Intel(R) PRO ネットワーク アダプタ WMI および CDM プロバイダ ユーザ ガイド

下記の情報は Dell 社による検証なしで、解説されたデバイスの出荷業者により提供され、下記の制限と免責条項 の条件が適用されます。

制限 および免責条項

すべての説明、警告、規制の認証および保証を含む本書の情報は、出荷業者によって提供されており、Dell 社は証明または検証をしていません。Dell 社は説明書に従って実行、または従わずに実行したため発生した破損には、いずれにも一切責任を負いません。

本書に典拠された所有権、有効性、速度、または品質に関するすべての記述は、出荷業者によって作られたものであり、Dell 社のものではありません。Dell 社は、それらの記述の正確性、完全性、または実証性の知識を免責条項とします。陳述または請求に関する質問またはコメントは、出荷業者に直接ご連絡ください。

初回リリース日：*2003* 年 *10* 月

目次に戻る

はじめに：Intel(R) PRO ネットワーク アダプタ WMI および CDM プロバイダ ユーザ ガイド

概要

*Intel(R) PRO ネットワーク アダプタ WMI および CDM プロバイダ ユーザ ガイド*へようこそ。本書では、Intel PRO ネットワーク アダプタ WMI および CDM プロバイダの概要を説明します。Windows Management Interface (WMI) Provider は、Network Configuration Services (NCS) のネットワーク設定ブロックです。NCS は、業界の標準的なメソッドを使用した、インテルのすべてのエンドステーション ネットワーキング テクノロジーを導入および管理するための手段です。Intel PRO の Common Diagnostic Model (CDM) Provider は、CIM 2.5 と WMI の標準に準拠した、上層インターフェイス API です。下層インターフェイスでは、CDM Provider によって PROSet ソフトウェア スタックの下層レイヤにクライアント インターフェイスが実装されます。これにより、すべての PROSet メカニズムでデータの整合性が保たれます。

WMI Provider と CDM Providerは、ソフトウェア コンポーネントのセットで、Intel WMI ネットワーク クラスを実装します。これらのクラスは Desktop Management Task Force (DMTF) CIM Schema バージョン 2.5 に基づいています。

本書では、本製品に付属の Managed Object Format (MOF) ファイルに含まれる情報を繰り返すことはしません。たとえば、個々の属性の詳細な意味については、MOF 属性の説明を参照してください。

本書では、Intel PROSet などの WMI アプリケーションが、どのようにクラスを使用してシステムのネットワークを設定するか、また、どのようにクラスを使用して Intel ネットワーク インターフェイス カードをテストするかを説明します。本文書は、WMI API と WMI SDK (http://www.microsoft.com/（英語） から参照可能) に精通したユーザを対象に書かれています。

先頭に戻る

関連文書

WMI テクノロジをよりよく理解するには、次の文書を参照してください。

- CIM スキーマ バージョン 2.0、2.2 (発行：Desktop Management Task Force (DMTF))。http://www.dmtf.org（英語）から参照可能。
- Microsoft Windows Management Interface (および、その他の管理に関する情報)。http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wmisdk/wmi/wmi_start_page.asp（英語） から参照可能。
- Web-Based Enterprise Management (WBEM) イニシアチブ (発行：DMTF)。 http://www.dmtf.org/standards/wbem（英語）から参照可能。
- WMI (Microsoft CIM 実装) SDK。http://msdn.microsoft.com/downloads/（英語）から参照可能。
- System Diagnostic Model White Paper (発行：DFTM)。 http://www.dmtf.org/standards/documents/CIM/DSP0138.pdf（英語）から参照可能。

警告：本製品には、コンピュータ システムやネットワークを攻撃、または動作不能にするために使用できる情報が含まれています。本製品を実装するには、Microsoft オペレーティング システムのセキュリティ機能について、十分に熟知していることが前提条件となります。また開発者やユーザは、本製品の実働環境における実装を使用する前に、セキュリティに関する問題について Microsoft 社に相談することを強く推奨します。

制限と免責条項をすべてお読みください。

目次に戻る 先頭に戻る

目次に戻る

WMI：Intel(R) PRO ネットワーク アダプタ WMI および CDM プロバイダ ユーザ ガイド

概要

WBEM (Web-Based Enterprise Management) は、企業のシステム マネージャに、標準化されたコスト効果の高いエンド ステーション管理の手段を提供するために設計された DMTF (Desktop Management Task Force (DMTF) のアプローチです。WBEM のアプローチは、簡単なワークステーションの設定から、複数のプラットフォームにわたる大規模な企業管理まで、多数のタスクを含んでいます。このアプローチの中心となるのは、一般的な管理環境に存在するオブジェクトを表す拡張可能なデータ モデル CIM (Common Information Model) と、モデル化されたデータの定義と保管のための言語 MOF (Managed Object Format) です。

WMI (Windows Management Instrumentation) は、Microsoft* Windows* プラットフォームのための WBEM の実装です。

WMI には次の 3 つの主なコンポーネントがあります。

- Core - これらのコンポーネントはオペレーティング システムの一部です。これらは WMI 対応のアプリケーションが機能するために必要で、SDK を使用するためにはインストールされていることが必要です。
- SDK - SDK には、WMI スキーマのブラウズ、スキーマの拡張、プロバイダの作成、WMI イベントの登録と使用のためのツールが含まれています。また、WMI を使用するアプリケーションの開発に有用な文書も含まれています。SDK は Microsoft プラットフォーム SDK のインストールの一環としてインストールされ、Windows NT4 SP4 または SP5、Windows 2000、Windows Me、Windows XP、および Windows Server 2003 でサポートされています。
- Tools - Microsoft WMI Tools は、まったく新しい世代の管理アプリケーションとソリューションを作成するために必要なツールを、開発者に提供します。Tools には、WMI から管理データにアクセスするプロセスをガイドする文書やツールが満載されています。

WMI アーキテクチャは次のコンポーネントで構成されています。

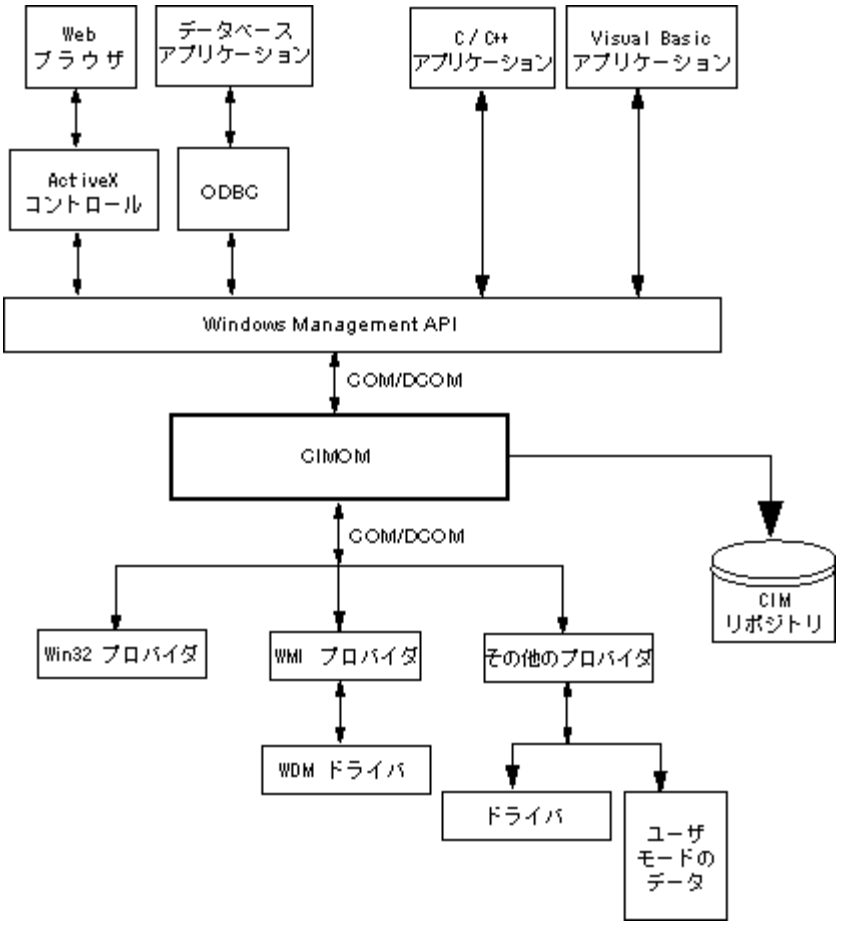
- 管理アプリケーション
- 管理されるオブジェクト
- プロバイダ
- 管理インフラストラクチャ (Windows Management と Windows Management リポジトリで構成)
- Windows Management API (COM/DCOM を使用して、プロバイダや管理アプリケーションが Windows 管理インフラストラクチャと通信できるようにする。)

管理アプリケーションでは、企業の論理的または物理的コンポーネントである、管理されるオブジェクトからのデータを処理、または表示します。これらのコンポーネントは CIM を使用して作成され、Windows Management を通じてアプリケーションからアクセスされます。プロバイダは Windows Management API を使用して、Windows Management に管理されるオブジェクトからのデータを供給し、アプリケーションからの要求を処理し、イベントの通知を生成します。

管理インフラストラクチャは Windows Management (管理アプリケーションとプロバイダ間の通信処理用) と Windows Management リポジトリ (データの整理用) で構成されます。Windows Management リポジトリには、静的な管理データが保管されます。動的なデータは、プロバイダの要求に応じて生成されます。データは、MOF 言語コンパイラか、Windows Management API を使用して、リポジトリに配置されます。

アプリケーションとプロバイダは、イベント通知やクエリー処理などのサービスを提供する Windows Management API を使用し、Windows Management を通じて通信します。

次の図に、WMI アーキテクチャ コンポーネントの相関関係を示します。



先頭に戻る

Common Information Model (CIM スキーマ)

Common Information Model (CIM) は、管理される環境における、すべてのタイプの論理的および物理的オブジェクトを、一定の規則にのっとって、一体化して表現します。管理されるオブジェクトは、クラスなどのオブジェクト指向の要素を使って表現されます。クラスには、データを記述するプロパティと、動作を記述するメソッドがあります。CIM は DMTF によって、オペレーティング システムとプラットフォームに依存しないように設計されています。WBEM テクノロジには、Microsoft Windows オペレーティング システム プラットフォームの拡張機能が含まれます。詳細については、DMTF Web サイトの DMTF CIM スキーマを参照してください。

CIM では、次の 3 つのレベルのクラスが定義されます。

- 管理のすべての領域に該当する、管理されるオブジェクトを表すクラス。これらのクラスでは、管理されるシステムを分析および記述するための基本的な語彙が提供されます。これらのクラスはコア モデルと呼ばれるものの一部です。
- 特定の管理領域に該当するが、特定の実装やテクノロジには依存しない、管理されるオブジェクトを表すクラス。これらのクラスは、共通モデルと呼ばれるものの一部です。
- 共通モデルに対し、テクノロジに特有なものとして追加される、管理されるオブジェクトを表すクラス。これらのクラスは一般的に特定のプラットフォーム (UNIX や Microsoft Win32 環境など) に該当し、拡張モデルと呼ばれます。

すべてのクラスは継承で関連付けることができます。継承では、子クラスが親クラスのデータとメソッドを含みます。継承による関係は、一般的にこれらの関係を使用する管理アプリケーションで確認されることはなく、アプリケーションでも継承の階層を認識する必要はありません。クラスの階層は、WMI Tools に付属のアプリケーションを使用して取得できます。詳細については、http://www.microsoft.com の WMI Tools を参照してください。

Windows Management では、関連クラスもサポートされています。関連クラスは 2 つの異なるクラスをリンクしてユーザ定義の関係を形成します。関連クラスは、管理アプリケーションで確認することができます。Windows Management では、関連クラスを定義してシステムのクラスをサポートします。サードパーティの開発者も、独自の管理環境のための関連クラスを定義することができます。

WBEM では、特定の管理環境で使用されるクラスやインスタンスをグループ化する、スキーマのコンセプトがサポートされています。 プラットフォーム SDK には、 CIM スキーマおよび Microsoft Win32 スキーマの 2 つのスキーマが含まれます。CIM スキーマには、CIM の最初の 2 つのレベルのクラス定義が含まれています。これらのクラスは、プラットフォームにかかわらず、すべての管理環境の一部である、管理されるオブジェクトを表します。Win32 スキーマには、一般的な Win32 環境の一部である、管理されるオブジェクトのクラス定義が含まれています。

CIM の詳細については、http://www.dmtf.org を参照してください。

制限と免責条項をすべてお読みください。

目次に戻る 先頭に戻る

目次に戻る

主な機能：Intel(R) PRO ネットワーク アダプタ WMI および CDM プロバイダ ユーザ ガイド

NCS WMI Provider の機能

WMI Provider の主な機能は次のとおりです。

アダプタに関する機能

- Intel(R) PROSet でサポートされているすべての物理アダプタの列挙
- インストールされているアダプタの設定の列挙
- インストールされているアダプタの設定の追加、削除、更新
- アダプタの物理デバイス情報の取得
- アダプタのシステム スロット デバイス情報の取得
- アダプタの IPv4 プロトコル設定の取得
- アダプタの Boot Agent と関連する設定の更新と変更
- アダプタのアンインストール

チームに関する機能

- Intel PROSet でサポートされているチームの列挙
- アダプタのチームの作成、削除
- チームの設定の追加、削除、更新
- チームのメンバー アダプタの追加、削除
- チームの IPv4 プロトコル設定の取得

VLAN に関する機能

- アダプタやチームの仮想 LAN の列挙
- 物理アダプタやアダプタのチームへの仮想 LAN の作成、削除
- VLAN の設定の追加、削除、更新
- チームの IPv4 プロトコル設定の取得

イベント通知機能

- クライアントの登録を許可
  - アダプタ ステータス イベント
  - アダプタ設定イベント
  - セッション イベント
  - チーム ステータス イベント
  - チーム設定イベント
  - VLAN 設定イベント

先頭に戻る

CDM Provider の機能

CDM Provider の主な機能は次のとおりです。

- 診断テストのタイプに依存しない、テストの実行と停止、およびテスト結果のクリア
- 汎用設定クラスの使用により、CDM ソフトウェアでは予期されないテストの制御を可能にする

- CDM Provider はアダプタのみに使用
- 汎用結果クラスの使用により、特定の結果メッセージを CDM Provider のコードから解放
- レジストリ エントリによって Provider の実行を制御
- テスト結果を結果ログ ファイルに書き込み

制限と免責条項をすべてお読みください。

目次に戻る 先頭に戻る

目次に戻る

インストールされるファイル：Intel(R) PRO ネットワーク アダプタ WMI および CDM プロバイダ ユーザ ガイド

WMI ファイル

実行可能ファイル

WMI Provider の実行可能ファイルは、次のとおりです。

- **NcsWmiCo.exe** - コア プロバイダ。IANet_NetService とコア イベント クラスを実装します。
- **NcsWmiIm.exe** - インスタンスとメソッドのプロバイダ。イーサネット アダプタ スキーマ、チーム化スキーマ、設定スキーマ、VLAN スキーマを実装します。
- **NcsWmiEv.exe** - イベント プロバイダ。アダプタ、チーム、VLAN のイベントを実装します。

MOF ファイル

言語に依存しないデータと、言語に特有なデータに対し、別々の MOF ファイルがあります。また、IntelNCS と CIMV2 ネームスペースに対し、別々の MOF ファイルがあります。詳細については、 ロケールとローカライゼーションおよび エラーのレポートを参照してください。

IntelNCS ネームスペースの MOF ファイルは次のとおりです。

- **NcsCmLn.mof** - NCS クラスが依存する CIM の基本クラス。
- **NcsCmEnu.mfl** - CIM 基本クラスのアメリカ英語版。
- **NcsCoLn.mof** - コア プロバイダで実装されるコア クラス。
- **NcsCoEnu.mfl** - コア クラスのアメリカ英語によるテキスト修正。
- **NcsIaLn.mof** - IEEE 802.3 アダプタ、チーム、VLAN のクラス。
- **NcsIaEnu.mfl** - 802.3 コア クラスのアメリカ英語によるテキスト修正。

CIMV2 ネームスペースの MOF ファイルは次のとおりです。

- **C2CmLn.mof** - NCS クラスが依存する CIM の基本クラス。
- **C2CmEnu.mfl** - CIM 基本クラスのアメリカ英語版。
- **C2CoLn.mof** - コア プロバイダで実装されるコア クラス。
- **C2CoEnu.mfl** - コア クラスのアメリカ英語によるテキスト修正。
- **C2IaLn.mof** - IEEE 802.3 アダプタ、チーム、VLAN のクラス。
- **C2IaEnu.mfl** - 802.3 コア クラスのアメリカ英語によるテキスト修正。

リソース ファイル

WMI Provider のリソース ファイルは次のとおりです。

- **ENU_8023.dll** - 英語 USA 8023 リソース。
- **ENU_NWRC.dll** - コア プロバイダの英語 USA WMI リソース。
- **ENU_NWR.dll** - 8023 プロバイダの英語 USA WMI リソース。

その他のローカライズされたリソース ファイルも、オンデマンドでロードできます。ローカライズされたリソース DLL の名前の一般的なパターンは「`_mwr.dll`」で、これがローカライゼーションの言語コードです。たとえば、標準フランス語は FRA です。

先頭に戻る

CDM Provider ファイル

実行可能ファイル

CDM Provider の実行可能ファイルは、次のとおりです。

- **Ncsdiag.exe** CDM 診断の主な実行可能ファイル。Microsoft* WMI インターフェイス仕様に準拠し、プロセス外の COM サーバとしてアクセスされます。
- Intel(R) PROSet ソフトウェア スタックのその他の実行可能ファイル

MOF ファイル

マスタ **.mof** ファイルは製品には付属していませんが、Microsoft* Windows* Management Instrumentation グローバル化モデルに基づいて、対応する言語に依存するコンポーネントや依存しないコンポーネントにコンパイルされています。詳細については、Microsoft* WMI SDK (プラットフォーム SDK のコンポーネント) の WMI ローカライゼーションの章を参照してください。特に、「*Compiling Localized MOF Files (*ローカライズされた *MOF* ファイルのコンパイル*)*」の項を注意して読んでください。

.mof ファイル (DNcsCdmN.mof) を削除すると、Intel から派生したクラス定義は削除されますが、DMTF で定義されたクラスは、削除してしまうとほかの既存のアプリケーションに障害が出る恐れがあるため、削除されません。

この CDM 実装は、典型的に CIMV2 ネームスペースに基づいて使用されます。IntelNCS ネームスペースの MOF ファイルは次のとおりです。

ファイル名	言語タイプ	説明
CdIa.mof	マスタ	Intel CDM 実装のクラス定義
CdIaLn.mof	言語に依存しない	Intel CDM 実装のクラス定義
CdIaEnu.mfl	英語に依存	Intel CDM 実装の言語拡張
CdCm.mof	マスタ	コア スーパーセット CDM クラスの定義
CdCmLn.mof	言語に依存しない	コア スーパーセット CDM クラスの定義
CdCmEnu.mfl	英語に依存	コア スーパーセット CDM クラス定義の言語拡張
DNcsCdmN.mof	該当せず	Intel CDM クラスを削除

CIMV2 ネームスペースの MOF ファイルは次のとおりです。

ファイル名	言語タイプ	説明
C2Icd.mof	マスタ	Intel CDM 実装のクラス定義
C2IcdLn.mof	言語に依存しない	Intel CDM 実装のクラス定義
C2IcdEnu.mfl	英語に依存	Intel CDM 実装の言語拡張
C2Cd.mof	マスタ	コア スーパーセット CDM クラスの定義
C2CdLn.mof	言語に依存しない	コア スーパーセット CDM クラスの定義
C2CdEnu.mfl	英語に依存	コア スーパーセット CDM クラス定義の言語拡張
DNcsCdm2.mof	該当せず	Intel CDM クラスを削除

注： ローカライゼーションでは、適切な言語に依存する **.mof** ファイルを追加する必要があります。

リソース ファイル

CDM Provider のリソース ファイルは次のとおりです。

- **ENU_Diag.dll** - Diagnostic Provider の英語 USA WMI リソース。

制限と免責条項をすべてお読みください。

目次に戻る 先頭に戻る

目次に戻る

セキュリティ：Intel(R) PRO ネットワーク アダプタ WMI および CDM プロバイダ ユーザ ガイド

WMI Provider と CDM Provider は、クライアントを擬人化してセキュリティを管理します。Provider へのすべての呼び出しは、クライアント自身のセキュリティ コンテキストで作成され、このコンテキストは下層レイヤへと渡されます。ターゲット マシンに管理権限がない場合、 1 つまたはすべての操作が失敗することがあります。

制限と免責条項をすべてお読みください。

目次に戻る 先頭に戻る

目次に戻る

ネームスペースとコンテキスト：Intel(R) PRO ネットワーク アダプタ WMI および CDM プロバイダ ユーザ ガイド

CIM クラスはネームスペース内に常駐します。Microsoft* の標準ネームスペースは **root/cimv2** と呼ばれ、CIM v2.2 または **root/default** に基づいています。WMI Provider と CDM Provider のクラスは、このネームスペースに追加できます。これらのプロバイダは CIM v2.5 に基づいています。このため、また、オブジェクトのキーの相違のため、両者のクラスは別のネームスペース、**root/IntelNCS** にあります。

WBEM コンテキスト

コンテキスト オブジェクトでは、WMI API メソッドにパラメータとして渡すことのできない追加情報が、Provider に提供されます。コンテキスト修飾子を登録するには、**IWbemContext** を使用します。コンテキスト オブジェクトのインターフェイス ポインタは、**IWbemServices** メソッドの最後のパラメータとして渡されます。

次の表に、Provider で使用されるコンテキスト修飾子 (指定された値) をあげます。SessionHandle などのほとんどの修飾子は、Provider の特定の領域の機能とともにのみ使用されますが、LocaleID、MachineName、ApplicationName は、すべての **IWbemServices** 呼び出しに設定できます。

Provider にコンテキストが渡されない場合は、Provider への **Initialize** 呼び出しで渡された LocaleID が使用されます。コンテキストを使用して行われるすべての読み取りでは、書き込みが実行されるまで、現在の設定が読み取られます。書き込み後の読み取りでは、書き込みが成功した後のシステムの状態が表示されます。読み取りには、NULL コンテキストを使用することができます。

コンテキスト修飾子	バリアントのタイプ	説明
SessionHandle	VT_BSTR	アプリケーションの IANet ネットワーク クラスを識別します。アプリケーションは、まずセッション ハンドルを確立しない限り、クラスやクラスの属性に変更を加えることはできません。セッション ハンドルの確立と使用については、IANet_NetService クラスの項を参照してください。この修飾子は、アプリケーションでクラスからデータを読み取るだけの場合は必要ありません。セッション ハンドルにより、NCS ソフトウェアが設定への複数の同時アクセスを管理できるようになり、1 人のユーザがほかのユーザをロックアウトしなくてもすむようになります。各セッションには変更された内容を保存するために、それぞれ別のキャッシュがあります。複数のユーザが同時に変更を行っている場合は、最初に変更を適用したユーザの変更が有効になります。その他のすべてのユーザのキャッシュは無効になります。
LocaleID	VT_BSTR	Microsoft のロケールの ID。これは、アプリケーションが Provider からローカライズされたテキスト文字列を必要とする場合に必要です。必須の LocalID が使用されない限り、すべてのエラーメッセージと警告は英語で表示されます。
ApplicationName	VT_BSTR	呼び出しを行ったアプリケーションの名前。これはログ記録に必要です。
MachineName	VT_BSTR	Provider に接続中のマシンの名前。これはログ記録に必要です。
PreCheck	VT_BOOL	このブール値は、クライアントが実際に操作を行う前に、操作が許可されているかどうかを確認しようとしていることを、Provider に告げるために使用されます。たとえば、チームへのアダプタの追加などです。 値： • TRUE = 操作が許可されていない場合、Provider は操作を実行せず、エラー コードと詳細なステータスを返します。 • FALSE = Provider は操作を実行します。 この修飾子がない場合は、属性が FALSE と同じ効果をもちます。

WarningErrorCode	VT_I4	一部の操作では、ユーザに警告を送信する必要の生じることがあります。たとえば、チームへアダプタを追加する際に、場合によってはチームを再ロードしなければならないことがあります。WMI では、このためのメカニズムは提供されていません。この修飾子が存在し、設定がゼロでない場合、操作が完了し、関連する警告がある際には、Provider から E_FAIL が返されます。クライアントで詳細ステータスを使って、警告のテキストを取得します。

制限と免責条項をすべてお読みください。

目次に戻る 先頭に戻る

目次に戻る

ロケールとローカライゼーション：Intel(R) PRO ネットワーク アダプタ WMI および CDM プロバイダ ユーザ ガイド

WMI Provider と CDM Provider のローカライゼーションには、2 つ要素があります。ローカライズされた MOF ファイルと、ローカライズされた属性データです。

ローカライズされた MOF ファイル

Provider で使用されるすべての MOF ファイルは、Microsoft Windows* Management Instrumentation (WMI) グローバル化モデルに基づいて、ローカライズされます。このために、各クラス定義は次のように分類されます。

- **.mof** ファイル：基本的なクラスの定義のみを含む、言語に依存しないバージョン。
- 対応する **.mfl** ファイル：ローカライズされた情報を含む、特定の言語のためのバージョン。ロケールに特有のプロパティの説明などです。

サポートされている言語

中国語 (台湾)
中国語 (中華人民共和国)
デンマーク語
オランダ語 (オランダ)
英語 (アメリカ合衆国)
フィンランド語
フランス語 (フランス)
ドイツ語 (ドイツ)
イタリア語 (イタリア)
日本語
ノルウェー語 (ブークモール)
ポルトガル語 (ブラジル)
スペイン語 (スペイン - 近代)
スウェーデン語

クラスの保管

言語に特定のクラス定義は、言語に依存しない基本クラス定義を含むネームスペースの下の、子サブネームスペースに保管されます。たとえば、WMI Provider と CDM Provider では、英語のロケールのために、子ネームスペース **ms_409** が **root/IntelIncs** ネームスペースの下に存在します。同様に、**root/IntelIncs** ネームスペースの下に、サポートされている各言語の子サブネームスペースがあります。

CIMV2 ネームスペースにおけるローカライズされた MOF のサポート

root/cimv2 ネームスペースでは、Provider のクラス (IANet_ classes) は WMI でこのネームスペースに追加された基本クラスから派生します。基本クラスの言語に特有なクラス定義を含むサブネームスペースは、**root/cimv2** ネームスペースの下にあらかじめ存在します。この既存の子ネームスペースに、IA_Net の言語に特定のクラス定義が追加されます。この基本クラスへの依存性のため、MOF のローカライズは、デフォルトのシステム ロケールでのみ行われます。

ランタイムのサポート

ローカライズされたデータを取得するために、WMI アプリケーションでは **SWbemLocator::ConnectServer** と **IWbemLocator::ConnectServer** の呼び出しで、strLocale パラメータを使用してロケールを指定できます。ロケールが指定されない場合は、システムのデフォルトのロケールが使用されます。(たとえば、アメリカ英語では MS_409。)このロケールは、英語の文字列に追加する際に、適切なネームスペースを選択するのに使用されます。

さらに、**IWbemServices::GetObject**、**SwbemServices.GetObject**、**IWbemServices:: ExecQuery**、および **SWbemServices.ExecQuery** は、WBEM_FLAG_USE_AMENDED_QUALIFIERS フラッグを指定して、ローカライズされたデータおよび基本的な定義を要求する必要があります。これは、値のマップ、表示の記述、またはその他の MOF ファイルからの修正修飾子を使用する、表示可能な値を生成するすべての機能で必須です。

先頭に戻る

ローカライズされた属性データ

エラー メッセージなどの、ローカライズされた属性データを取得するために、Provider はすべての呼び出しに対して、呼び出し機のロケールを知ることが必要です。これが適切に機能するためには、クライアントがすべての呼び出しに渡されるコンテキスト オブジェクトにロケールを追加することが必要です (WBEM コンテキストのネームスペースとコンテキストを参照)。Provider がローカライズ可能な文字列を返す場合、クライアントのロケールに対応したリソース DLL のロードが試みられます。対応するリソース DLL がない場合は、アメリカ英語の文字列が返されます。

制限と免責条項をすべてお読みください。

目次に戻る 先頭に戻る

目次に戻る

エラーのレポート：Intel(R) PRO ネットワーク アダプタ WMI および CDM プロバイダ ユーザ ガイド

概要

この項では、IANet_ExtendedStatus に関し、WMI Provider や CDM Provider で生成されるエラーの処理方法について説明します。エラー オブジェクトが返される方法や状況は、呼び出しが同時性、半同時性、非同時性のいずれかによって異なります。場合によっては、エラーが発生すると HRESULT が WBEM_E_FAILED に設定されます。ただし、この時点ではエラーを生成したのが WMI か Provider かは不明です。

同時性呼び出しのエラー オブジェクトを取得するには、GetErrorInfo() を使用して IErrorInfo object を取得します。QueryInterface() を使用して、エラー情報を含む IWbemClassObject を取得します。

非同時性呼び出しのエラー オブジェクトを取得するには、IWbemClassObject を最後の SetStatus() 呼び出しの最後のアイテムとして渡します。エラー オブジェクトのインスタンスを取得したら、__Class プロパティをチェックして、エラーの原因を判定します。IANet_ クラスに関連するエラーでは、WMI で __ExtendedStatus のインスタンスが、Provider で IANet_ExtendedStatus のインスタンスが作成されます。IANet_ExtendedStatus は __ExtendedStatus から派生し、次のエラー オブジェクト修飾子を含みます。

- Description - 現在のロケールにあわせたエラーの説明
- File - エラーが発生したコード ファイル
- Line - コード ファイル内でエラーのある行の番号
- ParameterInfo - エラーが発生した時点で使用されていたクラスまたは属性
- Operation - エラーが発生した際に処理中だった操作
- ProviderName - エラーを引き起こした Provider の名前
- StatusCode - 失敗した内部呼び出しから返されたコード
- SessionHandle - 処理に使用されたセッション ハンドル
- RuleFailureReasons - 処理が失敗した理由処理は、技術的な規則が失敗したために、失敗することがあります。たとえば、チームによっては、管理アダプタが必要な場合があります。

先頭に戻る

エラー コード

すべてのエラー コードに対し、Provider からロケールに合わせてカスタマイズされた説明が提供されます。エラー コードは HRESULT の形式で、重要度 1、機能が ITF に設定された状態になります。アプリケーションでは、次のコードに基づいて普及を行うことができます。

- 0x80040901 - "WMI: Put property failed"
- 0x80040902 - "WMI: No class object"
- 0x80040903 - "WMI: Failed to create class"
- 0x80040904 - "WMI: Failed to spawn instance of class"
- 0x80040905 - "WMI: Failed to create safe array"
- 0x80040906 - "WMI: Failed to put safe array"
- 0x80040907 - "WMI: Failed to return object to WMI"
- 0x80040908 - "WMI: Get property failed"
- 0x80040909 - "WMI: Unexpected type while getting property"
- 0x8004090A - "WMI: Class not implemented by this provider"
- 0x8004090B - "WMI: Unable to parse WQL statement"
- 0x8004090C - "WMI: Providers only support WQL"
- 0x8004090D - "WMI:0x8004090D - "WMI: Parameter in context has the wrong type"

- 0x8004090E - "WMI: Error formatting debug log"
- 0x8004090F - "WMI: bad object path"
- 0x80040910 - "WMI: Failed to update setting"
- 0x80040911 - "WMI: Null parameter passed to method"
- 0x80040912 - "Setting value too small."
- 0x80040913 - "Setting value too big."
- 0x80040914 - "Setting not in step"
- 0x80040915 - "String setting is too long"
- 0x80040916 - "Setting is not one of the allowed values"
- 0x80040917 - "WMI: Qualifier not found"
- 0x80040918 - "WMI: Qualifer set not found"
- 0x80040919 - "WMI: Safe array access failed"
- 0x8004091A - "WMI: Unhandled exception"
- 0x8004091B - "WMI:0x8004091B - "WMI: Operation is not supported for this class"
- 0x8004091C -"WMI: Unexpected event class"
- 0x8004091D -"WMI: Bad event data"
- 0x8004091E -"WMI: Operation succeeded with warnings"
- 0x8004081F -"WMI:0x8004081F - "WMI: The NCS Service has been stopped."

- 0x80040801 -"EAL: Internal exception"
- 0x80040802 -"EAL: General failure"
- 0x80040803 -"EAL: Not initialized"
- 0x80040804 -"EAL: Failed to initialize."
- 0x80040805 -"EAL: Session limits exceeded"
- 0x80040806 -"EAL: Out of memory"
- 0x80040807 -"EAL: Rule syntax error"
- 0x80040808 -"EAL: Unexpected end of list"
- 0x80040809 -"EAL: Rule link error"
- 0x8004080A -"EAL: Device Creation Failed"
- 0x8004080B -"EAL: Media service not found"
- 0x8004080C -"EAL: Device service not found"
- 0x8004080D -"EAL: PCI bus module not found"
- 0x8004080E - "EAL: Adapter is a member of a team"
- 0x8004080F -"EAL: Rule Access Point creation error"
- 0x80040810 -"EAL: Registry key error"
- 0x80040811 - "EAL: Registry XML file path error"
- 0x80040812 - "EAL: Unknown event class"
- 0x80040813 - "EAL: Unknown module id"
- 0x80040814 - "EAL: Rule service not found"
- 0x80040815 - "EAL: NULL input pointer"
- 0x80040816 - "EAL: Rule grammar error"
- 0x80040817 - "EAL: Rule failed"
- 0x80040818 - "EAL: Setting is already grouped"

- 0x80040220 - "Sync Layer: Team removal failed."
- 0x80040221 - "Sync Layer: Vlan creation failed."
- 0x80040222 - "Sync Layer: Vlan removal failed."
- 0x80040223 - "Sync Layer: Adapter removal failed."
- 0x80040224 - "Sync Layer: Setting Change/Creation/Removal failed."
- 0x80040225 - "Sync Layer: Parameter Change/Removal failed."
- 0x80040226 - "Sync Layer: NetConfig subsystem locked. "
- 0x80040227 - "Sync Layer: System Update In Progress. Please try again later."
- 0x80040228 - "Sync Layer: Adapter is Locked"
- 0x80040229 - "Sync Layer: Flash read failed."
- 0x8004022A - "Sync Layer:"

- 0x80040210 - "Sync Layer: Invalid event offset"
- 0x80040211 - "Sync Layer: Invalid input"
- 0x80040212 - "Sync Layer: Invalid key"
- 0x80040213 - "Sync Layer: Adapter not team member"

0x80040214 - "Sync Layer: Driver not loaded"
- 0x80040215 - "Sync Layer: Client impersonation failed"
- 0x80040216 - "Sync Layer: Caught exception"
- 0x80040217 - "Sync Layer: Session not locked"
- 0x80040218 - "Sync Layer: Hardware access layer is not available"
- 0x80040219 - "Sync Layer: Flash not available"
- 0x8004021A - "Sync Layer: Diagnostics not supported"
- 0x8004021B - "Sync Layer: Diagnostic test not running"
- 0x8004021C - "Sync Layer: Boot Agent update not available"
- 0x8004021D - "Sync Layer: Boot Agent corrupted."
- 0x8004021E - "Sync Layer: Flash write failed."
- 0x8004021F - "Sync Layer: Team creation failed."
- 0x80040201 - "Sync Layer: Initialization failed"
- 0x80040202 - "Sync Layer: Invalid initialization handle"
- 0x80040203 - "Sync Layer: Session handle already exists"
- 0x80040204 - "Sync Layer: Invalid session handle"
- 0x80040205 - "Sync Layer: The maximum number of sessions has been reached."
- 0x80040206 - "Sync Layer: The session lock handle already exists"
- 0x80040207 - "Sync Layer: Invalid session lock handle"
- 0x80040208 - "Sync Layer: Session already locked"
- 0x80040209 - "Sync Layer: Invalid media service module Id"
- 0x8004020A - "Sync Layer: Invalid Advanced Service Module Id"
- 0x8004020B - "Sync Layer: Invalid device service module Id"
- 0x8004020C - "Sync Layer: Invalid component type Id"
- 0x8004020D - "Sync Layer: Invalid bus interface module Id"
- 0x8004020E - "Sync Layer: Invalid sink window handle"
- 0x8004020F - "Sync Layer: Invalid event Id"

- 0x80040401 - "HAM PCI: Invalid memory map address"
- 0x80040402 - "HAM PCI: Configuration driver failed to load"
- 0x80040403 - "HAM PCI: Configuration driver version mismatch"
- 0x80040404 - "HAM PCI: Device slot not found"
- 0x80040405 - "HAM PCI: Diagnostic driver failed to load"
- 0x80040406 - "HAM PCI: Diagnostic driver version mismatch"
- 0x80040407 - "HAM PCI: Diagnostic driver initialization failed"
- 0x80040408 - "HAM PCI: Diagnostics not initialized"
- 0x80040409 - "HAM PCI: Diagnostics already initialized"
- 0x8004040A - "HAM PCI: Diagnostic test already running"
- 0x8004040B - "HAM PCI: Diagnostic test not running"
- 0x8004040C - "HAM PCI: Diagnostic test terminated"
- 0x8004040D - "HAM PCI: Diagnostic Invalid test number"
- 0x8004040E - "HAM PCI: Diagnostic hardware missing"
- 0x8004040F - "HAM PCI: Diagnostic send receive initialization failed"

- 0x80040511 - "Media Service: NDIS IO call failed"
- 0x80040512 - "Media Service: Miniport not loaded"
- 0x8004051B - "Media Service: Invalid device handle"
- 0x8004051C - "Media Service: Invalid adapter handle"
- 0x8004051D - "Media Service: Invalid team handle"
- 0x8004051E - "Media Service: Invalid VLAN handle"
- 0x8004051F - "Media Service: Device missing"
- 0x80040520 - "Media Service: Invalid setting type"
- 0x80040521 - "Media Service: Unknown invalid object"
- 0x80040522 - "Media Service: Invalid Setting Handle"
- 0x80040523 - "Media Service: Invalid Team Mode"
- 0x80040525 - "Media Service: Setting Already Exists"

- 0x80042001 - "RAP: Already initialized"
- 0x80042002 - "RAP: Invalid XML file"
- 0x80042003 - "RAP: XML load error"

0x80042004 - "RAP: Not initialized"

- 0x80042005 - "RAP: Rule not extracted before"
- 0x80042006 - "RAP: Conditions count mismatch"
- 0x80042007 - "RAP: Results apply error"
- 0x80042008 - "RAP: Invalid rule"
- 0x80042009 - "RAP: Node not found"
- 0x8004200A - "RAP: Error no single node"
- 0x8004200B - "RAP: No action rule"
- 0x8004200C - "RAP: Zero condition"
- 0x8004200D - "RAP: Zero action"
- 0x8004200E - "RAP: XML Decode error"

制限と免責条項をすべてお読みください。

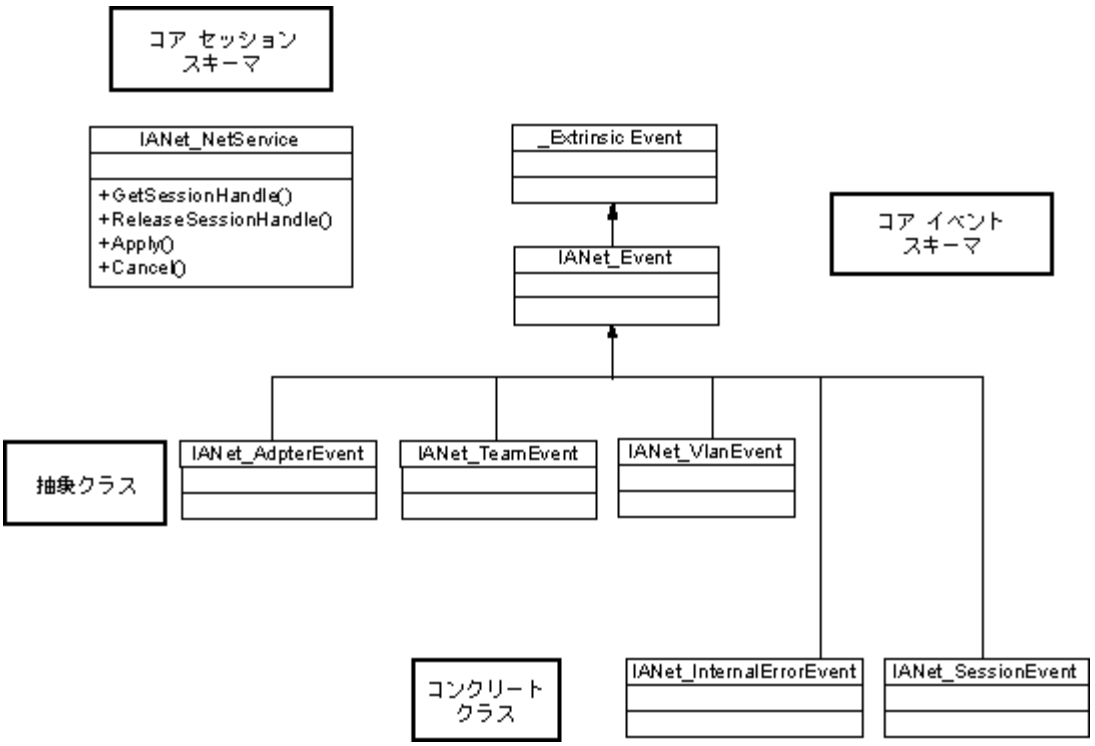
目次に戻る 先頭に戻る

目次に戻る

コア スキーマ：Intel(R) PRO ネットワーク アダプタ WMI および CDM プロバイダ ユーザ ガイド

概要

コア スキーマは、IANet_NetService クラスとコア イベント クラスで構成されます。



先頭に戻る

IANet_NetService

用途
IANet_NetService クラスは、IANet_ schema のルート オブジェクトです。このクラスにより、クライアントがセットを実行するのに必要なセッションにアクセスできるようになります。

インスタンス
このオブジェクトのインスタンスは 1 つです。クライアントでは、このクラスに使用されているキーに依存しないようにします。代わりに、IANet_NetService のすべてのインスタンスを列挙して、クラスのインスタンスを取得します。

インスタンスの作成
IANet_NetService のインスタンスを作成することはできません。

インスタンスの削除

IANet_NetService のインスタンスを削除することはできません。

プロパティの変更
このクラスには、ユーザが変更できるプロパティはありません。

サポートされている属性
このクラスでは、次の 2 つのクラスが実装されます。

- Version - コア プロバイダの現在のバージョンを格納する。
- InstallDate - プロバイダがインストールされた日付を格納する。

メソッド
セッションの管理には、次のメソッドを使用できます。

- **void GetSessionHandle(([OUT] string SessionHandle, [out] uint32 ActiveSessions)** - セッション ハンドル文字列の設定に使用します。セッション ハンドル文字列は、 SessionHandle 修飾子のコンテキスト オブジェクトに配置します。ActiveSessions では、このシステムのアクティブなセッションの数が返されます。これにより、クライアントが別のユーザがネットワーク設定を変更している可能性のある場合に、警告を発することができます。
- **void Apply([IN] string sSessionHandle, [OUT] uint32 FollowupAction);** - 特定のセッション ハンドルで行われた変更を適用します。返された uint32 引数は、WMI と CDM Provider で、アプリケーションに変更を適用するにはサーバを再起動する必要があることを告げるために使用されます。これは、Win32_OperatingSystem クラスで **Reboot** メソッドを呼び出して行うことができます。

  値：
  1 = システムの再起動が必要
  0 = 再起動は必要なし

- **void ReleaseSessionHandle ([IN] string SessionHandle)** - セッション ハンドルを使用後に解放します。このセッションで行った変更は失われます。この呼び出しの後はセッション ハンドルは無効となり、使用することはできません。
- **void Cancel([IN] string SessionHandle);** - セッションをキャンセルします。内部キャッシュはクリアされ、この呼び出しの後に読み込まれたすべてのデータは現在の設定を示します。

先頭に戻る

コア イベント

IANet_SessionEvent

用途
このイベントは、クライアントに NCS セッション API の使用を通知するために使用されます。クライアントはこのイベントを使用して、ほかのクライアントがセッションを作成または使用している際に通知を受けることができます。

トリガ
このイベントは、クライアントがセッションを作成または削除した場合や、セッションの **Apply** を呼び出した際にトリガされます。

イベント データ
EventType では、次の値のいずれかをとることができます。

- [New session (新規セッション)]は、当該のクライアント、またはその他のクライアントによって、新規セッションが作成されたことを示します。
- [End sesshon (セッションの終了)] は、クライアントがセッションを終了したことを示します。セッションを終了したのは、当該クライアントと、その他のクライアントの場合があります。
- [Cache invalidated (キャッシュ無効)] は、セッションで別のクライアントが **[Apply** (適用)] を呼び出したことを示します。すべてのほかのセッションは無効となり、セッションに関連するキャッシュは削除されています。
- [Configuration changed (設定の変更)] は、セッションの設定が変更されていることを示します。

SessionHandle には、イベントをトリガしたセッション ハンドルが含まれます。

OpenSessions には、オープンなセッションの数が含まれます。このデータ アイテムは [Cache invalidated] と [Configuration changed] のイベントでは NULL となります。

IANet_InternalErrorEvent

用途
このイベントは、イベント プロバイダで内部エラーが発生したことをクライアントに通知するために使用されます。場合によっては、イベント プロバイダがその後のイベントをレポートできないこともあります。

トリガ
このイベントは、次の場合に発生します。

- イベント プロバイダがイベント ソースから不明のイベントを受信した後。
- イベントを提供するソフトウェアがシャットダウンした後。
- イベント プロバイダがイベントを受信し、イベント ソースがそのイベントについてそれ以上のデータを取得できない場合。

イベント データ
EventType は、次のいずれかをとることができます。

- [Could not get event data (イベント データの取得不能)]：イベントが発生し、イベント ソースがそのイベントについて詳細な情報を取得できない場合です。
- [Event source has shut down (イベント ソースのシャット ダウン)]：イベントのデータ ソースがシャットダウンした場合です。この場合、イベント プロバイダもシャットダウンされ、ソースを再起動し、新しい通知クエリを作成するまで、新しいイベントは生成されません。
- [Unexpected message (予期しないメッセージ)]：イベント プロバイダが予期しないイベントを受信した場合です。

先頭に戻る

使用例

設定を変更するには、セッション ハンドルが必要です。セッション ハンドルにより、NCS ソフトウェアが設定への複数の同時アクセスを管理できるようになり、セッションがほかのユーザをロックアウトしなくてもすむようになります。各セッションには変更された内容を保存するために、それぞれ別のキャッシュがあります。複数のセッションで同時に変更が行われている場合は、最初に変更を適用するセッションで変更が有効になります。その他のすべてのセッションのキャッシュは無効になります。

セッション ハンドルの取得

セッション ハンドルにアクセスするには、まずクライアントが IANet_NetService の 1 つのインスタンスへのオブジェクト パスを取得することが必要です。**IWbemServices::CreateInstanceEnum** を呼び出して、クラス名 "IANet_NetService" を渡します。これは、**SELECT * FROM IANet_NetService** というクエリで **IWbemServices::ExecQuery** を呼び出すのと同じです。設定に変更を行うには、まずクライアントでセッション ハンドルを取得します。**GetSesssionHandle** メソッドを使用して、新しいセッションを開始します。

クライアントでは **IWbemServices::ExecMethod** を使用して CIM オブジェクトにメソッドを実行することができ、IANet_NetService のインスタンスの **__PATH** 属性からオブジェクト パスが必要になります。このメソッドでは、現在アクティブなセッションの数も返されます。NCS (Network Configuration Service) へのアクセスが独占でない場合、クライアントでは警告を発生する必要のある場合があります。

IWbemContext オブジェクトでのセッション ハンドルの使用

クライアントでセッション ハンドルを取得したら、IWbemContext オブジェクトを作成します。このオブジェクトの **SessionHandle** 修飾子にセッション ハンドルを保管します。この COM オブジェクトへのポインタは、IWbemServices へのすべての呼び出しに渡すようにします。IANet_NetService オブジェクトでは、セッション ハンドルを明示的な引数としてとるため、このオブジェクトへのアクセスの呼び出し時にはセッション ハンドルは必要ありません。

セッション ハンドルを使用した保留中の変更の読み取り

設定を読み取る場合、コンテキストにセッション ハンドルを渡すと、プロバイダから、保留中の変更がすでに適用されたかのように設定が返されます。つまり、アンインストールされたアダプタは欠けている状態になり、変更した設定では新しい値が返されます。

ただし、一部のオブジェクトは **[Apply** (適用)] を呼び出すまで表示されません (たとえば、IANet_IPProtcolEndpoints はプロトコルが適切なミニポートにバインドされるまで作成されません)。

セッション ハンドルでの終了

設定を変更したら、**Apply** メソッドを呼び出して、変更をコミットします。これにより、フォローアップ アクション コード (変更を適用するにはシステムを再起動する必要があるというメッセージ) が返されることがあります。

セッションが終了したら、必ず **ReleaseSessionHandle** を呼び出します。呼び出しを行わないと、行ったすべての変更は破棄されます。**Cancel** メソッドを呼び出した場合も、変更がすべて破棄されますが、クライアントでは同じセッション ハンドルが、新しく作成されたのと同じ状態で引き続き使用されます。

コア イベントの登録

アプリケーションは、**IWbemServices::ExecNotificationQuery** または **IWbemServices:: ExecNotificationQueryAsync** を使用して、イベントの通知を要求します。次に、イベント通知クエリの例をあげます。ただし、使用可能なクエリは、これ以外にも多数あります。

- **SELECT * FROM IANet_Event** - すべてのイベントを要求します。
- **SELECT * FROM IANet_SessionEvent** - すべてのセッション イベントを要求します。
- **SELECT * FROM IANet_InternalErrorEvent** - すべての内部イベントを要求します。

制限と免責条項をすべてお読みください。

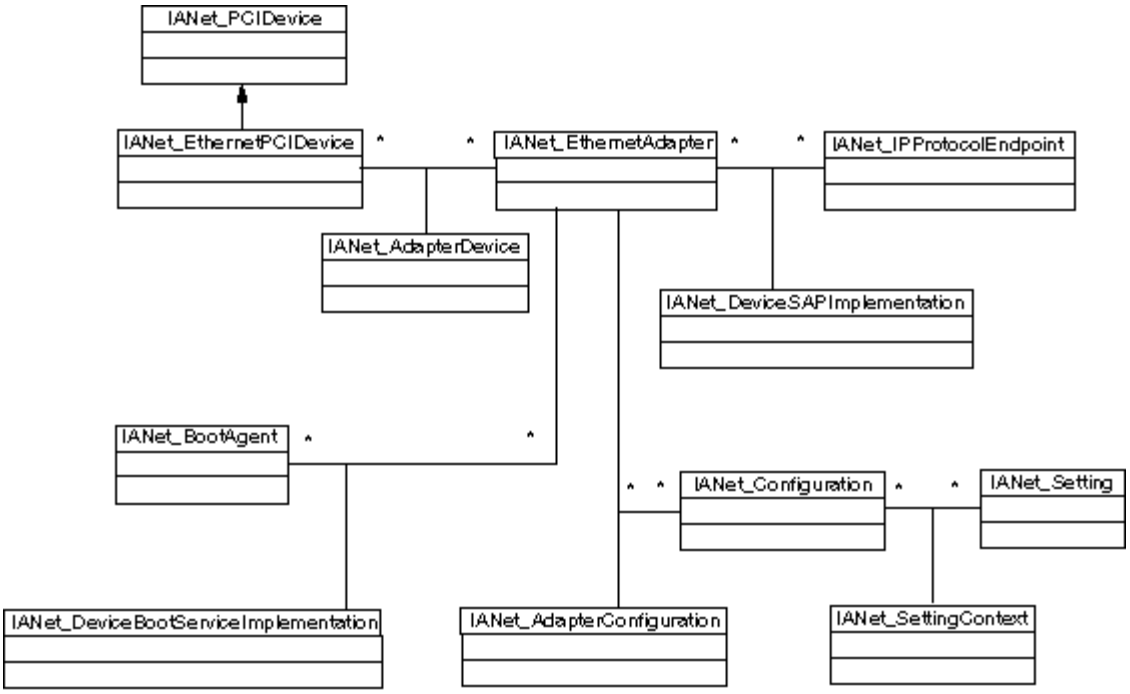
目次に戻る 先頭に戻る

目次に戻る

イーサネット アダプタ スキーマ：Intel(R) PRO ネットワーク アダプタ WMI および CDM プロバイダ ユーザ ガイド

概要

アダプタ スキーマは、設定可能なさまざまな Intel(R) PROSet イーサネット アダプタの設定に使用されます。このスキーマは、CIM v2.5 スキーマに基づいています。



先頭に戻る

IANet_EthernetAdapter

用途
IANet_EthernetAdapter は、インストールされているすべての Intel PRO ネットワーク アダプタと、インテルの中間アダプタを使用してチームを形成することのできるすべてのその他のアダプタの機能とステータスを定義します。このクラスは CIMv2.5 で定義される CIM_EthernetAdapter スーパークラスから派生します。CIM_EthernetAdapter は抽象クラスで、PermanentAddress、CurrentAddress、Speed of operation などの一般的なネットワーク ハードウェアのコンセプトを定義します。

インスタンス
このクラスのインスタンスは、次のものそれぞれに 1 つずつ存在します。

サポートされ、インストールされている Intel NIC
- インテルのマルチベンダー チームに参加できるサードパーティ製の NIC
- 作成された Intel アダプタ チーム

インスタンスの作成
IANet_EthernetAdapter のインスタンスを作成することはできません。

インスタンスの削除
IANet_EthernetAdapter のインスタンスを削除すると、物理アダプタがアンインストールされます。この方法でアンインストールできるのは、仮想アダプタではない、インテルのアダプタのみです。この操作には、セッション ハンドルが必要です。

プロパティの変更
このクラスには、ユーザが変更できるプロパティはありません。

サポートされていない属性
次の属性は Intel PROSet には必要でないため、サポートされていません。

- AutoSense (設定として露出)
- ErrorCleared
- OtherIdentifyingInfo
- IdentifyingDescriptions
- InstallDate
- LastErrorCode
- MaxDataSize
- MaxQuiesceTime
- PowerManagementCapabilities (メソッドとして露出)
- PowerManagementSupported (メソッドとして露出)
- PowerOnHours
- ShortFramesReceived
- SymbolErrors
- TotalPowerOnHours

メソッド
このクラスのインスタンスでは、次のメソッドがサポートされます。

- **IdentifyAdapter** アダプタのランプを数秒間点滅して、アダプタを識別します。このメソッドは、物理アダプタでのみ機能します。
- **HasVLANs** - このアダプタの VLAN の数を返します。
- **IsPowerMgmtSupported** - アダプタで電源管理がサポートされているかどうかを示します。
- **GetPowerUsage** - アダプタの全体的な電力使用量を検出します。
  0 = 通常電力
  1 = 低電力
- **SetPowerUsage** - アダプタの全体的な電力使用量を低減します。
  電力使用の設定は、システムの再起動やドライバの再ロードの際には維持されません。システムの再起動やドライバの再ロードを行うと、アダプタは自動的に通常の電力使用に戻ります。
- **GetPowerUsageOptions** - オプションの電力使用設定を検出します。スタンバイ時の電力使用や、バッテリーの使用などです。
- **SetPowerUsageOptions** - 電力使用オプションを変更します。スタンバイ時の電力使用を低減するためにメソッドを使用可能にする、バッテリーの使用などです。
  注： 電力使用の設定は保存され、再起動を行っても維持されます。
- **TestCable** - 特定のアダプタに診断テストを実行します。エラーが発生した際には、このメソッドによって考えられる問題、原因、解決策が返されます。
- **AdvancedTestCable** - 特定のアダプタに詳細なケーブル テストを実行します。このテスト スイートは、1000 Mbps アダプタで使用できます。このメソッドでは、各テストの名前と結果が返されます。
  注： **SpeedDuplex** が **Auto Negotiate** に設定されていないと、リンク エラーが発生する場合があります。このインスタンスでは、**SpeedAndDuplexNotAutomatic** 出力パラメータは TRUE です。
- **TestLinkSpeed** - アダプタが最大速度で実行しているかどうかを決定します。アダプタが 1 ギガビット未満を告知している場合は、このメソッドで考えられる原因が返されます。たとえば、「Link partner is not capable of running at 1000 Mbps (リンク パートナーが 1000 Mbps で実行不能)」などです。

先頭に戻る

IANet_IPProtocolEndpoint

用途
このクラスは、システムのプロトコル エンドポイントの IP 設定を記述するために使用されます。WMI Provider では、このほかのネットワーク プロトコルに関する情報は提供されません。このクラスは抽象クラス CIM_IPProtocolEndpoint から派生します。WMI Provider では、Intel PROSet で管理されるエンティティに関連するプロトコル情報のみが提供されます。

インスタンス
IANet_IPProtocolEndpoint のインスタンスは、IP プロトコル スタックからインテルでサポートされているエンドポイント (Intel アダプタや、Intel チームに参加できるアダプタや VLAN) への各バインドごとに存在します。チームに参加している一部のアダプタは、独自の IP アドレスを持たず、対応するアダプタのインスタンスに IANet_IPProtocolEndpoint が直接関連付けられてないことがあります。 IANet_IPProtocolEndpoint は、オペレーティング システムによってプロトコルがアダプタや VLAN にバインドされた後ではじめて存在します。 アダプタによっては複数の IP アドレスを持つものもありますが、これらのアドレスはすべて 1 つの IP プロトコル エンドポイント インスタンスに関連付けられます。このような高度な使用方法は Intel PROSet では必要なく、使用されることもないため、Provider ではサポートされていません。

インスタンスの作成
IANet_IPProtocolEndpoint のインスタンスを作成することはできません。インスタンスは、オペレーティング システムによってプロトコルがエンドポイントにバインドされた場合にのみ存在します。

インスタンス削除
IANet_IPProtocolEndpoint のインスタンスを削除することはできません。

プロパティの変更
このクラスには、ユーザが変更できるプロパティはありません。

関連
IANet_AdapterProtocolImplementation のインスタンスは、IANet_EthernetAdapter と IANet_IPProtocolEndpoint を関連付けるのに使用されます。IANet_VLANProtocolDependency のインスタンスは、VLAN を IANet_IPProtocolEndpoint に関連付けるのに使用されます。

注： チームは、チームの仮想アダプタを表すアダプタを通して、エンドポイントに関連付けられます。

サポートされている属性
Intel PROSet では、次の読み取り専用の属性が必要になります。

- Address
- AddressType
- DefaultGateway
- DHCPServerAddress
- DHCPAutoAssign
- IPVersionSupport
- SubnetMask

サポートされていない属性
次の属性は Intel PROSet には必要でないため、サポートされていません。

- Caption
- Description
- InstallDate
- NameFormat
- OtherTypeInformation
- ProtocolType
- Status

メソッド
なし

先頭に戻る

IANet_BootAgent

用途
このクラスは、アダプタのネットワーク起動機能に関する情報を取得するのに使用されます。たとえば、一部のインテルのアダプタでサポートされている PXE Boot Agent 設定などです。このクラスは CIM_BootService から派生します。

インスタンス
IANet_BootAgent のインスタンスは、Boot Agent 機能をサポートする各アダプタに対し 1 つずつ存在します。Boot Agent がインストールされていない場合でも同様です。

インスタンスの作成
IANet_BootAgent のインスタンスを作成することはできません。インスタンスは、アダプタが Boot Agent の機能をサポートする場合にのみ存在します。

インスタンスの削除
IANet_Setting のインスタンスを削除することはできません。

プロパティの変更
このクラスには、ユーザが変更できるプロパティはありません。

関連
IANet_DeviceBootServiceImplementation のインスタンスは、アダプタが Boot Agent をサポートする場合に、IANet_EthernetAdapter を IANet_BootAgent に関連付けるために使用されます。

サポートされている属性
Intel PROSet では、次の読み取り専用の属性が必要になります。

- InvalidImageSignature
- Version
- UpdateAvailable
- FlashImageType

サポートされていない属性
次の属性は Intel PROSet には必要でないため、サポートされていません。

- Caption
- Description
- InstallDate
- Started
- StartMode
- Status

メソッド
このクラスの次のメソッドを使用して、NIC の Flash ROM を更新できます。

uint32 ProgramFlash([IN, ValueMap {"0","1"} , Values {"Check Version", "Write Flash"}:Amended] uint32 Action, [IN] uint8 NewFlashData[], [OUT] string strErrorMessage);	このメソッドは、NIC の Flash ROM の更新に使用されます。これにより、フラッシュの更新中は NIC とネットワークの通信が停止します。
uint32 ReadFlash([OUT] uint8 FlashData[]);	このメソッドは NIC の Flash ROM を読み取ります。

先頭に戻る

IANet_PCIDevice

用途
このクラスは、システムのネットワーク デバイスの PCI デバイスのプロパティを記述するために使用されます。このクラスは CIM_PCIDevice から派生します。

インスタンス
このクラスのインスタンスは、システム内のネットワーク デバイスである各 PCI カードに 1 つ存在します。IA64 では、Intel PROSet でサポートされている PCI デバイスのみがインスタンスを持ちます。

インスタンスの作成
IANet_PCIDevice のインスタンスを作成することはできません。

インスタンスの削除
IANet_PCIDevice のインスタンスを削除することはできません。

プロパティの変更
このクラスには、ユーザが変更できるプロパティはありません。

関連
クラスの関連については、IANet_EthernetPCIDevice を参照してください。

メソッド
このクラスにサポートされているメソッドはありません。

サポートされていない属性
次の属性は WMI Provider ではサポートされていません。

- AdditionalAvailabitlity
- Capabilities
- CapabilityDescriptions
- Caption
- DeviceSelectTiming
- ErrorCleared
- ErrorDescription
- IdentifyingDescription
- InstallDate
- LastErrorCode
- MaxNumberController
- MaxQuiesceTime
- Name
- OtherIdentifyingInfo
- PowerManagementCapabilities
- PowerManagementSupported
- PowerOnHours
- ProtocolDescription
- ProtocolSupported
- SelfTestEnabled
- TimeOfLastReset
- TotalPowerOnHours

先頭に戻る

IANet_EthernetPCIDevice

用途

このクラスは、Intel PROSet でサポートされているイーサネット アダプタの PCI デバイスのプロパティを記述するために使用されます。これは IANet_PCIDevice のサブクラスです。このクラスには、Intel PROSet でサポートされている PCI デバイスのみが認識できる、追加の属性がいくつかあります。

インスタンス
このクラスのインスタンスは、Intel PROSet でサポートされているイーサネット アダプタである各 PCI カードに 1 つ存在します。

インスタンスの作成
IANet_EthernetPCIDevice のインスタンスを作成することはできません。

インスタンス削除
IANet_EthernetPCIDevice のインスタンスを削除することはできません。

プロパティの変更
このクラスには、ユーザが変更できるプロパティはありません。

関連
IANet_AdapterDevice のインスタンスは、IANet_PCIDevice と IANet_EthernetAdapter を関連付けるのに使用されます。仮想アダプタ (チームを表すために作成されたアダプタ) には、関連付けられた IANet_PCIDevice はありません。

サポートされていない属性
次の属性は WMI Provider ではサポートされていません。

- AdditionalAvailabitlity
- Capabilities
- CapabilityDescriptions
- Caption
- DeviceSelectTiming
- ErrorCleared
- ErrorDescription
- IdentifyingDescription
- InstallDate
- LastErrorCode
- MaxNumberController
- MaxQuiesceTime
- Name
- OtherIdentifyingInfo
- PowerManagementCapabilities
- PowerManagementSupported
- PowerOnHours
- ProtocolDescription
- ProtocolSupported
- SelfTestEnabled
- Status
- StatusInfo
- TimeOfLastReset
- TotalPowerOnHours

メソッド
このクラスにサポートされているメソッドはありません。

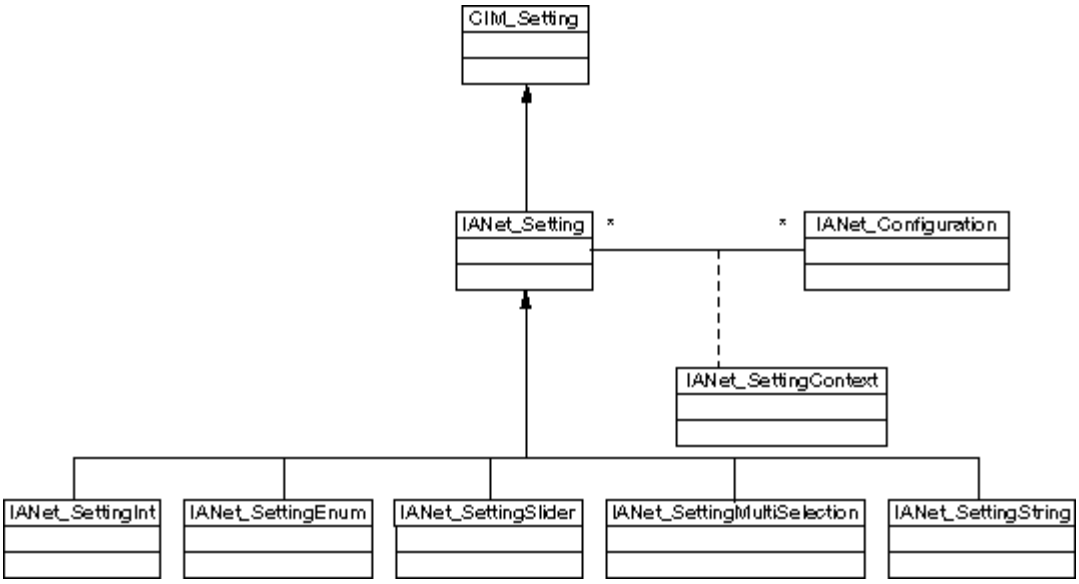
制限と免責条項をすべてお読みください。

目次に戻る 先頭に戻る

目次に戻る

設定スキーマ：Intel(R) PRO ネットワーク アダプタ WMI および CDM プロバイダ ユーザ ガイド

概要



先頭に戻る

IANet_Configuration

用途
このクラスは、IANet_Setting インスタンスのコレクションをグループ化するのに使用されます。このクラスは CIM_Configuration から派生します。

インスタンス
各アダプタ、VLAN、チームは、複数の関連付けられた IANet_Configuration インスタンスを持つことができます。各構成は、アダプタの異なる使用ケースに対応します。

今回 WMI Provider と CDM Provider のリリースでは、各アダプタ、VLAN、チームの IANet_Configuration インスタンスは、それぞれ 1 つのみです。

インスタンスの作成
IANet_Configuration のインスタンスを作成することはできません。

インスタンスの削除
IANet_Configuration のインスタンスを削除することはできません。

プロパティの変更
このクラスには、ユーザが変更できるプロパティはありません。

関連
IANet_AdapterConfiguration のインスタンスは、各アダプタ (IANet_EthernetAdapter) を対応する構成と関連付けるために存在します。IANet_VLANConfiguration インスタンスは、各 VLAN (IANet_VLAN) を対応する構成と関連付けるために存在します。IANet_BootAgentConfiguration インスタンスは、各 Boot Agent (IANet_BootAgent) を対応する構成と関連付けるために存在します。

メソッド
このクラスにサポートされているメソッドはありません。

サポートされていない属性
ありません。

先頭に戻る

IANet_Setting

用途
この抽象クラスは、構成内の設定可能なプロパティを記述するために使用されます。このクラスは CIM_Setting から派生します。

インスタンス
各アダプタ、VLAN、またはチームの各設定に対し、このクラスの独自のインスタンスが存在します。設定は構成間で共有されることはありません。

IANet_Setting にはいくつかのサブクラスがあります。サブクラスは、設定がとることのできる異なるタイプや値の範囲に対応します。各サブクラスは、設定の表示や変更に使用することのできる、異なるスタイルの GUI に対応します。

インスタンスの作成
IANet_Setting のインスタンスを作成することはできません。

インスタンスの削除
IANet_Setting のインスタンスを削除することはできません。

プロパティの変更
この抽象クラスには変更可能なプロパティはありませんが、子クラスには変更可能なプロパティがあります。以下を参照してください。

関連
各 IANet_Setting インスタンスは、IANet_SettingContext のインスタンスを使用して、 IANet_Configuration のインスタンスに関連付けられています。

メソッド
このクラスにサポートされているメソッドはありません。設定を変更するには、必要なプロパティを変更して、**PutInstance** を呼び出します。

サポートされていない属性
SettingID は使用されません。

先頭に戻る

IANet_SettingInt

用途
このクラスは、整数値をとる設定を形成します。整数値の形成に使用される IANet 設定には、いくつかのものがあります。これらのクラスの相違は、GUI が整数を表示および変更し、Provider が検証を行う方法と関係を持ちます。IANet_SettingInt では、GUI にスピン コントロールを持つ編集ボックスが表示されることが予想されます。

インスタンス
このクラスのインスタンスは、整数の編集ボックスとして表示される各設定に 1 つずつ存在します。

インスタンスの作成
このクラスのインスタンスを作成することはできません。

インスタンスの削除
このクラスのインスタンスを削除することはできません。

プロパティの変更
このクラスで変更可能な唯一のプロパティは「CurrentValue」属性です。このプロパティを編集するには、**IWbemClassObject::Put()** を使用して値を変更し、**IWbemServices::PutInstance()** を使用して設定を更新します。Provider では次のことが確認されます。

CurrentValue <= **max**
CurrentValue > = **min**
(**CurrentValue** - **min**) が **Step** の倍数であること

max、**min**、**CurrentValue**、**Step** はすべて IANet_SettingInt の属性です。

関連
各 IANet_SettingInt インスタンスは、IANet_SettingContext のインスタンスを使用して、 IANet_Configuration のインスタンスに関連付けられています。

サポートされていない属性
SettingID は使用されません。

メソッド
このクラスにサポートされているメソッドはありません。設定を変更するには、必要なプロパティを変更して、**PutInstance** を呼び出します。

先頭に戻る

IANet_SettingEnum

用途
このクラスは、整数値をとる設定を形成します。整数値の形成に使用される IANet 設定には、いくつかのものがあります。これらのクラスの相違は、GUI が整数を表示および変更し、Provider が検証を行う方法と関係を持ちます。IANet_SettingEnum では、GUI に少数の列挙された値にマップされる文字列のリスト (ドロップ リスト コンボ ボックスなど) が表示されることが予想されます。

インスタンス
このクラスのインスタンスは、列挙として表示される各設定に 1 つずつ存在します。

インスタンスの作成
このクラスのインスタンスを作成することはできません。

インスタンスの削除
このクラスのインスタンスを削除することはできません。

プロパティの変更
このクラスで変更可能な唯一のプロパティは **CurrentValue** 属性です。このプロパティを編集するには、**Put()** を使用して値を変更し、**PutInstance()** を使用して設定を更新します。Provider では **CurrentValue Є PossibleValues[]** であることが確認されます。

関連
各 IANet_SettingEnum インスタンスは、IANet_SettingContext のインスタンスを使用して、 IANet_Configuration のインスタンスに関連付けられています。

サポートされていない属性
SettingID は使用されません。

メソッド
このクラスでサポートされているメソッドはありません。設定を変更するには、必要なプロパティを変更して、**PutInstance** を呼び出します。

先頭に戻る

IANet_SettingSlider

用途
このクラスは、整数値をとる設定を形成します。整数値の形成に使用される IANet 設定には、いくつかのものがあります。これらのクラスの相違は、GUI が整数を表示および変更し、Provider が検証を行う方法と関係を持ちます。IANet_SettingSlider では、GUI にグラフィックを使用して値を選択できるスライダが表示されることが予想されます。また、実際に選択した値も表示する必要があります。

インスタンス
このクラスのインスタンスは、スライダとして表示される各設定に 1 つずつ存在します。

インスタンスの作成
このクラスのインスタンスを作成することはできません。

インスタンスの削除
このクラスのインスタンスを削除することはできません。

プロパティの変更
このクラスで変更可能な唯一のプロパティは **CurrentValue** 属性です。このプロパティを編集するには、**Put()** を使用して値を変更し、**PutInstance()** を使用して設定を更新します。Provider では **CurrentValue Є PossibleValues[]** であることが確認されます。

関連
各 IANet_SettingSlider インスタンスは、IANet_SettingContext のインスタンスを使用して、 IANet_Configuration のインスタンスに関連付けられています。

サポートされていない属性
SettingID は使用されません。

メソッド
このクラスでサポートされているメソッドはありません。設定を変更するには、必要なプロパティを変更して、**PutInstance** を呼び出します。

先頭に戻る

IANet_SettingMultiSelection

用途
このクラスは、オプションのリストから複数のオプションを選択できる設定を形成します。IANet_SettingMultiSelection では、GUI に任意のオプションを選択できる (また、オプションを 1 つも選択しないこともできる) 複数選択のリスト ボックスが表示されることが予想されます。

インスタンス
このクラスのインスタンスは、複数選択として表示される各設定に 1 つずつ存在します。

インスタンスの作成
このクラスのインスタンスを作成することはできません。

インスタンスの削除
このクラスのインスタンスを削除することはできません。

プロパティの変更

このクラスで変更可能な唯一のプロパティは **CurrentValue** 属性です。このプロパティを編集するには、**Put()** を使用して値を変更し、**PutInstance()** を呼び出して設定を更新します。Provider では **CurrentValue Є PossibleValues[]** であることが確認されます。

関連
各 IANet_SettingMultiSelection インスタンスは、IANet_SettingContext のインスタンスを使用して、 IANet_Configuration のインスタンスに関連付けられています。

サポートされていない属性
SettingID は使用されません。

メソッド
このクラスにサポートされているメソッドはありません。設定を変更するには、必要なプロパティを変更して、**PutInstance** を呼び出します。

先頭に戻る

IANet_SettingString

用途
このクラスは、自由形式の文字列値を入力できる設定を形成します。IANet_SettingMultiSelection では、GUI に編集ボックスが表示されることが予想されます。

インスタンス
このクラスのインスタンスは、編集ボックスとして表示される各設定に 1 つずつ存在します。

インスタンスの作成
このクラスのインスタンスを作成することはできません。

インスタンスの削除
このクラスのインスタンスを削除することはできません。

プロパティの変更
このクラスで変更可能な唯一のプロパティは **CurrentValue** 属性です。このプロパティを編集するには、**Put()** を使用して値を変更し、**PutInstance()** を使用して設定を更新します。

関連
各 IANet_SettingMultiSelection インスタンスは、IANet_SettingString のインスタンスを使用して、 IANet_ElementConfiguration のインスタンスに関連付けられています。

メソッド
このクラスにサポートされているメソッドはありません。

サポートされていない属性
SettingID は使用されません。

メソッド
このクラスにサポートされているメソッドはありません。設定を変更するには、必要なプロパティを変更して、**PutInstance** を呼び出します。

制限と免責条項をすべてお読みください。

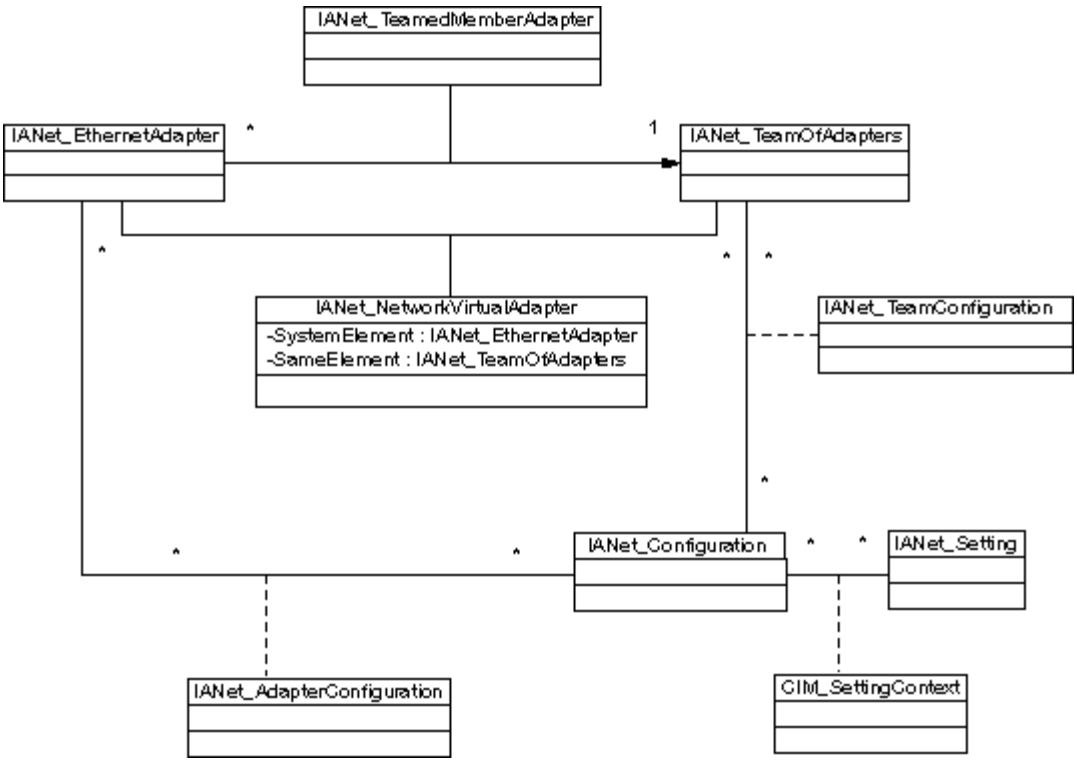
目次に戻る 先頭に戻る

目次に戻る

チーム スキーマ：Intel(R) PRO ネットワーク アダプタ WMI および CDM プロバイダ ユーザ ガイド

概要

チーム スキーマは、イーサネット アダプタをどのようにグループ化してチームを形成するかを記述します。



先頭に戻る

IANet_TeamOfAdapters

用途
このクラスは、CIM_RedundancyGroup クラスを実装します。このクラスは、チームのタイプ、チーム内のアダプタ数、およびチーム内に存在できる最大のアダプタ数を記述するメンバーを持ちます。

インスタンス
各インテル チームに対し、このクラスのインスタンス 1 つが存在します。

インスタンスの作成
空のチームを作成するには、IANet_TeamOfAdapters のインスタンスを作成します。Provider でオブジェクトを作成するには、**IWbemServices::PutInstance()** を呼び出す前に **TeamingMode** を適切に設定することが必要です。Providerswill では、新しいオブジェクトのオブジェクト パスを示す文字列が返されます。

インスタンスの削除
同様に、チームを削除するには、IANet_TeamOfAdapters のインスタンスを削除します。Providerswill により、チーム メンバーと仮想アダプタ、およびチームの設定の関連が削除されます。

プロパティの変更
Put() を使用して **TeamingMode** プロパティの値を変更し、次に **PutInstance()** を呼び出して、チームを更新します。

関連
チーム内の各アダプタは、IANet_TeamMemberAdapter のインスタンスを使用して、チームの IANet_TeamOfAdapters のインスタンスと関連付けられます。チーム仮想アダプタは、IA_NetNetworkVirtualAdapter のインスタンスを使用して、このクラスに関連付けられます。

メソッド
このクラスのインスタンスでは、次のメソッドがサポートされます。

TestSwitchConfiguration - スイッチの設定をテストして、チームがスイッチに対して適切に機能していることを確認します。このテストは、リンク パートナー (アダプタのリンク先のデバイス - 別のアダプタやハブ、スイッチなど) が選択したアダプタのチーム化モードをサポートしているかどうかを確認するのに使用できます。たとえば、アダプタがリンク アグリゲーション チームのメンバーの場合、このテストによって、リンク パートナーがリンク アグリゲーションをサポートするアダプタに接続していることを確認できます。

先頭に戻る

IANet_TeamedMemberAdapter

用途
このクラスは、アダプタをチームと関連付けるのに使用され、チーム内のアダプタの機能を判定し、チーム内でアダプタをアクティブにします。このクラスは、CIM クラスである CIM_NetworkAdapterRedundancyComponent を実装します。

インスタンス
このクラスのインスタンスは、チームのメンバーである各アダプタに 1 つずつ存在します。

インスタンスの作成
チームにアダプタを追加するには、IANet_TeamedMemberAdapter のインスタンスを作成して、アダプタをチームと関連付けます。

インスタンスの削除
チームからアダプタを削除するには、IANet_ TeamedMemberAdapter のインスタンスを削除します。**Apply()** 関数を呼び出すと、アダプタはチームの一部ではなくなり、IP プロトコル エンドポイントにバインドされることがあります。

プロパティの変更
このクラスの **AdapterFunction** プロパティは、チーム内でのアダプタの使用を記述するために、変更することができます。

関連
これは関連クラスです。

メソッド
このクラスでサポートされているメソッドはありません。

先頭に戻る

IANet_NetworkVirtualAdapter

用途
このクラスは、チームの IANet_TeamOfAdapters を、チームの仮想アダプタを表す IANet_EthernetAdapter と関連付けるのに使用されます。このクラスは、CIM クラスである CIM_ CIM_NetworkVirtualAdapter を実装します。

インスタンス
このクラスのインスタンスは、仮想アダプタにバインドされている各インテル チームに 1 つずつ存在します。

インスタンスの作成
このクラスのインスタンスを作成することはできません。チームを作成するには、IANet_TeamOfAdapters のインスタンスを作成します。このクラスは、有効なセッションのコンテキストで **IANet_NetService .Apply()** を呼び出し、IANet_EthernetAdapter のインスタンスを作成するまでは存在しません。

インスタンスの削除
このクラスのインスタンスを削除することはできません。

関連
これは関連クラスです。

メソッド
このクラスでサポートされているメソッドはありません。

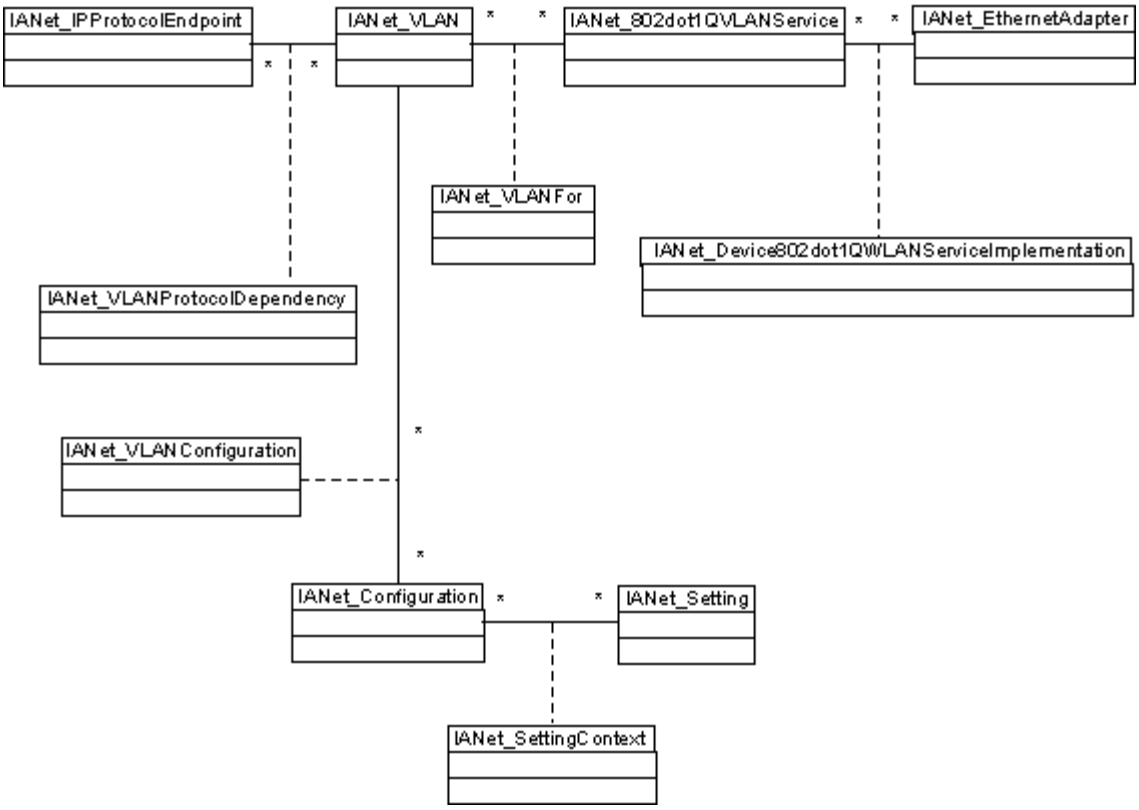
制限と免責条項をすべてお読みください。

目次に戻る 先頭に戻る

目次に戻る

VLAN スキーマ：Intel(R) PRO ネットワーク アダプタ WMI および CDM プロバイダ ユーザ ガイド

概要



先頭に戻る

IANet_802dot1QVLANService

用途
このクラスは、ネットワーク アダプタの IEEE 802.1Q プロパティを保持するのに使用されます。このクラスは、CIM クラスである CIM_ CIM_802dot1QVLANService を実装します。

インスタンス
このクラスのインスタンスは、IEEE 802.1Q をサポートする各アダプタまたはチームに 1 つずつ存在します。各アダプタは IANet_802dot1QVLANService を 1 つのみもつことができます。マルチベンダ フォルト トレラント チームなど、一部のチームではこのサービスはサポートされていません。

例外

VLAN をもたないチームでは、有効なセッションのコンテキストで VLAN を列挙しない限り、VLAN サービスはありません。チームでは、802.3QvlanService のインスタンスは次の場合にのみ存在します。

- チームにすでに VLAN がある場合。
- チームに VLAN がなく、使用しているセッション ハンドルのコンテキストでこのクラスが列挙されている場合。

インスタンスの作成
このクラスのインスタンスを作成することはできません。アダプタにインスタンスが関連付けられていない場合、そのアダプタはこのサービスをサポートしません。

インスタンスの削除
このクラスのインスタンスを削除することはできません。

プロパティの変更
このクラスには、変更できるプロパティはありません。

関連
このクラスの各インスタンスは、IANet_DeviceServiceImplementation を使用して、IANet_EthernetAdapter の 1 つに関連付けられます。

IANet_802dot1QVLANService の各インスタンスは、複数の VLAN をサポートできます。各 VLAN は IANet_VLANFor の関連を使用して、インスタンスに関連付けられます。

メソッド
uint16 CreateVLAN([in] uint32 VLANNumber, [in] string Name, [out] IANet_VLAN REF VLANpath); アダプタやチームに VLAN を作成します。クライアントは VLAN の番号と名前を入力する必要があります。新しく作成された VLAN のオブジェクト パスが返されます。

先頭に戻る

IANet_VLAN

用途
このクラスは、各インテル VLAN の情報を保持します。このクラスは CIM_VLAN を実装します。

インスタンス
各インテル VLAN ごとに、このクラスのインスタンス 1 つが存在します。

インスタンスの作成
VLAN を作成するには、適切な IANet_802dot1QVLANService のインスタンスに **CreateVLAN** を呼び出します。

インスタンスの削除
このクラスのインスタンスを削除すると、対応する VLAN が削除されます。

プロパティの変更
VLANNumber と Caption の属性を変更できます。

関連
各インスタンスは、IANet_802dot1QVLANService のインスタンスと関連付けられます。このため、IANet_VLANFor クラスを使用して IANet_EthernetAdapter のインスタンスに関連付けられます。

各インスタンスは、IANet_Configuration の複数のインスタンスと関連付けて、VLAN の一連の設定をグループ化することができます。今回の Provider のリリースでは、各 VLAN に対し IANet_Configuration は 1 つのみです。

各インスタンスは IANet_IPProtocolEndpoint の 1 つと関連付けて、IANet_VLANProtocolDependency クラスを使用して IP 設定を提供することができます。

メソッド
なし

目次に戻る

現在の設定の取得：Intel(R) PRO ネットワーク アダプタ WMI および CDM プロバイダ ユーザ ガイド

現在の設定を読み込むために、クライアントはセッション ハンドルを取得する必要はありません。クライアントは NULL コンテキストを使用することができますが、管理されているマシンでは、すべてのエラー メッセージはデフォルトの言語で返されます。次の表の {} で囲まれたアイテムは、オブジェクトのパスです。これらのパスは、以前の WQL クエリで取得されたと仮定します。クライアントでは、クエリなしでオブジェクト パスを作成する必要はありません。各オブジェクトの **__PATH** 属性には、そのオブジェクトのオブジェクト パスが含まれます。

以下の使用例では、WQL クエリの実行に **IWbemServices::ExecQuery** メソッド、または **IWbemServices::ExecQueryAsync** メソッドを使用しています。

物理アダプタの取得

タスク	**WQL** クエリ	結果のクラス	注釈
全アダプタの列挙	SELECT * FROM IANet_EthernetAdapter	IANet_EthernetAdapter	すべての IANet_EthernetAdapters を返す IWbemServices::CreateInstanceEnumAsync と同様
アダプタが仮想アダプタかどうかの判定	ASSOCIATORS OF {アダプタのパス} WHERE AssocClass = IANet_NetworkVirtualAdapter	IANet_TeamOfAdapters	クエリで何もクラスが返されない場合は、アダプタは実際のアダプタです。
アダプタがゴースト アダプタかどうかの判定	ASSOCIATORS OF {アダプタのパス} WHERE ResultClass = IANet_EthernetPCIDevice	IANet_EthernetPCIDevice	アダプタが仮想アダプタではなく、このクエリによってオブジェクトが返されない場合、アダプタはゴースト アダプタです。

アダプタの主なクラスは IANet_EthernetAdapter です。このクラスは物理アダプタと仮想アダプタの両方に使用されます。クライアントで両者を識別する方法がわかっていることが必要です。

先頭に戻る

PCI デバイスの取得

主なクラスは IANet_EthernetPCIDevice、IANet_PCIDevice、および IANet_AdapterDevice (アダプタをデバイスに関連付ける関連クラス) です。

この場合、関連クラスにはデータは含まれません。つまり、関連クラスにはデータはありません。IANet_EthernetPCIDevice は IANet_PCIDevice から継承され、イーサネット アダプタである PCI デバイスに特有な追加の属性を含みます。

タスク	**WQL** クエリ	結果のクラス	注釈
ネットワーク PCI デバイスの列挙	SELECT * FROM IANet_PCIDevice	IANet_PCIDevice	クエリで Intel(R) PROSet では設定されないモデムやその他のデバイスが返される場合があります。 IWbemServices::CreateInstanceEnumAsync と同様
Intel(R) PROSet で設定されたアダプタで使用されるすべての PCI デバイスを列挙	SELECT * FROM IANet_EthernetPCIDevice	IANet_EthernetPCIDevice	クエリでは、Intel(R) PROSet で管理される PCI デバイスのみが返されます。 IWbemServices::CreateInstanceEnumAsync と同様
アダプタがインストールされているかどうかの判定	該当なし	IANet_EthernetPCIDevice	IANet_EthernetPCIDevice の Availability 属性を確認します。この属性が 10 - Not Installed (インストールされていない) の場合、デバイスはインストールされていません。注：この時点では、デバイスについて確認されている情報はあまりありません。
アダプタに関連付けられている PCI デバイスの取得	ASSOCIATORS OF { IANet イーサネット アダプタのパス} WHERE ResultClass = IANet_EthernetPCIDevice	IANet_EthernetPCIDevice	オブジェクトが返されない場合、アダプタは仮想アダプタか、ゴースト アダプタです。

PCI デバイスに関連付けられたアダプタの取得	ASSOCIATORS OF {イーサネット PCI デバイスのパス} WHERE ResultClass = IANet_EthernetAdapter	IANet_EthernetAdapter	このクエリは、Provider に最適なものではありません。クライアントは、アダプタのある状態で起動するのが理想的です。

先頭に戻る

アダプタ設定の取得

設定オブジェクトは、アダプタと直接関連付けられてはいません。CIM 標準に準拠して、設定オブジェクトは構成オブジェクトと関連付けられ、構成オブジェクトがアダプタと関連付けられます。

スキーマのこの部分に関わるクラスは、IANet_EthernetAdapter、IANet_Configuration、IANet_SettingInt、IANet_SettingString、IANet_SettingEnum、IANet_SettingMultiSelection および IANetSettingSlider です。

関連クラスである IANet_AdapterConfiguration と IANet_SettingContext には、実際のデータは含まれません。これらのクラスは、設定と親オブジェクトの橋渡しの役割を果たします。

タスク	**WQL** クエリ	結果のクラス	注釈
アダプタの構成オブジェクトの取得	ASSOCIATORS OF { IANet イーサネット アダプタのパス} WHERE ResultClass = IANet_Configuration	IANet_Configuration	1 つのオブジェクトのみを返します。設定がない場合でも、常に構成オブジェクトは存在します。次のクエリでは、このオブジェクトからのオブジェクト パスが使用されます。
アダプタに関連付けられている設定の取得	ASSOCIATORS OF {IANet の構成パス} WHERE AssocClass = IANet_SettingContext	IANet_SettingInt、IANet_SettingString、IANet_SettingEnum、IANet_SettingMultiSelection、および IANet_SettingSlider が混在します。	アダプタに関連付けられているすべての設定クラスを返します。クライアントで各設定のタイプを判定するには、__CLASS 属性を使用します。

先頭に戻る

チーム構成の取得

チーム化スキーマの主なクラスには、IANet_EthernetAdapter、IANet_TeamOfAdapters、IANet_NetworkVirtualAdapter、および IANet_TeamedMemberAdapter があります。

このスキーマで問題になるのは、IANet_EthernetAdapter のインスタンスが各物理アダプタと各仮想アダプタに存在するという点です。クライアントでは、チームの仮想アダプタと、チームのメンバーであるアダプタを区別できることが必要です。

関連クラス IANet_NetworkVirtualAdapter には、実際に使用されるデータは含まれません。クライアントで問題となるのは、この関連のエンドポイントのみです。IANet_TeamedMemberAdapter には、チーム内でメンバー アダプタがどのように使用されるかについて、有用なデータは含まれていません。

タスク	**WQL** クエリ	結果のクラス	注釈
すべてのチームの列挙	SELECT * FROM IANet_TeamOfAdapters	IANet_TeamOfAdapters	各チームに対し、IANet_TeamOfAdapters のインスタンスが 1 つ存在します。 IWbemServices::CreateInstanceEnumAsync と同様
チームの仮	ASSOCIATORS OF {IANet アダプタ チームの	IANet_EthernetAdapter	チーム内の仮想アダプタのアダプタ オブジェクトのみを返します。

想アダプタの取得	パス} WHERE AssocClass = IANet_NetworkVirtualAdapter		チームが作成されていても、Apply が呼び出されていない場合は、このアダプタは存在しません(下記の設定の更新の項を参照してください)。
チームのメンバー アダプタの列挙	ASSOCIATORS OF {IANet アダプタ チームのパス} WHERE AssocClass = IANet_TeamedMemberAdapter	IANet_EthernetAdapter	チーム内のアダプタを返しますが、各アダプタの役割は表示されません。
チーム内のアダプタの役割の判定	REFERENCES OF {IANet イーサネット アダプタのパス} WHERE ResultClass = IANet_TeamedMemberAdapter	IANet_TeamedMemberAdapter	このクラスには、メンバー アダプタとチームの関係およびチーム内での現在の状態に関する情報が含まれます。

先頭に戻る

チーム設定の取得

設定オブジェクトは、チームと直接関連付けられてはいません。CIM 標準に準拠して、設定オブジェクトは構成オブジェクトと関連付けられ、構成オブジェクトがチームの仮想 IANet_EthernetAdapter と関連付けられます。また、同じ構成オブジェクトがチームの IANet_TeamOfAdapters オブジェクトにも関連付けられます。

スキーマのこの部分に関わるクラスは、IANet_EthernetAdapter、IANet_TeamOfAdapters、IANet_Configuration、IANet_SettingInt、IANet_SettingString、IANet_SettingEnum、IANet_SettingMultiSelection、および IANetSettingSlider です。

関連クラスである IANet_AdapterConfiguration と IANet_SettingContext には、実際のデータは含まれません。これらのクラスは、設定と親オブジェクトの橋渡しの役割を果たします。これは、アダプタ設定の場合と同じです。

タスク	**WQL** クエリ	結果のクラス	注釈
仮想アダプタから始まる、チームの構成オブジェクトの取得	ASSOCIATORS OF { IANet イーサネット アダプタのパス} WHERE ResultClass = IANet_Configuration	IANet_Configuration	1 つのオブジェクトのみを返します。設定がない場合でも、構成オブジェクトは常に存在します。次のクエリでは、このオブジェクトからのオブジェクト パスが使用されます。
アダプタのチームから始まる、チームの構成オブジェクトの取得	ASSOCIATORS OF { IANet アダプタ チームのパス} WHERE ResultClass = IANet_Configuration		
アダプタに関連付けられている設定の取得	ASSOCIATORS OF {IANet の構成パス} WHERE AssocClass = IANet_SettingContext	IANet_SettingInt、IANet_SettingString、IANet_SettingEnum、IANet_SettingMultiSelection、および IANet_SettingSlider が混在します。	アダプタに関連付けられているすべての設定クラスを返します。クライアントで各設定のタイプを判定するには、__CLASS 属性を使用します。

先頭に戻る

VLAN 設定の取得

VLAN をサポートする各アダプタには、関連クラス IANet_Device802do1QVVLANServiceImplementation を使用して、IANet_802dot1QVLANService が関連付けられています。アダプタにこのクラスのインスタンスが関連付けられていない場合、そのアダプタは VLAN をサポートしません。

各 VLAN は、IANet_VLAN のインスタンスで表されます。VLAN はアダプタと直接関連付けられてはいません。VLAN はアダプタの IANet_802dot1QVLANService に関連付けられます。

各 VLAN を適切な IANet_802dot1QVLANService と関連付けるために、関連クラス IANet_VLANFor が使用されます。このクラスには、ユーザに有用なデータは含まれていません。

タスク	**WQL** クエリ	結果のクラス	注釈

アダプタに関連付けられている 802.1q VLAN サービス オブジェクトの取得	ASSOCIATORS OF {IANet イーサネット アダプタのパス} WHERE ResultClass = IANet_802dot1QVLANService	IANet_802dot1QVLANService	1 つのオブジェクトを返すか、オブジェクトを返しません。
アダプタの VLAN の取得	ASSOCIATORS OF {IANet 802dot1QVLANService パス} WHERE ResultClass = IANet_VLAN	IANet_VLAN	VLAN がインストールされていない場合は、オブジェクトが返されません。

先頭に戻る

VLAN 設定の取得

設定オブジェクトは、VLAN と直接関連付けられてはいません。CIM 標準に準拠して、設定オブジェクトは構成オブジェクトと関連付けられ、構成オブジェクトが VLAN の IANet_VLAN オブジェクトと関連付けられます。

スキーマのこの部分に関わるクラスは、IANet_VLAN、IANet_Configuration、IANet_SettingInt、IANet_SettingString、IANet_SettingEnum、IANet_SettingMultiSelection および IANetSettingSlider です。

関連クラスである IANet_VLANConfiguration と IANet_SettingContext には、実際のデータは含まれません。これらのクラスは、設定と親オブジェクトの橋渡しの役割を果たします。これは、アダプタ設定の場合と同じです。

タスク	**WQL** クエリ	結果のクラス	注釈
VLAN の構成オブジェクトの取得	ASSOCIATORS OF { IANet VLAN のパス} WHERE ResultClass = IANet_Configuration	IANet_Configuration	1 つのオブジェクトのみを返します。設定がない場合でも、常に構成オブジェクトは存在します。次のクエリでは、このオブジェクトからのオブジェクト パスが使用されます。
VLAN に関連付けられている設定の取得	ASSOCIATORS OF {IANet の構成パス} WHERE AssocClass = IANet_SettingContext	IANet_SettingInt、IANet_SettingString、IANet_SettingEnum、IANet_SettingMultiSelection、および IANet_SettingSlider が混在します。	VLAN に関連付けられているすべての設定クラスを返します。クライアントで各設定のタイプを判定するには、__CLASS 属性を使用します。

先頭に戻る

IP プロトコル情報の取得

アダプタ、VLAN、およびチームに関連付けられた IP プロトコル エンドポイントについて、Provider から限られた情報が提供されます。このほかのプロトコルはサポートされていません。

プロトコル情報を含む主なクラスは IANet_IPProtocolEndpoint です。関連クラスには、IANet_VLANProtocolDependency と IANet_AdapterProtocolImplementation の 2 つがあります。チームの IP エンドポイントを取得するには、まずチームの仮想 IANet_EthernetAdapter (このインスタンスに関連付けられた IP エンドポイント) を取得します。

タスク	**WQL** クエリ	結果のクラス	注釈
アダプタに関連付けられている IP プロトコル エンドポイントの取得	ASSOCIATORS OF {IANet イーサネット アダプタのパス} WHERE ResultClass = IANet_IPProtocolEndpoint	IANet_IPProtocolEndpoint	アダプタによっては複数の IP アドレスを持つものもありますが、これらのアドレスはすべて 1 つの IP プロトコル エンドポイントに関連付けられます。
VLAN に関連付けられている IP プロトコル エンドポイントの取得	ASSOCIATORS OF {IANet VLAN のパス} WHERE ResultClass = IANet_IPProtocolEndpoint	IANet_IPProtocolEndpoint	VLAN に関連付けられている IP プロトコル エンドポイントは 1 つです。

先頭に戻る

Boot Agent 情報の取得

フラッシュ ROM の Boot Agent をサポートできる各アダプタには、関連クラス IANet_DeviceBootServiceImplementation を使用して、IANet_BootAgent インスタンス

が関連付けられています。

タスク	WQL クエリ	結果のクラス	注釈
アダプタに関連付けられている Boot Agent の取得	ASSOCIATORS OF {IANet イーサネット アダプタのパス} WHERE ResultClass = IANet_BootAgent	IANet_BootAgent	次の読み取り専用の属性では、このアダプタの起動 ROM イメージの情報が提供されます。InvalidImageSignature、Version、UpdateAvailable、FlashImageType

先頭に戻る

Boot Agent 設定の取得

設定オブジェクトは、Boot Agent と直接関連付けられてはいません。CIM 標準に準拠して、設定オブジェクトは構成オブジェクトと関連付けられ、構成オブジェクトが Boot Agent と関連付けられます。

スキーマのこの部分に関わるクラスは、IANet_BootAgent、IANet_Configuration、IANet_SettingInt、IANet_SettingString、IANet_SettingEnum、IANet_SettingMultiSelection および IANetSettingSlider です。

関連クラスである IANet_BootAgentConfiguration と IANet_SettingContext には、実際のデータは含まれません。これらのクラスは、設定と親オブジェクトの橋渡しの役割を果たします。

タスク	WQL クエリ	結果のクラス	注釈
Boot Agent の構成オブジェクトの取得	ASSOCIATORS OF { IANet Boot Agent のパス} WHERE ResultClass = IANet_Configuration	IANet_Configuration	1 つのオブジェクトのみを返します。設定がない場合でも、構成オブジェクトは常に存在します。次のクエリでは、このオブジェクトからのオブジェクト パスが使用されます。
Boot Agent に関連付けられている設定の取得	ASSOCIATORS OF {IANet の構成パス} WHERE AssocClass = IANet_SettingContext	IANet_SettingInt、IANet_SettingString、IANet_SettingEnum、IANet_SettingMultiSelection、および IANet_SettingSlider が混在します。	アダプタに関連付けられているすべての設定クラスを返します。クライアントで各設定のタイプを判定するには、__CLASS 属性を使用します。

制限と免責条項をすべてお読みください。

目次に戻る 先頭に戻る

目次に戻る

設定の更新：Intel(R) PRO ネットワーク アダプタ WMI および CDM プロバイダ ユーザ ガイド

概要

ほとんどの場合、設定を更新するには、クライアント アプリケーションが IANet_NetService クラスからセッション ハンドルを入手し、このハンドルを IWbemContext コンテキスト オブジェクトに保存する必要があります。設定は IANet_NetService に **Apply** メソッドが呼び出された場合にのみ変更されます。ただし、これにはいくつかの例外があります。

- Boot Agent への変更は、変更を行うとすぐに適用され、セッション ハンドルは必要ありません。
- 特定のメソッドの呼び出し (アダプタの識別など) では、**Apply** が呼び出される前に処理が実行されます。

一部の処理では、コンテキストに PreCheck 修飾子を使用して、処理が許可されているかどうかを確認することができます。これにより、必要に応じてユーザ インターフェイスで特定のコントロールやメニュー アイテムを無効にできます。

先頭に戻る

アダプタ、チーム、または VLAN 設定の変更

アダプタ、チーム、または VLAN 設定を変更するには：

- セッション ハンドルが必要です。
- PreCheck を使用できます。
- 処理を実行するには、**Apply** の呼び出しが必要です。

アダプタ、VLAN、またはチームの設定を変更するには、クライアントが変更する設定のオブジェクト パスをあらかじめ取得しておくことが必要です。このためには、オブジェクトの設定を列挙して、設定の **__PATH** 属性を保存する方法 (上記参照) をお勧めします。

クライアントで設定を更新するもっとも簡単な方法は次のとおりです。

□□□ WMI から設定オブジェクトのインスタンスを取得します。
□□□ **IWbemClassObject::Put()** を使用して、CurrentValue を変更します。
□□□ **IWbemServices::PutInstance()** を呼び出して、変更したインスタンスを WMI Provider に戻します。**PutInstance** を WBEM_FLAG_UPDATE_ONLY のフラグをつけて呼び出します。

WMI Provider により CurrentValue が検証され、検証に失敗した場合は WBEM_E_FAIL が返されます。IANet_ExtendedStatus オブジェクトの Description 属性では、失敗の理由が返されます。

設定に特有の説明には、次のものがあります。

- The integer setting value was less than the minimum allowed. (整数の設定値が許容最小値よりも小さい。)
- The integer setting value was greater than the maximum allowed. (整数の設定値が許容最大値よりも大きい。)
- The integer setting value is not one of the allowable steps. (整数の設定値が許可されているものではない。)
- The length of the string setting is bigger than the maximum allowed. (文字列の設定が、許容最大長さよりも長い。)
- The setting value is not one of the allowable values. (設定値が許可されているものではない。)

IANet_SettingEnum、IANet_SettingSlider、または IANet_SettingMultiSelection の現在値が許容値でない際に、上記の説明が返されます。

クライアントが変更可能な設定の唯一の属性は、CurrentValue です。 WMI Provider は、その他の値に加えられた変更を無視します。

メソッド このクラスでサポートされているメソッドはありません。設定を変更するには、必要なプロパティを変更して、**PutInstance** を呼び出します。

先頭に戻る

新しい空のチームの作成

新規チームの作成：

- セッション ハンドルが必要です。
- PreCheck を使用できます。
- 処理を実行するには、**Apply** の呼び出しが必要です。

新規チームを作成するには、IANet_TeamOfAdapters のインスタンスを作成して (たとえば、**IWbemServices::GetObject()** を使用し、IANet_TeamOfAdapters のクラス オブジェクトを取得してから、**IWbemServices::SpawnInstance()** を使用してこのオブジェクトのインスタンスを作成する)。

次に、**IWbemClassObject::Put** を使用し、希望のチーム タイプになるように (例： AFT) インスタンスの TeamMode 属性を設定します。 最後に、**IWbemServices::PutInstance()** を呼び出し、フラッグ WBEM_FLAG_CREATE_ONLY を渡してチームを作成します。

新規チームのオブジェクト パスは、**IWbemCallResultObject** に保管され、呼び出しが終了した際にユーザに戻されます。 メソッド **IWbemCallResult::GetResultString** は、新規のオブジェクト パスを取得します。

この操作が失敗した場合は、IANet_ExtendedStatus でエラー コードを確認してください。

チームの仮想 IANet_EthernetAdapter および IANet_IPProtocolEndpoint クラスは、**Apply** への呼び出しが実行されるまで利用できません。 チームの設定は、新しい IANet_TeamOfAdapters に関連付けられた IANet_Configuration オブジェクトを使用してアクセスできます。

先頭に戻る

アダプタのチームへの追加

アダプタのチームへの追加：

- セッション ハンドルが必要です。
- PreCheck を使用できます。
- 処理を実行するには、**Apply** の呼び出しが必要です。

アダプタをチームに追加するには、IANet_TeamedMemberAdapter のインスタンスを作成します。つまり、**IWbemServices::GetObject()** を使用して IANet_TeamedMemberAdapter のクラス オブジェクトを取得し、**IWbemServices::SpawnInstance()** を使用してこのオブジェクトのインスタンスを作成します。

IWbemClassObject::Put(): を使用して、オブジェクトの次の属性を設定する必要があります。

- GroupComponent：アダプタを追加する IANet_TeamOfAdapters の完全なオブジェクト パスを設定します。
- PartComponent：チームに追加する IANet_EthernetAdapter の完全なオブジェクト パスを設定します。

また、チーム内のアダプタの優先度を設定することもできます。最後に、**IWbemClassObject:: PutInstance** を呼び出して WBEM_FLAG_CREATE_ONLY フラグを渡して呼び出し、アダプタをチームに追加します。 この操作が失敗した場合は、IANet_ExtendedStatus でエラー コードを確認してください。

先頭に戻る

アダプタのチームからの削除

アダプタのチームからの削除：

- セッション ハンドルが必要です。
- PreCheck を使用できます。
- 処理を実行するには、**Apply** の呼び出しが必要です。

アダプタをチームから削除するには、**IWbemServices::DeleteInstance()** を使用して、アダプタをチームに関連付ける IANet_TeamedMemberAdapter を削除します。この操作が失敗した場合は、IANet_ExtendedStatus でエラー コードを確認してください。

先頭に戻る

チームの削除

チームの削除：

- セッション ハンドルが必要です。
- PreCheck を使用できます。
- 処理を実行するには、**Apply** の呼び出しが必要です。

チームを削除するには、**IWbemServices::DeleteInstance()** を使用して、IANet_TeamOfAdapters のインスタンスを削除します。 この操作が失敗した場合は、IANet_ExtendedStatus でエラー コードを確認してください。

先頭に戻る

チームのモードの変更

チームのモードの変更：

- セッション ハンドルが必要です。
- PreCheck を使用できます。
- 処理を実行するには、**Apply** の呼び出しが必要です。

チームのモードを変更するには、IANet_TeamOfAdapters のインスタンスを入手します。つまり、チームのオブジェクト パスを使用して、**IWbemServices::GetObject** を使います。 次に、**IWbemClassObject::Put** を使用して、チームの TeamMode 属性を変更します。 最後に、**IWbemClassObject::PutInstance** を呼び出して、WMI Provider にチーム モードの更新を告げ、WBEM_FLAG_UPDATE_ONLY フラグを渡します。 この操作が失敗した場合は、IANet_ExtendedStatus でエラー コードを確認してください。

先頭に戻る

アダプタのチーム内の優先度の変更

アダプタのチーム内での優先度の変更：

- セッション ハンドルが必要です。
- PreCheck を使用できます。
- 処理を実行するには、**Apply** の呼び出しが必要です。

アダプタの優先度を変更するには、まずクライアントでアダプタの IANet_TeamedMemberAdapter のインスタンスを入手します。たとえば、オブジェクト パスを使用して **IWbemServices::GetObject** を使います。次にクライアントで **IWbemClassObject::Put** を使用して、アダプタの AdapterFunction 属性を変更します。 最後に、**IWbemClassObject:: PutInstance** を呼び出し、WMI Provider にアダプタの優先度の更新を告げます。 この操作が失敗した場合は、IANet_ExtendedStatus でエラー コードを確認してください。

先頭に戻る

アダプタのアンインストール

アダプタのアンインストール：

- セッション ハンドルが必要です。
- PreCheck を使用できます。
- 処理を実行するには、**Apply** の呼び出しが必要です。

アダプタをアンインストールするには、アンインストールするアダプタのオブジェクト パスを渡して、**IWbemServices::DeleteInstance** を呼び出します。

先頭に戻る

VLAN の作成

VLAN の作成：

- セッション ハンドルが必要です。
- PreCheck を使用できます。
- 処理を実行するには、**Apply** の呼び出しが必要です。

VLAN を作成するには、VLAN を追加するアダプタの IANet_802dot1QVLANService に **CreateVLAN** メソッドを呼び出します。メソッドに次の引数を渡す必要があります。

- VLANNumber：VLAN の番号(範囲： 1-4094)
- Name：VLAN を識別するための、ユーザによって定義された名前。

この機能では、新しく作成された VLAN のオブジェクト パスが、Out パラメータ VLANpath で返されます。この操作が失敗した場合は、IANet_ExtendedStatus でエラー コードを確認してください。

先頭に戻る

VLAN の属性の変更

VLAN の属性の変更：

- セッション ハンドルが必要です。
- PreCheck を使用できます。
- 処理を実行するには、**Apply** の呼び出しが必要です。

クライアントでは、VLAN の VLANNumber 属性と VLANName 属性を変更できます。アダプタの優先度を変更するには、まずアダプタの IANet_VLAN のインスタンスを入手します。たとえば、オブジェクト パスを使って **IWbemServices::GetObject** を使用します。

次に、VLANNumber か VLANName を希望する値に変更します。 最後に、**IWbemClassObject::PutInstance** を呼び出して WMI Provider に属性の更新を告げ、WBEM_FLAG_UPDATE_ONLY フラグを渡します。 この操作が失敗した場合

は、IANet_ExtendedStatus でエラー コードを確認してください。

先頭に戻る

VLAN の削除

VLAN の削除：

- セッション ハンドルが必要です。
- PreCheck を使用できます。
- 処理を実行するには、**Apply** の呼び出しが必要です。

VLAN を削除するには、削除する VLANのオブジェクト パスを渡して、**IWbemServices::DeleteInstance** を呼び出します。

先頭に戻る

Boot Agent の更新

Boot Agent の更新：

- セッション ハンドルは必要ありません。
- PreCheck は使用できません。
- 処理を実行するために、**Apply** を呼び出す必要はありません。

クライアントは、メソッドの呼び出しを使用して Boot Agent のイメージを更新できます。フラッシュ イメージの読み取り/書き込みを行うには、まずアダプタの IANet_BootAgent のインスタンスを入手します。たとえば、オブジェクト パスを使って **IWbemServices::GetObject** を使用します。

次に、ReadFlash() を実行して既存のフラッシュ ブート ROM イメージを読み取るか、ProgramFlash() を実行してフラッシュ ブート ROM イメージを更新します。この操作が失敗した場合は、IANet_ExtendedStatus でエラー コードを確認してください。

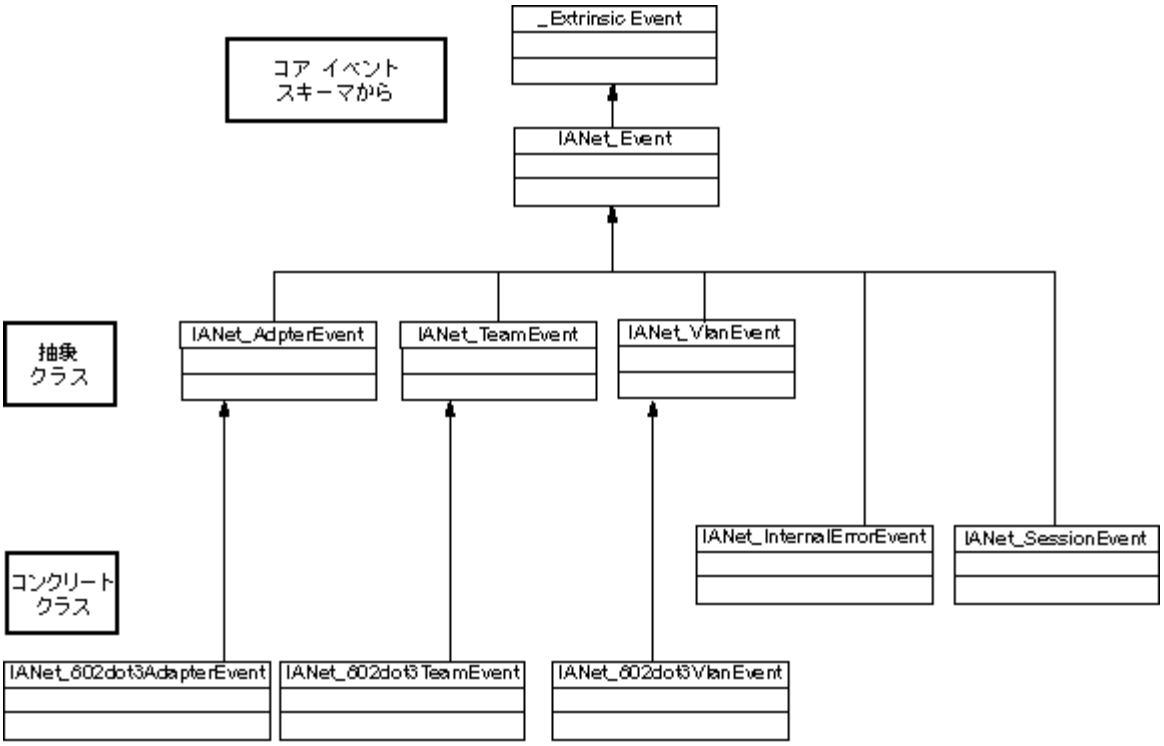
タスク	**WMI** メソッド	結果	注釈
アダプタのブート ROM イメージの更新または挿入	uint32 ProgramFlash([IN, ValueMap {"0","1"} , Values {"Check Version", "Write Flash"}:Amended] uint32 Action, [IN] uint8 NewFlashData[], [OUT] string strErrorMessage);	「Check Version」アクションを指定した場合、NewFlashData[] で更新するブート ROM イメージが現在 NIC にあるものよりも古いと、このメソッドでは警告が返されます。 「Write」アクションを指定した場合は、このメソッドによって NIC の FLASH ROM が NewFlashData[] で更新されます。	このメソッドは、NIC の Flash ROM の更新に使用されます。これにより、フラッシュの更新中は NIC とネットワークの通信が停止します。
ブート ROM イメージの読み取り	uint32 ReadFlash([OUT] uint8 FlashData[]);	FlashData[] には、NIC の Flash ROM イメージが含まれます。	このメソッドでは、NIC の Flash ROM が読み取られ、これをファイルに保存することができます。

制限と免責条項をすべてお読みください。

目次に戻る 先頭に戻る

目次に戻る

イベントの通知：Intel(R) PRO ネットワーク アダプタ WMI および CDM プロバイダ ユーザ ガイド



先頭に戻る

コンクリート イベント クラス

IANet_802dot3AdapterEvent

用途
このイベントは、アダプタのステータスや設定の変更について、クライアントに通知します。

トリガ
このイベントは、アダプタのステータスが変更された後、またはアダプタの設定を変更して、**Apply** を呼び出した後に発生します。

イベント データ
AdapterPath が、イベントを引き起こしたアダプタのオブジェクト パスを含みます。

IANet_802dot3TeamEvent

用途
このイベントは、チームのステータスや設定の変更について、クライアントに通知します。

トリガ
このイベントは、次の場合に発生します。

- アダプタのステータスが変更された後。
- チーム設定を変更し、**Apply** を呼び出した後。
- チームのコンフィグレーションを変更し、**Apply** を呼び出した後。

イベント データ
TeamPath が、イベントを引き起こしたチームのオブジェクト パスを含みます。

IANet_802dot3VlanEvent

用途
このイベントは、VLAN のステータスや設定の変更について、クライアントに通知します。

トリガ
このイベントは、次の場合に発生します。

- VLAN のステータスが変更された後。
- VLAN 設定を変更し、**Apply** を呼び出した後。
- VLAN のコンフィグレーションを変更し、**Apply** を呼び出した後。

イベント データ
VlanPath が、イベントを引き起こした VLAN のオブジェクト パスを含みます。

先頭に戻る

イベントの登録

アプリケーションは、**IWbemServices:: ExecNotificationQuery** または **IWbemServices:: ExecNotificationQueryAsync** を使用して、イベントの通知を要求します。.次に、イベント通知クエリの例をあげます。使用できるクエリには、このほかにも多数のものがあります。

- **SELECT * FROM IANet_Event** - すべてのイベントを要求します。
- **SELECT * FROM IANet_AdapterEvent** - すべてのアダプタ イベントを要求します。
- **SELECT * FROM IANet_TeamEvent** - すべてのチーム イベントを要求します。
- **SELECT * FROM IANet_SessionEvent** - すべてのセッション イベントを要求します。
- **SELECT * FROM IANet_VlanEvent** - すべての VLAN イベントを要求します。
- **SELECT * FROM IANet_InternalErrorEvent** - すべての内部イベントを要求します。
- **SELECT * FROM IANet_AdapterEvent WHERE AdapterPath={IANet_EthernetAdapter object path}** - 特定のアダプタのアダプタ イベントを要求します。

制限と免責条項をすべてお読みください。

目次に戻る 先頭に戻る

目次に戻る

最適化された WQL クエリ:Intel(R) PRO ネットワーク アダプタ WMI および CDM プロバイダ ユーザ ガイド

概要

WMI Provider は、アプリケーションがクエリを使用して設定を取得できるように最適化されています。WMI Provider は次のクエリを認識でき、一致するオブジェクトのみを返します。その他のすべてのクエリでは、WMI Provider ですべてのオブジェクトのすべての設定が取得され、CIMOM によってアプリケーションに届く前にこれらの設定がフィルタにかけられます。このため、複数のアダプタ、チーム、VLAN がある場合は、必要なデータの取得に数秒の遅延が生じることがあります。

先頭に戻る

特定のアダプタ、VLAN、チームの設定の取得

次のクエリでは、特定のアダプタ、VLAN、またはチームの設定のみが取得されます。WQL では、WHERE 句にこのほかの句を追加することはできません。

ASSOCIATORS OF {IANet の構成パス} WHERE AssocClass = IANet_SettingContext

先頭に戻る

単独の設定の取得

次のクエリでは、オブジェクトの単独の設定を取得することができます。すべての設定をクエリして取得する必要はありません。

SELECT * FROM [設定クラス] WHERE ParentId="[デバイス ID]" AND ParentType="[タイプ]" AND Caption="[設定名]"

注：

- クラスは、基本クラスではなく、正確な設定クラスであることが必要です (例：IANet_SettingInt)。
- 使用できる ParentType は、「NIC」、「Team」、 「VLAN」、または 「BootAgent」 です。
- ParentId は、オブジェクトを設定で一意に定義する GUID です (アダプタでは、これは DeviceId です)。
- これは、オブジェクトに関連する設定の取得に推奨する方法ではありません。できれば関連を使用するようにします。ただし、WQL では複雑なクエリはサポートされていないため、十分ではないことがあります。つまり、WQL では「ASSOCIATORS OF {IANet_Configuration path} WHERE AssocClass = IANet_SettingContext AND Caption="[SETTING NAME]"」というクエリはサポートされていません。

制限と免責条項をすべてお読みください。

目次に戻る 先頭に戻る

目次に戻る

診断：Intel(R) PRO ネットワーク アダプタ WMI および CDM プロバイダ ユーザ ガイド

診断クラス

IANet_DiagTest

用途
IANet_DiagTest は、CIM_DiagnosticTest のサブクラスです。このクラスでは、Intel(R) PROSet でサポートされているイーサネット アダプタの診断テストを実行およびコントロールするための一般的な手段が提供されます。スーパークラス CIM_DiagnosticTest は CIM 対応システムのすべてのコンピュータ ハードウェアのテストを一般的にサポートするように設計されています。クラスのプロパティは基本的に記述的で、テストのメカニズムは明示されたメソッドによって提供されます。

インスタンス
キーは Name で、このプロバイダでは参照されているアダプタの GUID でのテストの数字によるインデックスをつなげたものになります (例：1@{12345678-9ABC-DEF0-1234-123456789012})。このキー値は、アダプタとテストを参照するすべての情報が RunTest とその他のメソッドにオブジェクト パラメータとして渡されるため、ある意味で冗長な情報です。それでも、インスタンスはメソッドへのパラメータと一致することが必要で、一致しない場合はプロバイダによってコマンドが拒否されます。キャプション プロパティでは、このインスタンスが実行するテストの名前が提供されます。その他のプロパティでは、その他の記述的な情報と、実行時の情報が提供されます。

インスタンスの作成
IANet_DiagTest のインスタンスを作成することはできません。

インスタンスの削除
IANet_DiagTest のインスタンスを削除することはできません。

プロパティの変更
このクラスには、ユーザが変更できるプロパティはありません。

関連

- IANet_DiagTestForMSE のインスタンスは、IANet_ManagedSystemElement で IANet_DiagTest と関連をもちます。IANet_ManagedSystemElement は IANet_EthernetAdapter のインスタンスになります。
- IANet_DiagResultForTest のインスタンスは、IANet_DiagnosticResult で IANet_DiagTest と関連をもちます。
- IANetDiagSettingForTest のインスタンスは、IANet_DiagSetting で IANet_DiagTest と関連をもちます。

サポートされていないプロパティ
Install Date、OtherCharacteristicDescription

メソッド
このクラスでは、次のメソッドがサポートされます。

- RunTest - 次のものを参照する 3 つのパラメータで定義されるテストを実行します。
  - SystemElement - アダプタを定義します。このアダプタで SystemElement のインスタンスを参照してテストが実行されます。SystemElement は常に IANet_EthernetAdapter のサブクラスです。
  - Setting - 実行するテストを定義します。また、CIM_DiagnosticSetting のインスタンスを参照して実行方法を定義します。CIM_DiagnosticSetting は常に IANet_DiagSetting のサブクラスです。
  - DiagnosticResult - クラス CIM_DiagnosticResult のインスタンスを定義します。このクラスは常にクラス

IANet_DiagResult です。
- DiscontinueTest - CIM_ManagedSystemElement と CIM_DiagnosticResult を参照する 2 つのパラメータで定義される実行中の診断の停止を試みます。これらのパラメータは、RunTest と同様に機能します。3 つめのパラメータ TestingStopped では、テストを停止するコマンドに成功したかどうかを示すブール値が返されます。
- ClearResults - 次のパラメータを使用して、テスト結果をクリアします。
  - SystemElement
  - ResultsNotCleared

  参照されるパラメータ ManagedSystemElement は、このオブジェクトのパスとともに、削除される DiagnosticResultForMSE を参照します。また、DiagnosticResultForMSE で参照される DiagnosticResult オブジェクトのすべての参照も削除されます。さらに、DiagnosticResultForTest のすべてのインスタンス (削除される DiagnosticResult オブジェクトを参照) も削除されます。最後に、クリアできない DiagnosticResults のキーが、文字列配列 Output のパラメータ ResultsNotCleared にリストされます。

クラス階層

CimV2 用。未使用のプロパティとメソッドはあげてありません。

- CIM_ManagedElement:
  Caption
  Description
- CIM_ManagedSystemElement:
  Install Date
  Name
  Status
- CIM_LogicalElement
- CIM_Service:
  - キー
    - Name (string)
  - プロパティ
    - Caption (文字列)
    - CreationClassName (文字列)
    - Description (文字列)
    - Started (ブール値)
    - StartMode (文字列)
    - Status (文字列)
    - SystemCreationClass (文字列)
    - SystemName (文字列)
- CIM_DiagnosticTest:
  - プロパティ
    - Characteristics (uint16 配列)
    - IsInUse (ブール値)
    - ResourcesUsed (uint16 配列)
  - メソッド
    - RunTest
    - ClearResults
    - DiscontinueTest

WbemTest での RunTest やその他のメソッドの実行

MOF ファイルの RunTest メソッドは次のとおりです。

```
uint32
RunTest([IN] CIM_ManagedSystemElement ref SystemElement,
[IN] CIM_DiagnosticSetting ref Setting,
[OUT] CIM_DiagnosticResult ref Result);
```

最初の 2 つのパラメータは In パラメータです。参照されている両方のオブジェクトのオブジェクト パスを取得することが必要です。また、RunTest オブジェクトをエクスポートする IANet_DiagTest オブジェクトのオブジェクト パスも取得してください。

□□□ メインの WBEM テスト ダイアログ ボックスで **[Connect** (接続)] をクリックします。
□□□ 適切な "サーバ\ネームスペース" を入力します。サポートされているネームスペースは、IntelNCS と CimV2 です。
□□□ WBEM テストの **[Enum Instances**(インスタンスの列挙)] ボタンをクリックし、「IANet_DiagTest」と入力します。

□□□ IANet_DiagTest のインスタンスをダブルクリックします。名前には「X@[AdapterGUID}」の形式を使用します。X にはテスト名、AdapterGUID にはアダプタ名 (IANet_EthernetAdapter の Name キーと同じ) を入れます。
□□□ 次に、取得されるオブジェクト パスの例をあげます。
\\MYCOMPUTER\root\Cimv2:IANet_DiagTest.Name="1@{4A0CDABE-F6C3-45D0-B60D-F6E7BAFA2C2C}"
□□□ オブジェクト パスを保存します。
□□□ WBEM テストの **[Enum Instances** (インスタンスの列挙)] ボタンをクリックし、「IANet_EthernetAdapter」と入力します。
□□□ テストするアダプタをダブルクリックします。
□□□ 次に、取得されるオブジェクト パスの例をあげます。
\\MYCOMPUTER\root\cimv2:IANet_EthernetAdapter.DeviceID="{4A0CDABE-F6C3-45D0-B60D-F6E7BAFA2C2C}"
□□□□ オブジェクト パスを保存します。
□□□□ WBEM テストの **[Enum Instances** (インスタンスの列挙)] ボタンをクリックし、「IANet_DiagSetting」と入力します。
□□□□ アダプタ/テストの組み合わせを示す設定をダブルクリックします。
□□□□ 次に、取得されるオブジェクト パスの例をあげます。
\\MYCOMPUTER\root\cimv2:IANet_DiagSetting.SettingID="1@{4A0CDABE-F6C3-45D0-B60D-F6E7BAFA2C2C}"
□□□□ オブジェクト パスを保存します。
□□□□ メインの WBEM テスト ダイアログ ボックスで **[Execute Method (**メソッドの実行**)]** をクリックします。
□□□□ ダイアログ ボックスに IANet_DiagTest オブジェクトのパスを貼り付けます。[**OK**] をクリックします。
□□□□ メソッドの下にあるドロップダウン ボックスでテストを選択します。
□□□□ **[Edit In Parameters** (In パラメータの編集)] ボタンをクリックします。
□□□□ RunTest では、Setting と SystemElement が In パラメータです。保存しておいた Setting と Adapter のオブジェクト パスを貼り付けて、閉じます。
□□□□ **Execute** (実行)] ボタンをクリックします。
□□□□ In パラメータと同様の方法で、IANet_DiagResult クラスを列挙します。
□□□□ 必要に応じて、選択された結果オブジェクトを検討します。

IANet_DiagSetting

用途
IANet_DiagSetting のインスタンスは、特定のランタイム診断テストの指令を提供します。使用される指令は全テストに共通で、スーパークラス CIM_DiagnosticSetting にバインドされます。これらには ReportSoftErrors や HaltOnError などのプロパティが含まれます。サブクラス IANet_DiagSetting には追加のプロパティはありません。

インスタンスの作成
このクラスのインスタンスを作成することはできません。

インスタンスの削除
このクラスのインスタンスを削除することはできません。

プロパティの変更
UpdateInstanceAsync が実装され、テスト パラメータを "Halt On Error (エラー時に中断)"、"Report Soft Errors (ソフト エラーのレポート)"、"Report Status Messages (レポート ステータスのメッセージ)"、"Quick Mode (クイック モード)"、"Test Warning Level (警告レベルのテスト)" および "Percent Of Test Coverage (テストのカバー範囲の割合)" に設定するのに使用できます。

関連
IANetDiagSettingForTest のインスタンスは、IANet_DiagSetting で IANet_DiagTest と関連をもちます。

サポートされていないプロパティ
次のプロパティは NCS ではサポートされていません。

- Caption
- 説明

メソッド
なし

クラス階層

CimV2 用。未使用のプロパティとメソッドはあげてありません。

- CIM_ManagedElement

- CIM_Setting：
  - プロパティ
  - SettingID
  - メソッド (サポートされているものはなし)
  - VerifyOKToApplyToMSE
  - ApplyToMSE
  - VerifyOKToApplyToCollection
  - ApplyToCollection
  - VerifyOKToApplyIncrementalChangeToMSE
  - ApplyIncrementalChangesToMSE
  - ApplyIncrementalChangeToCollection
- CIM_DiagnosticSetting：
  - キー
    - SettingID (文字列)
  - プロパティ
    - TestWarningLevel (uint16)
    - ReportSoftErrors (ブール値)
    - ReportStatusMessages (ブール値)
    - HaltOnError (ブール値)
    - QuickMode (ブール値)
    - PercentOfTestCoverage (uint8)

IANet_DiagResult

用途
IANet_DiagResult のインスタンスは、特定のアダプタの特定のテストの実行に関する結果のデータを表示します。このクラスのインスタンスは、IANet_DiagTest と IANet_DiagSetting のインスタンスに完全に対応します。

インスタンス
IANet_DiagResult のインスタンスは、特定のアダプタの特定のテストの実行に関する結果に対応します。キーの形式は、IANet_DiagTest や IANet_DiagSetting と同じです。このインスタンスは、定義されたプロパティにあてはまらない任意のテスト結果を任意のデータとして保管し、TestResults Array に配置することができます。アダプタで新しいテストを実行するたびに、新規のインスタンスが、アダプタとテストの組み合わせの一致するテスト結果の既存のインスタンスを上書きします。

インスタンスの作成
このクラスのインスタンスを作成することはできません。

インスタンスの削除
このクラスのインスタンスを削除することはできません。

インスタンスの変更
このクラスのインスタンスを変更することはできません。

関連
IANet_DiagResultForTest のインスタンスは、IANet_DiagnosticResult で IANet_DiagTest と関連をもちます。

サポートされていないプロパティ
次のプロパティは NCS ではサポートされていません。

- EstimatedTimeOfPerforming
- HaltOnError
- OtherStateDescription
- ReportSoftErrors
- TestWarningLevel

メソッド
なし

クラス階層

CimV2 用。未使用のプロパティとメソッドはあげてありません。

- CIM_DiagnosticResult：
  - キー
    - DiagnosticCreationClassName (文字列)
    - DiagnosticName (文字列)
    - ExecutionID (文字列)
    - DiagSystemCreationClassName (文字列)
    - DiagSystemName (文字列)
  - プロパティ
    - TimeStamp (文字列)
    - IsPackage (ブール値)
    - TestStartTime (日付時間)
    - TestCompletionTime (日付時間)
    - TestState (uint16)
    - TestResults (文字列)
    - PercentComplete (uint8)
- IANet_DiagResult

先頭に戻る

レジストリのエントリ

インストール時に、**HKLM\Software\Intel\NETWORK_SERVICES\NCS\NcsDiag** に以下のレジストリ エントリが入力されます。これらのキーと値は、診断テストの実行をコントロールし、次のように定義されます。

次の表に、キーの値、タイプ、および使用方法に関する簡単な説明をリストします。

値	タイプ	デフォルト	使用方法
Check Time	REG_DWORD	2秒	送信機や応答機のテストの完了をチェックする間隔。
Enable	REG_DWORD	0	結果のログ ファイルを使用する (1) または使用しない (0)。
FileAppend	REG_DWORD	1	結果のログ ファイルを既存のログ ファイルに追加する (1)。0 の場合は、既存のファイルが削除されます。
LogFileName	REG_SZ	NcsDiag.log	結果のログ ファイルの名前。
MaxFileSize	REG_DWORD	0x10000	結果のログ ファイルの最大サイズ。
MaxPktsRcvd	REG_DWORD	200	Quick Mode (IANet_DiagSetting から) の場合、受信パケットの数がこの値よりも大きい際に送受信テストを中止します。
TimeoutSndRsp	REG_DWORD	100	テスト時間 (秒) がこの値を超えた場合、テストを終了します。

先頭に戻る

ログ記録：

結果のログ

結果のログは、主に情報を表示します。これらの情報は IANet_DiagResult オブジェクトから取得することもできます。両者の違いは、CIM ブラウザで入手する情報では、特定のアダプタにおける特定のテストの最後の結果のみが表示される点です。テストを繰り返し行うと、以前のテスト結果は新しいテスト結果で上書きされます。結果のログは、このようなテストを繰り返して実行する場合の設定や表示に便利です。

結果ログの使用

結果のログを有効にするには：

□□□ レジストリ キー「**HKLM\Software\Intel\NETWORK_SERVICES\NCS\NCSDiag**」で、Enable を 1 に設定します。
□□□ LogFileName の値を希望するログ ファイルの名前に変更するか、デフォルトの **NcsDiag.log** をそのまま使用します。
□□□ ログ ファイルは、InstalledDir の値が示すディレクトリに保存されます。

先頭に戻る

関連クラス

関連クラス	参照プロパティ/値	参照プロパティ/値
IANet_DiagTestForMSE	Antecedent = IANet_DiagTest	Dependent = IANet_EthernetAdapter
IANet_DiagResultForTest	DiagnosticResult = IANet_DiagResult	DiagnosticTest = IANet_DiagTest
IANet_DiagSettingForTest	Element = IANet_DiagTest	Setting = IANet_DiagSetting
IANet_DiagResultForTest	DiagnosticResult = IANetDiagResult	DiagnosticTest = IANet_DiagTest

先頭に戻る

テスト

実装したテストは、単独のマシンか、2 台のマシンで実行できます。CDM Provider はテストの内容に依存しない、一般的なテスト手段として設計されているため、テストの詳細な説明は本文書では扱いません。ただし、コードの一部で依存性が生じる部分があるため、この項ではその部分について説明します。

単独のアダプタのテスト

次のテストは単独のアダプタで実行され、ほかのアダプタとの通信は必要としません。

- EEPROM
- Control Registers
- MAC Loopback
- PHY Loopback
- Link

これらのテストからのエラー メッセージはすべて、スタック層の下層への呼び出しから返された HRESULT エラー コードからのものです。エラー コードはエラー コードとして内部で保管され、IANet_DiagResult オブジェクトの参照が列挙によって解除されるまで、または管理アプリケーションからオブジェクトへの呼び出しが受信されるまで、エラー メッセージに変換されることはありません。

2 つのアダプタを必要とするテスト

送信機と応答機のテストは、相互に依存するテストで、1 つのアダプタ (送信機) がパケットを別のアダプタ (応答機) に送信し、応答機から送信機にパケットが送り返されることで、ループが完了します。これらは、Intel(R) PROSet から実行できるのと同じテストです。ただし、Intel PROSet では CDM は使用されず、同じマシンに 2 つのテストを同時に実行することはできません。CDM では、同じマシンに複数のテストを同時に実行できます。

送信機/応答機テスト

送信機/応答機では、2 つの Intel アダプタが必要で、1 つは送信機、もう 1 つは応答機になります。このテストは、2 つめのスレッドに基づいてテストが実行され、このスレッドがテストが完了条件を満たして完了するまで、または 1 つめのスレッドによって停止されるまで実行し続ける、唯一のテストです。完了条件とは、テスト時間か、受信パケット数に基づくタイムアウトです。これらの値は両方とも、レジストリから取得されます。Quick モードがオンの場合は、テストは受信パケット数に基づいてのみ終了できます。Quick モードとは、IANet_DiagSetting クラスのプロパティで、アダプタごとに設定できます。CDM 応答機は PROSet 応答機に応答し、CDM 応答機は PROSet 応答機に応答します。

送信機/応答機テストで返されるエラー値には、2 つのタイプがあります。まず、下層レイヤから Error Code (HRESULT) が返されることがあります。次に、テストが実行中で、エラー コードが返されて早期に停止されない限り、テスト スレッドから次のような中間テスト結果や最終テスト結果が返されます。

- Link Status
- Using Auto-Negotiation
- Collisions
- Packets Received
- Packets Received Total
- Packets Sent
- Transmit Oks
- Receive Oks
- Transmit Errors
- Receive Errors
- Collisions
- Diagnostic Phase

制限と免責条項をすべてお読みください。

目次に戻る 先頭に戻る

IANet_DiagTest のメソッドの実行:Intel(R) PRO ネットワーク アダプタ WMI および CDM プロバイダ ユーザ ガイド

WbemTest での RunTest やその他のメソッドの実行

MOF ファイルの RunTest メソッドは次のとおりです。

```
uint32
RunTest([IN] CIM_ManagedSystemElement ref SystemElement,
[IN] CIM_DiagnosticSetting ref Setting,
[OUT] CIM_DiagnosticResult ref Result);
```

最初の 2 つのパラメータは In パラメータです。参照されている両方のオブジェクトのオブジェクト パスを入手することが必要です。また、RunTest オブジェクトをエクスポートする IANet_DiagTest オブジェクトのオブジェクト パスも取得してください。

□□□ メインの WBEM テスト ダイアログ ボックスで [**Connect** (接続)] をクリックします。
□□□ 適切な "サーバ\ネームスペース" を入力します。サポートされているネームスペースは、IntelNCS と CimV2 です。
□□□ WBEM テストの [**Enum Instances** (インスタンスの列挙)] ボタンをクリックし、「`IANet_DiagTest`」と入力します。
□□□ IANet_DiagTest のインスタンスをダブルクリックします。名前には「X@[AdapterGUID}」の形式を使用します。X にはテスト名、AdapterGUID にはアダプタ名 (IANet_EthernetAdapter の Name キーと同じ) を入れます。
□□□ 次に、取得されるオブジェクト パスの例をあげます。
`\\MYCOMPUTER\root\Cimv2:IANet_DiagTest.Name="1@{4A0CDABE-F6C3-45D0-B60D-F6E7BAFA2C2C}"`
□□□ オブジェクト パスを保存します。
□□□ WBEM テストの [**Enum Instances** (インスタンスの列挙)] ボタンをクリックし、「`IANet_EthernetAdapter`」と入力します。
□□□ テストするアダプタをダブルクリックします。
□□□ 次に、取得されるオブジェクト パスの例をあげます。
`\\MYCOMPUTER\root\cimv2:IANet_EthernetAdapter.DeviceID="{4A0CDABE-F6C3-45D0-B60D-F6E7BAFA2C2C}"`
□□□□ オブジェクト パスを保存します。
□□□□ WBEM テストの [**Enum Instances** (インスタンスの列挙)] ボタンをクリックし、「`IANet_DiagSetting`」と入力します。
□□□□ アダプタ/テストの組み合わせを示す設定をダブルクリックします。
□□□□ 次に、取得されるオブジェクト パスの例をあげます。
`\\MYCOMPUTER\root\cimv2:IANet_DiagSetting.SettingID="1@{4A0CDABE-F6C3-45D0-B60D-F6E7BAFA2C2C}"`
□□□□ オブジェクト パスを保存します。
□□□□ メインの WBEM テスト ダイアログ ボックスで [**Execute Method** (メソッドの実行)] をクリックします。
□□□□ +ダイアログ ボックスに IANet_DiagTest オブジェクトのパスを貼り付けます。[**OK**] をクリックします。
□□□□ メソッドの下にあるドロップダウン ボックスでテストを選択します。
□□□□ [**Edit In Parameters** (In パラメータの編集)] ボタンをクリックします。
□□□□ RunTest では、Setting と SystemElement が In パラメータです。保存しておいた Setting と Adapter のオブジェクト パスを貼り付けて、閉じます。
□□□□ [**Execute** (実行)] ボタンをクリックします。
□□□□ In パラメータと同様の方法で、IANet_DiagResult クラスを列挙します。
□□□□ 必要に応じて、選択された結果オブジェクトを検討します。

制限と免責条項をすべてお読みください。

目次に戻る

CIM クラスの概要：Intel(R) PRO ネットワーク アダプタ WMI および CDM プロバイダ ユーザ ガイド

クラス	作成	削除	実装されているメソッド	設定可能なプロパティ	サポートされていないプロパティ	インスタンスの数	関連する構成クラス
IANet_8023ad1Q1VLANService	不可	不可	CreateVLAN	GRVPEnabled、JoinTime	Description、Install Date、Started、StartMode、Status	VLAN をサポートするチームおよびシステム上で 1 つのインスタンス	IANet_Device8023ad1Q1VLANServiceImplementation、IANet_VLANFor
IANet_AdapterConfiguration	不可	不可	なし	なし	なし	各アダプタおよびシステム上で 1 つのインスタンス	このクラスは IANet_EthernetAdapter を IANet_Configuration と関連付けます。
IANet_AdapterDevice	不可	不可	なし	なし	なし	ゴースト以外の各アダプタおよびシステム上で 1 つのインスタンス	このクラスは IANet_EthernetAdapter を IANet_EthernetPCIDevice と関連付けます。
IANet_AdapterProtocolImplementation	不可	不可	なし	なし	なし	アダプタおよびハイパーバイザーから各 IP プロトコル エンドポイントに 1 つのインスタンス	このクラスは IANet_EthernetAdapter を IANet_IPProtocolEndpoint と関連付けます。
IANet_BootAgent	不可	不可	ProgramFlash ReadFlash	なし	Caption、Description、InstallDate、Started、StartMode、Status	Boot Agent 機能をサポートする各アダプタおよびシステム上で 1 つのインスタンス	IANet_DeviceBootServiceImplementation、IANet_BootAgentConfiguration
IANet_BootAgentConfiguration	不可	不可	なし	なし	なし	各 Boot Agent ごとおよびシステム上で 1 つのインスタンス	このクラスは IANet_BootAgent を IANet_Configuration と関連付けます。
IANet_Configuration	不可	不可	なし	なし	なし	各アダプタ、VLAN、およびシステム上で 1 つのインスタンス	IANet_AdapterConfiguration、IANet_VLANConfiguration、IANet_SettingControl
IANet_Device8023ad1Q1VLANServiceImplementation	不可	不可	なし	なし	なし	VLAN をサポートするアダプタおよびチームごとおよびシステム上で 1 つのインスタンス	このクラスは IANet_EthernetAdapter を IANet_8023ad1Q1VLANService と関連付けます。
IANet_DiagTest	不可	不可	RunTest、DiscontinueTest、ClearResults	なし	InstallDate、OtherCharacteristicsDescription	アダプタおよびシステムの各組み合わせに 1 つ	IANet_DiagTestForMSIE、IANet_DiagResultForTest、IANet_DiagSettingForTest
IANet_DiagSetting	不可	不可	HaltOnError、ReportSubErrors、ReportStatusMessages、QuickMode、PercentOfTestCoverage、TestWarningLevel	Caption、Description	なし	アダプタおよびシステムの各組み合わせに 1 つ	IANet_DiagSettingForTest
IANet_DiagResult	不可	不可	なし	EstimatedTimeOfPerforming、HaltOnError、OtherStateDescription、ReportSubErrors、TestWarningLevel	なし	アダプタおよびシステムの各組み合わせに 1 つ	IANet_DiagResultForTest、IANet_DiagResultForMSIEM
IANet_EthernetAdapter	不可	可	IdentifyAdapter HaltVLANs GPowerMgmtSupported GetPowerUsage GetPowerOptions SetPowerUsageOptions SetPowerUsageCyclons TestCable AdvancedTestCable TestLinkSpeed	Autosense（設定として）、ErrorCleared、OtherIdentifyingInfo、IdentifyingDescriptions、InstallDate、LastErrorCode、MaxDataSize、MaxQueueTime、PowerManagementCapabilities（メソッドとして）、PowerManagementSupported（メソッドとして）、PowerOnHours、ShortFramesReceived、SymbolErrors、TotalPowerOnHours	Intel PROSet サポートされ、インストールされているアダプタ、ゴースト アダプタ、および VLAN 以外のすべて	IANet_AdapterProtocolImplementation、IANet_AdapterDevice、IANet_AdapterConfiguration、IANet_TeamedMemberAdapter、IANet_NetworkVirtualAdapter、IANet_Device8023ad1Q1VLANServiceImplementation、IANet_DeviceBootServiceImplementation
IANet_EthernetPCIDevice	不可	不可	なし	なし	AdditionalAvailability、CapabilityDescriptions、Caption、Description、DeviceSelectTiming、ErrorCleared、ErrorDescription、IdentifyingDescription、InstallDate、LastErrorCode、MaxNumberController、MaxQueueTime、Name、OtherIdentifyingInfo、PowerManagementCapabilities、PowerManagementSupported、PowerOnHours、ProtocolDescription、ProtocolSupported、SelfTestEnabled、Status、StatusInfo、TimeOfLastReset、TotalPowerOnHours	Intel PROSet がサポートするイーサネット アダプタ PCI カードごとに 1 つのインスタンス	IANet_AdapterDevice
IANet_IPProtocolEndpoint	不可	不可	なし	なし	Caption、Description、InstallDate、NameFormat、OtherTypeInformation、Status	IP プロトコル スタックがサポートされていないエンドポイント以外に 1 つのインスタンス	IANet_AdapterProtocolImplementation、IANet_VLANProtocolDependency
IANet_NetService	不可	不可	GetSessionHandle、Apply、ReleaseSessionHandle、Cancel	なし	Caption、Description、Install Date、Started、Start Mode、Status	1 つの SA	なし
IANet_NetworkVirtualAdapter	不可	不可	なし	なし	なし	各チームに 1 つのインスタンス	このクラスは IANet_TeamOfAdapters を IANet_EthernetAdapter と関連付けます。
IANet_PCIDevice	不可	不可	なし	なし	AdditionalAvailability、CapabilityDescriptions、Caption、DeviceSelectTiming、ErrorCleared、ErrorDescription、IdentifyingDescription、InstallDate、LastErrorCode、MaxNumberController、MaxQueueTime、Name、OtherIdentifyingInfo、PowerManagementCapabilities、PowerManagementSupported、PowerOnHours、ProtocolDescription、ProtocolSupported、SelfTestEnabled、TimeOfLastReset、TotalPowerOnHours	システム内のネットワーク アダプタ PCI カードごとに 1 つのインスタンス	
IANet_Setting	不可	不可	なし	なし	SettingID		IANet_SettingControl
IANet_SettingContext	不可	不可	なし	なし	なし	各設定ごとに 1 つのインスタンス	このクラスは IANet_Setting を IANet_Configuration と関連付けます。
IANet_SettingInt	不可	不可	なし	CurrentValue	SettingID	各整数設定ごとに 1 つのインスタンス	IANet_SettingControl
IANet_SettingMultiSelection	不可	不可	なし	CurrentValue	SettingID	各複数選択設定ごとに 1 つのインスタンス	IANet_SettingControl
IANet_SettingSlider	不可	不可	なし	CurrentValue	SettingID	各スライダ設定ごとに 1 つのインスタンス	IANet_SettingControl
IANet_SettingString	不可	不可	なし	CurrentValue	SettingID	各文字列設定ごとに 1 つのインスタンス	IANet_SettingControl
IANet_TeamedMemberAdapter	可	可	なし	AdapterFunction	PrimaryAdapter、SoopetOfBalancing	チーム内の各アダプタごとに 1 つのインスタンス	このクラスは IANet_TeamOfAdapters を IANet_EthernetAdapter と関連付けます。
IANet_TeamOfAdapters	可	可	TestSwitchConfiguration	TeamingMode	Install Date、Status	各チームに 1 つのインスタンス	IANet_NetworkVirtualAdapter、IANet_TeamedMemberAdapter
IANet_VLAN	不可	可	なし	VLANNumber、Caption	Description、Install Date、StartMode、Status	各 VLAN に 1 つのインスタンス	IANet_VLANFor
IANet_VLANConfiguration	不可	不可	なし	なし	なし	各 VLAN に 1 つのインスタンス	このクラスは IANet_VLAN を IANet_Configuration と関連付けます。
IANet_VLANFor	不可	不可	なし	なし	なし	各 VLAN に 1 つのインスタンス	このクラスは IANet_VLAN を IANet_8023ad1Q1VLANService と関連付けます。
IANet_VLANProtocolDependency	不可	不可	なし	なし	なし	各 VLAN に 1 つのインスタンス	このクラスは IANet_VLAN を IANet_IPProtocolEndpoint と関連付けます。

制限と免責条項をすべてお読みください。

目次に戻る 先頭に戻る

目次に戻る

インテル ソフトウェア使用許諾契約書 （最終、使用許諾契約）：Intel(R) PRO ネットワーク アダプタ WMI および CDM プロバイダ ユーザ ガイド

重要 - コピー、インストール、または使用の前にお読みください。

ソフトウェアおよび関連資料 (以下、総称して 「本ソフトウェア」 といいます) を使用またはロードする前に、以下の条件を注意深くお読みください。本ソフトウェアをロードされますと、お客様は本契約の条件に同意されたことになります。同意されない場合は、本ソフトウェアをインストールまたは使用しないでください。

使用許諾契約：

- ネットワーク管理者には、以下の 「サイト使用許諾契約」 が該当します。
- エンドユーザには、「シングル ユーザ使用許諾契約」 が該当します。
- OEM (Original Equipment Manufacturer) には、「OEM 使用許諾契約」が該当します。

サイト使用許諾契約。お客様は、本ソフトウェアをお客様の組織が使用するためにその組織のコンピュータに本ソフトウェアをコピーすることができ、ソフトウェアのバックアップとして妥当な数のコピーを作成することができます。ただし、以下の条件に従うこととします。

□□□ 本ソフトウェアをインテル コンポーネント製品と関連して使用する場合のみ、使用を許諾します。本使用許諾契約では、インテル製品以外のコンポーネント製品と関連して本ソフトウェアを使用することを許諾するものではありません。
□□□ 本契約書で特に規定する場合を除き、本ソフトウェアのいかなる部分も、複製、変更、貸与、売却、配布、譲渡することは禁じられています。また、お客様は、本ソフトウェアの無断複製を防止することに合意するものとします。
□□□ 本ソフトウェアをリバース エンジニア、逆コンパイル、逆アセンブルすることを禁じます。
□□□ 本ソフトウェアの使用許諾を第三者に与えたり、複数ユーザによる本ソフトウェアの同時使用を許可することはできません。
□□□ 本ソフトウェアは、ここに記載する以外の条件で提供される部分を含む場合があり、その場合はその部分に付属の使用許諾契約が適用されます。

シングルユーザ使用許諾契約。 お客様は、本ソフトウェアを 1 台のコンピュータに非商用目的でコピーすることができ、ソフトウェアのバックアップ コピーを 1 部作成できます。その際、以下の条件が適用されます。

□□□ 本ソフトウェアをインテル コンポーネント製品と関連して使用する場合のみ、使用を許諾します。本使用許諾契約では、インテル製品以外のコンポーネント製品と関連して本ソフトウェアを使用することを許諾するものではありません。
□□□ 本契約書で特に規定する場合を除き、本ソフトウェアのいかなる部分も、複製、変更、貸与、売却、配布、譲渡することは禁じられています。また、お客様は、本ソフトウェアの無断複製を防止することに合意するものとします。
□□□ 本ソフトウェアをリバース エンジニア、逆コンパイル、逆アセンブルすることを禁じます。
□□□ 本ソフトウェアの使用許諾を第三者に与えたり、複数ユーザによる本ソフトウェアの同時使用を許可することはできません。
□□□ 本ソフトウェアは、ここに記載する以外の条件で提供される部分を含む場合があり、その場合はその部分に付属の使用許諾契約が適用されます。

OEM 使用許諾契約。 お客様は、お客様の製品に統合された、または組み込まれた部分として、あるいはお客様の製品の既存のユーザに対するスタンドアロンのソフトウェア メンテナンス アップデート (これ以外のすべてのスタンドアロン製品は除外) としてのみ、本ソフトウェアを複製および配布することができます。ただし、以下の条件に従うものとします。

□□□ 本ソフトウェアをインテル コンポーネント製品と関連して使用する場合のみ、使用を許諾します。本使用許諾契約では、インテル製品以外のコンポーネント製品と関連して本ソフトウェアを使用することを許諾するものではありません。
□□□ 本契約書で特に規定する場合を除き、本ソフトウェアのいかなる部分も、複製、変更、貸与、売却、配布、譲渡することは禁じられています。また、お客様は、本ソフトウェアの無断複製を防止することに合意するものとします。
□□□ 本ソフトウェアをリバース エンジニア、逆コンパイル、逆アセンブルすることを禁じます。
□□□ 本ソフトウェアをお客様の顧客に配布する際には、書面によるライセンス使用許諾契約書に従うものとします。この使用許諾契約書は、ソフトウェアの封印を開けることで、自動的に適用される使用許諾契約書の形式をとることもできます。このような使用許諾契約では、少なくともインテルの本ソフトウェアの所有権を保護するものとします。
□□□ 本ソフトウェアは、ここに記載する以外の条件で提供される部分を含む場合があり、その場合はその部分に付属の使用許諾契約が適用されます。

その他のすべての権利の除外。 本契約書に明示的に規定されているものを除き、インテル社はお客様に対し、インテル社が所有およびコントロールする当社独自の情報や特許権、著作権、マスク ワーク、商標、業務上の機密情報、その他の知的所有権に関し、明示的、暗示的を問わず、いかなる権利やライセンスも付与するものではありません。

ソフトウェアの所有権と著作権。本ソフトウェアの全コピーの所有権はインテルとその納入業者が保留します。本ソフトウェアは著作権が登録されており、米国と諸外国の法律、および国際条約によって保護されています。本ソフトウェアの著作権表示を抹消することはできません。インテルはでは、通知することなくいつでも本ソフトウェアまたはここに記載されている項目に変更を加えることができ、本ソフトウェアをサポートまたはアップデートする義務を負いません。特に明示的に規定がない限り、インテルでは明示・黙示を問わず、インテルの特許、著作権、商標、その他いかなる知的財産権も供与するものではありません。被譲渡人たる第三者が本契約の条件に完全に従うことに合意し、かつお客様が本ソフトウェアのコピーを一切手元に残さないことを条件として、本ソフトウェアを第三者に譲渡することができます。

媒体の限定保証。本ソフトウェアがインテルにより物理的な媒体上で配布された場合、インテルでは配達日より 90 日間、媒体に材質上および物理的な不具合がないことを保証いたします。万一そのような不具合が見つかった場合は、媒体をインテルまでご返送ください。インテルの選択によって、本ソフトウェアの交換または別の配布方法によって対応させていただきます。

その他の保証の除外。上述に規定する保証を除き、本ソフトウェアは「現状のまま」提供されます。商品性の保証、著作権の侵害がないこと、特定目的適合性の保証を含め、その他一切の保証には明示・黙示を問わず応じません。インテルでは、本ソフトウェアに含まれる情報、テキスト、グラフィック、リンク、その他の項目について、その正確性や完全性について一切保証せず、責任を負いません。

免責事項。どのような場合においても、インテルまたはその納入業者は、損害の可能性を指摘する通告が事前にあったとしても、本ソフトウェアの使用またはそれが使用できないことによって生じたいかなる損害 (遺失利益に起因する損害、業務の中断、情報の損失を含むがそれに限られるものではない) に対しても、一切責任を負いません。法管轄区によっては、黙示保証や間接的または付随的損害に対する制限や除外が禁止されている場合があり、したがって、上述の制限はお客様には適用されないことがあります。国または地域によりお客様は他の法的な権利を有する場合があります。

本契約の終了。お客様が本契約の条件に違反した場合、インテルは、本契約書をいつでも解消することができます。契約が解消された場合、お客様は直ちに本ソフトウェアを破棄するか、ソフトウェアのすべてのコピーをインテルに返還するものとします。

準拠法。本契約に関するすべての紛争については、抵触法の原則および物品売買契約に関する国際連合条約を例外とし、カリフォルニア州を準拠法とします。該当する輸出法および規制に違反して本ソフトウェアを輸出することはできません。インテルの承認する代表者が署名した書面がない限り、インテルはその他一切の契約下における責任を負いません。

合衆国政府よる制約。本ソフトウェアは、「制限付き権利」とともに提供されます。政府による使用、複製、開示については、FAR52.227-14 および DFAR252.227-7013 et seq 以降で制定されているとおり、制限に服従します。政府による本ソフトウェアの使用は、インテルの本ソフトウェアへの所有権の確認とみなされます。契約者または製造会社はインテルです。

制限と免責条項をすべてお読みください。

目次に戻る 先頭に戻る

目次に戻る

サポート：Intel(R) PRO ネットワーク アダプタ WMI および CDM プロバイダ ユーザ ガイド

Web サイト

http://www.jp.dell.com

カスタマ サポート技術担当者

本書のトラブルシューティング手順で問題を解決できない場合、技術的な問題は、Dell Computer Corporation までお問い合わせください (ご使用のシステムのマニュアルにあるサポートのセクションを参照してください)。

電話でお問い合わせの前に...

ソフトウェアを実行しているコンピュータ、および製品マニュアルをご用意ください。

技術者が次の質問をする可能性があります。

- 住所および電話番号
- お問い合わせの製品名およびモデル番号
- 製品のシリアル番号およびサービス タグ
- 製品を操作するために使用しているソフトウェア名およびバージョン番号
- ご使用のオペレーティング システム名およびバージョン番号
- コンピュータの種類 (製造元および品番)
- ご使用のコンピュータの拡張ボードまたはアドイン カード
- ご使用のコンピュータのメモリ容量

制限と免責条項をすべてお読みください。

目次に戻る