

CLUSTERPRO[®] システム構築ガイド

Ver6.0 (Windows[®] 2000 版)

API リファレンス

第6版 2002.12.12

改版履歷

版数	改版年月日	改版ページ	内容
第 1 版	2000. 2. 18		新規作成
第 2 版	2000. 3. 29	4	CLUSTERPRO ドキュメント体系を更新
第 3 版	2000. 7. 10	3	Windows 2000 対応について詳細変更
第 4 版	2000. 10. 6	183	6.1.5.1 ArmClientEnumCluster 「形式」の誤記訂正
第 5 版	2001. 8. 29	8	「1.2 CLUSTERPRO API の対応言語」C++で使用する際の注意点を追記
第 6 版	2002. 12. 12	全般	誤記等の訂正

はじめに

『CLUSTERPRO®システム構築ガイド』は、これからクラスタシステムを設計・導入しようとしているシステムエンジニアや、すでに導入されているクラスタシステムの保守・運用管理を行う管理者や保守員の方を対象にしています。

補足情報

【OS のアップグレードについて】

クラスタサーバの OS をアップグレードする場合、手順を誤ると予期せぬタイミングでフェイルオーバーが発生したり、最悪の場合、システムにダメージを与える可能性があります。

必ず製品添付のセットアップカードの手順に沿って OS をアップグレードしてください。
また、サービスパックの適用も上記に準じます。

Windows 2000 対応について

【ハードウェア】

クラスタシステムで使用するハードウェアの Windows 2000 対応状況については、製品通知などをご確認ください。

CLUSTERPRO®は日本電気株式会社の登録商標です。

Microsoft®, Windows®および 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。

CLARiiON ATF , CLARiiON Array Manager は米国 EMC 社 の商標です。

Oracle Parallel Server は米国オラクル社の商標です。

VERITAS , VERITAS ロゴおよび VERITAS Volume Manager は、VERITAS Software Corporation の登録商標または商標です。

その他のシステム名、社名、製品名等はそれぞれの会社の商標及び登録商標です。

CLUSTERPRO ドキュメント体系

CLUSTERPRO のドキュメントは、CLUSTERPRO をご利用になる局面や読者に応じて以下の通り分冊しています。初めてクラスタシステムを設計する場合は、システム構築ガイド【入門編】を最初に読んでください。

- セットアップカード (必須) 設計・構築・運用・保守
製品添付の資料で、製品構成や動作環境などについて記載しています。

- システム構築ガイド (必須) 設計・構築・運用・保守
【入門編】
クラスタシステムをはじめて設計・構築する方を対象にした入門書です。

- 【システム設計編(基本/共有ディスク,ミラーディスク)】 (必須) 設計・構築・運用・保守
クラスタシステムを設計・構築を行う上でほとんどのシステムで必要となる事項をまとめたノウハウ集です。構築前に知っておくべき情報、構築にあたっての注意事項などを説明しています。システム構成が共有ディスクシステムかミラーディスクシステムかで分冊しています。

- 【システム設計編(応用)】 (選択) 設計・構築・運用・保守
設計編(基本)で触れなかった CLUSTERPRO のより高度な機能を使用する場合に必要な事項をまとめたノウハウ集です。

- 【クラスタ生成ガイド(共有ディスク,ミラーディスク)】 (必須) 設計・構築・運用・保守
CLUSTERPRO のインストール後に行う環境設定を実際の作業手順に沿って分かりやすく説明しています。システム構成が共有ディスクシステムかミラーディスクシステムかで分冊しています。

- 【運用/保守編】 (必須) 設計・構築・運用・保守
クラスタシステムの運用を行う上で必要な知識と、障害発生時の対処方法やエラー一覧をまとめたドキュメントです。

- 【GUI リファレンス】 (必須) 設計・構築・運用・保守
クラスタシステムの運用を行う上で必要な CLUSTERPRO マネージャなどの操作方法をまとめたリファレンスです。

- 【コマンドリファレンス】 (選択) 設計・構築・運用・保守
CLUSTERPRO のスクリプトに記述できるコマンドやサーバまたはクライアントのコマンドプロンプトから実行できる運用管理コマンドについてのリファレンスです。

- 【API リファレンス】 (選択) 設計・構築・運用・保守
CLUSTERPRO が提供する API を利用してクラスタシステムと連携したアプリケーションを作成する場合にお使いいただくリファレンスです。

- 【PP 編】 (選択必須) 設計・構築・運用・保守
この編に記載されている各 PP は、CLUSTERPRO と連携して動作することができます。
各 PP が、CLUSTERPRO と連携する場合に必要な設定や、スクリプトの記述方法、注意事項などについて説明しています。使用する PP については必ずお読みください。

- 【注意制限事項集】 (選択) 設計・構築・運用・保守
クラスタシステム構築時、運用時、異常動作等障害対応時に注意しなければならない事項を記載したリファレンスです。必要に応じてお読み下さい。

目 次

1	はじめに.....	8
1.1	CLUSTERPRO 運用形態.....	8
1.2	CLUSTERPRO API の対応言語.....	8
2	CLUSTERPRO サーバ API 概要.....	9
2.1	CLUSTERPRO API 機能概要.....	9
2.2	Ver3.0 互換 API と Ver4.x ネイティブ API.....	10
2.3	開発環境.....	12
2.3.1	CLUSTERPRO Ver3.0 互換 API 開発キット.....	12
2.3.2	CLUSTERPRO Ver4.x API 開発キット.....	12
2.3.3	サーバ API 開発キットの使用について.....	13
2.4	CLUSTERPRO API 使用方法.....	14
2.5	注意事項.....	15
2.5.1	Ver4.0/4.1 以降の互換性に関する注意事項.....	15
2.5.2	フローティング IP アドレスに関する注意事項.....	15
2.5.3	CLUSTERPRO API を使用する上での注意事項.....	15
2.6	補足.....	16
2.6.1	構造体メンバのアライメント.....	16
3	サーバ API 詳細 — Ver4.x ネイティブ —.....	17
3.1	関数.....	18
3.1.1	API の初期化/終了処理.....	19
3.1.2	クラスタシステム定義情報の取得.....	23
3.1.3	クラスタ状態の取得/監視.....	44
3.1.4	レジストリの操作.....	56
3.2	構造体.....	57
3.2.1	ARM_SERVER_INFO — サーバ定義情報.....	58
3.2.2	ARM_GROUP_INFO — フェイルオーバーグループ定義情報.....	59
3.2.3	ARM_RESOURCE_INFO — リソース定義情報.....	60
3.2.4	ARM_GROUP_STATUS — フェイルオーバーグループ動作状態.....	68
3.2.5	ARM_RESOURCE_STATUS — リソース動作状態.....	69
3.2.6	ARM_EVENT_INFO — クラスタイベント情報.....	70
3.3	クラスタイベント.....	72
4	サーバ API 詳細 — Ver3.0 互換 —.....	73
4.1	関数.....	74
4.1.1	API の初期化/終了.....	75
4.1.2	クラスタシステム定義情報の取得.....	78
4.1.3	クラスタ状態の取得/監視.....	96
4.1.4	レジストリの操作.....	113
4.2	構造体.....	126
4.2.1	ARM_SERVER_INFO — サーバ定義情報.....	127
4.2.2	ARM_RESOURCE_INFO — リソース定義情報.....	128
4.2.3	ARM_EVENT_PARAM — サーバ状態変更通知パラメータ.....	135
5	CLUSTERPRO クライアント API 概要.....	136
5.1	CLUSTERPRO クライアント API 機能概要.....	136
5.1.1	主な機能.....	136
5.1.2	CLUSTERPRO クライアント API を使用するメリット.....	136
5.1.3	対応言語.....	136

5.2 CLUSTERPRO Ver3.0 からの変更点及び互換性留意事項.....	137
5.2.1 サーバ数の増加とフェイルオーバーグループの概念導入.....	137
5.2.2 Ver3.0 対応アプリケーションサービスを使用するための注意.....	139
5.2.3 追加されたクラスタイベントのクライアント API イベントへのマッピング.....	140
5.2.4 仮想コンピュータ名のサポート.....	141
5.2.5 フローティング IP アドレスのサポート.....	141
5.2.6 CLUSTERPRO Ver3.0 との差分一覧.....	142
5.2.7 マクロ定義.....	143
5.2.8 互換性留意事項.....	144
5.3 開発環境 (Visual C++)	146
5.3.1 CLUSTERPRO Ver4.x クライアント API 開発キット.....	146
5.3.2 CLUSTERPRO Ver1.0 互換クライアント API 開発キット.....	147
5.3.3 CLUSTERPRO クライアント API サンプルプログラム	148
5.4 開発環境 (Visual Basic)	154
5.4.1 CLUSTERPRO クライアント API 開発キット.....	154
5.4.2 CLUSTERPRO クライアント API サンプルプログラム	155
6 クライアント API 詳細 (Visual C++)	159
6.1 関数.....	159
6.1.1 API の初期化と終了.....	159
6.1.2 フェイルオーバーグループ状態問い合わせ.....	163
6.1.3 フェイルオーバーグループ監視.....	174
6.1.4 サーバ状態問い合わせ.....	179
6.1.5 クラスタ情報の取得.....	182
6.1.6 仮想 IP アドレス	197
6.1.7 仮想コンピュータ名からの情報の取得.....	207
6.1.8 論理サービス名.....	213
6.1.9 クラスタパラメータの取得.....	218
6.1.10 ローカルサーバ関連 API.....	222
6.2 構造体.....	226
6.2.1 ARMCLAPI_FAILOVER_INFO - フェイルオーバー情報.....	227
6.3 タイムチャート.....	230
6.3.1 フェイルオーバー時のタイムチャート.....	231
6.3.2 クラスタ復帰時のタイムチャート	232
6.3.3 クライアント起動時にバックアップ運用中である場合のタイムチャート.....	233
6.3.4 フェイルオーバー中にクライアントが起動した場合のタイムチャート.....	234
6.3.5 クラスタシャットダウン中にクライアントが起動した場合のタイムチャート.....	235
6.3.6 クライアントが起動した後、サーバBのみ起動した場合のタイムチャート.....	236
6.4 注意事項.....	237
6.4.1 32bit 版クライアントアプリケーションと 16bit 版クライアントアプリケーションを同一クライアント上で実行させる場合の注意.....	237
7 クライアント API 詳細 (Visual Basic)	238
7.1 関数.....	238
7.1.1 クライアント API の Visual Basic 版概要.....	238
7.1.2 API 一覧.....	239
7.1.3 API 詳細.....	240
7.2 ユーザー定義型 (構造体)	281
7.3 定数.....	282
7.3.1 拡張拡張エラー情報.....	282
7.3.2 ハンドル	282
7.3.3 イベント	282

7.3.4	フェイルオーバーグループの状態	282
7.3.5	サーバステータス	283
7.3.6	ホストタイプ	283
7.3.7	真理値	283
7.4	注意事項	284
7.4.1	関数の戻り値	284
7.4.2	API 関数での引数の文字列	284
7.4.3	コールバック関数での文字列の受け取り	284
7.4.4	拡張エラー情報	284
8	仮想 IP アドレス使用時の注意事項	285
8.1	問題点/制限事項	285
8.2	サーバからのブロードキャストをきっかけとした接続	290
9	関数・構造体索引	291

1 はじめに

1.1 CLUSTERPRO 運用形態

CLUSTERPROでは、片方向スタンバイクラスタ、双方向スタンバイクラスタとパラレルクラスタという3つのクラスタシステムを構築できます。

また、これら3つの運用とは別にVer3.0互換用にVer3.0互換運用のシステムを構築することができます。

1.2 CLUSTERPRO API の対応言語

CLUSTERPROサーバAPIは、Microsoft Visual C++ Ver4.2以降に対応しています。

CLUSTERPROクライアントAPIは、Microsoft Visual C++ Ver4.2以降、およびMicrosoft Visual Basic Ver5以降に対応しています。

なお、CLUSTERPROサーバAPIおよびCLUSTERPROクライアントAPI (Microsoft Visual C++)は、C言語形式となっています。C++から使用するためには、APIのヘッダファイルをインクルードしている前後に、次のような記載が必要です。記載がない場合、リンク時にエラー(外部シンボル xxxxは未解決です)となります。

```
#ifdef __cplusplus
extern "C" {
#endif

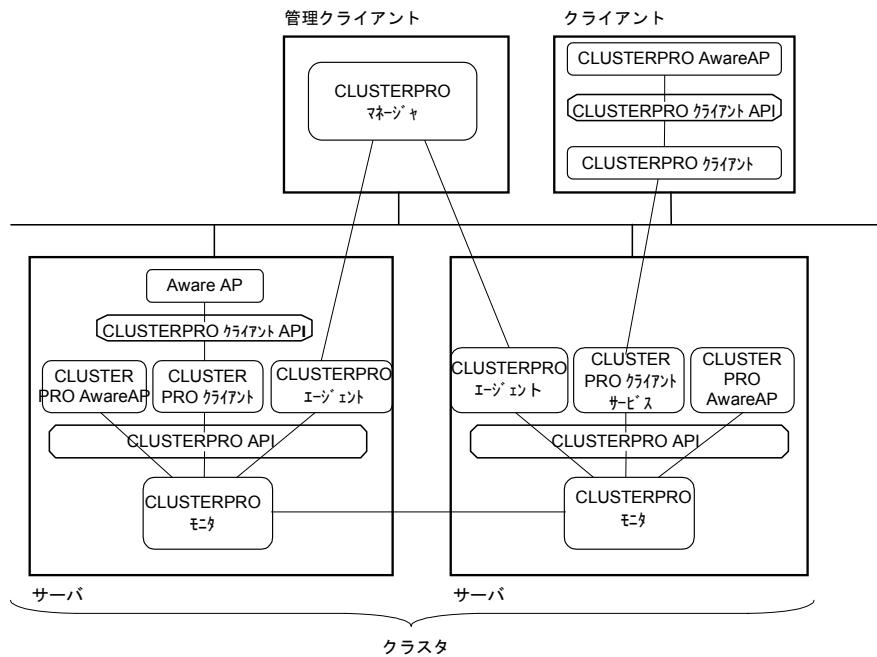
#include "armapi4.h"

#ifdef __cplusplus
}
#endif
```

2 CLUSTERPROサーバAPI概要

2.1 CLUSTERPRO API 機能概要

CLUSTERPRO APIは、サーバ上で動作するアプリケーションから、CLUSTERPROモニタの提供する機能を利用できるようにするためのものです。CLUSTERPRO APIでは、アプリケーションが動作しているサーバ（ローカルサーバ）の所属するクラスタ内の情報を扱うことができます。



2.2 Ver3.0 互換 API と Ver4.x ネイティブ API

CLUSTERPRO Ver6.0では、以下の2種類のAPI動作環境を提供します。

Ver3.0互換API

CLUSTERPRO Ver3.0用に開発されたアプリケーションをCLUSTERPRO Ver4.x以降で動作可能とする環境を提供します。Ver3.0用アプリケーションは、Ver3.0互換運用において、プログラムを変更することなくそのまま動作することが可能です。CLUSTERPRO Ver4.x以降を以下の条件で運用した場合にVer3.0互換運用となります。

- * **2サーバ構成であること。**
- * **クラスタ内には、各サーバをプライマリとするフェイルオーバーグループが1つずつ（クラスタ内で合計2つ）であること。**
- * **各フェイルオーバーグループのグループ名は、そのプライマリとなるサーバのサーバ名と同じであること。**

上記以外の運用形態(片方向スタンバイクラスタ、双方向スタンバイクラスタ、パラレルクラスタ)では、Ver3.0用アプリケーションの動作は保証しません。

Ver4.xネイティブAPI

CLUSTERPRO Ver4.x以降では、主に以下のような機能拡張が行なわれています。

- * **クラスタを構成するサーバ数の拡大（2サーバ ⇒ 16サーバ）**
- * **フェイルオーバー単位の変更（サーバ単位 ⇒ フェイルオーバーグループ単位）**

Ver4.xネイティブAPIは、上記の新機能に対応したAPIです。基本的にVer3.0以前のAPIとの互換性はありません。

Ver3.0 / Ver4.x API機能対応表

機能	Ver3.0	Ver4.x
CLUSTERPRO API初期化	ArmInitializeApi	ArmInitializeApi
CLUSTERPRO API終了	ArmTerminateApi	ArmTerminateApi
クラスタ名の取得	ArmGetClusterName	ArmGetClusterName
サーバの列挙	ArmEnumServer	ArmEnumServer
サーバ定義情報の取得	ArmGetServerInfo	ArmGetServerInfo
フェイルオーバーグループ定義情報の取得	—	ArmEnumGroup
フェイルオーバーグループの列挙	—	ArmGetGroupInfo
リソースの列挙	ArmEnumResource	ArmEnumResource
	ArmEnumResource2	
リソース定義情報の取得	ArmGetResourceInfo	ArmGetResourceInfo
	ArmGetResourceInfo2	
論理サービス名の列挙	ArmEnumLogicalService	ArmEnumLogicalService
論理サービス名定義情報の取得	ArmGetLogicalServiceInfo	
サーバ動作状態の取得	ArmGetServerStatus	ArmGetServerStatus
フェイルオーバーグループ動作状態の取得	—	ArmGetGroupStatus
リソース所有権の確認	ArmCheckResourceOwner	ArmGetResourceStatus
リソース動作状態の取得	—	
クラスタイベントの監視 (同期型)	ArmWatchServer	※
クラスタイベント監視の開始 (非同期コールバック)	ArmStartWatchServer	※
クラスタイベント監視の中止 (非同期コールバック)	ArmCancelWatchServer	※
クラスタイベント監視の開始 (イベントオブジェクト)	ArmRegisterWatchServer-Event	ArmStartEventWatch
クラスタイベント監視の中止 (イベントオブジェクト)	ArmCancelWatchServer-Event	ArmCancelEventWatch
クラスタイベント情報の取得 (イベントオブジェクト)	ArmGetServerEvent	ArmGetEventInfo
レジストリキーのオープン	ArmRegOpenKey	ArmRegOpenKey
レジストリキーの作成/オープン	ArmRegCreateKey	ArmRegCreateKey
レジストリキーの削除	ArmRegDeleteKey	ArmRegDeleteKey
レジストリのフラッシュ	ArmRegFlushKey	ArmRegFlushKey
レジストリキーのクローズ	ArmRegCloseKey	ArmRegCloseKey
レジストリキー値の設定	ArmRegSetValue	ArmRegSetValue
レジストリキー値の削除	ArmRegDeleteValue	ArmRegDeleteValue

※ Ver4.xでは、同期型／非同期コールバック型クラスタイベント監視は廃止します。クラスタイベント監視にはArmStartEventWatch()を使用してください。

2.3 開発環境

2.3.1 CLUSTERPRO Ver3.0 互換 API 開発キット

CLUSTERPRO Ver3.0用に開発したアプリケーションをCLUSTERPRO Ver4.x以降で使用するための開発キットです。新規にCLUSTERPRO上で動作するアプリケーションを開発する場合は、CLUSTERPRO Ver4.x API開発キットを使用してください。

Ver3.0用アプリケーションは、CLUSTERPRO Ver4.x以降でVer3.0互換運用をおこなっている場合のみ動作可能です。再コンパイル/再リンクは不要です。

- * armapi.h ヘッダファイルです。CLUSTERPRO APIを使用するソースファイルにインクルードしてください。armdef.h, armevt.h, armerr.h, armproto.hはarmapi.hの内部でインクルードされるので、armapi.hのみをインクルードしてください。
- * armapi.lib インポートライブラリです。CLUSTERPRO APIを使用するプログラムをリンクする際に使用してください。
armapist.libと同時に使用することはできません。
- * armapist.lib スタティックリンクライブラリです。CLUSTERPRO APIを使用するプログラムをリンクする際に使用してください。
armapi.libと同時に使用することはできません。
- * armapi.dll CLUSTERPRO API (Ver3.0互換) を提供するダイナミックリンクライブラリです。

2.3.2 CLUSTERPRO Ver4.x API 開発キット

- * armapi4.h ヘッダファイルです。CLUSTERPRO APIを使用するソースファイルにインクルードしてください。
- * armapi4.lib インポートライブラリです。CLUSTERPRO APIを使用するプログラムをリンクする際に使用してください。
armapi4s.libと同時に使用することはできません。
- * armapi4s.lib スタティックリンクライブラリです。CLUSTERPRO APIを使用するプログラムをリンクする際に使用してください。
armapi4.libと同時に使用することはできません。
- * armapi4.dll CLUSTERPRO API (Ver4.x) を提供するダイナミックリンクライブラリです。

2.3.3 サーバ API 開発キットの使用について

CLUSTERPROサーバAPI開発キットは、CLUSTERPROサーバをインストールすると自動的にインストールされます。CLUSTERPROサーバAPIを使用したAPを、CLUSTERPROサーバをインストールしたマシン以外で開発したい場合、CLUSTERPROサーバAPI開発キットをそのマシンにコピーしてください。

CLUSTERPROサーバAPI開発キットは、以下のフォルダにありますので、このフォルダ配下を全てコピーしてください。

(インストールディレクトリ)¥esmarm¥armapi

2.4 CLUSTERPRO API 使用方法

CLUSTERPRO APIは、ダイナミックリンクライブラリ (ARMAPI.DLL / ARMAPI4.DLL) で提供されます。ARMAPI.DLL / ARMAPI4.DLLは、以下のいずれかの方法でアプリケーションから使用することができます。

- * **CLUSTERPRO APIのインポートライブラリARMAPI.LIB / ARMAPI4.LIBをリンクしてアプリケーションを作成する。**

ARMAPI.DLL / ARMAPI4.DLLは、アプリケーション起動時にOSにより自動的にロードされます。ARMAPI.DLL / ARMAPI4.DLLが検索パス内に見つからない場合にはエラーとなり、アプリケーションは起動されません。したがって、CLUSTERPROがインストールされていない環境では、アプリケーションは動作できません。

- * **CLUSTERPRO APIのスタティックライブラリARMAPIST.LIB / ARMAPI4S.LIBをリンクしてアプリケーションを作成する。**

CLUSTERPRO API初期化関数 (ArmInitializeApi) 実行時に、ARMAPIST.LIB / ARMAPI4S.LIB内部でARMAPI.DLL / ARMAPI4.DLLを検索しロードします (CLUSTERPROのインストールディレクトリをレジストリから取得し、インストールディレクトリ下のARMAPI.DLL / ARMAPI4.DLLをロードします)。ARMAPI.DLL / ARMAPI4.DLLが見つからない場合 (CLUSTERPROがインストールされていない場合) は、初期化関数がエラーを返します。この場合、アプリケーションはCLUSTERPRO APIの提供する機能を使用することはできませんが、アプリケーション自体の動作を続行することには支障ありません。

APIのライブラリ以外に、システム標準のライブラリとしてKERNEL32.LIBおよびADVAPI.LIBが必要です。

- * **アプリケーション自身でLoadLibrary関数 (Win32API) を使用してARMAPI.DLL / ARMAPI4.DLLをロードする。**

ARMAPI.LIB / ARMAPI4.LIB, ARMAPIST.LIB / ARMAPI4S.LIBをリンクせず、Win32APIを使用してアプリケーション自身でARMAPI.DLL / ARMAPI4.DLLをロードしても構いません。

2.5 注意事項

2.5.1 Ver4.0/4.1 以降の互換性に関する注意事項

CLUSTERPRO Ver6.0 製品には、以下の開発環境が含まれます。

PP	開発環境
Ver 3.0	3.0API開発キット
Ver 4.0	3.0互換API開発キット/4.xAPI開発キット(4.0)
Ver 4.1	3.0互換API開発キット/4.xAPI開発キット(4.0/4.1)
Ver 4.2	3.0互換API開発キット/4.xAPI開発キット(4.0/4.1)
Ver 5.0	3.0互換API開発キット/4.xAPI開発キット(5.0)
Ver 6.0	3.0互換API開発キット/4.xAPI開発キット(6.0)

また、開発環境と実行環境の組み合わせは以下のようになります。

実行環境 API 開発キット	3.0/3.0 互換	4.x			
		4.0	4.1	5.0	6.0
3.0	○	×	×	×	×
4.0	×	○	○*1	○*1	○
4.0 (3.0互換運用)	○	○	○*1	○*1	○
4.1/4.2	×	○*2	○	○	○
4.1/4.2 (3.0互換運用)	○*2	○*2	○	○	○
5.0	×	○*2	○	○	○
5.0 (3.0互換運用)	○*2	○*2	○	○	○
6.0	×	○*2	○	○	○
6.0 (3.0互換運用)	○*2	○*2	○	○	○

*1:CLUSTERPRO4.08 (例えばESS RL1999/06(RUR適用を含む))以降にて、対応しています。

*2:切替ミラーディスクリソースはVer4.0以前の切替ディスクリソースとして扱われます。

実行環境が4.2以下のとき、クラスタIP (負荷分散IP) のリソース列挙を行うと、結果からクラスタIPリソースを除外します。

2.5.2 フローティング IP アドレスに関する注意事項

CLUSTERPRO Ver4.2 からサポートしているフローティング IP アドレスを使用する場合、フローティング IP アドレスを、仮想 IP アドレスの一形態として扱います。

2.5.3 CLUSTERPRO API を使用する上での注意事項

CLUSTERPRO APIによって得られるクラスタに関する情報は、クラスタとして正常動作しているサーバ上でのみ保証されます。

ArmGetServerStatus関数で、サーバがクラスタとして正常動作しているかどうかを確認の上、処理を行ってください。

2.6 補足

2.6.1 構造体メンバのアライメント

構造体メンバのアライメントは、8 バイトです。

3 サーバAPI詳細 — Ver4. xネイティブ —

CLUSTERPRO Ver4. x以降対応のAPIを提供します。
Ver3. 0以前のAPIとの互換性はありません。

3.1 関数

API初期化／終了処理

- | | |
|------------------|------------------------------|
| ArmInitializeApi | — CLUSTERPRO APIの初期化処理を行います。 |
| ArmTerminateApi | — CLUSTERPRO APIの終了処理を行います。 |

クラスタシステム情報の取得

- | | |
|------------------------|---------------------------------------|
| ArmGetClusterName | — ローカルサーバが所属するクラスタのクラスタ名を取得します。 |
| ArmEnumServer | — クラスタに所属するサーバを列挙します。 |
| ServerEnumProc | — サーバ列挙用コールバック関数。 |
| ArmGetServerInfo | — クラスタに所属する指定サーバのサーバ定義情報を取得します。 |
| ArmEnumGroup | — クラスタ内で定義されているフェイルオーバーグループを列挙します。 |
| GroupEnumProc | — フェイルオーバーグループ列挙用コールバック関数。 |
| ArmGetGroupInfo | — フェイルオーバーグループ定義情報を取得します。 |
| ArmEnumResource | — クラスタ内で定義されているリソースを列挙します。 |
| ResourceEnumProc | — リソース列挙用コールバック関数。 |
| ArmGetResourceInfo | — クラスタ内で定義されている指定リソースのリソース定義情報を取得します。 |
| ArmEnumLogicalService | — クラスタ内で定義されている論理サービス名を列挙します。 |
| LogicalServiceEnumProc | — 論理サービス名列挙用コールバック関数。 |

クラスタ動作状態の取得／監視

- | | |
|----------------------|----------------------------|
| ArmGetServerStatus | — サーバの動作状態を取得します。 |
| ArmGetGroupStatus | — フェイルオーバーグループの動作状態を取得します。 |
| ArmGetResourceStatus | — リソースの動作状態を取得します。 |
| ArmStartEventWatch | — クラスタイベントの監視を開始します。 |
| ArmCancelEventWatch | — クラスタイベントの監視を中止します。 |
| ArmGetEventInfo | — クラスタイベント情報を取得します。 |

レジストリ操作

- | | |
|-------------------|---------------------------|
| ArmRegOpenKey | — レジストリキーをオープンします。 |
| ArmRegCreateKey | — レジストリキー作成/オープンします。 |
| ArmRegDeleteKey | — レジストリキーを削除します。 |
| ArmRegFlushKey | — オープン中のキーをレジストリに書き込みます。 |
| ArmRegCloseKey | — レジストリキーをクローズします。 |
| ArmRegSetValue | — レジストリキーに値を設定します。 |
| ArmRegDeleteValue | — レジストリキーに設定されている値を削除します。 |

3.1.1 API の初期化／終了処理

CLUSTERPRO APIを使用するアプリケーションは、他のCLUSTERPRO API関数を使用する前に必ずArmInitializeApi関数を実行してCLUSTERPRO APIの初期化を行わなければなりません。また、アプリケーションを終了する前にArmTerminateApi関数を実行してCLUSTERPRO APIの終了処理を行わなければなりません。

- * ArmInitializeApi — CLUSTERPRO APIの初期化処理を行います。
- * ArmTerminateApi — CLUSTERPRO APIの終了処理を行います。

ArmInitializeApi, ArmTerminateApiは、同一プロセス内で複数独立に実行することが可能です。

3.1.1.1 ArmInitializeApi — CLUSTERPRO API の初期化

概要

CLUSTERPRO APIの初期化を行います。

形式

```
BOOL ArmInitializeApi (  
    LPDWORD lpdwArmApiHandle,  
    DWORD dwVersion,  
    DWORD dwTimeout  
)
```

パラメータ

- | | |
|-----------------------------|---|
| [o] <i>lpdwArmApiHandle</i> | CLUSTERPRO APIを使用するためのハンドルを受け取る変数へのポインタです。CLUSTERPRO APIハンドルは、クラスタイイベント監視およびCLUSTERPRO API終了処理に必要です。 |
| [i] <i>dwVersion</i> | アプリケーションの使用するCLUSTERPRO APIのバージョンを指定します。
ここには、CLUSTERPRO APIの提供するヘッダファイルで定義されているバージョン番号 ARMAPI_VERSION を指定してください。 |
| [i] <i>dwTimeout</i> | CLUSTERPROモニタの起動待ちタイムアウト時間を秒単位で指定します。
INFINITE を指定した場合には、タイムアウトは発生しません。 |

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_CANNOT_LOAD_DLL	armapi4.dllをロードできません。
ARMERR_VERSION_MISMATCH	armapi4.dllのバージョンが異なります。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_NOT_ENOUGH_MEMORY	メモリのアロケートに失敗しました。
ARMERR_CANNOT_ALLOCATE_RESOURCE	OS資源のアロケートに失敗しました。
ARMERR_CANNOT_CREATE_THREAD	スレッドの作成に失敗しました。

説明

本関数は、CLUSTERPRO APIの初期化を行います。CLUSTERPRO APIを使用するアプリケーションは、最初に本関数を実行して、CLUSTERPRO APIを初期化しなければなりません。本関数は同一プロセス内で複数回実行してもかまいません。

スタティックリンクライブラリ `armapi4s.lib` をリンクして作成されたアプリケーションでは、CLUSTERPROがインストールされていない (`armapi4.dll` をロードできない) 場合、本関数は `FALSE` を返却します。このとき拡張エラー情報で `ARMERR_CANNOT_LOAD_DLL` が返却されます。

本関数はインストールされている `armapi4.dll` のバージョンをチェックし、互換性が無い場合には `FALSE` を返却します。このとき拡張エラー情報で `ARMERR_VERSION_MISMATCH` が返却されます。

本関数は、CLUSTERPROモニタが動作していることを確認するために、CLUSTERPROモニタへの接続を試みます。CLUSTERPROモニタに接続できない場合、パラメータ `dwTimeout` で指定されたタイムアウトが経過するまで定期的に接続をリトライします。タイムアウトが経過しても接続できなかった場合には `FALSE` を返却します。このとき拡張エラー情報で `ARMERR_CANNOT_CONNECT` が返却されます。

参照

3.1.1.2 `ArmTerminateApi` — CLUSTERPRO APIの終了

3.1.1.2 ArmTerminateApi — CLUSTERPRO API の終了

概要

CLUSTERPRO APIの終了処理を行います。

形式

```
BOOL ArmTerminateApi (  
    DWORD dwArmApiHandle  
)
```

パラメータ

[i] *dwArmApiHandle* ArmInitializeApi関数で返却されたCLUSTERPRO APIハンドルを指定します。

戻り値

TRUE 関数が正常終了しました。
FALSE 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_INVALID_HANDLE	指定されたハンドルが不正です。

説明

本関数は、CLUSTERPRO APIの終了処理を行います。 ArmInitializeApi関数を実行してCLUSTERPRO APIの初期化を行なったアプリケーションは、アプリケーションを終了する前に本関数を実行してCLUSTERPRO APIの終了処理を行わなければなりません。 ArmInitializeApi関数を実行した回数と同じ回数だけ本関数を実行してください。

参照

3.1.1.1 ArmTerminateApi — CLUSTERPRO APIの終了

3.1.2 クラスタシステム定義情報の取得

クラスタシステムに関する定義情報を取得するためのAPIを提供します。

ArmGetClusterName	ー ローカルサーバが所属するクラスタのクラスタ名を取得します。
ArmEnumServer	ー クラスタに所属するサーバを列挙します。
ServerEnumProc	ー サーバ列挙用コールバック関数。
ArmGetServerInfo	ー クラスタに所属する指定サーバのサーバ定義情報を取得します。
ArmEnumGroup	ー クラスタ内で定義されているフェイルオーバーグループを列挙します。
GroupEnumProc	ー フェイルオーバーグループ列挙用コールバック関数。
ArmGetGroupInfo	ー フェイルオーバーグループ定義情報を取得します。
ArmEnumResource	ー クラスタ内で定義されているリソースを列挙します。
ResourceEnumProc	ー リソース列挙用コールバック関数。
ArmGetResourceInfo	ー クラスタ内で定義されている指定リソースのリソース定義情報を取得します。
ArmEnumLogicalService	ー クラスタ内で定義されている論理サービス名を列挙します。
LogicalServiceEnumProc	ー 論理サービス名列挙用コールバック関数。

3.1.2.1 ArmGetClusterName — クラスタ名の取得

概要

ローカルサーバが所属するクラスタの名前を取得します。

形式

```
BOOL ArmGetClusterName (  
    LPTSTR  lpszClusterName,  
    LPDWORD lpcchClusterName  
)
```

パラメータ

- | | |
|------------------------------|---|
| [o] <i>lpszClusterName</i> | クラスタ名（終端のNULL文字を含む）を受け取るバッファへのポインタです。 |
| [io] <i>lpcchClusterName</i> | バッファ <i>lpszClusterName</i> のサイズを示す変数へのポインタです。関数のリターン時には、バッファに格納された文字数（終端のNULL文字を含まない）がセットされます。また、指定バッファサイズが小さすぎた場合には、必要な文字数（終端のNULL文字を含む）がセットされ、拡張エラー情報で ARMERR_BUFFER_TOO_SMALL が返却されます。
(※ クラスタ名の最大長は MAX_CLUSTERNAME_LEN です) |

戻り値

- | | |
|-------|--|
| TRUE | 関数が正常終了しました。 |
| FALSE | 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。 |

拡張エラー情報

- | | |
|----------------------------------|-------------------------------|
| ARMERR_NOT_INITIALIZED | CLUSTERPRO APIが初期化されていません。 |
| ARMERR_CANNOT_CONNECT | CLUSTERPROモニタに接続できません。 |
| ARMERR_COMM_ERROR | CLUSTERPROモニタとの通信でエラーが発生しました。 |
| ARMERR_NOT_CLUSTER_MEMBER | ローカルサーバがクラスタメンバではありません。 |
| ARMERR_INVALID_PARAMETER | パラメータの指定が不正です。 |
| ARMERR_BUFFER_TOO_SMALL | データ返却用バッファが小さすぎます。 |

説明

本関数は、ローカルサーバが所属するクラスタの名前を取得し、*lpszClusterName*で指定されたバッファに格納します。

ローカルサーバがクラスタメンバでない場合、本関数はFALSEを返却します。このとき拡張エラー情報で **ARMERR_NOT_CLUSTER_MEMBER** が返却されます。

3.1.2.2 ArmEnumServer — サーバの列挙

概要

クラスタに所属するサーバを列挙します。

形式

```
BOOL ArmEnumServer (
    ARM_SRV_ENUM_PROC  ServerEnumProc,
    LPARAM  lParam
)
```

パラメータ

- [i] *ServerEnumProc* クラスタに所属するサーバのコンピュータ名を受け取る、アプリケーション定義のコールバック関数のアドレスです。
(3.1.2.3 ServerEnumProc関数参照)
- [i] *lParam* コールバック関数*ServerEnumProc*に渡される、アプリケーション定義の32ビット値です。

戻り値

- TRUE 関数が正常終了しました。
- FALSE 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_NOT_ENOUGH_MEMORY	メモリのアロケートに失敗しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_NOT_CLUSTER_MEMBER	ローカルサーバがクラスタメンバではありません。
ARMERR_OFF_CLUSTER	ローカルサーバがクラスタとして動作していません。
ARMERR_CANCELLED	コールバック関数から列挙の中止が返却されました。

説明

本関数は、ローカルサーバの所属するクラスタを構成しているサーバを列挙します。本関数は、アプリケーション定義のコールバック関数*ServerEnumProc*を順次呼び出し、各サーバのコンピュータ名を渡します。本関数は、クラスタを構成しているサーバをすべて列挙するか、またはコールバック関数がFALSEを返却するまで、列挙を続けます。

ローカルサーバがクラスタとして動作していない場合、サーバを列挙することはできません。

- * ローカルサーバがクラスタに登録されていない場合、本関数はFALSEを返却します。このとき拡張エラー情報でARMERR_NOT_CLUSTER_MEMBERが返却されます。
- * ダウン後再起動状態等でローカルサーバがクラスタから一時的に切り離されている場合、本関数はFALSEを返却します。このとき拡張エラー情報でARMERR_OFF_CLUSTERが返却されます。

参照

3.1.2.3 ServerEnumProc — サーバ列挙用コールバック関数

3.1.2.3 ServerEnumProc — サーバ列挙用コールバック関数

概要

ArmEnumServer関数で列挙されるサーバのコンピュータ名を受け取るための、アプリケーション定義のコールバック関数です。

形式

```
BOOL CALLBACK ServerEnumProc(  
    LPCTSTR lpszServerName,  
    LPARAM lParam  
)
```

パラメータ

[i] *lpszServerName* クラスタに所属するサーバのコンピュータ名を示す、NULLで終わる文字列へのポインタです。

[i] *lParam* ArmEnumServer関数で指定されたアプリケーション定義の値です。

戻り値

TRUE 列挙を継続する場合に返却します。
FALSE 列挙を中止する場合に返却します。

説明

ServerEnumProc関数は、ArmEnumServer関数の実行結果として、クラスタを構成するサーバのコンピュータ名を受け取るための、アプリケーション定義のコールバック関数です。サーバの列挙が完了するか、または本関数でFALSEを返却するまで、サーバのコンピュータ名が本関数に順次受け渡されます。アプリケーションは、ArmEnumServer関数実行時に本関数のアドレスを指定しなければなりません。

本関数に渡される *lpszServerName* の指す文字列の内容は、本関数からリターンするまでの間でのみ有効です。文字列の内容を保持する必要がある場合には、本関数内で文字列の内容を別の記憶領域へコピーしてください。

本関数の関数名 “ServerEnumProc” は、アプリケーション定義の関数名のプレースホルダです。

参照

3.1.2.2 ArmEnumServer — サーバの列挙

3.1.2.4 ArmGetServerInfo — サーバ定義情報の取得

概要

指定したサーバのサーバ定義情報を取得します。

形式

```
BOOL ArmGetServerInfo (
    LPCTSTR lpszServerName,
    LPARM_SERVER_INFO lpArmServerInfo
)
```

パラメータ

- | | |
|----------------------------|--|
| [i] <i>lpszServerName</i> | サーバ定義情報の取得対象とするサーバのコンピュータ名を指定する、NULLで終わる文字列へのポインタです。ローカルサーバを指定する場合には、定数ARM_LOCAL_SERVERを指定することができます。 |
| [o] <i>lpArmServerInfo</i> | サーバ定義情報を受け取るARM_SERVER_INFO構造体へのポインタです。 |

戻り値

- | | |
|-------|--|
| TRUE | 関数が正常終了しました。 |
| FALSE | 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。 |

拡張エラー情報

- | | |
|---------------------------|-------------------------------|
| ARMERR_NOT_INITIALIZED | CLUSTERPRO APIが初期化されていません。 |
| ARMERR_CANNOT_CONNECT | CLUSTERPROモニタに接続できません。 |
| ARMERR_COMM_ERROR | CLUSTERPROモニタとの通信でエラーが発生しました。 |
| ARMERR_NOT_ENOUGH_MEMORY | メモリのアロケートに失敗しました。 |
| ARMERR_INVALID_PARAMETER | パラメータの指定が不正です。 |
| ARMERR_NOT_CLUSTER_MEMBER | ローカルサーバがクラスタメンバではありません。 |
| ARMERR_OFF_CLUSTER | ローカルサーバがクラスタとして動作していません。 |
| ARMERR_SERVER_NOT_FOUND | 指定サーバが存在しません。 |

説明

本関数は、 *lpszServerName*で指定されたクラスタ内のサーバの定義情報を取得し、*lpArmServerInfo*で指定されたバッファに格納します。

ローカルサーバがクラスタとして動作していない場合、他サーバの定義情報を取得することはできません（ローカルサーバの定義情報は取得できます）。

- * ローカルサーバがクラスタに登録されていない場合、本関数はFALSEを返却します。このとき拡張エラー情報で**ARMERR_NOT_CLUSTER_MEMBER**が返却されます。
- * ダウン後再起動状態等でローカルサーバがクラスタから一時的に切り離されている場合、本関数はFALSEを返却します。このとき拡張エラー情報で**ARMERR_OFF_CLUSTER**が返却されます。

参照

3.2.1 ARM_SERVER_INFO — サーバ定義情報

3.1.2.5 ArmEnumGroup — フェイルオーバーグループの列挙

概要

クラスタ内で定義されているフェイルオーバーグループを列挙します。
論理サービス名をキーにフェイルオーバーグループを検索できます。

形式

```
BOOL ArmEnumGroup (  
    LPCTSTR lpzLogicalService,  
    ARM_GRP_ENUM_PROC GroupEnumProc,  
    LPARAM lParam  
)
```

パラメータ

- | | |
|------------------------------|---|
| [i] <i>lpzLogicalService</i> | このパラメータに論理サービス名を指定すると、その論理サービス名を持つフェイルオーバーグループのみが列挙対象となります。NULLまたは空文字列を指定した場合は、クラスタ内の全フェイルオーバーグループが列挙対象となります。 |
| [i] <i>GroupEnumProc</i> | フェイルオーバーグループ名を受け取る、アプリケーション定義のコールバック関数のアドレスです。
(3.1.2.6 GroupEnumProc関数参照) |
| [i] <i>lParam</i> | コールバック関数 <i>GroupEnumProc</i> に渡される、アプリケーション定義の32ビット値です。 |

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_NOT_ENOUGH_MEMORY	メモリのアロケートに失敗しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_NOT_CLUSTER_MEMBER	ローカルサーバがクラスタメンバではありません。
ARMERR_OFF_CLUSTER	ローカルサーバがクラスタとして動作していません。
ARMERR_GROUP_NOT_FOUND	列挙可能なフェイルオーバーグループが存在しません。
ARMERR_CANCELLED	コールバック関数から列挙の中止が返却されました。

説明

本関数は、ローカルサーバが所属するクラスタ内で定義されているフェイルオーバーグループを列挙します。本関数は、アプリケーション定義のコールバック関数 *GroupEnumProc* を順次呼び出し、各フェイルオーバーグループのグループ名を渡します。本関数は、列挙対象のフェイルオーバーグループをすべて列挙するか、またはコールバック関数がFALSEを返却するまで列挙を続けます。

本関数は、論理サービス名をキーにしてフェイルオーバーグループの列挙を行なうことができます。パラメータ *lpSzLogicalService* に論理サービス名を指定した場合、指定された論理サービス名を持つフェイルオーバーグループのみが列挙対象となります。この機能は、アプリケーションが自分の関係するフェイルオーバーグループを知るための手段として利用することができます。

ローカルサーバがクラスタとして動作していない場合、フェイルオーバーグループを列挙することはできません。

- * ローカルサーバがクラスタに登録されていない場合、本関数はFALSEを返却します。このとき拡張エラー情報で **ARMERR_NOT_CLUSTER_MEMBER** が返却されます。
- * ダウン後再起動状態等でローカルサーバがクラスタから一時的に切り離されている場合、本関数はFALSEを返却します。このとき拡張エラー情報で **ARMERR_OFF_CLUSTER** が返却されます。

参照

3.1.2.6 GroupEnumProc — フェイルオーバーグループ列挙用コールバック関数

3.1.2.6 GroupEnumProc — フェイルオーバーグループ列挙用コールバック関数

概要

ArmEnumGroup関数で列挙されるフェイルオーバーグループのグループ名を受け取るための、アプリケーション定義のコールバック関数です。

形式

```
BOOL CALLBACK GroupEnumProc (  
    LPCTSTR lpszGroupName,  
    LPCTSTR lpszLogicalService,  
    LPARAM lParam  
)
```

パラメータ

[i] <i>lpszGroupName</i>	フェイルオーバーグループ名を示すNULLで終わる文字列へのポインタです。
[i] <i>lpszLogicalService</i>	列挙対象を指定するキーとしてArmEnumGroup関数に渡された論理サービス名です。ArmEnumGroup関数に指定された値がそのまま渡されます。
[i] <i>lParam</i>	ArmEnumGroup関数で指定されたアプリケーション定義の値です。

戻り値

TRUE	列挙を継続する場合に返却します。
FALSE	列挙を中止する場合に返却します。

説明

GroupEnumProc関数は、ArmEnumGroup関数の実行結果として、クラスタ内で定義されているフェイルオーバーグループのグループ名を受け取るための、アプリケーション定義のコールバック関数です。フェイルオーバーグループの列挙が完了するか、または本関数でFALSEを返却するまで、グループ名が本関数に順次受け渡されます。アプリケーションは、ArmEnumGroup関数実行時に本関数のアドレスを指定しなければなりません。

本関数に渡される *lpszGroupName* の指す文字列の内容は、本関数からリターンするまでの間でのみ有効です。文字列の内容を保持する必要がある場合には、本関数内で文字列を別の記憶領域へコピーしてください。

本関数の関数名 “ GroupEnumProc ” は、アプリケーション定義の関数名のプレースホルダです。

参照

3.1.2.5 ArmEnumGroup — フェイルオーバーグループの列挙

3.1.2.7 ArmGetGroupInfo — フェイルオーバーグループ定義情報の取得

概要

指定したフェイルオーバーグループの定義情報を取得します。

形式

```
BOOL ArmGetGroupInfo (
    LPCTSTR lpszGroupName,
    LPARM_GROUP_INFO lpArmGroupInfo,
    LPDWORD lpcbArmGroupInfo
)
```

パラメータ

- | | |
|------------------------------|---|
| [i] <i>lpszGroupName</i> | 定義情報の取得対象とするフェイルオーバーグループのグループ名を指定する、NULLで終わる文字列へのポインタです。 |
| [o] <i>lpArmGroupInfo</i> | 定義情報（ARM_GROUP_INFO構造体）を受け取るバッファへのポインタです。 |
| [io] <i>lpcbArmGroupInfo</i> | バッファ <i>lpArmGroupInfo</i> のサイズを示す変数へのポインタです。
関数のリターン時には、バッファに格納されたバイト数がセットされます。また、指定バッファサイズが小さすぎた場合には、必要なバイト数がセットされ、拡張エラー情報で ARMERR_BUFFER_TOO_SMALL が返却されます。 |

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_NOT_ENOUGH_MEMORY	メモリのアロケートに失敗しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_NOT_CLUSTER_MEMBER	ローカルサーバがクラスタメンバではありません。
ARMERR_OFF_CLUSTER	ローカルサーバがクラスタとして動作していません。
ARMERR_GROUP_NOT_FOUND	指定したフェイルオーバーグループが存在しません。
ARMERR_BUFFER_TOO_SMALL	データ返却用バッファが小さすぎます。

説明

本関数は、ローカルサーバが所属するクラスタ内で定義されている、 *lpszGroupName*で指定されたフェイルオーバーグループの定義情報を取得し、 *lpArmGroupInfo*で指定されたバッファに格納します。

ローカルサーバがクラスタとして動作していない場合、フェイルオーバーグループの定義情報を取得することはできません。

- * ローカルサーバがクラスタに登録されていない場合、本関数はFALSEを返却します。このとき拡張エラー情報でARMERR_NOT_CLUSTER_MEMBERが返却されます。
- * ダウン後再起動状態等でローカルサーバがクラスタから一時的に切り離されている場合、本関数はFALSEを返却します。このとき拡張エラー情報でARMERR_OFF_CLUSTERが返却されます。

参照

3.2.2 ARM_GROUP_INFO — フェイルオーバーグループ定義情報

3.1.2.8 ArmEnumResource — リソースの列挙

概要

クラスタ内で定義されているリソースを列挙します。

形式

```
BOOL ArmEnumResource (
    LPCTSTR lpszGroupName,
    DWORD dwResourceType,
    ARM_RSC_ENUM_PROC ResourceEnumProc,
    LPARAM lParam
)
```

パラメータ

[i] <i>lpszGroupName</i>	このパラメータにフェイルオーバーグループ名を指定すると、そのフェイルオーバーグループに所属するリソースのみが列挙対象となります。NULLまたは空文字列を指定した場合は、クラスタ内の全リソースが列挙対象となります。																
[i] <i>dwResourceType</i>	列挙対象とするリソースタイプを以下の組み合わせで指定します。 <table><tr><td>ARM_RESTYPE_ALL</td><td>全リソースタイプ</td></tr><tr><td>ARM_RESTYPE_SWDISK</td><td>切替ディスク</td></tr><tr><td>ARM_RESTYPE_VIP</td><td>仮想IPアドレス</td></tr><tr><td>ARM_RESTYPE_VCOM</td><td>仮想コンピュータ名</td></tr><tr><td>ARM_RESTYPE_PRINTER</td><td>プリンタ</td></tr><tr><td>ARM_RESTYPE_LSW</td><td>回線切替装置</td></tr><tr><td>ARM_RESTYPE_NDR</td><td>切替ミラーディスク</td></tr><tr><td>ARM_RESTYPE_HF</td><td>クラスタIP（負荷分散IP）</td></tr></table>	ARM_RESTYPE_ALL	全リソースタイプ	ARM_RESTYPE_SWDISK	切替ディスク	ARM_RESTYPE_VIP	仮想IPアドレス	ARM_RESTYPE_VCOM	仮想コンピュータ名	ARM_RESTYPE_PRINTER	プリンタ	ARM_RESTYPE_LSW	回線切替装置	ARM_RESTYPE_NDR	切替ミラーディスク	ARM_RESTYPE_HF	クラスタIP（負荷分散IP）
ARM_RESTYPE_ALL	全リソースタイプ																
ARM_RESTYPE_SWDISK	切替ディスク																
ARM_RESTYPE_VIP	仮想IPアドレス																
ARM_RESTYPE_VCOM	仮想コンピュータ名																
ARM_RESTYPE_PRINTER	プリンタ																
ARM_RESTYPE_LSW	回線切替装置																
ARM_RESTYPE_NDR	切替ミラーディスク																
ARM_RESTYPE_HF	クラスタIP（負荷分散IP）																
[i] <i>ResourceEnumProc</i>	リソースの識別子を受け取る、アプリケーション定義のコールバック関数のアドレスです。 (3.1.2.9 ResourceEnumProc関数参照)																
[i] <i>lParam</i>	コールバック関数 <i>ResourceEnumProc</i> に渡される、アプリケーション定義の32ビット値です。																

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPRO モニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPRO モニタとの通信でエラーが発生しました。
ARMERR_NOT_ENOUGH_MEMORY	メモリのアロケートに失敗しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。

ARMERR_NOT_CLUSTER_MEMBER	ローカルサーバがクラスタメンバーではありません。
ARMERR_OFF_CLUSTER	ローカルサーバがクラスタとして動作していません。
ARMERR_RESOURCE_NOT_FOUND	列挙可能なリソースが存在しません。
ARMERR_CANCELLED	コールバック関数から列挙の中止が返却されました。

説明

本関数は、ローカルサーバが所属するクラスタ内で定義されているリソースを列挙します。本関数は、アプリケーション定義のコールバック関数 *ResourceEnumProc* を順次呼び出し、各リソースの識別子を渡します。本関数は、列挙対象のリソースをすべて列挙するか、またはコールバック関数が FALSE を返却するまで、列挙を続けます。

本関数は、フェイルオーバーグループ名をキーにしてリソースの列挙を行なうことができます。パラメータ *IpszGroupName* にフェイルオーバーグループ名を指定した場合、指定されたフェイルオーバーグループに所属するリソースのみが列挙対象となります。

ローカルサーバがクラスタとして動作していない場合、リソースグループを列挙することはできません。

- * ローカルサーバがクラスタに登録されていない場合、本関数は FALSE を返却します。このとき拡張エラー情報で **ARMERR_NOT_CLUSTER_MEMBER** が返却されます。
- * ダウン後再起動状態等でローカルサーバがクラスタから一時的に切り離されている場合、本関数は FALSE を返却します。このとき拡張エラー情報で **ARMERR_OFF_CLUSTER** が返却されます。

リソース列挙対象の指定にクラスタ IP が含まれている場合、API を呼び出したアプリケーションの対応バージョンが Ver 5.0 以降であれば、リソース列挙の結果としてクラスタ IP リソースを返却します。アプリケーションの対応バージョンが Ver 4.x 以前の場合は、リソース列挙の結果からクラスタ IP リソースを除外します。

参照

3.1.2.9 ResourceEnumProc — リソース列挙用コールバック関数

3.1.2.9 ResourceEnumProc — リソース列挙用コールバック関数

概要

ArmEnumResource関数で列挙されるリソースの識別子を受け取るための、アプリケーション定義のコールバック関数です。

形式

```
BOOL CALLBACK ResourceEnumProc (  
    LPCTSTR lpszResourceName,  
    DWORD dwResourceType,  
    LPCTSTR lpszGroupName,  
    LPARAM lParam  
)
```

パラメータ

[i] <i>lpszResourceName</i>	リソースの識別子を示すNULLで終わる文字列へのポインタです。
[i] <i>dwResourceType</i>	<i>lpszResourceName</i> で示されるリソースのリソースタイプです。 ARM_RESTYPE_SWDISK 切替ディスク ARM_RESTYPE_VIP 仮想IPアドレス ARM_RESTYPE_VCOM 仮想コンピュータ名 ARM_RESTYPE_PRINTER プリンタ ARM_RESTYPE_LSW 回線切替装置 ARM_RESTYPE_NDR 切替ミラーディスク ARM_RESTYPE_ クラスタIP（負荷分散IP）
[i] <i>lpszGroupName</i>	リソースが所属するフェイルオーバーグループのグループ名を示すNULLで終わる文字列へのポインタです。
[i] <i>lParam</i>	ArmEnumResource関数で指定されたアプリケーション定義の値です。

戻り値

TRUE	列挙を継続する場合に返却します。
FALSE	列挙を中止する場合に返却します。

説明

ResourceEnumProc関数は、ArmEnumResource関数の実行結果として、クラスタ内で定義されているリソースの識別子を受け取るための、アプリケーション定義のコールバック関数です。リソースの列挙が完了するか、または本関数でFALSEを返却するまで、リソースの識別子が本関数に順次受け渡されます。アプリケーションは、ArmEnumResource関数実行時に本関数のアドレスを指定しなければなりません。

本関数に渡される *lpszResourceName*, *lpszGroupName*の指す文字列の内容は、本関数からリターンするまでの間でのみ有効です。文字列の内容を保持する必要がある場合には、本関数内で文字列を別の記憶領域へコピーしてください。

本関数の関数名“ResourceEnumProc”は、アプリケーション定義の関数名のプレースホルダです。

ArmInitializeApi関数で指定されたバージョンがVer4.0以前の場合は、切替ミラーディスクは従来の切替ディスクとして列挙されます。

参照

3.1.2.8 ArmEnumResource — リソースの列挙

3.1.2.10 ArmGetResourceInfo — リソース定義情報の取得

概要

指定したリソースのリソース定義情報を取得します。

形式

```
BOOL ArmGetResourceInfo (
    LPCTSTR lpszResourceName,
    LPARM_RESOURCE_INFO lpArmResourceInfo
)
```

パラメータ

- | | |
|------------------------------|--|
| [i] <i>lpszResourceName</i> | リソース定義情報の取得対象とするリソースの識別子を指定する、NULLで終わる文字列へのポインタです。 |
| [o] <i>lpArmResourceInfo</i> | リソース定義情報を受け取るARM_RESOURCE_INFO構造体へのポインタです。 |

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_NOT_ENOUGH_MEMORY	メモリのアロケートに失敗しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_NOT_CLUSTER_MEMBER	ローカルサーバがクラスタメンバではありません。
ARMERR_OFF_CLUSTER	ローカルサーバがクラスタとして動作していません。
ARMERR_RESOURCE_NOT_FOUND	指定したリソースが存在しません。

説明

本関数は、ローカルサーバが所属するクラスタ内で定義されている、*lpszResourceName*で指定されたリソースの定義情報を取得し、*lpArmResourceInfo*で指定されたバッファに格納します。

ローカルサーバがクラスタとして動作していない場合、リソースの定義情報を取得することはできません。

- * ローカルサーバがクラスタに登録されていない場合、本関数はFALSEを返却します。このとき拡張エラー情報でARMERR_NOT_CLUSTER_MEMBERが返却されます。
- * ダウン後再起動状態等でローカルサーバがクラスタから一時的に切り離されている場合、本関数はFALSEを返却します。このとき拡張エラー情報でARMERR_OFF_CLUSTERが返却されます。

参照

3.2.3 ARM_RESOURCE_INFO — リソース定義情報

3.1.2.11 ArmEnumLogicalService — 論理サービス名の列挙

概要

クラスタ内で定義されている論理サービス名を列挙します。

形式

```
BOOL ArmEnumLogicalService (
    LPCTSTR lpszGroupName,
    ARM_LOGSRV_ENUM_PROC LogicalServiceEnumProc,
    LPARAM lParam
)
```

パラメータ

- | | |
|-----------------------------------|--|
| [i] <i>lpszGroupName</i> | このパラメータにフェイルオーバーグループ名を指定すると、そのフェイルオーバーグループに所属する論理サービス名のみが列挙対象となります。NULLまたは空文字列を指定した場合は、クラスタ内の全論理サービス名が列挙対象となります。 |
| [i] <i>LogicalServiceEnumProc</i> | 論理サービス名を受け取る、アプリケーション定義のコールバック関数のアドレスです。(3.1.2.12 LogicalServiceEnumProc関数参照) |
| [i] <i>lParam</i> | コールバック関数 <i>LogicalServiceEnumProc</i> に渡される、アプリケーション定義の32ビット値です。 |

戻り値

- | | |
|-------|--|
| TRUE | 関数が正常終了しました。 |
| FALSE | 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。 |

拡張エラー情報

- | | |
|---------------------------|-------------------------------|
| ARMERR_NOT_INITIALIZED | CLUSTERPRO APIが初期化されていません。 |
| ARMERR_CANNOT_CONNECT | CLUSTERPROモニタに接続できません。 |
| ARMERR_COMM_ERROR | CLUSTERPROモニタとの通信でエラーが発生しました。 |
| ARMERR_NOT_ENOUGH_MEMORY | メモリのアロケートに失敗しました。 |
| ARMERR_INVALID_PARAMETER | パラメータの指定が不正です。 |
| ARMERR_NOT_CLUSTER_MEMBER | ローカルサーバがクラスタメンバではありません。 |
| ARMERR_OFF_CLUSTER | ローカルサーバがクラスタとして動作していません。 |
| ARMERR_GROUP_NOT_FOUND | 指定したフェイルオーバーグループが存在しません。 |
| ARMERR_NOT_FOUND | 列挙可能な論理サービス名が存在しません。 |
| ARMERR_CANCELLED | コールバック関数から列挙の中止が返却されました。 |

説明

本関数は、ローカルサーバが所属するクラスタ内で定義されている論理サービス名を列挙します。本関数は、アプリケーション定義のコールバック関数`LogicalServiceEnumProc`を順次呼び出し、論理サービス名を渡します。本関数は、列挙対象の論理サービス名をすべて列挙するか、またはコールバック関数がFALSEを返却するまで、列挙を続けます。

本関数は、フェイルオーバーグループ名をキーにして論理サービス名の列挙を行なうことができます。パラメータ `lpszGroupName` にフェイルオーバーグループ名を指定した場合、指定されたフェイルオーバーグループに所属する論理サービス名のみが列挙対象となります。

ローカルサーバがクラスタとして動作していない場合、論理サービス名を列挙することはできません。

- * ローカルサーバがクラスタに登録されていない場合、本関数はFALSEを返却します。このとき拡張エラー情報でARMERR_NOT_CLUSTER_MEMBERが返却されます。
- * ダウン後再起動状態等でローカルサーバがクラスタから一時的に切り離されている場合、本関数はFALSEを返却します。このとき拡張エラー情報でARMERR_OFF_CLUSTERが返却されます。

参照

3.1.2.12 LogicalServiceEnumProc — 論理サービス名列挙用コールバック関数

3.1.2.12 LogicalServiceEnumProc — 論理サービス名列挙用コールバック関数

概要

ArmEnumLogicalService関数で列挙される論理サービス名を受け取るための、アプリケーション定義のコールバック関数です。

形式

```
BOOL CALLBACK LogicalServiceEnumProc (  
    LPCTSTR lpszLogicalServiceName,  
    LPCTSTR lpszGroupName,  
    LPARAM lParam  
)
```

パラメータ

- | | |
|-----------------------------------|---|
| [i] <i>lpszLogicalServiceName</i> | 論理サービス名を示すNULLで終わる文字列へのポインタです。 |
| [i] <i>lpszGroupName</i> | 論理サービス名が所属するフェイルオーバーグループのグループ名を示すNULLで終わる文字列へのポインタです。 |
| [i] <i>lParam</i> | ArmEnumLogicalService関数で指定されたアプリケーション定義の値です。 |

戻り値

- | | |
|-------|------------------|
| TRUE | 列挙を継続する場合に返却します。 |
| FALSE | 列挙を中止する場合に返却します。 |

説明

LogicalServiceEnumProc関数は、ArmEnumLogicalService関数の実行結果として、クラスタ内で定義されている論理サービス名を受け取るための、アプリケーション定義のコールバック関数です。論理サービス名の列挙が完了するか、または本関数でFALSEを返却するまで、論理サービス名が本関数に順次受け渡されます。アプリケーションは、ArmEnumLogicalService関数実行時に本関数のアドレスを指定しなければなりません。

本関数に渡される *lpszLogicalServiceName*, *lpszGroupName* の指す文字列の内容は、本関数からリターンするまでの間でのみ有効です。文字列の内容を保持する必要がある場合には、本関数内で文字列を別の記憶領域へコピーしてください。

本関数の関数名 “LogicalServiceEnumProc” は、アプリケーション定義の関数名のプレースホルダです。

参照

3.1.2.11 ArmEnumLogicalService — 論理サービス名の列挙

3.1.3 クラスタ状態の取得/監視

クラスタシステムの動作状態の取得、クラスタイベントの監視を行なうAPIを提供します。

- | | | |
|------------------------|---|--------------------------|
| * ArmGetServerStatus | — | サーバの動作状態を取得します。 |
| * ArmGetGroupStatus | — | フェイルオーバーグループの動作状態を取得します。 |
| * ArmGetResourceStatus | — | リソースの動作状態を取得します。 |
| * ArmStartEventWatch | — | クラスタイベントの監視を開始します。 |
| * ArmCancelEventWatch | — | クラスタイベントの監視を中止します。 |
| * ArmGetEventInfo | — | クラスタイベント情報を取得します。 |

3.1.3.1 ArmGetServerStatus — サーバ動作状態の取得

概要

指定したサーバの現在の動作状態を取得します。

形式

```
BOOL ArmGetServerStatus (
    LPCTSTR lpszServerName,
    LPDWORD lpdwStatus
)
```

パラメータ

[i] <i>lpszServerName</i>	動作状態の取得対象とするサーバのコンピュータ名を指定する、NULLで終わる文字列へのポインタです。ローカルサーバを指定する場合には、定数ARM_LOCAL_SERVERを指定することができます。
[o] <i>lpdwStatus</i>	指定したサーバの動作状態を受け取る変数へのポインタです。以下のサーバ動作状態が返却されます。 ARM_STATUS_SINGLE_SERVER クラスタに未登録のシングルサーバとして動作しています。 ARM_STATUS_CLUSTER クラスタとして正常に動作しています。 ARM_STATUS_OFF_CLUSTER クラスタには登録されているが、クラスタから切り離されて動作しています。（ダウン後再起動状態） ARM_STATUS_STOPPED 停止状態です。 ARM_STATUS_DOWN ダウン状態です。 ARM_STATUS_RECOVERING failover処理中です。 上記の状態に次の状態が組み合わせられます。 ARM_STATUS_BACKUP 他サーバからfailoverしてきたリソースがあります。 ARM_STATUS_FAIL 他サーバへfailoverしたリソースがあります。 ARM_STATUS_MISS failover処理の失敗したリソースがあります。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。

ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_OFF_CLUSTER	ローカルサーバがクラスタとして動作していません。
ARMERR_SERVER_NOT_FOUND	指定したサーバが存在しません。

説明

本関数は、ローカルサーバが所属するクラスタを構成している、*lpdwStatus*で指定されたサーバの動作状態を取得し、*lpArmServerStatus*の指す変数に格納します。

ローカルサーバがクラスタとして動作していない場合、他サーバの動作状態を取得することはできません。ダウン後再起動状態等でローカルサーバがクラスタから一時的に切り離されている場合、本関数はFALSEを返却します。このとき拡張エラー情報でARMERR_OFF_CLUSTERが返却されます。

3.1.3.2 ArmGetGroupStatus — フェイルオーバーグループ動作状態の取得

概要

指定したフェイルオーバーグループの現在の動作状態を取得します。

形式

```
BOOL ArmGetGroupStatus (
    LPCTSTR lpszGroupName,
    LPARM_GROUP_STATUS lpArmGroupStatus,
)
```

パラメータ

- [i] *lpszGroupName* 動作状態の取得対象とするフェイルオーバーグループのグループ名を指定する、NULLで終わる文字列へのポインタです。
- [o] *lpArmGroupStatus* 指定したフェイルオーバーグループの動作状態を受け取る ARM_GROUP_STATUS構造体へのポインタです。

戻り値

TRUE 関数が正常終了しました。
FALSE 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_NOT_CLUSTER_MEMBER	ローカルサーバがクラスタメンバーではありません。
ARMERR_OFF_CLUSTER	ローカルサーバがクラスタとして動作していません。
ARMERR_GROUP_NOT_FOUND	指定したフェイルオーバーグループが存在しません。

説明

本関数は、ローカルサーバが所属するクラスタ内で定義されている、*lpszGroupName*で指定されたフェイルオーバーグループの動作状態を取得し、*lpArmGroupStatus*で指定されたバッファに格納します。

ローカルサーバがクラスタとして動作していない場合、フェイルオーバーグループの動作状態を取得することはできません。

- * ローカルサーバがクラスタに登録されていない場合、本関数はFALSEを返却します。このとき拡張エラー情報でARMERR_NOT_CLUSTER_MEMBERが返却されます。
- * ダウン後再起動状態等でローカルサーバがクラスタから一時的に切り離されている場合、本関数はFALSEを返却します。このとき拡張エラー情報でARMERR_OFF_CLUSTERが返却されます。

*

参照

3.2.4 ARM_GROUP_STATUS — フェイルオーバーグループ動作状態

3.1.3.3 ArmGetResourceStatus — リソース動作状態の取得

概要

指定したリソースの現在の動作状態を取得します。

形式

```
BOOL ArmGetResourceStatus (
    LPCTSTR lpszResourceName,
    LPARM_RESOURCE_STATUS lpArmResourceStatus,
)
```

パラメータ

- [i] *lpszResourceName* 動作状態の取得対象とするリソースの識別子を指定する、NULLで終わる文字列へのポインタです。
- [o] *lpArmResourceStatus* 指定したリソースの動作状態を受け取る **ARM_RESOURCE_STATUS**構造体へのポインタです。

戻り値

TRUE 関数が正常終了しました。
FALSE 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_NOT_CLUSTER_MEMBER	ローカルサーバがクラスタメンバーではありません。
ARMERR_OFF_CLUSTER	ローカルサーバがクラスタとして動作していません。
ARMERR_RESOURCE_NOT_FOUND	指定したリソースが存在しません。

説明

本関数は、ローカルサーバが所属するクラスタ内で定義されている、*lpszResourceName*で指定されたリソースの動作状態を取得し、*lpArmResourceStatus*で指定されたバッファに格納します。

ローカルサーバがクラスタとして動作していない場合、リソースの動作状態を取得することはできません。

- * ローカルサーバがクラスタに登録されていない場合、本関数はFALSEを返却します。このとき拡張エラー情報で**ARMERR_NOT_CLUSTER_MEMBER**が返却されます。
- * ダウン後再起動状態等でローカルサーバがクラスタから一時的に切り離されている場合、本関数はFALSEを返却します。このとき拡張エラー情報で**ARMERR_OFF_CLUSTER**が返却されます。

参照

3.2.5 ARM_RESOURCE_STATUS — リソース動作状態

3.1.3.4 ArmStartEventWatch — クラスタイメント監視の開始

概要

CLUSTERPROモニタから通知されるクラスタイメントの監視を開始します。

形式

```
BOOL ArmServerEventWatch (  
    DWORD   dwArmApiHandle,  
    HANDLE  hEvent  
)
```

パラメータ

- [i] *dwArmApiHandle* ArmInitializeApiで返却されたCLUSTERPRO APIハンドルを指定します。
- [i] *hEvent* イベントオブジェクトハンドルを指定します。このイベントオブジェクトは、CLUSTERPROモニタからクラスタイメントが通知されたときにシグナル状態に設定されます。アプリケーションは、Win32 APIのWaitForSingleObject()等を用いてこのイベントオブジェクトの状態をチェックすることにより、クラスタイメント通知を監視することができます。

戻り値

- TRUE 関数が正常終了しました。
- FALSE 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_INVALID_HANDLE	指定されたCLUSTERPRO APIハンドルが不正です。
ARMERR_ALREADY_REGISTERED	指定したイベントオブジェクトは既に登録されています。

説明

本関数は、イベントオブジェクトを登録し、CLUSTERPROモニタからのクラスタイメント通知の監視を開始します。本関数で登録されたイベントオブジェクトは、クラスタイメント通知を受信したときにシグナル状態に設定されます。アプリケーションは、Win32 APIのWaitForSingleObject()等を用いてこのイベントオブジェクトの状態をチェックすることにより、クラスタイメント通知を監視することができます。

本関数により登録したイベントオブジェクトがシグナル状態に設定された場合には、ArmGetEventInfo関数を実行して、クラスタイメント情報を引き取ってください。クラスタイメント情報を引き取らないままにしておくと、以後に発生する別のクラスタイメント通知を受信できなくなります。

イベントオブジェクトは、ArmGetEventInfo関数を実行することにより、自動的に非シグナル状態に設定されます。

クラスタイベントの監視が不要になった場合には、ArmCancelEventWatch関数を実行して、クラスタイベント監視を中止してください。

本関数は、同一プロセス内で別のイベントオブジェクトを指定して複数回実行することが可能です。

ArmTerminateApiを実行してCLUSTERPRO APIの終了処理を行なうと、そのCLUSTERPRO APIハンドルに関連するクラスタイベント監視はすべてキャンセルされます。

参照

- 3.1.3.5 ArmCancelEventWatch — クラスタイベント監視の中止
- 3.1.3.6 ArmGetEventInfo — クラスタイベント情報の取得

3.1.3.5 ArmCancelEventWatch — クラスタイメント監視の中止

概要

ArmStartEventWatch関数により開始したクラスタイメント通知の監視を中止します。

形式

```
BOOL ArmCancelEventWatch (  
    HANDLE hEvent  
)
```

パラメータ

[i] *hEvent* ArmStartEventWatch関数で登録したイベントオブジェクトのハンドルを指定します。

戻り値

TRUE 関数が正常終了しました。
FALSE 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_NOT_REGISTERED	指定したイベントオブジェクトは登録されていません。

説明

本関数は、ArmStartEventWatch関数により開始したクラスタイメント通知の監視を中止し、イベントオブジェクトの登録を取り消します。本関数の実行後は、クラスタイメント通知の受信による指定イベントオブジェクトの状態の設定は行われません。また、本関数を実行すると、指定イベントオブジェクトに対する未取得のクラスタイメント情報は廃棄されます。

本関数は、指定したイベントオブジェクトによるクラスタイメント監視のみを中止します。複数のイベントオブジェクトを用いて、ArmStartEventWatch関数を複数回実行していた場合、他のイベントオブジェクトによるクラスタイメントの監視は継続します。すべての監視を中止する場合には、それぞれのイベントオブジェクトを指定して、本関数を必要な回数だけ実行してください。

参照

3.1.3.4 ArmStartEventWatch — クラスタイメント監視の開始

3.1.3.6 ArmGetEventInfo — クラスタイメント情報の取得

概要

クラスタイメント情報を取得します。

形式

```
BOOL ArmGetEventInfo (
    HANDLE hEvent,
    LPARM_EVENT_INFO lpArmEventInfo,
    LPDWORD lpcbArmEventInfo,
    LPBOOL lpfRemain
)
```

パラメータ

[i] <i>hEvent</i>	ArmStartEventWatch関数により登録した、クラスタイメント監視用イベントオブジェクトのハンドルを指定します。
[o] <i>lpArmEventInfo</i>	クラスタイメント情報を受け取る、ARM_EVENT_INFO構造体へのポインタです。
[io]	<i>lpcbArmEventInfo</i> <i>lpArmEventInfo</i> で指定されたバッファのサイズを指定する変数へのポインタです。関数のリターン時には、バッファに格納されたデータのバイト数がセットされます。 指定バッファサイズが小さすぎた場合には、必要なバイト数がセットされ、拡張エラー情報でARMERR_BUFFER_TOO_SMALLが返却されます。
[o] <i>lpfRemain</i>	まだ取得されていない他のクラスタイメント情報が残っているか否かを受け取る変数へのポインタです。以下の値がセットされます。 TRUE 取得されていないクラスタイメント情報が残っている FALSE 取得されていないクラスタイメント情報はない

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_NOT_REGISTERED	指定したイベントオブジェクトは登録されていません。
ARMERR_BUFFER_TOO_SMALL	データ返却用バッファが小さすぎます。
ARMERR_NOT_FOUND	取得できるクラスタイメント情報がありません。

説明

本関数は、ArmStartEventWatch関数により登録されたイベントオブジェクトがシグナル状態に設定された時に、クラスタイイベント情報を取得するために使用されます。

本関数は、クラスタイイベント情報を1つずつ取得します。本関数実行時に、2つ以上のクラスタイイベント情報が存在する場合には、他のクラスタイイベント情報が残っていることを示すために、パラメータ *lpfRemain* に **TRUE** がセットされます。*lpfRemain* で **TRUE** が返却された場合には、**FALSE** になるまで、繰り返し本関数を実行してください。クラスタイイベント情報を取得せずに残したままにしておくと、クラスタイイベントキューがオーバーフローする場合があります。

本関数を実行すると、イベントオブジェクトは自動的に非シグナル状態に設定されます。

参照

- 3.1.3.4 ArmStartEventWatch — クラスタイイベント監視の開始
- 3.2.6 ARM_EVENT_INFO — クラスタイイベント情報

3.1.4 レジストリの操作

クラスタ内で同期のとれたレジストリの操作を行うためのAPIを提供します。
レジストリ操作APIのインタフェースは、Ver3.0から変更ありません。

- | | | |
|---------------------|---------------------------|-------------|
| * ArmRegOpenKey | — レジストリキーをオープンします。 | (4.1.4.1参照) |
| * ArmRegCreateKey | — レジストリキー作成/オープンします。 | (4.1.4.2参照) |
| * ArmRegDeleteKey | — レジストリキーを削除します。 | (4.1.4.3参照) |
| * ArmRegFlushKey | — オープン中のキーをレジストリに書き込みます。 | (4.1.4.4参照) |
| * ArmRegCloseKey | — レジストリキーをクローズします。 | (4.1.4.5参照) |
| * ArmRegSetValue | — レジストリキーに値を設定します。 | (4.1.4.6参照) |
| * ArmRegDeleteValue | — レジストリキーに設定されている値を削除します。 | (4.1.4.7参照) |

- ※ レジストリキー/値の列挙や値の問い合わせ等にはWin32 APIを使用してください。
ただし、**ArmRegOpenKey関数およびArmRegCreateKey関数で返却されたオープンキーハンドルをWin32 APIで使用することはできません**。別途Win32 APIを使用してオープンキーハンドルを取得してください。
- ※ 上記APIによるレジストリの更新内容は、クラスタ内の他のサーバへ即時反映されます。ただし、更新内容が反映されるのは、クラスタ内の全サーバがクラスタ動作している場合のみです。クラスタ内にクラスタ動作していないサーバが1台でも存在した場合には、レジストリ更新操作は失敗します。
- ※ レジストリ同期の方法として、上記APIを使用する他に、同期させるレジストリキーをフェイルオーバーグループに登録する方法があります。この方法では、登録されたレジストリキー配下に対するWin32APIを使用したレジストリの更新が、フェイルオーバーのタイミングでフェイルオーバー先へ反映されます。

3.2 構造体

* ARM_SERVER_INFO	—	サーバ定義情報
* ARM_GROUP_INFO	—	フェイルオーバーグループ定義情報
* ARM_RESOURCE_INFO	—	リソース定義情報
* ARM_GROUP_STATUS	—	フェイルオーバーグループ動作状態
* ARM_RESOURCE_STATUS	—	リソース動作状態
* ARM_EVENT_INFO	—	クラスタイメント情報

3.2.1 ARM_SERVER_INFO — サーバ定義情報

概要

ARM_SERVER_INFO構造体は、サーバ定義情報を格納します。

形式

```
typedef struct _ARM_SERVER_INFO {  
    TCHAR    szOSVersion [MAX_VER_LEN];  
    TCHAR    szARMVersion [MAX_VER_LEN];  
    TCHAR    szHomeDir [MAX_PATH_LEN];  
} ARM_SERVER_INFO, *LARM_SERVER_INFO;
```

メンバ

szOSVersion	OS名およびバージョンを表すNULLで終わる文字列です。
szARMVersion	CLUSTERPROのバージョンを表すNULLで終わる文字列です。
szHomeDir	CLUSTERPROのインストールディレクトリのパスを表すNULLで終わる文字列です。

3.2.2 ARM_GROUP_INFO — フェイルオーバーグループ定義情報

概要

ARM_GROUP_INFO構造体は、フェイルオーバーグループ定義情報を格納します。

形式

```
typedef struct _ARM_GROUP_INFO {  
    DWORD      nServer;  
    LPTSTR      *ServerList;  
    DWORD      nRegistry;  
    LPTSTR      *RegistryList;  
    BYTE        bData [0];  
} ARM_GROUP_INFO, *LPARM_GROUP_INFO;
```

メンバ

nServer	フェイルオーバーグループが移動可能なサーバ数です。
ServerList	フェイルオーバーグループが移動可能なサーバのリスト（サーバ名を表す文字列へのポインタ配列）です。サーバ名は、フェイルオーバープライオリティの高い順に並べられています。最初(ServerList[0])のサーバはプライマリサーバとなります。
nRegistry	フェイルオーバー時に更新内容を同期させるレジストリキーの個数です。
RegistryList	フェイルオーバー時に更新内容を同期させるレジストリキーのリスト（レジストリキーを表す文字列へのポインタ配列）です。
bData	サーバリスト、レジストリキーリストの実データを格納するための領域です。

3.2.3 ARM_RESOURCE_INFO — リソース定義情報

概要

ARM_RESOURCE_INFO構造体は、リソース定義情報を格納します。

形式

```
typedef struct _ARM_RESOURCE_INFO {
    DWORD      dwType;
    TCHAR      szGroupName [MAX_GROUPNAME_LEN];
    union {
        ARM_RESOURCE_INFO_DISK          Disk;
        ARM_RESOURCE_INFO_VIP           VIP;
        ARM_RESOURCE_INFO_VCOM          VCom;
        ARM_RESOURCE_INFO_PRINTER       Printer;
        ARM_RESOURCE_INFO_LSW           LSW;
        ARM_RESOURCE_INFO_NDR           NDR;
        ARM_RESOURCE_INFO_HF            HF;
    } Parameter;
} ARM_RESOURCE_INFO, *LPARM_RESOURCE_INFO;
```

メンバ

dwType	リソースタイプです。 ARM_RESTYPE_SWDISK 切替ディスク ARM_RESTYPE_VIP 仮想IPアドレス ARM_RESTYPE_VCOM 仮想コンピュータ名 ARM_RESTYPE_PRINTER プリンタ ARM_RESTYPE_LSW 回線切替装置 ARM_RESTYPE_NDR 切替ミラーディスク ARM_RESTYPE_HF クラスタIP（負荷分散IP）
szGroupName	リソースが所属するフェイルオーバーグループのグループ名を表す NULLで終わる文字列です。
Parameter	リソースタイプ別の固有パラメータを格納する共用体です。
— Disk	— 切替ディスク固有パラメータ
— VIP	— 仮想IPアドレス固有パラメータ
— VCom	— 仮想コンピュータ名固有パラメータ
— Printer	— プリンタ固有パラメータ
— LSW	— 回線切替装置固有パラメータ
— NDR	— 切替ミラーディスク固有パラメータ
— HF	— クラスタIP固有パラメータ

3.2.3.1 ARM_RESOURCE_INFO_DISK — 切替ディスク固有パラメータ

概要

ARM_RESOURCE_INFO_DISK構造体は、切替ディスク固有のパラメータを格納します。

形式

```
typedef struct _ARM_RESOURCE_INFO_DISK {  
    TCHAR    szLetter [MAX_LETTER_LEN];  
} ARM_RESOURCE_INFO_DISK, *LPARM_RESOURCE_INFO_DISK;
```

メンバ

szLetter	切替ディスクパーティションに割り当てられたドライブ文字を表す文字列（“A:”～“Z:”）です。
----------	---

3.2.3.2 ARM_RESOURCE_INFO_VIP — 仮想 IP アドレス固有パラメータ

概要

ARM_RESOURCE_INFO_VIP構造体は、仮想IPアドレス固有のパラメータを格納します。

形式

```
typedef struct _ARM_RESOURCE_INFO_VIP {  
    DWORD    dwAddress;  
} ARM_RESOURCE_INFO_VIP, *LPARM_RESOURCE_INFO_VIP;
```

メンバ

dwAddress	仮想IPアドレス（ネットワークバイトオーダー）です。
-----------	----------------------------

3.2.3.3 ARM_RESOURCE_INFO_VCOM — 仮想コンピュータ名固有パラメータ

概要

ARM_RESOURCE_INFO_VCOM構造体は、仮想コンピュータ名固有のパラメータを格納します。

形式

```
typedef struct _ARM_RESOURCE_INFO_VCOM {  
    TCHAR    szCompName [MAX_VCOMNAME_LEN];  
} ARM_RESOURCE_INFO_VCOM, *LPARM_RESOURCE_INFO_VCOM;
```

メンバ

szCompName	仮想コンピュータ名を表すNULLで終わる文字列です。
------------	----------------------------

3. 2. 3. 4 ARM_RESOURCE_INFO_PRINTER — プリンタ固有パラメータ

概要

ARM_RESOURCE_INFO_PRINTER構造体は、プリンタ固有のパラメータを格納します。

形式

```
typedef struct _ARM_RESOURCE_INFO_PRINTER {  
    TCHAR    szPrinterName [MAX_PRNNAME_LEN];  
    TCHAR    szPortName [MAX_PRNPORT_LEN];  
    TCHAR    szShareName [MAX_PRNSHARE_LEN];  
    TCHAR    szSpoolDirectory [MAX_SPOOLPATH_LEN];  
} ARM_RESOURCE_INFO_PRINTER, *LPARM_RESOURCE_INFO_PRINTER;
```

メンバ

szPrinterName	プリンタ名を表すNULLで終わる文字列です。
szPortName	プリンタが接続されているポート名を表すNULLで終わる文字列です。
szShareName	プリンタのシェア名を表すNULLで終わる文字列です。
szSpoolDirectory	スプールディレクトリのパス名を表すNULLで終わる文字列です。

3. 2. 3. 5 ARM_RESOURCE_INFO_LSW — 回線切替装置固有パラメータ

概要

ARM_RESOURCE_INFO_LSW構造体は、回線切替装置固有のパラメータを格納します。

形式

```
typedef struct _ARM_RESOURCE_INFO_LSW {  
    TCHAR    szDeviceName [MAX_LSWDEVICE_LEN];  
    TCHAR    szPortName [MAX_LSWPORT_LEN];  
    DWORD    dwUnitNo;  
    DWORD    dwLineNo;  
} ARM_RESOURCE_INFO_LSW, *LPARM_RESOURCE_INFO_LSW;
```

メンバ

szDeviceName	回線切替装置の識別名を表すNULLで終わる文字列です。
szPortName	回線切替装置が接続されているシリアルポート名を表すNULLで終わる文字列（“COM1”，“COM2”）です。
dwUnitNo	ユニット番号（0～15）です。
dwLineNo	ライン番号（1～4）です。

3. 2. 3. 6 ARM_RESOURCE_INFO_NDR— 切替ミラーディスク固有パラメータ

概要

ARM_RESOURCE_INFO_NDR 構造体は、切替ミラーディスク固有のパラメータを格納します。

形式

```
typedef struct _ARM_RESOURCE_INFO_NDR {  
    TCHAR      szName [MAX_NDRNAME_LEN];  
    BYTE       bDrive [MAX_DRV_NUM];  
} ARM_RESOURCE_INFO_NDR, *LPARM_RESOURCE_INFO_NDR;
```

メンバ

szName	切替ミラーディスクに付けられた一意名を表す NULL で終わる文字列です。
bDrive	切替ミラーディスクに割り当てられたドライブ番号を示す配列です。bDrive[0]～bDrive[25]がドライブA～ドライブZを表し、この切替ディスクに割り当てられているドライブに対しては1、それ以外のドライブに対しては0がセットされます。

3. 2. 3. 7 ARM_RESOURCE_INFO_HFー クラスタ IP 固有パラメータ

概要

ARM_RESOURCE_INFO_HF 構造体は、クラスタ IP（負荷分散 IP）固有のパラメータを格納します。

形式

```
typedef struct _ARM_RESOURCE_INFO_HF {  
    DWORD    dwIPAddress;  
    DWORD    dwReserved1;  
    DWORD    dwReserved2;  
} ARM_RESOURCE_INFO_HF, *LPARM_RESOURCE_INFO_HF;
```

メンバ

dwIPAddress	クラスタ IP アドレス（ネットワークバイトオーダー）です。
dwReserved1	未使用。将来の拡張のために予約しています。
dwReserved2	未使用。将来の拡張のために予約しています。

3.2.4 ARM_GROUP_STATUS — フェイルオーバーグループ動作状態

概要

ARM_GROUP_STATUS構造体は、フェイルオーバーグループの動作状態を格納します。

形式

```
typedef struct _ARM_GROUP_STATUS {  
    DWORD      dwStatus;  
    TCHAR      szCurrentServer [MAX_SERVERNAME_LEN];  
} ARM_GROUP_STATUS, *LPARM_GROUP_STATUS;
```

メンバ

dwStatus	フェイルオーバーグループの動作状態です。以下の値の組み合わせになります。 ARM_GROUP_STATUS_ONLINE — 活性状態です。リソースの接続はすべて成功しています。 ARM_GROUP_STATUS_ONLINE_FAIL — 活性状態ですが、接続に失敗したリソースがあります。 ARM_GROUP_STATUS_OFFLINE — 非活性状態です。 ARM_GROUP_STATUS_INPROGRESS — リカバリ処理中です。
szCurrentServer	カレントサーバのサーバ名を表すNULLで終わる文字列です。

3.2.5 ARM_RESOURCE_STATUS — リソース動作状態

概要

ARM_RESOURCE_STATUS構造体は、リソースの動作状態を格納します。

形式

```
typedef struct _ARM_RESOURCE_STATUS {  
    DWORD      dwStatus;  
    TCHAR      szCurrentServer [MAX_SERVERNAME_LEN];  
} ARM_RESOURCE_STATUS, *LPARM_RESOURCE_STATUS ;
```

メンバ

dwStatus	リソースの動作状態です。 ARM_RESOURCE_STATUS_ONLINE — 活性状態。 ARM_RESOURCE_STATUS_OFFLINE — 非活性状態。
szCurrentServer	カレントサーバのサーバ名を表すNULLで終わる文字列です。

3.2.6 ARM_EVENT_INFO — クラスタイイベント情報

概要

ARM_EVENT_INFO構造体は、クラスタイイベント情報を格納します。

形式

```
typedef struct _ARM_EVENT_INFO {  
    DWORD    dwEvent;  
    DWORD    dwEventId;  
    TCHAR    szSourceName [MAX_EVENTSRCSRCNAME_LEN];  
    TCHAR    szSrcServer[MAX_SERVERNAME_LEN];  
    TCHAR    szDstServer[MAX_SERVERNAME_LEN];  
} ARM_EVENT_INFO, *LPARM_EVENT_INFO;
```

メンバ

dwEvent	イベントコードです。
dwEventId	イベントの発生順に付加される番号です。
szSourceName	イベントの発生元を表すNULLで終わる文字列です。イベントコードの種類に応じて、サーバ/フェイルオーバーグループ/リソースの名前が格納されます。
szSrcServer	移動元サーバ名を表すNULLで終わる文字列です。このメンバは、イベントコードの種類により使用されない場合があります。
szDstServer	移動先サーバ名を表すNULLで終わる文字列です。このメンバは、イベントコードの種類により使用されない場合があります。

イベントコードdwEventにセットされるクラスタイイベントは、以下のとおりです。

ARM_EVENT_SHUTDOWN	サーバがシャットダウンしました。
ARM_EVENT_DOWN	サーバ異常が発生しました。
ARM_EVENT_RETURN_TO_CLUSTER	サーバがクラスタへ復帰しました。
ARM_EVENT_FAILOVER_START	フェイルオーバー処理を開始しました。
ARM_EVENT_FAILOVER_END	フェイルオーバー処理が完了しました。
ARM_EVENT_GROUP_START	フェイルオーバーグループを起動しました。
ARM_EVENT_GROUP_STOP	フェイルオーバーグループを停止しました。
ARM_EVENT_GROUP_FAILOVER_START	フェイルオーバーグループのフェイルオーバー処理を開始しました。
ARM_EVENT_GROUP_FAILOVER_END	フェイルオーバーグループのフェイルオーバー処理が完了しました。
ARM_EVENT_GROUP_FAILOVER_FAILURE	フェイルオーバーグループのフェイルオーバー処理が失敗しました。
ARM_EVENT_GROUP_MOVE_START	フェイルオーバーグループの移動を開始しました。

ARM_EVENT_GROUP_MOVE_END	フェイルオーバーグループの移動が完了しました。
ARM_EVENT_GROUP_MOVE_FAILURE	フェイルオーバーグループの移動が失敗しました。
ARM_EVENT_LSW_SWITCHED	回線切替装置で、回線の切り替えが発生しました。

イベントコードによりメンバszSourceName, szSrcServer, szDstServerに格納される情報は、以下のようになります。

イベントコード	szSourceName	szSrcServer	szDstServer
ARM_EVENT_SHUTDOWN	シャットダウンしたサーバ	—	—
ARM_EVENT_DOWN	ダウンしたサーバ	—	—
ARM_EVENT_RETURN_TO_CLUSTER	復帰したサーバ	—	—
ARM_EVENT_FAILOVER_START ARM_EVENT_FAILOVER_END	フェイルオーバー先サーバ	—	—
ARM_EVENT_GROUP_START ARM_EVENT_GROUP_STOP	フェイルオーバーグループ	カレントサーバ	—
ARM_EVENT_GROUP_FAILOVER_START ARM_EVENT_GROUP_FAILOVER_END ARM_EVENT_GROUP_FAILOVER_FAILURE	フェイルオーバーグループ	フェイルオーバー元サーバ	フェイルオーバー先サーバ
ARM_EVENT_GROUP_MOVE_START ARM_EVENT_GROUP_MOVE_END ARM_EVENT_GROUP_MOVE_FAILURE	フェイルオーバーグループ	移動元サーバ	移動先サーバ
ARM_EVENT_LSW_SWITCHED	回線リソース	—	—

3.3 クラスタイレント

クラスタ状態遷移時に発生するクラスタイレントは、以下のようになります。

クラスタ起動	(1) ARM_EVENT_RETURN_TO_CLUSTER ^{*1} (2) ARM_EVENT_GROUP_STOP [×ファイルオーバグループ数]
単体サーバ起動	(1) ARM_EVENT_RETURN_TO_CLUSTER ^{*1}
クラスタシャットダウン	(1) ARM_EVENT_SHUTDOWN (2) ARM_EVENT_GROUP_STOP [×ファイルオーバグループ数]
サーバシャットダウン	(1) ARM_EVENT_SHUTDOWN
サーバダウン	(1) ARM_EVENT_DOWN (2) ARM_EVENT_FAILOVER_START (3) ARM_EVENT_GROUP_FAILOVER_START / END [×ファイルオーバグループ数] (4) ARM_EVENT_FAILOVER_END
グループフェイルオーバ	(1) ARM_EVENT_GROUP_FAILOVER_START (2) ARM_EVENT_GROUP_FAILOVER_END
メンテナンスモード設定	(1) ARM_EVENT_DOWN (2) ARM_EVENT_FAILOVER_START (3) ARM_EVENT_GROUP_FAILOVER_START / END [×ファイルオーバグループ数] (4) ARM_EVENT_FAILOVER_END
クラスタ復帰（手動）	(1) ARM_EVENT_RETURN_TO_CLUSTER
グループ開始（手動）	(1) ARM_EVENT_GROUP_START
グループ停止（手動）	(1) ARM_EVENT_GROUP_STOP
グループ移動（手動）	(1) ARM_EVENT_GROUP_MOVE_START (2) ARM_EVENT_GROUP_MOVE_END (3) ARM_EVENT_GROUP_MOVE_FAILURE
回線移動（手動）	(1) ARM_EVENT_LSW_SWITCHED

^{*1} 自動復帰モードONの場合のみ発生するイベントです。

4 サーバAPI詳細 — Ver3.0互換 —

CLUSTERPRO Ver3.0用アプリケーションをCLUSTERPRO Ver6.0上で動作可能とするためのVer3.0互換APIを提供します。

ただし、Ver3.0互換APIは、Ver 3.0互換運用時にのみ使用可能とします。Ver3.0互換運用時以外でのVer3.0互換APIの動作は保証しません。

4.1 関数

API初期化／終了処理

- * ArmInitializeApi
- * ArmTerminateApi

- CLUSTERPRO APIの初期化処理を行います。
- CLUSTERPRO APIの終了処理を行います。

クラスタシステム情報の取得

- * ArmGetClusterName
- * ArmEnumServer
- * EnumServerProc
- * ArmGetServerInfo
- * ArmEnumResource
- * ArmEnumResource2
- * EnumResourceProc
- * ArmGetResourceInfo
- * ArmGetResourceInfo2
- * ArmEnumLogicalService
- * ArmGetLogicalServiceInfo
- * GetSrvInfoProc

- ローカルサーバが所属するクラスタのクラスタ名を取得します。
- クラスタに所属するサーバを列挙します。
- サーバ列挙用コールバック関数。
- クラスタに所属する指定サーバのサーバ定義情報を取得します。
- クラスタ内で定義されているリソースを列挙します (Ver 2.x互換)。
- クラスタ内で定義されているリソースを列挙します (Ver3.0)。
- リソース列挙用コールバック関数。
- クラスタ内で定義されている指定リソースのリソース定義情報を取得します (Ver 2.x互換)。
- クラスタ内で定義されている指定リソースのリソース定義情報を取得します (Ver3.0)。
- クラスタ内で定義されている論理サービス名を列挙します。
- 論理サービス名の定義情報を取得します。
- 論理サービス名の定義情報取得用コールバック関数。

クラスタ動作状態の取得／監視

- * ArmGetServerStatus
- * ArmWatchServer
- * ArmStartWatchServer
- * ArmCancelWatchServer
- * WatchServerProc
- * ArmRegisterWatchServerEvent
- * ArmCancelWatchServerEvent
- * ArmGetServerEvent
- * ArmCheckResourceOwner

- サーバの動作状態を取得します。
- サーバの動作状態を監視します (同期型)。
- サーバの動作状態を監視します (非同期型)。
- 非同期サーバ監視を中止します。
- 非同期サーバ監視用コールバック関数。
- サーバ監視用イベントオブジェクトを登録します。
- サーバ監視用イベントオブジェクトの登録を取り消します。
- サーバ状態変更通知を取得します。
- ローカルサーバがリソースの所有権を持つか確認します。

レジストリ操作

- * ArmRegOpenKey
- * ArmRegCreateKey
- * ArmRegDeleteKey
- * ArmRegFlushKey
- * ArmRegCloseKey
- * ArmRegSetValue
- * ArmRegDeleteValue

- レジストリキーをオープンします。
- レジストリキー作成/オープンします。
- レジストリキーを削除します。
- オープン中のキーをレジストリに書き込みます。
- レジストリキーをクローズします。
- レジストリキーに値を設定します。
- レジストリキーに設定されている値を削除します。

4.1.1 API の初期化/終了

CLUSTERPRO APIを使用するアプリケーションは、他のCLUSTERPRO API関数を使用する前に必ずArmInitializeApi関数を実行してCLUSTERPRO APIの初期化を行わなければなりません。また、ArmInitializeApi関数を実行した場合には、アプリケーションを終了する前にArmTerminateApi関数を実行して、CLUSTERPRO APIの終了処理を行わなければなりません。

- * ArmInitializeApi — CLUSTERPRO APIの初期化を行います。
- * ArmTerminateApi — CLUSTERPRO APIの終了処理を行います。

CLUSTERPRO APIの初期化関数ArmInitializeApi()は、プロセス内で同時に複数実行することはありません。CLUSTERPRO APIを使用する複数のモジュールが同一プロセスとして動作する場合には、**いずれか1つのモジュールで、1回だけ**ArmInitializeApi()を実行してください。

4.1.1.1 ArmInitializeApi — CLUSTERPRO API の初期化

概要

CLUSTERPRO APIの初期化を行います。

形式

BOOL ArmInitializeApi (VOID)

パラメータ

なし

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_CANNOT_LOAD_DLL	armapi.dllをロードできません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_NOT_ENOUGH_MEMORY	メモリのアロケートに失敗しました。
ARMERR_CANNOT_ALLOCATE_RESOURCE	OS資源のアロケートに失敗しました。
ARMERR_CANNOT_CREATE_THREAD	スレッドの作成に失敗しました。
ARMERR_ALREADY_INITIALIZED	CLUSTERPRO APIは既に初期化されています。

説明

本関数は、CLUSTERPRO APIの初期化を行います。CLUSTERPRO APIを使用するアプリケーションは、最初に本関数を実行して、CLUSTERPRO APIを初期化しなければなりません。

armapist.libをリンクして作成されたアプリケーションでは、CLUSTERPROがインストールされていない（armapi.dllをロードできない）場合、本関数はFALSEを返却し、拡張エラー情報でARMERR_CANNOT_LOAD_DLLが返却されます。

注意

ArmInitializeApi関数は、プロセス内で同時に複数実行することはできません。

参照

4.1.1.2 ArmTerminateApi — CLUSTERPRO APIの終了

4.1.1.2 ArmTerminateApi — CLUSTERPRO API の終了

概要

CLUSTERPRO APIの終了処理を行います。

形式

BOOL ArmTerminateApi (*VOID*)

パラメータ

なし

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。

説明

本関数は、CLUSTERPRO APIの終了処理を行います。 ArmInitializeApi関数を実行してCLUSTERPRO APIの初期化を行なったアプリケーションは、アプリケーションを終了する前に本関数を実行してCLUSTERPRO APIの終了処理を行わなければなりません。

本関数の実行後は、再度ArmInitializeApi関数を実行するまで、他のCLUSTERPRO API関数は使用できません。

参照

4.1.1.1 ArmInitializeApi — CLUSTERPRO APIの初期化

4.1.2 クラスタシステム定義情報の取得

クラスタシステムに関する定義情報を取得するためのAPIを提供します。

* ArmGetClusterName	— ローカルサーバが所属するクラスタのクラスタ名を取得します。
* ArmEnumServer	— クラスタに所属するサーバを列挙します。
* EnumServerProc	— サーバ列挙用コールバック関数。
* ArmGetServerInfo	— クラスタに所属する指定サーバのサーバ定義情報を取得します。
* ArmEnumResource	— クラスタ内で定義されているリソースを列挙します (Ver 2. x互換)。
* ArmEnumResource2	— クラスタ内で定義されているリソースを列挙します (Ver3. 0)。
* EnumResourceProc	— リソース列挙用コールバック関数。
* ArmGetResourceInfo	— クラスタ内で定義されている指定リソースのリソース定義情報を取得します (Ver 2. x互換)。
* ArmGetResourceInfo2	— クラスタ内で定義されている指定リソースのリソース定義情報を取得します (Ver3. 0)。
* ArmEnumLogicalService	— クラスタ内で定義されている論理サービス名を列挙します。
* ArmGetLogicalServiceInfo	— 論理サービス名の定義情報を取得します。
* GetSrvInfoProc	— 論理サービス名の定義情報取得用コールバック関数。

4.1.2.1 ArmGetClusterName — クラスタ名の取得

概要

ローカルサーバが所属するクラスタの名前を取得します。

形式

```
BOOL ArmGetClusterName (
    LPTSTR  lpszClusterName,
    LPDWORD lpcchClusterName
)
```

パラメータ

[o] *lpszClusterName* クラスタ名（終端のNULL文字を含む）を受け取るバッファへのポインタです。

[io] *lpcchClusterName* バッファ *lpszClusterName* のサイズを示す変数へのポインタです。

関数のリターン時には、バッファに格納された文字数（終端のNULL文字を含まない）がセットされます。また、指定バッファサイズが小さすぎた場合には、必要な文字数（終端のNULL文字を含む）がセットされ、拡張エラー情報で **ARMERR_BUFFER_TOO_SMALL** が返却されます。

（※ クラスタ名の最大長は **MAX_CLUSTERNAME_LEN** です）

戻り値

TRUE 関数が正常終了しました。
FALSE 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_REGISTRY_ACCESS_FAILED	レジストリのアクセスに失敗しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_BUFFER_TOO_SMALL	データ返却用バッファが小さすぎます。

説明

本関数は、ローカルサーバが所属するクラスタの名前をレジストリから取得し、*lpszClusterName* で指定されたバッファに格納します。

4.1.2.2 ArmEnumServer — サーバの列挙

概要

クラスタに所属するサーバを列挙します。

形式

```
BOOL ArmEnumServer (
    ENUMSERVERPROC  enumproc,
    LPARAM  lParam
)
```

パラメータ

- [i] *enumproc* クラスタに所属するサーバのコンピュータ名を受け取る、アプリケーション定義のコールバック関数のアドレスです。
(4.1.2.3 EnumServerProc関数参照)
- [i] *lParam* コールバック関数*enumproc*に渡される、アプリケーション定義の32ビット値です。

戻り値

- TRUE 関数が正常終了しました。
- FALSE 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_NOT_ENOUGH_MEMORY	メモリのアロケートに失敗しました。
ARMERR_REGISTRY_ACCESS_FAILED	レジストリのアクセスに失敗しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_CANCELLED	コールバック関数から列挙の中止が返却されました。

説明

本関数は、ローカルサーバの所属するクラスタを構成しているサーバを列挙します。本関数は、アプリケーション定義のコールバック関数*enumproc*を順次呼び出し、各サーバのコンピュータ名を渡します。本関数は、クラスタを構成しているサーバをすべて列挙するか、またはコールバック関数がFALSEを返却するまで、列挙を続けます。

参照

- 4.1.2.3 EnumServerProc — サーバ列挙用コールバック関数

4.1.2.3 EnumServerProc — サーバ列挙用コールバック関数

概要

ArmEnumServer関数で列挙されるサーバのコンピュータ名を受け取るための、アプリケーション定義のコールバック関数です。

形式

```
BOOL CALLBACK EnumServerProc (  
    LPCTSTR lpszServerName,  
    LPARAM lParam  
)
```

パラメータ

[i] *lpszServerName* クラスタに所属するサーバのコンピュータ名を示す、NULLで終わる文字列へのポインタです。

[i] *lParam* ArmEnumServer関数で指定されたアプリケーション定義の値です。

戻り値

TRUE 列挙を継続する場合に返却します。
FALSE 列挙を中止する場合に返却します。

説明

EnumServerProc関数は、ArmEnumServer関数の実行結果として、クラスタを構成するサーバのコンピュータ名を受け取るための、アプリケーション定義のコールバック関数です。サーバの列挙が完了するか、または本関数でFALSEを返却するまで、サーバのコンピュータ名が本関数に順次受け渡されます。アプリケーションは、ArmEnumServer関数実行時に本関数のアドレスを指定しなければなりません。

本関数に渡される *lpszClusterName*, *lpszServerName*の指す文字列の内容は、本関数をリターンするまでの間でのみ有効です。これらの文字列の内容を保持する必要がある場合には、本関数内で文字列の内容を別の記憶領域へコピーしてください。

本関数の関数名 “EnumServerProc” は、アプリケーション定義の関数名のプレースホルダです。

参照

4.1.2.2 ArmEnumServer — サーバの列挙

4.1.2.4 ArmGetServerInfo — サーバ定義情報の取得

概要

指定したサーバのサーバ定義情報を取得します。

形式

```
BOOL ArmGetServerInfo (
    LPCTSTR lpszServerName,
    LPARM_SERVER_INFO lpArmServerInfo
)
```

パラメータ

[i] <i>lpszServerName</i>	サーバ定義情報の取得対象とするサーバのコンピュータ名を指定する、NULLで終わる文字列へのポインタです。ローカルサーバを指定する場合には、定数ARM_LOCAL_SERVERを指定することができます。
[o] <i>lpArmServerInfo</i>	サーバ定義情報を受け取るARM_SERVER_INFO構造体へのポインタです。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_NOT_ENOUGH_MEMORY	メモリのアロケートに失敗しました。
ARMERR_REGISTRY_ACCESS_FAILED	レジストリのアクセスに失敗しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_SERVER_NOT_FOUND	指定サーバが存在しません。

説明

本関数は、レジストリ上に設定されているサーバ定義情報を取得し、*lpArmServerInfo*で指定されたバッファに格納します。サーバ定義情報の取得対象とするサーバ*lpszServerName*には、ローカルサーバおよび、ローカルサーバの所属するクラスタ内の任意のサーバを指定することができます。

参照

4.2.1 ARM_SERVER_INFO — サーバ定義情報

4.1.2.5 ArmEnumResource — リソースの列挙 (Ver 2.x 互換)

概要

クラスタ内で定義されているリソースを列挙します。

形式

```
BOOL ArmEnumResource (  
    DWORD dwResourceType,  
    ENUMRESOURCEPROC enumproc,  
    LPARAM lParam  
)
```

パラメータ

[i] <i>dwResourceType</i>	列挙対象とするリソースタイプを指定します。 ARM_RESTYPE_ALL 全リソースタイプ ARM_RESTYPE_SCRIPT スクリプト ARM_RESTYPE_DISK 共有ディスク ARM_RESTYPE_VLAN 仮想IPアドレス ARM_RESTYPE_PRINTER プリンタ ARM_RESTYPE_LSW 回線切替装置
[i] <i>enumproc</i>	リソース識別子を受け取る、アプリケーション定義のコールバック関数のアドレスです。 (4.1.2.7 EnumResourceProc関数参照)
[i] <i>lParam</i>	コールバック関数 <i>enumproc</i> に渡される、アプリケーション定義の32ビット値です。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_NOT_ENOUGH_MEMORY	メモリのアロケートに失敗しました。
ARMERR_REGISTRY_ACCESS_FAILED	レジストリのアクセスに失敗しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_RESOURCE_NOT_FOUND	定義されているリソースがありません。
ARMERR_CANCELLED	コールバック関数から列挙の中止が返却されました。

説明

本関数は、ローカルサーバが所属するクラスタ内で定義されているリソースを列挙します。本関数は、アプリケーション定義のコールバック関数`enumproc`を順次呼び出し、各リソースのリソース識別子を渡します。本関数は、`dwResourceType`で指定されたリソースタイプのリソースをすべて列挙するか、またはコールバック関数がFALSEを返却するまで、列挙を続けます。

参照

- 4. 1. 2. 6 ArmEnumResource2 — リソースの列挙 (Ver3.0)
- 4. 1. 2. 7 EnumResourceProc — リソース列挙用コールバック関数

4.1.2.6 ArmEnumResource2 — リソースの列挙 (Ver3.0)

概要

クラスタ内で定義されているリソースを列挙します。

形式

```
BOOL ArmEnumResource2 (
    DWORD   dwResourceType,
    ENUMRESOURCEPROC  enumproc,
    LPARAM  lParam
)
```

パラメータ

[i] <i>dwResourceType</i>	列挙対象とするリソースタイプを指定します。 ARM_RESTYPE_ALL 全リソースタイプ ARM_RESTYPE_SCRIPT スクリプト ARM_RESTYPE_VLAN 仮想IPアドレス ARM_RESTYPE_PRINTER プリンタ ARM_RESTYPE_LSW 回線切替装置 ARM_RESTYPE_SWDISK 切替ディスクパーティション ARM_RESTYPE_SHDISK 共有ディスクパーティション ARM_RESTYPE_NPDISK CLUSTERPRO内部処理用共有ディスクパーティション
[i] <i>enumproc</i>	リソース識別子を受け取る、アプリケーション定義のコールバック関数のアドレスです。 (4.1.2.7 EnumResourceProc関数参照)
[i] <i>lParam</i>	コールバック関数 <i>enumproc</i> に渡される、アプリケーション定義の32ビット値です。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_NOT_ENOUGH_MEMORY	メモリのアロケートに失敗しました。
ARMERR_REGISTRY_ACCESS_FAILED	レジストリのアクセスに失敗しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_RESOURCE_NOT_FOUND	定義されているリソースがありません。

ARMERR_CANCELLED

コールバック関数から列挙の中止が返却されました。

説明

本関数は、ローカルサーバが所属するクラスタ内で定義されているリソースを列挙します。本関数は、アプリケーション定義のコールバック関数 *enumproc* を順次呼び出し、各リソースのリソース識別子を渡します。本関数は、*dwResourceType* で指定されたリソースタイプのリソースをすべて列挙するか、またはコールバック関数が FALSE を返却するまで、列挙を続けます。

本関数は、切替ディスクパーティションに対して、新しいリソースタイプ **ARM_RESTYPE_SWDISK** を返却します。従来のリソース列挙関数 *ArmEnumResource* では、切替ディスクパーティションに対して、リソースタイプ **ARM_RESTYPE_DISK** を返却します。

参照

- 4.1.2.5 *ArmEnumResource* — リソースの列挙 (Ver 2.x 互換)
- 4.1.2.7 *EnumResourceProc* — リソース列挙用コールバック関数

4.1.2.7 EnumResourceProc — リソース列挙用コールバック関数

概要

ArmEnumResource関数で列挙されるリソースのリソース識別子を受け取るための、アプリケーション定義のコールバック関数です。

形式

```
BOOL CALLBACK EnumResourceProc (  
    LPCTSTR lpszResourceName,  
    DWORD dwResourceType,  
    LPARAM lParam  
)
```

パラメータ

[i] <i>lpszResourceName</i>	リソース識別子を示すNULLで終わる文字列へのポインタです。
[i] <i>dwResourceType</i>	<i>lpszResourceName</i> で示されるリソースのリソースタイプです。 ARM_RESTYPE_SCRIPT スクリプト ARM_RESTYPE_DISK 共有ディスク ARM_RESTYPE_VLAN 仮想IPアドレス ARM_RESTYPE_PRINTER プリンタ ARM_RESTYPE_LSW 回線切替装置
[i] <i>lParam</i>	ArmEnumResource関数で指定されたアプリケーション定義の値です。

戻り値

TRUE	列挙を継続する場合に返却します。
FALSE	列挙を中止する場合に返却します。

説明

EnumResourceProc関数は、ArmEnumResource関数の実行結果として、クラスタ内で定義されているリソースのリソース識別子を受け取るための、アプリケーション定義のコールバック関数です。リソースの列挙が完了するか、または本関数でFALSEを返却するまで、リソース識別子が本関数に順次受け渡されます。アプリケーションは、ArmEnumResource関数実行時に本関数のアドレスを指定しなければなりません。

本関数に渡される*lpszResourceName*の指す文字列の内容は、本関数からリターンするまでの間でのみ有効です。文字列の内容を保持する必要がある場合には、本関数内で文字列を別の記憶領域へコピーしてください。

本関数の関数名“EnumResourceProc”は、アプリケーション定義の関数名のプレースホルダです。

参照

- 4.1.2.5 ArmEnumResource — リソースの列挙 (Ver2.x 互換)
- 4.1.2.6 ArmEnumResource2 — リソースの列挙 (Ver3.0)

4.1.2.8 ArmGetResourceInfo — リソース定義情報の取得 (Ver 2.x 互換)

概要

指定したリソースのリソース定義情報を取得します。

形式

```
BOOL ArmGetResourceInfo (
    LPCTSTR lpszResourceName,
    LPARM_RESOURCE_INFO lpArmResourceInfo
)
```

パラメータ

- | | |
|------------------------------|--|
| [i] <i>lpszResourceName</i> | リソース定義情報の取得対象とするリソースのリソース識別子を指定する、NULLで終わる文字列へのポインタです。 |
| [o] <i>lpArmResourceInfo</i> | リソース定義情報を受け取るARM_RESOURCE_INFO構造体へのポインタです。 |

戻り値

- | | |
|-------|--|
| TRUE | 関数が正常終了しました。 |
| FALSE | 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。 |

拡張エラー情報

- | | |
|-------------------------------|-------------------------------|
| ARMERR_NOT_INITIALIZED | CLUSTERPRO APIが初期化されていません。 |
| ARMERR_CANNOT_CONNECT | CLUSTERPROモニタに接続できません。 |
| ARMERR_COMM_ERROR | CLUSTERPROモニタとの通信でエラーが発生しました。 |
| ARMERR_NOT_ENOUGH_MEMORY | メモリのアロケートに失敗しました。 |
| ARMERR_REGISTRY_ACCESS_FAILED | レジストリのアクセスに失敗しました。 |
| ARMERR_INVALID_PARAMETER | パラメータの指定が不正です。 |
| ARMERR_RESOURCE_NOT_FOUND | 指定リソースが存在しません。 |

説明

本関数は、ローカルサーバが所属するクラスタ内で定義されている、*lpszResourceName*で指定されたリソースの定義情報をレジストリから取得し、*lpArmResourceInfo*で指定されたバッファに格納します。

参照

- 4.1.2.9 ArmGetResourceInfo2 — リソース定義情報の取得 (Ver3.0)
- 4.2.2 ARM_RESOURCE_INFO — リソース定義情報

4.1.2.9 ArmGetResourceInfo2 — リソース定義情報の取得 (Ver3.0)

概要

指定したリソースのリソース定義情報を取得します。

形式

```
BOOL ArmGetResourceInfo2 (
    LPCTSTR lpszResourceName,
    LPARM_RESOURCE_INFO lpArmResourceInfo
)
```

パラメータ

- | | |
|------------------------------|--|
| [i] <i>lpszResourceName</i> | リソース定義情報の取得対象とするリソースのリソース識別子を指定する、NULLで終わる文字列へのポインタです。 |
| [o] <i>lpArmResourceInfo</i> | リソース定義情報を受け取るARM_RESOURCE_INFO構造体へのポインタです。 |

戻り値

- | | |
|-------|--|
| TRUE | 関数が正常終了しました。 |
| FALSE | 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。 |

拡張エラー情報

- | | |
|-------------------------------|-------------------------------|
| ARMERR_NOT_INITIALIZED | CLUSTERPRO APIが初期化されていません。 |
| ARMERR_CANNOT_CONNECT | CLUSTERPROモニタに接続できません。 |
| ARMERR_COMM_ERROR | CLUSTERPROモニタとの通信でエラーが発生しました。 |
| ARMERR_NOT_ENOUGH_MEMORY | メモリのアロケートに失敗しました。 |
| ARMERR_REGISTRY_ACCESS_FAILED | レジストリのアクセスに失敗しました。 |
| ARMERR_INVALID_PARAMETER | パラメータの指定が不正です。 |
| ARMERR_RESOURCE_NOT_FOUND | 指定リソースが存在しません。 |

説明

本関数は、ローカルサーバが所属するクラスタ内で定義されている、*lpszResourceName*で指定されたリソースの定義情報をレジストリから取得し、*lpArmResourceInfo*で指定されたバッファに格納します。

本関数は、切替ディスクパーティションに対して、固有パラメータ構造体ARM_RESOURCE_INFO_DISK_2の形式で情報を返却します。リソース定義情報取得関数ArmGetResourceInfoでは、切替ディスクパーティションに対して、固有パラメータ構造体ARM_RESOURCE_INFO_DISKの形式で情報を返却します。

参照

- 4.1.2.8 ArmGetResourceInfo – リソース定義情報の取得 (Ver 2.x 互換)
- 4.2.2 ARM_RESOURCE_INFO – リソース定義情報

4.1.2.10 ArmEnumLogicalService — 論理サービス名の列挙

概要

クラスタ内で定義されている論理サービス名を列挙します。

形式

```
BOOL ArmEnumLogicalService (  
    LPCTSTR lpszPrimaryServer,  
    LPTSTR lpszCurrentServer,  
    LPDWORD lpcchCurrentServer,  
    LPTSTR lpszServiceList,  
    LPDWORD lpcchServiceList  
)
```

パラメータ

- | | |
|------------------------------|--|
| [i] <i>lpszPrimaryServer</i> | プライマリサーバ名を示す、NULLで終わる文字列へのポインタです。本関数は、このパラメータで指定されたサーバをプライマリサーバとする論理サービス名の一覧を返却します。 |
| [o] <i>lpszCurrentServer</i> | カレントサーバ名を受け取るバッファへのポインタです。本関数により返却される論理サービスに対し現在設定されているカレントサーバのサーバ名を示す、NULLで終わる文字列が格納されます。カレントサーバ名を受け取る必要がない場合には、本パラメータにNULLを指定することができます。 |
| [io] | <i>lpcchCurrentServer</i> バッファ <i>lpszCurrentServer</i> のサイズ（終端のNULL文字分を含む文字数）を示す変数へのポインタです。関数のリターン時には、バッファに格納された文字数（終端のNULL文字を含まない）がセットされます。また、指定されたバッファサイズが小さすぎた場合には、必要な文字数（終端のNULL文字分を含む）がセットされ、拡張エラー情報で ARMERR_BUFFER_TOO_SMALL が返却されます。
(※ サーバ名の最大長は MAX_SERVERNAME_LEN です)
<i>lpszCurrentServer</i> にNULLを指定した場合にのみ、本パラメータにNULLを指定することができます。 |
| [o] <i>lpszServiceList</i> | 論理サービス名一覧を受け取るバッファへのポインタです。バッファには、論理サービス名がNULL文字で区切られて並べられ、連続する2つのNULL文字により終了します。 |
| [io] | <i>lpcchServiceList</i> バッファ <i>lpszServiceList</i> のサイズ（終端のNULL文字分を含む文字数）を示す変数へのポインタです。関数のリターン時には、バッファに格納された文字数（終端のNULL文字分を含む）がセットされます。また、指定されたバッファサイズが小さすぎた場合には、必要な文字数（終端のNULL文字分を含む）がセットされ、拡張エラー情報で ARMERR_BUFFER_TOO_SMALL が返却されます。
(※ 論理サービス名の最大長は MAX_LOGSERVICENAME_LEN 、登録可能な論理サービスの最大個数は MAX_LOGSERVICE_NUM です) |

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_NOT_ENOUGH_MEMORY	メモリのアロケートに失敗しました。
ARMERR_REGISTRY_ACCESS_FAILED	レジストリのアクセスに失敗しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_SERVER_NOT_FOUND	指定したサーバは存在しません。
ARMERR_BUFFER_TOO_SMALL	データ返却用バッファが小さすぎます。

説明

本関数は、*lpszPrimaryServer*で指定されたサーバをプライマリサーバに持つ論理サービス名の一覧を取得し、*lpszServiceList*で指定されたバッファへ格納します。また、該当する論理サービス名に対し、現在設定されているカレントサーバのサーバ名を、*lpszCurrentServer*で指定されたバッファに格納します。

4.1.2.11 ArmGetLogicalServiceInfo — 論理サービス名定義情報の取得

概要

指定した論理サービス名の定義情報を取得します。

形式

```
BOOL ArmGetLogicalServiceInfo (
    LPCTSTR lpszServiceName,
    GETSRVINFOPROC GetSrvInfoProc,
    LPARAM lParam
)
```

パラメータ

[i] <i>lpszServiceName</i>	論理サービス名を指定する、NULLで終わる文字列へのポインタです。
[i] <i>GetSrvInfoProc</i>	<i>lpszServiceName</i> で指定された論理サービス名に関する定義情報を受け取る、アプリケーション定義のコールバック関数のアドレスです。（4.1.2.12 GetSrvInfoProc関数参照）
[i] <i>lParam</i>	コールバック関数GetSrvInfoProcに渡される、アプリケーション定義の32ビット値です。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_NOT_ENOUGH_MEMORY	メモリのアロケートに失敗しました。
ARMERR_REGISTRY_ACCESS_FAILED	レジストリのアクセスに失敗しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_NOT_FOUND	指定した論理サービス名は存在しません。
ARMERR_CANCELLED	コールバック関数から定義情報取得の中止が返却されました。

説明

本関数は、*lpszServiceName*で指定された論理サービス名に関する定義情報（プライマリサーバおよびカレントサーバのサーバ名）を取得します。本関数は、アプリケーション定義のコールバック関数GetSrvInfoProcを順次呼び出し、論理サービス名の定義情報を渡します。本関数は、指定された論理サービス名に関する定義情報をすべて取得するか、またはコールバック関数がFALSEを返却するまで、コールバック関数に定義情報を渡し続けます。

参照

4. 1. 2. 12 GetSrvInfoProcー 論理サービス名の定義情報取得用コールバック関数

4. 1. 2. 12 GetSrvInfoProc — 論理サービス名の定義情報取得用コールバック関数

概要

ArmGetLogicalServiceInfo関数で返却される論理サービス名の定義情報を受け取るための、アプリケーション定義のコールバック関数です。

形式

```
BOOL CALLBACK GetSrvInfoProc(  
    LPCTSTR lpszServiceName,  
    LPCTSTR lpszPrimaryServer,  
    LPCTSTR lpszCurrentServer,  
    LPARAM lParam  
)
```

パラメータ

[i] <i>lpszServiceName</i>	論理サービス名を示す、NULLで終わる文字列へのポインタです。
[i] <i>lpszPrimaryServer</i>	論理サービス名に対し設定されているプライマリサーバ名を示す、NULLで終わる文字列へのポインタです。
[i] <i>lpszCurrentServer</i>	論理サービス名に対し設定されているカレントサーバ名を示す、NULLで終わる文字列へのポインタです。
[i] <i>lParam</i>	ArmGetLogicalServiceInfo関数で指定された、アプリケーション定義の値です。

戻り値

TRUE	定義情報の取得を継続する場合に返却します。
FALSE	定義情報の取得を中止する場合に返却します。

説明

GetSrvInfoProc関数は、ArmGetLogicalServiceInfo関数の実行結果として、指定された論理サービス名に対し設定されている定義情報（プライマリサーバ名、カレントサーバ名）を受け取るための、アプリケーション定義のコールバック関数です。指定論理サービス名の定義情報をすべて返却するか、または本関数でFALSEを返却するまで、論理サービス名の定義情報が本関数に順次受け渡されます。

アプリケーションは、ArmGetLogicalServiceInfo実行時に本関数のアドレスを指定しなければなりません。

本関数に渡される *lpszPrimaryServer* および *lpszCurrentServer* の示す文字列の内容は、本関数からリターンするまでの間でのみ有効です。文字列の内容を保持する必要がある場合には、本関数内で文字列を別の記憶領域へコピーしてください。

本関数の関数名 “GetSrvInfoProc” は、アプリケーション定義の関数名のプレースホルダです。

参照

4. 1. 2. 11 ArmGetLogicalServiceInfo — 論理サービス名定義情報の取得

4.1.3 クラスタ状態の取得/監視

サーバ動作状態の取得/監視、クラスタ資源所有権の確認を行なうAPIを提供します。

* ArmGetServerStatus	—	サーバの動作状態を取得します。
* ArmWatchServer	—	サーバの動作状態を監視します（同期型）。
* ArmStartWatchServer	—	サーバの動作状態を監視します（非同期型）。
* ArmCancelWatchServer	—	非同期サーバ監視を中止します。
* WatchServerProc	—	非同期サーバ監視用コールバック関数。
* ArmRegisterWatchServerEvent	—	サーバ監視用イベントオブジェクトを登録します。
* ArmCancelWatchServerEvent	—	サーバ監視用イベントオブジェクトの登録を取り消します。
* ArmGetServerEvent	—	サーバ状態変更通知を取得します。
* ArmCheckResourceOwner	—	ローカルサーバがリソースの所有権を持つか確認します。

4.1.3.1 ArmGetServerStatus — サーバ動作状態の取得

概要

指定したサーバの現在の動作状態を取得します。

形式

```
BOOL ArmGetServerStatus (
    LPCTSTR lpszServerName,
    LPDWORD lpdwStatus
)
```

パラメータ

[i] <i>lpszServerName</i>	動作状態の取得対象とするサーバのコンピュータ名を指定する、NULLで終わる文字列へのポインタです。ローカルサーバを指定する場合には、定数ARM_LOCAL_SERVERを指定することができます。
[o] <i>lpdwStatus</i>	指定したサーバの動作状態を受け取る変数へのポインタです。以下のサーバ動作状態が返却されます。 ARM_STATUS_SINGLE_SERVER クラスタに所属していない、シングルサーバとして動作しています。 (クラスタの設定を行っていない状態) ARM_STATUS_CLUSTER クラスタとして正常に動作しています。 ARM_STATUS_OFF_CLUSTER クラスタには所属しているが、クラスタから切り離されて動作しています。 (ダウン後再起動状態) ARM_STATUS_STOPPED 停止状態です。 ARM_STATUS_DOWN ダウン状態です。 ARM_STATUS_RECOVERING failover処理を行なっています。 上記の状態に次の状態が組み合わされます。 ARM_STATUS_BACKUP 他サーバからfailoverしてきたリソースがあります。 ARM_STATUS_FAIL 他サーバへfailoverしたリソースがあります。 ARM_STATUS_MISS failover処理の失敗したリソースがあります。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_SERVER_NOT_FOUND	指定サーバが存在しません。

説明

本関数は、ローカルサーバが所属するクラスタを構成している、*IpszServerName*で指定されたサーバの動作状態を取得し、*IpArmServerStatus*の指す変数に格納します。

4.1.3.2 ArmWatchServer — サーバ動作状態の監視（同期型）

概要

指定したサーバの動作状態を監視します（動作状態の変更を検出するまで制御を戻しません）。

形式

```
BOOL ArmWatchServer(  
    LPCTSTR lpszTargetServer,  
    LPDWORD lpdwEvent,  
    LPDWORD lpdwEventId,  
    LPTSTR lpszCauseServer,  
    LPDWORD lpcchCauseServer,  
    DWORD dwTimeOut  
)
```

パラメータ

[i] <i>lpszTargetServer</i>	動作状態の監視対象とするサーバのコンピュータ名を指定する、NULLで終わる文字列へのポインタです。ローカルサーバを指定する場合には、定数ARM_LOCAL_SERVERを指定することができます。また、定数ARM_ALL_SERVERを指定した場合には、クラスタ内の全サーバを監視対象とします。																				
[o] <i>lpdwEvent</i>	指定サーバの検出された動作状態を受け取る変数へのポインタです。以下のイベントが返却されます。 <table><tr><td>ARM_EVENT_SHUTDOWN</td><td>シャットダウンが開始されました。</td></tr><tr><td>ARM_EVENT_DOWN</td><td>サーバダウンを検出しました。</td></tr><tr><td>ARM_EVENT_RETURN_TO_CLUSTER</td><td>クラスタへ復帰しました。</td></tr><tr><td>ARM_EVENT_FAILOVER_START</td><td>failover処理を開始しました。</td></tr><tr><td>ARM_EVENT_FAILOVER_END</td><td>failover処理が終了しました。</td></tr><tr><td>ARM_EVENT_FAILBACK_START</td><td>failback処理を開始しました。</td></tr><tr><td>ARM_EVENT_FAILBACK_END</td><td>failback処理が終了しました。</td></tr><tr><td>ARM_EVENT_FAILBACK_FAILURE</td><td>failback処理が失敗しました。</td></tr><tr><td>ARM_EVENT_LSW_SWITCHED</td><td>回線切替装置で、回線の切り替えが発生した。</td></tr><tr><td>ARM_EVENT_PRINTER_SWITCHED</td><td>プリンタの切り替えが発生した。</td></tr></table>	ARM_EVENT_SHUTDOWN	シャットダウンが開始されました。	ARM_EVENT_DOWN	サーバダウンを検出しました。	ARM_EVENT_RETURN_TO_CLUSTER	クラスタへ復帰しました。	ARM_EVENT_FAILOVER_START	failover処理を開始しました。	ARM_EVENT_FAILOVER_END	failover処理が終了しました。	ARM_EVENT_FAILBACK_START	failback処理を開始しました。	ARM_EVENT_FAILBACK_END	failback処理が終了しました。	ARM_EVENT_FAILBACK_FAILURE	failback処理が失敗しました。	ARM_EVENT_LSW_SWITCHED	回線切替装置で、回線の切り替えが発生した。	ARM_EVENT_PRINTER_SWITCHED	プリンタの切り替えが発生した。
ARM_EVENT_SHUTDOWN	シャットダウンが開始されました。																				
ARM_EVENT_DOWN	サーバダウンを検出しました。																				
ARM_EVENT_RETURN_TO_CLUSTER	クラスタへ復帰しました。																				
ARM_EVENT_FAILOVER_START	failover処理を開始しました。																				
ARM_EVENT_FAILOVER_END	failover処理が終了しました。																				
ARM_EVENT_FAILBACK_START	failback処理を開始しました。																				
ARM_EVENT_FAILBACK_END	failback処理が終了しました。																				
ARM_EVENT_FAILBACK_FAILURE	failback処理が失敗しました。																				
ARM_EVENT_LSW_SWITCHED	回線切替装置で、回線の切り替えが発生した。																				
ARM_EVENT_PRINTER_SWITCHED	プリンタの切り替えが発生した。																				

- [o] *lpdwEventId* サーバイベントに発生順に付加されるシーケンシャル番号を受け取る変数へのポインタです。
- [o] *lpzCauseServer* 動作状態の変更が検出されたサーバのコンピュータ名（終端のNULL文字を含む）を受け取るバッファへのポインタです。コンピュータ名が不要の場合には、NULLを指定できます。
- [io] *lpchCauseServer* *lpzCauseServer*で指定されたバッファのサイズを指定する変数へのポインタです。関数のリターン時には、バッファに格納された文字数（終端のNULL文字を含まない）がセットされます。
指定バッファサイズが小さすぎた場合には、必要な文字数（終端のNULL文字を含む）がセットされ、拡張エラー情報で **ARMERR_BUFFER_TOO_SMALL**が返却されます。
(※ サーバ名の最大長は**MAX_SERVERNAME_LEN**です)
*lpzCauseServer*にNULLを指定した場合にのみ、本パラメータにNULLを指定できます。
(※ サーバ名の最大長は**MAX_SERVERNAME_LEN**です)
- [i] *dwTimeout* タイムアウト時間をミリ秒単位で指定します。**INFINITE**を指定した場合には、タイムアウトは発生しません。

戻り値

- TRUE** 関数が正常終了しました。
- FALSE** 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_NOT_ENOUGH_MEMORY	メモリのアロケートに失敗しました。
ARMERR_CANNOT_ALLOCATE_RESOURCE	OS資源のアロケートに失敗しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_SERVER_NOT_FOUND	指定したサーバが存在しません。
ARMERR_TIMEOUT	タイムアウトが発生しました。
ARMERR_BUFFER_TOO_SMALL	データ返却用バッファが小さすぎます。

説明

本関数は、ローカルサーバの所属するクラスタを構成している、*IpszTargetServer*で指定されたサーバの動作状態の変更を監視します。本関数は、指定サーバの動作状態の変更を検出するか、または指定したタイムアウト時間が経過するまで、本関数を呼び出したアプリケーションプロセスに制御を戻しません。

監視対象サーバに**ARM_ALL_SERVER**を指定した場合には、ローカルサーバの所属するクラスタを構成しているすべてのサーバが監視対象になります。

注意

本関数は、本関数実行後に最初に検出したサーバ状態変更通知のみを取得します。本関数のリターン後、本関数を再実行するまでの間に発生したサーバ状態変更通知は捨てられます。サーバ動作状態の監視開始後のすべてのサーバ状態変更通知を取得したい場合には、**ArmStartWatchServer**関数または**ArmRegisterWatchServerEvent**関数を使用してください。

参照

- 4.1.3.3 **ArmStartWatchServer** — サーバ動作状態の監視（非同期型）
- 4.1.3.6 **ArmRegisterWatchServerEvent** — サーバ監視用イベントオブジェクトの登録

4.1.3.3 ArmStartWatchServer — サーバ動作状態の監視（非同期型）

概要

指定したサーバの動作状態を監視します（状態変更検出時に、非同期にコールバック関数を呼び出します）。

形式

```
BOOL ArmStartWatchServer (
    LPCTSTR lpszServerName,
    WATCHSERVERPROC WatchServerProc,
    LPDWORD lpdwWatchId
)
```

パラメータ

- | | |
|----------------------------|---|
| [i] <i>lpszServerName</i> | 動作状態の監視対象とするサーバのコンピュータ名を指定する、NULLで終わる文字列へのポインタです。ローカルサーバを指定する場合には、定数ARM_LOCAL_SERVERを指定することができます。また、定数 ARM_ALL_SERVERを指定した場合には、クラスタ内の全サーバを監視対象とします。 |
| [i] <i>WatchServerProc</i> | サーバ動作状態の変更を検出した時に呼び出される、アプリケーション定義のコールバック関数のアドレスです。
(4.1.3.5 WatchServerProc関数参照) |
| [o] <i>lpdwWatchId</i> | 監視IDを受け取る変数へのポインタです。 |

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_NOT_ENOUGH_MEMORY	メモリのアロケートに失敗しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_SERVER_NOT_FOUND	指定したサーバが存在しません。

説明

本関数は、ローカルサーバの所属するクラスタを構成している、*lpszServerName*で指定されたサーバの動作状態の変更を監視します。本関数は、アプリケーション定義のコールバック関数 *WatchServerProc*を登録した後、直ちに本関数を呼び出したアプリケーションプロセスに制御を戻します。コールバック関数 *WatchServerProc*は、サーバの動作状態の変更が検出された時に、非同期に呼び出されます。

参照

- 4.1.3.4 ArmCancelWatchServer — サーバ動作状態監視（非同期型）の中止
- 4.1.3.5 WatchServerProc — サーバ動作状態監視（非同期型）用コールバック関数

4.1.3.4 ArmCancelWatchServer — サーバ動作状態監視（非同期型）の中止

概要

ArmStartWatchServer関数によるサーバの動作状態の監視を中止します。

形式

```
BOOL ArmCancelWatchServer (  
    DWORD dwWatchId  
)
```

パラメータ

[i] *dwWatchId* ArmStartWatchServer関数で返却された監視IDを指定します。

戻り値

TRUE 関数が正常終了しました。
FALSE 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_INVALID_ID	指定した監視IDが不正です。

説明

本関数は、ArmStartWatchServer関数により登録されたサーバ動作状態監視を中止します。中止の対象とするサーバ動作状態変更監視は、監視ID *dwWatchId*で指定されます。

参照

4.1.3.3 ArmStartWatchServer — サーバ動作状態の監視（非同期型）

4.1.3.5 WatchServerProc — サーバ動作状態監視(非同期型)用コールバック関数

概要

ArmStartWatchServer関数により登録され、サーバの動作状態の変更検出時に呼び出される、アプリケーション定義のコールバック関数です。

形式

```
VOID CALLBACK WatchServerProc (
    DWORD    dwWatchId,
    LPARM_EVENT_PARAM  lpArmEventParam,
    DWORD    dwEvent,
    DWORD    dwEventId
)
```

パラメータ

[i] <i>dwWatchId</i>	対応するArmStartWatchServer関数を識別する監視IDです。	
[i] <i>lpArmEventParam</i>	サーバ状態変更通知に関する付加情報が格納された、 ARM_EVENT_PARAM 構造体へのポインタです。	
[i] <i>dwEvent</i>	ARM_EVENT_SHUTDOWN	シャットダウンが開始されました。
	ARM_EVENT_DOWN	サーバダウンを検出しました。
	ARM_EVENT_RETURN_TO_CLUSTER	クラスタへ復帰しました。
	ARM_EVENT_FAILOVER_START	failover処理を開始しました。
	ARM_EVENT_FAILOVER_END	failover処理が終了しました。
	ARM_EVENT_FAILBACK_START	failback処理を開始しました。
	ARM_EVENT_FAILBACK_END	failback処理が終了しました。
	ARM_EVENT_FAILBACK_FAILURE	failback処理が失敗しました。
	ARM_EVENT_LSW_SWITCHED	回線切替装置で、回線の切り替えが発生した。
	ARM_EVENT_PRINTER_SWITCHED	プリンタの切り替えが発生した。
[i] <i>dwEventId</i>	サーバイベントに発生順に付加されるシーケンシャル番号です。	

戻り値

なし

説明

本関数は、ArmStartWatchServer関数により登録されるコールバック関数であり、サーバの動作状態の変更が検出された時に非同期にコールバックされます。

本関数に渡される *lpArmEventParam* の指す **ARM_EVENT_PARAM** 構造体の内容は、本関数からリターンするまでの間でのみ有効です。**ARM_EVENT_PARAM** 構造体の内容を保持する必要がある場合には、本関数内で別の記憶領域へコピーしてください。

本関数の関数名 “WatchServerProc” は、アプリケーション定義の関数名のプレースホルダです。

注意

本コールバック関数は、CLUSTERPRO API内部のイベント処理スレッド上で実行されます。したがって、本コールバック関数内で、多くの処理時間を必要とする処理を実行すると、その間はCLUSTERPROモニタからのイベントを受信できなくなります。イベント処理に多くの処理時間を必要とする場合には、本コールバック関数内では、イベント処理開始の契機を取得するのみとし、アプリケーションの生成したスレッド上でメインの処理を実行するようにしてください。

参照

4.1.3.3 ArmStartWatchServer — サーバ動作状態の監視（非同期型）

4.1.3.6 ArmRegisterWatchServerEvent — サーバ監視用イベントオブジェクトの登録

概要

CLUSTERPROモニタからのサーバ状態変更通知と待つために使用する、イベントオブジェクトを登録します。

形式

```
BOOL ArmRegisterWatchServerEvent (
    HANDLE  hEvent,
    LPCTSTR lpszServerName
)
```

パラメータ

- | | |
|---------------------------|--|
| [i] <i>hEvent</i> | CLUSTERPROモニタからサーバ状態変更通知を受信したときにシグナル状態に設定される、イベントオブジェクトのハンドルです。 |
| [i] <i>lpszServerName</i> | 状態変更の監視対象とするサーバのコンピュータ名を指定する、NULLで終わる文字列へのポインタです。ローカルサーバを指定する場合には、定数ARM_LOCAL_SERVERを指定することができます。また、定数ARM_ALL_SERVERを指定した場合には、クラスタ内の全サーバを監視対象とします。 |

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_ALREADY_REGISTERED	指定したイベントオブジェクトは既に登録されています。
ARMERR_SERVER_NOT_FOUND	指定サーバが存在しません。

説明

本関数は、CLUSTERPROモニタからのサーバ状態変更通知と待つために使用する、イベントオブジェクトの登録を行ないます。本関数により登録されたイベントオブジェクトは、サーバ状態変更通知を受信したときにシグナル状態に設定されます。

注意

本関数により登録したイベントオブジェクトがシグナル状態に設定された場合には、必ずArmGetServerEvent関数を実行して、サーバ状態変更通知に関する情報を取得してください。サーバ状態変更通知情報を取得しないままにしておくと、以後に発生する他のサーバ状態変更通知を受信できなくなります。

サーバ状態変更の監視が不要になった場合には、ArmCancelWatchServerEvent関数を実行して、イベントオブジェクトの登録を取り消してください。

参照

- 4.1.3.7 ArmCancelWatchServerEvent — サーバ監視用イベントオブジェクトの登録取り消し
- 4.1.3.8 ArmGetServerEvent — サーバ状態変更通知の取得

4.1.3.7 ArmCancelWatchServerEvent — サーバ監視用イベントオブジェクトの登録取り消し

概要

ArmRegisterWatchServerEvent関数による、サーバ監視用イベントオブジェクトの登録を取り消します。

形式

```
BOOL ArmCancelWatchServerEvent (  
    HANDLE hEvent  
)
```

パラメータ

[i] *hEvent* ArmRegisterWatchServerEvent関数により登録した、サーバ監視用イベントオブジェクトのハンドルを指定します。

戻り値

TRUE 関数が正常終了しました。
FALSE 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_NOT_REGISTERED	指定したイベントオブジェクトは登録されていません。

説明

本関数は、ArmRegisterWatchServerEvent関数による、サーバ監視用イベントオブジェクトの登録を取り消します。本関数の実行後は、サーバ状態変更通知によるイベントオブジェクトの設定は行われません。

参照

4.1.3.6 ArmRegisterWatchServerEvent — サーバ監視用イベントオブジェクトの登録

4.1.3.8 ArmGetServerEvent — サーバ状態変更通知の取得

概要

サーバ状態変更通知に関する情報を取得します。

形式

```
BOOL ArmGetServerEvent (
    HANDLE    hEvent,
    LPDWORD   lpdwEvent,
    LPDWORD   lpdwEventId,
    LPARAM_EVENT_PARAM lpArmEventParam,
    LPDWORD   lpcbArmEventParam,
    LPBOOL    lpfRemain
)
```

パラメータ

[i] <i>hEvent</i>	ArmRegisterWatchServerEvent関数により登録した、サーバ監視用イベントオブジェクトのハンドルを指定します。																				
[o] <i>lpdwEvent</i>	検出されたサーバ状態変更通知を受け取る変数へのポインタです。 以下のサーバ状態変更通知が返却されます。 <table><tr><td>ARM_EVENT_SHUTDOWN</td><td>シャットダウンが開始されました。</td></tr><tr><td>ARM_EVENT_DOWN</td><td>サーバダウンを検出しました。</td></tr><tr><td>ARM_EVENT_RETURN_TO_CLUSTER</td><td>クラスタへ復帰しました。</td></tr><tr><td>ARM_EVENT_FAILOVER_START</td><td>failover処理を開始しました。</td></tr><tr><td>ARM_EVENT_FAILOVER_END</td><td>failover処理が終了しました。</td></tr><tr><td>ARM_EVENT_FAILBACK_START</td><td>failback処理を開始しました。</td></tr><tr><td>ARM_EVENT_FAILBACK_END</td><td>failback処理が終了しました。</td></tr><tr><td>ARM_EVENT_FAILBACK_FAILURE</td><td>failback処理が失敗しました。</td></tr><tr><td>ARM_EVENT_LSW_SWITCHED</td><td>回線切替装置で、回線の切り替えが発生した。</td></tr><tr><td>ARM_EVENT_PRINTER_SWITCHED</td><td>プリンタの切り替えが発生した。</td></tr></table>	ARM_EVENT_SHUTDOWN	シャットダウンが開始されました。	ARM_EVENT_DOWN	サーバダウンを検出しました。	ARM_EVENT_RETURN_TO_CLUSTER	クラスタへ復帰しました。	ARM_EVENT_FAILOVER_START	failover処理を開始しました。	ARM_EVENT_FAILOVER_END	failover処理が終了しました。	ARM_EVENT_FAILBACK_START	failback処理を開始しました。	ARM_EVENT_FAILBACK_END	failback処理が終了しました。	ARM_EVENT_FAILBACK_FAILURE	failback処理が失敗しました。	ARM_EVENT_LSW_SWITCHED	回線切替装置で、回線の切り替えが発生した。	ARM_EVENT_PRINTER_SWITCHED	プリンタの切り替えが発生した。
ARM_EVENT_SHUTDOWN	シャットダウンが開始されました。																				
ARM_EVENT_DOWN	サーバダウンを検出しました。																				
ARM_EVENT_RETURN_TO_CLUSTER	クラスタへ復帰しました。																				
ARM_EVENT_FAILOVER_START	failover処理を開始しました。																				
ARM_EVENT_FAILOVER_END	failover処理が終了しました。																				
ARM_EVENT_FAILBACK_START	failback処理を開始しました。																				
ARM_EVENT_FAILBACK_END	failback処理が終了しました。																				
ARM_EVENT_FAILBACK_FAILURE	failback処理が失敗しました。																				
ARM_EVENT_LSW_SWITCHED	回線切替装置で、回線の切り替えが発生した。																				
ARM_EVENT_PRINTER_SWITCHED	プリンタの切り替えが発生した。																				
[o] <i>lpdwEventId</i>	サーバイベントに発生順に付加されるシーケンシャル番号を受け取る変数へのポインタです。																				

- [o] *lpArmEventParam* サーバ状態変更通知に関する付加情報を受け取る、**ARM_EVENT_PARAM**構造体へのポインタです。付加情報が不要の場合には、NULLを指定できます。
- [io] *lpcbArmEventParam* *lpArmEventParam*で指定されたバッファのサイズを指定する変数へのポインタです。関数のリターン時には、バッファに格納されたデータのバイト数がセットされます。指定バッファサイズが小さすぎた場合には、必要なバイト数がセットされ、拡張エラー情報で**ARMERR_BUFFER_TOO_SMALL**が返却されます。*lpArmEventParam*にNULLを指定した場合にのみ、本パラメータにNULLを指定できます。
- [o] *lpfRemain* 他にもサーバ状態変更通知が残っているかどうかを受け取る変数へのポインタです。以下の値がセットされます。
TRUE 他のサーバ状態変更通知が残っている
FALSE 他のサーバ状態変更通知はない
 本パラメータで**TRUE**が返却された場合には、**FALSE**が返却されるまで本関数を繰り返し実行してください。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。
拡張エラー情報	
ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_NOT_REGISTERED	指定したイベントオブジェクトは登録されていません。
ARMERR_BUFFER_TOO_SMALL	データ返却用バッファが小さすぎます。
ARMERR_NOT_FOUND	取得できるサーバ状態変更通知はありません。

説明

本関数は、ArmRegisterWatchServerEvent関数により登録されたイベントオブジェクトがシグナル状態に設定された時に、サーバ状態変更通知に関する情報を取得するために使用されます。

本関数は、サーバ状態変更通知を1つずつ取得します。本関数実行時に、2つ以上のサーバ状態変更通知が存在する場合には、他のサーバ状態変更通知が残っていることを示すために、パラメータ *lpfRemain* に**TRUE**がセットされます。*lpfRemain*で**TRUE**が返却された場合には、**FALSE**になるまで、繰り返し本関数を実行してください。

参照

- 4.1.3.6 ArmRegisterWatchServerEvent — サーバ監視用イベントオブジェクトの登録
- 4.2.3 ARM_EVENT_PARAM — サーバ状態変更通知パラメータ

4.1.3.9 ArmCheckResourceOwner — クラスタ資源の所有権の確認

概要

指定したクラスタ資源に対し、ローカルサーバが所有権を持っているか確認します。

形式

```
BOOL ArmCheckResourceOwner (
    LPCTSTR lpszResourceName,
    LPBOOL lpfOwner
)
```

パラメータ

[i] <i>lpszResourceName</i>	所有権確認の対象とするクラスタ資源のリソース名を指定する、NULLで終わる文字列へのポインタです。
[o] <i>lpfOwner</i>	ローカルサーバがクラスタ資源の所有権を持つか否かを受け取る変数へのポインタです。 本関数のリターン時に以下の値がセットされます。 TRUE クラスタ資源の所有権を持ちます。 FALSE クラスタ資源の所有権を持ちません。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_REGISTRY_ACCESS_FAILED	レジストリのアクセスに失敗しました。
ARMERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMERR_RESOURCE_NOT_FOUND	指定したリソースが存在しません。

説明

本関数は、現在ローカルサーバが*lpszResourceName*で指定されたクラスタ資源の所有権を持つか否かを確認し、結果を*lpfOwner*の指す変数へ格納します。

4.1.4 レジストリの操作

クラスタ内で同期のとれたレジストリの操作を行うためのAPIを提供します。

* ArmRegOpenKey	— レジストリキーをオープンします。
* ArmRegCreateKey	— レジストリキー作成/オープンします。
* ArmRegDeleteKey	— レジストリキーを削除します。
* ArmRegFlushKey	— オープン中のキーをレジストリに書き込みます。
* ArmRegCloseKey	— レジストリキーをクローズします。
* ArmRegSetValue	— レジストリキーに値を設定します。
* ArmRegDeleteValue	— レジストリキーに設定されている値を削除します。

※ レジストリキー/値の列挙や値の問い合わせ等にはWin32 APIを使用してください。
ただし、**ArmRegOpenKey関数およびArmRegCreateKey関数で返却されたオープンキーハンドルをWin32 APIで使用することはできません**。別途Win32 APIを使用してオープンキーハンドルを取得してください。

※ 上記APIによるレジストリの更新内容は、クラスタ内の他のサーバへ即時反映されます。ただし、更新内容が反映されるのは、クラスタ内の全サーバがクラスタ動作している場合のみです。クラスタ内にクラスタ動作していないサーバが1台でも存在した場合には、レジストリ更新操作は失敗します。

4.1.4.1 ArmRegOpenKey — レジストリキーのオープン

概要

指定されたレジストリキーをオープンします。

形式

```
BOOL ArmRegOpenKey (  
    HKEY  hKey,  
    LPCTSTR lpszSubKey,  
    PHKEY phkResult  
)
```

パラメータ

- [i] *hKey* ArmRegOpenKey/ArmRegCreateKey関数により既にオープンされているレジストリキーのハンドル、または以下の定義済みハンドル値を指定します。
- HKEY_CLASSES_ROOT
HKEY_CURRENT_USER
HKEY_LOCAL_MACHINE
HKEY_USERS
- [i] *lpszSubKey* オープンするサブキー名を指定する、NULLで終わる文字列へのポインタです。
- [o] *phkResult* オープンされたキーハンドルを受け取る変数へのポインタです。

戻り値

- TRUE 関数が正常終了しました。
- FALSE 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_NOT_ENOUGH_MEMORY	メモリのアロケートに失敗しました。
ARMERR_FAILED_REGISTRY_OPEN	レジストリキーのオープンに失敗しました。

説明

本関数は、*hKey*で指定されたオープン済みレジストリキー配下の、*lpszSubKey*で指定されたサブキーをオープンし、そのキーハンドルを*phkResult*で指定された変数に格納します。

本関数で返却されるオープンキーハンドルは、ARM APIの提供するレジストリ操作関数でのみ使用可能です。Win32 APIのレジストリ操作関数で使用することはできません。

参照

- 4.1.4.2 ArmRegCreateKey — レジストリキーの作成/オープン
- 4.1.4.3 ArmRegDeleteKey — レジストリキーの削除
- 4.1.4.5 ArmRegCloseKey — レジストリキーのクローズ

4.1.4.2 ArmRegCreateKey — レジストリキーの作成/オープン

概要

指定されたレジストリキーを作成またはオープンします。

形式

```
BOOL ArmRegCreateKey (  
    HKEY  hKey,  
    LPCTSTR lpszSubKey,  
    PHKEY  phkResult,  
    LPDWORD lpdwDisposition  
)
```

パラメータ

- | | |
|----------------------------|--|
| [i] <i>hKey</i> | ArmRegOpenKey/ArmRegCreateKey関数により既にオープンされているレジストリキーのハンドル、または以下の定義済みハンドル値を指定します。
HKEY_CLASSES_ROOT
HKEY_CURRENT_USER
HKEY_LOCAL_MACHINE
HKEY_USERS |
| [i] <i>lpszSubKey</i> | 作成するサブキー名を指定する、NULLで終わる文字列へのポインタです。 |
| [o] <i>phkResult</i> | 作成/オープンされたキーハンドルを受け取る変数へのポインタです。 |
| [o] <i>lpdwDisposition</i> | 以下の処理結果のいずれかを受け取る変数へのポインタです。
ARM_REG_CREATED_NEW_KEY 指定されたサブキーが存在しなかったため、新たに作成した。
ARM_REG_OPENED_EXISTING_KEY 指定されたサブキーが既に存在しているため、オープンした。 |

戻り値

- | | |
|-------|--|
| TRUE | 関数が正常終了しました。 |
| FALSE | 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。 |

拡張エラー情報

- | | |
|-----------------------------|---|
| ARMERR_NOT_INITIALIZED | CLUSTERPRO APIが初期化されていません。 |
| ARMERR_CANNOT_CONNECT | CLUSTERPROモニタに接続できません。 |
| ARMERR_COMM_ERROR | CLUSTERPROモニタとの通信でエラーが発生しました。 |
| ARMERR_NOT_ENOUGH_MEMORY | メモリのアロケートに失敗しました。 |
| ARMERR_FAILED_REGISTRY_OPEN | ローカルサーバにおいてレジストリキーのオープンに失敗しました。サーバ間のレジストリ同期は不完全な状態です。 |

ARMERR_FAILED_REGISTRY_SYNCHRONIZE

リモートサーバでのレジストリ同期更新処理が失敗しました（ローカルサーバで更新処理は実行されません）。

説明

本関数は、*hKey*で指定されたオープン済みレジストリキー配下に、*lpszSubKey*で指定されたサブキーを作成します。指定サブキーが既に存在する場合には、サブキーをオープンします。

本関数で返却されるオープンキーハンドルは、CLUSTERPRO APIの提供するレジストリ操作関数でのみ使用可能です。Win32 APIのレジストリ操作関数で使用することはできません。

参照

- 4.1.4.1 ArmRegOpenKey — レジストリキーのオープン
- 4.1.4.3 ArmRegDeleteKey — レジストリキーの削除
- 4.1.4.5 ArmRegCloseKey — レジストリキーのクローズ

4.1.4.3 ArmRegDeleteKey — レジストリキーの削除

概要

指定されたレジストリキーを削除します。

形式

```
BOOL ArmRegDeleteKey (  
    HKEY hKey,  
    LPCTSTR lpszSubKey  
)
```

パラメータ

- [i] *hKey* ArmRegOpenKey/ArmRegCreateKey関数により既にオープンされているレジストリキーのハンドル、または以下の定義済みハンドル値を指定します。
- HKEY_CLASSES_ROOT
HKEY_CURRENT_USER
HKEY_LOCAL_MACHINE
HKEY_USERS
- [i] *lpszSubKey* 削除するサブキー名を指定する、NULLで終わる文字列へのポインタです。

戻り値

- TRUE 関数が正常終了しました。
FALSE 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_NOT_ENOUGH_MEMORY	メモリのアロケートに失敗しました。
ARMERR_NOT_EXIST_REGISTRY	レジストリキーが存在しません。
ARMERR_FAILED_REGISTRY_DELETE	ローカルサーバにおいてレジストリキーの削除に失敗しました。サーバ間のレジストリ同期は不完全な状態です。
ARMERR_FAILED_REGISTRY_SYNCHRONIZE	リモートサーバでのレジストリ同期更新処理が失敗しました（ローカルサーバで更新処理は実行されません）。

説明

本関数は、*hKey*で指定されたオープン済みレジストリキー配下の、*lpszSubKey*で指定されたサブキーを削除します。
本関数は、サブキーを持つキーを削除することはできません。

参照

- 4.1.4.1 ArmRegOpenKey — レジストリキーのオープン
- 4.1.4.2 ArmRegCreateKey — レジストリキーの作成/オープン
- 4.1.4.5 ArmRegCloseKey — レジストリキーのクローズ

4.1.4.4 ArmRegFlushKey — レジストリキーのフラッシュ

概要

オープン中のキーをレジストリに書き込みます。

形式

```
BOOL ArmRegFlushKey (  
    HKEY hKey  
)
```

パラメータ

[i] *hKey* ArmRegOpenKey/ArmRegCreateKey関数により既にオープンされているレジストリキーのハンドル、または以下の定義済みハンドル値を指定します。

```
HKEY_CLASSES_ROOT  
HKEY_CURRENT_USER  
HKEY_LOCAL_MACHINE  
HKEY_USERS
```

戻り値

TRUE 関数が正常終了しました。
FALSE 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_NOT_ENOUGH_MEMORY	メモリのアロケートに失敗しました。
ARMERR_NOT_OPEN_REGISTRY	レジストリキーがオープンされていません。
ARMERR_FAILED_REGISTRY_FLUSH	ローカルサーバにおいてレジストリの書き込みに失敗しました。サーバ間のレジストリ同期は不完全な状態です。
ARMERR_FAILED_REGISTRY_SYNCHRONIZE	リモートサーバでのレジストリ同期更新処理が失敗しました（ローカルサーバで更新処理は実行されません）。

説明

本関数は、*hKey*で指定されたオープンキーのすべての属性を、レジストリに書き込みます。両サーバ上でCLUSTERPROサービスが動作している場合にのみ、書き込みが可能です。

4.1.4.5 ArmRegCloseKey — レジストリキーのクローズ

概要

レジストリキーハンドルをクローズします。

形式

```
BOOL ArmRegCloseKey (  
    HKEY hKey  
)
```

パラメータ

[i] *hKey* クローズするレジストリキーハンドルを指定します。

戻り値

TRUE 関数が正常終了しました。
FALSE 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_NOT_ENOUGH_MEMORY	メモリのアロケートに失敗しました。
ARMERR_NOT_OPEN_REGISTRY	レジストリキーがオープンされていません。

説明

本関数は、*hKey*で指定されたレジストリキーハンドルをクローズします。

参照

- 4.1.4.1 ArmRegOpenKey — レジストリキーのオープン
- 4.1.4.2 ArmRegCreateKey — レジストリキーの作成/オープン

4.1.4.6 ArmRegSetValue — レジストリキー値の設定

概要

指定されたオープンレジストリキーに、値を設定します。

形式

```
BOOL ArmRegSetValue (  
    HKEY  hKey,  
    LPCTSTR lpszValueName,  
    DWORD  fdwType,  
    CONST BYTE * lpbData,  
    DWORD  cbData  
)
```

パラメータ

- | | |
|--------------------------|--|
| [i] <i>hKey</i> | ArmRegOpenKey/ArmRegCreateKey関数により既にオープンされているレジストリキーのハンドル、または以下の定義済みハンドル値を指定します。
HKEY_CLASSES_ROOT
HKEY_CURRENT_USER
HKEY_LOCAL_MACHINE
HKEY_USERS |
| [i] <i>lpszValueName</i> | 設定する値の名前を指定する、NULLで終わる文字列へのポインタです。 |
| [i] <i>fdwType</i> | 値のデータとして格納される情報の種類を指定します。
REG_BINARY 任意の形式のバイナリデータです。
REG_DWORD 32ビット値です。
REG_EXPAND_SZ 展開前の環境変数への参照が含まれているNULLで終わる文字列です。
REG_MULTI_SZ NULLで終わる文字列の配列です。2つのNULL終端文字で終わります。
REG_SZ NULLで終わる文字列です。 |
| [i] <i>lpbData</i> | 値として設定するデータの格納されたバッファへのポインタです。 |
| [i] <i>cbData</i> | バッファlpbDataに格納されたデータのサイズをバイト単位で指定します。データの種類のREG_SZ, REG_EXPAND_SZ, REG_MULTI_SZの場合には、終端のNULL文字もサイズに含めなければなりません。 |

戻り値

- | | |
|-------|--|
| TRUE | 関数が正常終了しました。 |
| FALSE | 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。 |
- 拡張エラー情報
- | | |
|------------------------|----------------------------|
| ARMERR_NOT_INITIALIZED | CLUSTERPRO APIが初期化されていません。 |
| ARMERR_CANNOT_CONNECT | CLUSTERPROモニタに接続できません。 |

ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_NOT_ENOUGH_MEMORY	メモリのアロケートに失敗しました。
ARMERR_NOT_OPEN_REGISTRY	レジストリキーがオープンされていません。
ARMERR_FAILED_SET_VALUE	ローカルサーバにおいて値の設定に失敗しました。サーバ間のレジストリ同期は不完全な状態です。
ARMERR_FAILED_REGISTRY_SYNCHRONIZE	リモートサーバでのレジストリ同期更新処理が失敗しました（ローカルサーバで更新処理は実行されません）。

説明

関数は、*hKey*で指定されたオープン済みレジストリキーの値フィールドに、データを格納します。

参照

4.1.4.7 ArmRegDeleteValue — レジストリキー値の削除

4.1.4.7 ArmRegDeleteValue — レジストリキー値の削除

概要

指定されたオープンレジストリキーに設定されている値を削除します。

形式

```
BOOL ArmRegDeleteValue (
    HKEY  hKey,
    LPCTSTR lpszValueName
)
```

パラメータ

[i] *hKey* ArmRegOpenKey/ArmRegCreateKey関数により既にオープンされているレジストリキーのハンドル、または以下の定義済みハンドル値を指定します。

HKEY_CLASSES_ROOT
HKEY_CURRENT_USER
HKEY_LOCAL_MACHINE
HKEY_USERS

[i] *lpszValueName* 削除する値の名前を指定する、NULLで終わる文字列へのポインタです。

戻り値

TRUE 関数が正常終了しました。
FALSE 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMERR_NOT_INITIALIZED	CLUSTERPRO APIが初期化されていません。
ARMERR_CANNOT_CONNECT	CLUSTERPROモニタに接続できません。
ARMERR_COMM_ERROR	CLUSTERPROモニタとの通信でエラーが発生しました。
ARMERR_NOT_ENOUGH_MEMORY	メモリのアロケートに失敗しました。
ARMERR_NOT_OPEN_REGISTRY	レジストリキーがオープンされていません。
ARMERR_FAILED_DELETE_VALUE	ローカルサーバにおいて値の削除に失敗しました。サーバ間のレジストリ同期は不完全な状態です。
ARMERR_FAILED_REGISTRY_SYNCHRONIZE	リモートサーバでのレジストリ同期更新処理が失敗しました（ローカルサーバで更新処理は実行されません）。

説明

本関数は、*hKey*で指定されたオープン済みレジストリキーに設定されている、*lpszValueName*で指定された値を削除します。

参照

4.1.4.6 ArmRegSetValue — レジストリキー値の設定

4.2 構造体

- * ARM_SERVER_INFO — サーバ定義情報
- * ARM_RESOURCE_INFO — リソース定義情報
- * ARM_EVENT_PARAM — サーバ状態変更通知パラメータ

4.2.1 ARM_SERVER_INFO — サーバ定義情報

概要

ARM_SERVER_INFO構造体は、サーバ定義情報を格納します。

形式

```
typedef struct _ARM_SERVER_INFO {
    TCHAR      OSVersion [MAX_VER_LEN];
    TCHAR      HomeDir [MAX_PATH_LEN];
    TCHAR      ARMVersion [MAX_VER_LEN];
    DWORD      nPublicLan;
    DWORD      PublicLAN [MAX_PUBLICLAN];
    DWORD      nInterConnect;
    TCHAR      InterConnect [MAX_INTERCONNECT][MAX_INTERCONNECT_LEN];
    DWORD      ClusterServerPortNo;
    DWORD      MonitorWatcherPortNo;
    DWORD      TimeSyncPortNo;
    DWORD      ClientServicePortNo;
} ARM_SERVER_INFO, *LPARM_SERVER_INFO;
```

メンバ

HomeDir	CLUSTERPROのインストールディレクトリのパスを示す文字列
OSVersion	OS名およびバージョンを示す文字列
ARMVersion	CLUSTERPROのバージョンを示す文字列
nPublicLAN	未使用
PublicLAN	未使用
nInterConnect	未使用
InterConnect	未使用
ClusterServerPortNo	未使用
MonitorWatcherPortNo	未使用
TimeSyncPortNo	未使用
ClientServicePortNo	未使用

4.2.2 ARM_RESOURCE_INFO — リソース定義情報

概要

ARM_RESOURCE_INFO構造体は、リソース定義情報を格納します。

形式

```
typedef struct _ARM_RESOURCE_INFO {
    DWORD      Type;
    TCHAR      CurrentServer [MAX_SERVERNAME_LEN];
    TCHAR      PrimaryServer [MAX_SERVERNAME_LEN];
    TCHAR      SecondaryServer [MAX_SERVERNAME_LEN];
    union {
        ARM_RESOURCE_INFO_SCRIPT      Script;
        ARM_RESOURCE_INFO_DISK        Disk;
        ARM_RESOURCE_INFO_DISK_2      Disk2;
        ARM_RESOURCE_INFO_VLAN        VLAN;
        ARM_RESOURCE_INFO_PRINTER     Printer;
        ARM_RESOURCE_INFO_LSW         LSW;
    } Parameter;
} ARM_RESOURCE_INFO, *LPARM_RESOURCE_INFO;
```

メンバ

Type	リソースタイプ	
	ARM_RESTYPE_SCRIPT	スクリプト
	ARM_RESTYPE_DISK	共有ディスク
	ARM_RESTYPE_VLAN	仮想IPアドレス
	ARM_RESTYPE_PRINTER	プリンタ
	ARM_RESTYPE_LSW	回線切替装置
	ARM_RESTYPE_SWDISK	切替ディスクパーティション
	ARM_RESTYPE_SHDISK	共有ディスクパーティション
	ARM_RESTYPE_NPD	CLUSTERPRO内部処理用共有ディスク
Current	現在リソースを所有するサーバのサーバ名を示す文字列(*)	
PrimaryServer	プライマリサーバのサーバ名を示す文字列(*)	
SecondaryServer	セカンダリサーバのサーバ名を示す文字列(*)	
Parameter	リソースタイプ別の固有パラメータを格納する共用体	
— Script	— スクリプト固有パラメータ	
— Disk	— 切替ディスク固有パラメータ (Ver2.x互換)	
— Disk2	— 共有ディスク固有パラメータ (Ver3.0)	
— VLAN	— 仮想IPアドレス固有パラメータ	
— Printer	— プリンタ固有パラメータ	
— LSW	— 回線切替装置固有パラメータ	

(*) リソースタイプがARM_RESTYPE_SHDISK / NPDISKの場合は空文字列となる。

4.2.2.1 ARM_RESOURCE_INFO_SCRIPT — スクリプト固有パラメータ

概要

ARM_RESOURCE_INFO_SCRIPT構造体は、スクリプト固有のパラメータを格納します。

形式

```
typedef struct _ARM_RESOURCE_INFO_SCRIPT {  
    TCHAR    StartScript [MAX_PATH_LEN];  
    TCHAR    StopScript  [MAX_PATH_LEN];  
} ARM_RESOURCE_INFO_SCRIPT, *LPARM_RESOURCE_INFO_SCRIPT;
```

メンバ

StartScript	起動時に実行されるスクリプトのファイル名を示す文字列
StopScript	終了時に実行されるスクリプトのファイル名を示す文字列

4.2.2.2 ARM_RESOURCE_INFO_DISK — 切替ディスク固有パラメータ (Ver 2.x 互換)

概要

ARM_RESOURCE_INFO_DISK構造体は、切替ディスク固有のパラメータ (Ver 2.x互換形式) を格納します。

形式

```
typedef struct _ARM_RESOURCE_INFO_DISK {
    DWORD  ScsiPortNo;
    DWORD  ScsiBus;
    DWORD  nUnit;
    struct {
        DWORD  ScsiId;
        DWORD  nPartition;
        DWORD  Drive [26];
    } Unit [MAX SCSI_ID];
} ARM_RESOURCE_INFO_DISK, *LPARM_RESOURCE_INFO_DISK;
```

メンバ

ScsiPortNo	未使用
ScsiBus	未使用
nUnit	ドライブユニット数 (固定値 1)
Unit	各ドライブユニットの情報 (最初の配列要素のみ使用)
— ScsiId	— 未使用
— nPartition	— パーティション数 (固定値 1)
— Drive	— 各ドライブ (Drive [0] = 'A' ドライブ, ... Drive [25] = 'Z' ドライブ) に割り当てられた、この共有ディスクのパーティション番号 この共有ディスクに関係しないドライブには、0xffffffffがセットされる

4. 2. 2. 3 ARM_RESOURCE_INFO_DISK_2 ― 共有ディスク固有パラメータ (Ver3.0)

概要

ARM_RESOURCE_INFO_DISK_2構造体は、共有ディスク固有のパラメータ (Ver3.0形式) を格納します。

形式

```
typedef struct _ARM_RESOURCE_INFO_DISK_2 {  
    DWORD      LType;  
    TCHAR      Letter [MAX_LETTER_LEN];  
} ARM_RESOURCE_INFO_DISK_2, *LPARM_RESOURCE_INFO_DISK_2;
```

メンバ

LType	未使用
Letter	・ リソースタイプがARM_RESTYPE_SWDISKの場合 → パーティションに割り当てられたドライブ文字 ("A:" ~ "Z:") ・ リソースタイプがARM_RESTYPE_SHDISK / NPDISKの場合 → パーティションに割り当てられたCLUSTERPRO内部識別名

4.2.2.4 ARM_RESOURCE_INFO_VLAN — 仮想 IP アドレス固有パラメータ

概要

ARM_RESOURCE_INFO_VLAN構造体は、仮想IPアドレス固有のパラメータを格納します。

形式

```
typedef struct _ARM_RESOURCE_INFO_VLAN {  
    DWORD    dwAddress;  
} ARM_RESOURCE_INFO_VLAN, *LPARM_RESOURCE_INFO_VLAN;
```

メンバ

dwAddress	仮想IPアドレス（ネットワークバイトオーダー）
-----------	-------------------------

4. 2. 2. 5 ARM_RESOURCE_INFO_PRINTER — プリンタ固有パラメータ

概要

ARM_RESOURCE_INFO_PRINTER構造体は、プリンタ固有のパラメータを格納します。

形式

```
typedef struct _ARM_RESOURCE_INFO_PRINTER {  
    TCHAR Port [MAX_PATH_LEN];  
    TCHAR PrinterName [MAX_PRINTERNAME_LEN];  
    TCHAR SpoolDirectory [MAX_PATH_LEN];  
    TCHAR ShareName [MAX_SHARENAME_LEN];  
} ARM_RESOURCE_INFO_PRINTER, *LPARM_RESOURCE_INFO_PRINTER;
```

メンバ

Port	プリンタが接続されているポート名を示す文字列
PrinterName	プリンタ名を示す文字列
SpoolDirectory	スプールディレクトリのパス名を示す文字列
ShareName	プリンタのシェア名を示す文字列

4. 2. 2. 6 ARM_RESOURCE_INFO_LSW — 回線切替装置固有パラメータ

概要

ARM_RESOURCE_INFO_LSW 構造体は、回線切替装置 (NEC の回線切替装置 (N8591-01/02 V. 24/X. 21回線切替装置、N8545-01/03 V. 24/X. 21回線切替ユニット、N8545-02/04 V. 24/X. 21回線切替拡張ユニット)あるいは、それと同等の回線切替装置)固有のパラメータを格納します。

形式

```
typedef struct _ARM_RESOURCE_INFO_LSW {  
    DWORD    UnitNo;  
    DWORD    LineNo;  
    TCHAR    Port [MAX_PORTNAME_LEN];  
} ARM_RESOURCE_INFO_LSW, *LPARM_RESOURCE_INFO_LSW;
```

メンバ

UnitNo	ユニット番号 (0～15)
LineNo	ライン番号 (1～4)
Port	回線切替装置が接続されているシリアルポート名を示す文字列 (“COM1” , “COM2”)

4.2.3 ARM_EVENT_PARAM — サーバ状態変更通知パラメータ

概要

ARM_EVENT_PARAM構造体は、サーバ状態変更通知に関する付加情報を格納します。

形式

```
typedef struct _ARM_EVENT_PARAM {
    TCHAR  szServerName [MAX_SERVERNAME_LEN];
    union {
        TCHAR  szResourceName [MAX_RESOURCENAME_LEN];
    } EventParam;
} ARM_EVENT_PARAM, *LPARM_EVENT_PARAM ;
```

メンバ

szServerName	状態変更の発生したサーバのサーバ名を示す文字列
EventParam	サーバ状態変更通知固有の付加情報を格納する構造体
— szResourceName	— プリンタ切替通知ARM_EVENT_PRINTER_SWITCHED、回線切替通知ARM_EVENT_LSW_SWITCHEDに関するリソースのリソース名を示す文字列

5 CLUSTERPRO クライアントAPI概要

5.1 CLUSTERPRO クライアント API 機能概要

5.1.1 主な機能

CLUSTERPROクライアントAPIは、CLUSTERPROクライアントの提供する機能を、クライアント上で動作するアプリケーションから利用可能にします。主な機能は以下の通りです。

- * 現在接続しているサーバの動作状態問い合わせ
- * 接続先ネットワークアドレス問い合わせ
- * フェイルオーバーを監視し、フェイルオーバー発生時にはクライアントアプリケーションが再接続に必要な情報を通知
- * クラスタ情報の取得
- * 仮想IPアドレスから、クラスタ名及びプライマリサーバ名取得
- * 論理サービスが設定されたサーバの一覧をクライアントから取得

5.1.2 CLUSTERPROクライアントAPIを使用するメリット

CLUSTERPROクライアントAPIを使用しなくても、仮想IPアドレスを使用することで、クラスタに接続し、フェイルオーバー時に接続先のサーバを切り替えることが可能です。しかし、CLUSTERPROクライアントAPIを使用することで、以下のような利点があります。

- * フェイルオーバーの開始や終了の通知が受けられるため、フェイルオーバーをきっかけとして、コネクションの切断及び再接続を行うクライアントAPを作成することが可能です。
- * サーバの動作状態を問い合わせることができるため、サーバが動作していない場合などにユーザーに通知するなどの処理を行うことが可能です。
- * 名前付きパイプなど、NetBIOSインタフェースを用いたAPで、サーバの切替を行うことが可能です。

5.1.3 対応言語

CLUSTERPROクライアントAPIは、Microsoft Visual C++ Ver4.2以降、およびMicrosoft Visual Basic Ver5以降に対応しています。

Visual Basic向けクライアントAPIは、CLUSTERPRO Ver5.0からの機能です。

5.2 CLUSTERPRO Ver3.0 からの変更点及び互換性留意事項

本節の内容は、Visual C++向けCLUSTERPROクライアントAPIにのみ当てはまります。
Visual Basic向けCLUSTERPROクライアントAPIは、CLUSTERPRO Ver5.0以降の対応です。

5.2.1 サーバ数の増加とフェイルオーバーグループの概念導入

CLUSTERPRO Ver3.0からの大きな変更点はサーバ数の増加と、フェイルオーバーグループの概念を明示的に取り入れた点です。

CLUSTERPRO Ver3.0ではクラスタを構成するサーバ数は2サーバ固定であったのに対して、CLUSTERPRO Ver4.xではサーバ数が2～4に対応しています。

CLUSTERPRO Ver5.0と6.0では、**Enterprise Edition**はサーバ数が2～16に対応、**Standard Edition**および**Lite!**はサーバ数は2サーバ固定です。

また、CLUSTERPRO Ver3.0では、フェイルオーバーの単位は、それぞれのサーバをプライマリとするクラスタ資源の集合でした。CLUSTERPRO Ver4.x以降ではフェイルオーバーの単位は、ユーザ定義可能なフェイルオーバーグループに変更されました。

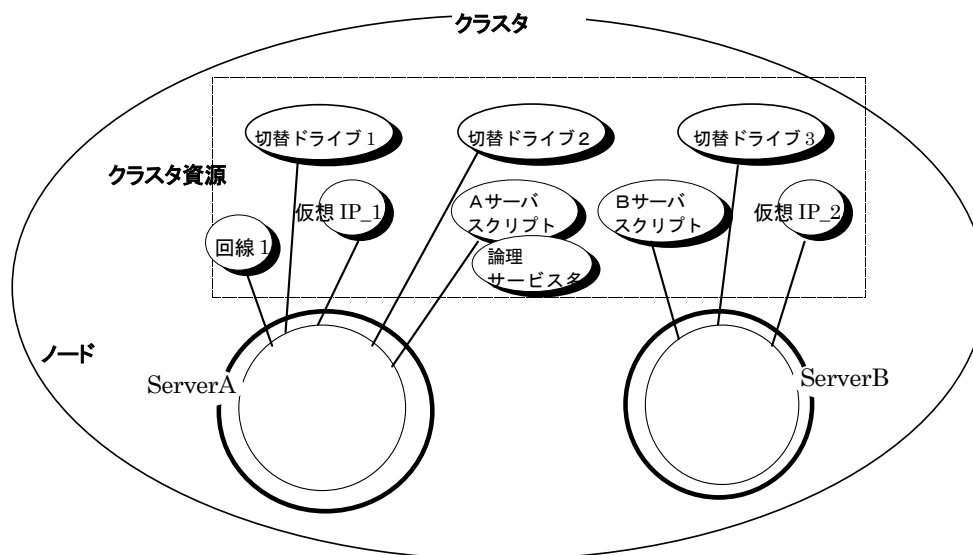
* クラスタ資源

H/W資源やH/Wを使用するためにつけられた論理的な名前でもフェイルオーバー可能な資源。かならずフェイルオーバーグループに所属します。

* フェイルオーバーグループ

1つの独立した業務を実現するためのクラスタ資源の集合。フェイルオーバーポリシー（どのサーバで実行可能か、サーバ間のフェイルオーバー順序）を持ちます。

(CLUSTERPRO Ver3.0)



* サーバ数は2

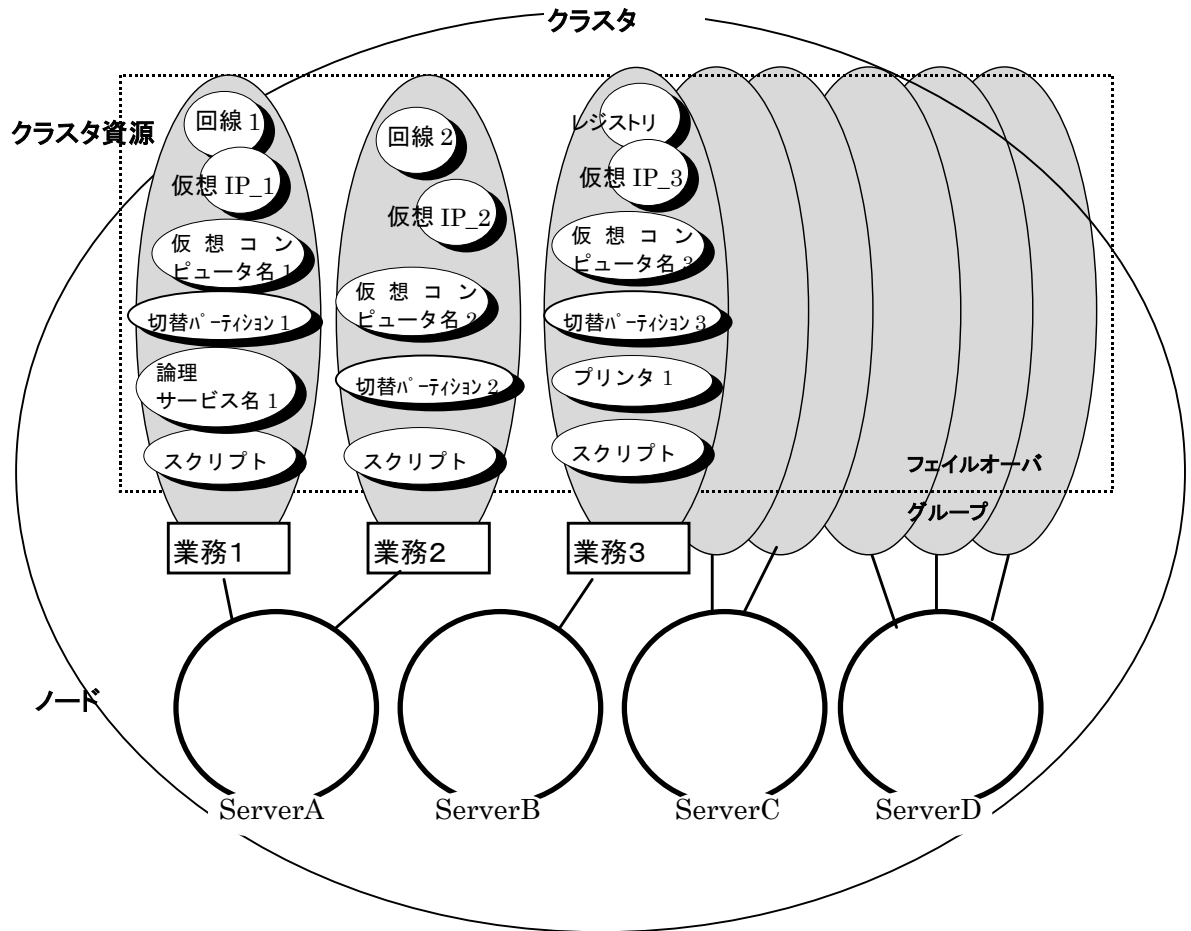
* 各クラスタ資源ごとにプライマリサーバを指定

* 1つのサーバをプライマリとするフェイルオーバーグループは1つ

* フェイルオーバーグループを代表する名前は明示的にはなし

* クライアントAPIではプライマリサーバ名をフェイルオーバーグループを代表する名前として使用

(CLUSTERPRO Ver4. x以降)



- * サーバ数は2～4
- * 1つのサーバをプライマリとするフェイルオーバーグループは複数
- * フェイルオーバーグループを代表する名前としてフェイルオーバーグループ名を明示的に定義
- * フェイルオーバーグループ毎にフェイルオーバーポリシーを定義

CLUSTERPROクライアントAPIでは、状態取得や監視の対象となるのはフェイルオーバーグループです。そのため、CLUSTERPRO Ver3. 0のクライアントAPIでは、状態取得や監視の対象となるフェイルオーバーグループを特定するためのキーとして、クラスタ名とプライマリサーバ名を使用していました。CLUSTERPRO Ver4. x以降では、クラスタ名とプライマリサーバ名の代わりにクラスタ名とフェイルオーバーグループ名を状態取得や監視のキーとします。フェイルオーバーグループ名は英数字とハイフンとアンダースコアによって構成される15文字以内（末尾のNUL文字を含まず）の文字列です。

CLUSTERPRO Ver3. 0で提供していたAPIのうち、プライマリサーバ名を引数に取るものは、全てフェイルオーバーグループ名に変更されます。

5.2.2 Ver3.0 対応アプリケーションサービスを使用するための注意

Ver3.0互換運用では、CLUSTERPRO Ver3.0に対応しているアプリケーション・サービスは、プログラムを変更することなくそのまま動作することが可能です。

一方、片方向スタンバイクラスタ、双方向スタンバイクラスタ、パラレルクラスタで正しく動作するためにプログラムの変更が必要となる場合があります。

詳細については、「5.2.8 互換性留意事項」を参照してください。

5.2.3 追加されたクラスタイベントのクライアント API イベントへのマッピング

CLUSTERPRO Ver3.0ではクライアントAPIのイベントの種類は以下の4種類のみでした。

- * ARMCLAPI_EVENT_BOOT (サーバ起動)
- * ARMCLAPI_EVENT_SHUTDOWN (シャットダウン)
- * ARMCLAPI_EVENT_FAILOVER_START (フェイルオーバー開始)
- * ARMCLAPI_EVENT_FAILOVER_END (フェイルオーバー完了)

CLUSTERPRO Ver4.x以降では、機能追加により、上記に加えて、以下のような場合にクラスタイベントが発生し、フェイルオーバーグループの状態や接続先が変化します。

- * アプリケーション・サービス異常によるフェイルオーバーグループ単位のフェイルオーバー
- * ユーザの指示によるフェイルオーバーグループの移動
- * ユーザの指示によるフェイルオーバーグループの起動・停止
- * サーバクライアント間のネットワークの監視によるネットワークの切り替え
- * サーバクライアント間のネットワークの監視によるネットワーク異常・回復の検出

これらの新しいクラスタイベントをCLUSTERPRO Ver3.0で定義した4種類のクライアントAPIイベントにマッピングします。

クライアントAPIイベント	クラスタイベント
ARMCLAPI_EVENT_FAILOVER_START	・サーバダウンによるフェイルオーバーの開始 ・アプリケーション・サービス異常によるフェイルオーバーの開始 ・フェイルオーバーグループ移動の開始 ・ネットワークの切り替え開始
ARMCLAPI_EVENT_FAILOVER_END	・サーバダウンによるフェイルオーバーの完了 ・アプリケーション・サービス異常によるフェイルオーバーの開始 ・フェイルオーバーグループ移動の完了 ・ネットワークの切り替え完了
ARMCLAPI_EVENT_SHUTDOWN	・クラスタシャットダウンによるフェイルオーバーグループの停止 ・ユーザの指示によるフェイルオーバーグループの停止 ・ネットワークの異常検出
ARMCLAPI_EVENT_BOOT	・サーバ起動によるフェイルオーバーグループの起動 ・ユーザの指示によるフェイルオーバーグループの起動 ・ネットワークの回復検出

このマッピングにより、CLUSTERPRO Ver3.0に対応しているアプリケーション・サービスは、クラスタイベントの追加を意識する必要はありません。

5.2.4 仮想コンピュータ名のサポート

CLUSTERPRO Ver4. x以降では、仮想コンピュータ名をサポートしています。仮想コンピュータ名を使用すれば、クライアントAPIを使用しなくても、フェイルオーバー時の切り替えは自動的に行われます。

クライアントAPIを使用すれば、仮想コンピュータ名が属するフェイルオーバーグループの状態取得や監視を行うことで、フェイルオーバー完了やグループ起動のタイミングでサーバに再接続するなど、よりクラスタを意識したアプリケーションを作成することができます。そこで、CLUSTERPRO Ver4. x以降では、仮想コンピュータ名から、監視や状態取得に必要なクラスタ名とフェイルオーバーグループ名を取得するAPI (ArmClientGetGroupFromVirtualComputer) を新規に提供します。また、仮想コンピュータ名を列挙するAPI (ArmClientEnumVirtualComputer) を提供します。

5.2.5 フローティング IP アドレスのサポート

CLUSTERPRO Ver4. 2 以降では、フローティング IP アドレスをサポートしています。CLUSTERPRO クライアント API では、フローティング IP アドレスを、仮想 IP アドレスの一形態として扱います。

5.2.6 CLUSTERPRO Ver3.0 との差分一覧

- * 4. x … CLUSTERPRO Ver4. x以降で提供するAPIです。
- * 無変更 … CLUSTERPRO Ver3.0から全く変更されていないAPIです。
- * S→G … プライマリサーバ名がフェイルオーバーグループ名に変更されたAPIです。
- * 互換のみ… 以前のバージョンからの互換性のために提供されているAPIです。

	4. x	無変更	S→G	互換のみ
ArmClientIntializeApiEx	—	○	—	—
ArmClientTerminateApiEx	—	○	—	—
ArmClientGetVersionInfo	—	○	—	—
ArmClientGetIPAddr	—	—	○	—
ArmClientGetComputerName	—	—	○	—
ArmClientGetGroupStatus	○	—	—	—
ArmClientGetServerStatus	—	—	○	○
ArmClientGetGroupStatusEx	○	—	—	—
ArmClientGetServerStatusEx	—	—	○	○
ArmClientStartWatchFailoverEx	—	—	○	—
ArmClientCancelWatchFailoverEx	—	○	—	—
WatchFailoverEx	—	○	—	—
ArmClientGetClusterServerStatus	○	—	—	—
ArmClientEnumCluster	○	—	—	—
EnumClusterProc	○	—	—	—
ArmClientEnumServer	—	○	—	—
EnumServerProc	—	○	—	—
ArmClientGetClusterFromServer	—	—	○	○
ArmClientGetClusterFromGroup	○	—	—	—
ArmClientGetClusterFromClusterServer	○	—	—	—
ArmClientEnumServerFromGroup	○	—	—	—
EnumServerFromGroupProc	○	—	—	—
ArmClientGetGroupFromVirtualIP	○	—	—	—
ArmClientGetServerFromVirtualIP	—	—	○	○
ArmClientGetClusterNameFromVirtualIP	—	○	—	○
ArmClientGetIPAddrFromVirtualIP	—	○	—	—
ArmClientEnumVirtualIP	○	—	—	—
EnumViaturalIPProc	○	—	—	—
ArmClientGetGroupFromVirtualComputer	○	—	—	—
ArmClientEnumVirtualComputer	○	—	—	—
EnumVirtualComputerProc	○	—	—	—
ArmClientEnumGroupFromLogicalService	○	—	—	—
ArmClientEnumServerFromLogicalService	—	—	○	○
EnumGroupFromLogicalServiceProc	○	—	—	—
ArmClientGetHeartbeatTimeout	—	○	—	—
ArmClientGetMaterServer	—	○	—	—
ArmClientCheckClusterServer	—	○	—	—
ArmClientGetServerName	—	○	—	—
ArmClientGetClusterName	—	○	—	—

5.2.7 マクロ定義

armcl2.hで定義されているマクロ定義をARMCLAPI_で開始されるように統一しました。CLUSTERPRO Ver3.0以前の定義もそのまま残してありますので、CLUSTERPRO Ver3.0用に作成したアプリケーションの再コンパイルには影響ありません。

CLUSTERPRO Ver4.xクライアントAPIを使用して、新規に作成するアプリケーションは、4.x用のマクロ定義を使用してください。

	CLUSTERPRO Ver3.0 API	CLUSTERPRO Ver4.x API
エラー定義	ARMERR_XXXX	ARMCLAPI_ERR_XXXX
イベント	ARM_EVENT_XXXX	ARMCLAPI_EVENT_XXXX
フェイルオーバーグループ動作状態	ARM_SERVER_STATUS_XXXX	ARMCLAPI_GROUP_STATUS_XXXX
ローカルホストタイプ	ARM_HOSTTYPE_XXXX	ARMCLAPI_HOSTTYPE_XXXX
フェイルオーバー構造体	ARM_FAILOVER_INFO_EX	ARMCLAPI_FAILOVER_INFO
ローカルクラスター指定	ARM_LOCAL_CLUSTER	ARMCLAPI_LOCAL_CLUSTER

5.2.8 互換性留意事項

(1) バイナリ互換性

CLUSTERPRO Ver3.0とCLUSTERPRO Ver4.x以降では、クライアントAPIとして提供するダイナミックリンクライブラリのファイル名は同一です。また、Ver3.0で提供した関数はすべてVer4.x以降でも提供します。そのため、CLUSTERPROをVer3.0からVer4.x以降にアップグレードした場合、Ver3.0対応クライアントアプリケーションは、再コンパイルの必要はありません。

ただし、片方向スタンバイクラスタ、双方向スタンバイクラスタ、パラレルクラスタでは、機能的な問題により、ソース修正が必要となる場合があります。(4)片方向スタンバイクラスタ、双方向スタンバイクラスタ、パラレルクラスタでの変更点をご覧ください。

(2) ソースファイル互換性

4.xクライアントAPI開発環境で提供するヘッダファイル及びスタティックリンクライブラリはVer3.0の上位互換となっています。そのため、Ver3.0用に開発したアプリケーションは、ソースファイルを変更することなく、4.xクライアントAPI開発環境で作成することができます。

ただし、片方向スタンバイクラスタ、双方向スタンバイクラスタ、パラレルクラスタでは、機能的な問題により、ソース修正が必要となる場合があります。(4)片方向スタンバイクラスタ、双方向スタンバイクラスタ、パラレルクラスタでの変更点をご覧ください。

(3) 開発環境の制約

4.xクライアントAPI開発環境で作成したアプリケーションがCLUSTERPRO Ver3.0の環境で動作することは保証しません。

CLUSTERPRO Ver3.0で正常に動作するアプリケーションは、Ver3.0クライアントAPI開発環境でコンパイルされた場合でも、4.xクライアントAPI開発環境で再コンパイルされた場合でも、Ver3.0互換運用で正常に動作することを保証します。

上記3点をまとめると、以下のようになります。

	Ver3.0対応AP (3.0開発環境で作成)	Ver3.0対応APを 4.x開発環境で 再コンパイル	4.x開発環境で 作成したAP
CLUSTERPRO Ver3.0	○	×	×
CLUSTERPRO Ver4.x以降 (Ver3.0互換運用)	○	○	○
CLUSTERPRO Ver4.x以降 (片方向スタンバイクラスタ、 双方向スタンバイクラスタ、パラレルク ラスタ)	△	△	○

(4) 片方向スタンバイクラスタ、双方向スタンバイクラスタ、パラレルクラスタでの変更点

片方向スタンバイクラスタ、双方向スタンバイクラスタ、パラレルクラスタでは、クラスタのサーバ数が2台固定で無くなったことと、プライマリサーバ名がフェイルオーバーグループ名に変更されたことが最もインパクトのある変更点です。

サーバ列挙関数(ArmClientEnumServer)を使用しており、サーバ数が2台固定であることを期待しているアプリケーションは、片方向スタンバイクラスタ、双方向スタンバイクラスタ、パラレルクラスタで正常に動作するためにはプログラムの変更が必要となります。

また、プライマリサーバ名が実際のサーバと同じ名前であることを前提とした処理を行っているアプリケーションは、プライマリサーバ名がフェイルオーバーグループ名に変更されたことで、片方向スタンバイクラスタ、双方向スタンバイクラスタ、パラレルクラスタで正常に動作するためにはプログラムの変更が必要となります。例えば以下のような処理が該当します。

- * プライマリサーバ名としてAPIの引数としている名前と同じ名前ですサーバと通信する処理
 サーバとの通信にはArmClientGetComputerNameで得られるコンピュータ名を用いてください。
- * プライマリサーバ名と接続先コンピュータ名の名前を比較するような処理
 フェイルオーバーグループが最も優先度が高いサーバで動作しているかどうかは、ArmClientEnumServerFromGroupによって得られるプライマリサーバ名と、ArmClientGetComputerNameによって得られるコンピュータ名を比較してください。
- * ArmClientEnumServerで列挙したサーバ名をプライマリサーバ名としてAPIに渡す処理
 ArmClientEnumGroupを使用してください。
 一方、以下のような方法でプライマリサーバ名を取得していたアプリケーションは、基本的にはプログラムの変更や再コンパイルの必要はありません。そのままシングルスタンバイ運用、双方向スタンバイクラスタ、パラレルクラスタで使用することが可能です。
- * プライマリサーバ名をユーザからの入力情報としている場合
 マニュアルの改訂で、プライマリサーバ名の代わりにフェイルオーバーグループ名をユーザ入力情報とすることで、そのまま片方向スタンバイクラスタ、双方向スタンバイクラスタ、クラスタ運用に対応できます。ただし、画面から入力を促す場合などで表示されるメッセージの変更が必要なケースがあります。
- * プライマリサーバ名は中間情報としてのみ使用している場合
 例えば仮想IPアドレスをユーザ入力としており、ArmClientGetPrimaryServerFromVirtualIPで得られたクラスタ名及びプライマリサーバ名を使用して、状態取得や監視を行う場合が相当します。フェイルオーバーグループ名は英数字とハイフンとアンダースコアによって構成される15文字以内（末尾のNUL文字を含まず）の文字列ですので、文字列サイズも変更不要です。

5.3 開発環境 (Visual C++)

5.3.1 CLUSTERPRO Ver4.xクライアントAPI開発キット

CLUSTERPRO Ver5.0以降では、16bit版は、実行環境（ダイナミックリンクライブラリ）のみのご提供になります。

5.3.1.1 32bit版

- armcl2.h ヘッダファイルです。CLUSTERPROクライアントAPIを使用するソースファイルにインクルードしてください。
- armcl232.lib インポートライブラリです。CLUSTERPROクライアントAPIを使用するプログラムをリンクする際に使用してください。armcl2st.libとは排他使用です。
- armcl2st.lib インポートライブラリです。CLUSTERPROクライアントAPIを使用するプログラムをリンクする際に使用してください。armcl2.libとは排他使用です。初期化関数 (ArmClientInitializeApiEx) 内で、CLUSTERPROクライアントAPIダイナミックリンクライブラリがロードできるか否かをLoadLibray関数を使用してチェックしています。ダイナミックリンクライブラリのファイル名は32bit版はarmcl232.dll, 16bit版はarmcl216.dllです。CLUSTERPROクライアントがインストールされていない場合には、ダイナミックリンクライブラリのロードに失敗するため、ArmClientInitializeApiExはエラーを返却します。この場合、CLUSTERPROクライアントAPIの機能は使用できませんが、アプリケーションの継続には支障ありません。
- armcl232.dll ダイナミックリンクライブラリです。

5.3.1.2 16bit版

- armcl216.dll ダイナミックリンクライブラリです。

5.3.2 CLUSTERPRO Ver1.0 互換クライアント API 開発キット

CLUSTERPRO Ver1.0用に開発したクライアントアプリケーションをCLUSTERPRO Ver4.x以降のVer3.0互換運用で使用するための開発キットです。互換性のために提供するものですので、新規にCLUSTERPRO Ver4.x以降で動作するクライアントアプリケーションを開発する場合にはCLUSTERPRO Ver4.xクライアントAPI開発キットを使用してください。CLUSTERPRO Ver1.0用に開発したクライアントアプリケーションをCLUSTERPRO Ver4.x以降で動作させるためには再コンパイル再リンクは不要です。片方向スタンバイクラスタ、双方向スタンバイクラスタ、パラレルクラスタでの動作は保証しません。

CLUSTERPRO Ver5.0以降では、16bit版は、実行環境（ダイナミックリンクライブラリ）のみのご提供になります。

5.3.2.1 32bit版

- | | |
|--------------|--|
| armclapi.h | ヘッダファイルです。CLUSTERPROクライアントAPIを使用するソースファイルにインクルードしてください。 |
| armclapi.lib | インポートライブラリです。CLUSTERPROクライアントAPIを使用するプログラムをリンクする際に使用してください。armclst.libとは排他使用です。 |
| armclst.lib | インポートライブラリです。CLUSTERPROクライアントAPIを使用するプログラムをリンクする際に使用してください。armclapi.libとは排他使用です。初期化関数(ArmClientInitializeAPI)内で、armclapi.dllがロードできるか否かをLoadLibrary関数を使用してチェックしています。armclapi.dllがインストールされていない場合、ArmClientInitializeAPIはエラーを返却します。この場合、CLUSTERPROクライアントAPIの機能は使用できませんが、アプリケーションの実行には支障ありません。 |
| armclapi.dll | ダイナミックリンクライブラリです。 |

5.3.2.2 16bit版

- | | |
|--------------|-------------------|
| armclapi.dll | ダイナミックリンクライブラリです。 |
|--------------|-------------------|

5.3.3 CLUSTERPROクライアントAPIサンプルプログラム

TCP/IPを使用し、サーバと通信するプログラムのサンプルです。サンプルプログラムの起動時に、現在動作しているサーバと接続します。また、フェイルオーバー時には、フェイルオーバー監視のコールバック関数が呼び出されます。サンプルプログラムでは、コールバックの内部で切り替わったサーバへ再接続を行います。

- * FailoverProc
フェイルオーバー発生・完了時のコールバック関数です。サーバへの再接続を行います。CLUSTERPROクライアントAPIからフェイルオーバー発生・完了時に呼び出されます。
- * ClientInitialize
APIの初期化、サーバへの接続、フェイルオーバー監視コールバック関数の登録を行います。アプリケーション起動時にInitInstance関数などから呼び出します。
- * ClientTerminate
APIの終了処理です。APの終了時に呼び出します。

```
/*
 * CLUSTERPRO Client API サンプルプログラム
 */

#include <windows.h>
#include <stdio.h>
#include "armcl2.h"

static DWORD Id = 0;
static DWORD ApiHandle = 0;

static BOOL ConnectFlag = FALSE;

/*
 * サーバとの接続を行う
 */

VOID ConnectToServer(DWORD IpAddr)
{
    union {
        DWORD IpAddr;
        char IpAddrc[4];
    } IpConv;

    if (ConnectFlag) {
        return;
    }

    IpConv.IpAddr = IpAddr;

    printf("サーバ%d. %d. %d. %dに接続します¥n",
        IpConv.IpAddrc[0],
        IpConv.IpAddrc[1],
        IpConv.IpAddrc[2],
```

```

        IpConv. IpAddrc[3]);

    /*
     * ここにサーバ接続の実処理を記述してください。
     */

    /*
     * フラグのセット
     */

    ConnectFlag = TRUE;
}

/*
 * サーバとの接続を切断する。
 */

VOID DisconnectToServer (VOID)
{
    if (ConnectFlag == FALSE) {
        return;
    }

    printf("サーバとの接続を切断します\n");

    /*
     * ここにサーバ切断の実処理を記述してください。
     */

    /*
     * フラグのリセット
     */

    ConnectFlag = FALSE;
}

/*
 * フェイルオーバーグループの監視コールバック関数
 */

VOID CALLBACK
#ifdef WIN32
_export
#endif
FailoverProc(
    DWORD event,
    DWORD watch,
    LPARMCLAPI_FAILOVER_INFO info
)
{

```

```

switch(info->Event) {
case ARMCLAPI_EVENT_BOOT:          /* 起動 */

    /*
     * サーバと接続し、通信を開始する
     */

    ConnectToServer(info->DstIP);

    return;

case ARMCLAPI_EVENT_SHUTDOWN:      /* シャットダウン */

    /*
     * サーバとの接続を切断する
     */

    DisconnectToServer();

    return;

case ARMCLAPI_EVENT_FAILOVER_START: /* フェイルオーバー発生 */

    /*
     * サーバとの通信を中断する
     */

    DisconnectToServer();

    return;

case ARMCLAPI_EVENT_FAILOVER_END: /* フェイルオーバー完了 */

    /*
     * サーバに再接続し、通信を再開する
     */

    ConnectToServer(info->DstIP);

    return;
}
}

/*
 * 初期化处理
 * 1. クライアントAPIの初期化
 * 2. フェイルオーバーグループ監視の開始
 * 3. 起動中であれば、サーバへの接続処理
 */

BOOL ClientInitialize(LPSTR ClusterName, LPSTR FailoverGroupName)

```

```

{
    /*
     * Arm Client Initialize
     */

    DWORD ret;
    DWORD IpAddr;
    DWORD StatusEx;

    /*
     * クライアントAPIの初期化
     */

    ApiHandle = ArmClientInitializeApiEx();

    if (ApiHandle == ARMCLAPI_ERROR_HANDLE) { /* 初期化に失敗した */
        ret = GetLastError();

        /*
         * エラー処理
         */

        fprintf(stderr, "ArmClientInitializeApiEx failed %lx\n", ret);

        return FALSE;
    }

    /*
     * フェイルオーバー監視を開始する
     */
    ret = ArmClientStartWatchFailoverEx(ApiHandle,
                                         ClusterName,
                                         FailoverGroupName,
                                         (PWATCHFAILOVERPROC) FailoverProc,
                                         &Id,
                                         &StatusEx);

    if (ret == FALSE) {
        ret = GetLastError();

        /*
         * エラー処理
         */

        fprintf(stderr, "ArmClientStartWatchFailoverEx failed %lx\n", ret);

        ret = ArmClientTerminateApiEx(ApiHandle);
        return FALSE;
    }

    if (StatusEx == ARMCLAPI_GROUP_STATUS_STOPPED ||
        StatusEx == ARMCLAPI_GROUP_STATUS_FAILOVER) {

```

```

        /*
        * フェイルオーバーグループは動作していない
        */

        return TRUE;
    }
    else {

        /*
        * フェイルオーバーグループは動作中
        */

        /*
        * 現在稼働しているサーバのIPアドレスを問い合わせ
        */

        ret = ArmClientGetIPAddr(ClusterName, FailoverGroupName, &IpAddr);
        if (ret == FALSE) {
            ret = GetLastError();

            /*
            * エラー処理
            */

            fprintf(stderr, "ArmClientGetIPAddr failed %lx\n", ret);
            ArmClientTerminateApiEx(ApiHandle);
            return FALSE;
        }

        /*
        * 得られたIPアドレスを用いてサーバと接続
        */

        ConnectToServer(IpAddr);

    }
    return TRUE;
}

/*
* 終了処理
* 1. 接続中であれば、サーバへの切断処理
* 2. フェイルオーバーグループ監視の停止
* 3. クライアントAPIの終了
*/

BOOL ClientTerminate(VOID)
{
    /*
    * クライアントAPIの終了処理
    */

```

```

    DisconnectToServer();
    ArmClientCancelWatchFailoverEx(Id);
    ArmClientTerminateApiEx(ApiHandle);
    return TRUE;
}

/*
 * メイン関数
 * 1. コマンドラインチェック
 * 2. 初期化関数をコール
 * 3. リターン入力を待つ（この間フェイルオーバーグループを監視している）
 * 4. 終了関数をコール
 */

int main(int argc, char **argv)
{
    char buf[256];

    if (argc != 3) {
        fprintf(stderr, "使用法: %s クラスタ名 フェイルオーバーグループ名\n",
            argv[0]);
        exit(1);
    }

    fprintf(stderr, "リターンを押すと終了します\n");
    if (ClientInitialize(argv[1], argv[2]) == FALSE) {
        fprintf(stderr, "初期化に失敗しました\n");
        exit(1);
    }
    gets(buf);
    ClientTerminate();
    return 0;
}

```

5.4 開発環境 (Visual Basic)

5.4.1 CLUSTERPRO クライアントAPI開発キット

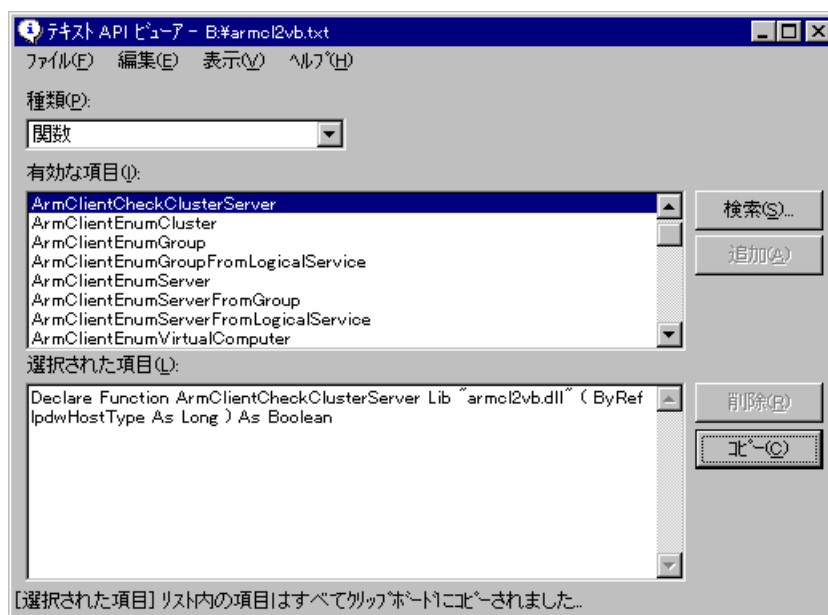
5.4.1.1 リリースモジュール

Visual Basic 向け 5.4.1 CLUSTERPRO クライアント API 開発キットは、32bit 版 API のみのリリースとなります。16bit 版 API はサポートしていません。

armcl2vb.bas VBリンク用インクルードファイルです。
ARMCL2VB.TXT APIビューア (APILOAD) 読み込み用ARMCL2VB.BASです。
armcl2vb.dll ダイナミックリンクライブラリです。

5.4.1.2 API ビューア (APILOAD) の利用について

Visual Basicには、APIビューア (APILOAD) が用意されています。このコマンドは、規定の書式の範囲で記述されたVisual Basicソース (*.txtファイル) からデータ、型、関数を別々に抽出して、クリップボードにコピーしてエディタなどでソースに張り付けしたり、データベースに登録することで、開発工数を削減する効果があります。ARMCL2VB.TXTを読み込んで利用してください。



5.4.2 CLUSTERPROクライアントAPIサンプルプログラム

```
rem      CLUSTERPRO Client API サンプルプログラム(Visual Basic用)

Attribute VB_Name = "Module1"
Public Const id      As Long = 0
Public ApiHandle      As Long
Public ConnectFlag As Long
Public Const logfile As String = ".¥¥armvbtst.log"
Public logid  As Long
Public fret As Long
Public IpAddr  As Long
Public StatusEx      As Long
Public title  As String
Public str As String
Public tmpstr  As String
Public ind As Byte

Declare Function WritePrivateProfileString Lib "kernel32" Alias _
    "WritePrivateProfileStringA" (ByVal lpApplicationName As String, _
    ByVal lpKeyName As Any, ByVal lpString As Any, ByVal lpFileName As String) As Long

rem      サーバとの接続を行う。

Function ConnectToServer(ByVal IpAddr As Long)
    Dim IpClass(4) As Byte
    For ind = 0 To 3
        IpClass(ind) = IpAddr Mod 256
        IpAddr = (IpAddr - IpClass(ind)) / 256
    Next ind
    If ConnectFlag Then Exit Function
    str = Format(IpClass(0), "000")
    For ind = 1 To 3
        tmpstr = Format(IpClass(ind), "000")
        str = str & "." & tmpstr
    Next ind
    str = "サーバIP:" & str & "と接続を行います。"
    title = Format(logid, "00000")
    ret = WritePrivateProfileString("logging.list", title, str, logfile)
    logid = logid + 1

rem      ここにサーバ接続の実処理を記述してください。

rem      フラグのセット

    ConnectFlag = ARMTRUE
    ConnectToServer = ret
End Function

rem      サーバとの接続を切断する。
```

```

Function DisconnectToServer()
    DisconnectToServer = 0
    If ConnectFlag = False Then Exit Function
    str = Format(logid, "00000")
    ret = WritePrivateProfileString("logging.list", str, _
        "サーバとの接続を切断します。", logfile)
    logid = logid + 1

rem      ここにサーバ切断の実処理を記述してください。

rem      フラグのリセット

    ConnectFlag = ARMFALSE
    DisconnectToServer = ret
End Function

Function FailoverProc(armevent As Long, armwatch As Long, _
    arminfo As ARMCLAPI_FAILOVER_INFO) As Long
    Select Case info.Event
        Case ARMCLAPI_EVENT_BOOT:rem      起動

rem      サーバと接続し、通信を開始する。

        ret = ConnectToServer(info.DstIP)
        Exit Function

        Case ARMCLAPI_EVENT_SHUTDOWN:rem      シャットダウン

rem      サーバとの接続を切断する。

        ret = DisconnectToServer()
        Exit Function

        Case ARMCLAPI_FAILOVER_START:rem      フェイルオーバー発生

rem      サーバとの通信を中断する。

        ret = DisconnectToServer()
        Exit Function

        Case ARMCLAPI_FAILOVER_END:rem      フェイルオーバー完了

rem      サーバに再接続し、通信を再開する。

        ret = ConnectToServer(info.DstIP)
        Exit Function
        Rem End to Failover
    End Select
End Function

rem      初期化处理
rem      1. クライアントAPIの初期化

```

```

rem      2. フェイルオーバーグループ監視の開始
rem      3. 起動中であれば、サーバへの接続処理

Function ClientInitialize(ByRef ClusterName As String, _
    ByRef FailoverGroupName As String) As Long

rem      Arm Client Initialize
rem      クライアントAPIの初期化

    ApiHandle = ArmClientInitializeApiEx()
    If ApiHandle = ARMCLAPI_ERROR_HANDLE Then
        fret = Err.LastDllError

rem      エラー処理

        str = Format(logid, "00000")
        ret = WritePrivateProfileString("logging.list", str, _
            "ArmClientInitializeApiEx():Failed.", logfile)
        logid = logid + 1
        Exit Function
    End If

rem      フェイルオーバー監視を開始する。

    fret = ArmClientStartWatchFailoverEx(ApiHandle, ClusterName, _
        FailoverGroupName, AddressOf FailoverProc, id, StatusEx)
    If fret = ARMFALSE Then
        fret = Err.LastDllError

rem      エラー処理

        str = Format(logid, "00000")
        ret = WritePrivateProfileString("logging.list", str,
"ArmClientStartWatchFailoverEx():Failed.", logfile)
        logid = logid + 1
        ret = ArmClientTerminateApiEx(ApiHandle)
        Exit Function
    End If
    If (StatusEx = ARMCLAPI_GROUP_STATUS_STOPPED) _
        Or (StatusEx = ARMCLAPI_GROUP_STATUS_FAILOVER) Then

rem      フェイルオーバーグループは動作していない。

        str = Format(logid, "00000")
        ret = WritePrivateProfileString("logging.list", str, _
            "ArmClientStartWatchFailoverEx():Failed(GROUP: _
            " & FailoverGroupName & "is STOPPED or during FAILOVER).", logfile)
        logid = logid + 1
        ClientInitialize = ARMTRUE
        Exit Function
    Else

```

```

rem      フェイルオーバーグループは動作中。

rem      現在稼動しているサーバのIPアドレスを問い合わせ。

      fret = ArmClientGetIPAddr(ClusterName, FailoverGroupName, IpAddr)
      If fret = ARMFALSE Then
          fret = Err.LastDllError

rem      エラー処理

      str = Format(logid, "00000")
      ret = WritePrivateProfileString("logging.list", str, _
          "ArmClientStartWatchFailoverEx():Failed(Communication).", _
          logfile)
      logid = logid + 1
      ret = ArmClientTerminateApiEx(ApiHandle)
      ClientInitialize = ARMFALSE
      Exit Function
      End If

rem      得られたIPアドレスを用いてサーバと接続。

      ConnectToServer (IpAddr)
      End If
      ClientInitialize = ARMTRUE
      str = Format(logid, "00000")
      ret = WritePrivateProfileString("logging.list", str, _
          "ARM Client Initialized,for Server:" & ClusterName & ".", logfile)
      logid = logid + 1
      End Function

rem      終了処理
rem      1. 接続中であれば、サーバへの切断処理
rem      2. フェイルオーバーグループ監視の停止
rem      3. クライアントAPIの終了

Function ClientTerminate() As Long

rem      クライアントAPIの終了処理

      DisconnectToServer
      ArmClientCancelWatchFailoverEx (id)
      ArmClientTerminateApiEx (ApiHandle)
      ClientTerminate = ARMTRUE
      str = Format(logid, "00000")
      ret = WritePrivateProfileString("logging.list", str, _
          "ARM Client Terminated.", logfile)
      logid = logid + 1
      End Function

```

6 クライアントAPI詳細 (Visual C++)

6.1 関数

6.1.1 APIの初期化と終了

CLUSTERPROクライアントAPIを使用するクライアントアプリケーションは、他のCLUSTERPROクライアントAPI関数を使用する前に必ずArmClientInitializeApiEx関数を実行してCLUSTERPROクライアントAPIの初期化を行わなければなりません。また、ArmClientInitializeApiEx関数を実行した場合には、アプリケーションを終了する前に、ArmClientTerminateApiEx関数を実行して、CLUSTERPROクライアントAPIの終了処理を行わなければなりません。

ArmClientInitializeApiExとArmClientTerminateApiExは、同一プロセスから、複数独立に実行することが可能です。

6.1.1.1 ArmClientInitializeApiEx — CLUSTERPRO クライアント API の初期化

ARM1.1 より変更なし

概要

CLUSTERPROクライアントAPIの初期化、資源の確保を行い、CLUSTERPROクライアントAPIを使用するためのハンドルを返却します。

形式

DWORD ArmClientInitializeApiEx(VOID)

パラメータ

なし

戻り値

ARMCLAPI_ERROR_HANDLE

関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

上記以外

CLUSTERPROクライアントAPIハンドル

拡張エラー情報

ARMCLAPI_ERR_CANNOT_LOAD_DLL

CLUSTERPRO クライアント API DLLをロードできません。

ARMCLAPI_ERR_CANNOT_CONNECT

CLUSTERPROクライアントに接続できません。

ARMCLAPI_ERR_COMM_ERROR

CLUSTERPROクライアントとの通信でエラーが発生しました。

ARMCLAPI_ERR_NOT_ENOUGH_MEMORY

メモリのアロケートに失敗しました。

ARMCLAPI_ERR_INVALID_PARAMETER

パラメータの指定が不正です。

ARMCLAPI_ERR_NOT_SUPPORTED

この関数はサポートされていません

説明

本関数は、CLUSTERPROクライアントAPIの初期化処理を行い、必要な資源を確保します。CLUSTERPROクライアントAPIを使用する場合には、最初に本関数を実行して、CLUSTERPROクライアントAPIの初期化を行ってください。戻り値として、CLUSTERPROクライアントAPIハンドルを返却します。CLUSTERPROクライアントAPIハンドルは、フェイルオーバーグループの監視や終了処理に必要です。

armcl2st.libをリンクしている場合には、CLUSTERPROクライアントがインストールされていない (CLUSTERPROクライアントAPI DLLをロードできない) 環境で実行した場合、本関数はARMCLAPI_ERROR_HANDLEを返却し、拡張エラー情報でARMCLAPI_ERR_CANNOT_LOAD_DLLが返却されます。

単一のプロセスから、独立に複数のArmClientInitializeApiExを実行する事が可能です。

参照

ArmClientTerminateApiEx

6.1.1.2 ArmClientTerminateApiEx — CLUSTERPRO クライアント API の終了

ARM1.1 より変更なし

概要

CLUSTERPROクライアントAPIの終了処理を行います。

形式

BOOL ArmClientTerminateApiEx (DWORD dwHandle)

パラメータ

[i]	dwHandle	ArmClientInitializeApiExで返却したCLUSTERPROクライアントAPIハンドルを指定します。
-----	----------	---

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPROクライアントAPIが初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPROクライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPROクライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_INVALID_HANDLE	指定されたハンドルが不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

CLUSTERPROクライアントAPIの終了処理を行います。ArmClientInitializeApiEx関数を実行してCLUSTERPROクライアントAPIの初期化を行ったアプリケーションは、CLUSTERPROクライアントAPIの機能を使用する必要が無くなった時に、CLUSTERPROクライアントAPIハンドルを指定して本関数を実行しなければいけません。

参照

ArmClientInitializeApiEx

6.1.1.3 ArmClientGetVersionInfo — CLUSTERPROクライアントAPIのバージョン情報の取得

ARM1.1 より変更なし

概要

CLUSTERPROクライアントAPIのバージョン情報を取得します。

形式

```
BOOL ArmClientGetVersionInfo(  
    LPDWORD dwMajor,  
    LPDWORD dwMinor,  
    LPDWORD dwBuild  
)
```

パラメータ

[o] dwMajor	メジャーバージョン番号。CLUSTERPRO Ver6.0では4を返却します。
[o] dwMinor	マイナーバージョン番号。CLUSTERPRO Ver6.0では0を返却します。
[o] dwBuild	ビルド番号。未使用。将来の拡張のために予約しています。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPROクライアントAPIが初期化されていません。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

CLUSTERPROクライアントAPIのバージョン情報を取得します。アプリケーションは、本関数を実行してCLUSTERPROクライアントAPIのバージョンを取得することができます。

互換性留意事項

CLUSTERPRO Ver3.0では、バージョン番号は3.0を返却していました。CLUSTERPRO Ver4.x以降では、Ver3.0互換運用、片方向スタンバイクラスタ、双方向スタンバイクラスタ、パラレルクラスタのいずれの構成でも、バージョン番号は4.0を返却します。

6.1.2 フェイルオーバーグループ状態問い合わせ

フェイルオーバーグループが動作しているサーバのネットワークアドレスや、フェイルオーバーグループが正常に動作しているかなどのフェイルオーバーグループ状態を問い合わせるAPIを提供します。問い合わせのキーは、クラスタ名とフェイルオーバーグループ名の組み合わせです。

6.1.2.1 ArmClientGetIPAddr — 実 IP アドレスの取得

プライマリサーバ名→フェイルオーバーグループ名

概要

接続するサーバの実IPアドレスを取得します。

形式

```
BOOL ArmClientGetIPAddr(  
    LPCTSTR lpszClusterName,  
    LPCTSTR lpszGroupName,  
    LPDWORD lpdwIPAddr  
)
```

パラメータ

[i] lpszClusterName	クラスタ名を指定する、NULLで終了する文字列へのポインタです。 ローカルサーバのクラスタを指定する場合は、定数 ARMCLAPI_LOCAL_CLUSTER を指定することができます。この場合、ローカルサーバがクラスタサーバでなければエラー (ARMCLAPI_ERR_CLUSTER_NOT_FOUND) が返却されます。
[i] lpszGroupName	フェイルオーバーグループ名を指定する、NULLで終了する文字列へのポインタです。
[o] lpdwIPAddr	IPアドレスを受け取る変数へのポインタです。 IPアドレスはネットワークバイトオーダーで格納されます。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPROクライアントAPIが初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPROクライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPROクライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	指定クラスタ名が存在しません。
ARMCLAPI_ERR_GROUP_NOT_FOUND	指定フェイルオーバーグループ名が存在しません。
ARMCLAPI_ERR_SERVER_NOT_FOUND	ARMCLAPI_ERR_GROUP_NOT_FOUNDの別名です。ARMVer3.0以前との互換性のために定義されています。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、クラスタ名、フェイルオーバーグループ名から、現在動作しているサーバの実IPアドレスを問い合わせます。TCP/IPでクラスタに接続する場合には、仮想IPアドレスを指定して接続する方法と、本APIを用いて実IPアドレスを求めて接続する方法の、2通りの方法を使用することができます。サーバと同一LAN上のWindows 3.1クライアントでは、仮想IPアドレスでの接続はできませんので、本APIを使用して、実IPアドレスで接続する必要があります。

互換性留意事項

CLUSTERPRO Ver3.0以前では、クラスタ名とプライマリサーバ名を入力パラメータとしていました。CLUSTERPRO Ver3.0互換運用では、プライマリサーバ名とフェイルオーバーグループ名は同一のため、アプリケーションの変更は不要です。
片方向スタンバイクラスタ、双方向スタンバイクラスタ、パラレルクラスタでは、プライマリサーバ名とフェイルオーバーグループ名は一般的に異なりますので注意が必要です。

6.1.2.2 ArmClientGetComputerName — サーバのLANManager コンピュータ名の取得

プライマリサーバ名→フェイルオーバーグループ名

概要

接続するサーバのLANManager コンピュータ名を取得します。

形式

```
BOOL ArmClientGetComputerName(  
    LPCTSTR lpszClusterName,  
    LPCTSTR lpszGroupName,  
    LPCTSTR lpszComputerName,  
    LPDWORD lpcchComputerName  
)
```

パラメータ

[i]	lpszClusterName	クラスタ名を指定する、NULLで終了する文字列へのポインタです。 ローカルサーバのクラスタを指定する場合は、定数 ARMCLAPI_LOCAL_CLUSTER を指定することができます。この場合、ローカルサーバがクラスタサーバでなければエラー (ARMCLAPI_ERR_CLUSTER_NOT_FOUND) が返却されます。
[i]	lpszGroupName	フェイルオーバーグループ名を指定する、NULLで終了する文字列へのポインタです。
[o]	lpszComputerName	現在動作しているサーバのコンピュータ名（終端のNULL文字を含む）を受け取るバッファへのポインタです。
[io]	lpcchComputerName	lpszComputerName で指定されるバッファのサイズを指定する変数へのポインタです。関数のリターン時には、バッファに格納された文字数（終端のNULL文字を含まない）がセットされます。指定バッファサイズが小さすぎた場合には、必要な文字数（終端のNULL文字を含む）がセットされ、拡張エラー情報で ARMCLAPI_ERR_BUFFER_TOO_SMALL が返却されます。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPROクライアントAPIが初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPROクライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPROクライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	指定クラスタ名が存在しません。
ARMCLAPI_ERR_GROUP_NOT_FOUND	指定フェイルオーバーグループ名が存在しません。

ARMCLAPI_ERR_SERVER_NOT_FOUND	ARMCLAPI_ERR_GROUP_NOT_FOUND の別名です。CLUSTERPRO Ver3.0以前との互換性のために定義されています。
ARMCLAPI_ERR_BUFFER_TOO_SMALL	データ返却用バッファが小さすぎます。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、クラスタ名、フェイルオーバーグループ名から、現在動作しているサーバのLANManager コンピュータ名を問い合わせます。

互換性留意事項

CLUSTERPRO Ver3.0以前では、クラスタ名とプライマリサーバ名を入力パラメータとしていました。CLUSTERPRO Ver3.0互換運用では、プライマリサーバ名とフェイルオーバーグループ名は同一のため、アプリケーションの変更は不要です。
片方向スタンバイクラスタ、双方向スタンバイクラスタ、パラレルクラスタでは、プライマリサーバ名とフェイルオーバーグループ名は一般的に異なりますので注意が必要です。

6.1.2.3 ArmClientGetGroupStatus — フェイルオーバーグループ動作状態の取得

ARM4.0 新規

概要

フェイルオーバーグループが、動作中/停止中のいずれの状態であるか取得します。

形式

```
BOOL ArmClientGetGroupStatus(  
    LPCTSTR lpszClusterName,  
    LPCTSTR lpszGroupName,  
    LPDWORD lpdwStatus  
)
```

パラメータ

[i]	lpszClusterName	クラスタ名を指定する、NULLで終了する文字列へのポインタです。 ローカルサーバのクラスタを指定する場合は、定数 ARMCLAPI_LOCAL_CLUSTERを指定することができます。この場合、 ローカルサーバがクラスタサーバでなければエラー (ARMCLAPI_ERR_CLUSTER_NOT_FOUND)が返却されます。
[i]	lpszGroupName	フェイルオーバーグループ名を指定する、NULLで終了する文字列へのポ インタです。
[o]	lpdwStatus	動作状態を受け取る変数へのポインタです。 以下の値が返却されます。 ARMCLAPI_GROUP_STATUS_STOPPED 停止中 ARMCLAPI_GROUP_STATUS_RUNNING 動作中

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPROクライアントAPIが初期化 されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPROクライアントに接続できま せん。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPROクライアントとの通信でエ ラーが発生しました。
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	指定クラスタ名が存在しません。
ARMCLAPI_ERR_GROUP_NOT_FOUND	指定フェイルオーバーグループ名が存在しま せん。
ARMCLAPI_ERR_SERVER_NOT_FOUND	ARMCLAPI_ERR_GROUP_NOT_FOUND の別名です。CLUSTERPRO Ver3.0以前との 互換性のために定義されています。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、クラスタ名、フェイルオーバーグループ名から、サーバが現在動作中か停止中かを問い合わせます。動作中とは、フェイルオーバーグループが、クライアントから接続可能なサーバのいずれかで動作している状態です。どのサーバで動作しているかは、ArmClientGetComputerName等で問い合わせてください。

停止中とは、フェイルオーバーグループが、クライアントから接続可能なサーバのいずれでも動作していない状態です。以下の場合は停止中となります。

- ・フェイルオーバーグループが移動可能なサーバが全て正常動作していない場合（サーバが停止あるいはダウン状態、CLUSTERPROが動作していない状態、クラスタから切り離された状態）
- ・フェイルオーバーグループが停止された状態
- ・フェイルオーバーグループが動作しているサーバとクライアントとの間のネットワーク経路の異常
- ・フェイルオーバー処理実行中

6.1.2.4 ArmClientGetServerStatus — フェイルオーバーグループ動作状態の取得

プライマリサーバ名→フェイルオーバーグループ名

ARM3.0 との互換性のためにのみ提供

概要

フェイルオーバーグループが、動作中/停止中のいずれの状態であるか取得します。
CLUSTERPRO Ver3.0 との互換性のためにのみ提供する関数です。ArmClientGetGroupStatus と関数名以外は全く同一です。

形式

```
BOOL ArmClientGetServerStatus(  
    LPCTSTR lpstrClusterName,  
    LPCTSTR lpstrGroupName,  
    LPDWORD lpdwStatus  
)
```

互換性留意事項

CLUSTERPRO Ver3.0 以前では、クラスタ名とプライマリサーバ名を入力パラメータとしていました。
CLUSTERPRO Ver3.0 互換運用では、プライマリサーバ名とフェイルオーバーグループ名は同一のため、アプリケーションの変更は不要です。
片方向スタンバイクラスタ、双方向スタンバイクラスタ、パラレルクラスタでは、プライマリサーバ名とフェイルオーバーグループ名は一般的に異なりますので注意が必要です。

参照

ArmClientGetGroupStatus

6.1.2.5 ArmClientGetGroupStatusEx — フェイルオーバーグループの詳細な動作状態の取得

ARM4.0 新規

概要

フェイルオーバーグループの動作状態を取得します。ArmClientGetGroupStatusより詳細な情報が得られます。

形式

```
BOOL ArmClientGetGroupStatusEx(  
    LPCTSTR lpszClusterName,  
    LPCTSTR lpszGroupName,  
    LPDWORD lpdwStatusEx  
)
```

パラメータ

[i] lpszClusterName	クラスタ名を指定する、NULLで終了する文字列へのポインタです。 ローカルサーバのクラスタを指定する場合は、定数 ARMCLAPI_LOCAL_CLUSTER を指定することができます。この場合、ローカルサーバがクラスタサーバでなければエラー (ARMCLAPI_ERR_CLUSTER_NOT_FOUND) が返却されます。								
[i] lpszGroupName	フェイルオーバーグループ名を指定する、NULLで終了する文字列へのポインタです。								
[o] lpdwStatusEx	動作状態を受け取る変数へのポインタです。 以下の値が返却されます。 <table><tr><td>ARMCLAPI_GROUP_STATUS_RUNNING</td><td>正常動作中</td></tr><tr><td>ARMCLAPI_GROUP_STATUS_MISS</td><td>資源獲得に失敗</td></tr><tr><td>ARMCLAPI_GROUP_STATUS_STOPPED</td><td>停止中</td></tr><tr><td>ARMCLAPI_GROUP_STATUS_FAILOVER</td><td>フェイルオーバー処理中</td></tr></table>	ARMCLAPI_GROUP_STATUS_RUNNING	正常動作中	ARMCLAPI_GROUP_STATUS_MISS	資源獲得に失敗	ARMCLAPI_GROUP_STATUS_STOPPED	停止中	ARMCLAPI_GROUP_STATUS_FAILOVER	フェイルオーバー処理中
ARMCLAPI_GROUP_STATUS_RUNNING	正常動作中								
ARMCLAPI_GROUP_STATUS_MISS	資源獲得に失敗								
ARMCLAPI_GROUP_STATUS_STOPPED	停止中								
ARMCLAPI_GROUP_STATUS_FAILOVER	フェイルオーバー処理中								

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアントAPIが初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	指定クラスタ名が存在しません。
ARMCLAPI_ERR_GROUP_NOT_FOUND	指定フェイルオーバーグループ名が存在しません。
ARMCLAPI_ERR_SERVER_NOT_FOUND	ARMCLAPI_ERR_GROUP_NOT_FOUND の別名です。CLUSTERPRO Ver3.0以前との互換性のために定義されています。

ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、クラスタ名、本来動作しているサーバのコンピュータ名から、サーバの状態を問い合わせます。正常動作中とは、フェイルオーバーグループが、クライアントから接続可能なサーバのいずれかで正常に動作している状態です。どのサーバで動作しているかは、ArmClientGetComputerName等で問い合わせてください。資源獲得に失敗とは、動作中であるが、切替ディスクの接続失敗など、クラスタ資源の獲得に失敗している状態です。

停止中とは、フェイルオーバーグループが、クライアントから接続可能なサーバのいずれでも動作していない状態です。フェイルオーバー処理実行中を除きます。以下の場合には停止中となります。

- ・フェイルオーバーグループが移動可能なサーバが全て正常動作していない場合（サーバが停止あるいはダウン状態、CLUSTERPROが動作していない状態、クラスタから切り離された状態）
- ・フェイルオーバーグループが停止された状態
- ・フェイルオーバーグループが動作しているサーバとクライアントとの間のネットワーク経路の異常

フェイルオーバー処理中とは、フェイルオーバーグループが動作していたサーバのダウンやフェイルオーバーグループの異常を検出してから、もう一方のサーバでの引き継ぎ処理が完了するまでの間です。また、CLUSTERPROマネージャからフェイルオーバーグループの移動を実行した場合は、移動を開始してから移動が完了するまでの間はフェイルオーバー処理中になります。

6.1.2.6 ArmClientGetServerStatusEx – フェイルオーバーグループの詳細な動作状態の取得

プライマリサーバ名→フェイルオーバーグループ名

ARM3.0 との互換性のためにのみ提供

概要

フェイルオーバーグループの状態を取得します。

CLUSTERPRO Ver3.0 との互換性のためにのみ提供する関数です。ArmClientGetGroupStatusEx と関数名以外は全く同一です。

形式

```
BOOL ArmClientGetServerStatusEx(  
    LPCTSTR lpzClusterName,  
    LPCTSTR lpzGroupName,  
    LPDWORD lpdwStatusEx  
)
```

互換性留意事項

CLUSTERPRO Ver3.0 以前では、クラスタ名とプライマリサーバ名を入力パラメータとしていました。

CLUSTERPRO Ver3.0 互換運用では、プライマリサーバ名とフェイルオーバーグループ名は同一のため、アプリケーションの変更は不要です。

片方向スタンバイクラスタ、双方向スタンバイクラスタ、パラレルクラスタでは、プライマリサーバ名とフェイルオーバーグループ名は一般的に異なりますので注意が必要です。

参照

ArmClientGetGroupStatusEx

6.1.3 フェイルオーバーグループ監視

フェイルオーバーグループの状態を監視します。フェイルオーバーグループの起動及び停止、フェイルオーバーの開始及び完了、フェイルオーバーグループ移動の開始及び完了、サーバとクライアント間のネットワーク経路の異常発生及び回復などで、グループの状態が変更された場合に、CLUSTERPROクライアントから通知を受け取るAPIを提供します。CLUSTERPROクライアントは、以下の契機でフェイルオーバーに関する情報の通知を行います。

クライアントAPIイベント	クラスタイベント
ARMCLAPI_EVENT_FAILOVER_START	・サーバダウンによるフェイルオーバーの開始 ・アプリケーション・サービス異常によるフェイルオーバーの開始 ・フェイルオーバーグループ移動の開始 ・ネットワークの切り替え開始
ARMCLAPI_EVENT_FAILOVER_END	・サーバダウンによるフェイルオーバーの完了 ・アプリケーション・サービス異常によるフェイルオーバーの開始 ・フェイルオーバーグループ移動の完了 ・ネットワークの切り替え完了
ARMCLAPI_EVENT_SHUTDOWN	・クラスタシャットダウンによるフェイルオーバーグループの停止 ・ユーザの指示によるフェイルオーバーグループの停止 ・ネットワークの異常検出
ARMCLAPI_EVENT_BOOT	・サーバ起動によるフェイルオーバーグループの起動 ・ユーザの指示によるフェイルオーバーグループの起動 ・ネットワークの回復検出

また、各イベントと共に通知されるのは、以下の情報です。

- * ARMCLAPI_EVENT_FAILOVER_START, ARMCLAPI_EVENT_SHUTDOWN
 - + フェイルオーバー・停止が発生したクラスタ名
 - + フェイルオーバー・停止するフェイルオーバーグループ名
 - + フェイルオーバー・停止するグループに割り付けられた仮想IPアドレス
 - + フェイルオーバー・停止時にグループが動作していたサーバのコンピュータ名
 - + フェイルオーバー・停止時にグループが動作していたサーバの実IPアドレス
- * **フェイルオーバー完了時、クラスタ起動時**
 - + フェイルオーバー・起動が発生したクラスタ名
 - + フェイルオーバー・起動したフェイルオーバーグループ名
 - + フェイルオーバー・起動したフェイルオーバーグループに割り付けられた仮想IPアドレス
 - + フェイルオーバー・起動でグループが動作しているサーバのコンピュータ名
 - + フェイルオーバー・起動でグループが動作しているサーバの実IPアドレス

6.1.3.1 ArmClientStartWatchFailoverEx — フェイルオーバーグループの監視

プライマリサーバ名→フェイルオーバーグループ名	フェイルオーバー要因・フェイルオーバー単位の追加
-------------------------	--------------------------

概要

フェイルオーバーの発生/完了などの、フェイルオーバーグループの状態変化の監視を開始します。グループの状態変化を検出した時に、登録されたコールバック関数を呼び出します。

形式

```
BOOL ArmClientStartWatchFailoverEx(  
    DWORD dwHandle  
    LPCTSTR lpszClusterName,  
    LPCTSTR lpszGroupName,  
    WATCHFAILOVERPROC WatchFailoverProc,  
    LPDWORD lpdwWatchId  
    LPDWORD lpdwStatusEx  
)
```

パラメータ

[i]	dwHandle	CLUSTERPROクライアントAPIハンドル
[i]	lpszClusterName	監視するフェイルオーバーグループが属するクラスタ名を指定する、NULLで終了する文字列へのポインタです。 ローカルサーバのクラスタを指定する場合は、定数 ARMCLAPI_LOCAL_CLUSTER を指定することができます。この場合、ローカルサーバがクラスタサーバでなければエラー (ARMCLAPI_ERR_CLUSTER_NOT_FOUND) が返却されます。
[i]	lpszGroupName	監視するフェイルオーバーグループ名を指定する、NULLで終了する文字列へのポインタです。
[i]	WatchFailoverProc	failoverの発生/完了を検出した時に呼び出される、アプリケーション定義のコールバック関数のアドレスです。 (WatchFailoverProc関数参照)
[o]	lpdwWatchId	監視IDを受け取る変数へのポインタです。
[o]	lpdwStatusEx	監視を開始した時のフェイルオーバーグループの状態です。 ArmClientGetGroupStatusExで返却するものと同じです。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPROクライアントAPIが初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPROクライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPROクライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	指定クラスタ名が存在しません。
ARMCLAPI_ERR_GROUP_NOT_FOUND	指定フェイルオーバーグループ名が存在しません。

ARMCLAPI_ERR_SERVER_NOT_FOUND	ARMCLAPI_ERR_GROUP_NOT_FOUND の別名です。CLUSTERPRO Ver3.0以前との互換性のために定義されています。
ARMCLAPI_ERR_INVALID_HANDLE	ハンドルが不正です。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、フェイルオーバーグループの状態変化の監視を開始します。監視する状態変化は、以下の通りです。

- * フェイルオーバーの開始
- * フェイルオーバーの完了
- * フェイルオーバーグループ移動の開始
- * フェイルオーバーグループ移動の完了
- * フェイルオーバーグループの起動
- * フェイルオーバーグループの停止
- * サーバクライアント間のネットワークの切り替え開始
- * サーバクライアント間のネットワークの切り替え完了
- * サーバクライアント間のネットワークの異常検出
- * サーバクライアント間のネットワークの回復検出

本関数は、アプリケーション定義のコールバック関数WatchFailoverProcを登録した後、直ちに本関数を呼び出したアプリケーションプロセスに制御を戻します。コールバック関数は、CLUSTERPRO クライアントによりフェイルオーバーグループの状態の変化が検出された時に、非同期に呼び出されます。

互換性留意事項

CLUSTERPRO Ver3.0以前では、クラスタ名とプライマリサーバ名を入力パラメータとしていました。CLUSTERPRO Ver3.0互換運用では、プライマリサーバ名とフェイルオーバーグループ名は同一のため、アプリケーションの変更は不要です。
片方向スタンバイクラスタ、双方向スタンバイクラスタ、パラレルクラスタでは、プライマリサーバ名とフェイルオーバーグループ名は一般的に異なりますので注意が必要です。

参照

ArmClientCancelWatchFailover , WatchFailoverProc, ArmClientGetServerStatusEx

6.1.3.2 ArmClientCancelWatchFailoverEx — フェイルオーバーグループ監視の中止

ARM1.1 より変更なし

概要

ArmClientStartWatchFailoverEx関数によるフェイルオーバーグループの発生/完了の監視を中止します。

形式

```
BOOL ArmClientCancelWatchFailoverEx(  
    DWORD dwWatchId  
)
```

パラメータ

[i] dwWatchId ArmClientStartWatchFailover関数で返却された監視IDを指定します。

戻り値

TRUE 関数が正常終了しました。
FALSE 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPROクライアントAPIが初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPROクライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPROクライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_INVALID_ID	指定した監視IDが不正です。

説明

本関数は、dwWatchIdで識別されるArmClientStartWatchFailover関数によるフェイルオーバーグループの監視を中止します。

参照

ArmClientStartWatchFailover

6.1.3.3 WatchFailoverProc — フェイルオーバーグループ監視用コールバック関数

ARM1.1 より変更なし

概要

ArmClientStartWatchFailoverEx関数により登録され、フェイルオーバーグループの状態変更時に呼び出される、アプリケーション定義のコールバック関数です。

形式

```
VOID CALLBACK WatchFailoverProc(  
    DWORD dwEventId,  
    DWORD dwWatchId,  
    LPARMCLAPI_FAILOVER_INFO lpArmFailoverInfoEx  
)
```

パラメータ

[i]	dwEventId	検出されたイベントの通し番号です。
[i]	dwWatchId	対応するArmClientStartWatchFailover関数を識別する監視IDです。
[i]	lpArmFailoverInfoEx	failover情報がセットされたARMCLAPI_FAILOVER_INFO構造体へのポインタです。

戻り値

なし

説明

本関数は、ArmClientStartWatchFailoverEx関数により登録される、アプリケーション定義のコールバック関数であり、CLUSTERPROクライアントがフェイルオーバーグループの状態の変更を検出した時に呼び出されます。

本関数の関数名“WatchFailoverProc”は、アプリケーション定義の関数名のプレースホルダです。

互換性留意事項

CLUSTERPRO Ver3.0以前では、ARMCLAPI_FAILOVER_INFOでイベントに対応したプライマリサーバ名が渡されました。CLUSTERPRO Ver4.x以降では、このメンバにはフェイルオーバーグループ名が入ります。

CLUSTERPRO Ver3.0互換運用では、プライマリサーバ名とフェイルオーバーグループ名は同一のため、アプリケーションの変更は不要です。

片方向スタンバイクラスタ、双方向スタンバイクラスタ、パラレルクラスタでは、プライマリサーバ名とフェイルオーバーグループ名は一般的に異なりますので注意が必要です。

参照

ArmClientStartWatchFailoverEx, ArmClientCancelWatchFailoverEx, ARMCLAPI_FAILOVER_INFO構造体

6.1.4 サーバ状態問い合わせ

サーバの状態を問い合わせるAPIを提供します。問い合わせのキーは、クラスタ名とサーバ名の組み合わせです。

6.1.4.1 ArmClientGetClusterServerStatus — サーバの動作状態の取得

ARM4.0 新規

概要

サーバの動作状態を取得します。

形式

```
BOOL ArmClientGetClusterServerStatus(  
    LPCTSTR lpzClusterName,  
    LPCTSTR lpzServerName,  
    LPDWORD lpdwServerStatus  
)
```

パラメータ

[I]	<i>lpzClusterName</i>	クラスタ名を指定する、NULLで終了する文字列へのポインタです。 ローカルサーバのクラスタを指定する場合は、定数 ARMCLAPI_LOCAL_CLUSTER を指定することができます。この場合、ローカルサーバがクラスタサーバでなければエラー (ARMCLAPI_ERR_CLUSTER_NOT_FOUND) が返却されます。
[I]	<i>lpzServerName</i>	サーバ名を指定する、NULLで終了する文字列へのポインタです。
[O]	<i>lpdwServerStatus</i>	動作状態を受け取る変数へのポインタです。 以下の値が返却されます。 ARMCLAPI_SERVER_STATUS_SINGLE_SERVE R クラスタに未登録のシングルサーバとして動作しています。 ARMCLAPI_SERVER_STATUS_CLUSTER クラスタとして正常に動作しています。 ARMCLAPI_SERVER_STATUS_OFF_CLUSTER クラスタには登録されているが、クラスタから切り離されて動作しています。 ARMCLAPI_SERVER_STATUS_STOPPED 停止状態です。 ARMCLAPI_SERVER_STATUS_DOWN ダウン状態です。 ARMCLAPI_SERVER_STATUS_RECOVERING フェイルオーバー処理中です。 上記の状態に、次の状態がbit論理和で組み合わせられます。 ARMCLAPI_SERVER_STATUS_BACKUP 他サーバからフェイルオーバーした資源があります。 ARMCLAPI_SERVER_STATUS_FAIL 他サーバへフェイルオーバーした資源があります。 ARMCLAPI_SERVER_STATUS_MISS 接続に失敗した資源があります。 例えばクラスタとして正常に動作しているかどうかを調べるためには、 lpdwServerStatus & ARMCLAPI_SERVER_STATUS_CLUSTER の値が0かどうかで判定することができます。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPROクライアントAPIが初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPROクライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPROクライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	指定クラスタ名が存在しません。
ARMCLAPI_ERR_SERVER_NOT_FOUND	指定サーバが存在しません。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、サーバの状態を問い合わせます。

6.1.5 クラスタ情報の取得

クラスタの情報を取得します。

6.1.5.1 ArmClientEnumCluster — 登録されたクラスタの列挙

ARM4.0 新規

概要

クライアントに登録されたクラスタを列挙します。

形式

```
BOOL ArmClientEnumCluster (
    CLIENTENUMCLUSTERPROC EnumClusterProc,
    LPARAM lParam
)
```

パラメータ

- | | |
|---------------------|---|
| [i] EnumClusterProc | クラスタ名を受け取る、アプリケーション定義のコールバック関数のアドレスです。
(EnumClusterProc関数参照) |
| [i] lParam | コールバック関数に渡される、アプリケーション定義の32ビット値です。 |

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPROクライアントAPIが初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPROクライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPROクライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。
ARMCLAPI_ERR_CANCELLED	コールバック関数から列挙の中止が返却されました。

説明

本関数は、クライアントに登録されたクラスタを列挙します。本関数は、アプリケーション定義のコールバック関数EnumClusterProcを順次呼び出し、各クラスタ名を渡します。本関数は、クラスタをすべて列挙するか、またはコールバック関数がFALSEを返却するまで、列挙を続けます。

参照

EnumClusterProc

6.1.5.2 EnumClusterProc — クラスタ列挙用コールバック関数

ARM4.0 新規

概要

ArmClientEnumCluster関数で列挙されるクラスタ名を受け取るための、アプリケーション定義のコールバック関数です。

形式

```
BOOL CALLBACK EnumClusterProc (  
    LPCTSTR lpszClusterName,  
    LPARAM lParam  
)
```

パラメータ

[i]	lpszClusterName	クラスタ名を示す、NULLで終わる文字列へのポインタです。
[i]	lParam	ArmClientEnumServer関数で指定されたアプリケーション定義の値です。

戻り値

TRUE	列挙を続ける場合に返却します。
FALSE	列挙を止める場合に返却します。

説明

EnumClusterProc関数は、ArmClientEnumCluster関数の実行結果として、クラスタ名を受け取るための、アプリケーション定義のコールバック関数です。列挙が完了するか、または本関数でFALSEを返却するまで、クラスタ名が本関数に順次受け渡されます。

アプリケーションは、ArmClientEnumCluster関数実行時に本関数のアドレスを指定しなければなりません。本関数に渡されるlpszClusterNameの指す文字列の内容は、本関数をリターンするまでの間でのみ有効です。これらの文字列の内容を保持する必要がある場合には、本関数内で文字列の内容を別の記憶領域へコピーしてください。

本関数の関数名“EnumClusterProc”は、アプリケーション定義の関数名のプレースホルダです。

参照

ArmClientEnumCluster

6.1.5.3 ArmClientEnumServer — クラスタに所属するサーバの列挙

クラスタ名に NULL 指定可能

概要

クラスタに所属するサーバを列挙します。

形式

```
BOOL ArmClientEnumServer (  
    LPCTSTR lpszClusterName,  
    ENUMSERVERPROC EnumServerProc,  
    LPARAM lParam  
)
```

パラメータ

- | | |
|---------------------|--|
| [i] lpszClusterName | クラスタ名を指定する、NULLで終了する文字列へのポインタです。
NULLあるいは空文字列を指定した場合は、登録された全てのクラスタが対象となります。
ローカルサーバのクラスタを指定する場合は、定数 ARMCLAPI_LOCAL_CLUSTER を指定することができます。この場合、ローカルサーバがクラスタサーバでなければエラー (ARMCLAPI_ERR_CLUSTER_NOT_FOUND) が返却されます。 |
| [i] EnumServerProc | サーバのコンピュータ名を受け取る、アプリケーション定義のコールバック関数のアドレスです。(EnumServerProc関数参照) |
| [i] lParam | コールバック関数に渡される、アプリケーション定義の32ビット値です。 |

戻り値

- | | |
|-------|--|
| TRUE | 関数が正常終了しました。 |
| FALSE | 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。 |

拡張エラー情報

- | | |
|---------------------------------------|----------------------------------|
| ARMCLAPI_ERR_NOT_INITIALIZED | CLUSTERPROクライアントAPIが初期化されていません。 |
| ARMCLAPI_ERR_CANNOT_CONNECT | CLUSTERPROクライアントに接続できません。 |
| ARMCLAPI_ERR_COMM_ERROR | CLUSTERPROクライアントとの通信でエラーが発生しました。 |
| ARMCLAPI_ERR_INVALID_PARAMETER | パラメータの指定が不正です。 |
| ARMCLAPI_ERR_NOT_SUPPORTED | この関数はサポートされていません。 |
| ARMCLAPI_ERR_CLUSTER_NOT_FOUND | 指定クラスタ名が存在しません。 |
| ARMCLAPI_ERR_CANCELLED | コールバック関数から列挙の中止が返却されました。 |

説明

本関数は、指定したクラスタを構成するクサーバを列挙します。本関数は、アプリケーション定義のコールバック関数EnumServerProcを順次呼び出し、各サーバのコンピュータ名を渡します。本関数は、サーバをすべて列挙するか、またはコールバック関数がFALSEを返却するまで、列挙を続けます。

参照

EnumServerProc

6.1.5.4 EnumServerProc — サーバ列挙用コールバック関数

ARM1.1 より変更なし

概要

ArmClientEnumServer関数で列挙されるサーバのコンピュータ名を受け取るための、アプリケーション定義のコールバック関数です。

形式

```
BOOL CALLBACK EnumServerProc (  
    LPCTSTR lpszClusterName,  
    LPCTSTR lpszServerName,  
    LPARAM lParam  
)
```

パラメータ

- | | | |
|-----|-----------------|--|
| [i] | lpszClusterName | クラスタ名を示す、NULLで終わる文字列へのポインタです。 |
| [i] | lpszServerName | サーバのコンピュータ名を示す、NULLで終わる文字列へのポインタです。 |
| [i] | lParam | ArmClientEnumServer関数で指定されたアプリケーション定義の値です。 |

戻り値

- | | |
|-------|-----------------|
| TRUE | 列挙を続ける場合に返却します。 |
| FALSE | 列挙を止める場合に返却します。 |

説明

EnumServerProc関数は、ArmClientEnumServer関数の実行結果として、クラスタを構成するサーバのコンピュータ名を受け取るための、アプリケーション定義のコールバック関数です。サーバの列挙が完了するか、または本関数でFALSEを返却するまで、サーバのコンピュータ名が本関数に順次受け渡されます。アプリケーションは、ArmClientEnumServer関数実行時に本関数のアドレスを指定しなければなりません。本関数に渡されるlpszClusterName、lpszServerNameの指す文字列の内容は、本関数をリターンするまでの間でのみ有効です。これらの文字列の内容を保持する必要がある場合には、本関数内で文字列の内容を別の記憶領域へコピーしてください。本関数の関数名“EnumServerProc”は、アプリケーション定義の関数名のプレースホルダです。

参照

ArmClientEnumServer

6.1.5.5 ArmClientEnumGroup — クラスタに所属するフェイルオーバーグループの列挙

ARM4.0 新規

概要

クラスタに所属するフェイルオーバーグループを列挙します。

形式

```
BOOL ArmClientEnumGroup (  
    LPCTSTR lpszClusterName,  
    ENUMGROUPPROC EnumGroupProc,  
    LPARAM lParam  
)
```

パラメータ

- | | |
|----------------------------|--|
| [i] <i>lpszClusterName</i> | クラスタ名を指定する、NULLで終了する文字列へのポインタです。
NULLあるいは空文字列を指定した場合は、登録された全てのクラスタが対象となります。
ローカルサーバのクラスタを指定する場合は、定数 ARMCLAPI_LOCAL_CLUSTER を指定することができます。この場合、ローカルサーバがクラスタサーバでなければエラー (ARMCLAPI_ERR_CLUSTER_NOT_FOUND) が返却されます。 |
| [i] <i>EnumGroupProc</i> | フェイルオーバーグループ名を受け取る、アプリケーション定義のコールバック関数のアドレスです。
(EnumGroupProc関数参照) |
| [i] <i>lParam</i> | コールバック関数に渡される、アプリケーション定義の32ビット値です。 |

戻り値

- | | |
|-------|--|
| TRUE | 関数が正常終了しました。 |
| FALSE | 関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。 |

拡張エラー情報

- | | |
|--------------------------------|----------------------------------|
| ARMCLAPI_ERR_NOT_INITIALIZED | CLUSTERPROクライアントAPIが初期化されていません。 |
| ARMCLAPI_ERR_CANNOT_CONNECT | CLUSTERPROクライアントに接続できません。 |
| ARMCLAPI_ERR_COMM_ERROR | CLUSTERPROクライアントとの通信でエラーが発生しました。 |
| ARMCLAPI_ERR_INVALID_PARAMETER | パラメータの指定が不正です。 |
| ARMCLAPI_ERR_NOT_SUPPORTED | この関数はサポートされていません。 |
| ARMCLAPI_ERR_CLUSTER_NOT_FOUND | 指定クラスタ名が存在しません。 |
| ARMCLAPI_ERR_CANCELLED | コールバック関数から列挙の中止が返却されました。 |

説明

本関数は、指定したクラスタに属するフェイルオーバーグループを列挙します。本関数は、アプリケーション定義のコールバック関数EnumGroupProcを順次呼び出し、各フェイルオーバーグループ名を渡します。本関数は、フェイルオーバーグループをすべて列挙するか、またはコールバック関数がFALSEを返却するまで、列挙を続けます。

参照

EnumGroupProc

6.1.5.6 EnumGroupProc — フェイルオーバーグループ列挙用コールバック関数

ARM4.0 新規

概要

ArmClientEnumGroup関数で列挙されるフェイルオーバーグループ名を受け取るための、アプリケーション定義のコールバック関数です。

形式

```
BOOL CALLBACK EnumGroupProc (  
    LPCTSTR lpszClusterName,  
    LPCTSTR lpszGroupName,  
    LPARAM lParam  
)
```

パラメータ

- | | | |
|-----|-----------------|--|
| [i] | lpszClusterName | クラスタ名を示す、NULLで終わる文字列へのポインタです。 |
| [i] | lpszGroupName | フェイルオーバーグループ名を示す文字列へのポインタです。 |
| [i] | lParam | ArmClientEnumServer関数で指定されたアプリケーション定義の値です。 |

戻り値

- | | |
|-------|-----------------|
| TRUE | 列挙を続ける場合に返却します。 |
| FALSE | 列挙を止める場合に返却します。 |

説明

EnumGroupProc関数は、ArmClientEnumGroup関数の実行結果として、クラスタを構成するサーバのコンピュータ名を受け取るための、アプリケーション定義のコールバック関数です。サーバの列挙が完了するか、または本関数でFALSEを返却するまで、フェイルオーバーグループ名が本関数に順次受け渡されます。

アプリケーションは、ArmClientEnumGroup関数実行時に本関数のアドレスを指定しなければなりません。

本関数に渡されるlpszClusterName, lpszGroupNameの指す文字列の内容は、本関数をリターンするまでの間でのみ有効です。これらの文字列の内容を保持する必要がある場合には、本関数内で文字列の内容を別の記憶領域へコピーしてください。

本関数の関数名“EnumGroupProc”は、アプリケーション定義の関数名のプレースホルダです。

参照

ArmClientEnumGroup

6.1.5.7 ArmClientGetClusterFromClusterServer — サーバが所属するクラスタ名の取得

ARM1.1 より変更なし

概要

サーバ名から、そのサーバが所属するクラスタのクラスタ名を取得します。

形式

```
BOOL ArmClientGetClusterFromClusterServer(  
    LPCSTR    lpzServerName,  
    LPSTR     lpzClusterName  
    LPDWORD   lpcchClusterName  
)
```

パラメータ

[i]	lpzServerName	サーバ名を指定する、NULLで終了する文字列へのポインタです。
[o]	lpzClusterName	現在動作しているサーバのコンピュータ名（終端のNULL文字を含む）を受け取るバッファへのポインタです。
[io]	lpcchClusterName	lpzClusterNameで指定されるバッファのサイズを指定する変数へのポインタです。関数のリターン時には、バッファに格納された文字数（終端のNULL文字を含まない）がセットされます。指定バッファサイズが小さすぎた場合には、必要な文字数（終端のNULL文字を含む）がセットされ、拡張エラー情報でARMCLAPI_ERR_BUFFER_TOO_SMALLが返却されます。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPROクライアントAPIが初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPROクライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPROクライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_SERVER_NOT_FOUND	指定されたサーバ名が見つかりません。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、サーバ名から、そのサーバが属しているクラスタ名を問い合わせます。ただし、複数のクラスタに同一の名前のサーバ名が存在する場合は、どのクラスタ名を返却するかは保証しません。

6.1.5.8 ArmClientGetClusterFromGroup — フェイルオーバーグループが所属するクラスタ名の取得

ARM4.0 新規

概要

フェイルオーバーグループ名から、そのサーバが所属するクラスタのクラスタ名を取得します。

形式

```
BOOL ArmClientGetClusterFromGroup(  
    LPCSTR    lpszGroupName,  
    LPSTR     lpszClusterName  
    LPDWORD   lpccsClusterName  
)
```

パラメータ

[i]	lpszGroupName	フェイルオーバーグループ名を指定する、NULLで終了する文字列へのポインタです。
[o]	lpszClusterName	クラスタ名（終端のNULL文字を含む）を受け取るバッファへのポインタです。
[io]	lpccsClusterName	lpszClusterNameで指定されるバッファのサイズを指定する変数へのポインタです。関数のリターン時には、バッファに格納された文字数（終端のNULL文字を含まない）がセットされます。指定バッファサイズが小さすぎた場合には、必要な文字数（終端のNULL文字を含む）がセットされ、拡張エラー情報でARMCLAPI_ERR_BUFFER_TOO_SMALLが返却されます。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPROクライアントAPIが初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPROクライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPROクライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_GROUP_NOT_FOUND	指定されたフェイルオーバーグループ名が見つかりません。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、フェイルオーバーグループ名から、そのサーバが属しているクラスタ名を問い合わせます。ただし、複数のクラスタに同一の名前のフェイルオーバーグループ名が存在する場合は、どのクラスタ名を返却するかは保証しません。

6.1.5.9 ArmClientGetClusterFromServer — フェイルオーバーグループが所属するクラスタ名の取得

プライマリサーバ名→フェイルオーバーグループ名

ARM3.0 との互換性のためにのみ提供

概要

フェイルオーバーグループ名から、そのサーバが所属するクラスタのクラスタ名を取得します。

形式

```
BOOL ArmClientGetClusterFromServer(  
    LPCSTR    lpzszGroupName,  
    LPSTR     lpzszClusterName  
    LPDWORD   lpccszClusterName  
)
```

パラメータ

[i]	lpzszGroupName	フェイルオーバーグループ名を指定する、NULLで終了する文字列へのポインタです。
[o]	lpzszClusterName	クラスタ名（終端のNULL文字を含む）を受け取るバッファへのポインタです。
[io]	lpccszClusterName	lpzszClusterNameで指定されるバッファのサイズを指定する変数へのポインタです。関数のリターン時には、バッファに格納された文字数（終端のNULL文字を含まない）がセットされます。指定バッファサイズが小さすぎた場合には、必要な文字数（終端のNULL文字を含む）がセットされ、拡張エラー情報でARMCLAPI_ERR_BUFFER_TOO_SMALLが返却されます。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPROクライアントAPIが初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPROクライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPROクライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_GROUP_NOT_FOUND	指定されたフェイルオーバーグループ名が見つかりません。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、フェイルオーバーグループ名から、そのサーバが属しているクラスタ名を問い合わせます。ただし、複数のクラスタに同一の名前のフェイルオーバーグループ名が存在する場合は、どのクラスタ名を返却するかは保証しません。

6.1.5.10 ArmClientEnumServerFromGroup – フェイルオーバーグループが移動可能なサーバの列挙

ARM4.0 新規

概要

フェイルオーバーグループが移動可能なサーバを列挙します。

形式

```
BOOL ArmClientEnumServerFromGroup (
    LPCTSTR lpszClusterName,
    LPCTSTR lpszGroupName,
    ENUMSERVERFROMGROUPPROC EnumServerFromGroupProc,
    LPARAM lParam
)
```

パラメータ

[i] lpszClusterName	クラスタ名を指定する、NULLで終了する文字列へのポインタです。ローカルサーバのクラスタを指定する場合は、定数 ARMCLAPI_LOCAL_CLUSTER を指定することができます。この場合、ローカルサーバがクラスタサーバでなければエラー (ARMCLAPI_ERR_CLUSTER_NOT_FOUND) が返却されます。
[i] lpszGroupName	フェイルオーバーグループ名を指定する、NULLで終了する文字列へのポインタです。
[i] EnumServerFromGroupProc	サーバのコンピュータ名を受け取る、アプリケーション定義のコールバック関数のアドレスです。 (EnumServerFromGroupProc関数参照)
[i] lParam	コールバック関数に渡される、アプリケーション定義の32ビット値です。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPROクライアントAPIが初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPROクライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPROクライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	指定クラスタ名が存在しません。
ARMCLAPI_ERR_CANCELLED	コールバック関数から列挙の中止が返却されました。

説明

本関数は、指定したフェイルオーバーグループが移動可能なサーバを列挙します。本関数は、アプリケーション定義のコールバック関数EnumServerFromGroupProcを順次呼び出し、各サーバのコンピュータ名を渡します。本関数は、サーバをすべて列挙するか、またはコールバック関数がFALSEを返却するまで、列挙を続けます。列挙の順は、フェイルオーバーグループのフェイルオーバーポリシーの優先度が高いサーバから順に列挙されます。

参照

EnumServerFromGroupProc

6.1.5.11 EnumServerFromGroupProc — サーバ列挙用コールバック関数

ARM4.0 新規

概要

ArmClientEnumServerFromGroup関数で列挙されるサーバのコンピュータ名を受け取るための、アプリケーション定義のコールバック関数です。

形式

```
BOOL CALLBACK EnumServerFromGroupProc (  
    LPCTSTR lpszClusterName,  
    LPCTSTR lpszGroupName,  
    LPCTSTR lpszServerName,  
    LPARAM lParam  
)
```

パラメータ

[i]	lpszClusterName	クラスタ名を示す、NULLで終わる文字列へのポインタです。
[i]	lpszGroupName	フェイルオーバーグループ名を示す、NULLで終わる文字列へのポインタです。
[i]	lpszServerName	サーバのコンピュータ名を示す、NULLで終わる文字列へのポインタです。
[i]	lParam	ArmClientEnumServer関数で指定されたアプリケーション定義の値です。

戻り値

TRUE	列挙を続ける場合に返却します。
FALSE	列挙を止める場合に返却します。

説明

EnumServerFromGroupProc関数は、ArmClientEnumServerFromGroup関数の実行結果として、フェイルオーバーグループが移動可能なサーバのコンピュータ名を受け取るための、アプリケーション定義のコールバック関数です。サーバの列挙が完了するか、または本関数でFALSEを返却するまで、サーバのコンピュータ名が本関数に順次受け渡されます。

アプリケーションは、ArmClientEnumServerFromGroup関数実行時に本関数のアドレスを指定しなければなりません。

本関数に渡されるlpszClusterName, lpszGroupName, lpszServerNameの指す文字列の内容は、本関数をリターンするまでの間でのみ有効です。これらの文字列の内容を保持する必要がある場合には、本関数内で文字列の内容を別の記憶領域へコピーしてください。

本関数の関数名“EnumServerFromGroupProc”は、アプリケーション定義の関数名のプレースホルダです。

参照

ArmClientEnumServerFromGroup

6.1.6 仮想 IP アドレス

仮想IPアドレスから、クラスタ名及びフェイルオーバーグループ名を取得します。また、仮想IPアドレスの列挙を行います。

6.1.6.1 ArmClientGetGroupFromVirtualIP – 仮想IPアドレスが所属しているフェイルオーバーグループ名の取得

ARM4.0 新規

概要

仮想IPアドレスが所属しているクラスタ名及びフェイルオーバーグループ名を取得します。

形式

```
BOOL ArmClientGetGroupFromVirtualIP (
    DWORD   dwVirtualIP,
    LPTSTR   lpszClusterName,
    LPDWORD  lpcchClusterName
    LPTSTR   lpszGroupName,
    LPDWORD  lpcchGroupName
)
```

パラメータ

[i]	dwVirtualIP	仮想IPアドレス。ネットワークバイトオーダーで指定します。
[o]	lpszClusterName	クラスタ名（終端のNULL文字を含む）を受け取るバッファへのポインタ。
[io]	lpcchClusterName	lpszClusterNameで指定されるバッファのサイズを指定する変数へのポインタ。関数のリターン時にはバッファに格納された文字数（終端のNULL文字を含まない）がセットされます。指定バッファサイズが小さすぎた場合には、必要な文字数（終端のNULL文字を含む）がセットされ、拡張エラー情報でARMERROR_BUFFER_TOO_SMALLが返却されます。
[o]	lpszGroupName	仮想IPアドレスが所属しているフェイルオーバーグループ名（終端のNULL文字を含む）を受け取るバッファへのポインタ。
[io]	lpcchGroupName	lpszGroupNameで指定されるバッファのサイズを指定する変数へのポインタ。関数のリターン時にはバッファに格納された文字数（終端のNULL文字を含まない）がセットされます。指定バッファサイズが小さすぎた場合には、必要な文字数（終端のNULL文字を含む）がセットされ、拡張エラー情報でARMERROR_BUFFER_TOO_SMALLが返却されます。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報を返却します。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPROクライアントAPIの初期化処理が行われていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPROクライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPROクライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_BUFFER_TOO_SMALL	文字列を格納するバッファのサイズが小さすぎます。

ARMCLAPI_ERR_VIRTUALIP_NOT_FOUND	指定仮想IPアドレスが存在しません。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は仮想IPアドレスを指定して、それに対応するクラスタ名及びフェイルオーバーグループ名を取得します。クラスタ名をバッファlpzClusterNameに格納します。フェイルオーバーグループ名をlpzGroupNameに格納します。仮想IPアドレスを監視する場合には、まずこの関数でクラスタ名及びフェイルオーバーグループ名を得ます。そして、それを、フェイルオーバーグループの状態監視開始の入力パラメータとして使用することで、監視が開始されます。

6.1.6.2 ArmClientGetPrimaryServerFromVirtualIP — 仮想IPアドレスが所属しているフェイルオーバーグループ名の取得

プライマリサーバ名→フェイルオーバーグループ名

ARM3.0 との互換性のためにのみ提供

概要

仮想IPアドレスを指定して、所属しているクラスタ名及びフェイルオーバーグループ名を取得します。CLUSTERPRO Ver3.0との互換性のためにのみ提供する関数です。ArmClientGetGroupFromVirtualIPと関数名以外は全く同一です。

形式

```
BOOL ArmClientGetPrimaryServerFromVirtualIP (  
    DWORD    dwVirtualIP,  
    LPTSTR    lpszClusterName,  
    LPDWORD   lpcchClusterName  
    LPTSTR    lpszGroupName,  
    LPDWORD   lpcchGroupName  
)
```

互換性留意事項

CLUSTERPRO Ver3.0以前では、本関数ではクラスタ名とプライマリサーバ名が取得されました。CLUSTERPRO Ver4.x以降では、プライマリサーバ名の代わりにフェイルオーバーグループ名が取得されます。。CLUSTERPRO Ver3.0互換運用では、プライマリサーバ名とフェイルオーバーグループ名は同一のため、アプリケーションの変更は不要です。
片方向スタンバイクラスタ、双方向スタンバイクラスタ、パラレルクラスタでは、プライマリサーバ名とフェイルオーバーグループ名は一般的に異なりますので注意が必要です。本関数で得られたフェイルオーバーグループ名をそのまま状態取得や監視に使用する場合は、状態取得関数及び監視関数も同様にプライマリサーバ名がフェイルオーバーグループ名に変更されていますので、アプリケーションの変更は不要です。

参照

ArmClientGetGroupFromVirtualIP

6.1.6.3 ArmClientGetClusterNameFromVirtualIP — 仮想 IP の属しているクラスタ名の取得

ARM1.1 より変更なし

ARM3.0 との互換性のためにのみ提供

概要

仮想IPアドレスを指定して、クラスタ名を取得します。

形式

```
BOOL ArmClientGetClusterNameFromVirtualIP (  
    DWORD    dwVirtualIP,  
    LPTSTR    lpszClusterName,  
    LPDWORD   lpdcchClusterName  
)
```

パラメータ

[i]	DwVirtualIP	仮想IPアドレス。ネットワークバイトオーダーで指定します。
[o]	LpszClusterName	クラスタ名（終端のNULL文字を含む）を受け取るバッファへのポインタ。
[io]	LpcchClusterName	lpszClusterNameで指定されるバッファのサイズを指定する変数へのポインタ。関数のリターン時にはバッファに格納された文字数（終端のNULL文字を含まない）がセットされます。指定バッファサイズが小さすぎた場合には、必要な文字数（終端のNULL文字を含む）がセットされ、拡張エラー情報で ARMERROR_BUFFER_TOO_SMALL が返却されます

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報を返却します。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアントAPIの初期化処理が行われていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_VIRTUALIP_NOT_FOUND	指定仮想IPアドレスが存在しません。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は指定した仮想IPアドレスが属するクラスタのクラスタ名を取得し、バッファlpszClusterNameに格納します。本関数はCLUSTERPRO Ver1.0との互換性のために用意されています。ArmClientGetGroupFromVirtualIPの機能の完全なサブセットであるため、ArmClientGetGroupFromVirtualIPを使用することを推奨します。

参照

ArmCliengGetGroupFromVirtualIP

6.1.6.4 ArmClientEnumVirtualIP — 仮想 IP アドレスの列挙

ARM4.0 新規

概要

クラスタに所属する仮想IPアドレスを列挙します。

形式

```
BOOL ArmClientEnumVirtualIP (
    LPCTSTR lpszClusterName,
    LPCTSTR lpszGroupName
    ENUMVIRTUALIPPROC EnumVirtualIPProc,
    LPARAM lParam
)
```

パラメータ

[i] lpszClusterName	クラスタ名を指定する、NULLで終了する文字列へのポインタです。 NULLあるいは空文字列を指定した場合は、登録された全てのクラスタが対象となります。 ローカルサーバのクラスタを指定する場合は、定数 ARMCLAPI_LOCAL_CLUSTER を指定することができます。この場合、ローカルサーバがクラスタサーバでなければエラー (ARMCLAPI_ERR_CLUSTER_NOT_FOUND) が返却されます。
[i] lpszGroupName	仮想IPアドレスが所属するフェイルオーバーグループ名を指定する、NULLで終了する文字列へのポインタです。 NULLあるいは空文字列を指定すると、クラスタに属する全てのグループが列挙の対象となります。
[i] EnumVirtualIPProc	フェイルオーバーグループ名を受け取る、アプリケーション定義のコールバック関数のアドレスです。(EnumVirtualIPProc関数参照)
[i] lParam	コールバック関数に渡される、アプリケーション定義の32ビット値です。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPROクライアントAPIが初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPROクライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPROクライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	指定クラスタ名が存在しません。
ARMCLAPI_ERR_GROUP_NOT_FOUND	フェイルオーバーグループ名が存在しません。
ARMCLAPI_ERR_CANCELLED	コールバック関数から列挙の中止が返却されました。

説明

本関数は、指定したクラスタ、フェイルオーバーグループに属する仮想IPアドレスを列挙します。本関数は、アプリケーション定義のコールバック関数EnumVirtualIPProcを順次呼び出し、各仮想IPアドレスを渡します。本関数は、仮想IPアドレスをすべて列挙するか、またはコールバック関数がFALSEを返却するまで、列挙を続けます。

参照

EnumVirtualIPProc

6.1.6.5 EnumVirtualIPProc — 仮想 IP アドレス列挙用コールバック関数

ARM4.0 新規

概要

ArmClientEnumVirtualIP関数で列挙される仮想IPアドレスを受け取るための、アプリケーション定義のコールバック関数です。

形式

```
BOOL CALLBACK EnumVirtualIPProc (  
    LPCTSTR lpszClusterName,  
    LPCTSTR lpszGroupName,  
    DWORD dwVirtualIP  
    LPARAM lParam  
)
```

パラメータ

[i]	lpszClusterName	クラスタ名を示す、NULLで終わる文字列へのポインタです。
[i]	lpszGroupName	フェイルオーバーグループ名を示す文字列へのポインタです。
[i]	dwVirtualIP	仮想IPアドレスです。ネットワークバイトオーダーで格納されます。
[i]	lParam	ArmClientEnumServer関数で指定されたアプリケーション定義の値です。

戻り値

TRUE	列挙を続ける場合に返却します。
FALSE	列挙を止める場合に返却します。

説明

EnumVirtualIPProc関数は、ArmClientEnumVirtualIP関数の実行結果として、クラスタに登録されている仮想IPアドレスを受け取るための、アプリケーション定義のコールバック関数です。仮想IPアドレスの列挙が完了するか、または本関数でFALSEを返却するまで、フェイルオーバーグループ名が本関数に順次受け渡されます。アプリケーションは、ArmClientEnumVirtualIP関数実行時に本関数のアドレスを指定しなければなりません。本関数に渡されるlpszClusterName、lpszGroupNameの指す文字列の内容は、本関数をリターンするまでの間でのみ有効です。これらの文字列の内容を保持する必要がある場合には、本関数内で文字列の内容を別の記憶領域へコピーしてください。

本関数の関数名“EnumVirtualIPProc”は、アプリケーション定義の関数名のプレースホルダです。

参照

ArmClientEnumVirtualIP

6.1.6.6 ArmClientGetIPAddrFromVirtualIP — 仮想 IP アドレスからの実 IP アドレスの取得

ARM1.1 より変更なし

概要

接続するサーバの実IPアドレスを取得します。

形式

```
BOOL ArmClientGetIPAddrFromVirtualIP(  
    DWORD dwVirtualIPAddr,  
    LPDWORD lpdwIPAddr  
)
```

パラメータ

[i] dwVirtualIPAddr	仮想IPアドレスです。ネットワークバイトオーダーで指定します。
[o] lpdwIPAddr	IPアドレスを受け取る変数へのポインタです。 IPアドレスはネットワークバイトオーダーで格納されます。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPROクライアントAPIが初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPROクライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPROクライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_VIRTUALIP_NOT_FOUND	指定された仮想IPアドレスが見つかりません。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、仮想IPアドレスから、現在動作しているサーバの実IPアドレスを問い合わせます。

6.1.7 仮想コンピュータ名からの情報の取得

仮想コンピュータ名から、クラスタ名及びフェイルオーバーグループ名を取得します。

6.1.7.1 ArmClientGetGroupFromVirtualComputer — 仮想コンピュータ名が所属しているフェイルオーバーグループ名の取得

ARM4.0 新規

概要

仮想コンピュータ名が所属しているクラスタ名及びフェイルオーバーグループ名を取得します。

形式

```
BOOL ArmClientGetGroupFromVirtualComputer (
    LPCTSTR lpszVirtualComputerName,
    LPCTSTR lpszClusterName,
    LPDWORD lpcchClusterName
    LPCTSTR lpszGroupName,
    LPDWORD lpcchGroupName
)
```

パラメータ

[i]	lpszVirtualComputerName	仮想コンピュータ名。ネットワークバイトオーダーで指定します。
[o]	lpszClusterName	クラスタ名（終端のNULL文字を含む）を受け取るバッファへのポインタです。
[io]	lpcchClusterName	lpszClusterNameで指定されるバッファのサイズを指定する変数へのポインタ。関数のリターン時にはバッファに格納された文字数（終端のNULL文字を含まない）がセットされます。指定バッファサイズが小さすぎた場合には、必要な文字数（終端のNULL文字を含む）がセットされ、拡張エラー情報でARMERROR_BUFFER_TOO_SMALLが返却されます。
[o]	lpszGroupName	仮想IPアドレスが所属しているフェイルオーバーグループ名（終端のNULL文字を含む）を受け取るバッファへのポインタ。
[io]	lpcchGroupName	lpszGroupNameで指定されるバッファのサイズを指定する変数へのポインタです。関数のリターン時にはバッファに格納された文字数（終端のNULL文字を含まない）がセットされます。指定バッファサイズが小さすぎた場合には、必要な文字数（終端のNULL文字を含む）がセットされ、拡張エラー情報でARMERROR_BUFFER_TOO_SMALLが返却されます。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報を返却します。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPROクライアントAPIの初期化処理が行われていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPROクライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPROクライアントとの通信でエラーが発生しました。

ARMCLAPI_ERR_BUFFER_TOO_SMALL	文字列を格納するバッファのサイズが小さすぎます。
ARMCLAPI_ERR_VCOM_NOT_FOUND	指定された仮想コンピュータ名が存在しません。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は仮想コンピュータ名を指定して、それに対応するクラスタ名及びフェイルオーバーグループ名を取得します。クラスタ名をバッファ `lpszClusterName` に格納します。フェイルオーバーグループ名を `lpszGroupName` に格納します。仮想コンピュータ名を監視する場合には、まずこの関数でクラスタ名及びフェイルオーバーグループ名を得ます。そして、それを、フェイルオーバーグループの状態監視開始の入力パラメータとして使用することで、監視が開始されます。

6.1.7.2 ArmClientEnumVirtualComputer — 仮想コンピュータ名の列挙

ARM4.0 新規

概要

クラスタに所属する仮想コンピュータ名を列挙します。

形式

```
BOOL ArmClientEnumVirtualComputer(  
    LPCTSTR lpszClusterName,  
    LPCTSTR lpszGroupName  
    ENUMVIRTUALCOMPUTERPROC EnumVirtualComputerProc,  
    LPARAM lParam  
)
```

パラメータ

[i] <i>lpszClusterName</i>	クラスタ名を指定する、NULLで終了する文字列へのポインタです。 NULLあるいは空文字列を指定した場合は、登録された全てのクラスタが対象となります。 ローカルサーバのクラスタを指定する場合は、定数 ARMCLAPI_LOCAL_CLUSTER を指定することができます。この場合、ローカルサーバがクラスタサーバでなければエラー (ARMCLAPI_ERR_CLUSTER_NOT_FOUND) が返却されます。
[i] <i>lpszGroupName</i>	仮想コンピュータ名が所属するフェイルオーバーグループ名を指定する、NULLで終了する文字列へのポインタです。 NULLあるいは空文字列を指定すると、クラスタに属する全てのグループが列挙の対象となります。
[i] <i>EnumVirtualComputerProc</i>	仮想コンピュータ名を受け取る、アプリケーション定義のコールバック関数のアドレスです。(EnumVirtualComputerProc関数参照)
[i] <i>Lparam</i>	コールバック関数に渡される、アプリケーション定義の32ビット値です。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPROクライアントAPIが初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPROクライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPROクライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	指定クラスタ名が存在しません。
ARMCLAPI_ERR_GROUP_NOT_FOUND	フェイルオーバーグループ名が存在しません。
ARMCLAPI_ERR_CANCELLED	コールバック関数から列挙の中止が返却されました。

説明

本関数は、指定したクラスタ、フェイルオーバーグループに属する仮想コンピュータ名を列挙します。本関数は、アプリケーション定義のコールバック関数EnumVirtualComputerProcを順次呼び出し、各仮想コンピュータ名を渡します。本関数は、仮想コンピュータ名をすべて列挙するか、またはコールバック関数がFALSEを返却するまで、列挙を続けます。

参照

EnumVirtualComputerProc

6.1.7.3 EnumVirtualComputerProc — 仮想コンピュータ名列挙用コールバック関数

ARM4.0 新規

概要

ArmClientEnumVirtualComputer関数で列挙される仮想コンピュータ名を受け取るための、アプリケーション定義のコールバック関数です。

形式

```
BOOL CALLBACK EnumVirtualComputerProc (  
    LPCTSTR lpszClusterName,  
    LPCTSTR lpszGroupName,  
    LPCTSTR lpszVirtualComputer  
    LPARAM lParam  
)
```

パラメータ

- | | | |
|-----|---------------------|--|
| [i] | lpszClusterName | クラスタ名を示す、NULLで終わる文字列へのポインタです。 |
| [i] | lpszGroupName | フェイルオーバーグループ名を示す文字列へのポインタです。 |
| [i] | lpszVirtualComputer | 仮想コンピュータ名を示す文字列へのポインタです。 |
| [i] | lParam | ArmClientEnumServer関数で指定されたアプリケーション定義の値です。 |

戻り値

- | | |
|-------|-----------------|
| TRUE | 列挙を続ける場合に返却します。 |
| FALSE | 列挙を止める場合に返却します。 |

説明

EnumVirtualComputerProcは、ArmClientEnumVirtualComputer関数の実行結果として、クラスタに登録されている仮想コンピュータ名を受け取るための、アプリケーション定義のコールバック関数です。仮想コンピュータ名の列挙が完了するか、または本関数でFALSEを返却するまで、仮想コンピュータ名が本関数に順次受け渡されます。

アプリケーションは、ArmClientEnumVirtualComputer関数実行時に本関数のアドレスを指定しなければなりません。

本関数に渡されるlpszClusterName、lpszGroupNameの指す文字列の内容は、本関数をリターンするまでの間でのみ有効です。これらの文字列の内容を保持する必要がある場合には、本関数内で文字列の内容を別の記憶領域へコピーしてください。

本関数の関数名“EnumVirtualComputerProc”は、アプリケーション定義の関数名のプレースホルダです。

参照

ArmClientEnumVirtualComputer

6.1.8 論理サービス名

論理サービスが設定されたフェイルオーバーグループの一覧を取得します。

6.1.8.1 ArmClientEnumGroupFromLogicalService — 論理サービス名に対応するフェイルオーバーグループの列挙

ARM4.0 新規

概要

論理サービス名が登録されているフェイルオーバーグループ名を列挙します。

形式

```
BOOL ArmClientEnumGroupFromLogicalService (
    LPCTSTR lpszClusterName,
    LPCTSTR lpszLogicalServiceName,
    ENUMGROUPFROMLOGICALSERVICEPROC EnumGroupFromLogicalServiceProc,
    LPARAM lParam
)
```

パラメータ

[i] lpszClusterName	クラスタ名を指定する、NULLで終了する文字列へのポインタです。 NULLあるいは空文字列を指定した場合は、登録された全てのクラスタが対象となります。 ローカルサーバのクラスタを指定する場合は、定数 ARMCLAPI_LOCAL_CLUSTER を指定することができます。この場合、ローカルサーバがクラスタサーバでなければエラー (ARMCLAPI_ERR_CLUSTER_NOT_FOUND) が返却されます。
[i] lpszLogicalServiceName	論理サービス名を指定する、NULLで終了する文字列へのポインタです。
[i] EnumGroupFromLogicalServiceProc	フェイルオーバーグループ名を受け取る、アプリケーション定義のコールバック関数のアドレスです。 (EnumGroupFromLogicalServiceProc関数参照)
[i] lParam	コールバック関数に渡される、アプリケーション定義の32ビット値です。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPROクライアントAPIが初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPROクライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPROクライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	指定されたクラスタ名が存在しません。

ARMCLAPI_ERR_LOGICAL_SERVICE_NOT_FOUND	指定された論理サービス名が見つかりません。
ARMCLAPI_ERR_CANCELLED	コールバック関数から列挙の中止が返却されました。

説明

本関数は、論理サービス名から、それに対応するフェイルオーバーグループ名を列挙します。本関数は、アプリケーション定義のコールバック関数EnumGroupFromLogicalServiceProcを順次呼び出し、各サーバのコンピュータ名を渡します。本関数は、該当するフェイルオーバーグループをすべて列挙するか、またはコールバック関数がFALSEを返却するまで、列挙を続けます。

参照

EnumGroupFromLogicalServiceProc

6.1.8.2 ArmClientEnumServerFromLogicalService — 論理サービス名に対応するフェイルオーバーグループの列挙

プライマリサーバ名→フェイルオーバーグループ名

ARM3.0 との互換性のためにのみ提供

クラスタ名に NULL 指定可能

概要

論理サービス名から、それに対応するフェイルオーバーグループ名を列挙します。CLUSTERPRO Ver3.0との互換性のためにのみ提供する関数です。ArmClientEnumGroupFromLogicalServiceと関数名以外は全く同一です。

形式

```
BOOL ArmClientEnumServerFromLogicalService (
    LPCTSTR IpszClusterName,
    LPCTSTR IpszLogicalServiceName,
    ENUMGROUPFROMLOGICALSERVICEPROC EnumGroupFromLogicalServiceProc,
    LPARAM lParam
)
```

互換性留意事項

CLUSTERPRO Ver3.0以前では、本関数ではプライマリサーバ名が取得されました。CLUSTERPRO Ver4.x以降では、プライマリサーバ名の代わりにフェイルオーバーグループ名が取得されます。。

CLUSTERPRO Ver3.0互換運用では、プライマリサーバ名とフェイルオーバーグループ名は同一のため、アプリケーションの変更は不要です。

片方向スタンバイクラスタ、双方向スタンバイクラスタ、パラレルクラスタでは、プライマリサーバ名とフェイルオーバーグループ名は一般的に異なりますので注意が必要です。本関数で得られたフェイルオーバーグループ名をそのまま状態取得や監視に使用する場合は、状態取得関数及び監視関数も同様にプライマリサーバ名がフェイルオーバーグループ名に変更されていますので、アプリケーションの変更は不要です。

参照

EnumGroupFromLogicalServiceProc

6.1.8.3 EnumGroupFromLogicalServiceProc — サーバ列挙用コールバック関数

プライマリサーバ名→フェイルオーバーグループ名

概要

ArmClientEnumGroupFromLogicalService関数で列挙されるフェイルオーバーグループ名を受け取るための、アプリケーション定義のコールバック関数です。

形式

```
BOOL CALLBACK EnumGroupFromLogicalServiceProc (  
    LPCTSTR lpszClusterName,  
    LPCTSTR lpszGroupName,  
    LPCTSTR lpszLogicalServiceName  
    LPARAM lParam  
)
```

パラメータ

[i]	lpszClusterName	クラスタ名です。NULLで終わる文字列へのポインタです。
[i]	lpszGroupName	指定された論理サービス名に対応するフェイルオーバーグループ名です。NULLで終わる文字列へのポインタです。
[i]	lpszLogicalServiceName	論理サービス名です。NULLで終わる文字列へのポインタです。
[i]	lParam	ArmClientEnumServerFromLogicalService関数で指定されたアプリケーション定義の値です。

戻り値

TRUE	列挙を続ける場合に返却します。
FALSE	列挙を止める場合に返却します。

説明

EnumGroupFromLogicalServiceProc関数は、ArmClientEnumGroupFromLogicalService関数の実行結果として、論理サービス名に対応するフェイルオーバーグループ名を受け取るための、アプリケーション定義のコールバック関数です。フェイルオーバーグループの列挙が完了するか、または本関数でFALSEを返却するまで、フェイルオーバーグループ名が本関数に順次受け渡されます。

アプリケーションは、ArmClientEnumGroupFromLogicalService関数実行時に本関数のアドレスを指定しなければなりません。

本関数に渡されるlpszClusterName, lpszGroupName, lpszLogicalServiceNameの指す文字列の内容は、本関数をリターンするまでの間でのみ有効です。これらの文字列の内容を保持する必要がある場合には、本関数内で文字列の内容を別の記憶領域へコピーしてください。

本関数の関数名“EnumGroupProcFromLogicalService”は、アプリケーション定義の関数名のプレースホルダーです。

参照

ArmClientEnumGroupFromLogicalService

6.1.9 クラスタパラメータの取得

指定したクラスタのユーザ指定パラメータを取得します。

6.1.9.1 ArmClientGetHeartbeatTimeout — ハートビートタイムアウト時間の取得

ARM2.0 より変更なし

概要

CLUSTERPROに設定されたハートビートタイムアウト時間(秒)を取得します。

形式

```
BOOL ArmClientGetHeartbeatTimeout(  
    LPCTSTR lpszClusterName,  
    LPDWORD lpdwTimeout  
)
```

パラメータ

[i] lpszClusterName	クラスタ名を指定する、NULLで終了する文字列へのポインタです。 ローカルサーバのクラスタを指定する場合は、定数 ARMCLAPI_LOCAL_CLUSTER を指定することができます。この場合、ローカルサーバがクラスタサーバでなければエラー (ARMCLAPI_ERR_CLUSTER_NOT_FOUND) が返却されます。
[o] lpdwTimeout	指定されたクラスタに設定されたハートビートタイムアウトを受け取る変数へのポインタです。ハートビートタイムアウトの単位は秒です。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報が返却されます。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPROクライアントAPIが初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPROクライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPROクライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	指定されたクラスタ名が存在しません。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、指定されたクラスタに設定されたハートビートタイムアウト時間を問い合わせます。ハートビートタイムアウト時間は、CLUSTERPROが相手サーバからのハートビートの停止から、相手サーバのダウンを判定する時間です。本関数で得られる時間は、クライアントAPがサーバとの通信異常を検出した場合に、フェイルオーバー発生通知を待つ場合の時間の目安として使用することができます。

補足

本関数は CLUSTERPRO Ver2.0 以降でサポートされます。拡張エラー情報 ARMCLAPI_ERR_NOT_SUPPORTEDは、CLUSTERPRO Ver2.0で提供するARMCL2ST.LIB をリンクしたAPが、CLUSTERPRO Ver1.1の環境で実行された場合に返却されるエラーです。

6.1.9.2 ArmClientGetMasterServer — マスタサーバの取得

ARM1.1 より変更なし

概要

クラスタのマスタサーバ名を取得します。

形式

```
BOOL ArmClientGetMasterServer (  
    LPCTSTR lpszClusterName,  
    LPTSTR lpszServerName,  
    LPDWORD lpcchServerName  
)
```

パラメータ

[i] lpszClusterName	クラスタ名を指定する、NULLで終了する文字列へのポインタです。 ローカルサーバのクラスタを指定する場合は、ARMCLAPI_LOCAL_CLUSTERを指定することができます。この場合、ローカルサーバがクラスタサーバでなければエラー (ARMCLAPI_ERR_CLUSTER_NOT_FOUND) を返却します。
[o] lpszServerName	主サーバ名 (終端のNULL文字を含む) を受け取るバッファへのポインタです。
[io] LpcchServerName	lpszServerNameで指定されるバッファのサイズを指定する変数へのポインタです。関数のリターン時にはバッファに格納された文字数 (終端のNULL文字を含まない) をセットします。 指定バッファサイズが小さすぎた場合には、必要な文字数 (終端のNULL文字を含む) がセットし、拡張エラー情報でARMERROR_BUFFER_TOO_SMALLが返却します。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報を返却します。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO APIの初期化処理が行われていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPROクライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPROクライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_BUFFER_TOO_SMALL	データ返却用バッファが小さすぎます。
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	指定クラスタ名が存在しません。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は指定したクラスタのマスタサーバのサーバ名を取得し、バッファlpszServerNameに格納します。マスタサーバとはクラスタを構成するサーバのうちの時刻同期の基準となるサーバです。

6.1.10 ローカルサーバ関連API

6.1.10.1 ArmClientCheckClusterServer — ローカルサーバの情報

ARM1.1 より変更なし

概要

ローカルコンピュータがクラスタサーバかどうかを判別します。

形式

```
BOOL ArmClientCheckClusterServer(  
    LPDWORD lpdwHostType  
)
```

パラメータ

[o] lpdwHostType	ローカルコンピュータの種別
	ARMCLAPI_HOSTTYPE_IS_SERVER クラスタサーバ
	ARMCLAPI_HOSTTYPE_IS_CLIENT CLUSTERPROクライアント
	ARMCLAPI_HOSTTYPE_IS_NONE 上記のいずれも該当しません。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報を返却します。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPROクライアントAPIの初期化処理が行われていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPROクライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPROクライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、実行されているコンピュータがクラスタサーバであるか、クライアントであるかを判別します。

6.1.10.2 ArmClientGetServerName — ローカルサーバ名の取得

ARM1.1 より変更なし

概要

ローカルサーバのサーバ名を取得します。

形式

```
BOOL ArmClientGetServerName (
    LPTSTR  lpszServerName,
    LPDWORD lpcchServerName
)
```

パラメータ

[o] lpszServerName	ローカルサーバのサーバ名（終端のNULL文字を含む）を受け取るバッファへのポインタです。
[io] lpcchServerName	lpszServerNameで指定されるバッファのサイズを指定する変数へのポインタです。関数のリターン時に、バッファに格納された文字数（終端のNULL文字を含まない）をセットします。指定バッファサイズが小さすぎた場合には、必要な文字数（終端のNULL文字を含む）をセットし、拡張エラー情報でARMCLAPI_ERR_BUFFER_TOO_SMALLを返却します。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報を返却します。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPROクライアントAPIの初期化処理が行われていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPROクライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPROクライアントの通信でエラーが発生しました。
ARMCLAPI_ERR_NOT_CLUSTER_SERVER	ローカルホストはクラスタサーバではありません。
ARMCLAPI_ERR_REGISTRY_ACCESS_FAILED	レジストリのアクセスに失敗しました。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_BUFFER_TOO_SMALL	データ返却用バッファが小さすぎます。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、ローカルサーバのサーバ名を取得し、バッファlpszServerNameに格納します。ローカルホストがクラスタに所属していない場合、本関数はFALSEを返却し拡張エラー情報でARMCLAPI_ERR_NOT_CLUSTER_SERVERを返却します。

6.1.10.3 ArmClientGetClusterName — クラスタ名の取得

ARM1.1 より変更なし

概要

ローカルサーバが所属するクラスタの名前を取得します。

形式

```
BOOL ArmClientGetClusterName(  
    LPTSTR lpszClusterName,  
    LPDWORD lpcchClusterName  
)
```

パラメータ

[o] lpszClusterName	クラスタ名（終端のNULL文字を含む）を受け取るバッファへのポインタです。
[io] lpcchClusterName	lpszClusterNameで指定されるバッファのサイズを指定する変数へのポインタです。関数のリターン時に、バッファに格納された文字数（終端のNULL文字を含まない）をセットします。指定バッファサイズが小さすぎた場合には、必要な文字数（終端のNULL文字を含む）をセットし、拡張エラー情報でARMCLAPI_ERR_BUFFER_TOO_SMALLを返却します。

戻り値

TRUE	関数が正常終了しました。
FALSE	関数が異常終了しました。GetLastError関数により拡張エラー情報を返却します。

拡張エラー情報

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPROクライアントAPIの初期化処理が行われていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPROクライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPROクライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_NOT_CLUSTER_SERVER	ローカルホストはクラスタサーバではありません。
ARMCLAPI_ERR_REGISTRY_ACCESS_FAILED	レジストリのアクセスに失敗しました。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_BUFFER_TOO_SMALL	データ返却用バッファが小さすぎます。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、ローカルサーバが所属するクラスタの名前を取得し、バッファlpszClusterNameに格納します。ローカルサーバがクラスタに所属していない場合、本関数はFALSEを返却し、拡張エラー情報でARMCLAPI_ERR_NOT_CLUSTER_SERVERを返却します。

6.2 構造体

6.2.1 ARMCLAPI_FAILOVER_INFO – フェイルオーバー情報

プライマリサーバ名→フェイルオーバーグループ名

概要

ARMCLAPI_FAILOVER_INFO構造体は、フェイルオーバー発生／終了時の情報を格納します。

形式

```
typedef struct _ARMCLAPI_FAILOVER_INFO{
    DWORD    Padding;
    TCHAR    ClusterName[CLUSTER_NAME_LEN];
    TCHAR    GroupName[GROUP_NAME_LEN];
    DWORD    VirtualIP;
    TCHAR    SrcComputerName[SERVER_NAME_LEN];
    DWORD    SrcIP;
    TCHAR    DstComputerName[SERVER_NAME_LEN];
    DWORD    DstIP;
    DWORD    Event;
    DWORD    StatusEx;
} ARMCLAPI_FAILOVER_INFO;
```

メンバ

Padding	未使用	
ClusterName	フェイルオーバーが発生したクラスタ名	
GroupName	フェイルオーバーグループ名	
VirtualIP	仮想IPアドレス名(ネットワークバイトオーダー、フェイルオーバーグループに複数仮想IPアドレスが登録されていても一つのみ返却する)	
SrcComputerName	フェイルオーバー開始時	フェイルオーバー元のサーバ名
	フェイルオーバー完了時	不定
	シャットダウン時	シャットダウンしたサーバ名
	起動時	不定
SrcIP	SrcComputerNameの実IP(ネットワークバイトオーダー)	
DstComputerName	フェイルオーバー開始時	不定
	フェイルオーバー完了時	フェイルオーバー先のサーバ名
	シャットダウン時	不定
	起動時	起動したサーバ名
DstIP	DstComputerNameの実IP(ネットワークバイトオーダー)	
Event	CLUSTERPROモニタが検出した状態変更の種類	
	ARMCLAPI_EVENT_FAILOVER_START	フェイルオーバー発生
	ARMCLAPI_EVENT_FAILOVER_END	フェイルオーバー終了
	ARMCLAPI_EVENT_SHUTDOWN	シャットダウン
	ARMCLAPI_EVENT_BOOT	サーバ起動
StatusEx	現在の状態	
	ARMCLAPI_GROUP_STATUS_RUNNING	正常動作中
	ARMCLAPI_GROUP_STATUS_MISS	資源獲得に失敗
	ARMCLAPI_GROUP_STATUS_STOPPED	停止中
	ARMCLAPI_GROUP_STATUS_FAILOVER	フェイルオーバー処理中

6.2.1.1 ARM_FAILOVER_INFO_EX – フェイルオーバー情報 (Ver3.0 互換用)

プライマリサーバ名→フェイルオーバーグループ名

ARM3.0 との互換性のためにのみ提供

概要

ARM_FAILOVER_INFO_EX構造体は、フェイルオーバー発生／終了時の情報を格納します。

形式

```
typedef struct _ARM_FAILOVER_INFO_EX{
    DWORD    Padding;
    TCHAR    ClusterName[CLUSTER_NAME_LEN];
    TCHAR    ServerName[GROUP_NAME_LEN];
    DWORD    VirtualIP;
    TCHAR    SrcComputerName[SERVER_NAME_LEN];
    DWORD    SrcIP;
    TCHAR    DstComputerName[SERVER_NAME_LEN];
    DWORD    DstIP;
    DWORD    Event;
    DWORD    StatusEx;
} ARM_FAILOVER_INFO_EX;
```

メンバ

Padding	未使用	
ClusterName	フェイルオーバーが発生したクラスタ名	
ServerName	フェイルオーバーグループ名 (Ver3.0互換運用ではプライマリサーバ名と同一)	
VirtualIP	仮想IPアドレス名 (ネットワークバイトオーダー、フェイルオーバーグループに複数仮想IPアドレスが登録されていても一つのみ返却する)	
SrcComputerName	フェイルオーバー開始時	フェイルオーバー元のサーバ名
	フェイルオーバー完了時	不定
	シャットダウン時	シャットダウンしたサーバ名
	起動時	不定
SrcIP	SrcComputerNameの実IP (ネットワークバイトオーダー)	
DstComputerName	フェイルオーバー開始時	不定
	フェイルオーバー完了時	フェイルオーバー先のサーバ名
	シャットダウン時	不定
	起動時	起動したサーバ名
DstIP	DstComputerNameの実IP (ネットワークバイトオーダー)	
Event	CLUSTERPROモニタが検出した状態変更の種類	
	ARMCLAPI_EVENT_FAILOVER_START	フェイルオーバー発生
	ARMCLAPI_EVENT_FAILOVER_END	フェイルオーバー終了
	ARMCLAPI_EVENT_SHUTDOWN	シャットダウン
	ARMCLAPI_EVENT_BOOT	サーバ起動
StatusEx	現在の状態	
	ARMCLAPI_GROUP_STATUS_RUNNING	正常動作中
	ARMCLAPI_GROUP_STATUS_MISS	資源獲得に失敗
	ARMCLAPI_GROUP_STATUS_STOPPED	停止中
	ARMCLAPI_GROUP_STATUS_FAILOVER	フェイルオーバー処理中

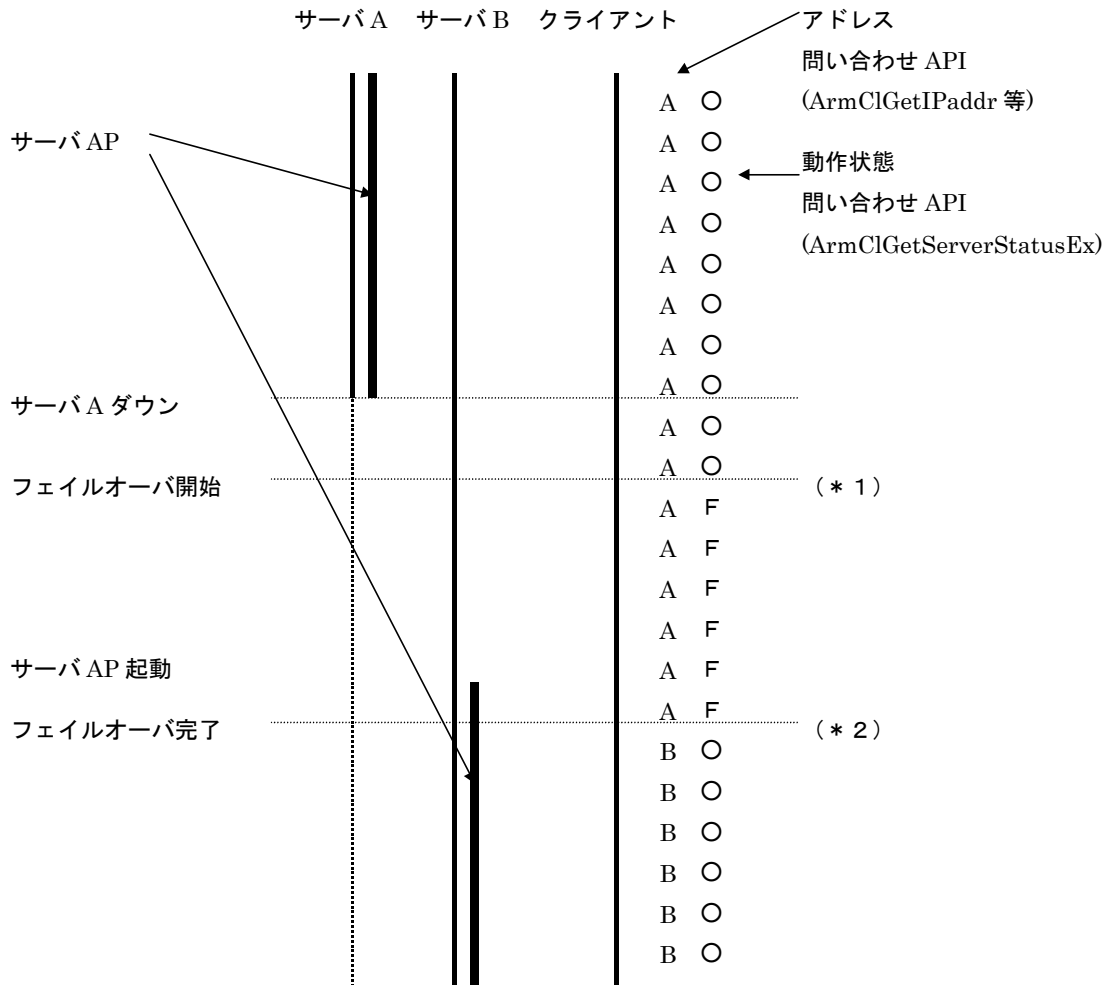
構造体名、及びメンバ名 (ServerName と GroupName) が異なるだけで、構成・内容は ARMCLAPI_FAILOVER_INFO 構造体と同じです。

6.3 タイムチャート

サーバの状態の変化に対するCLUSTERPROクライアントAPIの動作を、タイムチャートを用いて説明します。タイムチャート上で、動作状態問い合わせAPIの記号○×Fは、それぞれ下記の状態を示しています。

- ARMCLAPI_GROUP_STATUS_RUNNING 正常動作中
- × ARMCLAPI_GROUP_STATUS_STOPPED 停止中
- F ARMCLAPI_GROUP_STATUS_FAILOVER フェイルオーバー処理中

6.3.1 フェイルオーバー時のタイムチャート



(* 1) ポップアップメッセージ表示

「クラスタ**でグループ1のフェイルオーバーを開始しました.

サーバ A(x.x.x.x)→サーバ B(x.x.x.x)」

フェイルオーバー通知コールバック(ARM_EVENT_FAILOVER_START)

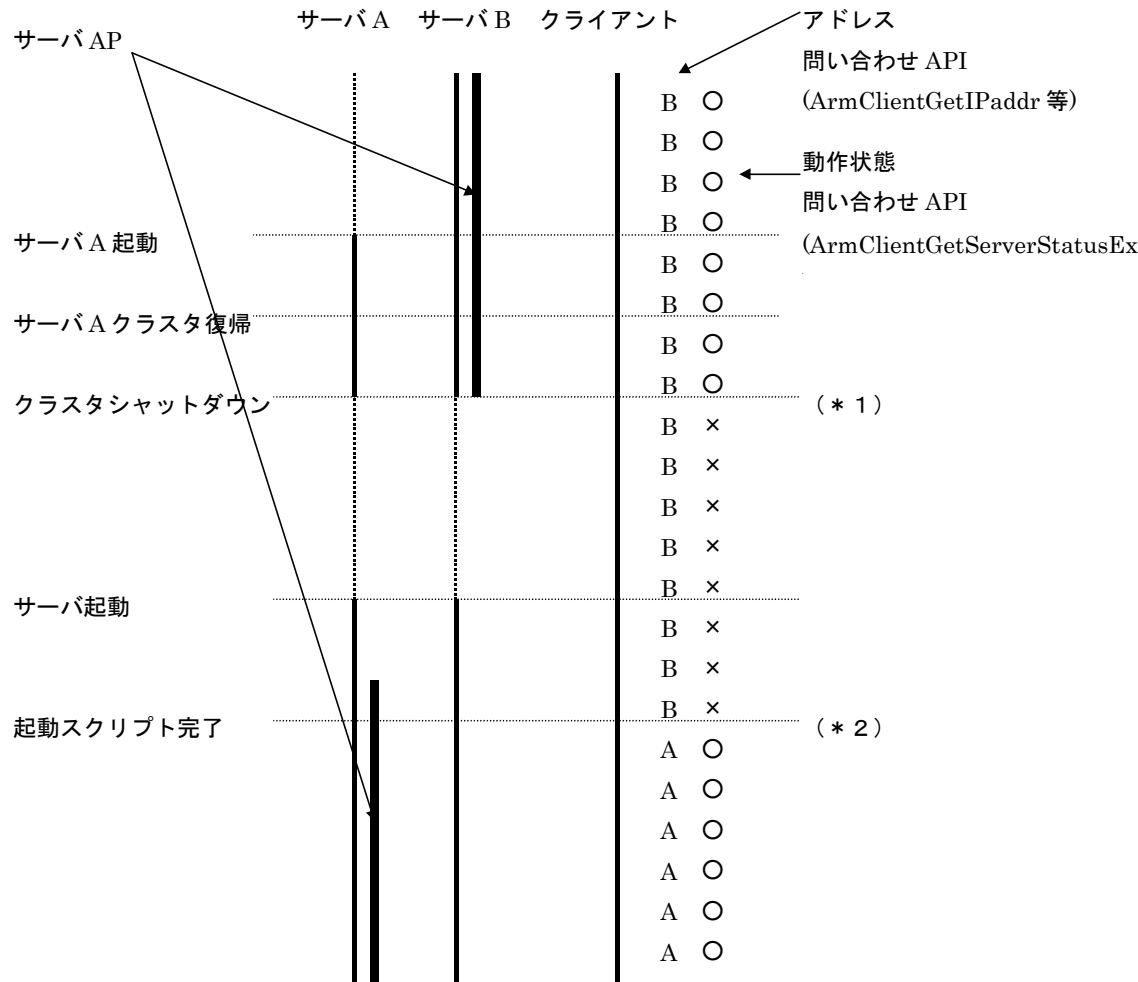
(* 2) ポップアップメッセージ表示

「クラスタ**でグループ1のフェイルオーバが終了しました.

サーバ A(x.x.x.x)→サーバ B(x.x.x.x)」

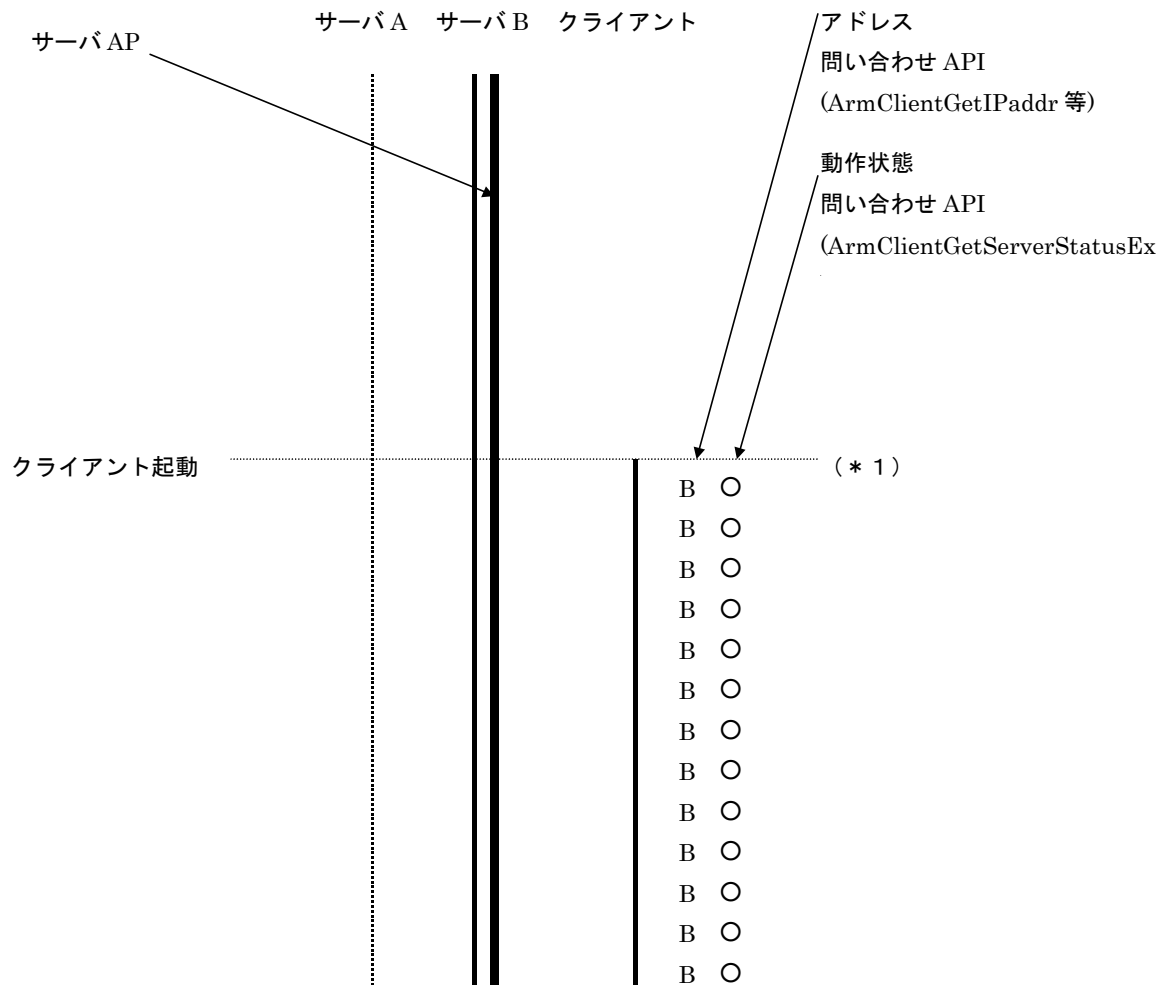
フェイルオーバー通知コールバック(ARM_EVENT_FAILOVER_END)

6.3.2 クラスタ復帰時のタイムチャート



- (* 1) ポップアップメッセージ表示
「クラスタ ** でグループ 1 が停止しました.
サーバ B(x.x.x.x) 」
フェイルオーバー通知コールバック(ARM_EVENT_SHUTDOWN)
- (* 2) ポップアップメッセージ表示
「クラスタ ** でグループ 1 が起動しました.
サーバ A(x.x.x.x)」

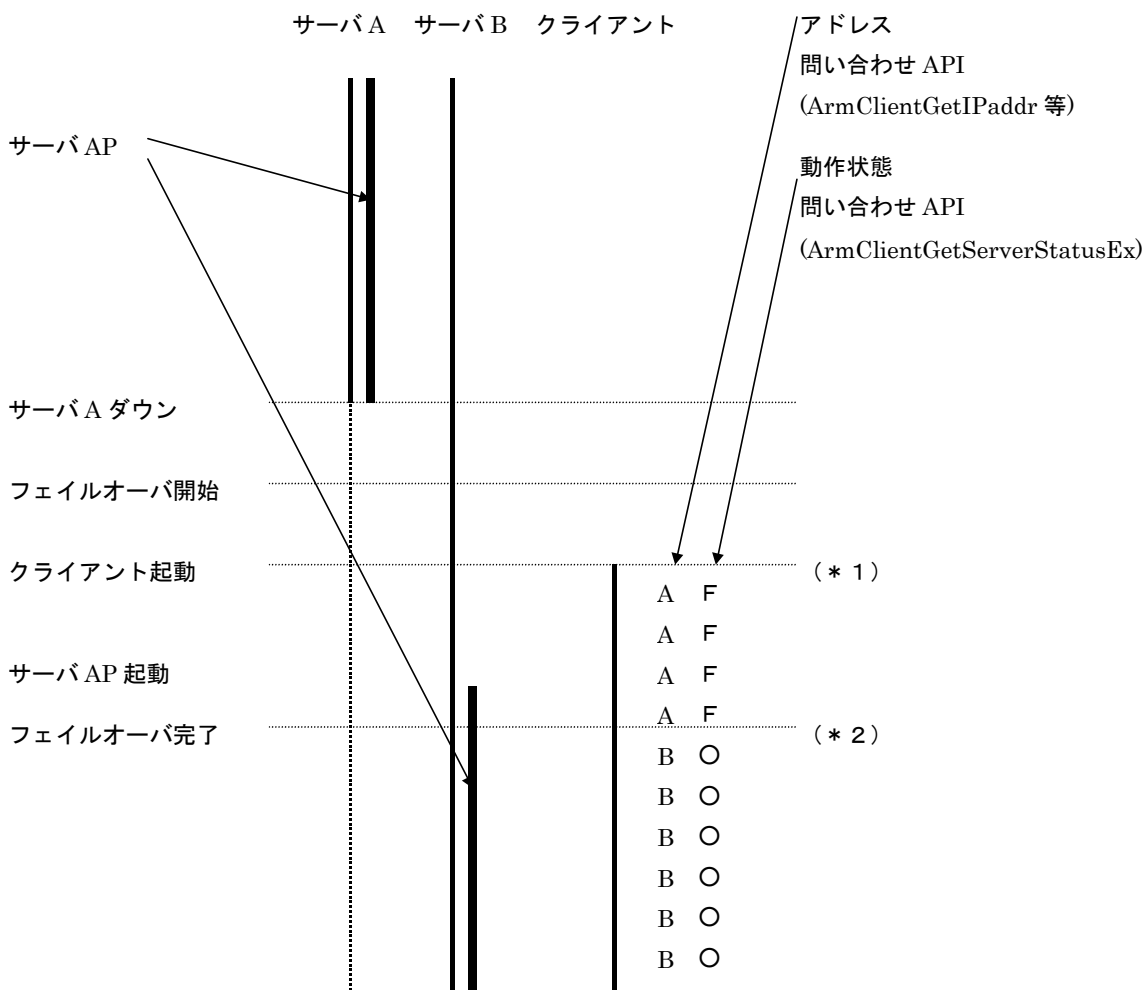
6.3.3 クライアント起動時にバックアップ運用中である場合のタイムチャート



(* 1) ポップアップメッセージ表示

「クラスタ**でグループ1がバックアップサーバ上で起動しました。
サーバ B(x.x.x.x)」

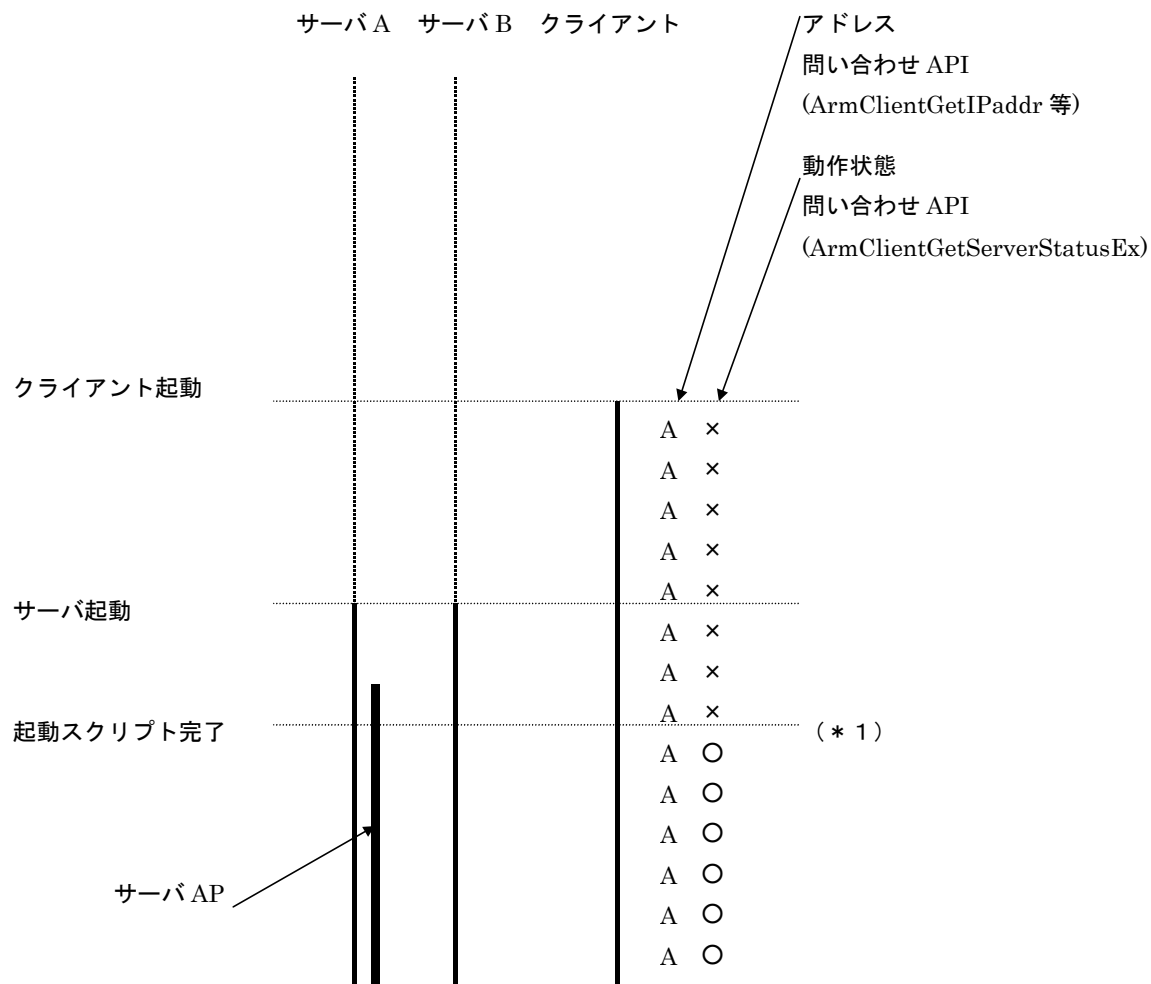
6.3.4 フェイルオーバー中にクライアントが起動した場合のタイムチャート



(* 1) ポップアップメッセージ表示
「クラスタ * * でグループ 1 のフェイルオーバーを開始しました.
サーバ A(x.x.x.x)→サーバ B(x.x.x.x)」

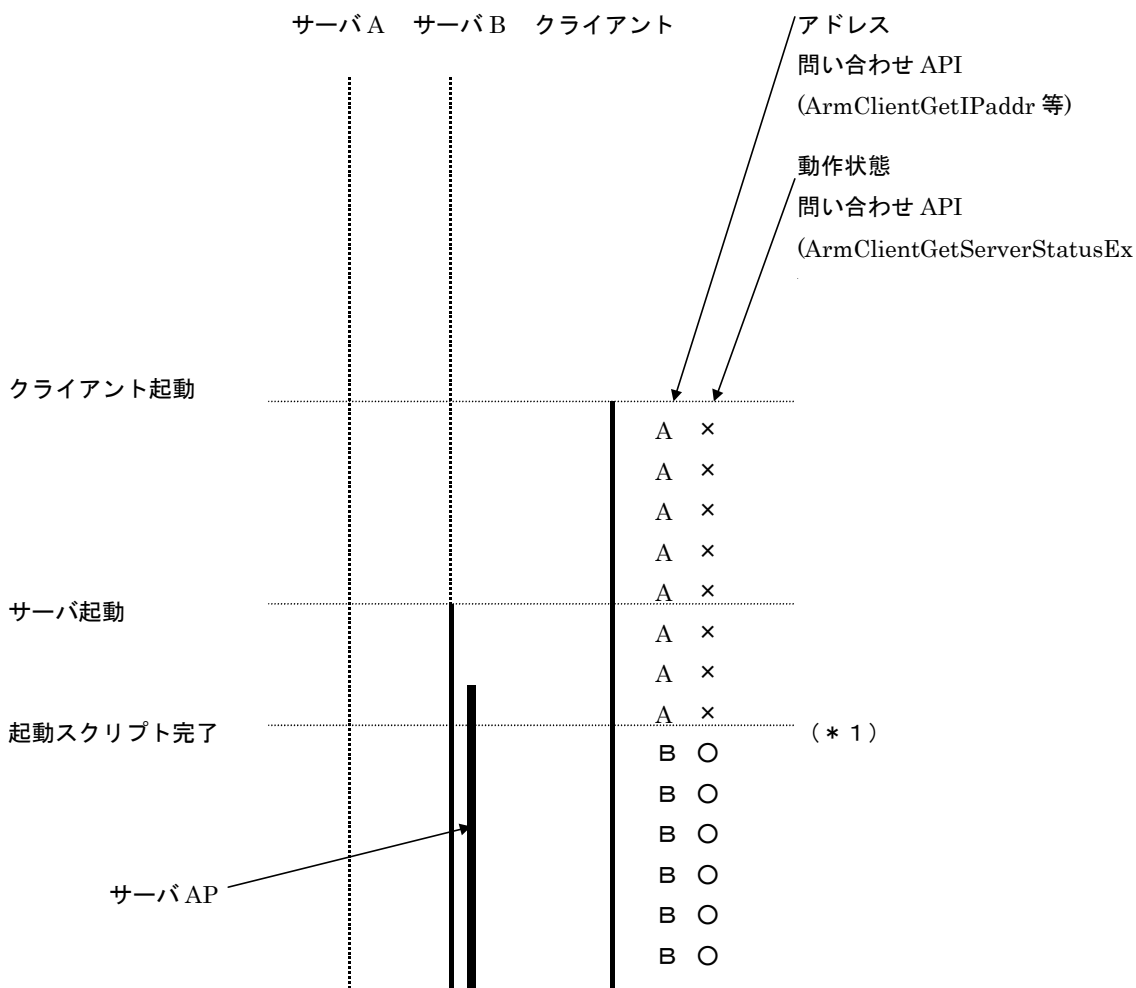
(* 2) ポップアップメッセージ表示
「クラスタ * * でグループ 1 のフェイルオーバーが終了しました.
サーバ A(x.x.x.x)→サーバ B(x.x.x.x)」
フェイルオーバー通知コールバック (ARM_EVENT_FAILOVER_END)

6.3.5 クラスタシャットダウン中にクライアントが起動した場合のタイムチャート



(* 1) フェイルオーバー通知コールバック (ARM_EVENT_BOOT)

6.3.6 クライアントが起動した後、サーバBのみ起動した場合のタイムチャート



(* 1) ポップアップメッセージ表示
「クラスタ ** のグループ 1 がバックアップ上で起動しました.
サーバ B(x.x.x.x)」
フェイルオーバー通知コールバック(ARM_EVENT_BOOT)

6.4 注意事項

6.4.1 32bit版クライアントアプリケーションと16bit版クライアントアプリケーションを同一クライアント上で実行させる場合の注意

CLUSTERPRO Ver4.x クライアントAPIは、同一クライアント上で、32bit版と16bit版を同時に実行させることが可能です。但し、以下のような注意が必要です。

- * CLUSTERPRO Ver1.0互換クライアントAPIを使用した16bit版クライアントアプリケーションは、Windows 3.1のみで動作可能です。
- * CLUSTERPRO Ver4.xクライアントAPIを使用した16bit版クライアントアプリケーションは、armcl2.lib(16bit版)あるいは、armcl2st.lib(16bit版)をリンク時に指定した場合に、Windows NT, 95上で動作させることが可能です。
- * CLUSTERPRO Ver4.xクライアントAPIを使用した16bit版クライアントアプリケーションで、armcl2.lib(16bit版) , armcl2st.lib(16bit版)を使用せず、クライアントアプリケーション内で独自にLoadLibrary関数でDLLをロードする場合は、DLL名にarmcl216.dllを指定してください。

7 クライアントAPI詳細 (Visual Basic)

7.1 関数

7.1.1 クライアントAPIのVisual Basic版概要

7.1.1.1 動作環境

Visual Basic 向け CLUSTERPRO クライアント API は、Visual Basic Ver5 以降に対応しています。

Visual Basic 向け CLUSTERPRO クライアント API は、CLUSTERPRO クライアントの動作している環境、もしくは、クラスタサーバの動作しているサーバ上でのみ、動作します。CLUSTERPRO クライアントではクライアント設定に登録されているクラスタについて利用可能です。クラスタサーバ上のクライアント設定に登録されているクラスタ、および、そのサーバの所属するローカルクラスタ (ARMCLAPI_LOCAL_CLUSTER) について利用可能です (したがって、同じクラスタでも利用環境によって結果が異なります)。

7.1.1.2 APIの初期化と終了

Visual Basic 向け CLUSTERPRO クライアント API を使用するクライアントアプリケーションは、ArmClientGetVersionInfo 関数以外の CLUSTERPRO クライアント API 関数を使用する前に、必ず ArmClientInitializeApiEx 関数を実行して CLUSTERPRO クライアント API の初期化を行わなければなりません。

また、ArmClientInitializeApiEx 関数を実行した場合には、アプリケーションを終了する前に、ArmClientTerminateApiEx 関数を実行して、CLUSTERPRO クライアント API の終了処理を行わなければなりません。

7.1.1.3 タイムチャート

「6.3 タイムチャート」を参照してください。Visual C++版と動作は同じです。

7.1.2 API 一覧

初期化／終了	
ArmClientInitializeApiEx()	クライアント API の初期化
ArmClientTerminateApiEx()	クライアント API の終了
一覧の列挙	
ArmClientEnumCluster()	登録されたクラスタの列挙
ArmClientEnumServer()	サーバの列挙
ArmClientEnumServerFromGroup()	フェイルオーバーグループが移動可能なサーバの列挙
ArmClientEnumGroup()	フェイルオーバーグループが移動可能なサーバの列挙
ArmClientEnumGroupFromLogicalService()	論理サービス名に対応するフェイルオーバーグループの列挙
ArmClientEnumVirtualComputer()	仮想コンピュータ名の列挙
ArmClientEnumVirtualIP()	仮想 IP アドレスの列挙
情報取得	
ArmClientGetClusterFromClusterServer()	サーバが所属するクラスタ名の取得
ArmClientGetClusterFromGroup()	フェイルオーバーグループが所属するクラスタ名の取得
ArmClientGetClusterName()	クラスタ名の取得
ArmClientGetClusterServerStatus()	サーバ動作状態の取得
ArmClientGetComputerName()	LAN Manager コンピュータ名の取得
ArmClientGetGroupFromVirtualComputer()	仮想コンピュータ名が所属しているフェイルオーバーグループ名の取得
ArmClientGetGroupFromVirtualIP()	仮想 IP アドレスが所属しているフェイルオーバーグループ名の取得
ArmClientGetGroupStatusEx()	フェイルオーバーグループの詳細な動作状態の取得
ArmClientGetHeartbeatTimeout()	ハートビートタイムアウト時間の取得
ArmClientGetIPAddr()	実 IP アドレスの取得
ArmClientGetIPAddrFromVirtualIP()	仮想 IP アドレスからの実 IP アドレスの取得
ArmClientGetMasterServer()	クラスタのマスタサーバ名の取得
ArmClientGetServerName()	ローカルサーバ名の取得
グループの監視	
ArmClientStartWatchFailoverEx()	フェイルオーバーグループの監視開始
ArmClientCancelWatchFailoverEx()	フェイルオーバーグループの監視中止
その他	
ArmClientCheckClusterServer()	クラスタサーバの確認
ArmClientGetVersionInfo()	CLUSTERPRO クライアント API のバージョン情報取得

7.1.3 API 詳細

7.1.3.1 ArmClientInitializeApiEx — CLUSTERPRO クライアント API の初期化

概要

CLUSTERPROクライアントAPIの初期化、資源の確保を行い、CLUSTERPROクライアントAPIを使用するためのハンドルを返却します。

形式

```
ArmClientInitializeApiEx(  
    ) As Long
```

パラメータ

なし。

戻り値

ARMCLAPI_ERROR_HANDLE	関数が異常終了しました(初期化失敗、または、ハンドルテーブルのオーバーフロー)。
上記以外	CLUSTERPROクライアントAPIハンドル

拡張エラー情報 (LastDLLError プロパティで取得)

ARMCLAPI_ERR_CANNOT_LOAD_DLL	CLUSTERPRO クライアント API DLL をロードできません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_NOT_ENOUGH_MEMORY	メモリのアロケートに失敗しました。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません

説明

本関数は、CLUSTERPROクライアントAPIの初期化処理を行い、必要な資源を確保します。CLUSTERPROクライアントAPIを使用する場合には、最初に本関数を実行して、CLUSTERPROクライアントAPIの初期化を行ってください。戻り値として、CLUSTERPROクライアントAPIハンドルを返却します。CLUSTERPROクライアントAPIハンドルは、フェイルオーバーグループの監視や終了処理に必要です。

単一のプロセスから、独立に複数のArmClientInitializeApiExを実行する事が可能です。

参照

ArmClientTerminateApiEx

7.1.3.2 ArmClientTerminateApiEx — CLUSTERPRO クライアント API の終了

概要

CLUSTERPROクライアントAPIの終了処理を行います。

形式

```
ArmClientTerminateApiEx(  
    ByVal      dwHandle As      Long  
) As          Long
```

パラメータ

[i] dwHandle	ArmClientInitializeApiExで返却したCLUSTERPROクライアントAPIハンドルを指定します。
--------------	---

戻り値

正常系：ARMTRUE
異常系：ARMFALSE

拡張エラー情報 (LastDLLError プロパティで取得)

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアント API が初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_INVALID_HANDLE	指定されたハンドルが不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

CLUSTERPROクライアントAPIの終了処理を行います。ArmClientInitializeApiEx関数を実行してCLUSTERPROクライアントAPIの初期化を行ったアプリケーションは、CLUSTERPROクライアントAPIの機能を使用する必要が無くなった時に、CLUSTERPROクライアントAPIハンドルを指定して本関数を実行しなければいけません。

参照

ArmClientInitializeApiEx

7.1.3.3 ArmClientGetIPAddr — 実 IP アドレスの取得

概要

接続するサーバの実IPアドレスを取得します。

形式

```
ArmClientGetIPAddr(  
    ByVal    lpszClusterName As String,  
    ByVal    lpszGroupName As String,  
    ByRef    lpdwIPAddr As Long  
) As Long
```

パラメータ

[i] lpszClusterName	クラスタ名を指定する、NULL で終了する文字列へのポインタです。 ローカルサーバのクラスタを指定する場合は、定数 ARMCLAPI_LOCAL_CLUSTER を指定することができます。この場合、ローカルサーバがクラスタサーバでなければエラー (ARMCLAPI_ERR_CLUSTER_NOT_FOUND) が返却されます。
[i] lpszGroupName	フェイルオーバーグループ名を指定する、NULL で終了する文字列へのポインタです。
[o] lpdwIPAddr	IP アドレスを受け取る変数へのポインタです。 IP アドレスはネットワークバイトオーダーで格納されます。

戻り値

正常系：ARMTRUE 関数が正常終了しました。
異常系：ARMFALSE 関数が異常終了しました。

拡張エラー情報 (LastDLLError プロパティで取得)

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアント API が初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	指定クラスタ名が存在しません。
ARMCLAPI_ERR_GROUP_NOT_FOUND	指定フェイルオーバーグループ名が存在しません。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、クラスタ名、フェイルオーバーグループ名から、現在動作しているサーバの実IPアドレスを問い合わせます。TCP/IPでクラスタに接続する場合には、仮想IPアドレスを指定して接続する方法と、本APIを用いて実IPアドレスを求めて接続する方法の、2通りの方法を使用することができます。

7.1.3.4 ArmClientGetComputerName — コンピュータ名を取得

概要

接続するサーバのLANManagerコンピュータ名を取得します。

形式

```
ArmClientGetComputerName(  
    ByVal    lpszClusterName As String,  
    ByVal    lpszGroupName  As String,  
    ByVal    lpszComputerName As String,  
    ByRef    lpcchComputerName As Long  
) As Long
```

パラメータ

[i] lpszClusterName	クラスタ名を指定する、NULL で終了する文字列へのポインタです。 ローカルサーバのクラスタを指定する場合は、定数 ARMCLAPI_LOCAL_CLUSTER を指定することができます。 この場合、ローカルサーバがクラスタサーバでなければエラー (ARMCLAPI_ERR_CLUSTER_NOT_FOUND) が返却されます。
[i] lpszGroupName	フェイルオーバーグループ名を指定する、NULL で終了する文字列へのポインタです。
[o] lpszComputerName	現在動作しているサーバのコンピュータ名（終端の NULL 文字を含む）を受け取るバッファへのポインタです。
[io] lpcchComputerName	lpszComputerName で指定されるバッファのサイズを指定する変数へのポインタです。関数のリターン時には、バッファに格納された文字数（終端の NULL 文字を含まない）がセットされます。指定バッファサイズが小さすぎた場合には、必要な文字数（終端の NULL 文字を含む）がセットされ、拡張エラー情報で ARMCLAPI_ERR_BUFFER_TOO_SMALL が返却されます。

戻り値

正常系：ARMTRUE 関数が正常終了しました。
異常系：ARMFALSE 関数が異常終了しました。

拡張エラー情報 (LastDLLError プロパティで取得)

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアント API が初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	指定クラスタ名が存在しません。

説明

本関数は、クラスタ名、フェイルオーバーグループ名から、現在動作しているサーバのLANManagerコンピュータ名を問い合わせます。

7.1.3.5 ArmClientGetIPAddrFromVirtualIP — 仮想 IP アドレスからの実 IP アドレスの取得

概要

接続するサーバの実IPアドレスを取得します。

形式

```
ArmClientGetIPAddrFromVirtualIP(  
    ByVal    dwVirtualIPAddr As Long,  
    ByRef    lpdwIPAddr      As Long  
) As Long
```

パラメータ

[i]	dwVirtualIPAddr	仮想IPアドレスです。ネットワークバイトオーダーで指定します。
[o]	lpdwIPAddr	IPアドレスを受け取る変数へのポインタです。 IPアドレスはネットワークバイトオーダーで格納されます。

戻り値

正常系：ARMTRUE 関数が正常終了しました。
異常系：ARMFALSE 関数が異常終了しました。

拡張エラー情報 (LastDLLError プロパティで取得)

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアント API が初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_VIRTUALIP_NOT_FOUND	指定された仮想 IP アドレスが見つかりません。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、仮想IPアドレスから、現在動作しているサーバの実IPアドレスを問い合わせます。

7.1.3.6 ArmClientStartWatchFailoverEx – フェイルオーバーグループの監視の開始

概要

フェイルオーバーの発生/完了などの、フェイルオーバーグループの状態変化の監視を開始します。グループの状態変化を検出した時に、登録されたコールバック関数を呼び出します。

形式

```
ArmClientStartWatchFailoverEx(  
    ByVal    dwHandle As Long,  
    ByVal    lpszClusterName As String,  
    ByVal    lpszGroupName As String,  
    ByVal    WATCHFAILOVERPROC As Long,  
    ByRef    lpdwWatchId As Long,  
    ByRef    lpdwStatusEx As Long  
) As Long
```

パラメータ

[i]	dwHandle	CLUSTERPRO クライアント API ハンドル
[i]	lpszClusterName	監視するフェイルオーバーグループが属するクラスタ名を指定する、NULL で終了する文字列へのポインタです。 ローカルサーバのクラスタを指定する場合は、定数 ARMCLAPI_LOCAL_CLUSTER を指定することができます。この場合、ローカルサーバがクラスタサーバでなければエラー (ARMCLAPI_ERR_CLUSTER_NOT_FOUND) が返却されます。
[i]	lpszGroupName	監視するフェイルオーバーグループ名を指定する、NULL で終了する文字列へのポインタです。
[i]	WatchFailoverProc	フェイルオーバーの発生/完了を検出した時に呼び出される、アプリケーション定義のコールバック関数のアドレスです (WatchFailoverProc 関数参照)
[o]	lpdwWatchId	監視 ID を受け取る変数へのポインタです。
[o]	lpdwStatusEx	監視を開始した時のフェイルオーバーグループの状態です。ArmClientGetServerStatusEx で返却するものと同じです。

戻り値

正常系：ARMTRUE 関数が正常終了しました。
異常系：ARMFALSE 関数が異常終了しました。

拡張エラー情報 (LastDLLError プロパティで取得)

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアント API が初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	指定クラスタ名が存在しません。
ARMCLAPI_ERR_GROUP_NOT_FOUND	指定フェイルオーバーグループ名が存在しません。
ARMCLAPI_ERR_INVALID_HANDLE	ハンドルが不正です。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、フェイルオーバーグループの状態変化の監視を開始します。監視する状態変化は、以下の通りです。

- フェイルオーバーの開始
- フェイルオーバーの完了
- フェイルオーバーグループ移動の開始
- フェイルオーバーグループ移動の完了
- フェイルオーバーグループの起動
- フェイルオーバーグループの停止
- サーバクライアント間のネットワークの切り替え開始
- サーバクライアント間のネットワークの切り替え完了
- サーバクライアント間のネットワークの異常検出
- サーバクライアント間のネットワークの回復検出

本関数は、アプリケーション定義のコールバック関数WatchFailoverProcを登録した後、直ちに本関数を呼び出したアプリケーションプロセスに制御を戻します。コールバック関数は、CLUSTERPROクライアントによりフェイルオーバーグループの状態の変化が検出された時に、非同期に呼び出されます。

参照

ArmClientCancelWatchFailover, WatchFailoverProc, ArmClientGetServerStatusEx

7.1.3.7 ArmClientCancelWatchFailoverEx – フェイルオーバーグループ監視の中止

概要

ArmClientStartWatchFailoverEx関数によるフェイルオーバーグループの発生/完了の監視を中止します。

形式

```
ArmClientCancelWatchFailoverEx(  
    ByVal    dwWatchId    As    Long  
) As    Long
```

パラメータ

[i] dwWatchId ArmClientStartWatchFailover関数で返却された監視IDを指定します。

戻り値

正常系：ARMTRUE 関数が正常終了しました。
異常系：ARMFALSE 関数が異常終了しました。

拡張エラー情報 (LastDLLError プロパティで取得)

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアント API が初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_INVALID_ID	指定した監視 ID が不正です。

説明

本関数は、dwWatchIdで識別されるArmClientStartWatchFailover関数によるフェイルオーバーグループの監視を中止します。

参照

ArmClientStartWatchFailover

7.1.3.8 ArmClientEnumCluster — 登録されたクラスタの列挙

概要

クライアントに登録されたクラスタを列挙します。

形式

```
ArmClientEnumCluster(  
    ByVal EnumClusterProc As Long,  
    ByVal lParam As Long  
) As Long
```

パラメータ

[i]	EnumClusterProc	クラスタ名を受け取る、アプリケーション定義のコールバック関数のアドレスです (EnumClusterProc関数参照)
[i]	lParam	コールバック関数に渡される、アプリケーション定義の32ビット値です。

戻り値

正常系：ARMTRUE 関数が正常終了しました。
異常系：ARMFALSE 関数が異常終了しました。

拡張エラー情報 (LastDLLError プロパティで取得)

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアント API が初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。
ARMCLAPI_ERR_CANCELLED	コールバック関数から列挙の中止が返却されました。

説明

本関数は、クライアントに登録されたクラスタを列挙します。本関数は、アプリケーション定義のコールバック関数EnumClusterProcを順次呼び出し、各クラスタ名を渡します。本関数は、クラスタをすべて列挙するか、またはコールバック関数がARMFALSEを返却するまで、列挙を続けます。

参照

EnumClusterProc

7.1.3.9 ArmClientEnumServer — サーバの列挙

概要

クラスタに所属するサーバを列挙します。

形式

```
ArmClientEnumServer(  
    ByVal lpszClusterName As String,  
    ByVal EnumServerProc As Long,  
    ByVal lParam As Long  
) As Long
```

パラメータ

[i]	lpszClusterName	サーバを列挙するクラスタ名を指定します。
[i]	EnumServerProc	クラスタに所属するサーバのコンピュータ名を受け取る、アプリケーション定義のコールバック関数のアドレスです。(EnumServerProc 関数参照)
[i]	lParam	コールバック関数 EnumServerProc に渡される、アプリケーション定義の 32 ビット値です。

戻り値

正常系：ARMTRUE 関数が正常終了しました。
異常系：ARMFALSE 関数が異常終了しました。

拡張エラー情報 (LastDLLError プロパティで取得)

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO API が初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO モニタに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO モニタとの通信でエラーが発生しました。
ARMCLAPI_ERR_NOT_ENOUGH_MEMORY	メモリのアロケートに失敗しました。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_CLUSTER_MEMBER	ローカルサーバがクラスタメンバではありません。
ARMCLAPI_ERR_OFF_CLUSTER	ローカルサーバがクラスタとして動作していません。
ARMCLAPI_ERR_CANCELLED	コールバック関数から列挙の中止が返却されました。

説明

本関数は、ローカルサーバの所属するクラスタを構成しているサーバを列挙します。本関数は、アプリケーション定義のコールバック関数EnumServerProcを順次呼び出し、各サーバのコンピュータ名を渡します。本関数は、クラスタを構成しているサーバをすべて列挙するか、またはコールバック関数がARMFALSEを返却するまで、列挙を続けます。

ローカルサーバがクラスタとして動作していない場合、サーバを列挙することはできません。

- ローカルサーバがクラスタに登録されていない場合、本関数は ARMFALSE を返却します。このとき拡張エラー情報で ARMCLAPI_ERR_NOT_CLUSTER_MEMBER が返却されます。
- ダウン後再起動状態等でローカルサーバがクラスタから一時的に切り離されている場合、本関数は ARMFALSE を返却します。このとき拡張エラー情報で ARMCLAPI_ERR_OFF_CLUSTER が返却されます。

参照

EnumServerProc — サーバ列挙用コールバック関数

7.1.3.10 ArmClientEnumGroup — フェイルオーバーグループの列挙

概要

指定されたクラスタのフェイルオーバーグループを列挙します。

形式

```
ArmClientEnumGroup(  
    ByVal    lpszClusterName As String,  
    ByVal    EnumGroupProc As Long,  
    ByVal    lParam As Long  
) As Long
```

パラメータ

[I]	lpszClusterName	クラスタ名を指定する、NULL で終了する文字列へのポインタです。 ローカルサーバのクラスタを指定する場合は、定数 ARMCLAPI_LOCAL_CLUSTER を指定することができます。この場合、ローカルサーバがクラスタサーバでなければエラー (ARMCLAPI_ERR_CLUSTER_NOT_FOUND) が返却されます。
[I]	EnumGroupProc	サーバのコンピュータ名を受け取る、アプリケーション定義のコールバック関数のアドレスです。(EnumGroupProc 関数参照)
[I]	lParam	コールバック関数に渡される、アプリケーション定義の 32 ビット値です。

戻り値

正常系：ARMTRUE 関数が正常終了しました。
異常系：ARMFALSE 関数が異常終了しました。

拡張エラー情報 (LastDLLError プロパティで取得)

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアント API が初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	指定クラスタ名が存在しません。
ARMCLAPI_ERR_CANCELLED	コールバック関数から列挙の中止が返却されました。

説明

本関数は、指定したフェイルオーバーグループが移動可能なサーバを列挙します。本関数は、アプリケーション定義のコールバック関数EnumGroupProcを順次呼び出し、各グループ名を渡します。本関数は、サーバをすべて列挙するか、またはコールバック関数がARMFALSEを返却するまで、列挙を続けます。
列挙の順は、フェイルオーバーグループのフェイルオーバーポリシーの優先度が高いサーバから順に列挙されます。

参照

EnumGroupProc

7.1.3.11 ArmClientEnumServerFromGroup — フェイルオーバーグループが移動可能なサーバの列挙

概要

フェイルオーバーグループが移動可能サーバを列挙します。

形式

```
ArmClientEnumServerFromGroup(  
    ByVal    lpszClusterName As String,  
    ByVal    lpszGroupName As String,  
    ByVal    EnumServerFromGroupProc As Long  
    ByVal    lParam As Long  
) As Long
```

パラメータ

[i] lpszClusterName	クラスタ名を指定する、NULL で終了する文字列へのポインタです。 ローカルサーバのクラスタを指定する場合は、定数 ARMCLAPI_LOCAL_CLUSTER を指定することができます。この場合、ローカルサーバがクラスタサーバでなければエラー (ARMCLAPI_ERR_CLUSTER_NOT_FOUND) が返却されます。
[i] lpszGroupName	フェイルオーバーグループ名を指定する、NULL で終了する文字列へのポインタです。
[i] EnumServerFromGroupProc	サーバのコンピュータ名を受け取る、アプリケーション定義のコールバック関数のアドレスです。 (EnumServerFromGroupProc 関数参照)
[i] lParam	コールバック関数に渡される、アプリケーション定義の 32 ビット値です。

戻り値

正常系：ARMTRUE 関数が正常終了しました。

異常系：ARMFALSE 関数が異常終了しました。

拡張エラー情報 (LastDLLError プロパティで取得)

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアント API が初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	指定クラスタ名が存在しません。
ARMCLAPI_ERR_CANCELLED	コールバック関数から列挙の中止が返却されました。

説明

本関数は、指定したフェイルオーバーグループが移動可能なサーバを列挙します。本関数は、アプリケーション定義のコールバック関数EnumServerFromGroupProcを順次呼び出し、各サーバのコンピュータ名を渡します。本関数は、サーバをすべて列挙するか、またはコールバック関数がARMFALSEを返却するまで、列挙を続けます。列挙の順は、フェイルオーバーグループのフェイルオーバーポリシーの優先度が高いサーバから順に列挙されます。

参照

EnumServerFromGroupProc

7.1.3.12 ArmClientGetVersionInfo — CLUSTERPRO クライアント API のバージョン情報の取得

概要

CLUSTERPROクライアントAPIのバージョン情報を取得します。

形式

```
ArmClientGetVersionInfo(  
    ByRef    dwMajor As Long,  
    ByRef    dwMinor As Long,  
    ByRef    dwBuild As Long  
) As Long
```

パラメータ

[o]	dwMajor	メジャーバージョン番号。CLUSTERPRO Ver5.0では仕様番号上位4を返却します。
[o]	dwMinor	マイナーバージョン番号。CLUSTERPRO Ver5.0では仕様番号下位0を返却します。
[o]	dwBuild	ビルド番号。未使用。将来の拡張のために予約しています。

戻り値

正常系：ARMTRUE 関数が正常終了しました。
異常系：ARMFALSE 関数が異常終了しました。

拡張エラー情報 (LastDLLError プロパティで取得)

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアント API が初期化されていません。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

CLUSTERPROクライアントAPIのバージョン情報を取得します。アプリケーションは、本関数を実行してCLUSTERPROクライアントAPIのバージョンを取得することができます。

ArmClientGetVersionInfoにて取得できるバージョン情報は、CLUSTERPROクライアントAPIのバージョン情報です。CLUSTERPRO本体やarmcl2vb.dllのバージョン情報ではありません。

7.1.3.13 ArmClientGetGroupStatusEx — フェイルオーバーグループの詳細な動作状態の取得

概要

フェイルオーバーグループの動作状態を取得します。ArmClientGetGroupStatusより詳細な情報が得られます。

形式

```
ArmClientGetGroupStatusEx(  
    ByVal    lpszClusterName As String,  
    ByVal    lpszGroupname As String,  
    ByRef    lpdwStatusEx As Long  
) As Long
```

パラメータ

[i]	lpszClusterName	クラスタ名を指定する、NULL で終了する文字列へのポインタです。 ローカルサーバのクラスタを指定する場合は、定数 ARMCLAPI_LOCAL_CLUSTER を指定することができます。 この場合、ローカルサーバがクラスタサーバでなければエラー (ARMCLAPI_ERR_CLUSTER_NOT_FOUND) が返却されます。								
[i]	lpszGroupName	フェイルオーバーグループ名を指定する、NULL で終了する文字列へのポ インタです。								
[o]	lpdwStatusEx	動作状態を受け取る変数へのポインタです。 以下の値が返却されます。 <table><tr><td>ARMCLAPI_GROUP_STATUS_RUNNING</td><td>正常動作中</td></tr><tr><td>ARMCLAPI_GROUP_STATUS_MISS</td><td>資源獲得に失敗</td></tr><tr><td>ARMCLAPI_GROUP_STATUS_STOPPED</td><td>停止中</td></tr><tr><td>ARMCLAPI_GROUP_STATUS_FAILOVER</td><td>フェイルオーバー処理中</td></tr></table>	ARMCLAPI_GROUP_STATUS_RUNNING	正常動作中	ARMCLAPI_GROUP_STATUS_MISS	資源獲得に失敗	ARMCLAPI_GROUP_STATUS_STOPPED	停止中	ARMCLAPI_GROUP_STATUS_FAILOVER	フェイルオーバー処理中
ARMCLAPI_GROUP_STATUS_RUNNING	正常動作中									
ARMCLAPI_GROUP_STATUS_MISS	資源獲得に失敗									
ARMCLAPI_GROUP_STATUS_STOPPED	停止中									
ARMCLAPI_GROUP_STATUS_FAILOVER	フェイルオーバー処理中									

戻り値

正常系：ARMTRUE 関数が正常終了しました。
異常系：ARMFALSE 関数が異常終了しました。

拡張エラー情報 (LastDLLError プロパティで取得)

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアント API が初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しまし た。
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	指定クラスタ名が存在しません。
ARMCLAPI_ERR_GROUP_NOT_FOUND	指定フェイルオーバーグループ名が存在しません。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、クラスタ名、本来動作しているサーバのコンピュータ名から、サーバの状態を問い合わせます。正常動作中とは、フェイルオーバーグループが、クライアントから接続可能なサーバのいずれかで正常に動作している状態です。どのサーバで動作しているかは、ArmClientGetComputerName等で問い合わせてください。資源獲得に失敗とは、動作中であるが、切替ディスクの接続失敗など、クラスタ資源の獲得に失敗している状態です。

停止中とは、フェイルオーバーグループが、クライアントから接続可能なサーバのいずれでも動作していない状態です。フェイルオーバー処理実行中を除きます。以下の場合には停止中となります。

- ・フェイルオーバーグループが移動可能なサーバが全て正常動作していない場合（サーバが停止あるいはダウン状態、CLUSTERPROが動作していない状態、クラスタから切り離された状態）
- ・フェイルオーバーグループが停止された状態
- ・フェイルオーバーグループが動作しているサーバとクライアントとの間のネットワーク経路の異常

フェイルオーバー処理中とは、フェイルオーバーグループが動作していたサーバのダウンやフェイルオーバーグループの異常を検出してから、もう一方のサーバでの引き継ぎ処理が完了するまでの間です。また、CLUSTERPRO マネージャからフェイルオーバーグループの移動を実行した場合は、移動を開始してから移動が完了するまでの間はフェイルオーバー処理中になります。

7.1.3.14 ArmClientGetClusterServerStatus — サーバの動作状態の取得

概要

サーバの動作状態を取得します。

形式

```
ArmClientGetClusterServerStatus(  
    ByVal    lpszClusterName As String,  
    ByVal    lpszServerName As String,  
    ByRef    lpdwStatusEx As Long  
) As Long
```

パラメータ

[I]	lpszClusterName	クラスタ名を指定する、NULLで終了する文字列へのポインタです。 ローカルサーバのクラスタを指定する場合は、定数ARMCLAPI_LOCAL_CLUSTERを指定することができます。この場合、ローカルサーバがクラスタサーバでなければエラー(ARMCLAPI_ERR_CLUSTER_NOT_FOUND)が返却されます。
[I]	lpszServerName	サーバ名を指定する、NULLで終了する文字列へのポインタです。
[o]	lpdwStatusEx	動作状態を受け取る変数へのポインタです。 以下の値が返却されます。 ARMCLAPI_SERVER_STATUS_SINGLE_SERVER クラスタに未登録のシングルサーバとして動作しています。 ARMCLAPI_SERVER_STATUS_CLUSTER クラスタとして正常に動作しています。 ARMCLAPI_SERVER_STATUS_OFF_CLUSTER クラスタには登録されているが、クラスタから切り離されて動作しています。 ARMCLAPI_SERVER_STATUS_STOPPED 停止状態です。 ARMCLAPI_SERVER_STATUS_DOWN ダウン状態です。 ARMCLAPI_SERVER_STATUS_RECOVERING フェイルオーバー処理中です。 ARMCLAPI_SERVER_STATUS_FAILBACK 上記の状態に、次の状態がbit論理和で組み合わせられます。 ARMCLAPI_SERVER_STATUS_BACKUP 他サーバからフェイルオーバーした資源があります。 ARMCLAPI_SERVER_STATUS_FAIL 他サーバへフェイルオーバーした資源があります。 ARMCLAPI_SERVER_STATUS_FAILMISS 接続に失敗した資源があります。 例えばクラスタとして正常に動作しているかどうかを調べるためには、lpdwStatusEx & ARMCLAPI_SERVER_STATUS_CLUSTER の値が 0 かどうかで判定することができます。

戻り値

正常系：ARMTRUE 関数が正常終了しました。
異常系：ARMFALSE 関数が異常終了しました。

拡張エラー情報 (LastDLLError プロパティで取得)

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアント API が初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	指定クラスタ名が存在しません。
ARMCLAPI_ERR_SERVER_NOT_FOUND	指定サーバが存在しません。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、サーバの状態を問い合わせます。

7.1.3.15 ArmClientGetClusterFromGroup — フェイルオーバーグループが所属するクラスタ名の取得

概要

フェイルオーバーグループ名から、そのサーバが所属するクラスタのクラスタ名を取得します。

形式

```
ArmClientGetClusterFromGroup(  
    ByVal    lpszGroupName As String,  
    ByVal    lpszClusterName As String,  
    ByRef    lpcchClusterName As Long  
) As Long
```

パラメータ

[i]	lpszGroupName	フェイルオーバーグループ名を指定する、NULLで終了する文字列へのポインタです。
[o]	lpszClusterName	クラスタ名（終端のNULL文字を含む）を受け取るバッファへのポインタです。
[io]	lpcchClusterName	lpszClusterNameで指定されるバッファのサイズを指定する変数へのポインタです。関数のリターン時には、バッファに格納された文字数（終端のNULL文字を含まない）がセットされます。指定バッファサイズが小さすぎた場合には、必要な文字数（終端のNULL文字を含む）がセットされ、拡張エラー情報でARMCLAPI_ERR_BUFFER_TOO_SMALLが返却されます。

戻り値

正常系：ARMTRUE 関数が正常終了しました。
異常系：ARMFALSE 関数が異常終了しました。

拡張エラー情報（LastDLLError プロパティで取得）

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアント API が初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_GROUP_NOT_FOUND	指定されたフェイルオーバーグループ名が見つかりません。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、フェイルオーバーグループ名から、そのサーバが属しているクラスタ名を問い合わせます。ただし、複数のクラスタに同一の名前のフェイルオーバーグループ名が存在する場合は、どのクラスタ名を返却するかは保証しません。
クライアント設定されているクラスタが複数存在し、同一名のグループが二つ以上のクラスタに存在する場合、最初に見つかったグループの登録されているグループの所属するクラスタ名を返却します（クライアント設定に登録されている順に検索します）。

7.1.3.16 ArmClientGetClusterFromClusterServer — サーバが所属するクラスタ名の取得

概要

サーバ名から、そのサーバが所属するクラスタのクラスタ名を取得します。

形式

```
ArmClientGetClusterFromClusterServer(  
    ByVal    lpszServerName As String,  
    ByVal    lpszClusterName As String,  
    ByRef    lpccchClusterName As Long  
) As Long
```

パラメータ

[i]	lpszServerName	サーバ名を指定する、NULLで終了する文字列へのポインタです。
[o]	lpszClusterName	現在動作しているサーバのコンピュータ名（終端のNULL文字を含む）を受け取るバッファへのポインタです。
[io]	lpccchClusterName	lpszClusterNameで指定されるバッファのサイズを指定する変数へのポインタです。関数のリターン時には、バッファに格納された文字数（終端のNULL文字を含まない）がセットされます。指定バッファサイズが小さすぎた場合には、必要な文字数（終端のNULL文字を含む）がセットされ、拡張エラー情報でARMCLAPI_ERR_BUFFER_TOO_SMALLが返却されます。

戻り値

正常系：ARMTRUE 関数が正常終了しました。
異常系：ARMFALSE 関数が異常終了しました。

拡張エラー情報（LastDLLError プロパティで取得）

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアント API が初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_SERVER_NOT_FOUND	指定されたサーバ名が見つかりません。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、サーバ名から、そのサーバが属しているクラスタ名を問い合わせます。ただし、複数のクラスタに同一の名前のサーバ名が存在する場合は、どのクラスタ名を返却するかは保証しません。

7.1.3.17 ArmClientGetGroupFromVirtualIP – 仮想 IP アドレスが所属しているフェイルオーバーグループ名の取得

概要

仮想IPアドレスが所属しているクラスタ名及びフェイルオーバーグループ名を取得します。

形式

```
ArmClientGetGroupFromVirtualIP(  
    ByVal dwVirtualIP As Long,  
    ByVal lpszClusterName As String,  
    ByRef lpcchClusterName As Long,  
    ByVal lpszGroupName As String,  
    ByRef lpcchGroupName As Long  
) As Long
```

パラメータ

[i]	dwVirtualIP	仮想IPアドレス。ネットワークバイトオーダーで指定します。
[o]	lpszClusterName	クラスタ名（終端のNULL文字を含む）を受け取るバッファへのポインタ。
[io]	lpcchClusterName	lpszClusterNameで指定されるバッファのサイズを指定する変数へのポインタ。関数のリターン時にはバッファに格納された文字数（終端のNULL文字を含まない）がセットされます。指定バッファサイズが小さすぎた場合には、必要な文字数（終端のNULL文字を含む）がセットされ、拡張エラー情報でARMCLAPI_ERR_BUFFER_TOO_SMALLが返却されます。
[o]	lpszGroupName	仮想IPアドレスが所属しているフェイルオーバーグループ名（終端のNULL文字を含む）を受け取るバッファへのポインタ。
[io]	lpcchGroupName	lpszGroupNameで指定されるバッファのサイズを指定する変数へのポインタ。関数のリターン時にはバッファに格納された文字数（終端のNULL文字を含まない）がセットされます。指定バッファサイズが小さすぎた場合には、必要な文字数（終端のNULL文字を含む）がセットされ、拡張エラー情報でARMCLAPI_ERR_BUFFER_TOO_SMALLが返却されます。

戻り値

正常系：ARMTRUE 関数が正常終了しました。

異常系：ARMFALSE 関数が異常終了しました。

拡張エラー情報（LastDLLError プロパティで取得）

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアント API の初期化処理が行われていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_BUFFER_TOO_SMALL	文字列を格納するバッファのサイズが小さすぎます。
ARMCLAPI_ERR_VIRTUALIP_NOT_FOUND	指定仮想 IP アドレスが存在しません。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は仮想IPアドレスを指定して、それに対応するクラスタ名及びフェイルオーバーグループ名を取得します。クラスタ名をバッファlpszClusterNameに格納します。フェイルオーバーグループ名をlpszGroupNameに格納します。仮想IPアドレスを監視する場合には、まずこの関数でクラスタ名及びフェイルオーバーグループ名を得ます。そして、それを、フェイルオーバーグループの状態監視開始の入力パラメータとして使用することで、監視が開始されます。

7.1.3.18 ArmClientEnumVirtualIP — 仮想 IP アドレスの列挙

概要

グループ、またはクラスタに所属する仮想IPアドレスを列挙します。

形式

```
ArmClientEnumVirtualIP(  
    ByVal    lpszClusterName As String,  
    ByVal    lpszGroupName As String,  
    ByVal    EnumVirtualIPProc As Long,  
    ByVal    lParam As Long  
) As Long
```

パラメータ

[i]	lpszClusterName	クラスタ名を指定する、NULLで終了する文字列へのポインタです。 NULLあるいは空文字列を指定した場合は、登録された全てのクラスタが対象となります。 ローカルサーバのクラスタを指定する場合は、定数ARMCLAPI_LOCAL_CLUSTERを指定することができます。この場合、ローカルサーバがクラスタサーバでなければエラー(ARMCLAPI_ERR_CLUSTER_NOT_FOUND)が返却されます。
[i]	lpszGroupName	仮想IPアドレスが所属するフェイルオーバーグループ名を指定する、NULLで終了する文字列へのポインタです。 NULLあるいは空文字列を指定すると、クラスタに属する全てのグループが列挙の対象となります。
[i]	EnumVirtualIPProc	フェイルオーバーグループ名を受け取る、アプリケーション定義のコールバック関数のアドレスです。(EnumVirtualIPProc関数参照)
[i]	lParam	コールバック関数に渡される、アプリケーション定義の32ビット値です。

戻り値

正常系：ARMTRUE	関数が正常終了しました。
異常系：ARMFALSE	関数が異常終了しました。
拡張エラー情報 (LastDLLError プロパティで取得)	
ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアント API が初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	指定クラスタ名が存在しません。
ARMCLAPI_ERR_GROUP_NOT_FOUND	フェイルオーバーグループ名が存在しません。

説明

本関数は、指定したクラスタ、フェイルオーバーグループに属する仮想IPアドレスを列挙します。本関数は、アプリケーション定義のコールバック関数EnumVirtualIPProcを順次呼び出し、各仮想IPアドレスを渡します。本関数は、仮想IPアドレスをすべて列挙するか、またはコールバック関数がFALSEを返却するまで、列挙を続けます。

参照

EnumVirtualIPProc

7.1.3.19 ArmClientGetGroupFromVirtualComputer — 仮想コンピュータ名が所属しているフェイルオーバーグループ名の取得

概要

仮想コンピュータ名が所属しているクラスタ名及びフェイルオーバーグループ名を取得します。

形式

```
ArmClientGetGroupFromVirtualComputer(  
    ByVal    lpszVirtualComputerName As String,  
    ByVal    lpszClusterName As String,  
    ByRef    lpcchClusterName As Long,  
    ByVal    lpszGroupName As String,  
    ByRef    lpcchGroupName As Long  
) As Long
```

パラメータ

[i]	lpszVirtualComputerName	仮想コンピュータ名。ネットワークバイトオーダで指定します。
[o]	lpszClusterName	クラスタ名（終端のNULL文字を含む）を受け取るバッファへのポインタです。
[io]	lpcchClusterName	lpszClusterNameで指定されるバッファのサイズを指定する変数へのポインタ。関数のリターン時にはバッファに格納された文字数（終端のNULL文字を含まない）がセットされます。 指定バッファサイズが小さすぎた場合には、必要な文字数（終端のNULL文字を含む）がセットされ、拡張エラー情報で ARMERROR_BUFFER_TOO_SMALL が返却されます。
[o]	lpszGroupName	仮想IPアドレスが所属しているフェイルオーバーグループ名（終端のNULL文字を含む）を受け取るバッファへのポインタ。
[io]	lpcchGroupName	lpszGroupNameで指定されるバッファのサイズを指定する変数へのポインタです。関数のリターン時にはバッファに格納された文字数（終端のNULL文字を含まない）がセットされます。 指定バッファサイズが小さすぎた場合には、必要な文字数（終端のNULL文字を含む）がセットされ、拡張エラー情報で ARMERROR_BUFFER_TOO_SMALL が返却されます。

戻り値

正常系：ARMTRUE 関数が正常終了しました。
異常系：ARMFALSE 関数が異常終了しました。

拡張エラー情報（LastDLLError プロパティで取得）

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアント API の初期化処理が行われていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_BUFFER_TOO_SMALL	文字列を格納するバッファのサイズが小さすぎます。
ARMCLAPI_ERR_VCOM_NOT_FOUND	指定された仮想コンピュータ名が存在しません。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は仮想コンピュータ名を指定して、それに対応するクラスタ名及びフェイルオーバーグループ名を取得します。クラスタ名をバッファlpszClusterNameに格納します。フェイルオーバーグループ名をlpszGroupNameに格納します。仮想コンピュータ名を監視する場合には、まずこの関数でクラスタ名及びフェイルオーバーグループ名を得ます。そして、それを、フェイルオーバーグループの状態監視開始の入力パラメータとして使用することで、監視が開始されます。

7.1.3.20 ArmClientEnumVirtualComputer — 仮想コンピュータ名の列挙

概要

クラスタに所属する仮想コンピュータ名を列挙します。

形式

```
ArmClientEnumVirtualComputer(  
    ByVal lpszClusterName As String,  
    ByVal lpszGroupName As String,  
    ByVal EnumVirtualComputerProc As Long,  
    ByVal lParam As Long  
) As Long
```

パラメータ

[i] lpszClusterName	クラスタ名を指定する、NULLで終了する文字列へのポインタです。 NULLあるいは空文字列を指定した場合は、登録された全てのクラスタが対象となります。 ローカルサーバのクラスタを指定する場合は、定数 ARMCLAPI_LOCAL_CLUSTER を指定することができます。 この場合、ローカルサーバがクラスタサーバでなければエラー (ARMCLAPI_ERR_CLUSTER_NOT_FOUND) が返却されます。
[i] lpszGroupName	仮想コンピュータ名が所属するフェイルオーバーグループ名を指定する、NULLで終了する文字列へのポインタです。 NULLあるいは空文字列を指定すると、クラスタに属する全てのグループが列挙の対象となります。
[i] EnumVirtualComputerProc	仮想コンピュータ名を受け取る、アプリケーション定義のコールバック関数のアドレスです。(EnumVirtualComputerProc関数参照)
[i] lParam	コールバック関数に渡される、アプリケーション定義の32ビット値です。

戻り値

正常系：ARMTRUE 関数が正常終了しました。
異常系：ARMFALSE 関数が異常終了しました。

拡張エラー情報 (LastDLLError プロパティで取得)

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアント API が初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	指定クラスタ名が存在しません。
ARMCLAPI_ERR_GROUP_NOT_FOUND	フェイルオーバーグループ名が存在しません。
ARMCLAPI_ERR_CANCELLED	コールバック関数から列挙の中止が返却されました。

説明

本関数は、指定したクラスタ、フェイルオーバーグループに属する仮想コンピュータ名を列挙します。本関数は、アプリケーション定義のコールバック関数 EnumVirtualComputerProc を順次呼び出し、各仮想コンピュータ名を渡します。本関数は、仮想コンピュータ名をすべて列挙するか、またはコールバック関数が ARMFALSE を返却するまで、列挙を続けます。

参照

EnumVirtualComputerProc

7.1.3.21 ArmClientEnumGroupFromLogicalService — 論理サービス名に対応するフェイルオーバーグループの列挙

概要

論理サービス名が登録されているフェイルオーバーグループ名を列挙します。

形式

```
ArmClientEnumGroupFromLogicalService(  
    ByVal    lpszClusterName As      String,  
    ByVal    lpszLogicalServiceName As  String,  
    ByVal    EnumGroupFromLogicalServiceProc As      Long,  
    ByVal    lParam As      Long  
) As      Long
```

パラメータ

[i] lpszClusterName	クラスタ名を指定する、NULLで終了する文字列へのポインタです。 NULLあるいは空文字列を指定した場合は、登録された全てのクラスタが対象となります。 ローカルサーバのクラスタを指定する場合は、定数 ARMCLAPI_LOCAL_CLUSTER を指定することができます。この場合、ローカルサーバがクラスタサーバでなければエラー (ARMCLAPI_ERR_CLUSTER_NOT_FOUND) が返却されます。
[i] lpszLogicalServiceName	論理サービス名を指定する、NULLで終了する文字列へのポインタです。
[i] EnumGroupFromLogicalServiceProc	フェイルオーバーグループ名を受け取る、アプリケーション定義のコールバック関数のアドレスです。 (EnumGroupFromLogicalServiceProc 関数参照)
[i] lParam	コールバック関数に渡される、アプリケーション定義の32ビット値です。

戻り値

正常系：ARMTRUE 関数が正常終了しました。
異常系：ARMFALSE 関数が異常終了しました。

拡張エラー情報 (LastDLLError プロパティで取得)

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアント API が初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	指定されたクラスタ名が存在しません。
ARMCLAPI_ERR_LOGICAL_SERVICE_NOT_FOUND	指定された論理サービス名が見つかりません。
ARMCLAPI_ERR_CANCELLED	コールバック関数から列挙の中止が返却されました。

説明

本関数は、論理サービス名から、それに対応するフェイルオーバーグループ名を列挙します。本関数は、アプリケーション定義のコールバック関数EnumGroupFromLogicalServiceProcを順次呼び出し、各サーバのコンピュータ名を渡します。本関数は、該当するフェイルオーバーグループをすべて列挙するか、またはコールバック関数がARMFALSEを返却するまで、列挙を続けます。

参照

EnumGroupFromLogicalServiceProc

7.1.3.22 ArmClientGetHeartbeatTimeout — ハートビートタイムアウト時間の取得

概要

CLUSTERPROに設定されたハートビートタイムアウト時間(秒)を取得します。

形式

```
ArmClientGetHeartbeatTimeout(  
    ByVal    lpszClusterName As String,  
    ByRef    lpdwTimeout As Long  
) As Long
```

パラメータ

[i] lpszClusterName	クラスタ名を指定する、NULLで終了する文字列へのポインタです。 ローカルサーバのクラスタを指定する場合は、定数 ARMCLAPI_LOCAL_CLUSTERを指定することができます。 この場合、ローカルサーバがクラスタサーバでなければエラー (ARMCLAPI_ERR_CLUSTER_NOT_FOUND)が返却されます。
[o] lpdwTimeout	指定されたクラスタに設定されたハートビートタイムアウトを受け取る 変数へのポインタです。ハートビートタイムアウトの単位は秒です。

戻り値

正常系：ARMTRUE 関数が正常終了しました。
異常系：ARMFALSE 関数が異常終了しました。

拡張エラー情報 (LastDLLError プロパティで取得)

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアント API が初期化されていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	指定されたクラスタ名が存在しません。

説明

本関数は、指定されたクラスタに設定されたハートビートタイムアウト時間を問い合わせます。
ハートビートタイムアウト時間は、CLUSTERPROが相手サーバからのハートビートの停止から、相手サーバのダウンを判定する時間です。本関数で得られる時間は、クライアントAPがサーバとの通信異常を検出した場合に、フェイルオーバー発生通知を待つ場合の時間の目安として使用することができます。

7.1.3.23 ArmClientGetMasterServer — マスタサーバ名の取得

概要

クラスタのマスタサーバ名を取得します。

形式

```
ArmClientGetMasterServer(  
    ByVal    lpszClusterName As String,  
    ByVal    lpszServerName As String,  
    ByRef    lpcchServerName As Long  
) As Long
```

パラメータ

[i] lpszClusterName	クラスタ名を指定する、NULLで終了する文字列へのポインタです。 ローカルサーバのクラスタを指定する場合は、ARMCLAPI_LOCAL_CLUSTERを指定することができます。この場合、ローカルサーバがクラスタサーバでなければエラー (ARMCLAPI_ERR_CLUSTER_NOT_FOUND) を返却します。
[o] lpszServerName	主サーバ名 (終端のNULL文字を含む) を受け取るバッファへのポインタです。
[io] lpcchServerName	lpszServerNameで指定されるバッファのサイズを指定する変数へのポインタです。関数のリターン時にはバッファに格納された文字数 (終端のNULL文字を含まない) をセットします。 指定バッファサイズが小さすぎた場合には、必要な文字数 (終端のNULL文字を含む) がセットし、拡張エラー情報でARMCLAPI_ERR_BUFFER_TOO_SMALLが返却します。

戻り値

正常系：ARMTRUE 関数が正常終了しました。
異常系：ARMFALSE 関数が異常終了しました。

拡張エラー情報 (LastDLLError プロパティで取得)

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO API の初期化処理が行われていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_BUFFER_TOO_SMALL	データ返却用バッファが小さすぎます。
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	指定クラスタ名が存在しません。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、ローカルサーバのサーバ名を取得し、バッファlpszServerNameに格納します。
ローカルホストがクラスタに所属していない場合、本関数はARMFALSEを返却し拡張エラー情報でARMCLAPI_ERR_NOT_CLUSTER_SERVERを返却します。

7.1.3.24 ArmClientCheckClusterServer — ローカルサーバの情報

概要

ローカルコンピュータがクラスタサーバかどうかを判別します。

形式

```
ArmClientCheckClusterServer(  
    ByRef    lpdwHostType    As    Long  
) As    Long
```

パラメータ

[o] lpdwHostType	ローカルコンピュータの種別	
	ARMCLAPI_HOSTTYPE_IS_SERVER	クラスタサーバ
	ARMCLAPI_HOSTTYPE_IS_CLIENT	CLUSTERPROクライアント環境
	ARMCLAPI_HOSTTYPE_IS_NONE	上記のいずれも該当しません。

戻り値

正常系 : ARMTRUE 関数が正常終了しました。
異常系 : ARMFALSE 関数が異常終了しました。

拡張エラー情報 (LastDLLError プロパティで取得)

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアント API の初期化処理が行われていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、実行されているコンピュータがクラスタサーバであるか、クライアントであるかを判別します。

7.1.3.25 ArmClientGetServerName — ローカルサーバ名の取得

概要

ローカルサーバのサーバ名を取得します。

形式

```
ArmClientGetServerName(  
    ByVal    lpszServerName  As    String,  
    ByRef    lpchServerName  As    Long  
) As    Long
```

パラメータ

[o] lpszServerName	サーバ名（終端のNULL文字を含む）を受け取るバッファへのポインタです。
[io] lpchServerName	lpszServerNameで指定されるバッファのサイズを指定する変数へのポインタです。関数のリターン時に、バッファに格納された文字数（終端のNULL文字を含まない）をセットします。指定バッファサイズが小さすぎた場合には、必要な文字数（終端のNULL文字を含む）をセットし、拡張エラー情報でARMCLAPI_ERR_BUFFER_TOO_SMALLを返却します。

戻り値

正常系：ARMTRUE 関数が正常終了しました。
異常系：ARMFALSE 関数が異常終了しました。

拡張エラー情報（LastDLLError プロパティで取得）

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアント API の初期化処理が行われていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_NOT_CLUSTER_SERVER	ローカルホストはクラスタサーバではありません。
ARMCLAPI_ERR_REGISTRY_ACCESS_FAILED	レジストリのアクセスに失敗しました。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_BUFFER_TOO_SMALL	データ返却用バッファが小さすぎます。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、ローカルサーバのサーバ名を取得し、バッファlpszServerNameに格納します。
ローカルホストがクラスタに所属していない場合、本関数はARMFALSEを返却し、拡張エラー情報でARMCLAPI_ERR_NOT_CLUSTER_SERVERを返却します。

7.1.3.26 ArmClientGetClusterName — クラスタ名の取得

概要

ローカルサーバが所属するクラスタの名前を取得します。

形式

```
ArmClientGetClusterName(  
    ByVal    lpszClusterName As String,  
    ByRef    lpccchClusterName As Long  
) As Long
```

パラメータ

[o] lpszClusterName	クラスタ名（終端のNULL文字を含む）を受け取るバッファへのポインタです。lpszClusterNameで指定されるバッファのサイズを指定する変数へのポインタです。関数のリターン時に、バッファに格納された文字数（終端のNULL文字を含まない）をセットします。指定バッファサイズが小さすぎた場合には、必要な文字数（終端のNULL文字を含む）をセットし、拡張エラー情報でARMCLAPI_ERR_BUFFER_TOO_SMALLを返却します。
[io] lpccchClusterName	

戻り値

正常系：ARMTRUE 関数が正常終了しました。
異常系：ARMFALSE 関数が異常終了しました。

拡張エラー情報（LastDLLError プロパティで取得）

ARMCLAPI_ERR_NOT_INITIALIZED	CLUSTERPRO クライアント API の初期化処理が行われていません。
ARMCLAPI_ERR_CANNOT_CONNECT	CLUSTERPRO クライアントに接続できません。
ARMCLAPI_ERR_COMM_ERROR	CLUSTERPRO クライアントとの通信でエラーが発生しました。
ARMCLAPI_ERR_NOT_CLUSTER_SERVER	ローカルホストはクラスタサーバではありません。
ARMCLAPI_ERR_REGISTRY_ACCESS_FAILED	レジストリのアクセスに失敗しました。
ARMCLAPI_ERR_INVALID_PARAMETER	パラメータの指定が不正です。
ARMCLAPI_ERR_BUFFER_TOO_SMALL	データ返却用バッファが小さすぎます。
ARMCLAPI_ERR_NOT_SUPPORTED	この関数はサポートされていません。

説明

本関数は、ローカルサーバが所属するクラスタの名前を取得し、バッファlpszClusterNameに格納します。ローカルサーバがクラスタに所属していない場合、本関数はARMFALSEを返却し、拡張エラー情報でARMCLAPI_ERR_NOT_CLUSTER_SERVERを返却します。

7.1.3.27 WATCHFAILOVERPROC — フェイルオーバーグループ監視用コールバック関数

概要

ArmClientStartWatchFailoverEx関数により登録され、フェイルオーバーグループの状態変更時に呼び出される、アプリケーション定義のコールバック関数です。

形式

```
WATCHFAILOVERPROC(  
    ByVal    dwEventID    As    Long,  
    ByVal    dwWatchId    As    Long,  
    ByRef    lpArmFailoverInfo    As    ARMCLAPI_FAILOVER_INFO  
) As    Long
```

パラメータ

[i]	dwEventId	検出されたイベントの通し番号です。
[i]	dwWatchId	対応するArmClientStartWatchFailover関数を識別する監視IDです。
[i]	lpArmFailoverInfoEx	フェイルオーバー情報がセットされたユーザ定義型ARMCLAPI_FAILOVER_INFOへのポインタです。

戻り値

ARMTRUE 常に、この値を返却します。

説明

本関数は、ArmClientStartWatchFailoverEx関数により登録される、アプリケーション定義のコールバック関数であり、CLUSTERPROクライアントがフェイルオーバーグループの状態の変更を検出した時に呼び出されます。

本関数の関数名 “WatchFailoverProc” は、アプリケーション定義の関数名のプレースホルダです。

参照

ArmClientStartWatchFailoverEx, ArmClientCancelWatchFailoverEx,
ARMCLAPI_FAILOVER_INFO型

7.1.3.28 EnumClusterProc — クラスタ列挙用コールバック関数

概要

ArmClientEnumCluster関数で列挙されるクラスタ名を受け取るための、アプリケーション定義のコールバック関数です。

形式

```
EnumClusterProc(  
    ByVal    lpszClusterName As    String,  
    ByVal    lParam As    Long  
) As    Long
```

パラメータ

[i]	lpszClusterName	クラスタ名を示す、NULLで終わる文字列へのポインタです。
[i]	lParam	ArmClientEnumCluster関数で指定されたアプリケーション定義の値です。

戻り値

正常系：ARMTRUE	列挙を続ける場合に返却します。
異常系：ARMFALSE	列挙を止める場合に返却します。

説明

EnumClusterProc関数は、ArmClientEnumCluster関数の実行結果として、クラスタ名を受け取るための、アプリケーション定義のコールバック関数です。列挙が完了するか、または本関数でARMFALSEを返却するまで、クラスタ名が本関数に順次受け渡されます。

アプリケーションは、ArmClientEnumCluster関数実行時に本関数のアドレスを指定しなければなりません。本関数に渡されるlpszClusterNameの指す文字列の内容は、本関数をリターンするまでの間でのみ有効です。これらの文字列の内容を保持する必要がある場合には、本関数内で文字列の内容を別の記憶領域へコピーしてください。

本関数の関数名“EnumClusterProc”は、アプリケーション定義の関数名のプレースホルダです。

参照

ArmClientEnumCluster

7.1.3.29 EnumServerProc — サーバ列挙用コールバック関数

概要

ArmClientEnumServer関数で列挙されるサーバのコンピュータ名を受け取るための、アプリケーション定義のコールバック関数です。

形式

```
EnumServerProc(  
    ByVal lpszClusterName As String,  
    ByVal lpszServerName As String,  
    ByVal lParam As Long  
) As Long
```

パラメータ

[i]	lpszClusterName	クラスタ名を示す、NULLで終わる文字列へのポインタです。
[i]	lpszServerName	サーバのコンピュータ名を示す、NULLで終わる文字列へのポインタです。
[i]	lParam	ArmClientEnumServer関数で指定されたアプリケーション定義の値です。

戻り値

正常系：ARMTRUE 列挙を続ける場合に返却します。
異常系：ARMFALSE 列挙を止める場合に返却します。

説明

EnumServerProc関数は、ArmClientEnumServer関数の実行結果として、クラスタを構成するサーバのコンピュータ名を受け取るための、アプリケーション定義のコールバック関数です。サーバの列挙が完了するか、または本関数でFALSEを返却するまで、サーバのコンピュータ名が本関数に順次受け渡されます。アプリケーションは、ArmClientEnumServer関数実行時に本関数のアドレスを指定しなければなりません。本関数に渡されるlpszClusterName、lpszServerNameの指す文字列の内容は、本関数をリターンするまでの間でのみ有効です。これらの文字列の内容を保持する必要がある場合には、本関数内で文字列の内容を別の記憶領域へコピーしてください。本関数の関数名“EnumServerProc”は、アプリケーション定義の関数名のプレースホルダです。

参照

ArmClientEnumServer

7.1.3.30 EnumGroupProc — フェイルオーバーグループ列挙用コールバック関数

概要

ArmClientEnumGroup関数で列挙されるフェイルオーバーグループ名を受け取るための、アプリケーション定義のコールバック関数です。

形式

```
EnumGroupProc(  
    ByVal    lpszClusterName As String,  
    ByVal    lpszGroupName As String,  
    ByVal    lParam As Long  
) As Long
```

パラメータ

[i]	lpszClusterName	クラスタ名を示す、NULLで終わる文字列へのポインタです。
[i]	lpszGroupName	フェイルオーバーグループ名を示す文字列へのポインタです。
[i]	lParam	ArmClientEnumGroup関数で指定されたアプリケーション定義の値です。

戻り値

正常系：ARMTRUE 列挙を続ける場合に返却します。
異常系：ARMFALSE 列挙を止める場合に返却します。

説明

EnumGroupProc関数は、ArmClientEnumGroup関数の実行結果として、クラスタを構成するサーバのLAN Manager コンピュータ名を受け取るための、アプリケーション定義のコールバック関数です。サーバの列挙が完了するか、または本関数でARMFALSEを返却するまで、フェイルオーバーグループ名が本関数に順次受け渡されます。アプリケーションは、ArmClientEnumGroup関数実行時に本関数のアドレスを指定しなければなりません。本関数に渡されるlpszClusterName、lpszGroupNameの指す文字列の内容は、本関数をリターンするまでの間でのみ有効です。これらの文字列の内容を保持する必要がある場合には、本関数内で文字列の内容を別の記憶領域へコピーしてください。本関数の関数名“EnumGroupProc”は、アプリケーション定義の関数名のプレースホルダです。

参照

ArmClientEnumGroup

7.1.3.31 EnumServerFromGroupProc — サーバ列挙用コールバック関数

概要

ArmClientEnumServerFromGroup関数で列挙されるサーバのコンピュータ名を受け取るための、アプリケーション定義のコールバック関数です。

形式

```
EnumServerFromGroupProc (  
    ByVal    lpszClusterName As    String,  
    ByVal    lpszGroupName  As    String,  
    ByVal    lpszServerName  As    String,  
    ByVal    lParam As    Long  
) As    Long
```

パラメータ

[i]	lpszClusterName	クラスタ名を示す、NULLで終わる文字列へのポインタです。
[i]	lpszGroupName	フェイルオーバーグループ名を示す、NULLで終わる文字列へのポインタです。
[i]	lpszServerName	サーバのLAN Manager コンピュータ名を示す、NULLで終わる文字列へのポインタです。
[i]	lParam	ArmClientEnumServerFromGroup関数で指定されたアプリケーション定義の値です。

戻り値

正常系：ARMTRUE 列挙を続ける場合に返却します。
異常系：ARMFALSE 列挙を止める場合に返却します。

説明

EnumServerFromGroupProc関数は、ArmClientEnumServerFromGroup関数の実行結果として、フェイルオーバーグループが移動可能なサーバのコンピュータ名を受け取るための、アプリケーション定義のコールバック関数です。サーバの列挙が完了するか、または本関数でARMFALSEを返却するまで、サーバのコンピュータ名が本関数に順次受け渡されます。

アプリケーションは、ArmClientEnumServerFromGroup関数実行時に本関数のアドレスを指定しなければなりません。

本関数に渡されるlpszClusterName, lpszGroupName, lpszServerNameの指す文字列の内容は、本関数をリターンするまでの間でのみ有効です。これらの文字列の内容を保持する必要がある場合には、本関数内で文字列の内容を別の記憶領域へコピーしてください。

本関数の関数名“EnumServerFromGroupProc”は、アプリケーション定義の関数名のプレースホルダです。

参照

ArmClientEnumServerFromGroup

7.1.3.32 EnumVirtualIPProc — 仮想 IP アドレス列挙用コールバック関数

概要

ArmClientEnumVirtualIP関数で列挙される仮想IPアドレスを受け取るための、アプリケーション定義のコールバック関数です。

形式

```
EnumVirtualIPProc(  
    ByVal    lpszClusterName As String,  
    ByVal    lpszGroupName As String,  
    ByVal    dwVirtualIP As Long,  
    ByVal    lParam As Long  
) As Long
```

パラメータ

[i]	lpszClusterName	クラスタ名を示す、NULLで終わる文字列へのポインタです。
[i]	lpszGroupName	フェイルオーバーグループ名を示す文字列へのポインタです。
[i]	dwVirtualIP	仮想IPアドレスです。ネットワークバイトオーダーで格納されます。
[i]	lParam	ArmClientEnumVirtualIP関数で指定されたアプリケーション定義の値です。

戻り値

正常系：ARMTRUE 列挙を続ける場合に返却します。
異常系：ARMFALSE 列挙を止める場合に返却します。

説明

EnumVirtualIPProc関数は、ArmClientEnumVirtualIP関数の実行結果として、クラスタに登録されている仮想IPアドレスを受け取るための、アプリケーション定義のコールバック関数です。仮想IPアドレスの列挙が完了するか、または本関数でARMFALSEを返却するまで、フェイルオーバーグループ名が本関数に順次受け渡されます。

アプリケーションは、ArmClientEnumVirtualIP関数実行時に本関数のアドレスを指定しなければなりません。本関数に渡されるlpszClusterName、lpszGroupNameの指す文字列の内容は、本関数をリターンするまでの間でのみ有効です。これらの文字列の内容を保持する必要がある場合には、本関数内で文字列の内容を別の記憶領域へコピーしてください。

本関数の関数名“EnumVirtualIPProc”は、アプリケーション定義の関数名のプレースホルダです。

参照

ArmClientEnumVirtualIP

7.1.3.33 EnumVirtualComputerProc — 仮想コンピュータ名列挙用コールバック関数

概要

ArmClientEnumVirtualComputer関数で列挙される仮想コンピュータ名を受け取るための、アプリケーション定義のコールバック関数です。

形式

```
EnumVirtualComputerProc(  
    ByVal lpszClusterName As String,  
    ByVal lpszGroupName As String,  
    ByVal lpszVirtualComputer As String,  
    ByVal lParam As Long  
) As Long
```

パラメータ

- | | | |
|-----|---------------------|---|
| [i] | lpszClusterName | クラスタ名を示す、NULLで終わる文字列へのポインタです。 |
| [i] | lpszGroupName | フェイルオーバーグループ名を示す文字列へのポインタです。 |
| [i] | lpszVirtualComputer | 仮想コンピュータ名を示す文字列へのポインタです。 |
| [i] | lParam | ArmClientEnumVirtualComputer関数で指定されたアプリケーション定義の値です。 |

戻り値

- | | |
|--------------|-----------------|
| 正常系：ARMTRUE | 列挙を続ける場合に返却します。 |
| 異常系：ARMFALSE | 列挙を止める場合に返却します。 |

説明

EnumVirtualComputerProcは、ArmClientEnumVirtualComputer関数の実行結果として、クラスタに登録されている仮想コンピュータ名を受け取るための、アプリケーション定義のコールバック関数です。仮想コンピュータ名の列挙が完了するか、または本関数でARMFALSEを返却するまで、仮想コンピュータ名が本関数に順次受け渡されます。

アプリケーションは、ArmClientEnumVirtualComputer関数実行時に本関数のアドレスを指定しなければなりません。

本関数に渡されるlpszClusterName、lpszGroupNameの指す文字列の内容は、本関数をリターンするまでの間でのみ有効です。これらの文字列の内容を保持する必要がある場合には、本関数内で文字列の内容を別の記憶領域へコピーしてください。

本関数の関数名“EnumVirtualComputerProc”は、アプリケーション定義の関数名のプレースホルダです。

参照

ArmClientEnumVirtualComputer

7.1.3.34 EnumGroupFromLogicalServiceProc — 論理サービス名列挙用コールバック関数

概要

ArmClientEnumGroupFromLogicalService関数で列挙されるフェイルオーバーグループ名を受け取るための、アプリケーション定義のコールバック関数です。

形式

```
EnumGroupFromLogicalServiceProc(  
    ByVal lpszClusterName As String,  
    ByVal lpszServerName As String,  
    ByVal lpszLogicalServiceName As String,  
    ByVal lParam As Long  
) As Long
```

パラメータ

[i]	lpszClusterName	クラスタ名です。NULLで終わる文字列へのポインタです。
[i]	lpszGroupName	指定された論理サービス名に対応するフェイルオーバーグループ名です。NULLで終わる文字列へのポインタです。
[i]	lpszLogicalServiceName	論理サービス名です。NULLで終わる文字列へのポインタです。
[i]	lParam	ArmClientEnumServerFromLogicalService関数で指定されたアプリケーション定義の値です。

戻り値

正常系：ARMTRUE 列挙を続ける場合に返却します。
異常系：ARMFALSE 列挙を止める場合に返却します。

説明

EnumGroupFromLogicalServiceProc関数は、ArmClientEnumGroupFromLogicalService関数の実行結果として、論理サービス名に対応するフェイルオーバーグループ名を受け取るための、アプリケーション定義のコールバック関数です。フェイルオーバーグループの列挙が完了するか、または本関数でARMFALSEを返却するまで、フェイルオーバーグループ名が本関数に順次受け渡されます。

アプリケーションは、ArmClientEnumGroupFromLogicalService関数実行時に本関数のアドレスを指定しなければなりません。

本関数に渡されるlpszClusterName、lpszGroupName、lpszLogicalServiceNameの指す文字列の内容は、本関数をリターンするまでの間でのみ有効です。これらの文字列の内容を保持する必要がある場合には、本関数内で文字列の内容を別の記憶領域へコピーしてください。

本関数の関数名“EnumGroupProcFromLogicalService”は、アプリケーション定義の関数名のプレースホルダーです。

参照

ArmClientEnumGroupFromLogicalService

7.2 ユーザー定義型（構造体）

ARMCLAPI_FAILOVER_INFO ユーザー定義型

概要

ARMCLAPI_FAILOVER_INFO型は、フェイルオーバー発生／完了時の情報を格納します。

形式

```
Public Type ARMCLAPI_FAILOVER_INFO
    Status      As Long
    ClusterName As String * CLUSTER_NAME_LEN
    GroupName   As String * SERVER_NAME_LEN
    VirtualIP   As Long
    SrcComputerName As String * SERVER_NAME_LEN
    SrcIP       As Long
    DstComputerName As String * SERVER_NAME_LEN
    DstIP       As Long
    StatusEx    As Long
    Event       As Long
End Type
```

メンバ

ClusterName	フェイルオーバーが発生したクラスタ名	
GroupName	フェイルオーバーグループ名	
VirtualIP	仮想IPアドレス名(ネットワークバイトオーダー、フェイルオーバーグループに複数仮想IPアドレスが登録されていても一つのみ返却する)	
SrcComputerName	フェイルオーバー開始時	フェイルオーバー元のサーバ名
	フェイルオーバー完了時	不定
	シャットダウン時	シャットダウンしたサーバ名
	起動時	不定
SrcIP	SrcComputerNameの実IP(ネットワークバイトオーダー)	
DstComputerName	フェイルオーバー開始時	不定
	フェイルオーバー完了時	フェイルオーバー先のサーバ名
	シャットダウン時	不定
	起動時	起動したサーバ名
DstIP	DstComputerNameの実IP(ネットワークバイトオーダー)	
Event	CLUSTERPROモニタが検出した状態変更の種類	
	ARMCLAPI_EVENT_FAILOVER_START	フェイルオーバー発生
	ARMCLAPI_EVENT_FAILOVER_END	フェイルオーバー終了
	ARMCLAPI_EVENT_SHUTDOWN	シャットダウン
	ARMCLAPI_EVENT_BOOT	サーバ起動
StatusEx	現在の状態	
	ARMCLAPI_GROUP_STATUS_RUNNING	正常動作中
	ARMCLAPI_GROUP_STATUS_MISS	資源獲得に失敗
	ARMCLAPI_GROUP_STATUS_STOPPED	停止中
	ARMCLAPI_GROUP_STATUS_FAILOVER	フェイルオーバー処理中

7.3 定数

7.3.1 拡張拡張エラー情報

定数名	型	値
ARMCLAPI_ERR_CANNOT_LOAD_DLL	Long	&H20000001
ARMCLAPI_ERR_CANNOT_CONNECT	Long	&H20000004
ARMCLAPI_ERR_COMM_ERROR	Long	&H20000005
ARMCLAPI_ERR_NOT_ENOUGH_MEMORY	Long	&H20000010
ARMCLAPI_ERR_BUFFER_TOO_SMALL	Long	&H20000011
ARMCLAPI_ERR_ALREADY_INITIALIZED	Long	&H20000020
ARMCLAPI_ERR_NOT_INITIALIZED	Long	&H20000021
ARMCLAPI_ERR_INVALID_PARAMETER	Long	&H20000050
ARMCLAPI_ERR_INVALID_ID	Long	&H20000051
ARMCLAPI_ERR_GROUP_NOT_FOUND	Long	&H20000071
ARMCLAPI_ERR_SERVER_NOT_FOUND	Long	ARMCLAPI_ERR_GROUP_NOT_FOUND
ARMCLAPI_ERR_CLUSTER_NOT_FOUND	Long	&H20000075
ARMCLAPI_ERR_VIRTUALIP_NOT_FOUND	Long	&H20000076
ARMCLAPI_ERR_NOT_CLUSTER_SERVER	Long	&H20000077
ARMCLAPI_ERR_LOGICAL_SERVICE_NOT_FOUND	Long	&H20000078
ARMCLAPI_ERR_VCOM_NOT_FOUND	Long	&H20000079
ARMCLAPI_ERR_CANCELLED	Long	&H20000080
ARMCLAPI_ERR_REGISTRY_ACCESS_FAILED	Long	&H200000B0
ARMCLAPI_ERR_INVALID_HANDLE	Long	&H20000090
ARMCLAPI_ERR_NOT_SUPPORTED	Long	&H200000A0

7.3.2 ハンドル

定数名	型	値
ARMCLAPI_ERROR_HANDLE	Long	0

7.3.3 イベント

定数名	型	値
ARMCLAPI_EVENT_SHUTDOWN	Long	&H1
ARMCLAPI_EVENT_FAILOVER_START	Long	&H4
ARMCLAPI_EVENT_FAILOVER_END	Long	&H5
ARMCLAPI_EVENT_BOOT	Long	&HB

7.3.4 フェイルオーバーグループの状態

定数名	型	値
ARMCLAPI_GROUP_STATUS_STOPPED	Long	0
ARMCLAPI_GROUP_STATUS_RUNNING	Long	1
ARMCLAPI_GROUP_STATUS_FAILOVER	Long	2
ARMCLAPI_GROUP_STATUS_MISS	Long	3

7.3.5 サーバステータス

定数名	型	値
ARMCLAPI_SERVER_STATUS_SINGLE_SERVER	Long	&H1000000
ARMCLAPI_SERVER_STATUS_CLUSTER	Long	&H2000000
ARMCLAPI_SERVER_STATUS_OFF_CLUSTER	Long	&H4000000
ARMCLAPI_SERVER_STATUS_STOPPED	Long	&H10000
ARMCLAPI_SERVER_STATUS_DOWN	Long	&H20000
ARMCLAPI_SERVER_STATUS_RECOVERING	Long	&H40000
ARMCLAPI_SERVER_STATUS_FAILBACK	Long	&H80000
ARMCLAPI_SERVER_STATUS_BACKUP	Long	&H1
ARMCLAPI_SERVER_STATUS_FAIL	Long	&H2
ARMCLAPI_SERVER_STATUS_FAILMISS	Long	&H4

7.3.6 ホストタイプ

定数名	型	値
ARMCLAPI_HOSTTYPE_IS_NONE	Long	0
ARMCLAPI_HOSTTYPE_IS_CLIENT	Long	1
ARMCLAPI_HOSTTYPE_IS_SERVER	Long	2

7.3.7 真理値

定数名	型	値
ARMTRUE	Long	1
ARMFALSE	Long	0

7.4 注意事項

7.4.1 関数の戻り値

- * CLUSTERPRO クライアントAPIは、成功したか失敗したかについて、WIN32APIのBOOL値(32Bit値)を戻り値として返します。そのため、Visual Basic側では、As Booleanではなく、As Longで宣言された変数で受ける必要があります。
また、値としてWIN32APIのTRUEが渡されるので、Visual BasicのTrueは返却されません。クライアントAPIで提供している定数ARMTRUE/ARMTFALSEを用いるようにしてください。

7.4.2 API 関数での引数の文字列

- * APIに文字列変数を渡して、文字列を取得する際は、全ての領域を初期化してください。
- * 引数に渡す文字列は、ANSI文字列として扱ってください。Visual Basic上の文字列はUnicodeで扱われますが、APIとのインタフェースでは、一般にANSI文字列を利用します。(シフトJISはANSI文字列です。)

7.4.3 コールバック関数での文字列の受け取り

- * コールバック関数でAPIから文字列を受け取る際は、Byte型配列のユーザー定義型で受け取ってください。
- * コールバック関数でAPIから受け取った文字列は、通常、Unicode(WIDE文字列)ではありません(ANSI文字列が返却されます)。しかし、Visual Basicで作成されたアプリケーションは文字列をUnicodeで扱いますので、いったんUnicodeに変換しないと、表示や印字で文字化けが発生したり、文字列の比較などが行えない等の問題が発生します。文字列の変換は、変換後文字列=StrConv(変換前文字列,vbUnicode)等とすることで実現できます。
- * コールバック関数でAPIから受け取った文字列を切り出したときは、本来、終端となるべき位置に終端があるとは限りません。ANSI文字列は¥0で終了しますが、WIDE文字列は¥0¥0で終了するため、一つだけ¥0があっても終端とみなさないのが原因です。
したがって、受け取った文字列の¥0の位置を探し(InStrB(文字列,CharB(0))とする)、その手前の位置で文字列を切り出す(Left(文字列,手前の位置))必要があります。

7.4.4 拡張エラー情報

- * APIの拡張エラー情報は、WIN32APIのGetLastError関数ではなく、Visual Basic標準のErrオブジェクトのLastDLLErrorプロパティを参照してください。

本章で、本文中に表記している NULL は、Visual Basic の &H0 に相当します。
--

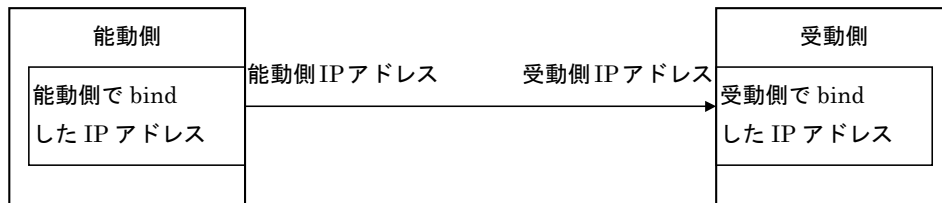
8 仮想IPアドレス使用時の注意事項

仮想IPアドレスを用いる必要のあるアプリケーションを開発する場合は、本節の内容を把握した上で行ってください。

8.1 問題点/制限事項

仮想IPアドレスを用いること、及びクラスタサーバのIPアドレスが切り替わることによって、サーバ・クライアントアプリケーション上で生じる問題点と、その対応策について説明します。

コネクションを要求する側を能動側、要求を待ち受ける側を受動側として説明します。
図の矢印は、接続要求の方向を示しています。

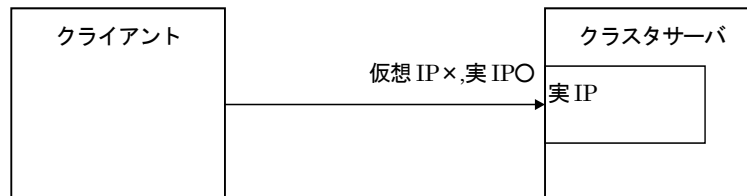


(1) クラスタサーバが受動側の場合

【問題点】

クラスタサーバ側でIPアドレスを明示的にbindした場合、接続できないことがあります。

- * クラスタサーバ側のAPが、実IPを明示的にbindした場合



実IPに対する接続は確立できますが、仮想IPに対する接続はできません。

- * クラスタサーバ側のAPが、仮想IPを明示的にbindした場合



仮想IPに対する接続は確立できるが、実IPに対する接続はできません。

【解決策および制限事項】

クラスタサーバで動作し、TCP/IPの接続を受けるAPは、IPアドレスを明示的にbindしないでください。

(2) クラスタサーバが能動側の場合

【問題点】

クラスタサーバ側のIPアドレスが切り替わることを考慮する必要があります。

- * bindを行わないか、bind時にIPアドレスを明示的に指定しない場合



接続は実IPで行われます。

- * 実IPを明示的にbindした場合



接続はbindした実IPで行われます。

- * 仮想IPを明示的にbindした場合



接続はbindした仮想IPで行われます。

クラスタサーバ側でIPアドレスを明示的にbindしなかった場合（通常はbindしない）には、クラスタサーバからの実IPによる接続要求となります。そのため、接続先のAPやサービスに、以下のような性質がある場合に問題が発生します。

- * セキュリティなどの理由で、接続元のIPアドレスを限定している場合
- * アカウンティングなどで、接続元のIPアドレスを認証の手段に用いている場合
- * 再接続の際に、前回の接続元のIPアドレスと同一のIPアドレスからの接続でないと正しい動作とならない場合

【解決策および制限事項】

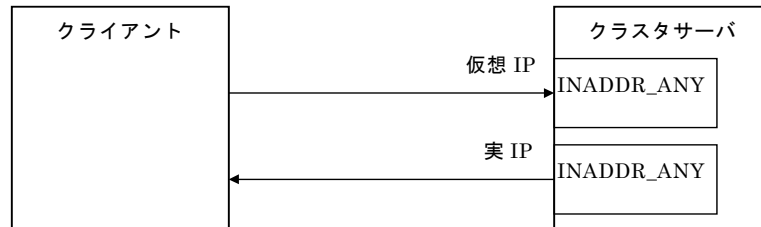
クラスタサーバから他サーバに接続する場合には、以下の条件を満たさなければなりません。

- * 接続先のサービスやAPが接続元のIPアドレスを限定する場合は、両クラスタサーバの実IP、仮想IPいずれの接続も許可するように設定すること
- * アカウンティングなどで、接続元のIPアドレスを認証の手段に用いる場合は、以下のようにすること
(以下の2つの方法のうち、前者を推奨する)
 - + 接続先のサービスやAPで、クラスタ構成の仮想IPアドレスおよび実IPアドレスは全て同一とみなすようにすること
 - + クラスタサーバ側で接続要求をするソケットについて、仮想IPを明示的にbindする。但し、接続先ホストが仮想IPとの接続が問題なく行えること
- * 再接続の際に、接続元のIPアドレスと同一のIPアドレスからの接続でないと正しい動作とならない場合は上記と同様です。

(3) クラスタサーバとクライアント間で複数のコネクションを張る場合

【問題点】

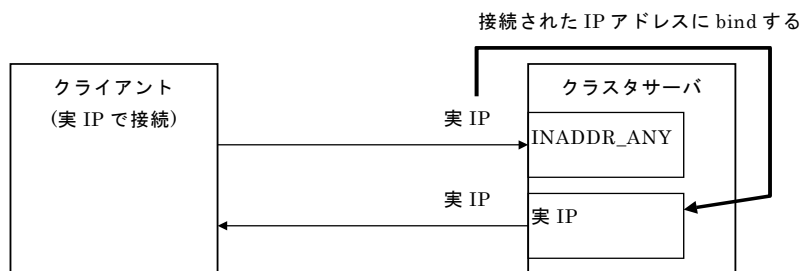
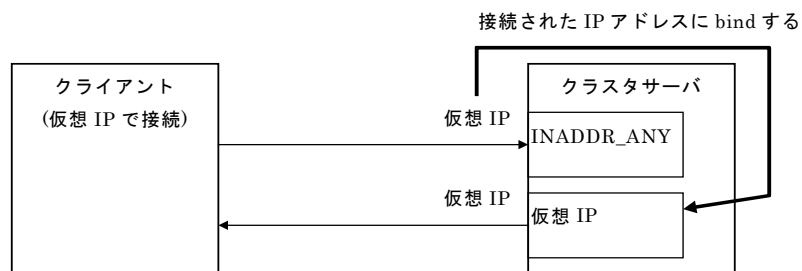
クライアントからクラスタサーバに接続し、要求によって新たにクラスタサーバからクライアントに接続する場合、2つのコネクションが別のIPアドレスとなって、うまく動作しない場合があります。



【解決策および制限事項】

クラスタサーバのAP/サービスが、以下の条件を満たす必要があります。

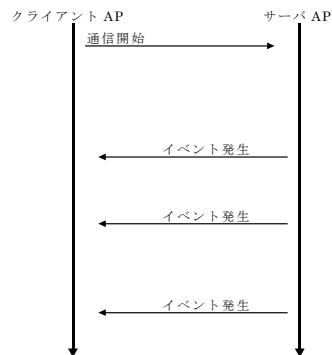
- * 2本目以降のコネクションを張る際に、1本目のコネクションでクライアントから指定されたIPアドレスを新たなコネクションのsocketに明示的にbindすること



(4) UDPを使用する場合

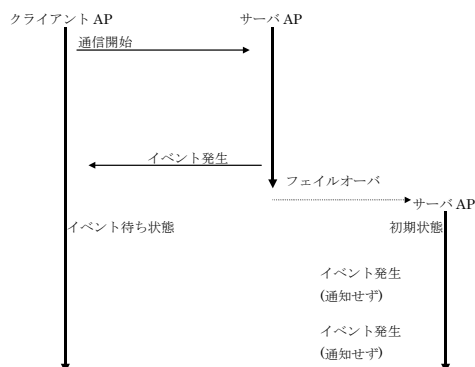
【問題点】

UDPはコネクションレスであるため、サーバの切替が発生した後も、切替わったことが判別できず、サーバ側とクライアント側で異なる状態とならないように注意が必要です。
下図のように、サーバ側のイベント発生時など、任意のタイミングでサーバからクライアントに通知するような通信を行うクライアント・サーバAPを例に述べます。



クラスタを意識していないAPの場合、下図のように、サーバがフェイルオーバーした場合、サーバがフェイルオーバーする際にうまく動作しないことがあります。

- * サーバAPは初期状態となり、クライアントの存在を関知していない。
- * クライアントはサーバAPからの通知を待ち受けたままとなる。

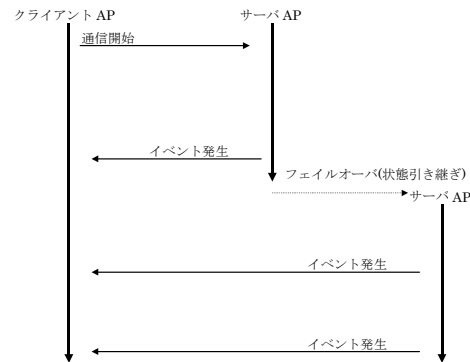


【解決策】

サーバ側で対処する方法と、クライアント側で対処する方法があります。

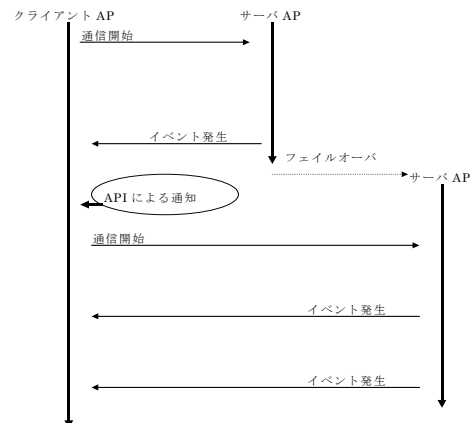
① サーバAPで、状態を切替ディスクを用いて引き継ぐ方法

- * サーバAPは、通知すべきクライアントを切替ディスクに保存しておく。
- * フェイルオーバー後に、サーバAPが起動する時に、切替ディスクから、イベント発生を通知する先のクライアント一覧を読み取って、フェイルオーバー前と同じ状態で再開する。
- * クライアントはどちらのサーバからのイベント発生通知も同様に受け取ることが可能であること。



② クライアントAPでAPIを使用する方法

- * サーバAPは、フェイルオーバー後の再起動は初期状態で起動する。
- * クライアントAPは、CLUSTERPROクライアントAPIを使用して、フェイルオーバーを監視する。フェイルオーバーが完了すると、サーバAPに対して、再度通信を開始する。



8.2 サーバからのブロードキャストをきっかけとした接続

接続形態

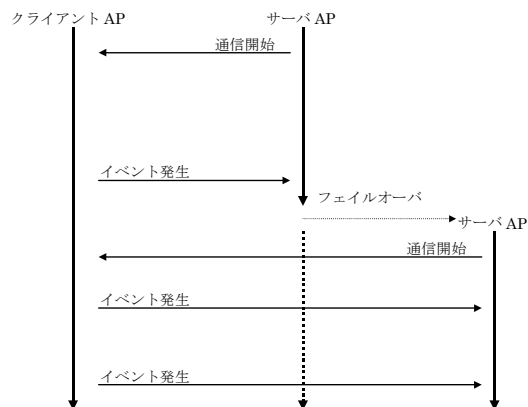
サーバAPは起動時等に、ブロードキャストメッセージを送信します。クライアントはブロードキャストメッセージの受信をきっかけに、サーバと通信を開始します。

接続方法

サーバ側はブロードキャストメッセージの発信アドレスは明示的にbindしない。(実IPアドレスが用いられる) クライアントは、ブロードキャストメッセージの送信元に対して接続する。

フェイルオーバー時の動作

フェイルオーバーが発生すると、フェイルオーバー先でサーバAPが起動されます。そのとき、フェイルオーバー先の実IPアドレスを発信元IPアドレスとするブロードキャストメッセージが送信されるため、クライアントAPは正しくフェイルオーバー先のサーバAPと通信することができます。



補足説明

サーバ側のブロードキャストメッセージがきっかけとなる場合など、サーバ側からの要求で接続が開始する場合は、クライアント側でどのサーバに接続すればよいかの情報が得られるため、仮想IPアドレスを使用する必要はありません。

9 関数・構造体索引

ARM_EVENT_INFO	70	ArmClientTerminateApiEx	161
ARM_EVENT_PARAM	135	ArmEnumGroup	30
ARM_FAILOVER_INFO_EX	228	ArmEnumLogicalService	41, 91
ARM_GROUP_INFO	59	ArmEnumResource	35, 83
ARM_GROUP_STATUS	68	ArmEnumResource2	85
ARM_RESOURCE_INFO	60, 128	ArmEnumServer	25, 80
ARM_RESOURCE_STATUS	69	ArmGetClusterName	24, 79
ARM_SERVER_INFO	58, 127	ArmGetEventInfo	54
ArmCancelEventWatch	53	ArmGetGroupInfo	33
ArmCancelWatchServer	104	ArmGetGroupStatus	47
ArmCancelWatchServerEvent	109	ArmGetLogicalServiceInfo	93
ArmCheckResourceOwner	112	ArmGetResourceInfo	39, 88
ARMCLAPI_FAILOVER_INFO	227	ArmGetResourceInfo2	89
ArmClientCancelWatchFailoverEx	177	ArmGetResourceStatus	49
ArmClientCheckClusterServer	223	ArmGetServerEvent	110
ArmClientEnumCluster	183	ArmGetServerInfo	28, 82
ArmClientEnumGroup	188	ArmGetServerStatus	45, 97
ArmClientEnumGroupFromLogicalService	214	ArmInitializeApi	20, 76
ArmClientEnumServer	185	ArmRegCloseKey	121
ArmClientEnumServerFromGroup	194	ArmRegCreateKey	116
ArmClientEnumServerFromLogicalService	216	ArmRegDeleteKey	118
ArmClientEnumVirtualComputer	210	ArmRegDeleteValue	124
ArmClientEnumVirtualIP	203	ArmRegFlushKey	120
ArmClientGetClusterFromClusterServer	191	ArmRegisterWatchServerEvent	107
ArmClientGetClusterFromGroup	192	ArmRegOpenKey	114
ArmClientGetClusterFromServer	193	ArmRegSetValue	122
ArmClientGetClusterName	225	ArmStartEventWatch	51
ArmClientGetClusterNameFromVirtualIP	201	ArmStartWatchServer	102
ArmClientGetClusterServerStatus	180	ArmTerminateApi	22, 77
ArmClientGetComputerName	166	ArmWatchServer	99
ArmClientGetGroupFromVirtualComputer	208	EnumClusterProc	184
ArmClientGetGroupFromVirtualIP	198	EnumGroupFromLogicalServiceProc	217
ArmClientGetGroupStatus	168	EnumGroupProc	190
ArmClientGetGroupStatusEx	171	EnumResourceProc	87
ArmClientGetHeartbeatTimeout	219	EnumServerFromGroupProc	196
ArmClientGetIPAddr	164	EnumServerProc	81, 187
ArmClientGetIPAddrFromVirtualIP	206	EnumVirtualComputerProc	212
ArmClientGetMasterServer	221	EnumVirtualIPProc	205
ArmClientGetPrimaryServerFromVirtualIP	200	GetSrvInfoProc	95
ArmClientGetServerName	224	GroupEnumProc	32
ArmClientGetServerStatus	170	LogicalServiceEnumProc	43
ArmClientGetServerStatusEx	173	ResourceEnumProc	37
ArmClientGetVersionInfo	162	ServerEnumProc	27
ArmClientInitializeApiEx	160	WatchFailoverProc	178
ArmClientStartWatchFailoverEx	175	WatchServerProc	105