

Guía del usuario de proveedores WMI y CDM para adaptadores de red Intel® PRO

El proveedor del dispositivo mencionado suministró la información que se incluye a continuación, pero carece de la verificación independiente de la empresa Dell y está sujeta a las [restricciones y renuncia de responsabilidad](#) indicadas en este documento.

- [Introducción](#)
 - [WMI](#)
 - [Características principales](#)
 - [Archivos instalados](#)
 - [Seguridad](#)
 - [Espacio de nombres y contexto](#)
 - [Configuración regional y localización](#)
 - [Informe de errores](#)
 - [Esquema de núcleo](#)
 - [Esquema de adaptadores Ethernet](#)
 - [Esquema de configuración](#)
 - [Esquema de equipo](#)
 - [Esquema VLAN](#)
 - [Obtener la configuración actual](#)
 - [Actualización de la configuración](#)
 - [Notificación de eventos](#)
 - [Consultas WQL optimizadas](#)
 - [Diagnósticos](#)
 - [Ejecución de métodos de IANet_DiagTest](#)
 - [Resumen de clases CIM](#)
 - [Licencia del software](#)
 - [Asistencia al cliente](#)
-

La información de este documento está sujeta a cambios sin previo aviso.

© 2003 Intel Corporation. Reservados todos los derechos.

Marcas comerciales que se utilizan en este texto: *Dell* y el logotipo de *DELL* son marcas comerciales de Dell Computer Corporation; *Intel* es una marca comercial o una marca comercial registrada de Intel Corporation o sus filiales en EE.UU. y en otros países.

* En este documento se podrían utilizar marcas comerciales y nombres comerciales para hacer referencia a las entidades que reclaman las marcas y nombres o sus productos. Intel Corporation rechaza cualquier interés sobre la propiedad de marcas registradas y nombres comerciales distintos de los suyos propios.

Restricciones y renuncia de responsabilidad

El proveedor del dispositivo mencionado suministró la información que se incluye en este documento, incluyendo las instrucciones, advertencias, aprobaciones y certificaciones normativas, pero carece de la verificación y las pruebas independientes de la empresa Dell. Dell no se responsabiliza por ningún daño causado como resultado de seguir o ignorar estas instrucciones.

El autor de todas las descripciones o afirmaciones relacionadas con las propiedades, la capacidad, la velocidad y las calificaciones del producto mencionadas en este documento es el proveedor y no la empresa Dell. Dell específicamente declara que desconoce la precisión de dichas descripciones, así como cuán completas sean o si se han comprobado. Se debe dirigir cualquier pregunta o comentario relacionado con dichas descripciones o afirmaciones directamente al proveedor.

Versión inicial: octubre de 2003

Introducción: Guía del usuario de proveedores WMI y CDM para adaptadores de red Intel® PRO

Visión general

Bienvenidos a la *Guía del usuario de proveedores WMI y CDM para adaptadores de red Intel® PRO*. Este documento describe la vista general de los proveedores WMI y CDM para adaptadores de red Intel® PRO. El proveedor Windows Management Interface (WMI) es un bloque de configuración de red de Network Configuration Services (NCS), que ofrece una manera de instalar y gestionar todas las tecnologías de red de las estaciones terminales Intel, utilizando métodos estándar en la industria. El proveedor Intel PRO Common Diagnostic Model (CDM) es una interfaz API superior que cumple con los estándares CIM 2.5 y WMI. En la interfaz inferior, el proveedor CDM implementa una interfaz de cliente en los niveles inferiores de la pila del software PROSet. Debido a lo anterior, se mantienen todos los mecanismos de integridad de datos de PROSet.

Los proveedores WMI y CDM son conjuntos de componentes de software que implementan las clases de red Intel WMI. Estas clases se basan en el esquema de Desktop Management Task Force (DMTF) CIM versión 2.5

Este documento no repite la información incluida en los archivos Managed Object Format (MOF) suministrados con este producto (por ejemplo, detalles del significado de los atributos individuales que se encuentran en las descripciones de los atributos MOF).


Este documento describe la manera en que una aplicación WMI, tal como sería Intel PROSet, usa las clases para configurar la red de un sistema y cómo una aplicación WMI usa las clases para probar una tarjeta de interfaz de red de Intel. Los lectores deberían estar familiarizados con API WMI y SDK WMI (disponible en <http://www.microsoft.com/>).

[Principio](#)

Documentos relacionados

Se pueden utilizar los documentos siguientes para lograr una comprensión mejor de la tecnología WMI:

- Esquema CIM versión 2.0, 2.2 publicada por Desktop Management Task Force (DMTF). Disponible en <http://www.dmtf.org>.
- Microsoft Windows Management Interface (e información adicional acerca de la manejabilidad). Disponible en http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wmisdk/wmi/wmi_start_page.asp.
- La iniciativa Web-Based Enterprise Management (WBEM) de DMTF. Disponible en <http://www.dmtf.org/standards/wbem>.
- SDK WMI (Implementación Microsoft CIM). Disponible en <http://msdn.microsoft.com/downloads/>.
- Documento informativo del System Diagnostic Model de DMTF. Disponible en <http://www.dmtf.org/standards/documents/CIM/DSP0138.pdf>.

 **AVISO:** Este producto incluye información que se podría utilizar para atacar o inhabilitar sistemas informáticos o redes. Un conocimiento profundo de las características de seguridad del sistema operativo de Microsoft debería ser un requisito previo a la implementación de este producto, y se ruega encarecidamente a los desarrolladores y usuarios que se pongan en contacto con Microsoft en relación con cualquier duda sobre la seguridad que pudiesen tener antes de implementar este producto en un entorno de producción.

Sírvase leer todas las [restricciones y renuncias de responsabilidad](#).

WMI: Guía del usuario de proveedores WMI y CDM para adaptadores de red Intel® PRO

[Visión general](#)

[Common Information Model \(esquema CIM\)](#)

Visión general

Web-Based Enterprise Management (Administración empresarial basada en Web, WBEM) es una iniciativa de Desktop Management Task Force (Comunidad de expertos en administración de escritorio, DMTF) concebida para proporcionarle a los administradores de sistemas de empresas un método rentable y estandarizado para la gestión de las estaciones terminales. La iniciativa WBEM comprende una variedad de tareas, que van desde la configuración simple de una estación de trabajo hasta la administración de empresas de gran escala y por varias plataformas. Una de las piezas clave de la iniciativa es el Common Information Model (Modelo de información común, CIM), un modelo de datos ampliable para la representación de objetos que existen en los entornos administrativos típicos, y siendo la otra el lenguaje Managed Object Format (Formato de objeto gestionado, MOF) para definir y almacenar los datos modelados.

Windows Management Instrumentation (WMI) es una implementación de la iniciativa WBEM para las plataformas Microsoft Windows*.

WMI consiste de tres componentes principales:

- Núcleo: Estos componentes son parte del sistema operativo. Son necesarios para que una aplicación preparada para WMI funcione, y deberán estar instalados para poder utilizar el SDK.
- SDK: El SDK incluye herramientas para navegar por el esquema WMI, ampliar el esquema, crear proveedores, y registrar y utilizar los eventos WMI. Proporciona también documentación muy útil para el desarrollo de aplicaciones que utilizarán el WMI. EL SDK se instala como un componente del proceso de instalación del SDK de la plataforma Microsoft y es compatible con Windows NT4 SP4 o SP5, Windows 2000, Windows Me, Windows XP y Windows Server 2003.
- Herramientas: Las Microsoft WMI Tools (herramientas WMI de Microsoft) les brindan a los desarrolladores todas las herramientas necesarias para crear una generación nueva de aplicaciones y soluciones de gestión. Está repleto de documentación y herramientas para guiarles por el proceso de acceder datos de gestión de WMI.

La arquitectura WMI consiste de los componentes siguientes:

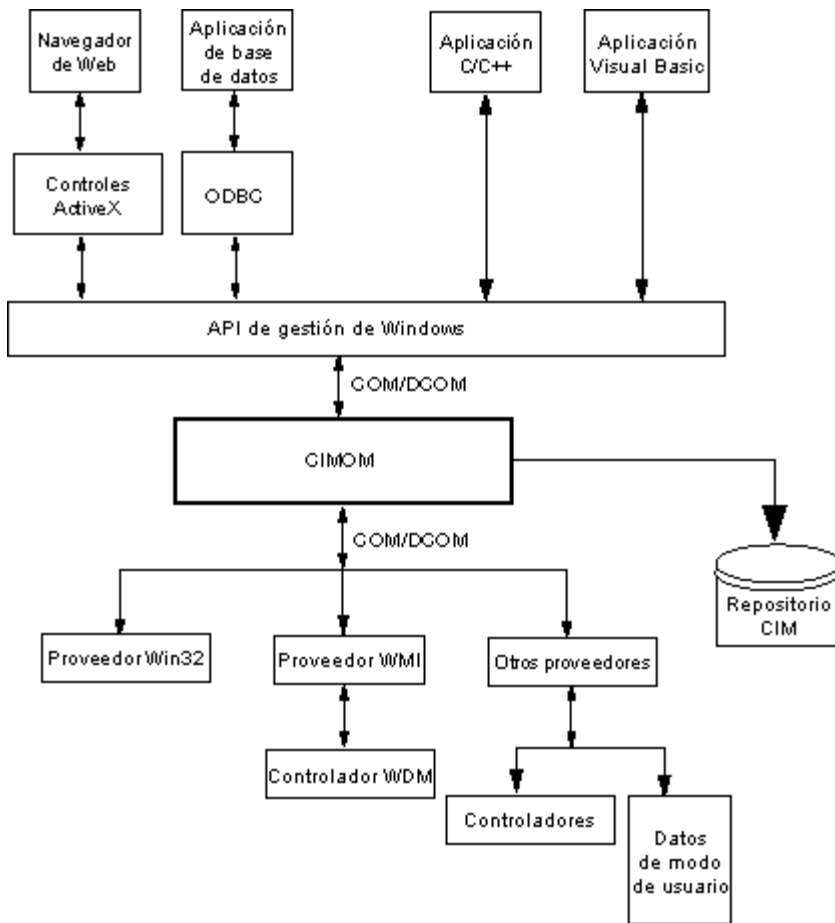
- Aplicaciones de gestión
- Objetos gestionados
- Proveedores
- Infraestructura de gestión (consiste de Windows Management y del repositorio de Windows Management)
- La API Windows Management (la cual utiliza COM/DCOM para habilitar la comunicación entre los proveedores y las aplicaciones de gestión y la infraestructura Windows Management).

El proceso de aplicaciones de gestión o la visualización de datos de objetos gestionados, que son componentes lógicos o físicos de la empresa. Estos componentes se han modelado usando CIM y pueden ser accedidos por las aplicaciones a través de Windows Management. Los proveedores utilizan la API de Windows Management para suministrarle a Windows Management los datos de objetos gestionados, para manejar las solicitudes de las aplicaciones y para generar las notificaciones de eventos.

La infraestructura de gestión consiste de Windows Management (para manejar las comunicaciones entre las aplicaciones gestionadas y los proveedores) y el repositorio de Windows Management (para almacenar los datos). El repositorio de Windows Management contiene datos estáticos de gestión. Los datos dinámicos se generan a solicitud de los proveedores únicamente. Los datos se colocan en el repositorio, ya sea utilizando el compilador de lenguaje MOF o la API de Windows Management.

Las aplicaciones y los proveedores se comunican mediante Windows Management utilizando la API de Windows Management, que proporciona servicios tales como la notificación de eventos y el procesamiento de consultas.

El diagrama que aparece a continuación muestra la interrelación entre los componentes de la arquitectura WMI.



[Principio](#)

Common Information Model (esquema CIM)

El Common Information Model (Modelo de información común, CIM) presenta una visión consistente y unificada de todos los tipos de objetos lógicos o físicos de un entorno gestionado. Los objetos gestionados se representan mediante teoremas orientados a los objetos, como serían las clases. Las clases incluyen las propiedades que describen a los datos y métodos que describen el comportamiento. La CIM está diseñada por la DMTF para ser independiente del sistema operativo o de la plataforma. La tecnología WBEM incluye una ampliación del CIM para las plataformas de sistemas operativos de Microsoft Windows. Si desea más información, consulte el esquema CIM de DMTF en el sitio Web de DMTF.

El CIM define tres niveles de clases:

- Las clases que representan objetos gestionados que son aplicables a todas las áreas de gestión. Estas clases proporcionan un vocabulario básico para analizar y describir sistemas gestionados, y son una parte de lo que se conoce como el modelo núcleo.
- Las clases que representan a objetos gestionados que son aplicables a un área de gestión específica, pero son independientes de una implementación o tecnología en particular. Estas clases son una parte de lo que se conoce como el modelo común.
- Las clases que representan objetos gestionados, los cuales son adiciones de tecnología específica al modelo común. Normalmente, estas clases corresponden a plataformas específicas, como sería el entorno UNIX o Microsoft Win32, que se conoce como el modelo ampliado.

Todas las clases pueden estar relacionadas por herencia, donde una clase secundaria incluye datos y métodos de su clase primaria. Por lo general, las relaciones transferidas son invisibles para las aplicaciones de gestión que las están utilizando, y no es necesario que las aplicaciones conozcan la jerarquía de la transferencia. Las jerarquías de clase pueden obtenerse utilizando las aplicaciones que se incluyen en las herramientas WMI (si desea más información, consulte las herramientas WMI en <http://www.microsoft.com>).

Windows Management acepta también las clases de asociación. Las clases de asociación ligan dos clases diferentes para modelar una relación definida por el usuario y son visibles para todas las aplicaciones de gestión. Windows Management define las clases de asociación para su compatibilidad con las clases de sistema. Los desarrolladores de terceras empresas pueden definir las clases de asociación para sus entornos de gestión igualmente.

WBEM acepta el concepto de esquemas para agrupar clases e instancias que se utilizan dentro de un entorno de gestión en particular. La plataforma SDK incluye dos esquemas: el esquema CIM y el esquema Win32 de Microsoft. El esquema CIM incluye las definiciones de clase para los dos primeros niveles del CIM. Estas clases representan objetos gestionados que forman parte de todos los entornos gestionados, independientemente de la plataforma. El esquema Win32 incluye las definiciones de clases para objetos gestionados que forman parte de un ambiente Win32 típico.

Si desea más información sobre CIM, visite <http://www.dmtf.org>.

Sírvase leer todas las [restricciones y renunciaciones de responsabilidad](#).

[Regreso a la pantalla de contenido](#) [Principio](#)

Características principales: Guía del usuario de proveedores WMI y CDM para adaptadores de red Intel® PRO

[Características de proveedores WMI NCS](#)

[Características de proveedores CDM](#)

Características de proveedores WMI NCS

Las características principales del Proveedor WMI NCS son las siguientes:

Características del adaptador

- Enumerar todos los adaptadores físicos compatibles con Intel® PROSet.
- Enumerar la configuración de un adaptador instalado.
- Agregar, eliminar o actualizar la configuración de un adaptador instalado.
- Obtener la información de dispositivo físico de un adaptador.
- Obtener la información de un dispositivo de ranura del sistema de un adaptador.
- Obtener la configuración de protocolo IPv4 de un adaptador.
- Actualizar y modificar la configuración asociada y de agente de inicio de un adaptador.
- Desinstalar un adaptador.

Características del equipo

- Enumerar los equipos compatibles con Intel PROSet.
- Crear o eliminar un equipo de adaptadores.
- Agregar, eliminar o actualizar la configuración de un equipo.
- Agregar o eliminar adaptadores de los miembros de un equipo.
- Obtener la configuración de protocolo IPv4 de un equipo.

Características VLAN

- Enumerar las LAN virtuales de un adaptador o equipo.
- Crear o eliminar las LAN virtuales de un adaptador físico o de un equipo de adaptadores.
- Agregar, eliminar o actualizar la configuración de la VLAN.
- Obtener la configuración de protocolo IPv4 de un equipo.

Características de notificación de eventos

- Permitir que el cliente se registre.
 - Eventos del estado de adaptadores.
 - Eventos de configuración de adaptadores.
 - Eventos de sesión.
 - Eventos del estado de equipos.
 - Eventos de configuración de equipos.
 - Eventos de configuración de VLAN.

[Principio](#)

Características de proveedores CDM

Las características principales del Proveedor CDM son las siguientes:

- Ejecutar pruebas, detener las pruebas y limpiar los resultados de las pruebas independientemente del tipo de prueba de diagnóstico.

El uso de clases genéricas de configuración deberá permitir el control de pruebas de maneras no anticipadas por el software CDM mismo.

- El proveedor CDM se usa para adaptadores únicamente.
 - El uso de resultados genéricos desliga mensajes de resultados específicos del código del proveedor CDM.
 - Las entradas al registro controlan la ejecución del proveedor.
 - Los resultados de las pruebas se escriben en archivos de registro de resultados.
-

Sírvase leer todas las [restricciones y renunciaciones de responsabilidad](#).

[Regreso a la pantalla de contenido](#) [Principio](#)

Archivos instalados: Guía del usuario de proveedores WMI y CDM para adaptadores de red Intel® PRO

[Archivos WMI](#)

[Archivos de proveedor CDM](#)

Archivos WMI

Ejecutables

Los ejecutables del Proveedor WMI son los siguientes:

- **NcsWmiCo.exe**: El proveedor de núcleo. Implementa IANet_NetService y las clases de eventos de núcleo.
- **NcsWmilm.exe**: El proveedor de instancias y métodos. Implementa el esquema de adaptadores Ethernet, el esquema de generación de equipos, el esquema de configuración y el esquema de VLAN.
- **NcsWmiEv.exe**: El proveedor de eventos. Implementa los eventos de adaptadores, equipos y VLAN.

Archivos MOF

Existen archivos MOF separados para los datos de idioma neutral e idioma específico. Asimismo, hay archivos MOF separados para los espacios de nombre IntelNCS y CIMV2. Si desea más información, consulte [Configuración regional y localización e Informe de errores](#).

Los archivos MOF para los espacios de nombre IntelNCS son los siguientes:

- **NcsCmLn.mof**: Clases de base CIM de las cuales dependen las clases NCS.
- **NcsCmEnu.mfl**: Versión en inglés de EE.UU. de las clases de base CIM.
- **NcsCoLn.mof**: Clases de núcleo implementadas por el proveedor de núcleo.
- **NcsCoEnu.mfl**: Enmiendas textuales en inglés de EE.UU. a las clases de núcleo.
- **NcsIaLn.mof**: Clases para los adaptadores, equipos y VLAN IEEE 802.3.
- **NcsCoEnu.mfl**: Enmiendas textuales en inglés de EE.UU. a las clases de núcleo 802.3.

Los archivos MOF para los espacios de nombre CIMV2 son los siguientes:

- **C2CmLn.mof**: Clases de base CIM de las cuales dependen las clases NCS.
- **C2CmEnu.mfl**: Versión en inglés de EE.UU. de las clases de base CIM.
- **C2CoLn.mof**: Clases de núcleo implementadas por el proveedor de núcleo.
- **C2CoEnu.mfl**: Enmiendas textuales en inglés de EE.UU. a las clases de núcleo.
- **C2IaLn.mof**: Clases para los adaptadores, equipos y VLAN IEEE 802.3.
- **C2IaEnu.mfl**: Enmiendas textuales en inglés de EE.UU. a las clases de núcleo 802.3.

Archivos de recursos

Los archivos de recursos del Proveedor WMI son las siguientes:

- **ENU_8023.dll**: Recurso 8023 en inglés de EE.UU.
- **ENU_NWRC.dll**: Recursos WMI en inglés de EE.UU. para los proveedores de núcleo.
- **ENU_NWRC.dll**: Recursos WMI en inglés de EE.UU. para los proveedores 8023.

Los otros recursos localizados pueden descargarse a solicitud. El patrón general de los nombres de DLL de archivos localizados es "_mwr.dll" que es el código de idioma para la localización (por ejemplo, FRA para el francés estándar).

[Principio](#)

Archivos de proveedor CDM

Ejecutables

Los ejecutables del Proveedor CDM son los siguientes:

- **Ncsdiag.exe** es el ejecutable principal para los diagnósticos CDM. Cumple con la especificación de la interfaz WMI de Microsoft* y se accede como un servidor COM fuera de proceso.
- Otros ejecutables de la pila de software de Intel® PROSet

Archivos MOF

Los archivos maestros **.mof** no se envían con el producto, sino que se compilan en los componentes dependientes del idioma o idioma neutral correspondientes, según el modelo de globalización de Microsoft* Windows* Management Instrumentation. Si desea más información, consulte el capítulo sobre localización de WMI de Microsoft* WMI SDK (componente de Plataforma SDK). Preste atención especial a la sección *Compiling Localized MOF Files*(Compilación de archivos MOF localizados)..

Si se elimina un archivo **.mof** (DNcsCdmN.mof), se eliminan las definiciones de clase derivadas de Intel, pero no las clases definidas DMTF, ya que esto pondría en peligro a otras aplicaciones existentes.

El uso típico de esta implementación CDM está basada en el espacio de nombre CIMV2. Los archivos MOF para los espacios de nombre IntelNCS son los siguientes:

Nombre del archivo	Tipo de idioma	Descripción
Cdla.mof	Maestro	Definiciones de clase para la implementación CDM Intel
CdlaLn.mof	Idioma neutral	Definiciones de clase para la implementación CDM Intel
CdlaEnu.mfl	Depende del idioma inglés	Extensiones de idioma para la implementación Intel CDM
CdCm.mof	Maestro	Definiciones de clase CDM del superconjunto de núcleo
CdCmLn.mof	Idioma neutral	Definiciones de clase CDM del superconjunto de núcleo
CdCmEnu.mfl	Depende del idioma inglés	Extensiones de idioma para las definiciones CDM del superconjunto de núcleo
DNcsCdmN.mof	No corresponde	Elimina las clases Intel CDM

Los archivos MOF para los espacios de nombre CIMV2 son los siguientes:

Nombre del archivo	Tipo de idioma	Descripción
C2lcd.mof	Maestro	Definiciones de clase para la implementación CDM Intel
C2lcdLn.mof	Idioma neutral	Definiciones de clase para la implementación CDM Intel
C2lcdEnu.mfl	Depende del idioma inglés	Extensiones de idioma para la implementación Intel CDM
C2Cd.mof	Maestro	Definiciones de clase CDM del superconjunto de núcleo
C2CdLn.mof	Idioma neutral	Definiciones de clase CDM del superconjunto de núcleo

C2CdEnu.mfl	Depende del idioma inglés	Extensiones de idioma para las definiciones CDM del superconjunto de núcleo
DNcsCdm2.mof	No corresponde	Elimina las clases Intel CDM

Nota: La localización requiere que se agregue el archivo **.mof** correcto que depende del idioma.

Archivos de recursos

Los archivos de recursos del Proveedor CDM son las siguientes:

- **ENU_Diag.dll** - Recursos WMI en inglés de EE.UU. para el proveedor de diagnóstico

Sírvase leer todas las [restricciones y renunciaciones de responsabilidad](#).

[Regreso a la pantalla de contenido](#) [Principio](#)

[Regreso a la pantalla de contenido](#)

Seguridad: Guía del usuario de proveedores WMI y CDM para adaptadores de red Intel® PRO

Los proveedores WMI y CDM utilizan representaciones de los clientes para gestionar la seguridad. Cada llamada a los proveedores se llevará a cabo en el contexto de seguridad propia del cliente, que se pasa a los niveles inferiores. Una o todas las operaciones podrían fallar si no tiene derechos de administrador en el equipo objeto.

Sírvase leer todas las [restricciones y renunciaciones de responsabilidad](#).

[Regreso a la pantalla de contenido](#) [Principio](#)

Espacio de nombre y contexto: Guía del usuario de proveedores WMI y CDM para adaptadores de red Intel® PRO

Las clases CIM residen en un espacio de nombre. El espacio de nombre estándar de Microsoft* se llama **root/cimv2** y está basado en CIM v2.2 o **root/default**. Las clases de proveedores WMI y CDM puede agregarse a este espacio de nombre. Los proveedores se basan en CIM v2.5. Debido a eso y a diferencias en las claves de los objetos, las clases de proveedor están ubicadas en un espacio de nombre separado, **root/IntelNCS**.

Contexto WBEM

Los objetos de contexto proporcionan información adicional a los proveedores, misma que no puede pasarse a un método WMI API como un parámetro. Para registrar calificadores de contexto, use el **IWbemContext** para registrar los calificadores de contexto. El puntero de interfaz del objeto de contexto se pasa como el último parámetro de un método **IWbemServices**.

La tabla siguiente contiene los calificadores de contexto (valores nombrados) utilizado por el Proveedor. La mayoría de los calificadores, como SessionHandle, se utilizan solamente junto con áreas funcionales específicas de los proveedores, mientras que LocaleID, MachineName y ApplicationName pueden establecerse para todas las llamadas de **IWbemServices**.

Si no se pasa algún contexto a los proveedores, utilizarán la LocaleId que se pasó en la llamada **Initialize** a los proveedores. Cualquier lectura de un contexto que se efectuó leerá la unidad de configuración actual hasta que se ejecute una operación de escritura. Las lecturas subsecuentes mostrarán el sistema tal como estaría después de una escritura satisfactoria. Se puede utilizar un contexto NULO para todas las lecturas.

Calificador de contexto	Tipo de variante	Descripción
SessionHandle	VT_BSTR	Identifica la copia de la aplicación de las clases de red de IANet. La aplicación no puede efectuar ningún cambio a las clases o a sus atributos sin establecer primero un identificador de sesión. Consulte la sección correspondiente a la clase IANet_NetService para ver la forma en la que se establece y se utiliza un identificador de sesión. Este calificador no es necesario si la aplicación va a leer solamente datos de las clases. El identificador de sesión permite que el software NCS gestione simultáneamente varios accesos a la configuración, impidiendo así que un usuario pueda bloquear a todas las demás. Cada sesión cuenta con un caché separado para guardar cualquier cambio que se haya efectuado. Si hay varios usuarios haciendo cambios simultáneamente, entonces el primer usuario que aplique sus cambios será el que lo logre. Se invalidarán todas las otras memorias caché de los usuarios.
LocaleID	VT_BSTR	El identificador de Microsoft para una configuración regional. Es necesario si es que la aplicación necesita cadenas de texto localizadas de los proveedores. Todos los mensajes y advertencias de error estarán en inglés, a menos que se utilice el LocaleID necesario.
ApplicationName	VT_BSTR	El nombre de la aplicación que hizo la llamada. Esto es necesario para el registro.
MachineName	VT_BSTR	El nombre de la máquina que se conecta a los proveedores. Esto es necesario para el registro.
PreCheck	VT_BOOL	Este valor booleano se utiliza para indicar a los proveedores que el cliente está intentando verificar que una estación se ha permitido antes de llevar a cabo la operación. Por ejemplo, agregar un adaptador al equipo. Valores: <ul style="list-style-type: none">• TRUE = El proveedor no ejecutará la operación, pero devolverá un código de error y un estado ampliado si no se permitió la operación.• FALSE = El proveedor va a ejecutar la operación.

		Si falta este calificador, tendría el mismo efecto que si el atributo fuera FALSE.
WarningErrorCode	VT_I4	Algunas operaciones podrían requerir del envío de advertencias al usuario (por ejemplo, en algunas circunstancias, agregar un adaptador a un equipo podría hacer necesario que el equipo se vuelva a cargar). WMI no proporciona un mecanismo para este efecto. Si está presente este calificador y no es cero, el Proveedor va a devolver E_FAIL si es que la operación fue satisfactoria, pero hubo una advertencia asociada. El cliente deberá utilizar el estado extendido para recibir el texto de la advertencia.

Sírvase leer todas las [restricciones y renunciaciones de responsabilidad](#).

[Regreso a la pantalla de contenido](#) [Principio](#)

Configuración regional y localización: Guía del usuario de proveedores WMI y CDM para adaptadores de red Intel® PRO

[Archivos MOF localizados](#)

[Datos de atributos localizados](#)

Existen dos aspectos de la localización de WMI y del proveedor CDM: los archivos MOF localizados y los datos de atributos localizados.

Archivos MOF localizados

Todos los archivos MOF utilizados por los Proveedores han sido localizados de acuerdo con el modelo de globalización de Microsoft Windows* Management Instrumentation (WMI). Para lograr lo anterior, cada definición de clase deberá separarse como se muestra a continuación:

- Un versión de idioma neutral que incluye la definición de clase básica en el archivo **.mof**.
- Un versión de idioma específico que incluye la información localizada, como serían descripciones de propiedades que son específicas a la configuración regional en el archivo **.mfl** correspondiente.

Idiomas compatibles

Chino (Taiwán)
Chino (RPC)
Danés
Holandés (Países Bajos)
Inglés (EE.UU.)
Finlandés
Francés (Francia)
Alemán (Alemania)
Italiano (Italia)
Japonés
Noruego (Bokmal)
Portugués (Brasil)
Español (España - Moderno)
Sueco

Almacenamiento de clases

Las definiciones de clases en idiomas específicos se guardan en un subespacio de nombre secundario debajo del espacio de nombre que incluye la definición de clase básica en idioma neutral. Por ejemplo, para el proveedor WMI y CDM, existirá un espacio de nombre **ms_409** debajo del espacio de nombre **root/IntelIncs** para la configuración regional inglesa. De la misma manera, existe un subespacio de nombre secundario para cada idioma compatible debajo del espacio de nombre **root/IntelIncs**.

Compatibilidad localizada de MOF en el espacio de nombre cimv2

Para los espacios de nombre **root/cimv2**, las clases de los proveedores (es decir, las clases IANet_) se han derivado de clases de base agregadas por WMI a ese espacio de nombre. Ya existe un subespacio de nombre con definiciones de clase de idioma específico para las clases de base debajo del espacio de nombre **root/cimv2**. Las definiciones de clase de idioma específico de IA_Net se agregan a ese espacio de nombre secundario existente. Debido a esta dependencia en clases de base, la localización de MOF se hace en la configuración regional predeterminada.

Compatibilidad de tiempo de ejecución

Para la recuperación de datos localizados, una aplicación WMI puede especificar la configuración regional utilizando un parámetro strLocale en las llamadas a **SWbemLocator::ConnectServer** y a **IWbemLocator::ConnectServer**. Si no se especificó la configuración regional, se utilizará la configuración regional predeterminada para ese sistema. (por ejemplo, MS_409 para inglés de EE.UU). Esta configuración regional se usa para seleccionar el espacio de nombre correcto cuando se agregan las cadenas en inglés.

Además, **IWbemServices::GetObject**, **SwbemServices.GetObject**, **IWbemServices::ExecQuery** y **SWbemServices.ExecQuery** deben especificar el indicador WBEM_FLAG_USE_AMENDED_QUALIFIERS para solicitar los datos localizados junto con la definición básica. Esto es necesario en todas las funciones que producen valores que se pueden visualizar utilizando los mapas de valores, descripciones de visualización o algunos otros calificadores corregidos de los archivos MOF.

[Principio](#)

Datos de atributo localizados

Para obtener datos de atributo localizados (como serían los mensajes de error), los Proveedores deben de saber la configuración regional del llamante de cada llamada. Para que esto funcione correctamente, el cliente deberá agregar la configuración genérica al objeto de contexto que se pasa con cada llamada (consulte [Espacio de nombre y contexto](#) en el contexto WBEM). Si los Proveedores deben devolver una cadena localizable, intentarían cargar un DLL de recurso que sea el apropiado para la configuración regional del cliente. Si no existe un DLL de recurso apropiado, entonces los proveedores van a devolver las cadenas en inglés de EE.UU.

Sírvase leer todas las [restricciones y renunciaciones de responsabilidad](#).

[Regreso a la pantalla de contenido](#) [Principio](#)

Informe de errores: Guía del usuario de proveedores WMI y CDM para adaptadores de red Intel® PRO

[Visión general](#)
[Códigos de error](#)

Visión general

Esta sección sobre IANet_ExtendedStatus describe el modo de manejo de los errores generados por los proveedores WMI y CDM. La manera y el momento en que se devuelve un objeto error dependen de si la llamada fue sincrónica, semisincrónica o asincrónica. En la mayoría de los casos, el HRESULT se establece en WBEM_E_FAILED cuando ocurre un error. Sin embargo, en este momento, se desconoce si el error fue generado por WMI o por el proveedor.

Para obtener el objeto de error para llamadas sincrónicas, use GetErrorInfo() para obtener el objeto IErrorInfo. Use QueryInterface() para obtener el objeto IWbemClassObject que incluye la información de error.

Para obtener el objeto de error para llamadas asincrónicas, el IWbemClassObject se retorna como el último elemento de la última llamada SetStatus(). Después de obtener la instancia del objeto de error, puede comprobar la propiedad __Class para determinar el origen del error. WMI crea una instancia de __ExtendedStatus, y el Proveedor crea una instancia de IANet_ExtendedStatus para los errores relativos a clases IANet_. IANet_ExtendedStatus se deriva de __ExtendedStatus e incluye los calificadores de objeto de error siguientes:

- Description: Descripción del error personalizada a la configuración regional actual.
- File: Archivo de código donde se generó el error.
- Line: Número de línea del archivo de código con el error.
- ParameterInfo: Clase o atributo que estaba en uso cuando ocurrió el error.
- Operation: La operación intentada cuando ocurrió el error.
- ProviderName: Nombre del proveedor que ocasionó el error.
- StatusCode: Código devuelto por la llamada interna que falló.
- SessionHandle: Identificador de sesión utilizado para la operación.
- RuleFailureReasons: Razón por la cual falló la operación. Una operación puede fallar debido a que falló una regla técnica (por ejemplo, se debe tener un adaptador administrativo en ciertos equipos).

[Principio](#)

Códigos de error

Para todos los errores, los proveedores pueden dar una descripción personalizada de la configuración regional. Los códigos de error están en forma de HRESULT con la severidad establecida en uno (1) y la facilidad en ITF. Una aplicación puede usar los códigos siguientes como una base para una acción de recuperación:

- 0x80040901: "WMI: Error al colocar propiedades"
- 0x80040902: "WMI: Sin objeto de clase"
- 0x80040903: "WMI: Error en la creación de clase"
- 0x80040904: "WMI: No pudo generar instancias de clase"
- 0x80040905: "WMI: Error en la creación de una matriz segura"
- 0x80040906: "WMI: Error en la colocación de una matriz segura"
- 0x80040907: "WMI: No se pudo devolver el objeto a WMI"
- 0x80040908: "WMI: Error al obtener propiedades"
- 0x80040909: "WMI: Tipo inesperado mientras obtenía propiedades"
- 0x8004090A: "WMI: La clase no ha sido implementada por este proveedor"
- 0x8004090B: "WMI: No es posible analizar la frase WQL"
- 0x8004090C: "WMI: Los proveedores sólo admiten WQL"

- 0x8004090D: "WMI: El parámetro en contexto tiene el tipo equivocado"
- 0x8004090E: "WMI: Error al formatear el registro de depuración"
- 0x8004090F: "WMI: ruta de objeto incorrecta"
- 0x80040910: "WMI: No se puede actualizar la configuración"
- 0x80040911: "WMI: Parámetro nulo transferido al método"
- 0x80040912: "Valor demasiado pequeño"
- 0x80040913: "Valor demasiado grande"
- 0x80040914: "Configuración no sincronizada"
- 0x80040915: "Valor de cadena demasiado largo"
- 0x80040916: "El valor no es uno de los valores permitidos"
- 0x80040917: "WMI: No se encontró el calificador"
- 0x80040918: "WMI: No se encontró el conjunto de calificadores"
- 0x80040919: "WMI: Error al acceder la matriz segura"
- 0x8004091A: "WMI: Excepción no controlada"
- 0x8004091B: "WMI: La operación no es compatible con esta clase"
- 0x8004091C: "WMI: Clase de evento inesperado"
- 0x8004091D: "WMI: Datos de evento erróneos"
- 0x8004091E: "WMI: Se ha logrado la operación con advertencias"
- 0x8004081F: "WMI: Se ha cancelado el servicio NCS"

- 0x80040801: "EAL: Excepción interna"
- 0x80040802: "EAL: Fallo general"
- 0x80040803: "EAL: No ha sido inicializado"
- 0x80040804: "EAL: Falló la inicialización"
- 0x80040805: "EAL: Se ha excedido el límite de sesiones"
- 0x80040806: "EAL: No hay memoria suficiente"
- 0x80040807: "EAL: Error de regla de sintaxis"
- 0x80040808: "EAL: Final de la lista inesperado"
- 0x80040809: "EAL: Error de regla de enlace"
- 0x8004080A: "EAL: Fallo en la creación del dispositivo"
- 0x8004080B: "EAL: No se encontraron los servicios de medios"
- 0x8004080C: "EAL: No se encontró el servicio de dispositivos"
- 0x8004080D: "EAL: No se encontró el módulo de bus PCI"
- 0x8004080E: "EAL: El adaptador pertenece a un equipo"
- 0x8004080F: "EAL: Error de regla de creación del punto de acceso"
- 0x80040810: "EAL: Error de clave de registro"
- 0x80040811: "EAL: Error de ruta de archivo XML de registro"
- 0x80040812: "EAL: Clase de evento inesperada"
- 0x80040813: "EAL: Identificación de módulo desconocido"
- 0x80040814: "EAL: No se encontró el servicio de reglas"
- 0x80040815: "EAL: Puntero de entrada NULO"
- 0x80040816: "EAL: Error de regla gramatical"
- 0x80040817: "EAL: Fallo en la regla"
- 0x80040818: "EAL: La configuración ya ha sido agrupada"

- 0x80040220: "Nivel de sincronización: La eliminación del equipo ha fallado"
- 0x80040221: "Nivel de sincronización: La creación de VLAN ha fallado"
- 0x80040222: "Nivel de sincronización: La eliminación de VLAN ha fallado"
- 0x80040223: "Nivel de sincronización: La eliminación del adaptador ha fallado"
- 0x80040224: "Nivel de sincronización: La configuración de Cambio/Creación/Eliminación ha fallado"
- 0x80040225: "Nivel de sincronización: El parámetro Cambio/Creación/Eliminación ha fallado"
- 0x80040226: "Nivel de sincronización: Subsistema de NetConfig bloqueado" "
- 0x80040227: "Nivel de sincronización: Actualización de sistema en curso Intente de nuevo más tarde."
- 0x80040228: "Nivel de sincronización: El adaptador está bloqueado"
- 0x80040229: "Nivel de sincronización: La lectura de flash ha fallado"
- 0x8004022A: "Nivel de sincronización:

- 0x80040210: "Nivel de sincronización: Desplazamiento de evento no válido"
- 0x80040211: "Nivel de sincronización: Entrada no válida"
- 0x80040212: "Nivel de sincronización: Clave no válida"

- 0x80040213: "Nivel de sincronización: El adaptador no pertenece al equipo"
- 0x80040214: "Nivel de sincronización: El controlador no está cargado"
- 0x80040215: "Nivel de sincronización: La representación de cliente ha fallado"
- 0x80040216: "Nivel de sincronización: Excepción capturada"
- 0x80040217: "Nivel de sincronización: La sesión no ha sido bloqueada"
- 0x80040218: "Nivel de sincronización: El nivel de acceso de hardware no está disponible"
- 0x80040219: "Nivel de sincronización: El flash no está disponible"
- 0x8004021A: "Nivel de sincronización: Diagnósticos no compatibles"
- 0x8004021B: "Nivel de sincronización: La prueba de diagnóstico no está en ejecución"
- 0x8004021C: "Nivel de sincronización: La actualización del agente de inicio no está disponible"
- 0x8004021D: "Nivel de sincronización: El agente de inicio está dañado"
- 0x8004021E: "Nivel de sincronización: La escritura de flash ha fallado"
- 0x8004021F— "Nivel de sincronización: La eliminación del equipo ha fallado"
- 0x80040201: "Nivel de sincronización: La inicialización ha fallado"
- 0x80040202: "Nivel de sincronización: Controlador de inicialización no válido"
- 0x80040203: "Nivel de sincronización: El controlador de sesión ya existe"
- 0x80040204: "Nivel de sincronización: Controlador de sesión no válido"
- 0x80040205: "Nivel de sincronización: Se ha alcanzado la cantidad máxima de sesiones."
- 0x80040206: "Nivel de sincronización: El controlador de bloque de sesión ya existe"
- 0x80040207: "Nivel de sincronización: Controlador de sesión no válido"
- 0x80040208: "Nivel de sincronización: La sesión ya ha sido bloqueada"
- 0x80040209: "Nivel de sincronización: Identificación de módulo de servicio medio no válida"
- 0x8004020A: "Nivel de sincronización: Identificación de módulo de servicio avanzado no válida"
- 0x8004020B: "Nivel de sincronización: Identificación del dispositivo de servicio medio no válida"
- 0x8004020C: "Nivel de sincronización: Identificación de tipo de componente no válida"
- 0x8004020D: "Nivel de sincronización: Identificación del módulo de interfaz de bus no válida"
- 0x8004020E: "Nivel de sincronización: Controlador de ventana de recepción no válido"
- 0x8004020F: "Nivel de sincronización: Información de evento no válido"

- 0x80040401: "HAM PCI: Dirección de asignación de memoria no válida"
- 0x80040402: "HAM PCI: El controlador de configuración no se ha cargado"
- 0x80040403: "HAM PCI: La versión del controlador de configuración no coincide"
- 0x80040404: "HAM PCI: No se encontró la ranura del dispositivo"
- 0x80040405: "HAM PCI: El controlador de diagnóstico no se ha cargado"
- 0x80040406: "HAM PCI: La versión del controlador de diagnóstico no coincide"
- 0x80040407: "HAM PCI: La inicialización del controlador de diagnóstico ha fallado"
- 0x80040408: "HAM PCI: No se han inicializado los diagnósticos"
- 0x80040409: "HAM PCI: Ya se han inicializado los diagnósticos"
- 0x8004040A: "HAM PCI: La prueba de diagnóstico ya está en ejecución"
- 0x8004040B: "HAM PCI: La prueba de diagnóstico no está en ejecución"
- 0x8004040C: "HAM PCI: La prueba de diagnóstico ha sido cancelada"
- 0x8004040D: "HAM PCI: Diagnóstico, número de prueba no válido"
- 0x8004040E: "HAM PCI: Falta el hardware de diagnóstico"
- 0x8004040F: "HAM PCI: La inicialización del envío y recibo de diagnóstico ha fallado"

- 0x80040511: "Servicios de medios: Fallo de llamada E/S NDIS"
- 0x80040512: "Servicios de medios: No se ha cargado el minipuerto"
- 0x8004051B: "Servicios de medios: Controlador de dispositivo no válido"
- 0x8004051C: "Servicios de medios: Controlador de adaptador no válido"
- 0x8004051D: "Servicios de medios: Controlador de equipo no válido"
- 0x8004051E: "Servicios de medios: Controlador de VLAN no válido"
- 0x8004051F: "Servicios de medios: Falta el dispositivo"
- 0x80040520: "Servicios de medios: Tipo de configuración no válido"
- 0x80040521: "Servicios de medios: Objeto desconocido no válido"
- 0x80040522: "Servicios de medios: Controlador de configuración no válido"
- 0x80040523: "Servicios de medios: Modo de equipo no válido"
- 0x80040525: "Servicios de medios: La configuración ya existe"

- 0x80042001: "RAP: Ya se ha inicializado"
- 0x80042002: "RAP: Archivo XML no válido"

- 0x80042003: "RAP: Error de carga XML"
 - 0x80042004: "RAP: No ha sido inicializado"
 - 0x80042005: "RAP: La regla no fue extraída antes"
 - 0x80042006: "RAP: El conteo de condiciones no coincide"
 - 0x80042007: "RAP: Error de aplicación de resultados"
 - 0x80042008: "RAP: Regla no válida"
 - 0x80042009: "RAP: No se ha encontrado el nodo"
 - 0x8004200A: "RAP: Error, no es nodo único"
 - 0x8004200B: "RAP: Ninguna regla de acción"
 - 0x8004200C: "RAP: Cero condiciones"
 - 0x8004200D: "RAP: Cero acciones"
 - 0x8004200E: "RAP: Error de decodificación XML"
-

Sírvase leer todas las [restricciones y renunciaciones de responsabilidad](#).

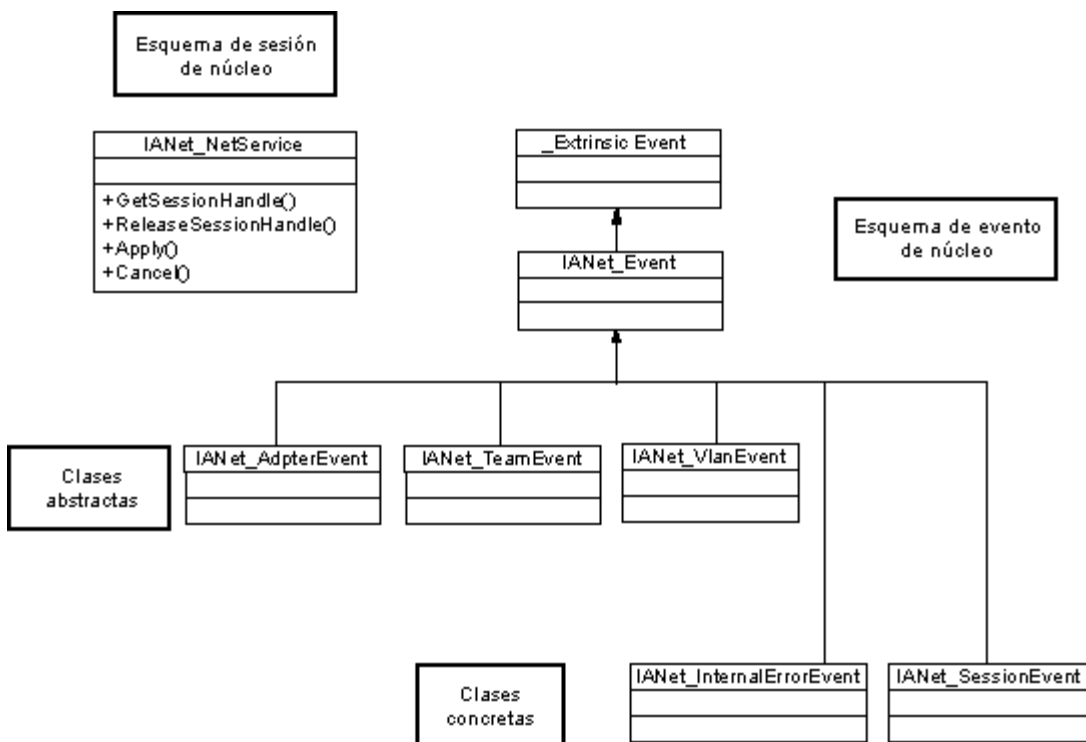
[Regreso a la pantalla de contenido](#) [Principio](#)

El esquema de núcleo: Guía del usuario de proveedores WMI y CDM para adaptadores de red Intel® PRO

- [Visión general](#)
- [IANet_NetService](#)
- [Eventos de núcleo](#)
- [Casos de uso](#)

Visión general

El Esquema de núcleo consiste de la clase IANet_NetService y de las clases del evento de núcleo.



[Principio](#)

IANet_NetService

Propósito

La clase IANet_NetService es el objeto raíz del esquema IANet_. Esta clase le permite al cliente acceder a las sesiones necesarias para realizar conjuntos.

Instancias

Hay una sola instancia de este objeto. El cliente no debería depender de la clave que se usa para esta clase. En su lugar, el cliente debe obtener la instancia de la clase enumerando todas las instancias de IANet_NetService.

Creación de instancias

No se pueden crear instancias de IANet_NetService.

Eliminación de instancias

No se puede eliminar la instancia de `IANet_NetService`.

Modificación de propiedades

No hay propiedades que el usuario pueda modificar en esta clase.

Atributos compatibles

Esta clase implementa dos atributos:

- `Version` (Versión): Incluye la versión actual del proveedor de núcleo.
- `InstallDate` (Fecha de instalación): Incluye la fecha en la que se instalaron los proveedores.

Métodos

Se pueden utilizar los métodos siguientes para gestionar las sesiones:

- **`void GetSessionHandle([OUT] string SessionHandle, [out] uint32 ActiveSessions)`**: Se usa para fijar una cadena de controlador de la sesión, misma que debería ubicarse en el objeto de contexto en el calificador de `SessionHandle`. `ActiveSessions` devuelve el número de sesiones activas del sistema. Eso permite que los clientes notifiquen que otros están modificando los valores de la red.
- **`void Apply([IN] string sSessionHandle, [OUT] uint32 FollowupAction)`**: Aplica los cambios efectuados con un controlador de sesión en particular. El argumento `uint32` devuelto es utilizado por los proveedores WMI y CDM para informar a la aplicación que el servidor debería reiniciarse antes de que los cambios entren en vigor. Esto puede lograrse llamando al método **`Reboot`** (Reiniciar) en la clase `Win32_OperatingSystem`.

Valores:

1 = Reiniciación del sistema requerida
0 = No es necesario reiniciar

- **`void ReleaseSessionHandle ([IN] string SessionHandle)`**: Libera un controlador de sesión después de haberlo utilizado. Cualquier cambio efectuado con esta sesión se pierde. El controlador de sesión no tendrá validez después de esta llamada y no puede utilizarse más.
- **`void Cancel([IN] string SessionHandle)`**: Cancela una sesión. El caché interno se limpia y cualquier dato que se lea después de esta llamada mostrará la configuración actual.

[Principio](#)

Eventos de núcleo

IANet_SessionEvent

Propósito

Este evento se usa para notificar al cliente acerca del uso de la sesión API de NCS. Los clientes pueden utilizar este evento para recibir información si es que otros clientes están creando o utilizando sesiones.

Desencadenadores

Este evento se desencadena cuando un cliente crea una sesión, elimina una sesión o llama a **`Apply`** (Aplicar) para una sesión.

Datos del evento

El **`EventType`** (tipo de evento) puede tener uno de los valores siguientes:

- "New session" (sesión nueva) indica que el cliente, o algún otro cliente, ha creado una sesión nueva.
- "End session" (Fin de sesión) indica que un cliente ha concluido una sesión. La sesión puede haber sido terminada por el cliente o por otro cliente.
- "Cache invalidated" (Caché no es válido) indica que otro cliente ha llamado a **`Apply`** en una sesión.. Todas las otras sesiones son declaradas no válidas y el caché asociado con sus sesiones se ha eliminado.
- "Configuration changed" (Configuración ha cambiado) indica que se ha cambiado la configuración de la sesión.

La **`SessionHandle`** (Controlador de la sesión) incluye el controlador de la sesión que desencadenó el evento.

`OpenSessions` (Sesiones abiertas) incluye el número de sesiones abiertas. Este artículo de datos es NULO para los eventos "Cache invalidated" y "Configuration changed".

IANet_InternalErrorEvent

Propósito

Este evento se usa para notificar al cliente que ha ocurrido un error interno en los proveedores de evento. En algunos casos, significa que el proveedor del evento no tiene más la posibilidad de informar acerca de eventos futuros.

Desencadenadores

Este evento ocurrirá:

- Después de que el proveedor de eventos recibe un evento desconocido de una fuente de eventos
- Después de que el software que proporcionó el evento ha sido desactivado
- Después de que el proveedor del evento recibe un evento pero la fuente del evento no puede obtener más datos acerca del evento

Datos del evento

El **EventType** (tipo de evento) puede ser uno de los siguientes:

- "Could not get event data" (No se han podido obtener los datos del evento) Ocurrió un evento, pero la fuente del evento no puede obtener más datos acerca del evento.
- "Event source has shut down" (La fuente de eventos se ha desactivado). La fuente de datos del evento se ha desactivado. En este caso, el proveedor de eventos será desactivado también y no se van a generar más eventos hasta que se haya reiniciado la fuente y se hayan efectuado nuevas consultas de notificación.
- "Unexpected message" (Mensaje inesperado). El proveedor de eventos ha recibido un evento inesperado.

[Principio](#)

Casos de uso

Se requiere un controlador de sesión para cambiar la configuración. El controlador de sesión permite que el software NCS gestione simultáneamente varios accesos a la configuración, impidiendo así que una sesión pueda bloquear a todas las demás. Cada sesión cuenta con un caché separado para guardar cualquier cambio que se haya efectuado. Si hay varias sesiones haciendo cambios simultáneamente, entonces el primero en aplicar sus cambios será el que lo logre. Todos los otros cachés de la sesión no serán válidos.

Obtener un controlador de sesión

El cliente debe obtener la ruta del objeto de la única instancia de `IANet_NetService` antes de poder tener acceso al controlador de sesión. Llame a `IWbemServices::CreateInstanceEnum` y pase el nombre de la clase: `IANet_NetService`. Esto es equivalente a llamar a `IWbemServices::ExecQuery` con la consulta `SELECT * FROM IANet_NetService`. Antes de hacer ningún cambio a la configuración, el cliente debe obtener un controlador de sesión. Use el método `GetSessionHandle` para iniciar una sesión nueva.

El cliente puede usar `IWbemServices::ExecMethod` para ejecutar un método en un objeto CIM y va a necesitar la ruta del objeto, del atributo `__PATH` de la instancia de `IANet_NetService`. Este método devuelve también el número de sesiones activas actualmente. Tal vez el cliente desee emitir una notificación si no tiene acceso exclusivo a Network Configuration Service (NCS).

Usar un controlador de sesión en el objeto `IWbemContext`

Después de obtener un controlador de sesión, el cliente debe crear un objeto `IWbemContext`. Guarde el controlador de sesión en el calificador `SessionHandle` de este objeto. Se debe pasar un puntero a este objeto COM a cada llamada dentro de `IWbemService`. No se necesita el controlador de sesión al hacer llamadas para tener acceso al objeto `IANet_NetService`, ya que toma el controlador como un argumento explícito.

Leer los cambios pendientes usando un controlador de sesión

Durante la lectura de la configuración, si pasa el controlador de sesión en el contexto, entonces los proveedores devolverán la configuración como si las actualizaciones pendientes fueron aplicadas (por ejemplo, van a faltar adaptadores no instalados y los

valores modificados devolverán sus valores nuevos). Sin embargo, algunos objetos no van a aparecer hasta que se haya llamado a **Apply** (por ejemplo, los `IANet_IPProtocolEndpoints` se crearán hasta que el protocolo se haya ligado al mini puerto correspondiente).

Finalizar con un controlador de sesión

Después de cambiar la configuración, llame al método **Apply** para activar los cambios. Eso puede devolver un código de acción de seguimiento (por ejemplo, reiniciar el sistema antes de que los cambios entren en vigor).

Siempre llame a **ReleaseSessionHandle** después de finalizar una sesión, ya que de otro modo, se podrían descartar todos los cambios. Si llama al método **Cancel**, se descartará cualquier cambio efectuado, pero el cliente puede seguir utilizando el controlador de sesión tal como si se hubiera creado en ese momento.

Registro de los Eventos de núcleo

Las aplicaciones deberían usar `IWbemServices::ExecNotificationQuery` o `IWbemServices::ExecNotificationQueryAsync` para solicitar notificación de los eventos. Las consultas siguientes son ejemplos de consultas de notificación de eventos (esta lista no está completa, ya que son posibles muchas consultas):

- **SELECT * FROM IANet_Event**: Solicitar todos los eventos.
- **SELECT * FROM IANet_SessionEvent**: Solicitar todos los eventos de sesión.
- **SELECT * FROM IANet_InternalErrorEvent**: Solicitar todos los eventos internos.

Sírvase leer todas las [restricciones y renunciaciones de responsabilidad](#).

[Regreso a la pantalla de contenido](#) [Principio](#)

El esquema de adaptadores Ethernet: Guía del usuario de proveedores WMI y CDM para adaptadores de red Intel® PRO

[Visión general](#)

[IANet_EthernetAdapter](#)

[IANet_IPProtocolEndpoint](#)

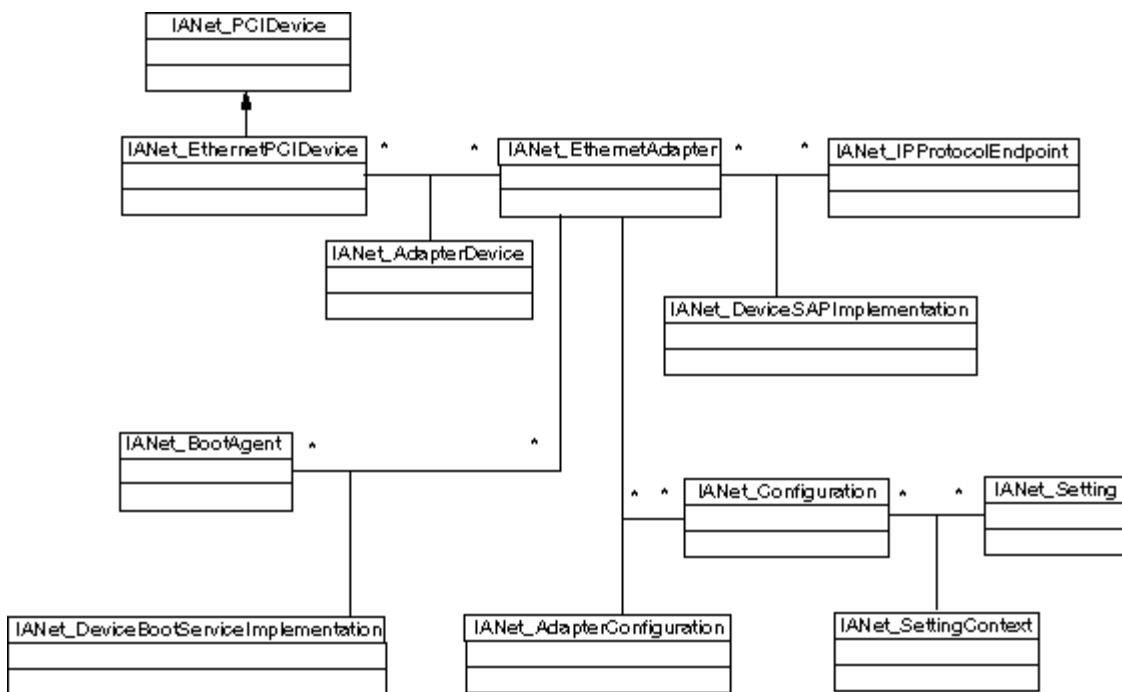
[IANet_BootAgent](#)

[IANet_PCIDevice](#)

[IANet_EthernetPCIDevice](#)

Visión general

El esquema de adaptadores se usa para modelar los distintos adaptadores Intel® PROSet Ethernet configurables. Este esquema se basa en el esquema CIM v2.5.



[Principio](#)

IANet_EthernetAdapter

Propósito

IANet_EthernetAdapter define la capacidad y el estado de todos los adaptadores de red Intel PRO instalados, así como cualquier otro adaptador que pudiese pertenecer a un equipo utilizando los controladores intermedios Intel. La clase se deriva de la superclase de CIM_EthernetAdapter que se ha definido en CIMv2.5. CIM_EthernetAdapter se deriva de CIM_NetworkAdapter, una clase abstracta que define conceptos generales de hardware de red, como serían PermanentAddress, CurrentAddress, Speed of operation, etc.

Instancias

Habrán instancias de esta clase para cada uno de los siguientes:

- NIC Intel compatibles e instaladas
- NIC que no son de Intel que pueden participar en un equipo de varios fabricantes de Intel
- Equipo de adaptadores Intel creado

Creación de instancias

No se pueden crear instancias de IANet_EthernetAdapter.

Eliminación de instancias

La eliminación de una instancia de IANet_EthernetAdapter desinstalará los adaptadores físicos. Solamente los adaptadores que no son de Intel pueden desinstalarse de esta manera. Esta operación requiere de un controlador de sesión.

Modificación de propiedades

No hay propiedades que el usuario pueda modificar en esta clase.

Atributos no compatibles

Los atributos siguientes no son necesarios para Intel PROSet y, en consecuencia, no son compatibles:

- AutoSense (esto se expone como un valor)
- ErrorCleared
- OtherIdentifyingInfo
- IdentifyingDescriptions
- InstallDate
- LastErrorCode
- MaxDataSize
- MaxQuiesceTime
- PowerManagementCapabilities (esto se expone como un método)
- PowerManagementSupported (esto se expone como un método)
- PowerOnHours
- ShortFramesReceived
- SymbolErrors
- TotalPowerOnHours

Métodos

Esta instancia de clase acepta los siguientes métodos:

- **IdentifyAdapter**: Identifica el adaptador haciendo parpadear la luz del adaptador por unos cuantos segundos. Este método funcionará solamente con los adaptadores físicos.
- **HasVLANs**: Especifica el número de LAN virtuales en este adaptador.
- **IsPowerMgmtSupported**: Indica si el adaptador es compatible con la gestión de energía.
- **GetPowerUsage**: Detecta el consumo de energía total del adaptador.
0 = Energía normal
1 = Energía reducida
- **SetPowerUsage**: Reduce el consumo de energía total del adaptador. El valor del consumo de energía no se conserva entre los reinicios del sistema o las recargas de los controladores. Cuando el sistema reinicia o se recarga el controlador, el adaptador regresa al consumo normal de energía automáticamente.
- **GetPowerUsageOptions**: Detecta las configuraciones optativas del consumo de energía (por ejemplo, el consumo de energía para el modo de espera, la operación con batería, etc.)
- **SetPowerUsageOptions**: Cambia las configuraciones del consumo de energía (por ejemplo, el método puede utilizarse para reducir el consumo de energía para el modo de espera, la operación con batería, etc.)
Nota: Los valores de consumo de energía se guardan y se utilizan para los reinicios subsecuentes.
- **TestCable**: Ejecuta pruebas de diagnóstico en un adaptador en particular. Si hay una falla, este método devuelve los problemas, causas y soluciones posibles.
- **AdvancedTestCable**: Ejecuta pruebas de cable avanzadas en un adaptador en particular. Este conjunto de pruebas está disponible con los adaptadores de 1000 Mbps. El método devuelve los nombres de las pruebas y sus resultados respectivos.
Nota: Puede ocurrir una falla de enlace cuando **SpeedDuplex** no se establece en **Auto Negotiate**. En esta instancia, **SpeedAndDuplexNotAutomatic** el parámetro de salida tendrá el valor TRUE.
- **TestLinkSpeed**: Determina si el adaptador está funcionando a la velocidad completa. Si el adaptador anuncia menos de 1 Gigabit, entonces el método especifica las razones posibles (por ejemplo , "El interlocutor de enlace no puede funcionar a 1000 Mbps").

IANet_IPProtocolEndpoint

Propósito

Esta clase se usa para describir la configuración IP para un extremo del protocolo en el sistema. El Proveedor WMI no suministra información sobre ningún otro tipo de protocolo de red. La clase se deriva de la clase abstracta CIM_IPProtocolEndpoint. El proveedor WMI proporciona información de protocolo solamente cuando se refiere a una entidad gestionada por Intel PROSet.

Instancias

Habrá una instancia de IANet_IPProtocolEndpoint para cada una de las ligas de la pila del protocolo IP a un extremo compatible con Intel (es decir, adaptadores Intel, adaptadores Intel preparados para equipos y VLAN). Algunos adaptadores que pertenecen a un equipo no cuentan con su dirección IP propia, por lo que no tienen un IANet_IPProtocolEndpoint asociado directamente con su instancia de adaptador. El IANet_IPProtocolEndpoint existe solamente después de que el sistema operativo ha ligado el protocolo al adaptador o VLAN. A pesar de que algunos adaptadores tendrán más de una dirección IP, estarán asociados solamente con una instancia del extremo del protocolo IP. El proveedor no acepta este uso avanzado ya que no sólo no es necesario sino que Intel PROSet no lo utiliza.

Creación de instancias

No se pueden crear instancias de IANet_IPProtocolEndpoint. La instancia existirá solamente si el sistema operativo ha ligado el protocolo al extremo.

Eliminación de instancias

No se pueden eliminar instancias de IANet_IPProtocolEndpoint.

Modificación de propiedades

No hay propiedades que el usuario pueda modificar en esta clase.

Asociaciones

Se usa una instancia IANet_AdapterProtocolImplementation para asociar un IANet_EthernetAdapter con un IANet_IPProtocolEndpoint. Una instancia de IANet_VLANProtocolDependency se usa para asociar una VLAN con un IANet_IPProtocolEndpoint.

Nota: Los equipos se asocian al extremo mediante el adaptador que representa el adaptador virtual del equipo.

Atributos compatibles

Los atributos de sólo lectura siguientes son necesarios para Intel PROSet:

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

Atributos no compatibles

Los atributos siguientes no son necesarios para Intel PROSet y, en consecuencia, no son compatibles:

- Caption
- Description
- InstallDate
- NameFormat
- OtherTypeInfo
- ProtocolType
- Status

Métodos

Ninguno

[Principio](#)

IANet_BootAgent

Propósito

Esta clase se usa para capturar la información sobre la capacidad de inicio de red de un adaptador (por ejemplo, la configuración del agente de inicio PXE compatible con algunos adaptadores Intel). Esta clase se deriva de CIM_BootService.

Instancias

Existe una instancia de IANet_BootAgent para cada adaptador que es compatible con la capacidad de agente de inicio, aunque no se haya instalado actualmente el agente de inicio.

Creación de instancias

No se pueden crear instancias de IANet_BootAgent. Existe una instancia solamente si el adaptador acepta la funcionalidad de agente de inicio.

Eliminación de instancias

No se pueden eliminar instancias de IANet_BootAgent.

Modificación de propiedades

No hay propiedades que el usuario pueda modificar en esta clase.

Asociaciones

Se usa una instancia de IANet_DeviceBootServiceImplementation para asociar un IANet_EthernetAdapter con un IANet_BootAgent, si lo acepta el adaptador.

Atributos compatibles

Los atributos de sólo lectura siguientes son necesarios para Intel PROSet:

- InvalidImageSignature
- Version
- UpdateAvailable
- FlashImageType

Atributos no compatibles

Los atributos siguientes no son necesarios para Intel PROSet y, en consecuencia, no son compatibles:

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

Métodos

Los métodos siguientes de esta clase se pueden usar para actualizar la Flash ROM de la NIC:

<pre>uint32 ProgramFlash([IN, ValueMap {"0","1"} , Values {"Check Version", "Write Flash"}: Amended] uint32 Action, [IN] uint8 NewFlashData[], [OUT] string strErrorMessage);</pre>	<p>Este método se usa para actualizar la Flash ROM de la NIC. La NIC dejará de comunicarse con la red mientras se actualiza la memoria Flash.</p>
--	---

```
uint32 ReadFlash( [OUT] uint8  
FlashData[] );
```

Este método lee la Flash ROM en la NIC.

[Principio](#)

IANet_PCIDevice

Propósito

Esta clase se usa para describir las propiedades de un dispositivo PCI en un dispositivo de red del sistema. Esta clase se deriva de CIM_PCIDevice.

Instancias

Habrà una instancia de esta clase para cada tarjeta PCI que es un dispositivo de red en el sistema. Para IA64, solamente tendrán instancias los dispositivos PCI que son adaptadores compatibles de Intel PROSet.

Creación de instancias

No se pueden crear instancias de IANet_PCIDevice.

Eliminación de instancias

No se pueden eliminar instancias de IANet_PCIDevice.

Modificación de propiedades

No hay propiedades que el usuario pueda modificar en esta clase.

Asociaciones

Consulte IANet_EthernetPCIDevice para las asociaciones de clase.

Métodos

No hay métodos compatibles con esta clase.

Atributos no compatibles

Los atributos siguientes no son compatibles con el proveedor WMI:

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

[Principio](#)

IANet_EthernetPCIDevice

Propósito

Esta clase se usa para describir las propiedades de un dispositivo PCI para un adaptador Ethernet compatible con Intel PROSet. Esta es una subclase de IANet_PCIDevice. La clase incluye algunos atributos adicionales que son conocidos solamente por los dispositivos PCI compatibles con Intel PROSet.

Instancias

Habrà una instancia de esta clase para cada tarjeta PCI que es un adaptador Ethernet compatible con Intel PROSet.

Creación de instancias

No se pueden crear instancias de IANet_EthernetPCIDevice.

Eliminación de instancias

No se pueden eliminar instancias de IANet_EthernetPCIDevice.

Modificación de propiedades

No hay propiedades que el usuario pueda modificar en esta clase.

Asociaciones

Se usa una instancia IANet_AdapterDevice para asociar un IANet_PCIDevice con un IANet_EthernetAdapter. Los adaptadores virtuales (es decir, adaptadores creados para representar equipos) no tienen un IANet_PCIDevice asociado.

Atributos no compatibles

Los atributos siguientes no son compatibles con el proveedor WMI:

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

Métodos

No hay métodos compatibles con esta clase.

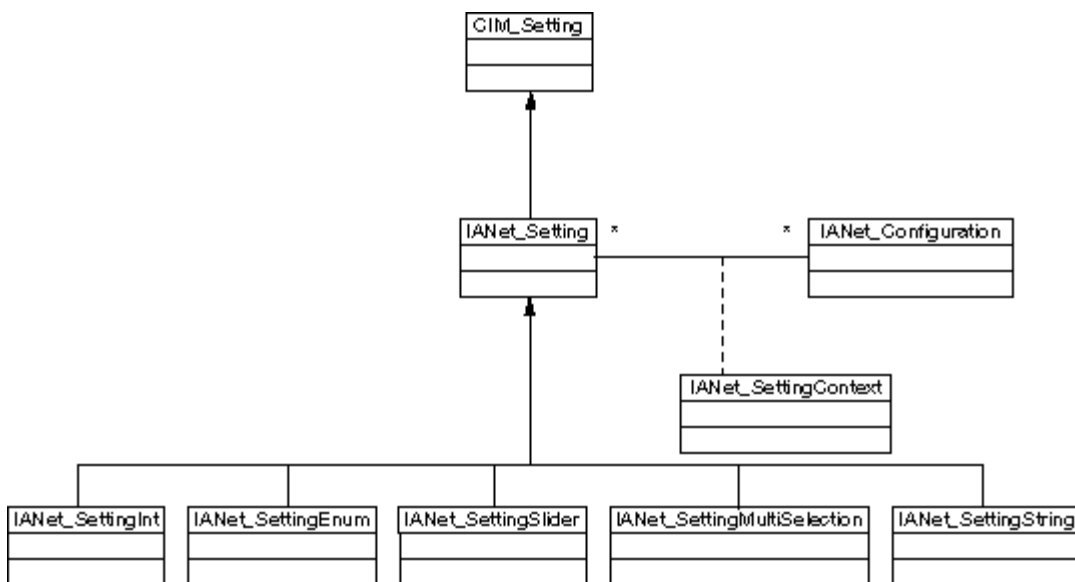
Sírvase leer todas las [restricciones y renunciaciones de responsabilidad](#).

[Regreso a la pantalla de contenido](#) [Principio](#)

Esquema de configuración: Guía del usuario de proveedores WMI y CDM para adaptadores de red Intel® PRO

- [Visión general](#)
- [IANet_Configuration](#)
- [IANet_Setting](#)
- [IANet_SettingInt](#)
- [IANet_SettingEnum](#)
- [IANet_SettingSlider](#)
- [IANet_SettingMultiSelection](#)
- [IANet_SettingString](#)

Visión general



Principio

IANet_Configuration

Propósito

Esta clase se utiliza para agrupar una colección de instancias de IANet_Setting. Esta clase se deriva de CIM_Configuration.

Instancias

Cada adaptador, VLAN o equipo puede tener varias instancias asociadas de IANet_Configuration (cada configuración corresponde a un escenario de utilización diferente del adaptador).

Para esta versión de los proveedores WMI y CDM, habrá una sola instancia de IANet_Configuration por adaptador, VLAN o equipo.

Creación de instancias

No se pueden crear instancias de IANet_Configuration.

Eliminación de instancias

No se pueden eliminar instancias de IANet_Configuration.

Modificación de propiedades

No hay propiedades que el usuario pueda modificar en esta clase.

Asociaciones

Existirá una instancia de IANet_AdapterConfiguration para asociar a cada adaptador (IANet_EthernetAdapter) con su configuración. Existirá una instancia de IANet_VLANConfiguration para asociar a cada VLAN (IANet_VLAN) con su configuración. Existirá una instancia de IANet_BootAgentConfiguration para asociar a cada agente de inicio (IANet_BootAgent) con su configuración.

Métodos

No hay métodos compatibles con esta clase.

Atributos no compatibles

Ninguno.

[Principio](#)

IANet_Setting

Propósito

Esta clase abstracta se usa para describir un propiedad configurable en una configuración. Esta clase se deriva de CIM_Setting.

Instancias

Una instancia separada de esta clase existirá para cada configuración de cada adaptador, VLAN o equipo. Los valores no se comparten entre las configuraciones.

Hay varias subclases de IANet_Setting. Las subclases corresponden a los tipos y gamas de valores distintos que pueden adoptar las configuraciones. Cada subclase corresponde a un estilo diferente de GUI que puede utilizarse para visualizar o cambiar las configuraciones.

Creación de instancias

No se pueden crear instancias de IANet_Setting.

Eliminación de instancias

No se pueden eliminar instancias de IANet_Setting.

Modificación de propiedades

Esta clase abstracta no tiene propiedades modificables, sin embargo las clases secundarias sí tiene propiedades modificables (vea a continuación).

Asociaciones

Cada instancia de IANet_Setting está asociada con una instancia de IANet_Configuration que usa una instancia de IANet_SettingContext.

Métodos

No hay métodos compatibles con esta clase. Para cambiar una configuración, modifique la propiedad necesaria y llame a **PutInstance**.

Atributos no compatibles

No se usa SettingID.

[Principio](#)

IANet_SettingInt

Propósito

Esta clase modela una configuración que adquiere un valor de número entero. Existen varias clases de configuración de IANet que se utilizan para modelar números enteros. Las diferencias entre estas clases se relacionan con la forma en la que la GUI visualiza y modifica el número entero y la manera en la cual los proveedores lo validan. Para IANet_SettingInt, se espera que la GUI

mostrará un cuadro de edición con un control numérico.

Instancias

Existe una instancia de esta clase para cada configuración que debe visualizarse como un cuadro de edición de número entero.

Creación de instancias

No se pueden crear instancias de esta clase.

Eliminación de instancias

No se pueden eliminar instancias de esta clase.

Modificación de propiedades

El atributo "CurrentValue" es la única propiedad de esta clase que puede modificarse. Se puede modificar esta propiedad usando **IWbemClassObject::Put()** para cambiar el valor y luego llame a **IWbemServices::PutInstance()** para actualizar esa configuración. Los proveedores verificarán que:

CurrentValue <= **max**

CurrentValue >= **min**

(**CurrentValue** - **min**) es un múltiplo de **Step**

Dónde **max**, **min**, **CurrentValue** y **Step** son todos atributos de **IANet_SettingInt**.

Asociaciones

Cada instancia de **IANet_SettingInt** está asociada con una instancia de **IANet_Configuration** que usa una instancia de **IANet_SettingContext**.

Atributos no compatibles

No se usa **SettingID**.

Métodos

No hay métodos compatibles con esta clase. Para cambiar una configuración, modifique la propiedad necesaria y llame a **PutInstance**.

[Principio](#)

IANet_SettingEnum

Propósito

Esta clase modela una configuración que adquiere un valor de número entero. Existen varias clases de configuración de **IANet** que se utilizan para modelar números enteros. Las diferencias entre estas clases se relacionan con la forma en la que la GUI visualiza y modifica el número entero y la manera en la cual los proveedores lo validan. Para **IANet_SettingEnum**, se espera que la GUI visualizará una lista de cadenas que se asignan a un número reducido de valores enumerados (por ejemplo, un cuadro combinado con una lista desplegable).

Instancias

Existe una instancia de esta clase para cada configuración que debe visualizarse como una enumeración.

Creación de instancias

No se pueden crear instancias de esta clase.

Eliminación de instancias

No se pueden eliminar instancias de esta clase.

Modificación de propiedades

El atributo **CurrentValue** es la única propiedad de esta clase que puede modificarse. A fin de modificar esta propiedad use **Put()** para cambiar el valor y luego llame a **PutInstance()** para actualizar esa configuración. Los proveedores van a verificar que **CurrentValue** ? **PossibleValues[]**

Asociaciones

Cada instancia de **IANet_SettingEnum** está asociada con una instancia de **IANet_Configuration** que usa una instancia de **IANet_SettingContext**.

Atributos no compatibles

No se usa SettingID.

Métodos

No hay métodos compatibles con esta clase. Para cambiar una configuración, modifique la propiedad necesaria y llame a **PutInstance**.

[Principio](#)

IANet_SettingSlider

Propósito

Esta clase modela una configuración que adquiere un valor de número entero. Existen varias clases de configuración de IANet que se utilizan para modelar números enteros. Las diferencias entre estas clases se relacionan con la forma en la que la GUI visualiza y modifica el número entero y la manera en la cual los proveedores lo validan. Para IANet_SettingSlider, se espera que la GUI visualizará un deslizador que le permitirá elegir el valor de manera gráfica ya que el valor elegido realmente no necesita mostrarse.

Instancias

Existe una instancia de esta clase para cada configuración que debe visualizarse como un deslizador.

Creación de instancias

No se pueden crear instancias de esta clase.

Eliminación de instancias

No se pueden eliminar instancias de esta clase.

Modificación de propiedades

El atributo **CurrentValue** es la única propiedad de esta clase que puede modificarse. A fin de modificar esta propiedad use **Put()** para cambiar el valor y luego llame a **PutInstance()** para actualizar esa configuración. Los proveedores van a verificar que **CurrentValue ? PossibleValues[]**

Asociaciones

Cada instancia de IANet_SettingSlider está asociada con una instancia de IANet_Configuration que usa una instancia de IANet_SettingContext.

Atributos no compatibles

No se usa SettingID.

Métodos

No hay métodos compatibles con esta clase. Para cambiar una configuración, modifique la propiedad necesaria y llame a **PutInstance**.

[Principio](#)

IANet_SettingMultiSelection

Propósito

Esta clase modela una configuración en la cual se pueden seleccionar varias opciones de una lista de opciones. Para IANet_SettingMultiSelection, se espera que la GUI visualizará un cuadro de lista de varias selecciones que le permiten elegir cualquiera (o ninguna) de las opciones.

Instancias

Existe una instancia de esta clase para cada configuración que debe visualizarse como una selección variada.

Creación de instancias

No se pueden crear instancias de esta clase.

Eliminación de instancias

No se pueden eliminar instancias de esta clase.

Modificación de propiedades

El atributo **CurrentValue** es la única propiedad de esta clase que puede modificarse. A fin de modificar esta propiedad use **Put()** para cambiar el valor y luego llame a **PutInstance()** para actualizar esa configuración. Los proveedores van a verificar que **CurrentValue ? PossibleValues[]**.

Asociaciones

Cada instancia de IANet_SettingMultiSelection está asociada con una instancia de IANet_Configuration que usa una instancia de IANet_SettingContext.

Atributos no compatibles

No se usa SettingID.

Métodos

No hay métodos compatibles con esta clase. Para cambiar una configuración, modifique la propiedad necesaria y llame a **PutInstance**.

[Principio](#)

IANet_SettingString

Propósito

Esta clase modela una configuración que permite escribir una cadena de valor de formato libre. Para IANet_SettingMultiSelection, se espera que la GUI visualizará un cuadro de edición.

Instancias

Existe una instancia de esta clase para cada configuración que debe visualizarse como un cuadro de edición.

Creación de instancias

No se pueden crear instancias de esta clase.

Eliminación de instancias

No se pueden eliminar instancias de esta clase.

Modificación de propiedades

El atributo **CurrentValue** es la única propiedad de esta clase que puede modificarse. A fin de modificar esta propiedad use **Put()** para cambiar el valor y luego llame a **PutInstance()** para actualizar esa configuración.

Asociaciones

Cada instancia de IANet_SettingMultiSelection está asociada con una instancia de IANet_Configuration que usa una instancia de IANet_SettingString.

Métodos

No hay métodos compatibles con esta clase.

Atributos no compatibles

No se usa SettingID.

Métodos

No hay métodos compatibles con esta clase. Para cambiar una configuración, modifique la propiedad necesaria y luego llame a **PutInstance**.

Sírvase leer todas las [restricciones y renunciaciones de responsabilidad](#).

[Regreso a la pantalla de contenido](#) [Principio](#)

Esquema de equipo: Guía del usuario de proveedores WMI y CDM para adaptadores de red Intel® PRO

[Visión general](#)

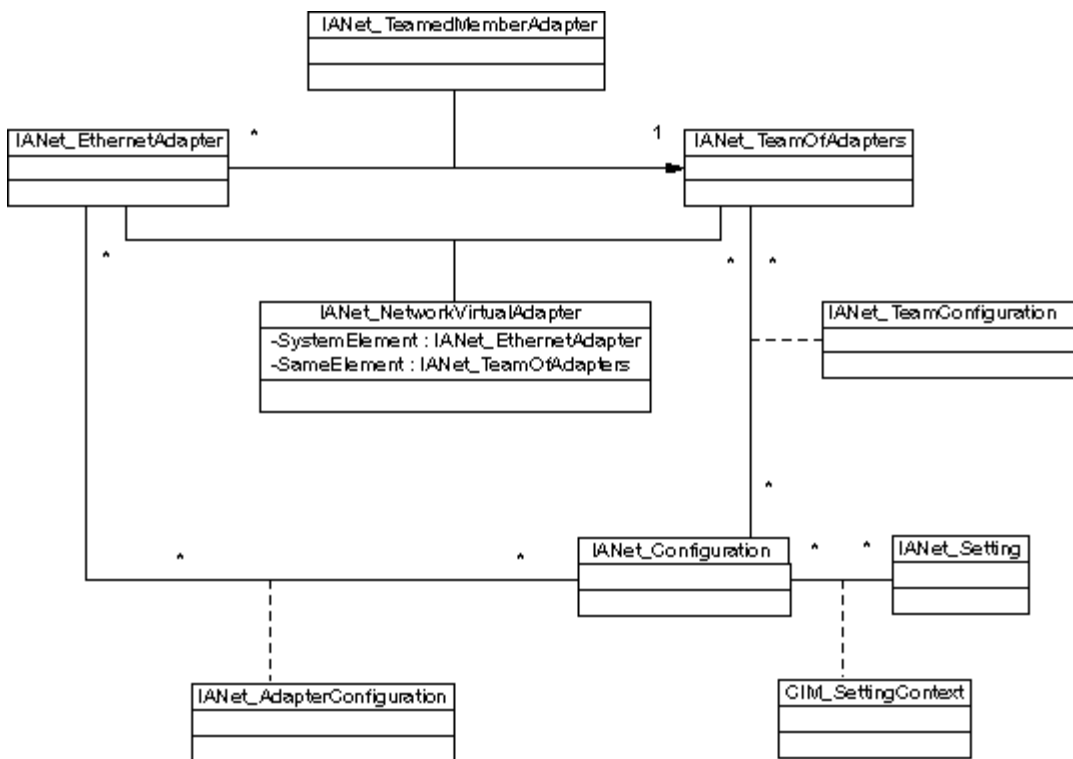
[IANet_TeamOfAdapters](#)

[IANet_TeamedMemberAdapter](#)

[IANet_NetworkVirtualAdapter](#)

Visión general

El esquema de equipo describe la manera como los adaptadores Ethernet se agrupan en equipos.



[Principio](#)

IANet_TeamOfAdapters

Propósito

Esta clase implementa la clase CIM_RedundancyGroup. Esta clase tiene miembros que describen el tipo de equipo, el número de adaptadores en el equipo y el número máximo de adaptadores que pueden pertenecer al equipo.

Instancias

Hay una instancia de esta clase para cada equipo Intel.

Creación de instancias

Para crear un equipo vacío, se debe crear una instancia de IANet_TeamOfAdapters. Debe fijar el **TeamingMode** correcto antes de llamar a **IWbemServices::PutInstance()** para crear el objeto en los proveedores. Los proveedores van a devolver una cadena que incluye la ruta al nuevo objeto.

Eliminación de instancias

De forma correspondiente, para eliminar un equipo, elimine la instancia de IANet_TeamOfAdapters. Los proveedores van a eliminar las asociaciones a los miembros del equipo y el adaptador virtual, y la configuración del equipo.

Modificación de propiedades

Use **Put()** para cambiar el valor de la propiedad **TeamingMode**, luego llame a **PutInstance()** para actualizar el equipo.

Asociaciones

Cada adaptador de un equipo está asociado con la instancia IANet_TeamOfAdapters del equipo que usando una instancia de IANet_TeamMemberAdapter. El adaptador virtual del equipo está asociado con esta clase usando una instancia de IA_NetNetworkVirtualAdapter.

Métodos

Esta instancia de clase acepta los siguientes métodos:

TestSwitchConfiguration: Prueba la configuración del conmutador para cerciorarse de que el equipo está funcionando correctamente con el conmutador. Esta prueba se puede utilizar para comprobar que los interlocutores de enlace (es decir, un dispositivo al cual se enlaza el adaptador, como sería otro adaptador, concentrador, conmutador, etc.) es compatible con el modo de generación de equipos del adaptador que se seleccionó. Por ejemplo, si el adaptador es un miembro de un equipo de Agregación de enlaces, mediante esta prueba se puede verificar que los interlocutores de enlace conectados a ese adaptador son compatibles con la Agregación de enlaces.

[Principio](#)

IANet_TeamedMemberAdapter

Propósito

Esta clase se usa para asociar al adaptador con el equipo y determina la función de ese adaptador en el equipo, así como establece que el adaptador está activo actualmente en el equipo. Esta clase implementa la clase CIM de CIM_NetworkAdapterRedundancyComponent.

Instancias

Existe una instancia de esta clase para cada adaptador que es miembro de un equipo.

Creación de instancias

Para agregar un adaptador a un equipo, se debe crear una instancia de IANet_TeamedMemberAdapter a fin de asociar el adaptador con el equipo.

Eliminación de instancias

Para eliminar un adaptador del equipo, elimine la instancia de IANet_TeamedMemberAdapter. El adaptador dejará de ser parte del equipo y podrá ligarse a un extremo de protocolo IP después de llamar a la función **Apply()**.

Modificación de propiedades

La propiedad **AdapterFunction** de esta clase se puede modificar para describir cómo se utilizará el adaptador dentro del equipo.

Asociaciones

Esta es una clase de asociación.

Métodos

No hay métodos compatibles con esta clase.

[Principio](#)

IANet_NetworkVirtualAdapter

Propósito

Esta clase se utiliza para asociar IA_Net_TeamOfAdapters del equipo con el IANet_EthernetAdapter que representa el adaptador virtual del equipo. Esta clase implementa la clase CIM de CIM_CIM_NetworkVirtualAdapter.

Instancias

Habr  una instancia de esta clase para cada equipo Intel que se ha ligado a un adaptador virtual.

Creaci3n de instancias

No se pueden crear instancias de esta clase. Para crear un equipo se debe crear una instancia de IANet_TeamOfAdapters. Esta clase existir  hasta despu s de haber llamado a **IANet_NetService .Apply()** dentro del contexto de una sesi3n v lida y despu s de que se haya creado una instancia de IANet_EthernetAdapter.

Eliminaci3n de instancias

No se pueden eliminar instancias de esta clase.

Asociaciones

Esta es una clase de asociaci3n.

M todos

No hay m todos compatibles con esta clase.

S rvase leer todas las [restricciones y renunciaciones de responsabilidad](#).

[Regreso a la pantalla de contenido](#) [Principio](#)

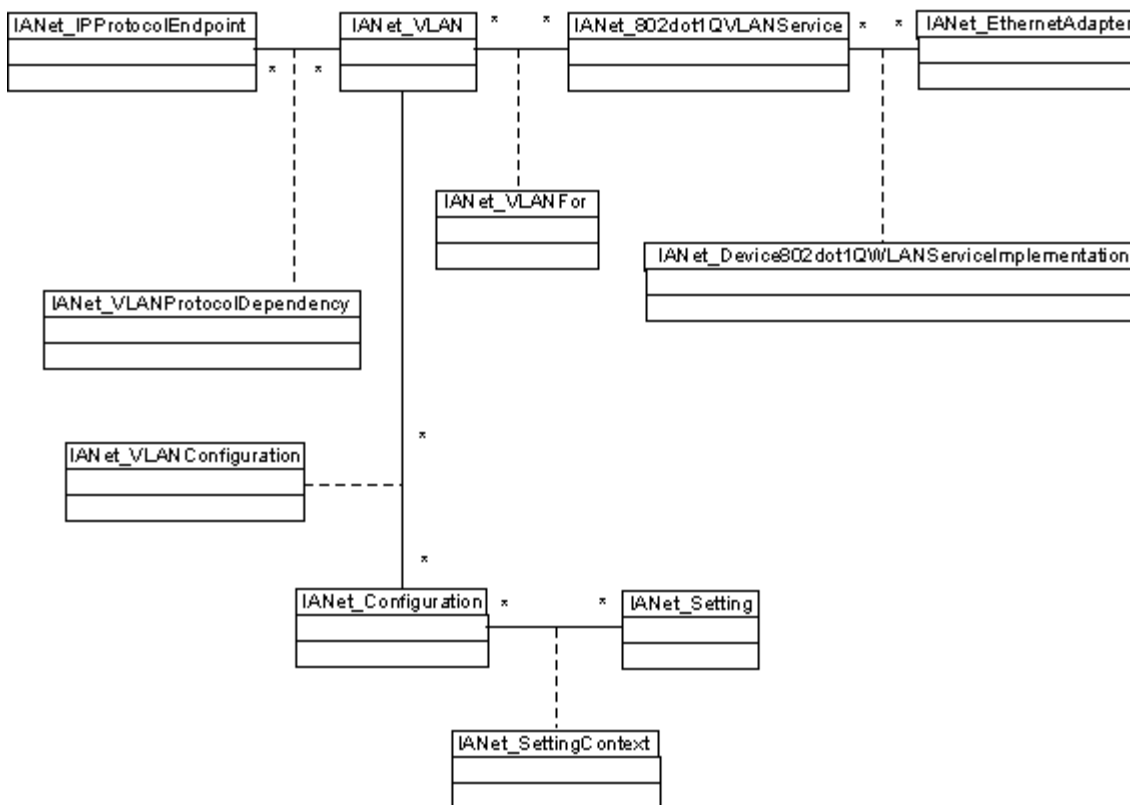
Esquema VLAN: Guía del usuario de proveedores WMI y CDM para adaptadores de red Intel® PRO

[Visión general](#)

[IANet_802dot1QVLANService](#)

[IANet_VLAN](#)

Visión general



[Principio](#)

IANet_802dot1QVLANService

Propósito

Esta clase se utiliza para retener las propiedades IEEE 802.1Q de un adaptador de red. La clase implementa la clase CIM de CIM_802dot1QVLANService.

Instancias

Existe una instancia de esta clase para cada adaptador o equipo compatible con IEEE 802.1Q. Cada adaptador puede tener solamente un IANet_802dot1QVLANService. Algunos equipos, como los equipos de tolerancia a fallos para varios vendedores no son compatibles con este servicio.

Anomalía

Un equipo sin VLAN no tiene un servicio VLAN salvo que se enumere dentro del contexto de una sesión válida. Para los equipos,

la instancia 802.3QVlanService aparecerá en las situaciones siguientes únicamente:

- Si es equipo ya tiene VLAN
- Si el equipo no tiene VLAN y se utiliza un controlador de sesión en el contexto mientras se está enumerando esta clase.

Creación de instancias

No se pueden crear instancias de esta clase. Si un adaptador no tiene una instancia de esta clase asociada, el adaptador no es compatible con este servicio.

Eliminación de instancias

No se pueden eliminar instancias de esta clase.

Modificación de propiedades

No hay propiedades que se puedan modificar en esta clase.

Asociaciones

Cada instancia de la clase será asociada con un IANet_EthernetAdapter utilizando una instancia de IANet_DeviceServiceImplementation.

Cada instancia de IANet_802dot1QVLANService puede aceptar varias VLAN; cada VLAN se asociará con la instancia utilizando la asociación IANet_VLANFor.

Métodos

uint16 CreateVLAN([in] uint32 VLANNumber, [in] string Name, [out] IANet_VLAN REF VLANpath); Se usa para crear una VLAN en el adaptador o en el equipo. El cliente deberá proporcionar el número y el nombre de la VLAN y obtener la ruta del objeto de la VLAN recién creada.

[Principio](#)

IANet_VLAN

Propósito

Esta clase retiene la información de cada VLAN Intel. Esta clase implementa CIM_VLAN.

Instancias

Va a existir una instancia de esta clase para cada VLAN Intel.

Creación de instancias

Para crear una VLAN, llame a **CreateVLAN** en la instancia apropiada de IANet_802dot1QVLANService.

Eliminar instancias

Se puede eliminar una instancia de esta clase para eliminar la VLAN correspondiente.

Modificación de propiedades

Se puede modificar el atributo VLANNumber y Caption.

Asociaciones

Cada instancia está asociada con una instancia de IANet_802dot1QVLANService, y, en consecuencia, a una instancia de IANet_EthernetAdapter, utilizando la clase IANet_VLANFor.

Cada instancia puede asociarse con varias instancias IANet_Configuration para agrupar un conjunto de configuraciones de la VLAN. Para esta versión del proveedor, hay solamente un objeto IANet_Configuration por cada VLAN.

Cada instancia puede asociarse con un IANet_IPProtocolEndpoint para proporcionar la configuración IP para la VLAN que utiliza la clase IANet_VLANProtocolDependency.

Métodos

Ninguno

Sírvase leer todas las [restricciones y renunciaciones de responsabilidad](#).

Obtener la configuración actual: Guía del usuario de proveedores WMI y CDM para adaptadores de red Intel® PRO

- [Obtener los adaptadores físicos](#)
- [Obtener los dispositivos PCI](#)
- [Obtener la configuración de los adaptadores](#)
- [Obtener la configuración del equipo](#)
- [Obtener la configuración del equipo](#)
- [Obtener la configuración de la VLAN](#)
- [Obtener los valores de configuración de la VLAN](#)
- [Obtener la información del protocolo IP](#)
- [Obtener la información del agente de inicio](#)
- [Obtener la configuración del agente de inicio](#)

El cliente no necesita un controlador de sesión para leer la configuración actual. Los clientes pueden utilizar un contexto NULO, sin embargo cualquier mensaje de error aparecerá en el idioma predeterminado de la máquina gestionada. En la tabla que sigue, los elementos entre { } son rutas de objetos. Se supone que estas rutas se obtuvieron de consultas WQL previas. El cliente nunca debería necesitar construir una ruta de objeto sin hacer una consulta. El atributo **__PATH** de todos los objetos incluye la ruta del objeto para ese objeto.

En todos los casos de utilización siguientes, los métodos **IWbemServices::ExecQuery** o **IWbemServices::ExecQueryAsync** se usan para ejecutar las consultas WQL.

Obtener los adaptadores físicos

Tarea	Consulta WQL	Clase resultante	Comentario
Enumerar todos los adaptadores	SELECT * FROM IANet_EthernetAdapter	IANet_EthernetAdapter	Devuelve todas las IANet_EthernetAdapters. Es equivalente a IWbemServices::CreateInstanceEnumAsync.
Determinar si el adaptador es virtual	ASSOCIATORS OF {ruta al adaptador} WHERE AssocClass = IANet_NetworkVirtualAdapter	IANet_TeamOfAdapters	Si la consulta responde sin clases, entonces el adaptador es un adaptador real.
Determinar si el adaptador es un adaptador fantasma	ASSOCIATORS OF {ruta al adaptador} WHERE ResultClass = IANet_EthernetPCIDevice	IANet_EthernetPCIDevice	Si el adaptador no es virtual y la consulta responde sin objetos, entonces el adaptador es un adaptador fantasma.

La clase principal de los adaptadores es IANet_EthernetAdapter. Esta clase se utiliza tanto para los adaptadores físicos como para los virtuales y el cliente debe de saber cómo distinguir entre ambos.

[Principio](#)

Obtener los dispositivos PCI

Las clases principales son IANet_EthernetPCIDevice, IANet_PCIDevice e IANet_AdapterDevice (una clase de asociación para asociar a un adaptador con su dispositivo).

Para este caso, la clase de asociación no incluye datos, es decir, no tiene un valor por sí misma, IANet_EthernetPCIDevice

hereda de IANet_PCIDevice e incluye atributos adicionales que son específicos para un dispositivo PCI que es un adaptador Ethernet.

Tarea	Consulta WQL	Clase resultante	Comentario
Enumerar los dispositivos PCI de la red	SELECT * FROM IANet_PCIDevice	IANet_PCIDevice	Esta consulta podrá responder con módems y otros dispositivos no configurados por Intel® PROSet. Es equivalente a IWbemServices::CreateInstanceEnumAsync.
Enumerar todos los dispositivos PCI utilizados por los adaptadores configurados por Intel® PROSet.	SELECT * FROM IANet_EthernetPCIDevice	IANet_EthernetPCIDevice	Esta consulta responde solamente con los dispositivos PCI gestionados por Intel® PROSet. Es equivalente a IWbemServices::CreateInstanceEnumAsync.
Determinar si el adaptador está instalado.	N/D	IANet_EthernetPCIDevice	Verifique el atributo "Availability" (Disponibilidad) de IANet_EthernetPCIDevice. Si es igual a 10 - "Not Installed" (No instalado), entonces el dispositivo no ha sido instalado. Nota: En este momento no se conoce mucha información acerca del dispositivo.
Obtener el el dispositivo PCI asociado con un adaptador	ASSOCIATORS OF {ruta a IANet_EthernetAdapter} WHERE ResultClass = IANet_EthernetPCIDevice	IANet_EthernetPCIDevice	Si esto responde sin objetos, entonces el adaptador es virtual o es un adaptador fantasma.
Obtener el adaptador asociado con un dispositivo PCI	ASSOCIATORS OF {ruta al dispositivo Ethernet PCI} WHERE ResultClass = IANet_EthernetAdapter	IANet_EthernetAdapter	Esta consulta no es óptima para el Proveedor, es preferible que el cliente comience con el adaptador.

Principio

Obtener la configuración del adaptador

Los objetos de configuración no se asocian directamente con el adaptador. De acuerdo con el estándar CIM, se asocian con un objeto de configuración, que está asociado con el adaptador.

Las clases que participan en esta parte del esquema son IANet_EthernetAdapter, IANet_Configuration, IANet_SettingInt, IANet_SettingString, IANet_SettingEnum, IANet_SettingMultiSelection e IANetSettingSlider.

Las clases de asociación IANet_AdapterConfiguration e IANet_SettingContext no contienen datos reales y actúan como aglutinadotes entre los valores y el objeto del cual dependen.

Tarea	Consultas WQL	Clase resultante	Comentarios
Obtener el objeto de Configuración para el adaptador	ASSOCIATORS OF {ruta a IANet_EthernetAdapter} WHERE ResultClass = IANet_Configuration	IANet_Configuration	Responde con un objeto exclusivamente, aun cuando no existan valores, siempre habrá un objeto de configuración. La ruta al objeto de este objeto se utiliza en la consulta siguiente.

Obtener la configuración asociada con un adaptador	ASSOCIATORS OF {ruta a IANet_Configuration} WHERE AssocClass = IANet_SettingContext	Una mezcla de IANet_SettingInt, IANet_SettingString, IANet_SettingEnum, IANet_SettingMultiSelection, IANet_SettingSlider	Responde con todas las clases de configuración asociadas con el adaptador. El cliente debería usar el atributo __CLASS para determinar ese tipo de cada valor.
--	--	--	--

[Principio](#)

Obtener la configuración del equipo

Las clases principales en el esquema de generación de equipos son IANet_EthernetAdapter, IANet_TeamOfAdapters, IANet_NetworkVirtualAdapter e IANet_TeamedMemberAdapter.

El desafío de este esquema consiste en que existe una instancia de IANet_EthernetAdapter para cada adaptador físico y para cada adaptador virtual. El cliente debe de tener la capacidad de distinguir entre el adaptador virtual de un equipo y los adaptadores que son miembros del equipo.

La clase de asociación IANet_NetworkVirtualAdapter no incluye ningún dato útil, los clientes están interesados solamente en los extremos de esta asociación. IANet_TeamedMemberAdapter sí incluye datos útiles sobre la forma como el adaptador miembro se utiliza dentro del equipo.

Tarea	Consultas WQL	Clase resultante	Comentarios
Enumerar todos los equipos	SELECT * FROM IANet_TeamOfAdapters	IANet_TeamOfAdapters	Existe una instancia de IANet_TeamOfAdapters para cada equipo. Es equivalente a IWbemServices::CreateInstanceEnumAsync.
Obtener el adaptador virtual de un equipo	ASSOCIATORS OF {ruta a IANet_TeamOfAdapters} WHERE AssocClass = IANet_NetworkVirtualAdapter	IANet_EthernetAdapter	Responde solamente con el objeto adaptador para el adaptador virtual del equipo. Este adaptador no va a existir si el equipo ha sido creado pero no se ha llamado a Apply (Aplicar). (consulte más adelante sobre la actualización de la configuración).
Enumerar los adaptadores miembros del equipo	ASSOCIATORS OF {ruta a IANet_TeamOfAdapters} WHERE AssocClass = IANet_NetworkVirtualAdapter	IANet_EthernetAdapter	Responde con los adaptadores que forman parte del equipo, pero no describe el papel que juega el adaptador.
Determinar el papel que juega un adaptador en el equipo	REFERENCES OF {ruta a IANet_EthernetAdapter} WHERE ResultClass = IANet_TeamedMemberAdapter	IANet_TeamedMemberAdapter	La clase incluye la información sobre cómo el adaptador se relaciona con el equipo y su estado actual dentro del equipo.

[Principio](#)

Obtener la configuración del equipo

Los objetos de configuración no se asocian directamente con el equipo. De acuerdo con el estándar CIM, se asocian con un objeto de configuración, que está asociado con el objeto IANet_EthernetAdapter virtual del equipo. El mismo objeto de configuración está asociado también con el objeto IANet_TeamOfAdapters del equipo.

Las clases que participan en esta parte del esquema son IANet_EthernetAdapter, IANet_Configuration, IANet_SettingInt, IANet_SettingString, IANet_SettingEnum, IANet_SettingMultiSelection e IANetSettingSlider.

Las clases de asociación IANet_AdapterConfiguration e IANet_SettingContext no contienen datos reales y actúan como aglutinadores entre los valores y el objeto del cual dependen. Es exactamente lo mismo que en el caso de los valores de los adaptadores.

Tarea	Consultas WQL	Clase resultante	Comentarios
Obtener el objeto de configuración del equipo, comenzando con el adaptador virtual	ASSOCIATORS OF {ruta a IANet_EthernetAdapter} WHERE ResultClass = IANet_Configuration	IANet_Configuration	Responde con un objeto exactamente. Aun cuando no existan los valores, siempre existirá un objeto de configuración. La ruta al objeto de este objeto se utiliza en la consulta siguiente.
Obtener el objeto de configuración del equipo, comenzando con el equipo de adaptadores	ASSOCIATORS OF {ruta a IANet_EthernetAdapter} WHERE ResultClass = IANet_Configuration		
Obtener la configuración asociada con un adaptador	ASSOCIATORS OF {ruta a IANet_Configuration} WHERE AssocClass = IANet_SettingContext	Una mezcla de IANet_SettingInt, IANet_SettingString, IANet_SettingEnum, IANet_SettingMultiSelection, IANet_SettingSlider	Responde con todas las clases de configuración asociadas con el adaptador. El cliente debería usar el atributo __CLASS para determinar ese tipo de cada valor.

[Principio](#)

Obtener la configuración de la VLAN

Cada adaptador compatible con VLAN tiene una IANet_802dot1QVLANService asociada, usando la clase de asociación IANet_Device802do1QVLANServiceImplementation. Si un adaptador no tiene una instancia de esta clase asociada, no es compatible con VLAN.

Cada VLAN está representada por una instancia de IANet_VLAN. La VLAN no está asociada directamente con el adaptador, está asociada con IANet_802dot1QVLANService del adaptador.

La clase de asociación IANet_VLANFor se usa para asociar cada instancia de VLAN con la IANet_802dot1QVLANService correcta. Esta clase no incluye ningún dato útil para el usuario.

Tarea	Consultas WQL	Clase resultante	Comentarios
Obtener el objeto de servicio 802.1q VLAN asociado con un adaptador	ASSOCIATORS OF {ruta a IANet_EthernetAdapter} WHERE ResultClass = IANet_802dot1QVLANService	IANet_802dot1QVLANService	Responde con un objeto o ninguno.
Obtener las VLAN en un adaptador	ASSOCIATORS OF {ruta a IANet_802dot1QVLANService} WHERE ResultClass = IANet_VLAN	IANet_VLAN	Esto puede responder con no objetos si no hay VLAN instaladas.

[Principio](#)

Obtener los valores de la configuración de las VLAN

Los objetos de configuración no se asocian directamente con la VLAN. De acuerdo con el estándar CIM, se asocian con un objeto de configuración, que está asociado con el objeto IANet_VLAN para la VLAN.

Las clases que participan en esta parte del esquema son IANet_EthernetAdapter, IANet_Configuration, IANet_SettingInt, IANet_SettingString, IANet_SettingEnum, IANet_SettingMultiSelection e IANetSettingSlider.

Las clases de asociación IANet_VLANConfiguration e IANet_SettingContext no contienen datos reales, actúan como aglutinadores entre los valores y el objeto del cual dependen. Es exactamente lo mismo que en el caso de los valores de los adaptadores.

Tarea	Consultas WQL	Clase resultante	Comentarios
Obtener el objeto de Configuración para VLAN	ASSOCIATORS OF {ruta a IANet_EthernetAdapter} WHERE ResultClass = IANet_Configuration	IANet_Configuration	Responde con un objeto exclusivamente, aun cuando no existan valores, siempre habrá un objeto de configuración. La ruta al objeto de este objeto se utiliza en la consulta siguiente.
Obtener la configuración asociada con una VLAN	ASSOCIATORS OF {ruta a IANet_Configuration} WHERE AssocClass = IANet_SettingContext	Una mezcla de IANet_SettingInt, IANet_SettingString, IANet_SettingEnum, IANet_SettingMultiSelection, IANet_SettingSlider	Responde con todas las clases de configuración asociadas con la VLAN. El cliente debería usar el atributo __CLASS para determinar el tipo de cada valor.

[Principio](#)

Obtener la información del protocolo IP

El Proveedor ofrecerá alguna información limitada acerca de los extremos de protocolo IP asociados con adaptadores, VLAN y equipos. No hay compatibilidad con ningún otro protocolo.

La clase principal que incluye la información del protocolo es IANet_IPProtocolEndpoint. Hay dos clases de asociación: IANet_VLANProtocolDependency e IANet_AdapterProtocolImplementation. Para obtener el extremo IP de un equipo, obtenga primero el IANet_EthernetAdapter virtual del equipo, es decir, el extremo IP asociado con esta instancia.

Tarea	Consultas WQL	Clase resultante	Comentarios
Obtener los extremos de protocolo IP asociados con un adaptador	ASSOCIATORS OF {ruta a IANet_EthernetAdapter} WHERE ResultClass = IANet_IPProtocolEndpoint	IANet_IPProtocolEndpoint	A pesar de que algunos adaptadores tendrán más de una dirección IP, estarán asociados solamente con una instancia de extremo de protocolo IP.
Obtener los extremos de protocolo IP asociados con una VLAN.	ASSOCIATORS OF {ruta a IANet_VLAN} WHERE ResultClass = IANet_IPProtocolEndpoint	IANet_IPProtocolEndpoint	Habrà solamente un IP Protocol Endpoint asociado con la VLAN.

[Principio](#)

Obtener la información del agente de inicio

Cada adaptador que es compatible con un agente de inicio en ROM flash tendrá una instancia de IANet_BootAgent asociada con el mismo utilizando la clase de asociación IANet_DeviceBootServiceImplementation

Tarea	Consultas WQL	Clase resultante	Comentarios
Obtener el agente de inicio asociado con un adaptador	ASSOCIATORS OF {ruta a IANet_EthernetAdapter} WHERE ResultClass = IANet_BootAgent	IANet_BootAgent	Los atributos de sólo lectura proporcionan la información de la imagen ROM de inicio para este adaptador. InvalidImageSignature, Version, UpdateAvailable, FlashImageType

[Principio](#)

Obtener la configuración del agente de inicio

Los objetos de configuración no se asocian directamente con el agente de inicio. De acuerdo con el estándar CIM, se asocian con un objeto de configuración, que está asociado con agente de inicio.

Las clases que participan en esta parte del esquema son IANet_BootAgent, IANet_Configuration, IANet_SettingInt, IANet_SettingString, IANet_SettingEnum, IANet_SettingMultiSelection e IANetSettingSlider.

Las clases de asociación IANet_VLANConfiguration e IANet_SettingContext no contienen datos reales, actúan como aglutinadotes entre los valores y el objeto del cual dependen.

Tarea	Consultas WQL	Clase resultante	Comentarios
Obtener el objeto de Configuración para el agente de inicio	ASSOCIATORS OF {ruta a IANet_BootAgent} WHERE ResultClass = IANet_Configuration	IANet_Configuration	Responde con un objeto exactamente. Aun cuando no existan los valores, siempre existirá un objeto de configuración. La ruta de objeto de este objeto se utiliza en la consulta siguiente.
Obtener la configuración asociada con un agente de inicio	ASSOCIATORS OF {ruta a IANet_Configuration} WHERE AssocClass = IANet_SettingContext	Una mezcla de IANet_SettingInt, IANet_SettingString, IANet_SettingEnum, IANet_SettingMultiSelection, IANet_SettingSlider	Responde con todas las clases de configuración asociadas con el adaptador. El cliente debería usar el atributo __CLASS para determinar ese tipo de cada valor.

Sírvase leer todas las [restricciones y renunciaciones de responsabilidad](#).

[Regreso a la pantalla de contenido](#) [Principio](#)

Actualización de la configuración: Guía del usuario de proveedores WMI y CDM para adaptadores de red Intel® PRO

[Visión general](#)

[Cambiar la configuración de un adaptador, equipo o VLAN](#)

[Crear un equipo nuevo \(vacío\)](#)

[Agregar un adaptador a un equipo](#)

[Eliminar un adaptador de un equipo](#)

[Eliminar un equipo](#)

[Cambiar el modo de un equipo](#)

[Cambiar la prioridad de un adaptador dentro de un equipo](#)

[Desinstalar un adaptador](#)

[Crear una VLAN](#)

[Cambiar los atributos de una VLAN](#)

[Eliminar una VLAN](#)

[Actualizar el agente de inicio](#)

Visión general

En la mayoría de los casos, para actualizar la configuración, la aplicación cliente deberá obtener un controlador de sesión de la clase IANet_NetService y guardar esta identificación en un objeto de contexto IWbemContext. Los cambios de la configuración ocurrirán solamente cuando se llama el método **Apply** en IANet_NetService. Este requisito tiene algunas excepciones:

- Los cambios a la clase de agente de inicio ocurrirán inmediatamente en cuanto se producen y no requieren un controlador de sesión.
- Algunas llamadas de método (por ejemplo, identify adapter) ocasionan que se lleve a cabo una operación antes de que se llame a **Apply**.

Para algunas operaciones, se puede utilizar el calificador PreCheck en el contexto para verificar si es que se permitió un operación. Esto permite la utilización de una interfaz de usuario a fin de inhabilitar ciertos controles o elementos del menú si fuera necesario.

[Principio](#)

Cambiar la configuración del adaptador, equipo o VLAN

El cambio de la configuración de un adaptador, equipo o VLAN:

- Requiere de un controlador de sesión.
- PreCheck está disponible.
- Requiere una llamada a **Apply** antes de que se ejecute la operación..

Para cambiar la configuración de un adaptador, VLAN o equipo, el cliente deberá obtener primero la ruta del objeto de la configuración que se va a cambiar. La mejor manera de lograrlo es enumerar las configuraciones en el objeto y guardar el atributo **__PATH** de la configuración (vea lo indicado anteriormente).

Para el cliente la forma más fácil de actualizar una configuración es:

1. Obtener una instancia del objeto de configuración del WMI.
2. Modificar el atributo CurrentValue (usando **IWbemClassObject::Put()**).
3. Llamar a **IWbemServices::PutInstance()** para pasar la instancia modificada al proveedor WMI. Deberá llamarse a **PutInstance** con el indicador WBEM_FLAG_UPDATE_ONLY.

El Proveedor WMI validará el CurrentValue y devolverá WBEM_E_FAIL si es que falló la validación. La razón exacta de la falla

será devuelta en el atributo `Description` del objeto `IANet_ExtendedStatus`.

El establecimiento de descripciones específicas de los valores incluye:

- El valor del número entero era menor que el mínimo permitido.
- El valor de configuración de un número entero era mayor que el máximo permitido.
- El valor de configuración del número entero no es un paso permitido.
- La longitud de la configuración de cadena es mayor que el máximo permitido.
- El valor de configuración no es uno de los valores permitidos.

La última descripción es devuelta cuando el valor actual de `IANet_SettingEnum`, `IANet_SettingSlider` o `IANet_SettingMultiSelection` no es uno de los valores permitidos.

El único atributo de una configuración que puede cambiar el cliente es `CurrentValue`. El proveedor WMI ignorará los cambios que se hicieron a cualesquiera de los otros valores.

No hay métodos compatibles con esta clase. Para cambiar una configuración, modifique la propiedad `CurrentValue` y luego llame a **PutInstance**.

[Principio](#)

Crear un equipo nuevo (vacío)

La creación de un equipo nuevo:

- Requiere de un controlador de sesión.
- `PreCheck` está disponible.
- Requiere una llamada a **Apply** antes de que se ejecute la operación.

A fin de crear un equipo nuevo, se debe crear una instancia de `IANet_TeamOfAdapters` (es decir, use **`IWbemServices::GetObject()`** para obtener un objeto de clase para `IANet_TeamOfAdapters` y luego use **`IWbemServices::SpawnInstance()`** para crear una instancia de este objeto).

Luego, use **`IWbemClassObject::Put`** para fijar el atributo `TeamMode` en la instancia que será el tipo de equipo deseado (por ejemplo, `AFT`). Finalmente, llame a **`IWbemServices::PutInstance()`** para crear el equipo, pasando el indicador `WBEM_FLAG_CREATE_ONLY`.

La ruta al objeto del nuevo equipo se guarda en el **`IWbemCallResultObject`**, el cual se le transfiere de regreso en cuanto se completa la llamada. El método **`IWbemCallResult::GetResultString`** va a obtener la nueva ruta al objeto.

Si fallara esta acción, el cliente debería verificar `IANet_ExtendedStatus` para obtener las razones de la falla.

Las clases de `IANet_EthernetAdapter` virtual e `IANet_IPProtocolEndpoint` para el equipo están disponibles hasta después de la llamada a **Apply**. Se puede acceder a la configuración del equipo usando el objeto `IANet_Configuration` asociado con el `IANet_TeamOfAdapters` nuevo.

[Principio](#)

Agregar un adaptador a un equipo

La adición de un adaptador a un equipo

- Requiere de un controlador de sesión.
- `PreCheck` está disponible.
- Requiere una llamada a **Apply** antes de que se ejecute la operación.

Para crear un equipo nuevo, se debe crear una instancia de `IANet_TeamedMemberAdapter` (es decir, use **`IWbemServices::GetObject()`** para obtener un objeto de clase para `IANet_TeamedMemberAdapter` y luego use **`IWbemServices::SpawnInstance()`** para crear una instancia de este objeto).

Se deben establecer los atributos siguientes en el objeto utilizando `IWbemClassObject::Put()`:

- `GroupComponent` debe establecerse como la ruta de objeto completa de los `IANet_TeamOfAdapters` a los que se va a agregar el adaptador.
- `PartComponent` deberá establecerse en la ruta completa del objeto del `IANet_EthernetAdapter` que se va a agregar al equipo.

También se puede fijar la prioridad del adaptador en el equipo. Finalmente, llame a `IWbemServices::PutInstance()` para agregar el adaptador, pasando el indicador `WBEM_FLAG_CREATE_ONLY`. Si esta acción falla, busque el código de error en `IANet_ExtendedStatus`.

[Principio](#)

Eliminar un adaptador de un equipo existente

La eliminación de un adaptador de un equipo existente:

- Requiere de un controlador de sesión.
- `PreCheck` está disponible.
- Requiere una llamada a **Apply** antes de que se ejecute la operación.

Para eliminar un adaptador de un equipo, elimine la instancia de `IANet_TeamedMemberAdapter` que asocia el adaptador al equipo usando `IWbemServices::DeleteInstance()`. Si esta acción falla, busque el código de error en `IANet_ExtendedStatus`.

[Principio](#)

Eliminar un equipo

La eliminación de un equipo:

- Requiere de un controlador de sesión.
- `PreCheck` está disponible.
- Requiere una llamada a **Apply** antes de que se ejecute la operación.

Para eliminar un equipo, elimine la instancia `IANet_TeamOfAdapters` usando `IWbemServices::DeleteInstance()`. Si esta acción falla, busque el código de error en `IANet_ExtendedStatus`.

[Principio](#)

Cambiar el modo de un equipo

El cambio del modo de un equipo:

- Requiere de un controlador de sesión.
- `PreCheck` está disponible.
- Requiere una llamada a **Apply** antes de que se ejecute la operación.

Para cambiar el modo de un equipo, obtenga la instancia de `IANet_TeamOfAdapters` para el equipo (por ejemplo, use `IWbemServices::GetObject` utilizando la ruta de objeto del equipo). Luego, use `IWbemClassObject::Put` para cambiar el atributo `TeamMode` del equipo. Finalmente, llame a `IWbemClassObject::PutInstance` para indicar al proveedor WMI que actualice el modo del equipo, pasando el indicador `WBEM_FLAG_UPDATE_ONLY`. Si esta acción falla, busque el código de error en `IANet_ExtendedStatus`.

[Principio](#)

Cambiar la prioridad de un adaptador en un equipo

El cambio de la prioridad de un adaptador en un equipo:

- Requiere de un controlador de sesión.
- PreCheck está disponible.
- Requiere una llamada a **Apply** antes de que se ejecute la operación.

Para cambiar la prioridad de un adaptador, el cliente deberá obtener primero la instancia del adaptador de IANet_TeamedMemberAdapter. (por ejemplo, use **IWbemServices::GetObject** utilizando la ruta del objeto). Luego, el cliente puede usar **IWbemClassObject::Put** para cambiar el atributo AdapterFunction del adaptador. Finalmente, el cliente deberá llamar a **IWbemClassObject::PutInstance** para indicar al proveedor WMI que actualice la prioridad del adaptador. Si esta acción falla, busque el código de error en IANet_ExtendedStatus.

[Principio](#)

Desinstalar un adaptador

La desinstalación de un adaptador:

- Requiere de un controlador de sesión.
- PreCheck está disponible.
- Requiere una llamada a **Apply** antes de que se ejecute la operación.

Para desinstalar un adaptador, llame a **IWbemServices::DeleteInstance**, pasando la ruta del objeto del adaptador que se va a desinstalar.

[Principio](#)

Crear una VLAN

La creación de una VLAN:

- Requiere de un controlador de sesión.
- PreCheck está disponible.
- Requiere una llamada a **Apply** antes de que se ejecute la operación.

Para crear una VLAN, llame el método **CreateVLAN** en el IANet_802dot1QVLANService para el adaptador al que se va agregar la VLAN. Se deben pasar los argumentos siguientes al método:

- VLANNumber, que es el número de la VLAN. (Rango 1- 4094)
- Name, que es un nombre que define el usuario para identificar a la VLAN.

La función devolverá la ruta del objeto de la VLAN recién creada en el parámetro Out de VLANpath. Si esta acción falla, busque el código de error en IANet_ExtendedStatus.

[Principio](#)

Cambiar los atributos de una VLAN

El cambio de atributos de una VLAN:

- Requiere de un controlador de sesión.
- PreCheck está disponible.
- Requiere una llamada a **Apply** antes de que se ejecute la operación.

El cliente puede cambiar los atributos VLANNumber y VLANName de una VLAN. Para cambiar la prioridad de un adaptador, obtenga primero una instancia de IANet_VLAN para el adaptador (por ejemplo, use **IWbemServices::GetObject** utilizando la ruta del objeto).

Luego, cambie VLANNumber o VLANName a los valores deseados. Finalmente, llame a **IWbemClassObject:: PutInstance** para indicar al proveedor WMI que actualice el modo del equipo, pasando el indicador WBEM_FLAG_UPDATE_ONLY. Si esta acción falla, busque el código de error en IANet_ExtendedStatus.

[Principio](#)

Eliminar una VLAN

La eliminación de una VLAN:

- Requiere de un controlador de sesión.
- PreCheck está disponible.
- Requiere una llamada a **Apply** antes de que se ejecute la operación.

Para eliminar una VLAN llame a **IWbemServices::DeleteInstance**, pasando la ruta del objeto de VLAN que se va a eliminar.

[Principio](#)

Actualizar el agente de inicio

La actualización del agente de inicio:

- No requiere de un controlador de sesión.
- PreCheck no está disponible.
- No requiere una llamada a **Apply** antes de que se ejecute la operación.

El cliente puede actualizar la imagen del agente de inicio usando llamadas de método. Para leer o escribir la imagen flash, obtenga primero una instancia de IANet_BootAgent para el adaptador (por ejemplo, use **IWbemServices::GetObject** utilizando la ruta del objeto).

Luego, ejecute ReadFlash() para leer la imagen flash de inicio de ROM existente o ProgramFlash() para actualizar la imagen flash de inicio de ROM. Si esta acción falla, busque el código de error en IANet_ExtendedStatus.

Tarea	Métodos WMI	Resultado	Comentarios
Actualizar o insertar una imagen de inicio ROM para el adaptador	uint32 ProgramFlash([IN, ValueMap {"0", "1"}, Values { "Check Version", "Write Flash"}: Amended] uint32 Action, [IN] uint8 NewFlashData[], [OUT]	Si se especificó la acción "Check Version", el método devolverá un mensaje de advertencia si es que la imagen de inicio de ROM que se actualizaría como en un NewFlashData[] es más antigua que la versión que ya está presente en la NIC. Si se especificó la acción "Write", se actualiza la FLASH ROM en la NIC con NewFlashData[].	Este método se usa para actualizar la flash ROM de la NIC. La NIC dejará de comunicarse con la red mientras se actualiza la memoria flash.

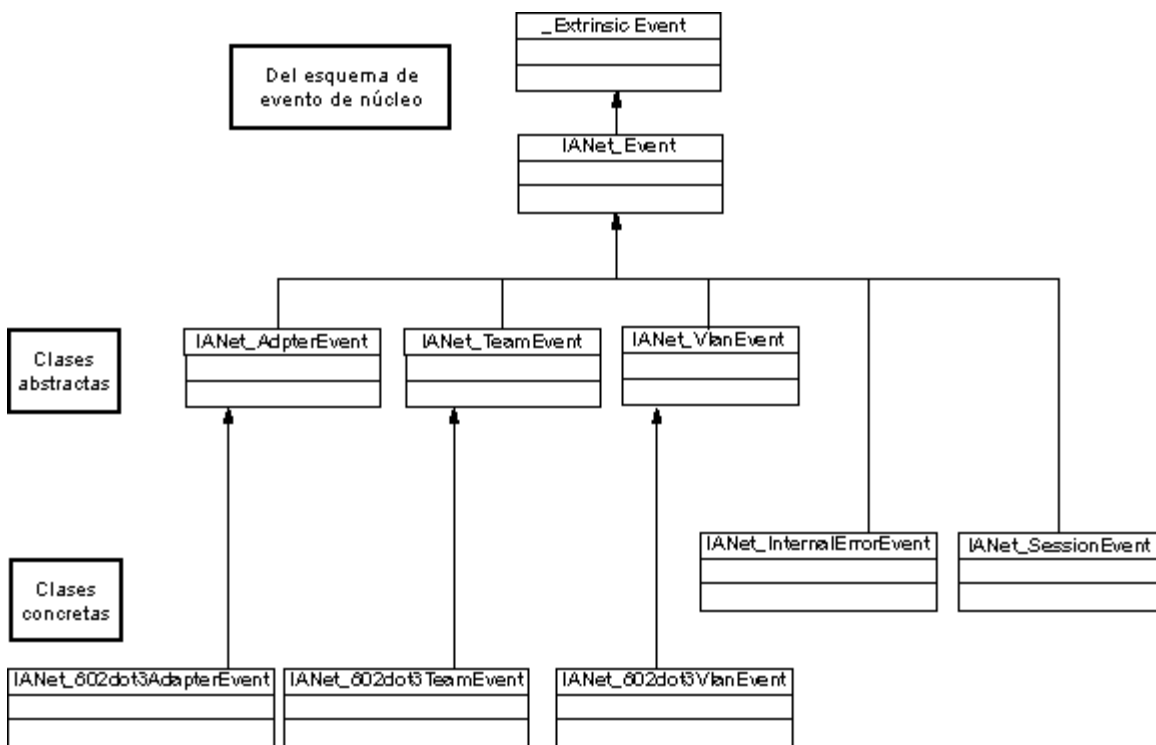
	<pre>string strErrorMessage);</pre>		
Leer la imagen de inicio de ROM	<pre>uint32 ReadFlash([OUT] uint8 FlashData[]);</pre>	FlashData[] incluye la imagen flash ROM de la NIC.	Este método lee la flash ROM de la NIC, que puede guardarse en un archivo.

Sírvase leer todas las [restricciones y renuncias de responsabilidad](#).

[Regreso a la pantalla de contenido](#) [Principio](#)

Notificación de eventos: Guía del usuario de proveedores WMI y CDM para adaptadores de red Intel® PRO

[Clases de eventos concretos](#)
[Registro para eventos](#)



[Principio](#)

Clases de eventos concretos

IANet_802dot3AdapterEvent

Propósito

Este evento le notifica al cliente sobre un cambio en el estado o configuración de un adaptador.

Desencadenadores

El evento ocurre después de un cambio del estado de un adaptador o después de haber cambiado la configuración de un adaptador y se ha llamado a **Apply**.

Datos del evento

AdapterPath incluye la ruta al objeto del adaptador que causó el evento.

IANet_802dot3TeamEvent

Propósito

Este evento le notifica al cliente sobre un cambio en el estado o configuración de un equipo.

Desencadenadores

El evento ocurrirá:

- Después de un cambio en el estado de un adaptador.
- Después de cambiar la configuración de un equipo y se llama a **Aplicar**.
- Después de haberse cambiado la configuración del equipo y se llama a **Aplicar**.

Datos del evento

TeamPath incluye la ruta al objeto del equipo que causó el evento.

IANet_802dot3VlanEvent

Propósito

Este evento le notifica al cliente sobre un cambio en el estado o configuración de una VLAN.

Desencadenadores

El evento ocurrirá:

- Después de un cambio en el estado de una VLAN.
- Después de cambiar un valor de una VLAN y se llama a **Aplicar**.
- Después de haberse cambiado la configuración de la VLAN y se llama a **Aplicar**.

Datos del evento

VlanPath incluye la ruta al objeto del VLAN que causó el evento.

[Principio](#)

Registro para eventos

Las aplicaciones deberán usar **IWbemServices:: ExecNotificationQuery** o **IWbemServices:: ExecNotificationQueryAsync** para solicitar notificación de los eventos. Las consultas siguientes son algunos ejemplos de las consultas de notificación de eventos. Esta lista no está completa, ya que son posibles muchas consultas.

- **SELECT * FROM IANet_Event**: Solicitar todos los eventos.
 - **SELECT * FROM IANet_AdapterEvent**: Solicitar todos los eventos de adaptadores.
 - **SELECT * FROM IANet_TeamEvent**: Solicitar todos los eventos de equipos.
 - **SELECT * FROM IANet_SessionEvent**: Solicitar todos los eventos de sesiones.
 - **SELECT * FROM IANet_VlanEvent**: Solicitar todos los eventos VLAN.
 - **SELECT * FROM IANet_InternalErrorEvent**: Solicitar todos los eventos internos.
 - **SELECT * FROM IANet_AdapterEvent WHERE AdapterPath={ruta al objeto IANet_EthernetAdapter}**: Solicitar los eventos de adaptador de un adaptador en particular.
-

Sírvase leer todas las [restricciones y renunciaciones de responsabilidad](#).

[Regreso a la pantalla de contenido](#) [Principio](#)

Consultas WQL optimizadas: Guía del usuario de proveedores WMI y CDM para adaptadores de red Intel® PRO

[Visión general](#)

[Obtener la configuración para un adaptador, VLAN, o equipo en particular](#)

[Obtener una configuración](#)

Visión general

El Proveedor WMI está optimizado para permitir que las aplicaciones obtengan configuraciones usando consultas. El Proveedor WMI puede reconocer las consultas siguientes y devolverá solamente aquellos objetos que coincidan. Todas las otras consultas darán por resultado que el Proveedor WMI obtenga todas las configuraciones de todos los objetos y CIMOM las filtre antes de que alcancen la aplicación. Si existen varios adaptadores, equipos y VLAN se ocasionará un retraso de varios segundos antes de recuperar los datos requeridos.

[Principio](#)

Obtener la configuración para un adaptador, VLAN o equipo en particular

La consulta siguiente va a obtener la configuración para un adaptador, VLAN o equipo en particular únicamente. WQL no permite ninguna cláusula adicional en la cláusula WHERE.

```
ASSOCIATORS OF {ruta a IANet_Configuration} WHERE AssocClass = IANet_SettingContext
```

[Principio](#)

Obtener una configuración

La consulta siguiente se puede utilizar para obtener una configuración única para un objeto sin consultar para obtenerlas a todas.

```
SELECT * FROM [CLASE DE CONFIGURACIÓN] WHERE ParentId="[ID de dispositivo]" AND ParentType="[tipo]" AND Caption="[NOMBRE DE LA CONFIGURACIÓN]"
```

Notas:

- Esta clase deberá ser la clase de configuración exacta y no una clase de base (por ejemplo, IANet_SettingInt)
- Los ParentTypes aceptables son "NIC", "Team" , "VLAN" y "BootAgent".
- ParentID es la GUID que define de forma individual al objeto con la configuración (para un adaptador, sería DeviceId).
- Este no es un método recomendado para obtener la configuración asociada con un objeto; el método preferido es utilizar asociaciones. Sin embargo, WQL no es compatible con la consulta compleja necesaria (es decir, WQL no es compatible con ASSOCIATORS OF {ruta a IANet_Configuration} WHERE AssocClass = IANet_SettingContext AND Caption="[NOMBRE DE LA CONFIGURACIÓN]")

Sírvase leer todas las [restricciones y renunciaciones de responsabilidad](#).

[Regreso a la pantalla de contenido](#) [Principio](#)

Diagnósticos: Guía del usuario de proveedores WMI y CDM para adaptadores de red Intel® PRO

[Clases de diagnóstico](#)

[Entradas del registro](#)

[Registro](#)

[Clases asociativas](#)

[Pruebas](#)

Clases de diagnóstico

IANet_DiagTest

Propósito

IANet_DiagTest está incluida en una subclase de CIM_DiagnosticTest. La clase proporciona un vehículo genérico para ejecutar y controlar las pruebas de diagnóstico para un adaptador Ethernet compatible con Intel® PROSet. La superclase, CIM_DiagnosticTest, se ha concebido para aceptar, de manera genérica, el conjunto de pruebas de cualquier hardware de PC en un sistema CIM activado. Las propiedades de la clase son de naturaleza descriptiva y el mecanismo de la prueba lo proporcionan los métodos expuestos.

Instancias

La clave es un nombre y en este proveedor es la concatenación de un índice numérico de la prueba en el GUID del adaptador de referencia (por ejemplo, 1@{12345678-9ABC-DEF0-1234-123456789012}). Este valor de la clave es información redundante en un sentido, ya que toda la información para referirse a un adaptador y una prueba es pasada como parámetros de objeto del método RunTest y otros. A pesar de eso, la instancia debe ser consistente con los parámetros para el método o, de lo contrario, el proveedor podría rechazar el comando. La propiedad "caption" (título) proporciona el nombre de la prueba que se ejecutará en esta instancia. Otras propiedades proporcionan información de descripción y de tiempo de ejecución.

Creación de instancias

No se pueden crear instancias de IANet_DiagTest.

Eliminación de instancias

No se pueden eliminar instancias de IANet_DiagTest.

Modificación de propiedades

No hay propiedades que el usuario pueda modificar en esta clase.

Asociaciones

- Una instancia de IANet_DiagTestForMSE asocia un IANet_DiagTest con un IANet_ManagedSystemElement. El IANet_ManagedSystemElement será una instancia de IANet_EthernetAdapter.
- Una instancia de IANet_DiagResultForTest asocia un IANet_DiagTest con una instancia de IANet_DiagnosticResult.
- Una instancia de IANetDiagSettingForTest asocia una IANet_DiagTest con un IANet_DiagSetting..

Propiedades no compatibles

InstallDate, OtherCharacteristicDescription

Métodos

Esta clase acepta los siguientes métodos:

- RunTest: Ejecuta una prueba que es definida por tres parámetros, que se refieren a:
 - SystemElement: define al adaptador en el cual habrá que ejecutar la prueba refiriéndose a una instancia de SystemElement, que siempre será la subclase IANet_EthernetAdapter.
 - Setting: define la prueba que ha de ejecutarse y la manera como se ejecuta refiriéndose a una instancia de CIM_DiagnosticSetting, que siempre será la subclase IANet_DiagSetting.

- DiagnosticResult: define una instancia de la clase CIM_DiagnosticResult, que siempre será la clase IANet_DiagResult.
- DiscontinueTest: Intenta detener una prueba de diagnóstico que está en ejecución definida por dos parámetros que hacen referencia a CIM_ManagedSystemElement y CIM_DiagnosticResult. Estos parámetros funcionan igual que RunTest. Un tercer parámetro, TestingStopped devuelve un valor BOOLEANO, que indica si es que el comando logró detener la prueba.
- ClearResults: Borra los resultados de una prueba usando los parámetros:
 - SystemElement
 - ResultsNotCleared

El parámetro referido ManagedSystemElement, combinado con la ruta de objeto de este objeto se combina para referirse a instancias de DiagnosticResultForMSE, misma que será eliminada. También, todas las referencias a objetos DiagnosticResult referidas por DiagnosticResultForMSE serán eliminadas. También, todas las instancias de DiagnosticResultForTest referidas a los objetos DiagnosticResult eliminados, serán eliminadas. Finalmente, el parámetro de matriz de cadenas ResultsNotCleared de Output va a enumerar las claves de DiagnosticResults, mismas que no pudieron eliminarse.

Jerarquía de clases

Para CimV2. No se muestran las propiedades y métodos que no se utilizan.

- CIM_ManagedElement:
 - Caption
 - Description
- CIM_ManagedSystemElement:
 - Install Date
 - Name
 - Status
- CIM_LogicalElement
- CIM_Service:
 - Clave
 - Name (cadena)
 - Propiedades
 - Caption (cadena)
 - CreationClassName (cadena)
 - Description (cadena)
 - Started (booleano)
 - StartMode (cadena)
 - Status (cadena)
 - SystemCreationClass (cadena)
 - SystemName (cadena)
- CIM_DiagnosticTest:
 - Propiedades
 - Characteristics (matriz uint16)
 - IsInUse (Booleano)
 - ResourcesUsed (matriz uint16)
 - Métodos
 - RunTest
 - ClearResults
 - DiscontinueTest

La ejecución de RunTest y otros métodos en WbemTest

El método RunTest del archivo MOF es como sigue:

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

Los dos primeros parámetros son parámetros de entrada. Debe obtener la ruta de objeto de ambos objetos referenciados. También debe obtener la ruta al objeto IANet_DiagTest, que está exportando el objeto RunTest.

1. En el cuadro de diálogo principal de la prueba WBEM, haga clic en **Connect** (Conectar).
2. Escriba el servidor o el espacio de nombre apropiado. Se aceptan los espacios de nombre IntelINCS y CimV2.
3. Haga clic en el botón **Enum Instances** (Enumerar instancias) de la prueba WBEM y escriba `IANet_DiagTest`.
4. Haga doble clic en la instancia `IANet_DiagTest`. El nombre aparecerá en la forma de `X@[GUID_del_adaptador]`, donde X es el nombre de la prueba y GUID_del_adaptador es el nombre del adaptador, igual que la clave Name de `IANet_EthernetAdapter`.
5. A continuación aparece un ejemplo de la ruta de objeto recuperada:
`\\MYCOMPUTER\root\Cimv2:IANet_DiagTest.Name="1@{4A0CDABE-F6C3-45D0-B60D-F6E7BAFA2C2C}"`
6. Guarde la ruta del objeto.
7. Haga clic en el botón **Enum Instances** de la prueba WBEM y escriba `IANet_EthernetAdapter`.
8. Haga doble clic en el adaptador que se va a probar.
9. A continuación aparece un ejemplo de la ruta de objeto recuperada:
`\\MYCOMPUTER\root\cimv2:IANet_EthernetAdapter.DeviceID="{4A0CDABE-F6C3-45D0-B60D-F6E7BAFA2C2C}"`
10. Guarde la ruta del objeto.
11. Haga clic en el botón **Enum Instances** de la prueba WBEM y escriba `IANet_DiagSetting`.
12. Haga doble clic en el valor que representa la combinación de adaptador y prueba.
13. A continuación aparece un ejemplo de la ruta de objeto recuperada:
`\\MYCOMPUTER\root\cimv2:IANet_DiagSetting.SettingID="1@{4A0CDABE-F6C3-45D0-B60D-F6E7BAFA2C2C}"`
14. Guarde la ruta del objeto.
15. En el cuadro de diálogo principal de la prueba WBEM, haga clic en **Execute Method** (Ejecutar método).
16. Pegue la ruta del objeto `IANet_DiagTest` en el cuadro de diálogo. Haga clic en **Aceptar**.
17. Debajo de method en el cuadro de lista desplegable, seleccione la prueba.
18. Haga clic en el botón **Edit In Parameters** (Editar los parámetros de entrada).
19. Para RunTest, los parámetros de entrada son Setting y SystemElement Pegue las rutas de los objetos Settings y Adapter guardadas previamente y cierre.
20. Haga clic en el botón **Execute** (Ejecutar).
21. Enumere la clase `IANet_DiagResult`, de la misma forma que los parámetros de entrada.
22. Examine el objeto resultante seleccionado según sea necesario.

IANet_DiagSetting

Propósito

Las instancias de `IANet_DiagSetting` proporcionan directrices de pruebas de diagnóstico en tiempo de ejecución específicas. Las directrices utilizadas son comunes para todas las pruebas y están ligadas a la superclase `CIM_DiagnosticSetting`. Incluyen propiedades, como serían `ReportSoftErrors` y `HaltOnError`. No hay propiedades adicionales agregadas a la subclase `IANet_DiagSetting`.

Creación de instancias

No se pueden crear instancias de esta clase.

Eliminación de instancias

No se pueden eliminar instancias de esta clase.

Modificación de propiedades

Se implementa `UpdateInstanceAsync` y puede usarse para fijar parámetros de prueba de "Halt On Error", "Report Soft Errors", "Report Status Messages", "Quick Mode", "Test Warning Level" y "Percent Of Test Coverage".

Asociaciones

Una instancia de `IANetDiagSettingForTest` asocia una `IANet_DiagTest` con una `IANet_DiagSetting`.

Propiedades no compatibles

Las propiedades siguientes no son aceptadas por NCS:

- Caption
- Description

Métodos

Ninguno

Jerarquía de clases

Para CimV2. No se muestran las propiedades y los métodos que no se utilizan.

- CIM_ManagedElement:
- CIM_Setting:
 - Propiedades
 - SettingID
 - Methods (ninguno es aceptado)
 - VerifyOKToApplyToMSE
 - ApplyToMSE
 - VerifyOKToApplyToCollection
 - ApplyToCollection
 - VerifyOKToApplyIncrementalChangeToMSE
 - ApplyIncrementalChangesToMSE
 - ApplyIncrementalChangeToCollection
- CIM_DiagnosticSetting:
 - Clave
 - SettingID (cadena)
 - Propiedades
 - TestWarningLevel (uint16)
 - ReportSoftErrors (Booleano)
 - ReportStatusMessages (Booleano)
 - HaltOnError (Booleano)
 - QuickMode (Booleano)
 - PercentOfTestCoverage (uint8)

IANet_DiagResult

Propósito

Las instancias de IANet_DiagResult muestran los datos resultantes de una prueba en particular que se ejecutó en un adaptador específico. Las instancias de esta clase son idénticas a las instancias de IANet_DiagTest y IANet_DiagSetting.

Instancias

Las instancias de IANet_DiagResult corresponden a los resultados de una prueba en particular que se ejecutó en un adaptador específico. El formato para la clave es el mismo que IANet_DiagTest y IANet_DiagSetting. La instancia puede guardar resultados arbitrarios de pruebas como cualquier tipo de datos, que no se ajusta a las propiedades definidas, puede ubicarse en la propiedad TestResults Array. En cualquier momento que se ejecuta una prueba de un adaptador, la instancia nueva sobrescribe la instancia de resultados de prueba existentes correspondientes a la combinación de ese adaptador y esa prueba.

Creación de instancias

No se pueden crear instancias de esta clase.

Eliminación de instancias

No se pueden eliminar instancias de esta clase.

Modificación de propiedades

No se pueden modificar instancias de esta clase.

Asociaciones

Una instancia de IANet_DiagResultForTest asocia un IANet_DiagTest con una instancia de IANet_DiagnosticResult.

Propiedades no compatibles

Las propiedades siguientes no son aceptadas por NCS:

- EstimatedTimeOfPerforming
- HaltOnError
- OtherStateDescription
- ReportSoftErrors
- TestWarningLevel

Métodos

Ninguno

Jerarquía de clases

Para CimV2. No se muestran las propiedades y los métodos que no se utilizan.

- CIM_DiagnosticResult:
 - Claves
 - DiagnosticCreationClassName (cadena)
 - DiagnosticName (cadena)
 - ExecutionID (cadena)
 - DiagnosticCreationClassName (cadena)
 - DiagSystemName (cadena)
 - Propiedades
 - TimeStamp (cadena)
 - IsPackage (Booleano)
 - TestStartTime (fecha y hora)
 - TestCompletionTime (fecha y hora)
 - TestState (uint16)
 - TestResults (cadena)
 - PercentComplete (uint8)
- IANet_DiagResult

[Principio](#)

Entradas del registro

Las entradas siguientes se escriben en el registro durante la instalación bajo

HKLM\Software\Intel\NETWORK_SERVICES\NCS\NcsDiag. Estas claves y valores controlan la ejecución de las pruebas de diagnóstico y se definen a continuación.

La tabla siguiente incluye los valores de la clave, su tipo y una explicación breve de su uso.

Valor	Tipo	Valor predeterminado	Uso
Check Time	REG_DWORD	2 segundos	La duración del lapso de tiempo entre las verificaciones para la conclusión de la prueba del remitente o destinatario
Habilitar	REG_DWORD	0	Habilitar (1) o Inhabilitar (0) uso de un archivo de registro de resultados.
FileAppend	REG_DWORD	1	Agregar (1) el archivo de registro de resultados al archivo de registro de resultados existente. Si es 0, elimina el archivo existente.
LogFileName	REG_SZ	NcsDiag.log	El nombre del archivo de registro de resultados.
MaxFileSize	REG_DWORD	0x10000	El tamaño máximo del archivo de registro de resultados.
MaxPktsRcvd	REG_DWORD	200	En Quick Mode (Modo rápido) (de IANet_DiagSetting), la prueba enviar / recibir será terminada en cuanto el número de paquetes recibidos es mayor que este valor.
TimeoutSndRsp	REG_DWORD	100	La prueba se termina en cuanto la duración de la prueba (en segundos) sobrepasa este valor.

[Principio](#)

Registro

Registro de resultados

El registro de resultados muestra, principalmente, información que podía haberse obtenido de objetos IANet_DiagResult. La diferencia consiste en que esta información que se obtuvo de un navegador CIM, mostrará solamente los últimos resultados inmediatamente anteriores de pruebas particulares en un adaptador en particular. Las pruebas subsecuentes sobrescriben los resultados de pruebas anteriores. El registro de resultados es más conveniente para configurar y examinar varias ejecuciones de tal prueba.

Habilitación del registro de resultados

Para habilitar el registro de resultados:

1. En la clave del registro **HKLM\Software\Intel\NETWORK_SERVICES\NCS\NCSDiag**, fije el valor de habilitación en 1.
2. Establezca el valor LogFileName en el nombre preferido del archivo de registro, o deje el predeterminado **NcsDiag.log**.
3. El archivo de registro puede encontrarse en el directorio indicado por el valor de InstalledDir.

[Principio](#)

Clases asociativas

Clases asociativas	Propiedad o valor de referencia	Propiedad o valor de referencia
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

[Principio](#)

Pruebas

Las pruebas implementadas pueden ejecutarse en una sola máquina o en dos. La descripción detallada de las pruebas van más allá del ámbito de este documento, ya que el proveedor CDM se ha concebido como un vehículo de pruebas genérico, que no depende de los detalles de las pruebas. Sin embargo, existe cierta dependencia construida en el código y que se explica en esta sección.

Pruebas de un solo adaptador

Las pruebas siguientes se ejecutan en un solo adaptador y no requieren de interacción con otros adaptadores.

- EEPROM
- Registros de control
- Bucles cerrados MAC
- Bucles cerrados PHY
- Enlace

Todos los mensajes de error de estas pruebas dan como resultado códigos de error HRESULT devueltos de llamadas a las capas inferiores de la pila. Los códigos de error se guardan internamente como códigos de error y se traducen como mensajes de error hasta que se elimina la referencia de los objetos IANet_DiagResult por enumeración o por llamadas de objeto recibidas de una aplicación de gestión.

Pruebas que requieren de dos adaptadores

Las pruebas de remitente y destinatario son pruebas codependientes en las cuales uno de los adaptadores (el remitente) envía paquetes a otro adaptador (el destinatario), el cual envía paquetes de regreso al remitente, completando así el bucle. Estas son las mismas pruebas que pueden ejecutarse desde Intel® PROSet. Sin embargo, Intel PROSet no utiliza CDM y no le permite

ejecutar dos pruebas en la misma máquina a la vez. CDM permitirá que se ejecuten pruebas concurrentemente en la misma máquina.

Prueba de remitente y destinatario

La prueba de remitente y destinatario requiere de dos adaptadores Intel, uno para el remitente y el otro para el destinatario. Esta prueba es la única en la cual la prueba se ejecuta basada en un segundo subproceso que continúa en ejecución hasta que se termina la prueba de acuerdo con los criterios de terminación o es detenida por un subproceso primario. Los criterios de terminación son tiempos de espera que están basados en la duración del tiempo de la prueba o el número de paquetes recibidos. Ambos valores se pueden obtener del registro. La prueba podrá terminar solamente sobre la base del número de paquetes recibidos si es que se habilitó el Quick Mode. Quick Mode es una propiedad de la clase IANet_DiagSetting y, en consecuencia, podría establecerse sobre la base de cada adaptador. Los destinatarios CDM le contestan a los destinatarios PROSet y viceversa.

Se devuelven dos tipos de valores de error de la prueba de remitente y destinatario. Primero, puede devolverse un código de error (HRESULT) de las capas inferiores. En segundo lugar, mientras se ejecuta la prueba y, salvo que la prueba se termine prematuramente por un código de error devuelto, el subproceso de prueba devolverá estadísticas de la prueba intermedias y después las finales, las cuales incluyen lo siguiente:

- Estado de enlace
- Uso de negociación automática
- Colisiones
- Paquetes recibidos
- Total de paquetes recibidos
- Paquetes enviados
- Transmisiones correctas
- Recepciones correctas
- Errores de transmisión
- Errores de recepción
- Colisiones
- Fase de diagnóstico

Sírvase leer todas las [restricciones y renunciaciones de responsabilidad](#).

[Regreso a la pantalla de contenido](#) [Principio](#)

Ejecución de métodos en IANet_DiagTest: Guía del usuario de proveedores WMI y CDM para adaptadores de red Intel® PRO

Ejecución de RunTest y otros métodos en WbemTest

El método RunTest del archivo MOF es como sigue:

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

Los dos primeros parámetros son parámetros de entrada. Debe obtener las rutas a ambos objetos referenciados. También debe obtener la ruta al objeto IANet_DiagTest, que está exportando el objeto RunTest.

1. En el cuadro de diálogo principal de la prueba WBEM, haga clic en **Connect** (Conectar).
2. Escriba el servidor o el espacio de nombre apropiado. Se aceptan los espacios de nombre IntelNCS y CimV2.
3. Haga clic en el botón **Enum Instances** (Enumerar instancias) de la prueba WBEM y escriba IANet_DiagTest.
4. Haga doble clic en la instancia IANet_DiagTest. El nombre aparecerá en la forma de X@[GUID_del_adaptador], donde X es el nombre de la prueba y GUID_del_adaptador es el nombre del adaptador, igual que la clave Name de IANet_EthernetAdapter.
5. A continuación aparece un ejemplo de la ruta de objeto recuperada:
\\MIPC\root\Cimv2:IANet_DiagTest.Name="1@{4A0CDABE-F6C3-45D0-B60D-F6E7BAFA2C2C}"
6. Guarde la ruta del objeto.
7. Haga clic en el botón **Enum Instances** de la prueba WBEM y escriba IANet_EthernetAdapter.
8. Haga doble clic en el adaptador que se va a probar.
9. A continuación aparece un ejemplo de la ruta de objeto recuperada:
\\MIPC\root\cimv2:IANet_EthernetAdapter.DeviceID="{4A0CDABE-F6C3-45D0-B60D-F6E7BAFA2C2C}"
10. Guarde la ruta del objeto.
11. Haga clic en el botón **Enum Instances** de la prueba WBEM y escriba IANet_DiagSetting.
12. Haga doble clic en el valor que representa la combinación de adaptador y prueba.
13. A continuación aparece un ejemplo de la ruta de objeto recuperada:
\\MIPC\root\cimv2:IANet_DiagSetting.SettingID="1@{4A0CDABE-F6C3-45D0-B60D-F6E7BAFA2C2C}"
14. Guarde la ruta del objeto.
15. En el cuadro de diálogo principal de la prueba WBEM, haga clic en **Execute Method** (Ejecutar método).
16. Pegue la ruta del objeto IANet_DiagTest en el cuadro de diálogo. Haga clic en **Acceptar**.
17. Debajo de method en el cuadro de lista desplegable, seleccione la prueba.
18. Haga clic en el botón **Edit In Parameters** (Editar los parámetros de entrada).
19. Para RunTest, los parámetros de entrada son Setting y SystemElement Pegue las rutas de los objetos Settings y Adapter guardadas previamente y cierre.
20. Haga clic en el botón **Execute** (Ejecutar).
21. Enumere la clase IANet_DiagResult, de la misma forma que los parámetros de entrada.
22. Examine el objeto resultante seleccionado según sea necesario.

Sírvase leer todas las [restricciones y renunciaciones de responsabilidad](#).

Resumen de clases CIM: Guía del usuario de proveedores WMI y CDM para adaptadores de red Intel® PRO

Clase	¿Puede crear?	¿Puede eliminar?	Métodos implementados	Propiedades configurables	Propiedades no compatibles	Conteo de instancias	Clases de asociación relacionadas
IANet_802dot1QVLANService	N	N	CreateVLAN	GRVPEEnabled, JoinTime	Description, Install Date, Started, StartMode, Status	Una instancia para cada equipo o adaptador que es compatible con VLAN	IANet_Device802dot1QVLANServiceImplementation, IANet_VLANFor
IANet_AdapterConfiguration	N	N	Ninguno	Ninguna	Ninguna	Una instancia para cada adaptador	Esta clase asocia IANet_EthernetAdapter con IANet_Configuration.
IANet_AdapterDevice	N	N	Ninguno	Ninguna	Ninguna	Una instancia para cada adaptador no fantasma	Esta clase asocia IANet_EthernetAdapter con IANet_EthernetPCIDevice.
IANet_AdapterProtocolImplementation	N	N	Ninguno	Ninguna	Ninguna	Una instancia para cada extremo de protocolo IP que está ligado a un adaptador	Esta clase asocia IANet_EthernetAdapter con IANet_IPProtocolEndpoint.
IANet_BootAgent	N	N	ProgramFlash ReadFlash	Ninguna	Caption, Description, InstallDate, Started, StartMode, Status	Una instancia para cada adaptador compatible con la capacidad de agente de inicio	IANet_DeviceBootServiceImplementation, IANet_BootAgentConfiguration.
IANet_BootAgentConfiguration	N	N	Ninguno	Ninguna	Ninguna	Una instancia para cada agente de inicio.	Esta clase asocia IANet_BootAgent con IANet_Configuration.
IANet_Configuration	N	N	Ninguno	Ninguna	Ninguna	Una instancia para cada adaptador, VLAN y equipo	IANet_AdapterConfiguration, IANet_VLANConfiguration, IANet_SettingContext.
IANet_Device802dot1QVLANServiceImplementation	N	N	Ninguno	Ninguna	Ninguna	Una instancia para cada equipo o adaptador que es compatible con VLAN	Esta clase asocia IANet_EthernetAdapter con IANet_802dot1QVLANService.
IANet_DiagTest	N	N	RunTest, DiscontinueTest, ClearResults	Ninguna	InstallDate, OtherCharacteristicsDescription	Uno para cada combinación de adaptador y prueba	IANet_DiagTestForMSE, IANet_DiagResultForTest, IANet_DiagSettingForTest

IA net_DiagSetting	N	N	Ninguno	HaltOnError, ReportSoftErrors, ReportStatusMessages, QuickMode, PercentOfTestCoverage, TestWarningLevel	Caption, Description	Uno para cada combinación de adaptador y prueba	IA net_DiagSettingForTest
IA net_DiagResult	N	N	Ninguno	Ninguna	EstimatedTimeOfPerforming, HaltOnError, OtherStateDescription, ReportSoftErrors, TestWarningLevel	Uno para cada combinación de adaptador y prueba	IA net_DiagResultForTest, IA net_DiagResultForMSEM
IA net_EthernetAdapter	N	Y	IdentifyAdapter HasVLANs IsPowerMgmtSupported GetPowerUsage SetPowerUsage GetPowerUsageOptions SetPowerUsageOptions TestCable AdvancedTestCable TestLinkSpeed	Ninguna	AutoSense – (se expone como una configuración), ErrorCleared, OtherIdentifyingInfo, IdentifyingDescriptions, InstallDate, LastErrorCode, MaxDataSize, MaxQuiesceTime, PowerManagementCapabilities– (se expone como un método), PowerManagementSupported– (se expone como un método), PowerOnHours , ShortFramesReceived, SymbolErrors, TotalPowerOnHours	Uno para cada adaptador compatible con Intel®PROSet instalado, adaptador fantasma y equipo	IA net_AdapterProtocolImplementation, IA net_AdapterDevice, IA net_AdapterConfiguration, IA net_TeamedMemberAdapter, IA net_NetworkVirtualAdapter, IA net_Device802dot1QVLANServiceImplementation, IA net_DeviceBootServiceImplementation
IA net_EthernetPCIDevice	N	N	Ninguno	Ninguna	AdditionalAvailability, Capabilities, CapabilityDescriptions, Caption, Description, DeviceSelectTiming, ErrorCleared, ErrorDescription, IdentifyingDescription, InstallDate, LastErrorCode, MaxNumberController, MaxQuiesceTime, Name, OtherIdentifyingInfo, PowerManagementCapabilities, PowerManagementSupported, PowerOnHours, ProtocolDescription, ProtocolSupported, SelfTestEnabled, Status, StatusInfo, TimeOfLastReset, TotalPowerOnHours	Una instancia para cada tarjeta PCI que es un adaptador Ethernet compatible con Intel PROSet	IA net_AdapterDevice
IA net_IPProtocolEndpoint	N	N	Ninguno	Ninguna	Caption, Description, InstallDate, NameFormat, OtherTypeInfoInformation, Status	Una instancia para cada liga de la pila del Protocolo IP a un extremo compatible con Intel	IA net_AdapterProtocolImplementation, IA net_VLANProtocolDependency
IA net_NetService	N	N	GetSessionHandle, Apply, ReleaseSessionHandle, Cancel	Ninguna	Caption, Description, Install Date, Started, Start Mode, Status	Exactamente uno.	Ninguno
IA net_NetworkVirtualAdapter	N	N	Ninguno	Ninguna	Ninguna	Una instancia para cada equipo.	Esta clase asocia IA net_TeamOfAdapters con un IA net_EthernetAdapter.
IA net_PCIDevice	N	N	Ninguno	Ninguna	AdditionalAvailability,	Una instancia	65

					Capabilities, CapabilityDescriptions, Caption, DeviceSelectTiming, ErrorCleared, ErrorDescription, IdentifyingDescription, InstallDate, LastErrorCode, MaxNumberController, MaxQuiesceTime, Name, OtherIdentifyingInfo, PowerManagementCapabilities, PowerManagementSupported, PowerOnHours, ProtocolDescription, ProtocolSupported, SelfTestEnabled, TimeOfLastReset, TotalPowerOnHours	para cada tarjeta PCI que es un dispositivo de red en el sistema	
IANet_Setting	N	N	Ninguno	Ninguna	SettingID	Esta es una clase abstracta.	IANet_SettingContext
IANet_SettingContext	N	N	Ninguno	Ninguna	Ninguna	Una instancia para cada configuración	Esta clase asocia IANet_EthernetAdapter con una IANet_Configuration.
IANet_SettingInt	N	N	Ninguno	CurrentValue	SettingID	Una instancia para cada configuración de número entero	IANet_SettingContext
IANet_SettingMultiSelection	N	N	Ninguno	CurrentValue	SettingID	Una instancia para cada configuración de selección múltiple	IANet_SettingContext
IANet_SettingSlider	N	N	Ninguno	CurrentValue	SettingID	Una instancia para cada configuración de deslizador	IANet_SettingContext
IANet_SettingString	N	N	Ninguno	CurrentValue	SettingID	Una instancia para cada configuración de cadena	IANet_SettingContext
IANet_TeamedMemberAdapter	Y	Y	Ninguno	AdapterFunction	PrimaryAdapter, ScopeOfBalancing	Una instancia para cada adaptador que es miembro de un equipo	Esta clase asocia IANet_TeamOfAdapters con un IANet_EthernetAdapter.
IANet_TeamOfAdapters	Y	Y	TestSwitchConfiguration	TeamingMode	Install Date, Status	Una instancia para cada equipo.	IANet_NetworkVirtualAdapter, IANet_TeamedMemberAdapter
IANet_VLAN	N	Y	Ninguno	VLANNumber, Caption	Description, Install Date, StartMode, Status	Una instancia para cada VLAN	IANet_VLANFor
IANet_VLANConfiguration	N	N	Ninguno	Ninguna	Ninguna	Una instancia para cada VLAN	Esta clase asocia IANet_VLAN con IANet_Configuration.
IANet_VLANFor	N	N	Ninguno	Ninguna	Ninguna	Una instancia para cada VLAN	Esta clase asocia IANet_VLAN con IANet_802dot1QVLANService.

IANet_VLANProtocolDependency	N	N	Ninguno	Ninguna	Ninguna	Una instancia para cada VLAN	Esta clase asocia IANet_VLAN con IANet_IPProtocolEndpoint.
------------------------------	---	---	---------	---------	---------	------------------------------	--

Sírvase leer todas las [restricciones y renunciaciones de responsabilidad](#).

[Regreso a la pantalla de contenido](#) [Principio](#)

Contrato de licencia del software de Intel (final, licencia): Guía del usuario de proveedores WMI y CDM para adaptadores de red Intel® PRO

IMPORTANTE: LÉASE ANTES DE COPIAR, INSTALAR O UTILIZAR

No utilice ni cargue este Software ni cualquier material asociado (denominado conjuntamente "el Software") hasta que haya leído con detenimiento los siguientes términos y condiciones. Al cargar o usar el Software, conviene en observar los términos de este Contrato. Si no desea aceptarlos, no instale ni utilice el Software.

LICENCIAS:

- Si es usted un administrador de redes, le corresponde la sección "Licencia de sitio".
- Si es un usuario final, le corresponde la sección "Licencia de usuario individual".
- Si es usted un fabricante de equipo original (OEM), le corresponde la sección "Licencia de OEM".

Licencia de sitio. Podrá copiar el software en los sistemas de su organización para el uso de ésta, y podrá efectuar un número prudente de copias de seguridad del software, sujeto a las siguientes condiciones:

1. Se autoriza el uso de este Software únicamente en conjunto con los productos de componentes Intel. Este documento no otorga licencia para el uso del software con productos de componentes ajenos a Intel.
2. No puede copiar, modificar, alquilar, vender, distribuir ni transferir parte alguna del Software, excepto según se especifique en este Contrato, y se aviene a impedir la copia no autorizada del Software.
3. No puede aplicar técnicas de ingeniería inversa, descompilar ni desensamblar el Software.
4. No puede otorgar licencia de uso del Software ni permitir que más de un usuario lo utilice simultáneamente.
5. El software podrá incluir porciones que se ofrecen en términos agregados a los estipulados en el presente contrato, según se expresan en la licencia que acompaña a esas porciones.

Licencia de usuario individual. Puede copiar el Software en un único sistema para su uso personal, no comercial, y realizar una copia de seguridad del Software con estas condiciones:

1. Se autoriza el uso de este Software únicamente en conjunto con los productos de componentes Intel. Este documento no otorga licencia para el uso del software con productos de componentes ajenos a Intel.
2. No puede copiar, modificar, alquilar, vender, distribuir ni transferir parte alguna del Software, excepto según se especifique en este Contrato, y se aviene a impedir la copia no autorizada del Software.
3. No puede aplicar técnicas de ingeniería inversa, descompilar ni desensamblar el Software.
4. No puede otorgar licencia de uso del Software ni permitir que más de un usuario lo utilice simultáneamente.
5. El software podrá incluir porciones que se ofrecen en términos agregados a los estipulados en el presente contrato, según se expresan en la licencia que acompaña a esas porciones.

Licencia de OEM. Usted puede copiar y distribuir el Software únicamente como una parte integral o incorporada en su Producto o como una actualización autónoma de mantenimiento del Software solamente para usuarios finales existentes de Sus productos, a excepción de cualquier otro producto autónomo, y sujeto a las condiciones siguientes:

1. Se autoriza el uso de este Software únicamente en conjunto con los productos de componentes Intel. Este documento no otorga licencia para el uso del software con productos de componentes ajenos a Intel.
2. No puede copiar, modificar, alquilar, vender, distribuir ni transferir parte alguna del Software, excepto según se especifique en este Contrato, y se aviene a impedir la copia no autorizada del Software.
3. No puede aplicar técnicas de ingeniería inversa, descompilar ni desensamblar el Software.
4. Solamente podrá distribuir el Software a sus clientes, conforme a un contrato de licencia por escrito. Ese contrato de licencia puede tener el carácter de contrato de licencia por "ruptura del sello". Cuando menos, esa licencia va a salvaguardar los derechos de propiedad sobre el Software de Intel.
5. El software podrá incluir porciones que se ofrecen en términos agregados a los estipulados en el presente contrato, según se expresan en la licencia que acompaña a esas porciones.

NO EXISTEN OTROS DERECHOS. Intel no le concede otros derechos o licencias, de manera expresa o implícita, con respecto a

cualquier información de titularidad o patente, copyright, combinación, marca comercial, secretos comerciales, u otros derechos de titularidad de propiedad intelectual o controlados por Intel, a excepción de lo que se establece expresamente en este Contrato.

PROPIEDAD DEL SOFTWARE Y DERECHOS DE AUTOR. Intel o sus proveedores mantienen el título de propiedad de todas las copias del Software. El Software dispone de sus correspondientes derechos de autor, y está amparado por la legislación estadounidense y de otros países, así como por disposiciones de tratados internacionales. No está permitido eliminar del Software ninguna nota sobre los derechos de autor. Intel puede introducir modificaciones en el Software, o en elementos a los que se hace referencia en el mismo, en cualquier momento y sin previo aviso, sin tener obligación de prestar asistencia técnica o actualizar el Software. Excepto si se indica lo contrario, Intel no concede derechos explícitos o implícitos sobre las patentes, derechos de autor, marcas registradas u otros derechos de la propiedad intelectual de Intel. Sólo podrá transferir el Software si el receptor acepta por completo estos términos y usted no retiene copia alguna del mismo.

GARANTÍA LIMITADA DEL MEDIO. Si Intel ha entregado el Software en un soporte físico, Intel garantiza que dicho soporte está libre de defectos físicos de material durante los noventa días después de la entrega. Si encuentra algún defecto semejante, devuelva el soporte a Intel para reemplazarlo o para que Intel le entregue el Software de otro modo, a discreción de Intel.

EXCLUSIÓN DE OTRAS GARANTÍAS. EXCEPTO LO YA DISPUESTO, SE PROPORCIONA EL SOFTWARE EN EL ESTADO QUE SE ENCUENTRA Y SIN GARANTÍAS EXPRESAS O IMPLÍCITAS DE NINGÚN TIPO, ENTRE OTRAS, GARANTÍAS DE COMERCIALIZACIÓN, AUSENCIA DE VIOLACIONES O IDONEIDAD PARA UN FIN DETERMINADO. Intel no garantiza ni asume responsabilidad en cuanto a la exactitud o integridad de cualquier información, texto, gráficos, enlaces u otros elementos contenidos en el Software.

LIMITACIÓN DE RESPONSABILIDAD. INTEL Y SUS PROVEEDORES NO SERÁN RESPONSABLES EN NINGÚN CASO DE NINGUNA CLASE DE DAÑO (INCLUYENDO, PERO SIN LIMITARSE A ELLO, LA PÉRDIDA DE BENEFICIOS, LA INTERRUPTIÓN DE LA ACTIVIDAD COMERCIAL O LA PÉRDIDA DE INFORMACIÓN) DERIVADO DE LA UTILIZACIÓN O DE LA IMPOSIBILIDAD DE UTILIZAR EL SOFTWARE, INCLUSO EN EL CASO DE QUE INTEL HAYA TENIDO CONOCIMIENTO DE LA POSIBILIDAD DE DICHO DAÑO. EN CIERTAS JURISDICCIONES SE PROHÍBE LA EXCLUSIÓN O LIMITACIÓN DE RESPONSABILIDAD RESPECTO A GARANTÍAS IMPLÍCITAS O DAÑOS INDIRECTOS O CASUALES, POR LO QUE LAS LIMITACIONES MENCIONADAS PUEDEN NO SER APLICABLES EN SU CASO. ASIMISMO, ES POSIBLE QUE TENGA OTROS DERECHOS LEGALES QUE VARIÉN DE UNA JURISDICCIÓN A OTRA.

RESCISIÓN DE ESTE CONTRATO. Si viola los términos de este Contrato, Intel puede rescindirlo en cualquier momento. Tras la rescisión, usted deberá destruir inmediatamente el Software o devolver a Intel todas las copias del mismo.

LEGISLACIÓN APLICABLE. Cualquier litigio o reclamación que se derive de este contrato se regirá por la legislación del Estado de California, con exclusión de sus principios de conflicto de leyes y la Convención de las Naciones Unidas sobre la venta internacional de mercaderías. Se prohíbe exportar el Software si dicho acto violara cualquier ley o reglamento aplicable sobre exportaciones. Intel no está obligada por ningún otro contrato, a menos que éste se haya realizado por escrito y esté firmado por un representante autorizado de Intel.

DERECHOS RESTRINGIDOS DEL ESTADO. Este Software se suministra sujeto a "DERECHOS RESTRINGIDOS". Su uso, reproducción o divulgación por el Gobierno está sujeto a las restricciones establecidas en FAR52.227-14 y DFAR252.227-7013 y siguientes, o sus sucesoras. El uso del Software por parte del Gobierno constituye una aceptación de los derechos de propiedad de Intel sobre el Software. El contratista o fabricante es Intel.

Sírvase leer todas las [restricciones y renunciaciones de responsabilidad](#).

[Regreso a la pantalla de contenido](#) [Principio](#)

[Regreso a la pantalla de contenido](#)

Asistencia técnica: Guía del usuario de proveedores WMI y CDM para adaptadores de red Intel® PRO

Sitios Web e Internet

<http://www.dell.com>

Técnicos de asistencia al cliente

Si los procedimientos de resolución de problemas de este documento no solucionan el problema, póngase en contacto con Dell Computer Corporation para recibir asistencia técnica (consulte la sección "Obtención de ayuda" en la documentación del sistema).

Antes de llamar...

Debe estar delante de su equipo, con el software en ejecución y la documentación del producto a la mano.

Es posible que el técnico le pregunte lo siguiente:

- Su dirección y número de teléfono
 - El nombre y el número de modelo del producto sobre el que está llamando
 - El número de serie y la etiqueta de servicio del producto
 - Los nombres y los números de versión del software que está usando para trabajar con el producto
 - El nombre y número de versión del sistema operativo que está utilizando
 - El tipo de sistema (fabricante y número de modelo)
 - Las tarjetas de expansión o adicionales del sistema
 - La cantidad de memoria de su sistema
-

Sírvase leer todas las [restricciones y renunciaciones de responsabilidad](#).

[Regreso a la pantalla de contenido](#)