> : 時, <i>mm</i> : 分, <i>ss</i> : 秒
パッケージの形式	C	1	DMGAI_CMDE_RSCFORM	パッケージの形式を示す H: 圧縮暗号化してパッケージングしたパッケージ K: UNIX/Windows 共用資源 空白: 圧縮暗号化しないでパッケージングしたパッケージ その他: 上記以外のパッケージ
組み込み前提バージョン	C	6	DMGAI_CMDE_PREVER	組み込みの前提バージョン

参照できるエリア名	型	サイズ (バイト)	設定する識別子	エリアの説明
インストールスクリプトファイル有無	C	1	DMGAI_CMDE_SCRIPTF	インストールスクリプトファイルの有無を示す Y: 有り N: 無し
ゼネレータ区分	C	1	DMGAI_CMDE_GENCLS	ゼネレータ区分を示す C: 標準ゼネレータ Y: 組み込み支援機能による組み込み
リンクージファイル名	C	40	DMGAI_CMDE_LINKNAME	リンクージファイル名
組み込み後起動 UAP 名	C	40	DMGAI_CMDE_UAPNAME	組み込み後起動 UAP 名
組み込みファイル名	C	14	DMGAI_CMDE_INSFIL	組み込みファイル名
資源サイズ	X	4	DMGAI_CMDE_DATASIZE	パッケージの配布状況を表示するときに転送 (配布) するパッケージの大きさ
全転送サイズ	X	4	DMGAI_CMDE_TTRANSIZE	パッケージの配布状況を表示するときに転送できた大きさ
分割サイズ	X	4	DMGAI_CMDE_SPLITSIZE	分割配布のときに指定した分割サイズ
転送サイズ	X	4	DMGAI_CMDE_TRANSIZE	分割配布のときに転送できた大きさ
分割回数	X	2	DMGAI_CMDE_SPLITNO	分割配布の回数

(凡例)

C: char 型

UC: unsigned char 型

X: signed int 型

注※1

配布先名が 64 バイトを超える場合は、255 バイトになります。

注※2

Windows 用の差分パッケージを含みます。

注※3

ユーザコメント、ユーザ組み込み処理名、または PP インストール後処理名が格納されます。ユーザ組み込み前処理とユーザ組み込み後処理を両方使用した場合、ユーザ組み込み処理のエリアには、次の順序でユーザ組み込み処理の名称が格納されます。

ユーザ組み込み後処理名	NULL	ユーザ組み込み前処理名
-------------	------	-------------

付録 A.3 配布状況ファイル

配布状況ファイルのレコードエリアを参照するときに使う_gaiDLVReferArea()で参照できる参照エリアと識別子の一覧を表 A-4 に示します。

表 A-4 参照エリアと識別子の一覧

参照できるエリア名	型	サイズ	識別子	エリアの内容説明
指令種別	C	1	DMWGAI_DLV_CMDKIND	'D' : 配布・組み込み指令
組み込み種別	C	1	DMWGAI_DLV_TIMING	'△' : 通常の組み込み 'F' : 強制組み込み 'U' : UAP の組み込み
配布先ホスト名	C	64*1	DMWGAI_DLV_DLVNM	パッケージの配布先ホスト名
配布先指令番号	X	8	DMWGAI_DLV_CMDNO	指令に付加される番号
エンドステータス	C	1	DMWGAI_DLV_ENDSTS	JP1/NETM/DM Client (基本機能) の情報
ユーザステータス	C	1	DMWGAI_DLV_USRSTS	ユーザステータス
エンドステータス 2	C	1	DMWGAI_DLV_ENDSTS2	JP1/NETM/DM Client (基本機能) の保守情報
ユーザ組み込み処理または PP インストール後処理終了コード	C	1	DMWGAI_DLV_UPRGSTS	ユーザ組み込み処理または PP インストール後処理終了コード
リカバリステータス	C	1	DMWGAI_DLV_RCVSTS	リカバリステータス
配布状況ステータス	C	1	DMWGAI_DLV_DLVSTS	10 : 指令ファイル受信の完了 (KEEPING) 20 : 指令保管の完了 (KEEP) 2f : パッケージ保管の失敗 (KEEP_ERROR) 30 : クライアント起動の完了 (CLT_START) 3f : クライアント起動の失敗 (CLT_START_ERROR) 90 : クライアント起動待ち (CLT_WAIT) 40 : 組み込み待ち (INS_WAIT) 50 : ゼネレータの起動 (GEN_START) 60 : 実行完了 (COMPLETE) 70 : 実行失敗 (ERROR) 80 : セットアップエラー (SETUP_ERROR)
階層レベル	C	1	DMWGAI_DLV_DWNLVL	階層レベル
DM 種別	C	1	DMWGAI_DLV_DMKIND	資源登録システムの DM 種別 'C' : UNIX 'D' : Windows
グループ	C	2	DMWGAI_DLV_GROUP	パッケージのグループ
資源名	C	44	DMWGAI_DLV_RSCNM	資源名

参照できるエリア名	型	サイズ	識別子	エリアの内容説明
バージョン/リビジョン	C	8	DMWGAI_DLV_VERREV	パッケージのバージョン/リビジョン
世代番号	C	4	DMWGAI_DLV_GENNO	パッケージの世代番号
配布日	C	6	DMWGAI_DLV_DLVDAY	パッケージの配布日 yy:年(西暦下2けた) mm:月 dd:日
適用日	C	12	DMWGAI_DLV_WORKDAY	パッケージの組み込み日時 yy:年(西暦下2けた) mm:月 dd:日 hh:時 mm:分 ss:秒
リカバリ処理	C	1	DMWGAI_DLV_RECOVERY	インストール障害時のリカバリ処理 'Y':あり 'N':なし
プログラム名	C	50	DMWGAI_DLV_PROGNM	登録するソフトウェアに付ける一般名称
容量	X	4	DMWGAI_DLV_RSCSIZE	パッケージの容量
ゼネレータ区分	C	1	DMWGAI_DLV_GENCLS	ゼネレータの種類 'U':標準ゼネレータ 'Y':組み込み支援機能のゼネレータ
ユーザコメント	C	64	DMWGAI_DLV_USRCMT	ユーザコメント
ユーザ組み込み処理または PP インストール後処理 ^{※2}	C	64 ^{※3}	DMWGAI_DLV_USRPROG	ユーザ組み込み処理または PP インストール後処理
所有者	C	14	DMWGAI_DLV_OWNER	パッケージの登録者, 開発者
インストールタイミング	C	1	DMWGAI_DLV_INSTIM	インストールのタイミング 'B':システム起動時の組み込み 'E':システム稼働時の組み込み 'S':システム停止時の組み込み 'U':UAPの組み込み
ユーザ後処理監視時間	X	2	DMWGAI_DLV_APWATCH	ユーザ後処理プログラムの監視時間
満了日付	C	6	DMWGAI_DLV_KEEPDATE	保管庫に登録したパッケージの保管期限 yy:年(西暦下2けた) mm:月 dd:日
配布先ディレクトリ	C	64	DMWGAI_DLV_INSDIR	パッケージの配布先のディレクトリ
配布元ホスト名	C	64 ^{※1}	DMWGAI_DLV_HOSTNM	配布管理システムのホスト名

参照できるエリア名	型	サイズ	識別子	エリアの内容説明
前提バージョン	C	6	DMWGAI_DLV_PREVER	組み込むパッケージの前提バージョン

(凡例)

C : char 型

X : signed int 型

注※1

ホスト名が 64 バイトを超える場合、このエリアのサイズは 255 バイトになります。

注※2

ユーザ組み込み処理は、ゼネレータ区分が標準ゼネレータのときだけ参照できます。

注※3

ユーザ組み込み前処理とユーザ組み込み後処理を両方使用した場合、ユーザ組み込み処理のエリアには、次の順序でユーザ組み込み処理の名称が格納されます。

ユーザ組み込み後処理名	NULL	ユーザ組み込み前処理名
-------------	------	-------------

付録 B リターンコード

付録 B.1 配布管理の関数のリターンコード

詳細情報エリアに設定されている, API の各関数の処理結果を示す概要コードおよび詳細コードを関数の種類ごとに表 B-1~表 B-3 に示します。

表 B-1 概要コードと詳細コード (開始・終了関数)

概要コード	意味	詳細コード	意味
DMGAICOM	通信エラー	DMGAICOM_COMMUNICATE	ソケット通信エラー
DMGAIGETSD	ソケット記述子の獲得失敗	DMGAIGETSD_CONNECTION	JP1/NETM/DM Client との接続の確立に失敗した, または JP1/NETM/DM Client が起動されていない
		DMGAIGETSD_GETHOSTNM	ホスト名の取り出しに失敗した
		DMGAIGETSD_OPENSOCKET	ソケットのオープンに失敗した
		DMGAIGETSD_SERVNM	サービス名が誤っている
DMGAINVALID	引数不正	引数の指定順番の番号	引数の内容が不正である
DMGAINRM	正常終了	DMGAINRM_NORMAL	正常終了した
DMGAIREFUSE	要求が拒否された	DMGAIREFUSE_REQREFUSE	JP1/NETM/DM Client が計画停止中で, 要求が拒否された
DMGAISEQ	要求順序不正	DMGAISEQ_CALLEDSTART	API 開始関数はすでに発行されている
DMGAISYS	システムエラー	DMGAISYS_SYSTEM	JP1/NETM/DM Client でシステムエラーが発生した

表 B-2 概要コードと詳細コード (ソフトウェア操作・指令操作関数)

概要コード	意味	詳細コード	意味
DMGAIACCESS	ファイルアクセスエラー	DMGAIACCESS_FILEACCESS	ファイルアクセスエラー
DMGAICOM	通信エラー	DMGAICOM_COMMUNICATE	ソケット通信エラー
DMGAIERR	処理失敗	DMGAIERR_ATTRINV	資源属性が不正である
		DMGAIERR_BKUPATTRINV	バックアップリストの属性が不正である
		DMGAIERR_BKUPNOEXIST	バックアップリストがない
		DMGAIERR_EXIST	同一資源名がある
		DMGAIERR_FILENOTSUPPORT	パッケージとして登録できないファイルがある
		DMGAIERR_GRPNOTDEF	グループ名が定義されていない
		DMGAIERR_NOEXIST	パッケージのパスが存在しない
		DMGAIERR_RSCOVER	グループ内のパッケージ数が上限を超えた

概要コード	意味	詳細コード	意味
DMGAIERR	処理失敗	DMGAIERR_SCPATTRINV	スクリプトファイルの属性が不正である
		DMGAIERR_SCPNOEXIST	スクリプトファイルがない
		DMGAIERR_SYSFULL	ファイルシステムが満杯である
		DMGAIERR_USENMNOTDEF	ユーザ名が定義されていない
		DMGAIERR_USING	パッケージは使用中である
		DMGAIERR_VTFDELETE	仮想ファイルの削除に失敗した
DMGAIINVALID	引数不正	DMGAIINVALID_OPRCNT	引数の数が誤っている
		DMGAIINVALID_OPRCOMBI	引数の組み合わせが誤っている
		引数の指定順番の番号	引数の内容が不正である
DMGAINODATA	データがない	DMGAINODATA_NODATA	該当するデータがない
		DMGAINODATA_UNCOMPLETE	指定したパッケージまたはファイルは登録されていない
DMGAINRM	正常終了	DMGAINRM_INSTALLINF	指令を登録したが、組み込み情報は更新できなかった
		DMGAINRM_NORMAL	正常終了した
DMGAIREFUSE	要求が拒否された	DMGAIREFUSE_PASSWD	パッケージを登録したパスワードが不正である
		DMGAIREFUSE_LOCK	指定されたグループのパッケージは登録中または削除中である
		DMGAIREFUSE_STOPING	JP1/NETM/DM Client が停止処理中で要求が拒否された
		DMGAIREFUSE_WFSEND	起動ファイルの送信が拒否された
DMGAISYS	システムエラー	DMGAISYS_SYSTEM	JP1/NETM/DM Client でシステムエラーが発生した

表 B-3 概要コードと詳細コード (参照関数)

概要コード	意味	詳細コード	意味
DMGAIACCESS	ファイルアクセスエラー	DMGAIACCESS_FILEACCESS	ファイルアクセスエラー
DMGAICOM	通信エラー	DMGAICOM_COMMUNICATE	ソケット通信エラー
DMGAIINVALID	引数不正	引数の指定順番の番号	引数の内容が不正である
DMGAINODATA	データがない	DMGAINODATA_NODATA	該当するデータがない
DMGAINRM	正常終了	DMGAINRM_CMDKIND_D	配布・組み込み指令の入力が正常終了した
		DMGAINRM_HEAD	ヘッダレコードの入力が正常終了した
		DMGAINRM_NORMAL	正常終了した

概要コード	意味	詳細コード	意味
DMGAINRM	正常終了	DMGAINRM_ORIGINAL	オリジナルレコードの入力が正常終了した
DMGAIREFUSE	要求が拒否された	DMGAIREFUSE_REQREFUSE	JPI/NETM/DM Client が計画停止中で、要求が拒否された
DMGAISEQ	要求順序不正	DMGAISEQ_CALLEDSTART	API 開始関数はすでに発行されている
		DMGAISEQ_NOCALLFR	関数の発行順序が正しくない (_gaiXXXNext() は, _gaiXXXRewind()または_gaiXXXFind()の次に発行する*)
DMGAISYS	システムエラー	DMGAISYS_SYSTEM	JPI/NETM/DM Client でシステムエラーが発生した

注※

「XXX」には、「CMD」または「RSC」が入ります。

付録 B.2 配布状況ファイルアクセス関数のリターンコード

処理結果の詳細情報の格納エリアを参照するときに使う、処理結果の参照関数 (_gaiDLVGetMainCode(), _gaiDLVGetSubCode()) で参照できる概要コードと詳細情報の一覧を表 B-4 に示します。

表 B-4 概要コードと詳細コード (配布状況ファイルアクセス関数)

概要コード	意味	詳細コード	意味
DMWGAINRM	正常終了	DMWGAINRM_NORMAL	正常終了
DMWGAISYS	システムエラー	DMWGAISYS_SYSTEM	要求受け付け拒否
DMWGAACCESS	ファイルアクセスエラー	DMWGAACCESS_FILEACCESS	ファイルアクセスエラー
DMWGAINODATA	データなし	DMWGAINODATA_NODATA	入力データなし
DMWGAINVALID	引数間違い	引数の指定順番の番号	引数の内容間違い
DMWGAISEQ	要求順序の間違い	DMWGAISEQ_ALREADYSTART	配布状況ファイルオープン済み
		DMWGAISEQ_NOCALLFR	要求順序の間違い
DMWGAIGETSD	ソケット記述子	DMWGAIGETSD_OPENSOCKET	ソケットオープン失敗
—	獲得失敗	—	—
DMWGAICOM	ソケット通信エラー	DMWGAICOM_COMMUNICATE	ソケット通信エラー

(凡例)

—：該当しない

付録 C 各バージョンの変更内容

ここでは、各バージョンの変更内容を示します。

(1) 10-10 の変更内容

- 機能追加なし。

(2) 09-00 の変更内容

- JP1/NETM/DM SubManager を廃止し、ソフトウェアの中継機能を JP1/NETM/DM Client に統合した。そのため、マニュアル中での JP1/NETM/DM SubManager の記述を JP1/NETM/DM Client に変更した。

(3) 08-00 の変更内容

- OS が HP-UX(IPF)の場合に使用する共有ライブラリの記述を変更した。

(4) 07-50 の変更内容

- OS が HP-UX(IPF)の場合に使用する共有ライブラリの記述を追加した。
- OS が HP-UX(IPF)の場合に使用するライブラリ関数の形式の記述を追加した。

索引

記号

_gaiAPIEnd() 13
_gaiAPIStart() 12
_gaiCMDCreate() 20
_gaiCMDDelete() 24
_gaiCMDFind() 32
_gaiCMDFindNext() 33
_gaiCMDFindUnique() 34
_gaiCMDReferArea() 35
_gaiCMDRewind() 31
_gaiCMDRun() 23
_gaiDLVDel() 52
_gaiDLVDelUnique() 54
_gaiDLVEnd() 43
_gaiDLVFind() 46
_gaiDLVFindNext() 48
_gaiDLVFindUnique() 49
_gaiDLVGetMainCode() 56
_gaiDLVGetSubCode() 57
_gaiDLVReferArea() 50
_gaiDLVRewind() 45
_gaiDLVStart() 42
_gaiRSCCreateStore() 15
_gaiRSCDelete() 19
_gaiRSCFind() 27
_gaiRSCFindNext() 28
_gaiRSCFindUnique() 29
_gaiRSCReferArea() 30
_gaiRSCRewind() 26
_gaiRTNGetMainCode() 37
_gaiRTNGetSubCode() 38

A

API の使用開始 12
API の使用終了 13
API のリターンコードを参照する関数 36, 55

D

DM 種別コード 5

か

開始・終了の関数 11, 41
概要コードと詳細コード (開始・終了関数) 70
概要コードと詳細コード (参照関数) 71

概要コードと詳細コード (ソフトウェア操作・指令操作関数) 70
概要コードと詳細コード (配布状況ファイルアクセス関数) 72
概要コードの参照 37, 56
各バージョンの変更内容 73
完全資源名 5
管理ファイルの参照エリアと識別子 60

<

グループ id 5
グループ名 6

さ

参照エリアと識別子の一覧 67
参照関数の使用方法 6
参照の関数 25

し

資源属性ファイル 60
資源属性ファイルの参照 6, 25
資源属性ファイルの参照 (前回参照した次のレコードの参照) 28
資源属性ファイルの参照 (先頭レコードの参照) 27
資源属性ファイルの参照 (特定レコードの参照) 29
資源属性ファイルの参照 (ファイルの先頭レコードの参照) 26
資源属性ファイルの参照 (レコード中の特定エリアの参照) 30
資源属性ファイルの参照エリアと識別子 (エントリ部) 60
資源属性ファイルの参照エリアと識別子 (ヘッダ部) 60
資源名 6
実行管理ファイル 63
実行管理ファイルの参照 7, 25
実行管理ファイルの参照 (前回参照した次のレコードの参照) 33
実行管理ファイルの参照 (先頭レコードの参照) 32
実行管理ファイルの参照 (特定レコードの参照) 34
実行管理ファイルの参照 (ファイルの先頭レコードの参照) 31
実行管理ファイルの参照 (レコード中の特定エリアの参照) 35

使用言語 5
詳細コードの参照 38, 57
指令操作 14
指令の削除 24
指令の作成・登録 20
指令の実行 23

せ

世代番号 6

そ

操作の関数 14
ソフトウェア操作 14
ソフトウェアの登録（ソフトウェアを資源保管庫に登録する） 15

は

バージョン/リビジョン 6
配布管理の関数 2, 9
配布管理の関数の一覧 2
配布管理の関数のリターンコード 70
配布先指令番号 6
配布先名 5
配布状況ファイル 67
配布状況ファイルアクセス関数 3, 39
配布状況ファイルアクセス関数のリターンコード 72
配布状況ファイルの前回参照した次のレコードの参照 48
配布状況ファイルの先頭レコードの参照 45
配布状況ファイルの特定のレコードの削除 54
配布状況ファイルの特定レコードの参照 49
配布状況ファイルの任意のレコードの削除 52
配布状況ファイルの任意のレコードの参照 46
配布状況ファイルのレコード中の特定エリアの参照 50
配布状況ファイルへのアクセスの開始 42
配布状況ファイルへのアクセスの終了 43
配布指令の参照エリアと識別子 63
パッケージの削除 19

ひ

引数の設定方法 5

ら

ライブラリ関数の種類 2
ライブラリ関数の使用方法 5

り

リターンコード 70

れ

レコードの削除の関数 51
レコードの参照の関数 44