

# Benutzerhandbuch für Intel® PRO Netzwerkadapter – WMI und CDM Provider

---

Die nachfolgenden Informationen wurden vom Zulieferer des jeweils erwähnten Gerätes ohne unabhängige Prüfung durch Dell zur Verfügung gestellt und unterliegen unten aufgeführten [Einschränkungen und Beschränkungen](#).

---

- [Einführung](#)
  - [WMI](#)
  - [Hauptfunktionen](#)
  - [Installierte Dateien](#)
  - [Sicherheit](#)
  - [Namespace und Kontext](#)
  - [Standorte und Lokalisierung](#)
  - [Fehlermeldungen](#)
  - [Das Kernschema](#)
  - [Ethernet-Adapterschema](#)
  - [Einstellungsschema](#)
  - [Gruppenschema](#)
  - [VLAN-Schema](#)
  - [Ermitteln der aktuellen Konfiguration](#)
  - [Aktualisieren der Konfiguration](#)
  - [Ereignisbenachrichtigungen](#)
  - [Optimierte WQL-Abfragen](#)
  - [Diagnose](#)
  - [Ausführen von Methoden in IANet\\_DiagTest](#)
  - [Zusammenfassung der CIM-Klassen](#)
  - [Softwarelizenz](#)
  - [Kundensupport](#)
- 

**Änderungen der in diesem Handbuch enthaltenen Informationen vorbehalten.**

**© 2003 Intel Corporation. Alle Rechte vorbehalten.**

In diesem Text werden folgende Marken verwendet: *Dell* und das *DELL* Logo sind Marken der Dell Computer Corporation; *Intel* ist eine Marke der Intel Corporation oder ihrer Tochtergesellschaften in den USA oder anderen Ländern.

\* Zusätzliche Marken und Handelsnamen können in diesem Dokument verwendet werden, um entweder auf das Unternehmen oder das Produkt zu verweisen, zu dem die Marken oder Handelsnamen gehören. Die Intel Corporation erhebt mit Ausnahme der eigenen Marken und Handelsnamen keinerlei Anspruch bezüglich des Eigentums an Marken und Handelsnamen Dritter.

---

## **Einschränkungen und Beschränkungen**

Sämtliche in diesem Dokument enthaltenen Informationen, einschließlich aller Anweisungen, Hinweise, Bestimmungsgenehmigungen und Zertifizierungen, wurden vom Zulieferer zur Verfügung gestellt und wurden von Dell weder unabhängig geprüft noch getestet. Dell übernimmt keine Verantwortung für Schäden, die aus der Befolgung oder Nichtbefolgung dieser Anweisungen entstehen.

Alle Angaben oder Ansprüche bezüglich der Eigenschaften, Fähigkeiten, Geschwindigkeiten oder Qualifikationen eines in diesem Dokument erwähnten Teiles stammen vom Zulieferer und nicht von Dell. Dell lehnt insbesondere eine Bewertung der Genauigkeit, Vollständigkeit oder Begründung solcher Aussagen ab. Alle Fragen oder Anmerkungen bezüglich derartiger Angaben oder Ansprüche müssen an den Zulieferer gerichtet werden.

---

*Erste Veröffentlichung: Oktober 2003*

# Einführung: Benutzerhandbuch für Intel® PRO Netzwerkadapter – WMI und CDM Provider

---

## Überblick

Willkommen beim *Benutzerhandbuch für Intel® PRO Netzwerkadapter – WMI und CDM Provider*. Dieses Dokument behandelt die externe Ansicht der WMI und CDM Provider für Intel PRO Netzwerkadapter. Der WMI Provider (Windows Management Interface – Windows-Verwaltungsschnittstelle) ist ein Netzwerkkonfigurationsblock der NCS (Network Configuration Services – Netzwerkkonfigurationsservices). Diese Netzwerkkonfigurationsservices ermöglichen das Einrichten und Verwalten aller Netzwerktechnologien von Intel Endstationen mithilfe branchenüblicher Methoden. Der Intel PRO Common Diagnostic Model (CDM) Provider ist eine höhere Schnittstellen-API, die den CIM 2.5- und WMI-Normen entspricht. Auf der unteren Schnittstelle implementiert der CDM Provider eine Client-Schnittstelle in den unteren Schichten des PROSet-Softwarestacks. Aus diesem Grund werden alle PROSet-Mechanismen für die Datenintegrität beibehalten.

Bei den WMI und CDM Providern handelt es sich um Softwarekomponentensätze, die die Intel WMI-Netzwerkclassen implementieren. Diese Klassen beruhen auf Version 2.5 des Desktop Management Task Force (DMTF) CIM-Schemas.

In diesem Dokument werden die Informationen, die in den MOF-Dateien (Managed Object Format) für dieses Produkt enthalten sind, nicht wiederholt (detaillierte Angaben zur Bedeutung einzelner Attribute können z. B. in den MOF-Attributbeschreibungen gefunden werden).

Dieses Dokument beschreibt, wie eine WMI-Anwendung wie z. B. Intel PROSet Klassen zur Konfiguration eines Systemnetzwerks verwendet und wie eine WMI-Anwendung Klassen zum Testen einer Intel Netzwerkschnittstellenkarte einsetzt. Es wird vorausgesetzt, dass Sie mit WMI APIs und dem WMI SDK (erhältlich unter <http://www.microsoft.com/>) vertraut sind.

[Zurück zum Seitenanfang](#)

---

## Ähnliche Dokumente

Die folgenden Dokumente ermöglichen ein besseres Verständnis der WMI-Technologie:

- CIM-Schema Version 2.0, 2.2 – veröffentlicht von Desktop Management Task Force (DMTF). Erhältlich unter <http://www.dmtf.org>.
- Microsoft Windows Management Interface (und andere Verwaltungsinformationen). Erhältlich unter [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wmisdk/wmi/wmi\\_start\\_page.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wmisdk/wmi/wmi_start_page.asp).
- Web-Based Enterprise Management (WBEM) – eine Initiative von DMTF. Erhältlich unter <http://www.dmtf.org/standards/wbem>.
- WMI (Microsoft CIM-Implementierung) SDK. Erhältlich unter <http://msdn.microsoft.com/downloads/>.
- System Diagnostic Model White Paper von DMTF. Erhältlich unter <http://www.dmtf.org/standards/documents/CIM/DSP0138.pdf>.

 **WARNUNG:** Dieses Produkt enthält Informationen, die dazu missbraucht werden könnten, Computersysteme oder Netzwerke zu attackieren und/oder zu deaktivieren. Die Implementierung dieses Produkts setzt gründliche Kenntnisse der Sicherheitsfunktionen der Betriebssysteme von Microsoft voraus. Es wird Entwicklern und Benutzern dringend geraten, eventuelle Sicherheitsbedenken vor einer Implementierung dieses Produkts in einer Produktionsumgebung durch eine Anfrage bei Microsoft zu klären.

---

Lesen Sie bitte alle [Einschränkungen und Haftungsausschlüsse](#).

---



# WMI: Benutzerhandbuch für Intel® PRO Netzwerkadapter – WMI und CDM Provider

[Überblick](#)  
[Common Information Model \(CIM-Schema\)](#)

---

## Überblick

Web-Based Enterprise Management (WBEM) ist eine DMTF-Initiative (Desktop Management Task Force), die darauf abzielt, Enterprise-Systemmanagern eine standardisierte, kosteneffektive Methode zur Verwaltung von Endstationen anzubieten. Die WBEM-Initiative schließt eine Vielzahl von Aufgaben ein, die von der einfachen Konfiguration einer Arbeitsstation bis zum umfangreichen Enterprise-Management auf mehreren Plattformen reicht. Im Mittelpunkt dieser Initiative stehen das CIM-Modell (Common Information Model), ein ausdehnbares Datenmodell zur Darstellung von Objekten, die in typischen Verwaltungsumgebungen existieren, und die MOF-Sprache (Managed Object Format) zum Definieren und Speichern der modellierten Daten.

WMI (Windows Management Instrumentation) ist eine Implementierung der WBEM-Initiative für Microsoft\* Windows\*-Plattformen.

WMI besteht aus drei Hauptkomponenten:

- Kern – Diese Komponenten sind Bestandteil des Betriebssystems. Ohne sie ist eine WMI-aktivierte Anwendung nicht funktionsfähig, und sie müssen installiert sein, damit das SDK verwendet werden kann.
- SDK – Das SDK (Software Development Kit – Softwareentwicklungs-Kit) enthält Tools für das Durchsuchen des WMI-Schemas, das Erweitern des Schemas, das Erstellen von Providern sowie das Registrieren und Verwenden von WMI-Ereignissen. Es bietet außerdem Dokumentationsmaterialien, die bei der Entwicklung von WMI-aktivierten Anwendungen nützlich sind. Das SDK wird als Teil der Microsoft Platform SDK-Installation installiert und wird unter Windows NT4 SP4 oder SP5, Windows 2000, Windows Me, Windows XP und Windows Server 2003 unterstützt.
- Tools – Die Microsoft WMI Tools bieten Entwicklern die erforderlichen Hilfsmittel zum Erstellen einer vollkommen neuen Generation von Verwaltungsanwendungen und -lösungen. Dieses Tools-Paket enthält Dokumente und Tools, die Ihnen beim Zugreifen auf die Verwaltungsdaten von WMI helfen.

Die WMI-Architektur besteht aus den folgenden Komponenten:

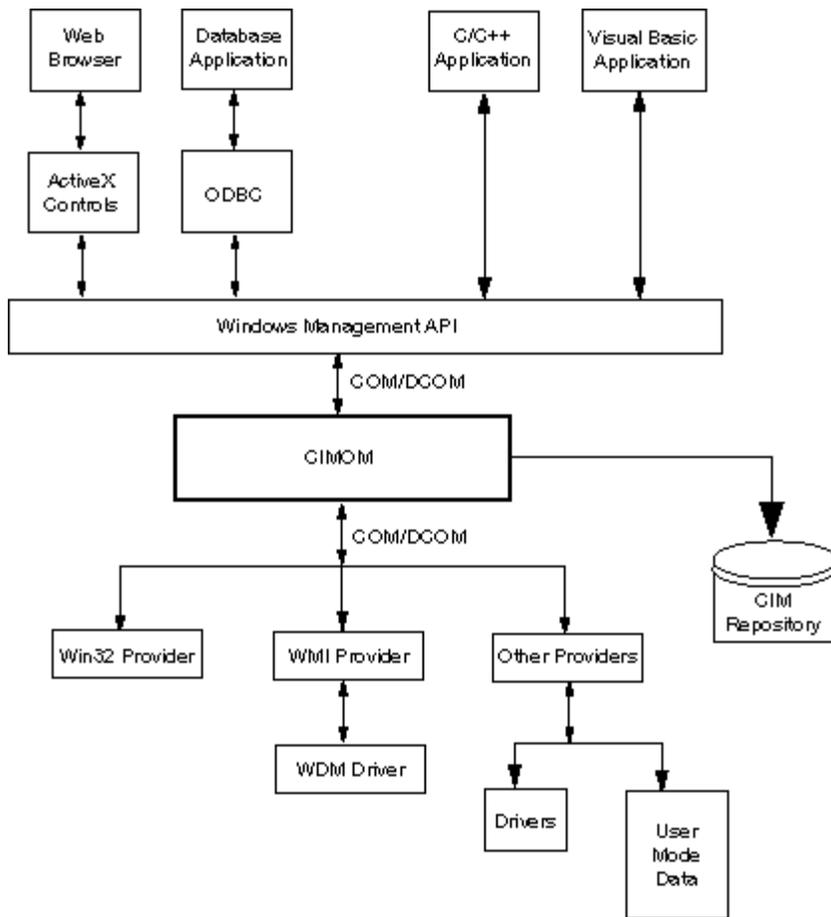
- Verwaltungsanwendungen
- Verwaltete Objekte
- Provider
- Verwaltungsinfrastruktur (besteht aus Windows Management und Windows Management-Repository)
- Windows Management API (verwendet COM/DCOM, um Provider und Verwaltungsanwendungen die Kommunikation mit der Windows Management-Infrastruktur zu ermöglichen)

Verwaltungsanwendungen dienen dazu, Daten von verwalteten Objekten (logische oder reale Enterprise-Komponenten) zu verarbeiten oder anzuzeigen. Diese Komponenten werden mithilfe von CIM modelliert, und Anwendungen greifen auf sie über Windows Management zu. Provider verwenden die Windows Management API, um Windows Management mit Daten von den verwalteten Objekten zu versorgen, Anfragen von Anwendungen abzuwickeln und Ereignisbenachrichtigungen zu generieren.

Die Verwaltungsinfrastruktur besteht aus Windows Management (zum Abwickeln der Kommunikation zwischen den Verwaltungsanwendungen und den Providern) und dem Windows Management-Repository (Depot zum Speichern von Daten). Im Windows Management-Repository werden die statischen Verwaltungsdaten festgehalten. Dynamische Daten werden nur bei Anfrage von Providern generiert. Die Daten werden entweder mithilfe des MOF-Sprachencompilers oder der Windows Management API im Repository abgelegt.

Anwendungen und Provider kommunizieren über Windows Management miteinander, wobei die Windows Management API die Services für z. B. Ereignisbenachrichtigung und Abfrageverarbeitung bereitstellt.

Das folgende Diagramm zeigt die Wechselbeziehung zwischen den Komponenten der WMI-Architektur:



[Zurück zum Seitenanfang](#)

## Common Information Model (CIM-Schema)

Das CIM-Modell (Common Information Model) präsentiert eine konsistente und einheitliche Ansicht der verschiedenen logischen und realen Objekte in einer verwalteten Umgebung. Verwaltete Objekte werden mithilfe objektorientierter Konstrukte wie z. B. Klassen dargestellt. Die Klassen schließen Eigenschaften ein, die Daten beschreiben, und Methoden, die Verhalten beschreiben. Das CIM wurde von DMTF so konzipiert, dass es sowohl betriebssystem- als auch plattformunabhängig verwendbar ist. Die WBEM-Technologie schließt eine Erweiterung des CIM für die Betriebssystemplattformen von Microsoft Windows ein. Weitere Informationen zum DMTF CIM-Schema finden Sie auf der DMTF-Website.

Das CIM definiert drei Klassenebenen:

- Klassen, die verwaltete Objekte repräsentieren, die für alle Verwaltungsbereiche gelten. Diese Klassen bieten ein Basisvokabular zur Analyse und Beschreibung verwalteter System und sind ein Bestandteil des so genannten Kernmodells.
- Klassen, die verwaltete Objekte repräsentieren, die für einen bestimmten Verwaltungsbereich gelten, aber unabhängig von einer bestimmten Implementierung oder Technologie sind. Diese Klassen sind Bestandteil des so genannten allgemeinen Modells.
- Klassen, die verwaltete Objekte repräsentieren, die technologiespezifische Ergänzungen zum allgemeinen Modell darstellen. Diese Klassen gelten in der Regel für spezielle Plattformen wie z. B. für UNIX oder die Microsoft Win32-Umgebung. Sie werden als erweitertes Modell bezeichnet.

Alle Klassen können durch eine Art "Erbfolge" miteinander verwandt sein, wobei die untergeordnete Klasse Daten und Methoden der übergeordneten Klasse einschließt. Solche Vererbungsbeziehungen werden in der Regel von der Verwaltungsanwendung, die sie verwendet, nicht erkannt. Es besteht allerdings auch keine Notwendigkeit für die Anwendungen, die Vererbungshierarchie zu berücksichtigen. Klassenhierarchien können mithilfe von Anwendungen im WMI Tools-Paket ermittelt werden (siehe WMI Tools unter <http://www.microsoft.com> für weitere Informationen).

Windows Management unterstützt auch Assoziationsklassen. Assoziationsklassen verbinden zwei verschiedene Klassen, um eine

benutzerdefinierte Beziehung zu modellieren. Sie sind sichtbar für Verwaltungsanwendungen. Windows Management definiert Assoziationsklassen, um Systemklassen zu unterstützen. Entwickler von Drittunternehmen können ebenfalls Assoziationsklassen für ihre Verwaltungsumgebung definieren.

WBEM unterstützt das Schemenkonzept zur Gruppierung der Klassen und Instanzen, die in einer bestimmten Verwaltungsumgebung verwendet werden. Das Platform SDK enthält zwei Schemen: das CIM-Schema und das Microsoft Win32-Schema. Das CIM-Schema enthält die Klassendefinitionen für die ersten beiden Ebenen des CIM. Diese Klassen repräsentieren verwaltete Objekte, die unabhängig von der Plattform zu jeder Verwaltungsumgebung gehören. Das Win32-Schema enthält Klassendefinitionen für verwaltete Objekte, die zu einer typischen Win32-Umgebung gehören.

Weitere Informationen über CIM finden Sie unter <http://www.dmtf.org>.

---

Lesen Sie bitte alle [Einschränkungen und Haftungsausschlüsse](#).

---

[Zurück zur Inhaltsseite](#)   [Zurück zum Seitenanfang](#)

# Hauptfunktionen: Benutzerhandbuch für Intel® PRO Netzwerkadapter – WMI und CDM Provider

[NCS WMI Provider – Funktionen](#)

[CDM Provider – Funktionen](#)

---

## NCS WMI Provider – Funktionen

WMI Provider bietet die folgenden Hauptfunktionen:

### Adapterfunktionen

- Auflisten aller realen Adapter, die von Intel® PROSet unterstützt werden
- Auflisten der Einstellungen eines installierten Adapters
- Hinzufügen/Entfernen/Aktualisieren der Einstellungen eines installierten Adapters
- Ermitteln der realen Geräteinformationen eines Adapters
- Ermitteln der Geräteinformationen für den Systemsteckplatz eines Adapters
- Ermitteln der IPv4-Protokolleinstellungen für einen Adapter
- Aktualisieren und Ändern des Boot Agent eines Adapters und der damit zusammenhängenden Einstellungen
- Deinstallieren eines Adapters

### Gruppenfunktionen

- Auflisten der Gruppen, die von Intel® PROSet unterstützt werden
- Erstellen/Entfernen einer Gruppe von Adaptern
- Hinzufügen/Entfernen/Aktualisieren der Einstellungen einer Gruppe
- Hinzufügen/Entfernen von Mitgliedsadaptern in einer Gruppe
- Ermitteln der IPv4-Protokolleinstellungen für eine Gruppe

### VLAN-Funktionen

- Auflisten der virtuellen LANs eines Adapters oder einer Gruppe
- Erstellen/Entfernen der virtuellen LANs eines realen Adapters oder einer Adaptergruppe
- Hinzufügen/Entfernen/Aktualisieren der Einstellungen des VLAN
- Ermitteln der IPv4-Protokolleinstellungen für eine Gruppe

### Ereignisbenachrichtigungsfunktionen

- Erteilen der Registriererlaubnis für den Client:
  - Adapterstatusereignisse
  - Adapterkonfigurationsereignisse
  - Sitzungsereignisse
  - Gruppenstatusereignisse
  - Gruppenkonfigurationsereignisse
  - VLAN-Konfigurationsereignisse

[Zurück zum Seitenanfang](#)

---

## CDM Provider – Funktionen

CDM Provider bietet die folgenden Hauptfunktionen:

- Tests können ausgeführt und angehalten und Testergebnisse können unabhängig vom Typ des Diagnosetests gelöscht werden.

Durch Verwendung der generellen Einstellungsklasse sollte die Kontrolle der Tests in einer Weise möglich sein, die von der CDM-Software eventuell so nicht vorhergesehen wurde.

- CDM Provider wird nur für Adapter verwendet.
  - Durch Verwendung der generellen Ergebnisklasse werden spezifische Ergebnismeldungen aus dem CDM-Providercode freigegeben.
  - Die Registrierungseinträge steuern die Provider-Ausführung.
  - Die Testergebnisse werden in die Ergebnisprotokolldatei geschrieben.
- 

Lesen Sie bitte alle [Einschränkungen und Haftungsausschlüsse](#).

---

[Zurück zur Inhaltsseite](#)   [Zurück zum Seitenanfang](#)

# Installierte Dateien: Benutzerhandbuch für Intel® PRO Netzwerkadapter – WMI und CDM Provider

[WMI-Dateien](#)

[CDM Provider-Dateien](#)

---

## WMI-Dateien

### Ausführbare Dateien

Der WMI Provider enthält die folgenden ausführbaren Dateien:

- **NcsWmiCo.exe** – Der Kern-Provider. Implementiert IANet\_NetService und die Kernereignisklassen.
- **NcsWmilm.exe** – Der Instanzen- und Methoden-Provider. Implementiert das Ethernet-Adapterschema, das Gruppenbildungsschema, das Einstellungsschema und das VLAN-Schema.
- **NcsWmiEv.exe** – Der Kern-Provider. Implementiert die Adapter-, Gruppen- und VLAN-Ereignisse.

## MOF-Dateien

Für sprachenneutrale und sprachenspezifische Daten werden unterschiedliche MOF-Dateien verwendet. Für den IntelNCS- und CIMV2-Namespace sind ebenfalls verschiedene MOF-Dateien vorhanden. Weitere Informationen finden Sie unter [Standorte und Lokalisierung](#) und [Fehlermeldungen](#).

Die folgenden MOF-Dateien sind für den IntelNCS-Namespace vorhanden:

- **NcsCmLn.mof** – CIM-Basisklassen, auf denen die NCS-Klassen aufbauen.
- **NcsCmEnu.mfl** – US-amerikanische Version der CIM-Basisklassen.
- **NcsCoLn.mof** – Kernklassen, die vom Kern-Provider implementiert werden.
- **NcsCoEnu.mfl** – US-amerikanische Textergänzungen für die Kernklassen.
- **NcslaLn.mof** – Klassen für die IEEE 802.3-Adapter, -Gruppen und -VLANs.
- **NcslaEnu.mfl** – US-amerikanische Textergänzungen für die 802.3-Kernklassen.

Die folgenden MOF-Dateien sind für den CIMV2-Namespace vorhanden:

- **C2CmLn.mof** – CIM-Basisklassen, auf denen die NCS-Klassen aufbauen.
- **C2CmEnu.mfl** – US-amerikanische Version der CIM-Basisklassen.
- **C2CoLn.mof** – Kernklassen, die vom Kern-Provider implementiert werden.
- **C2CoEnu.mfl** – US-amerikanische Textergänzungen für die Kernklassen.
- **C2laLn.mof** – Klassen für die IEEE 802.3-Adapter, -Gruppen und -VLANs.
- **C2laEnu.mfl** – US-amerikanische Textergänzungen für die 802.3-Kernklassen.

## Ressourcendateien

Die folgenden Ressourcendateien sind für WIM Provider vorhanden:

- **ENU\_8023.dll** – 8023-Ressource Englisch/USA.
- **ENU\_NWRC.dll** – WMI-Ressourcen für Kern-Provider Englisch/USA.
- **ENU\_NWR.dll** – WMI-Ressourcen für 8023-Provider Englisch/USA.

Die anderen lokalisierten Ressourcendateien können bei Bedarf geladen werden. Die Namen der lokalisierten Ressourcen-DLLs haben im Allgemeinen folgendes Muster: "\_mwr.dll". Dabei handelt es sich um den Sprachencode für die Lokalisierung (z. B. FRA für Standard-Französisch).

## CDM Provider-Dateien

### Ausführbare Dateien

Der CDM Provider enthält die folgenden ausführbaren Dateien:

- **Ncsdiag.exe** ist die wichtigste ausführbare Datei für die CDM-Diagnose. Diese Datei entspricht der WMI-Schnittstellenspezifikation von Microsoft\*. Auf sie wird als prozessexterner COM-Server zugegriffen.
- Andere ausführbare Dateien des Intel® PROSet-Softwarestacks

### MOF-Dateien

Die **.mof**-Hauptdateien werden nicht mit dem Produkt ausgeliefert, sondern werden in Übereinstimmung mit dem Microsoft\* WMI-Globalisierungsmodell in sprachenabhängige und sprachenneutrale Komponenten kompiliert. Weitere Informationen finden Sie im Microsoft\* WMI SDK-Kapitel (Plattform-SDK-Komponente) über WMI-Lokalisierung. Beachten Sie dort besonders den Abschnitt *Compiling Localized MOF Files (Kompilieren lokalisierter MOF-Dateien)*.

Durch Löschen einer **.mof**-Datei (DNcsCdmN.mof) werden zwar die Intel-abgeleiteten Klassendefinitionen gelöscht, aber nicht die DMTF-definierten Klassen, da andernfalls andere vorhandene Anwendungen gefährdet werden könnten.

Die typische Verwendung dieser CDM-Implementierung basiert auf dem CIMV2-Namespace. Die folgenden MOF-Dateien sind für einen IntelNCS-Namespace vorhanden:

Dateiname	Sprachentyp	Beschreibung
Cdla.mof	Master	Klassendefinitionen für Intel CDM-Implementierung
CdlaLn.mof	Sprachenneutral	Klassendefinitionen für Intel CDM-Implementierung
CdlaEnu.mfl	Setzt Englisch voraus	Sprachenerweiterungen für Intel CDM-Implementierung
CdCm.mof	Master	Kern-Superset der CDM-Klassendefinitionen
CdCmLn.mof	Sprachenneutral	Kern-Superset der CDM-Klassendefinitionen
CdCmEnu.mfl	Setzt Englisch voraus	Sprachenerweiterungen für Kern-Superset der CDM-Klassendefinitionen
DNcsCdmN.mof	Nicht zutreffend	Löscht Intel CDM-Klassen

Die folgenden MOF-Dateien sind für einen CIMV2-Namespace vorhanden:

Dateiname	Sprachentyp	Beschreibung
C2lcd.mof	Master	Klassendefinitionen für Intel CDM-Implementierung
C2lcdLn.mof	Sprachenneutral	Klassendefinitionen für Intel CDM-Implementierung
C2lcdEnu.mfl	Setzt Englisch voraus	Sprachenerweiterungen für Intel CDM-Implementierung
C2Cd.mof	Master	Kern-Superset der CDM-Klassendefinitionen
C2CdLn.mof	Sprachenneutral	Kern-Superset der CDM-Klassendefinitionen
C2CdEnu.mfl	Setzt Englisch voraus	Sprachenerweiterungen für Kern-Superset der CDM-Klassendefinitionen
DNcsCdm2.mof	Nicht zutreffend	Löscht Intel CDM-Klassen

**Hinweis:** Zur Lokalisierung muss die korrekte sprachenabhängige **.mof**-Datei hinzugefügt werden.

## Ressourcendateien

Die folgenden Ressourcendateien sind für den CDM Provider vorhanden:

- **ENU\_Diag.dll** – WMI-Ressourcen für Diagnose-Provider Englisch/USA.
- 

Lesen Sie bitte alle [Einschränkungen und Haftungsausschlüsse](#).

---

[Zurück zur Inhaltsseite](#)   [Zurück zum Seitenanfang](#)

[Zurück zur Inhaltsseite](#)

## **Sicherheit: Benutzerhandbuch für Intel® PRO Netzwerkadapter – WMI und CDM Provider**

---

Die WMI und CDM Provider verwenden Client-Impersonation für die Sicherheitsverwaltung. Jeder an die Provider gerichtete Aufruf wird im Client-Sicherheitskontext ausgeführt, der an die unteren Schichten weitergeleitet wird. Ein einzelner oder alle Vorgänge können versagen, wenn Sie nicht über Administratorrechte auf dem Zielgerät verfügen.

---

Lesen Sie bitte alle [Einschränkungen und Haftungsausschlüsse](#).

---

[Zurück zur Inhaltsseite](#)   [Zurück zum Seitenanfang](#)

# Namespace und Kontext: Benutzerhandbuch für Intel® PRO Netzwerkadapter – WMI und CDM Provider

Die CIM-Klassen befinden sich in einem Namespace. Der reguläre Microsoft\* Namespace lautet **root/cimv2** und beruht auf CIM v2.2 oder **root/default**. Die Klassen von WMI und CDM Provider können diesem Namespace hinzugefügt werden. Die Provider beruhen auf CIM v2.5. Aus diesem Grund und aufgrund der Unterschiede in den Schlüsseln befinden sich die Provider-Klassen in dem separaten Namespace **root/IntelINCS**.

## WBEM-Kontext

Kontextobjekte bieten zusätzliche Informationen zu den Providern, die nicht als Parameter an eine WMI API-Methode geleitet werden können. Verwenden Sie **IWbemContext**, um Kontext-Qualifizierungsmerkmale zu registrieren. Der Schnittstellenpointer für das Kontextobjekt wird als der letzte Parameter einer **IWbemServices**-Methode weitergegeben.

Die folgende Tabelle enthält die Kontext-Qualifizierungsmerkmale (bezeichnete Werte), die von den Providern verwendet werden. Die meisten Qualifizierungsmerkmale, wie z. B. SessionHandle, werden nur in Verbindung mit speziellen funktionalen Bereichen der Provider verwendet. LocaleID, MachineName und ApplicationName können dagegen für alle **IWbemServices**-Aufrufe eingestellt werden.

Wenn kein Kontext an die Provider weitergegeben wird, verwenden sie die LocaleID, die im **Initialize**-Aufruf der Provider weitergegeben wurde. Jeder mit einem Kontext ausgeführte Lesevorgang erfasst die aktuelle Konfiguration bis zu dem Zeitpunkt, an dem ein Schreibvorgang ausgeführt wird. Nachfolgende Lesevorgänge zeigen das System so, wie es nach dem erfolgreichen Schreibvorgang aussehen würde. Ein NULL-Kontext kann für Lesevorgänge verwendet werden.

Kontext-Qualifizierungsmerkmale	Variantentyp	Beschreibung
SessionHandle	VT_BSTR	Identifiziert die Kopie der IANet-Netzwerkclasses, die die Anwendung verwendet. Die Anwendung kann keine Änderungen an den Klassen oder ihren Attributen durchführen, ohne zuerst ein Sitzungshandle einzurichten. Im Abschnitt über die IANet_NetService-Klasse erfahren Sie, wie ein Sitzungshandle eingerichtet und verwendet wird. Dieses Qualifizierungsmerkmal ist nicht erforderlich, wenn die Anwendung die Daten aus den Klassen nur liest. Mit dem Sitzungshandle ist die NCS-Software in der Lage, den gleichzeitigen mehrfachen Zugang zur Konfiguration zu verwalten, ohne dass ein Benutzer alle anderen Benutzer "aussperrt". Jede Sitzung verwendet einen anderen Cache, um die durchgeführten Änderungen zu speichern. Wenn von mehreren Benutzern gleichzeitig Änderungen durchgeführt werden, ist der Benutzer erfolgreich, dessen Änderungen zuerst übernommen werden. Alle anderen Benutzer-Caches werden dadurch ungültig.
LocaleID	VT_BSTR	Microsoft ID für einen Standort. Dieser Wert ist erforderlich, wenn für die Anwendung lokalisierte Textzeichenfolgen von den Providern notwendig sind. Alle Fehlermeldungen und Warnungen werden in Englisch angezeigt, es sei denn, die erforderliche LocaleID wird verwendet.
ApplicationName	VT_BSTR	Der Name der Anwendung die den Aufruf ausgeführt hat. Diese Angabe ist für das Protokoll erforderlich.
MachineName	VT_BSTR	Der Name der Maschine, die die Verbindung zu den Providern herstellt. Diese Angabe ist für das Protokoll erforderlich.
PreCheck	VT_BOOL	Dieser Boolesche Wert wird verwendet, um den Providern mitzuteilen, dass der Client vor dem Ausführen eines Vorgangs zu überprüfen versucht, ob dieser Vorgang zulässig ist. Ein solcher Vorgang ist beispielsweise das Hinzufügen eines Adapters zu einer Gruppe.

		<p>Werte:</p> <ul style="list-style-type: none"> <li>• TRUE (WAHR) = Provider führt den Vorgang nicht aus, aber gibt einen Fehlercode und den erweiterten Status zurück, falls der Vorgang nicht erlaubt ist.</li> <li>• FALSE (FALSCH) = Provider führt den Vorgang aus.</li> </ul> <p>Das Fehlen dieses Qualifizierungsmerkmals wird so interpretiert, als ob das Attribut FALSE (FALSCH) wäre.</p>
WarningErrorCode	VT_I4	<p>Bei einigen Vorgängen ist es unter Umständen notwendig, Warnungen an den Benutzer zu senden (beim Hinzufügen eines Adapters muss eine Gruppe z. B. unter gewissen Umständen neu geladen werden). WMI bietet keinen Mechanismus für diese Aktion. Wenn dieses Qualifizierungsmerkmal vorhanden und nicht gleich null ist, gibt der Provider E_FAIL zurück, sofern der Vorgang zwar gelungen ist, aber eine Warnung damit verbunden ist. Der Client sollte den erweiterten Status verwenden, um den Text jeder Warnung zu sehen.</p>

Lesen Sie bitte alle [Einschränkungen und Haftungsausschlüsse](#).

[Zurück zur Inhaltsseite](#)   [Zurück zum Seitenanfang](#)

# Standorte und Lokalisierung: Benutzerhandbuch für Intel® PRO Netzwerkadapter – WMI und CDM Provider

[Lokalisierte MOF-Dateien](#)  
[Lokalisierte Attributdateien](#)

---

Die Lokalisierung von WMI und CDM Provider ist durch zwei Kategorien geprägt: lokalisierte MOF-Dateien und lokalisierte Attributdaten.

## Lokalisierte MOF-Dateien

Alle MOF-Dateien, die von den Providern verwendet werden, werden entsprechend dem Microsoft Windows\* Management Instrumentation (WMI) Globalisierungsmodell lokalisiert. Zu diesem Zweck wird jede Klassendefinition folgendermaßen aufgetrennt:

- Eine sprachenneutrale Version, die nur die Basisklassendefinition in der **.mof**-Datei enthält.
- Eine sprachenspezifische Version, die lokalisierte Informationen enthält, z. B. Eigenschaftsbeschreibungen, die sich auf einen bestimmten Standort in der entsprechenden **.mfl**-Datei beziehen.

## Unterstützte Sprachen

Chinesisch (Taiwan)  
Chinesisch (Volksrepublik)  
Dänisch  
Niederländisch (Niederlande)  
Englisch (USA)  
Finnisch  
Französisch (Frankreich)  
Deutsch (Deutschland)  
Italienisch (Italien)  
Japanisch  
Norwegisch (Bokmal)  
Portugiesisch (Brasilien)  
Spanisch (Spanien – Modern)  
Schwedisch

## Klassenspeicherung

Die sprachenspezifischen Klassendefinitionen werden in einem untergeordneten Unter-Namespace unterhalb des Namespace gespeichert, der eine sprachenneutrale Basisklassendefinition enthält. Für WMI und CDM Provider ist beispielsweise der untergeordnete Namespace **ms\_409** unterhalb des Namespace **root/IntelIncs** für den englischsprachigen Standort vorhanden. Für jede unterstützte Sprache existiert ein ähnlicher untergeordneter Unter-Namespace unterhalb des Namespace **root/IntelIncs**.

## Lokalisierte MOF-Support in cimv2-Namespace

Für **root/cimv2**-Namespaces werden die Provider-Klassen (z. B. **IA\_Net\_classes**) von den Basisklassen abgeleitet, die diesem Namespace von WMI hinzugefügt werden. Ein Unter-Namespace mit sprachenspezifischen Klassendefinitionen für Basisklassen existiert bereits unterhalb des Namespace **root/cimv2**. Die sprachenspezifischen **IA\_Net**-Klassendefinitionen werden diesem bereits vorhandenen, untergeordneten Namespace hinzugefügt. Aufgrund dieser Abhängigkeit von Basisklassen wird die MOF-Lokalisierung nur im Standard-Systemstandort durchgeführt.

## Runtime-Support

Zum Abrufen lokalisierter Daten kann eine WMI-Anwendung den Standort mithilfe eines strLocale-Parameters in **SWbemLocator::ConnectServer**- und **IWbemLocator::ConnectServer**-Aufrufen kennzeichnen. Wenn der Standort nicht angegeben ist, wird der Standardstandort für das System verwendet (z. B. MS\_409 für US-Englisch). Dieser Standort wird zur Auswahl des korrekten Namespace verwendet, wenn die englischen Zeichenfolgen hinzugefügt werden.

Darüber hinaus müssen **IWbemServices::GetObject**, **SwbemServices.GetObject**, **IWbemServices::ExecQuery** und **SWbemServices.ExecQuery** das WBEM\_FLAG\_USE\_AMENDED\_QUALIFIERS-Kennzeichen angeben, um lokalisierte Daten mit der Basisdefinition anzufordern. Dies ist bei allen Funktionen erforderlich, die mithilfe von Wertzuordnungen, Anzeigebeschreibungen oder anderen ergänzten Qualifizierungsmerkmalen aus den MOF-Dateien anzeigbare Werte erzeugen.

[Zurück zum Seitenanfang](#)

---

## Lokalisierte Attributdaten

Um lokalisierte Attributdaten (z. B. Fehlermeldungen) erhalten zu können, müssen die Provider bei jedem Aufruf über den Standort des Aufrufinitiators informiert sein. Damit dieser Vorgang korrekt ablaufen kann, muss der Client den Standort zum Kontextobjekt hinzufügen, das bei jedem Aufruf weitergegeben wird (siehe die Informationen zum WBEM-Kontext unter [Namespace und Kontext](#)). Falls die Provider eine lokalisierbare Zeichenfolge zurückgeben müssen, versuchen sie, eine Ressourcen-DLL zu laden, die für den Client-Standort geeignet ist. Falls keine passende Ressourcen-DLL vorhanden ist, geben die Provider Zeichenfolgen in Englisch zurück.

Lesen Sie bitte alle [Einschränkungen und Haftungsausschlüsse](#).

---

[Zurück zur Inhaltsseite](#)   [Zurück zum Seitenanfang](#)

# Fehlermeldungen: Benutzerhandbuch für Intel® PRO Netzwerkadapter – WMI und CDM Provider

[Überblick](#)  
[Fehlercodes](#)

---

## Überblick

In diesem Abschnitt über IANet\_ExtendedStatus wird beschrieben, wie Sie mit Fehlern umgehen, die von WMI und CDM Provider generiert werden. Auf welche Weise und wann ein Fehlerobjekt zurückgegeben wird, hängt davon ab, ob ein Aufruf synchron, semi-synchron oder asynchron erfolgt. In den meisten Fällen ist HRESULT bei Auftreten eines Fehler auf WBEM\_E\_FAILED eingestellt. Zu diesem Zeitpunkt ist jedoch nicht bekannt, ob WMI oder die Provider den Fehler generiert haben.

Verwenden Sie bei synchronen Aufrufen GetErrorInfo(), um das IErrorInfo-Objekt zu ermitteln. Verwenden Sie QueryInterface(), um das IWbemClassObject zu erhalten, das die Fehlerinformationen enthält.

Bei asynchronen Aufrufen wird das Fehlerobjekt als IWbemClassObject zurückgegeben. Es ist das letzte Element im letzten SetStatus()-Aufruf. Nachdem Sie die Fehlerobjekt-Instanz ermittelt haben, können Sie die \_\_Class-Eigenschaft überprüfen, um den Ursprung des Fehlers zu bestimmen. WMI erstellt eine Instanz von \_\_ExtendedStatus, und die Provider erstellen eine Instanz von IANet\_ExtendedStatus für Fehler, die mit IANet\_classes zusammenhängen. IANet\_ExtendedStatus wird von \_\_ExtendedStatus abgeleitet und enthält die folgenden Fehlerobjekt-Qualifizierungsmerkmale:

- Description – Beschreibung des Fehlers, die auf den aktuellen Standort zugeschnitten ist.
- File – Codedatei, in der der Fehler generiert wurde.
- Line – Nummer der Zeile, in der sich der Fehler in der Codedatei befindet.
- ParameterInfo – Klasse bzw. Attribut, die bzw. das verwendet wurde, als der Fehler auftrat.
- Operation – Vorgang, bei dessen versuchter Ausführung der Fehler auftrat.
- ProviderName – Name des Providers, der den Fehler verursacht hat.
- StatusCode – Code, der vom internen fehlgeschlagenen Aufruf zurückgegeben wurde.
- SessionHandle – Sitzungshandle, das für den Vorgang verwendet wurde.
- RuleFailureReasons – Grund für das Versagen des Vorgangs. Ein Vorgang kann fehlschlagen, weil eine technische Regel versagt hat (bei bestimmten Gruppen benötigen Sie z. B. einen Verwaltungsadapter).

[Zurück zum Seitenanfang](#)

---

## Fehlercodes

Für alle Fehlercodes geben die Provider eine Beschreibung aus, die auf den jeweiligen Standort zugeschnitten ist. Fehlercodes werden im Format HRESULT angegeben, wobei der Schweregrad auf 1 und die Fähigkeit auf ITF eingestellt sind. Eine Anwendung kann die folgenden Codes als Basis für eine Wiederherstellungsaktion verwenden:

- 0x80040901 – "WMI: Put property failed" (WMI: Put-Eigenschaft fehlgeschlagen)
- 0x80040902 – "WMI: No class object" (WMI: Kein Klassenobjekt)
- 0x80040903 – "WMI: Failed to create class" (WMI: Klasse konnte nicht erstellt werden)
- 0x80040904 – "WMI: Failed to spawn instance of class" (WMI: Instanz der Klasse konnte nicht erzeugt werden)
- 0x80040905 – "WMI: Failed to create safe array" (WMI: Sicheres Array konnte nicht erstellt werden)
- 0x80040906 – "WMI: Failed to create safe array" (WMI: Sicheres Array konnte nicht platziert werden)
- 0x80040907 – "WMI: Failed to return object to WMI" (WMI: Objekt konnte nicht an WMI zurückgegeben werden)
- 0x80040908 – "WMI: Put property failed" (WMI: Get-Eigenschaft fehlgeschlagen)
- 0x80040909 – "WMI: Unexpected type while getting property" (WMI: Unerwarteter Typ beim Ermitteln der Eigenschaft)
- 0x8004090A – "WMI: Class not implemented by this provider" (WMI: Klasse wurde von diesem Provider nicht implementiert)
- 0x8004090B – "WMI: Unable to parse WQL statement" (WMI: WQL-Aussage konnte nicht analysiert werden)
- 0x8004090C – "WMI: Providers only support WQL" (WMI: Provider unterstützen nur WQL)

- 0x8004090D – "WMI: Parameter in context has the wrong type" (WMI: Parameter im Kontext ist vom falschen Typ)
- 0x8004090E – "WMI: Error formatting debug log" (WMI: Fehler beim Formatieren des Debug-Protokolls)
- 0x8004090F – "WMI: bad object path" (WMI: Unbrauchbarer Objektpfad)
- 0x80040910 – "WMI: Failed to update setting" (WMI: Einstellung konnte nicht aktualisiert werden)
- 0x80040911 – "WMI: Null parameter passed to method" (WMI: Null-Parameter an Methode geleitet)
- 0x80040912 – "Setting value too small." (Einstellungswert zu klein)
- 0x80040913 – "Setting value too big." (Einstellungswert zu groß)
- 0x80040914 – "Setting not in step" (Einstellung nicht konform)
- 0x80040915 – "String setting is too long" (Zeichenfolgen-Einstellung ist zu lang)
- 0x80040916 – "Setting is not one of the allowed values" (Einstellung ist keiner der zulässigen Werte)
- 0x80040917 – "WMI: Qualifier not found" (WMI: Qualifizierungsmerkmal nicht gefunden)
- 0x80040918 – "WMI: Qualifier set not found" (WMI: Qualifizierungsmerkmal-Satz nicht gefunden)
- 0x80040919 – "WMI: Safe array access failed" (WMI: Sicherer Array-Zugriff fehlgeschlagen)
- 0x8004091A – "WMI: Unhandled exception" (WMI: Unbehandelte Ausnahme)
- 0x8004091B – "WMI: Operation is not supported for this class" (WMI: Vorgang wird für diese Klasse nicht unterstützt)
- 0x8004091C – "WMI: Unexpected event class" (WMI: Unerwartete Ereignisklasse)
- 0x8004091D – "WMI: Bad event data" (WMI: Unbrauchbare Ereignisdaten)
- 0x8004091E – "WMI: Operation succeeded with warnings" (WMI: Vorgang wurde mit Warnungen ausgeführt)
- 0x8004081F – "WMI: The NCS Service has been stopped." (WMI: Der NCS-Service wurde angehalten)
  
- 0x80040801 – "EAL: Internal exception" (EAL: Interne Ausnahme)
- 0x80040802 – "EAL: General failure" (EAL: Allgemeiner Fehler)
- 0x80040803 – "EAL: Not initialized" (EAL: Nicht initialisiert)
- 0x80040804 – "EAL: Failed to initialize." (EAL: Initialisierung nicht gelungen)
- 0x80040805 – "EAL: Session limits exceeded" (EAL: Sitzungsgrenzen wurden überschritten)
- 0x80040806 – "EAL: Out of memory" (EAL: Nicht genügend Speicher)
- 0x80040807 – "EAL: Rule syntax error" (EAL: Regelsyntaxfehler)
- 0x80040808 – "EAL: Unexpected end of list" (EAL: Unerwartetes Listenende)
- 0x80040809 – "EAL: Rule link error" (EAL: Regelverbindungsfehler)
- 0x8004080A – "EAL: Device Creation Failed" (EAL: Geräteerstellung fehlgeschlagen)
- 0x8004080B – "EAL: Media service not found" (EAL: Medienservice nicht auffindbar)
- 0x8004080C – "EAL: Device service not found" (EAL: Geräteservice nicht auffindbar)
- 0x8004080D – "EAL: PCI bus module not found" (EAL: PCI-Busmodul nicht auffindbar)
- 0x8004080E – "EAL: Adapter is a member of a team" (EAL: Adapter ist Mitglied einer Gruppe)
- 0x8004080F – "EAL: Rule Access Point creation error" (EAL: Fehler bei der Erstellung des Regelzugriffspunkts)
- 0x80040810 – "EAL: Registry key error" (EAL: Registrierungsschlüsselfehler)
- 0x80040811 – "EAL: Registry XML file path error" (EAL: Pfadfehler für Registrierungs-XML-Datei)
- 0x80040812 – "EAL: Unknown event class" (EAL: Unbekannte Ereignisklasse)
- 0x80040813 – "EAL: Unknown module id" (EAL: Unbekannte Modul-ID)
- 0x80040814 – "EAL: Rule service not found" (EAL: Regelservice nicht auffindbar)
- 0x80040815 – "EAL: NULL input pointer" (EAL: NULL-Eingabezeiger)
- 0x80040816 – "EAL: Rule grammar error" (EAL: Regelgrammatikfehler)
- 0x80040817 – "EAL: Rule failed" (EAL: Regel fehlgeschlagen)
- 0x80040818 – "EAL: Setting is already grouped" (EAL: Einstellung ist bereits gruppiert)
  
- 0x80040220 – "Sync Layer: Team removal failed." (Sync-Schicht: Gruppenentfernung fehlgeschlagen)
- 0x80040221 – "Sync Layer: Vlan creation failed." (Sync-Schicht: VLAN-Erstellung fehlgeschlagen)
- 0x80040222 – "Sync Layer: Vlan removal failed." (Sync-Schicht: VLAN-Entfernung fehlgeschlagen)
- 0x80040223 – "Sync Layer: Adapter removal failed." (Sync-Schicht: Adapterentfernung fehlgeschlagen)
- 0x80040224 – "Sync Layer: Setting Change/Creation/Removal failed." (Sync-Schicht: Ändern/Erstellen/Entfernen der Einstellung fehlgeschlagen)
- 0x80040225 – "Sync Layer: Parameter Change/Removal failed." (Sync-Schicht: Ändern/Entfernen des Parameters fehlgeschlagen)
- 0x80040226 – "Sync Layer: NetConfig subsystem locked. (Sync-Schicht: NetConfig-Subsystem gesperrt) "
- 0x80040227 – "Sync Layer: System Update In Progress. Please try again later." (Sync-Schicht: System wird derzeit aktualisiert. Versuchen Sie es später noch einmal.)
- 0x80040228 – "Sync Layer: Adapter is Locked" (Sync-Schicht: Adapter ist gesperrt)
- 0x80040229 – "Sync Layer: Flash read failed." (Sync-Schicht: Flash-Lesevorgang fehlgeschlagen)
- 0x8004022A – "Sync Layer:"

- 0x80040210 – "Sync Layer: Invalid event offset" (Sync-Schicht: Ungültiger Ereignisversatz)
- 0x80040211 – "Sync Layer: Invalid input" (Sync-Schicht: Ungültige Eingabe)
- 0x80040212 – "Sync Layer: Invalid key" (Sync-Schicht: Ungültiger Schlüssel)
- 0x80040213 – "Sync Layer: Adapter not team member" (Sync-Schicht: Adapter ist kein Gruppenmitglied)
- 0x80040214 – "Sync Layer: Driver not loaded" (Sync-Schicht: Treiber nicht geladen)
- 0x80040215 – "Sync Layer: Client impersonation failed" (Sync-Schicht: Client-Impersonation fehlgeschlagen)
- 0x80040216 – "Sync Layer: Caught exception" (Sync-Schicht: Ausnahme erfasst)
- 0x80040217 – "Sync Layer: Session not locked" (Sync-Schicht: Sitzung nicht gesperrt)
- 0x80040218 – "Sync Layer: Hardware access layer is not available" (Sync-Schicht: Hardware-Zugriffsschicht ist nicht verfügbar)
- 0x80040219 – "Sync Layer: Flash not available" (Sync-Schicht: Flash nicht verfügbar)
- 0x8004021A – "Sync Layer: Diagnostics not supported" (Sync-Schicht: Diagnose wird nicht unterstützt)
- 0x8004021B – "Sync Layer: Diagnostic test not running" (Sync-Schicht: Diagnosetest wird nicht ausgeführt)
- 0x8004021C – "Sync Layer: Boot Agent update not available" (Sync-Schicht: Aktualisierung für Boot Agent nicht verfügbar)
- 0x8004021D – "Sync Layer: Boot Agent corrupted." (Sync-Schicht: Boot Agent beschädigt)
- 0x8004021E – "Sync Layer: Flash write failed." (Sync-Schicht: Flash-Schreibvorgang fehlgeschlagen)
- 0x8004021F – "Sync Layer: Team creation failed." (Sync-Schicht: Gruppenerstellung fehlgeschlagen)
- 0x80040201 – "Sync Layer: Initialization failed" (Sync-Schicht: Initialisierung fehlgeschlagen)
- 0x80040202 – "Sync Layer: Invalid initialization handle" (Sync-Schicht: Ungültiges Initialisierungshandle)
- 0x80040203 – "Sync Layer: Session handle already exists" (Sync-Schicht: Sitzungshandle ist bereits vorhanden)
- 0x80040204 – "Sync Layer: Invalid session handle" (Sync-Schicht: Ungültiges Sitzungshandle)
- 0x80040205 – "Sync Layer: The maximum number of sessions has been reached." (Sync-Schicht: Die maximale Sitzungsanzahl wurde erreicht)
- 0x80040206 – "Sync Layer: The session lock handle already exists" (Sync-Schicht: Das Sitzungssperrhandle ist bereits vorhanden)
- 0x80040207 – "Sync Layer: Invalid session lock handle" (Sync-Schicht: Ungültiges Sitzungssperrhandle)
- 0x80040208 – "Sync Layer: Session already locked" (Sync-Schicht: Sitzung ist bereits gesperrt)
- 0x80040209 – "Sync Layer: Invalid media service module Id" (Sync-Schicht: Ungültige ID für Medienservicemodul)
- 0x8004020A – "Sync Layer: Invalid Advanced Service Module Id" (Sync-Schicht: Ungültige ID für erweitertes Servicemodul)
- 0x8004020B – "Sync Layer: Invalid device service module Id" (Sync-Schicht: Ungültige ID für Geräteservicemodul)
- 0x8004020C – "Sync Layer: Invalid component type Id" (Sync-Schicht: Ungültige ID für Komponententyp)
- 0x8004020D – "Sync Layer: Invalid bus interface module Id" (Sync-Schicht: Ungültige ID für Busschnittstellenmodul)
- 0x8004020E – "Sync Layer: Invalid sink window handle" (Sync-Schicht: Ungültiges Sinkfensterhandle)
- 0x8004020F – "Sync Layer: Invalid event Id" (Sync-Schicht: Ungültige Ereignis-ID)
  
- 0x80040401 – "HAM PCI: Invalid memory map address" (HAM PCI: Ungültige Speicherzuordnungsadresse)
- 0x80040402 – "HAM PCI: Configuration driver failed to load" (HAM PCI: Konfigurationstreiber wurde nicht geladen)
- 0x80040403 – "HAM PCI: Configuration driver version mismatch" (HAM PCI: Konfigurationstreiberversion stimmt nicht überein)
- 0x80040404 – "HAM PCI: Device slot not found" (HAM PCI: Gerätesteckplatz nicht auffindbar)
- 0x80040405 – "HAM PCI: Diagnostic driver failed to load" (HAM PCI: Diagnosetreiber wurde nicht geladen)
- 0x80040406 – "HAM PCI: Diagnostic driver version mismatch" (HAM PCI: Diagnosetreiberversion stimmt nicht überein)
- 0x80040407 – "HAM PCI: Diagnostic driver initialization failed" (HAM PCI: Diagnosetreiberinitialisierung fehlgeschlagen)
- 0x80040408 – "HAM PCI: Diagnostics not initialized" (HAM PCI: Diagnose wurde nicht initialisiert)
- 0x80040409 – "HAM PCI: Diagnostics already initialized" (HAM PCI: Diagnose ist bereits initialisiert)
- 0x8004040A – "HAM PCI: Diagnostic test already running" (HAM PCI: Diagnosetest wird bereits ausgeführt)
- 0x8004040B – "HAM PCI: Diagnostic test not running" (HAM PCI: Diagnosetest wird nicht ausgeführt)
- 0x8004040C – "HAM PCI: Diagnostic test terminated" (HAM PCI: Diagnosetest wurde beendet)
- 0x8004040D – "HAM PCI: Diagnostic Invalid test number" (HAM PCI: Ungültige Diagnosetestnummer)
- 0x8004040E – "HAM PCI: Diagnostic hardware missing" (HAM PCI: Diagnosehardware fehlt)
- 0x8004040F – "HAM PCI: Diagnostic send receive initialization failed" (HAM PCI: Senden/Empfangen-Initialisierung für Diagnose fehlgeschlagen)
  
- 0x80040511 – "Media Service: NDIS IO call failed" (Medienservice: NDIS IO-Aufruf fehlgeschlagen)
- 0x80040512 – "Media Service: Miniport not loaded" (Medienservice: Miniport nicht geladen)
- 0x8004051B – "Media Service: Invalid device handle" (Medienservice: Ungültiges Gerätehandle)
- 0x8004051C – "Media Service: Invalid adapter handle" (Medienservice: Ungültiges Adapterhandle)
- 0x8004051D – "Media Service: Invalid team handle" (Medienservice: Ungültiges Gruppenhandle)
- 0x8004051E – "Media Service: Invalid VLAN handle" (Medienservice: Ungültiges VLAN-Handle)
- 0x8004051F – "Media Service: Device missing" (Medienservice: Gerät fehlt)

- 0x80040520 – "Media Service: Invalid setting type" (Medienservice: Ungültiger Einstellungstyp)
  - 0x80040521 – "Media Service: Unknown invalid object" (Medienservice: Unbekanntes ungültiges Objekt)
  - 0x80040522 – "Media Service: Invalid Setting Handle" (Medienservice: Ungültiges Einstellungshandle)
  - 0x80040523 – "Media Service: Invalid Team Mode" (Medienservice: Ungültiger Gruppenmodus)
  - 0x80040525 – "Media Service: Setting Already Exists" (Medienservice: Einstellung ist bereits vorhanden)
  
  - 0x80042001 – "RAP: Already initialized" (RAP: Bereits initialisiert)
  - 0x80042002 – "RAP: Invalid XML file" (RAP: Ungültige XML-Datei)
  - 0x80042003 – "RAP: XML load error" (RAP: XML-Ladefehler)
  - 0x80042004 – "RAP: Not initialized" (RAP: Nicht initialisiert)
  - 0x80042005 – "RAP: Rule not extracted before" (RAP: Regel nicht zuvor extrahiert)
  - 0x80042006 – "RAP: Conditions count mismatch" (RAP: Bedingungsähler stimmt nicht überein)
  - 0x80042007 – "RAP: Results apply error" (RAP: Ergebnisanwendungsfehler)
  - 0x80042008 – "RAP: Invalid rule" (RAP: Ungültige Regel)
  - 0x80042009 – "RAP: Node not found" (RAP: Knoten nicht auffindbar)
  - 0x8004200A – "RAP: Error no single node" (RAP: Fehler – kein Einzelknoten)
  - 0x8004200B – "RAP: No action rule" (RAP: Keine Aktionsregel)
  - 0x8004200C – "RAP: Zero condition" (RAP: Null-Bedingung)
  - 0x8004200D – "RAP: Zero action" (RAP: Null-Aktion)
  - 0x8004200E – "RAP: XML Decode error" (RAP: XML-Dekodierfehler)
- 

Lesen Sie bitte alle [Einschränkungen und Haftungsausschlüsse](#).

---

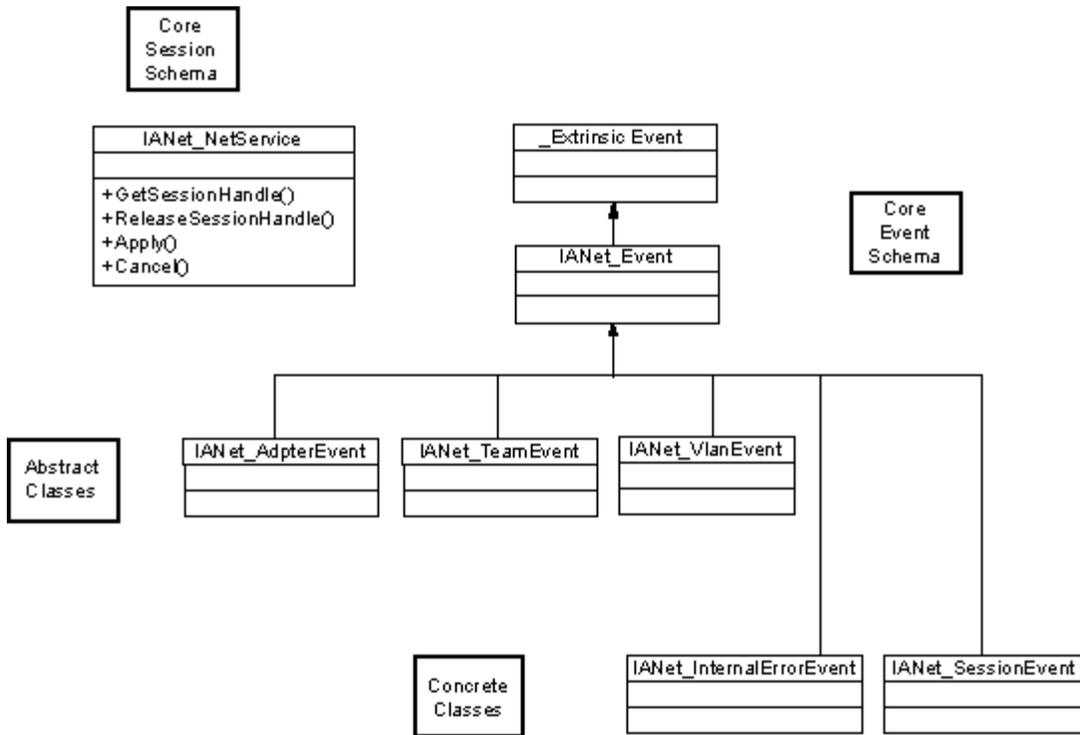
[Zurück zur Inhaltsseite](#)   [Zurück zum Seitenanfang](#)

# Das Kernschema: Benutzerhandbuch für Intel® PRO Netzwerkadapter – WMI und CDM Provider

- [Überblick](#)
- [IANet\\_NetService](#)
- [Kernereignisse](#)
- [Verwendungsfälle](#)

## Überblick

Das Kernschema besteht aus der IANet\_NetService-Klasse und den Kernereignisklassen.



## IANet\_NetService

### Zweck

Die IANet\_NetService-Klasse ist das Stammobjekt des IANet\_-Schemas. Diese Klasse ermöglicht es dem Client, auf die Sitzungen zuzugreifen, die zur Ausführung von Sätzen erforderlich sind.

### Instanzen

Es gibt eine Instanz dieses Objekts. Der Client sollte sich nicht auf den für diese Klasse verwendeten Schlüssel verlassen. Der Client sollte stattdessen die Instanz der Klasse durch Auflisten aller Instanzen von IANet\_NetService ermitteln.

### Erstellen von Instanzen

Sie können keine Instanzen von IANet\_NetService erstellen.

### Entfernen von Instanzen

Sie können die Instanz von `IANet_NetService` nicht löschen.

### Ändern von Eigenschaften

Für diese Klasse sind keine durch den Benutzer veränderbaren Eigenschaften vorhanden.

### Unterstützte Attribute

Durch diese Klasse werden zwei Attribute implementiert:

- `Version` – Enthält die aktuelle Version des Kern-Providers.
- `InstallDate` – Enthält das Datum, an dem die Provider installiert wurden.

### Methoden

Die folgenden Methoden können zur Verwaltung von Sitzungen verwendet werden:

- **`void GetSessionHandle([OUT] string SessionHandle, [out] uint32 ActiveSessions)`** – Dient zum Einstellen einer Handle-Zeichenfolge für die Sitzung. Diese Zeichenfolge sollte im Kontextobjekt im `SessionHandle`-Qualifizierungsmerkmal platziert werden. `ActiveSessions` gibt die Anzahl der aktiven Sitzungen für dieses System zurück. Dadurch ist es möglich, Clients davor zu warnen, dass andere Benutzer eventuell die Netzwerkeinstellungen ändern.
- **`void Apply([IN] string sSessionHandle, [OUT] uint32 FollowupAction)`**; – Wendet Änderungen an, die mit einem speziellen Sitzungshandle durchgeführt wurden. Das zurückgegebene `uint32`-Argument wird von den WMI und CDM Providern verwendet, um der Anwendung mitzuteilen, dass ein Serverneustart erforderlich ist, bevor die Änderungen wirksam werden. Dies wird durch Aufrufen der **`Reboot`**-Methode (Neustart) für die Klasse `Win32_OperatingSystem` erreicht.

Werte:

1 = Systemneustart erforderlich

0 = Kein Neustart erforderlich

- **`void ReleaseSessionHandle ([IN] string SessionHandle)`** – Gibt ein Sitzungshandle nach der Verwendung frei. Alle Änderungen, die bei dieser Sitzung durchgeführt wurden, gehen verloren. Das Sitzungshandle ist nach diesem Aufruf nicht mehr gültig und kann nicht mehr verwendet werden.
- **`void Cancel([IN] string SessionHandle)`**; – Bricht eine Sitzung ab. Der interne Cache-Speicher wird gelöscht, und alle nach diesem Aufruf gelesenen Daten zeigen die aktuelle Konfiguration.

[Zurück zum Seitenanfang](#)

---

## Kernereignisse

### IANet\_SessionEvent

#### Zweck

Dieses Ereignis wird verwendet, um den Client über die Verwendung der NCS-Sitzungs-API zu informieren. Clients können dieses Ereignis verwenden, um darüber informiert zu werden, wenn andere Clients Sitzungen erstellen oder verwenden.

#### Auslöser

Dieses Ereignis wird ausgelöst, wenn ein Client eine Sitzung erstellt, löscht oder **`Apply`** (Übernehmen) für eine Sitzung aufruft.

#### Ereignisdaten

Als **`EventType`** (Ereignistyp) kommt einer der folgenden Werte infrage:

- "New session" (Neue Sitzung) weist darauf hin, dass eine neue Sitzung durch den Client oder einen anderen Client erstellt wurde.
- "End session" (Ende der Sitzung) weist darauf hin, dass ein Client eine Sitzung beendet hat. Die Sitzung kann vom Client oder einem anderen Client beendet worden sein.
- "Cache invalidated" (Cache ungültig) weist darauf hin, dass ein anderer Client den Aufruf **`Apply`** (Übernehmen) auf eine Sitzung angewendet hat. Alle anderen Sitzungen verlieren dadurch ihre Gültigkeit und der mit ihren Sitzungen assoziierte Cache wird gelöscht.
- "Configuration changed" (Konfiguration geändert) weist darauf hin, dass die Konfiguration der Sitzung sich geändert hat.

Als **`SessionHandle`** (Sitzungshandle) wird das Sitzungshandle angezeigt, das das Ereignis ausgelöst hat.

**OpenSessions** (Offene Sitzungen) enthält die Anzahl der geöffneten Sitzungen. Dieses Datenelement ist NULL für die Ereignisse "Cache invalidated" (Cache ungültig) und "Configuration changed" (Konfiguration geändert).

## IANet\_InternalErrorEvent

### Zweck

Dieses Ereignis wird verwendet, um den Client darüber zu informieren, dass in den Ereignis-Providern ein interner Fehler aufgetreten ist. In manchen Fällen bedeutet dies, dass der Ereignis-Provider nicht in der Lage ist, weitere Ereignisse zu melden.

### Auslöser

Dieses Ereignis tritt in folgenden Situationen ein:

- Nachdem der Ereignis-Provider ein unbekanntes Ereignis von einer Ereignisquelle erhält
- Nachdem die Software, die die Ereignisse bereitstellt, beendet wurde
- Nachdem der Ereignis-Provider ein Ereignis erhalten hat, zu dem die Ereignisquelle keine weiteren Daten ermitteln kann

### Ereignisdaten

Als **EventType** (Ereignistyp) kommt eine der folgenden Angaben infrage:

- "Could not get event data" (Konnte Ereignisdaten nicht ermitteln). Ein Ereignis ist eingetreten, aber die Ereignisquelle kann keine weiteren Daten zum Ereignis ermitteln.
- "Event source has shut down" (Ereignisquelle wurde geschlossen). Die Datenquelle für das Ereignis wurde geschlossen. In diesem Fall wird der Ereignis-Provider ebenfalls geschlossen, und weitere Ereignisse werden erst dann wieder generiert, wenn die Quelle neu gestartet wurde und neue Benachrichtigungsabfragen durchgeführt wurden.
- "Unexpected message" (Unerwartete Meldung). Der Ereignis-Provider hat ein unerwartetes Ereignis erhalten.

[Zurück zum Seitenanfang](#)

---

## Verwendungsfälle

Zur Änderung der Konfiguration ist ein Sitzungshandle erforderlich. Mit dem Sitzungshandle ist die NCS-Software in der Lage, den gleichzeitigen mehrfachen Zugang zur Konfiguration zu verwalten. Dadurch wird verhindert, dass eine Sitzung alle anderen Sitzungen "aussperrt". Jede Sitzung verwendet einen anderen Cache, um die durchgeführten Änderungen zu speichern. Wenn von mehreren Sitzungen gleichzeitig Änderungen durchgeführt werden, ist die Sitzung erfolgreich, deren Änderungen zuerst übernommen werden. Alle anderen Sitzungs-Caches werden dadurch ungültig.

### Ermitteln eines Sitzungshandle

Der Client muss vor dem Zugriff auf das Sitzungshandle den Objektpfad der Einzelinstanz von IANet\_NetService ermitteln. Rufen Sie **IWbemServices::CreateInstanceEnum** auf, und geben Sie den Namen der Klasse weiter: IANet\_NetService. Dies entspricht dem Aufrufen von **IWbemServices::ExecQuery** mit der Abfrage **SELECT \* FROM IANet\_NetService**. Bevor Änderungen durchgeführt werden können, muss der Client ein Sitzungshandle ermitteln. Verwenden Sie die Methode **GetSessionHandle**, um eine neue Sitzung zu starten.

Der Client kann **IWbemServices::ExecMethod** verwenden, um eine Methode auf ein CIM-Objekt anzuwenden. Er benötigt dazu den Objektpfad aus dem **\_\_PATH**-Attribut der IANet\_NetService-Instanz. Bei dieser Methode wird auch die Anzahl der zurzeit aktiven Sitzungen zurückgegeben. Der Client gibt unter Umständen eine Warnung aus, wenn er keinen exklusiven Zugriff auf den Netzwerkkonfigurationsservice (NCS) hat.

### Verwenden eines Sitzungshandle im IWbemContext-Objekt

Nachdem der Client ein Sitzungshandle erhalten hat, muss er ein IWbemContext-Objekt erstellen. Das Sitzungshandle sollte im **SessionHandle**-Qualifizierungsmerkmal dieses Objekts gespeichert werden. Ein Verweis auf dieses COM-Objekt sollte an jeden an IWbemServices gerichteten Aufruf übertragen werden. Das Sitzungshandle ist nicht erforderlich, wenn Aufrufe für den Zugriff auf das IANet\_NetService-Objekt ausgeführt werden, da dabei das Handle als explizites Argument verwendet wird.

### Lesen von anstehenden Änderungen mithilfe eines Sitzungshandle

Wenn beim Lesen der Konfiguration das Sitzungshandle in den Kontext übertragen wird, geben die Provider die Konfiguration so zurück, als ob die anstehenden Aktualisierungen bereits übernommen worden wären (es fehlen also z. B. deinstallierte Adapter und für geänderte Einstellungen werden die neuen Werte angezeigt). Manche Objekte werden jedoch erst dann angezeigt, wenn der Aufruf **Apply** (Übernehmen) angewendet wird (IANet\_IPProtocolEndpoints werden z.B. erst dann erstellt, wenn das Protokoll an den entsprechenden Miniport angebunden wurde).

## Beenden mit einem Sitzungshandle

Nach dem Ändern der Konfiguration müssen die Änderungen mithilfe der Methode **Apply** (Übernehmen) angewendet werden, damit sie wirksam werden. Dabei wird unter Umständen ein Code für eine Folgeaktion ausgegeben (z.B. eine Aufforderung zum Neustart, damit die Änderungen wirksam werden).

Rufen Sie nach Abschluss einer Sitzung immer den Handle **ReleaseSessionHandle** auf. Andernfalls werden die vorgenommenen Änderungen verworfen. Durch Aufrufen der Methode **Cancel** (Abbrechen) werden die vorgenommenen Änderungen ebenfalls verworfen. Der Client kann das Sitzungshandle jedoch so weiterverwenden, als ob es gerade erstellt worden wäre.

## Registrieren für Kernereignisse

Anwendungen sollten **IWbemServices::ExecNotificationQuery** oder **IWbemServices:: ExecNotificationQueryAsync** verwenden, um eine Ereignisbenachrichtigung anzufordern. Die folgenden Abfragen sind Beispiele für Ereignisbenachrichtigungsabfragen (diese Liste ist unvollständig, da eine Vielzahl von Anfragen möglich ist):

- **SELECT \* FROM IANet\_Event** – Fordert alle Ereignisse an.
- **SELECT \* FROM IANet\_SessionEvent** – Fordert alle Sitzungsereignisse an.
- **SELECT \* FROM IANet\_InternalErrorEvent** – Fordert alle internen Ereignisse an.

---

Lesen Sie bitte alle [Einschränkungen und Haftungsausschlüsse](#).

---

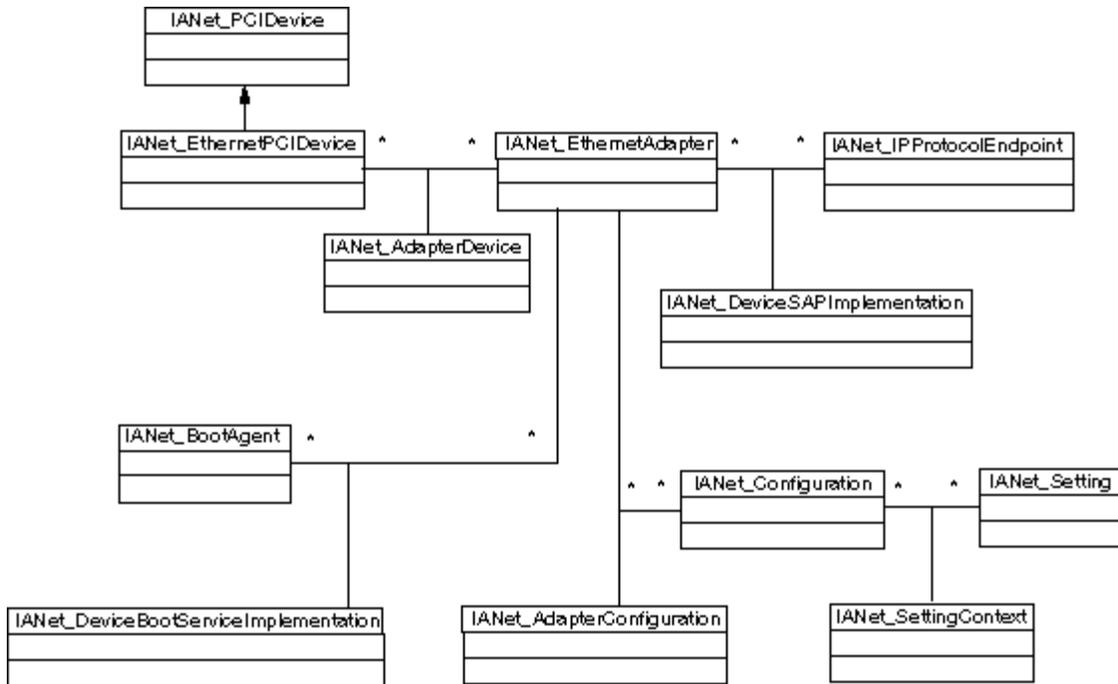
[Zurück zur Inhaltsseite](#)   [Zurück zum Seitenanfang](#)

# Das Ethernet-Adapterschema: Benutzerhandbuch für Intel® PRO Netzwerkadapter – WMI und CDM Provider

- [Überblick](#)
- [IANet\\_EthernetAdapter](#)
- [IANet\\_IPProtocolEndpoint](#)
- [IANet\\_BootAgent](#)
- [IANet\\_PCIDevice](#)
- [IANet\\_EthernetPCIDevice](#)

## Überblick

Das Adapterschema dient als Modell für die verschiedenen konfigurierbaren Intel® PROSet Ethernet-Adapter. Dieses Schema beruht auf dem CIM v2.5-Schema.



## IANet\_EthernetAdapter

### Zweck

Durch IANet\_EthernetAdapter werden die Fähigkeiten und der Status aller installierten Intel PRO-Netzwerkadapter sowie anderer Adapter definiert, die mit Intel Zwischentreibern in eine Gruppe eingebunden werden können. Die Klasse ist von der CIM\_EthernetAdapter-Überklasse abgeleitet, die in CIMv2.5 definiert ist. CIM\_EthernetAdapter ist von CIM\_NetworkAdapter abgeleitet, einer abstrakten Klasse, die allgemeine Netzwerk-Hardwarekonzepte definiert, wie z. B. PermanentAddress, CurrentAddress, Betriebsgeschwindigkeit usw.

### Instanzen

Instanzen dieser Klasse sind vorhanden für:

- Unterstützte und installierte Intel NICs
- NICs von Drittherstellern, die an einer Intel Multi-Vendor-Gruppe beteiligt sein können
- Erstellte Intel Adaptergruppen

### Erstellen von Instanzen

Sie können keine Instanzen von IANet\_EthernetAdapter erstellen.

### Entfernen von Instanzen

Durch Löschen einer Instanz von IANet\_EthernetAdapter werden reale Adapter deinstalliert. Nur nicht-virtuelle Intel Adapter dürfen auf diese Weise deinstalliert werden. Für diesen Vorgang ist ein Sitzungshandle erforderlich.

### Ändern von Eigenschaften

Für diese Klasse sind keine durch den Benutzer veränderbaren Eigenschaften vorhanden.

### Nicht unterstützte Attribute

Die folgenden Attribute sind für Intel PROSet nicht erforderlich und werden deshalb nicht unterstützt:

- AutoSense (dies ist als Einstellung erkennbar)
- ErrorCleared
- OtherIdentifyingInfo
- IdentifyingDescriptions
- InstallDate
- LastErrorCode
- MaxDataSize
- MaxQuiesceTime
- PowerManagementCapabilities (dies ist als Methode erkennbar)
- PowerManagementSupported (dies ist als Methode erkennbar)
- PowerOnHours
- ShortFramesReceived
- SymbolErrors
- TotalPowerOnHours

### Methoden

Diese Klasseninstanz unterstützt die folgenden Methoden:

- **IdentifyAdapter** – Identifiziert Adapter durch mehrsekündiges Blinken des Lämpchens am Adapter. Diese Methode ist nur bei realen Adaptern anwendbar.
- **HasVLANs** – Gibt die Anzahl der VLANs auf diesem Adapter zurück.
- **IsPowerMgmtSupported** – Zeigt an, ob die Energieverwaltung auf dem Adapter unterstützt wird.
- **GetPowerUsage** – Ermittelt den gesamten Energieverbrauch des Adapters.  
0 = Normaler Energieverbrauch  
1 = Niedriger Energieverbrauch
- **SetPowerUsage** – Reduziert den gesamten Energieverbrauch des Adapters. Die Energieverbrauchseinstellung geht beim Systemneustart oder beim Neuladen der Treiber verloren. Wenn das System neu gestartet oder der Treiber neu geladen wird, kehrt der Adapter automatisch zur Einstellung für normalen Energieverbrauch zurück.
- **GetPowerUsageOptions** – Ermittelt optionale Energieverbrauchseinstellungen (z. B. Energieverbrauchseinstellungen für Standby-Modus, Batteriebetrieb usw.).
- **SetPowerUsageOptions** – Ändert optionale Energieverbrauchseinstellungen (diese Methode kann z. B. zur Verringerung des Energieverbrauchs im Standby-Modus, Batteriebetrieb usw. verwendet werden).  
**Hinweis:** Die Energieverbrauchseinstellungen werden gespeichert und bei nachfolgenden Neustarts aktiviert.
- **TestCable** – Führt einen Diagnosetest für einen bestimmten Adapter aus. Bei nicht bestandenem Test werden bei dieser Methode mögliche Probleme, Ursachen und Lösungen angezeigt.
- **AdvancedTestCable** – Führt einen erweiterten Kabeltest für einen bestimmten Adapter aus. Diese Testsuite steht für 1000 Mbit/s-Adapter zur Verfügung. Durch diese Methode werden die Namen der Tests und die jeweiligen Ergebnisse zurückgegeben.  
**Hinweis:** Ein Verbindungsfehler kann auftreten, wenn **SpeedDuplex** nicht auf **Auto Negotiate** (Automatische Absprache) eingestellt ist. In diesem Fall ist der **SpeedAndDuplexNotAutomatic**-Out-Parameter TRUE (WAHR).
- **TestLinkSpeed** – Stellt fest, ob der Adapter mit voller Geschwindigkeit arbeitet. Falls der Adapter weniger als 1 Gigabit meldet, werden durch diese Methode mögliche Gründe angegeben (z. B. "Link partner is not capable of running at 1000 Mbps" (Verknüpfungspartner ist nicht in der Lage, mit 1000 Mbit/s zu arbeiten)).

---

# IANet\_IPProtocolEndpoint

## Zweck

Diese Klasse wird dazu verwendet, die IP-Einstellungen für einen Protokollendpunkt im System zu beschreiben. Der WMI Provider stellt keine Informationen für andere Netzwerkprotokollarten bereit. Die Klasse ist von der abstrakten Klasse CIM\_IPProtocolEndpoint abgeleitet. Der WMI Provider liefert nur Protokollinformationen, wenn sie eine von Intel PROSet verwaltete Einheit betreffen.

## Instanzen

Eine IANet\_IPProtocolEndpoint-Instanz existiert für jede Bindung des IP-Protokollstapels an einen von Intel unterstützten Endpunkt (z. B. Intel Adapter, gruppenfähige Intel Adapter und VLANs). Einige in Gruppen integrierte Adapter verfügen über keine eigene IP-Adresse. Aus diesem Grund ist kein IANet\_IPProtocolEndpoint direkt mit ihrer Adapterinstanz assoziiert. Der IANet\_IPProtocolEndpoint existiert erst dann, wenn das Betriebssystem das Protokoll an den Adapter oder das VLAN angebunden hat. Es kann zwar vorkommen, dass Adapter mehrere IP-Adressen haben, sie sind jedoch alle nur mit einer Instanz eines IP-Protokollendpunkts assoziiert. Der Provider unterstützt diese erweiterte Verwendungsweise nicht, da sie nicht erforderlich ist bzw. von Intel PROSet nicht verwendet wird.

## Erstellen von Instanzen

Sie können keine Instanzen von IANet\_IPProtocolEndpoint erstellen. Die Instanz existiert nur dann, wenn das Betriebssystem das Protokoll an den Endpunkt angebunden hat.

## Entfernen von Instanzen

Sie können keine Instanzen von IANet\_IPProtocolEndpoint löschen.

## Ändern von Eigenschaften

Für diese Klasse sind keine durch den Benutzer veränderbaren Eigenschaften vorhanden.

## Assoziationen

Eine Instanz von IANet\_AdapterProtocolImplementation wird verwendet, um einen IANet\_EthernetAdapter mit einem IANet\_IPProtocolEndpoint zu assoziieren. Eine Instanz von IANet\_VLANProtocolDependency wird verwendet, um ein VLAN mit einem IANet\_IPProtocolEndpoint zu assoziieren.

**Hinweis:** Gruppen sind mit dem Endpunkt über den Adapter assoziiert, der als virtueller Adapter für die Gruppe fungiert.

## Unterstützte Attribute

Die folgenden schreibgeschützten Attribute sind notwendig für Intel PROSet:

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

## Nicht unterstützte Attribute

Die folgenden Attribute sind für Intel PROSet nicht erforderlich und werden deshalb nicht unterstützt:

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

## Methoden

Keine

[Zurück zum Seitenanfang](#)

# IANet\_BootAgent

## Zweck

Diese Klasse wird zum Erfassen von Informationen über die Netzwerk-Startfähigkeiten eines Adapters (z. B. Einstellungen für den von einigen Intel Adaptern unterstützten PXE Boot Agent) verwendet. Diese Klasse ist von CIM\_BootService abgeleitet.

## Instanzen

Eine IANet\_BootAgent-Instanz ist für jeden Adapter vorhanden, der die Boot Agent-Fähigkeit unterstützt, und zwar auch dann, wenn der Boot Agent derzeit nicht installiert ist.

## Erstellen von Instanzen

Sie können keine Instanzen von IANet\_BootAgent erstellen. Eine Instanz existiert nur dann, wenn der Adapter die Boot Agent-Fähigkeit unterstützt.

## Entfernen von Instanzen

Sie können keine Instanzen von IANet\_BootAgent löschen.

## Ändern von Eigenschaften

Für diese Klasse sind keine durch den Benutzer veränderbaren Eigenschaften vorhanden.

## Assoziationen

Eine Instanz von IANet\_DeviceBootServiceImplementation wird verwendet, um einen IANet\_EthernetAdapter mit einem IANet\_BootAgent zu assoziieren, sofern der Adapter dies unterstützt.

## Unterstützte Attribute

Die folgenden schreibgeschützten Attribute sind notwendig für Intel PROSet:

- InvalidImageSignature
- Version
- UpdateAvailable
- FlashImageType

## Nicht unterstützte Attribute

Die folgenden Attribute sind für Intel PROSet nicht erforderlich und werden deshalb nicht unterstützt:

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

## Methoden

Die folgenden Methoden für diese Klasse können verwendet werden, um den Flash-ROM-Speicher auf dem NIC zu aktualisieren:

<pre>uint32 ProgramFlash(     [IN,     ValueMap {"0","1"} ,     Values {"Check Version", "Write Flash"}: Amended     ]     uint32 Action,     [IN]     uint8 NewFlashData[],     [OUT]     string</pre>	<p>Diese Methode wird zur Aktualisierung des Flash-ROM-Speichers auf dem NIC verwendet. Während der Aktualisierung des Flash-Speichers unterbricht der NIC die Kommunikation mit dem Netzwerk.</p>
---	--

strErrorMessage );	
uint32 ReadFlash( [OUT] uint8 FlashData[] );	Diese Methode dient zum Lesen des Flash-ROM-Speichers auf dem NIC.

[Zurück zum Seitenanfang](#)

## IANet\_PCIDevice

### Zweck

Diese Klasse wird dazu verwendet, die Eigenschaften des PCI-Geräts für ein Netzwerkgerät im System zu beschreiben. Diese Klasse ist von CIM\_PCIDevice abgeleitet.

### Instanzen

Eine Instanz dieser Klasse ist für jede PCI-Karte vorhanden, die als Netzwerkgerät im System eingesetzt wird. Bei IA64 weisen nur PCI-Geräte Instanzen auf, die von Intel PROSet unterstützte Adapter darstellen.

### Erstellen von Instanzen

Sie können keine Instanzen von IANet\_PCIDevice erstellen.

### Löschen von Instanzen

Sie können keine Instanzen von IANet\_PCIDevice löschen.

### Ändern von Eigenschaften

Für diese Klasse sind keine durch den Benutzer veränderbaren Eigenschaften vorhanden.

### Assoziationen

Informationen zu den Klassenassoziationen finden Sie unter IANet\_EthernetPCIDevice.

### Methoden

Für diese Klasse sind keine unterstützten Methoden vorhanden.

### Nicht unterstützte Attribute

Die folgenden Attribute werden vom WMI Provider nicht unterstützt:

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

[Zurück zum Seitenanfang](#)

---

## IANet\_EthernetPCIDevice

### Zweck

Diese Klasse wird dazu verwendet, die Eigenschaften eines PCI-Geräts für einen von Intel PROSet unterstützten Ethernet-Adapter zu beschreiben. Es handelt sich dabei um eine Unterklasse von IANet\_PCIDevice. Die Klasse enthält einige zusätzliche Attribute, die nur für von Intel PROSet unterstützte PCI-Geräte bekannt sind.

### Instanzen

Eine Instanz dieser Klasse ist für jede PCI-Karte vorhanden, die als von Intel PROSet unterstützter Ethernet-Adapter fungiert.

### Erstellen von Instanzen

Sie können keine Instanzen von IANet\_EthernetPCIDevice erstellen.

### Entfernen von Instanzen

Sie können keine Instanzen von IANet\_EthernetPCIDevice löschen.

### Ändern von Eigenschaften

Für diese Klasse sind keine durch den Benutzer veränderbaren Eigenschaften vorhanden.

### Assoziationen

Eine IANet\_AdapterDevice-Instanz wird verwendet, um IANet\_PCIDevice mit IANet\_EthernetAdapter zu assoziieren. Virtuelle Adapter (z. B. Adapter, die als Repräsentanten von Gruppen erstellt werden) sind nicht mit IANet\_PCIDevice assoziiert.

### Nicht unterstützte Attribute

Die folgenden Attribute werden vom WMI Provider nicht unterstützt:

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

### Methoden

Für diese Klasse sind keine unterstützten Methoden vorhanden.

---

Lesen Sie bitte alle [Einschränkungen und Haftungsausschlüsse](#).

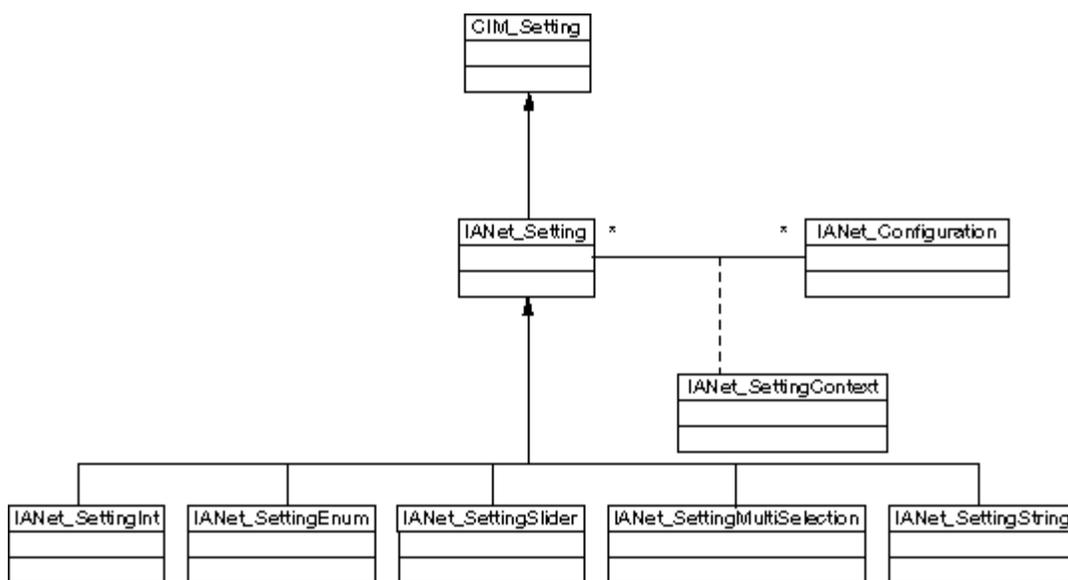
---

[Zurück zur Inhaltsseite](#)   [Zurück zum Seitenanfang](#)

# Das Einstellungsschema: Benutzerhandbuch für Intel® PRO Netzwerkkadpter – WMI und CDM Provider

- [Überblick](#)
- [IANet\\_Configuration](#)
- [IANet\\_Setting](#)
- [IANet\\_SettingInt](#)
- [IANet\\_SettingEnum](#)
- [IANet\\_SettingSlider](#)
- [IANet\\_SettingMultiSelection](#)
- [IANet\\_SettingString](#)

## Überblick



## IANet\_Configuration

### Zweck

Diese Klasse dient dazu, eine Sammlung von `IANet_Setting`-Instanzen zu einer Gruppe zusammenzufassen. Diese Klasse ist von `CIM_Configuration` abgeleitet.

### Instanzen

Jeder Adapter, jedes VLAN und jede Gruppe können mehrere assoziierte `IANet_Configuration`-Instanzen aufweisen (jede Konfiguration entspricht einem anderen Verwendungsszenario für den Adapter).

Bei dieser Ausgabe der WMI und CDM Provider gibt es nur jeweils eine `IANet_Configuration`-Instanz pro Adapter, VLAN oder Gruppe.

### Erstellen von Instanzen

Sie können keine Instanzen von `IANet_Configuration` erstellen.

### Entfernen von Instanzen

Sie können keine Instanzen von IANet\_Configuration löschen.

### Ändern von Eigenschaften

Für diese Klasse sind keine durch den Benutzer veränderbaren Eigenschaften vorhanden.

### Assoziationen

Jeweils eine IANet\_AdapterConfiguration-Instanz ist vorhanden, um jeden Adapter (IANet\_EthernetAdapter) mit seiner Konfiguration zu assoziieren. Jeweils eine IANet\_VLANConfiguration-Instanz ist vorhanden, um jedes VLAN (IANet\_VLAN) mit seiner Konfiguration zu assoziieren. Jeweils eine IANet\_BootAgentConfiguration-Instanz ist vorhanden, um jeden Boot Agent (IANet\_BootAgent) mit seiner Konfiguration zu assoziieren.

### Methoden

Für diese Klasse sind keine unterstützten Methoden vorhanden.

### Nicht unterstützte Attribute

Keine

[Zurück zum Seitenanfang](#)

---

## IANet\_Setting

### Zweck

Diese abstrakte Klasse wird verwendet, um eine einstellbare Eigenschaft in einer Konfiguration zu beschreiben. Diese Klasse ist von CIM\_Setting abgeleitet.

### Instanzen

Eine separate Instanz dieser Klasse ist für jede Einstellung auf jedem Adapter, VLAN bzw. jeder Gruppe vorhanden. Die Einstellungen werden nicht von verschiedenen Konfigurationen gemeinsam verwendet.

Es gibt mehrere Unterklassen von IANet\_Setting. Die Unterklassen entsprechen den verschiedenen Typen und Wertebereichen, die für Einstellungen infrage kommen. Jede Unterklasse entspricht einem anderen GUI-Stil für das Anzeigen oder Ändern der Einstellungen.

### Erstellen von Instanzen

Sie können keine Instanzen von IANet\_Setting erstellen.

### Entfernen von Instanzen

Sie können keine Instanzen von IANet\_Setting löschen.

### Ändern von Eigenschaften

Diese abstrakte Klasse besitzt keine veränderbaren Eigenschaften; die Eigenschaften der untergeordneten Klassen sind jedoch veränderbar (siehe unten).

### Assoziationen

Jede IANet\_Setting-Instanz ist mittels einer IANet\_SettingContext-Instanz mit einer IANet\_Configuration-Instanz assoziiert.

### Methoden

Für diese Klasse sind keine unterstützten Methoden vorhanden. Wenn Sie Änderungen an einer Einstellung durchführen möchten, ändern Sie die erforderliche Eigenschaft und wenden Sie **PutInstance** an.

### Nicht unterstützte Attribute

SettingID wird nicht verwendet.

[Zurück zum Seitenanfang](#)

---

## IANet\_SettingInt

### Zweck

Diese Klasse modelliert eine Einstellung, die einen Ganzzahlwert aufnimmt. Zur Modellierung von Ganzzahlen werden mehrere

IANet-Einstellungsklassen verwendet. Die Unterschiede zwischen diesen Klassen stehen in engem Zusammenhang mit der Art und Weise, wie die GUI die Ganzzahl anzeigt und verändert und wie die Provider die Validierung durchführen. Bei IANet\_SettingInt wird erwartet, dass die GUI ein Bearbeitungsfeld mit Einstellregler anzeigt.

### Instanzen

Eine Instanz dieser Klasse ist für jede Einstellung vorhanden, die als Ganzzahl-Bearbeitungsfeld angezeigt werden sollte.

### Erstellen von Instanzen

Sie können keine Instanzen dieser Klasse erstellen.

### Entfernen von Instanzen

Sie können keine Instanzen dieser Klasse löschen.

### Ändern von Eigenschaften

Die einzige veränderbare Eigenschaft dieser Klasse ist das Attribut "CurrentValue". Sie können diese Eigenschaft ändern, indem Sie **IWbemClassObject::Put()** zum Ändern des Werts verwenden und dann **IWbemServices::PutInstance()** aufrufen, um die Einstellung zu aktualisieren. Die Provider überprüfen daraufhin Folgendes:

**CurrentValue** <= **max**

**CurrentValue** >= **min**

(**CurrentValue** - **min**) ist ein Vielfaches von **Step**

**max**, **min**, **CurrentValue** (Aktueller Wert) und **Step** (Schritt) sind Attribute von IANet\_SettingInt.

### Assoziationen

Jede IANet\_SettingInt-Instanz ist mittels einer IANet\_SettingContext-Instanz mit einer IANet\_Configuration-Instanz assoziiert.

### Nicht unterstützte Attribute

SettingID wird nicht verwendet.

### Methoden

Für diese Klasse sind keine unterstützten Methoden vorhanden. Wenn Sie Änderungen an einer Einstellung durchführen möchten, ändern Sie die erforderliche Eigenschaft und wenden Sie **PutInstance** an.

[Zurück zum Seitenanfang](#)

---

## IANet\_SettingEnum

### Zweck

Diese Klasse modelliert eine Einstellung, die einen Ganzzahlwert aufnimmt. Zur Modellierung von Ganzzahlen werden mehrere IANet-Einstellungsklassen verwendet. Die Unterschiede zwischen diesen Klassen stehen in engem Zusammenhang mit der Art und Weise, wie die GUI die Ganzzahl anzeigt und verändert und wie die Provider die Validierung durchführen. Bei IANet\_SettingEnum wird erwartet, dass die GUI eine Liste der Zeichenfolgen anzeigt, die einer kleinen Anzahl aufgelisteter Werte zugeordnet sind (z. B. eine Einblendliste in einem Kombinationsfeld).

### Instanzen

Eine Instanz dieser Klasse ist für jede Einstellung vorhanden, die als Auflistung angezeigt wird.

### Erstellen von Instanzen

Sie können keine Instanzen dieser Klasse erstellen.

### Entfernen von Instanzen

Sie können keine Instanzen dieser Klasse löschen.

### Ändern von Eigenschaften

Die einzige veränderbare Eigenschaft dieser Klasse ist das Attribut **CurrentValue**. Sie können diese Eigenschaft ändern, indem Sie **Put()** zum Ändern des Werts verwenden und dann **PutInstance()** aufrufen, um die Einstellung zu aktualisieren. Die Provider überprüfen daraufhin Folgendes: **CurrentValue** & **PossibleValues[]**

### Assoziationen

Jede IANet\_SettingEnum-Instanz ist mittels einer IANet\_SettingContext-Instanz mit einer IANet\_Configuration-Instanz assoziiert.

### Nicht unterstützte Attribute

SettingID wird nicht verwendet.

### Methoden

Für diese Klasse sind keine unterstützten Methoden vorhanden. Wenn Sie Änderungen an einer Einstellung durchführen möchten, ändern Sie die erforderliche Eigenschaft und wenden Sie **PutInstance** an.

[Zurück zum Seitenanfang](#)

---

## IANet\_SettingSlider

### Zweck

Diese Klasse modelliert eine Einstellung, die einen Ganzzahlwert aufnimmt. Zur Modellierung von Ganzzahlen werden mehrere IANet-Einstellungsklassen verwendet. Die Unterschiede zwischen diesen Klassen stehen in engem Zusammenhang mit der Art und Weise, wie die GUI die Ganzzahl anzeigt und verändert und wie die Provider die Validierung durchführen. Bei IANet\_SettingSlider wird erwartet, dass die GUI einen Regler anzeigt, der eine grafische Methode zur Auswahl des Werts anbietet (der tatsächlich gewählte Wert muss dabei nicht angezeigt werden).

### Instanzen

Eine Instanz dieser Klasse ist für jede Einstellung vorhanden, die als Regler angezeigt wird.

### Erstellen von Instanzen

Sie können keine Instanzen dieser Klasse erstellen.

### Entfernen von Instanzen

Sie können keine Instanzen dieser Klasse löschen.

### Ändern von Eigenschaften

Die einzige veränderbare Eigenschaft dieser Klasse ist das Attribut **CurrentValue**. Sie können diese Eigenschaft ändern, indem Sie **Put()** zum Ändern des Werts verwenden und dann **PutInstance()** aufrufen, um die Einstellung zu aktualisieren. Die Provider überprüfen daraufhin Folgendes: **CurrentValue € PossibleValues[]**

### Assoziationen

Jede IANet\_SettingSlider-Instanz ist mittels einer IANet\_SettingContext-Instanz mit einer IANet\_Configuration-Instanz assoziiert.

### Nicht unterstützte Attribute

SettingID wird nicht verwendet.

### Methoden

Für diese Klasse sind keine unterstützten Methoden vorhanden. Wenn Sie Änderungen an einer Einstellung durchführen möchten, ändern Sie die erforderliche Eigenschaft und wenden Sie **PutInstance** an.

[Zurück zum Seitenanfang](#)

---

## IANet\_SettingMultiSelection

### Zweck

Diese Klasse modelliert eine Einstellung, für die Sie mehrere Optionen aus einer Liste mit Optionen auswählen können. Bei IANet\_SettingMultiSelection wird erwartet, dass die GUI ein Listenfeld mit der Möglichkeit zur mehrfachen Auswahl anzeigt, so dass Sie eine oder mehrere Optionen auswählen oder auf die Auswahl einer Option verzichten können.

### Instanzen

Eine Instanz dieser Klasse ist für jede Einstellung vorhanden, die als Mehrfachauswahl angezeigt wird.

### Erstellen von Instanzen

Sie können keine Instanzen dieser Klasse erstellen.

### Entfernen von Instanzen

Sie können keine Instanzen dieser Klasse löschen.

### Ändern von Eigenschaften

Die einzige veränderbare Eigenschaft dieser Klasse ist das Attribut **CurrentValue**. Sie können diese Eigenschaft ändern, indem Sie **Put()** zum Ändern des Werts verwenden und dann **PutInstance()** aufrufen, um die Einstellung zu aktualisieren. Die Provider überprüfen daraufhin Folgendes: **CurrentValue & PossibleValues[]**

### Assoziationen

Jede `IANet_SettingMultiSelection`-Instanz ist mittels einer `IANet_SettingContext`-Instanz mit einer `IANet_Configuration`-Instanz assoziiert.

### Nicht unterstützte Attribute

SettingID wird nicht verwendet.

### Methoden

Für diese Klasse sind keine unterstützten Methoden vorhanden. Wenn Sie Änderungen an einer Einstellung durchführen möchten, ändern Sie die erforderliche Eigenschaft und wenden Sie **PutInstance** an.

[Zurück zum Seitenanfang](#)

---

## IANet\_SettingString

### Zweck

Diese Klasse modelliert eine Einstellung, für die Sie in einem Bearbeitungsfeld eine Zeichenfolge beliebiger Form eingeben können. Bei `IANet_SettingMultiSelection` wird erwartet, dass die GUI ein Bearbeitungsfeld anzeigt.

### Instanzen

Eine Instanz dieser Klasse ist für jede Einstellung vorhanden, die als Bearbeitungsfeld angezeigt wird.

### Erstellen von Instanzen

Sie können keine Instanzen dieser Klasse erstellen.

### Entfernen von Instanzen

Sie können keine Instanzen dieser Klasse löschen.

### Ändern von Eigenschaften

Die einzige veränderbare Eigenschaft dieser Klasse ist das Attribut **CurrentValue**. Sie können diese Eigenschaft ändern, indem Sie **Put()** zum Ändern des Werts verwenden und dann **PutInstance()** aufrufen, um die Einstellung zu aktualisieren.

### Assoziationen

Jede `IANet_SettingMultiSelection`-Instanz ist mittels einer `IANet_SettingString`-Instanz mit einer `IANet_ElementConfiguration`-Instanz assoziiert.

### Methoden

Für diese Klasse sind keine unterstützten Methoden vorhanden.

### Nicht unterstützte Attribute

SettingID wird nicht verwendet.

### Methoden

Für diese Klasse sind keine unterstützten Methoden vorhanden. Wenn Sie Änderungen an einer Einstellung durchführen möchten, ändern Sie die erforderliche Eigenschaft und wenden Sie dann **PutInstance** an.

---

Lesen Sie bitte alle [Einschränkungen und Haftungsausschlüsse](#).

---

[Zurück zur Inhaltsseite](#)   [Zurück zum Seitenanfang](#)

# Das Gruppenschema: Benutzerhandbuch für Intel® PRO Netzwerkadapter – WMI und CDM Provider

[Überblick](#)

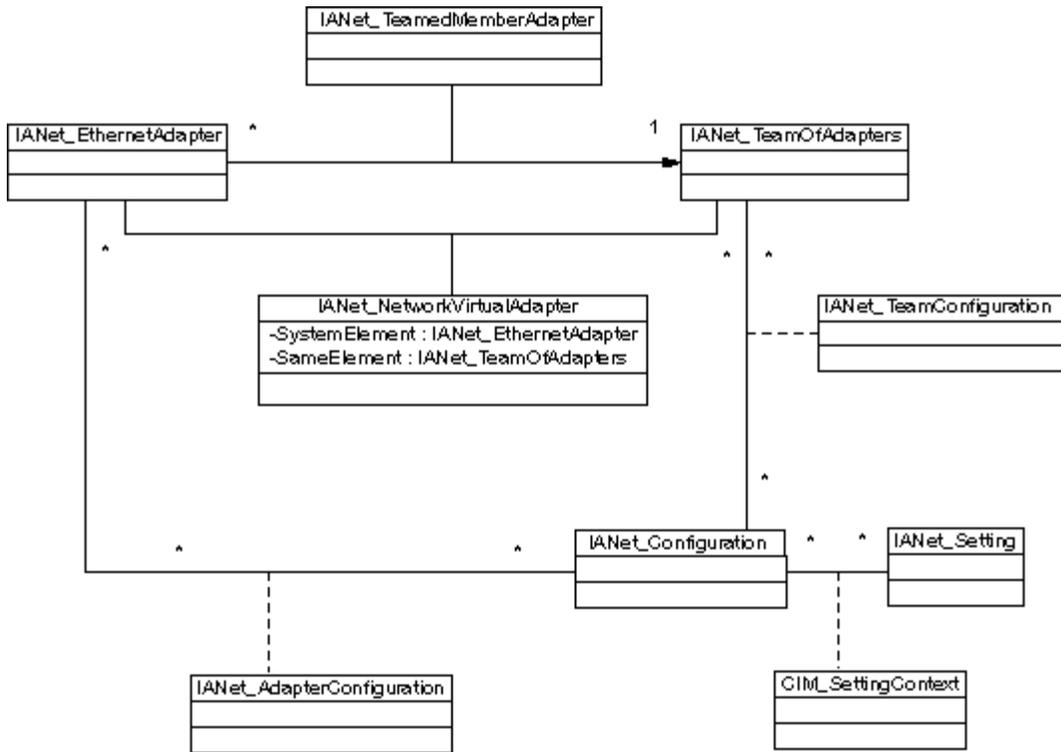
[IANet\\_TeamOfAdapters](#)

[IANet\\_TeamedMemberAdapter](#)

[IANet\\_NetworkVirtualAdapter](#)

## Überblick

Das Gruppenschema beschreibt, wie die Ethernet-Adapter in Gruppen zusammengefasst werden.



[Zurück zum Seitenanfang](#)

## IANet\_TeamOfAdapters

### Zweck

Mit dieser Klasse wird die CIM\_RedundancyGroup-Klasse implementiert. Diese Klasse weist Mitglieder auf, die den Gruppentyp, die Anzahl der Adapter in der Gruppe und die Höchstzahl der in der Gruppe möglichen Adapter beschreibt.

### Instanzen

Für jede Intel Gruppe ist eine Instanz dieser Klasse vorhanden.

### Erstellen von Instanzen

Erstellen Sie eine Instanz von IANet\_TeamOfAdapters, um eine leere Gruppe zu erstellen. Sie müssen die korrekte Einstellung für **TeamingMode** wählen, bevor Sie **IWbemServices::PutInstance()** aufrufen, um das Objekt in den Providern zu erstellen. Die Provider geben eine Zeichenfolge zurück, die den Objekt Pfad für das neue Objekt enthält.

### Entfernen von Instanzen

Zum Entfernen einer Gruppe müssen Sie auf entsprechende Weise die IANet\_TeamOfAdapters-Instanz entfernen. Die Provider löschen die Assoziationen mit den Gruppenmitgliedern und den virtuellen Adapter sowie die Einstellungen für die Gruppe.

### Ändern von Eigenschaften

Verwenden Sie **Put()**, um den Wert der **TeamingMode**-Eigenschaft zu ändern, und rufen Sie dann **PutInstance()** auf, um die Gruppe zu aktualisieren.

### Assoziationen

Jeder Adapter in einer Gruppe ist mittels einer Instanz von IANet\_TeamMemberAdapter mit der IANet\_TeamOfAdapters-Instanz der Gruppe assoziiert. Der virtuelle Adapter für die Gruppe ist mit dieser Klasse mittels einer Instanz von IA\_NetNetworkVirtualAdapter assoziiert.

### Methoden

Diese Klasseninstanz unterstützt die folgende Methode:

**TestSwitchConfiguration** – Testet dieSwitch-Konfiguration, um sicherzustellen, dass die Gruppe korrekt mit dem Switch funktioniert. Mit diesem Test kann überprüft werden, ob die Verbindungspartner (d. h. ein Gerät, mit dem ein Adapter verbunden ist, z. B. ein anderer Adapter, Hub, Switch usw.) den gewählten Adapter-Gruppenbildungsmodus unterstützen. Wenn der Adapter z. B. Mitglied einer Link-Aggregation-Gruppe ist, kann mit diesem Test sichergestellt werden, dass die mit dem Adapter verbundenen Verknüpfungspartner Link Aggregation unterstützen.

[Zurück zum Seitenanfang](#)

---

## IANet\_TeamedMemberAdapter

### Zweck

Diese Klasse wird verwendet, um den Adapter mit der Gruppe zu assoziieren. Sie bestimmt die Funktion des Adapters in der Gruppe und stellt sicher, dass der Adapter derzeit in der Gruppe aktiv ist. Diese Klasse implementiert die CIM-Klasse CIM\_NetworkAdapterRedundancyComponent.

### Instanzen

Eine Instanz dieser Klasse ist für jeden Adapter vorhanden, der ein Mitglied der Gruppe ist.

### Erstellen von Instanzen

Wenn Sie einen Adapter einer Gruppe hinzufügen möchten, erstellen Sie eine Instanz von IANet\_TeamedMemberAdapter, um den Adapter mit der Gruppe zu assoziieren.

### Entfernen von Instanzen

Wenn Sie einen Adapter aus der Gruppe entfernen möchten, entfernen Sie die IANet\_TeamedMemberAdapter-Instanz. Der Adapter gehört danach nicht mehr zur Gruppe und kann an einen IP-Protokollendpunkt angebunden werden, nachdem die Funktion **Apply()** (Übernehmen) aufgerufen wurde.

### Ändern von Eigenschaften

Die **AdapterFunction**-Eigenschaft dieser Klasse kann so abgeändert werden, dass sie beschreibt, wie der Adapter in der Gruppe verwendet wird.

### Assoziationen

Dies ist eine Assoziationsklasse.

### Methoden

Für diese Klasse sind keine unterstützten Methoden vorhanden.

[Zurück zum Seitenanfang](#)

---

## IANet\_NetworkVirtualAdapter

### Zweck

Diese Klasse wird verwendet, um IANet\_TeamOfAdapters des Teams mit dem IANet\_EthernetAdapter zu assoziieren, der den

virtuellen Adapter für die Gruppe darstellt. Diese Klasse implementiert die CIM-Klasse CIM\_NetworkVirtualAdapter.

### **Instanzen**

Eine Instanz dieser Klasse ist für jede Intel Gruppe vorhanden, die an einen virtuellen Adapter angebunden wurde.

### **Erstellen von Instanzen**

Sie können keine Instanzen dieser Klasse erstellen. Erstellen Sie eine Instanz von IANet\_TeamOfAdapters, um eine Gruppe zu erstellen. Diese Klasse ist erst dann vorhanden, wenn Sie **IANet\_NetService .Apply()** im Kontext der gültigen Sitzung aufgerufen haben und die IANet\_EthernetAdapter-Instanz erstellt wurde.

### **Entfernen von Instanzen**

Sie können keine Instanzen dieser Klasse löschen.

### **Assoziationen**

Dies ist eine Assoziationsklasse.

### **Methoden**

Für diese Klasse sind keine unterstützten Methoden vorhanden.

---

Lesen Sie bitte alle [Einschränkungen und Haftungsausschlüsse](#).

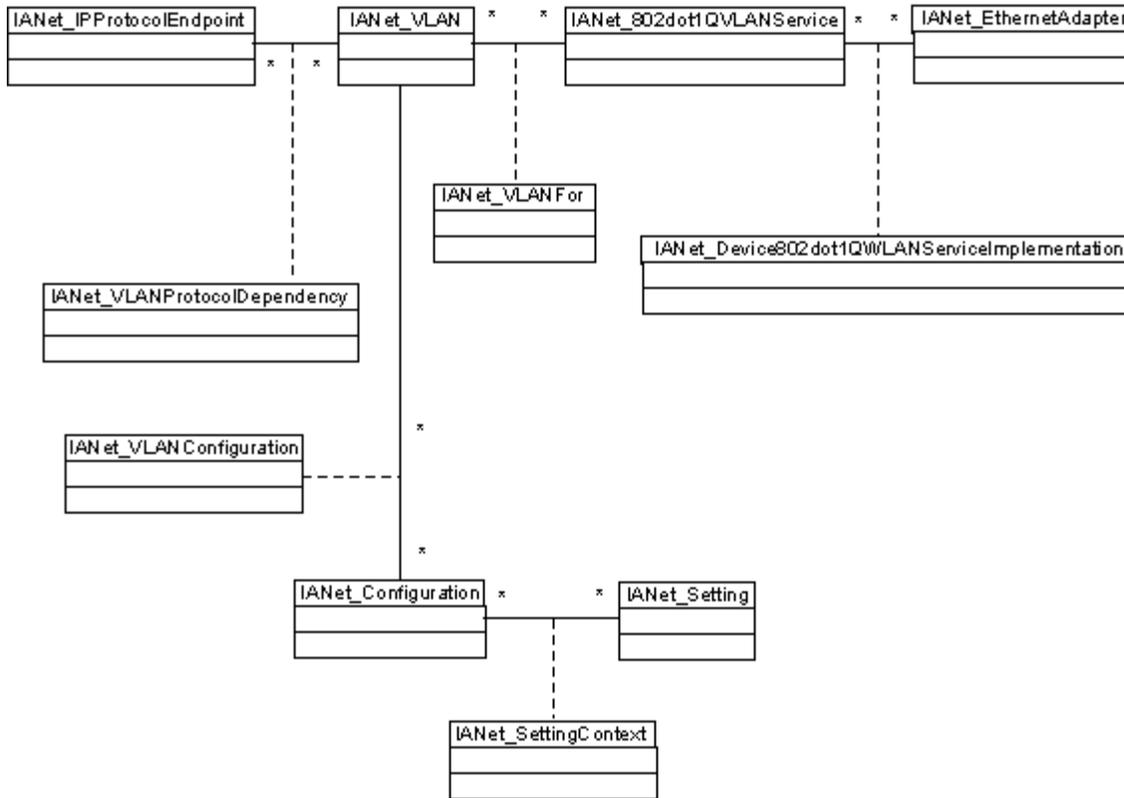
---

[Zurück zur Inhaltsseite](#)   [Zurück zum Seitenanfang](#)

# Das VLAN-Schema: Benutzerhandbuch für Intel® PRO Netzwerkkadpter – WMI und CDM Provider

[Überblick](#)  
[IANet\\_802dot1QVLANService](#)  
[IANet\\_VLAN](#)

## Überblick



## IANet\_802dot1QVLANService

### Zweck

Diese Klasse wird zum Festhalten der IEEE 802.1Q-Eigenschaften eines Netzwerkkadpters verwendet. Die Klasse implementiert die CIM-Klasse CIM\_802dot1QVLANService.

### Instanzen

Eine Instanz dieser Klasse ist für jeden Adapter bzw. jede Gruppe vorhanden, der bzw. die IEEE 802.1Q unterstützt. Jeder Adapter kann nur eine einzelne IANet\_802dot1QVLANService-Instanz haben. Manche Gruppen, wie z. B. fehlertolerante Multi-Vendor-Gruppen, unterstützen diesen Service nicht.

### Anomalie

Eine Gruppe ohne VLAN verfügt über keinen VLAN-Service, es sei denn, Sie listen ihn im Kontext einer gültigen Sitzung auf. Bei

Gruppen erscheint die 802.3QvlanService-Instanz nur in den folgenden Situationen:

- Wenn die Gruppe bereits über VLANs verfügt
- Wenn die Gruppe keine VLANs besitzt und Sie beim Aufzählen dieser Klasse ein Sitzungshandle im Kontext verwenden

### **Erstellen von Instanzen**

Sie können keine Instanzen dieser Klasse erstellen. Falls mit dem Adapter keine Instanz dieser Klasse assoziiert ist, unterstützt der Adapter diesen Service nicht.

### **Entfernen von Instanzen**

Sie können keine Instanzen dieser Klasse löschen.

### **Ändern von Eigenschaften**

Für diese Klasse sind keine veränderbaren Eigenschaften vorhanden.

### **Assoziationen**

Jede Instanz dieser Klasse wird mittels einer IANet\_DeviceServiceImplementation-Instanz mit einer IANet\_EthernetAdapter-Instanz assoziiert.

Jede Instanz von IANet\_802dot1QVLANService kann mehrere VLANs unterstützen; jedes VLAN ist mittels IANet\_VLANFor-Assoziation mit der Instanz assoziiert.

### **Methoden**

**uint16 CreateVLAN( [in] uint32 VLANNumber, [in] string Name, [out] IANet\_VLAN REF VLANpath );** – Wird zum Erstellen eines VLAN für den Adapter oder die Gruppe verwendet. Der Client muss die VLAN-Nummer und den VLAN-Namen bereitstellen, und er erhält den Objektpfad des neu erstellten VLAN.

[Zurück zum Seitenanfang](#)

---

## **IANet\_VLAN**

### **Zweck**

Diese Klasse enthält die Informationen für jedes Intel VLAN. Diese Klasse implementiert CIM\_VLAN.

### **Instanzen**

Für jedes Intel VLAN ist eine Instanz dieser Klasse vorhanden.

### **Erstellen von Instanzen**

Rufen Sie zum Erstellen eines VLAN **CreateVLAN** auf der zutreffenden IANet\_802dot1QVLANService-Instanz auf.

### **Entfernen von Instanzen**

Sie können eine Instanz dieser Klasse löschen, um das entsprechende VLAN zu entfernen.

### **Ändern von Eigenschaften**

Sie können das VLANNumber- und Caption-Attribut ändern.

### **Assoziationen**

Jede Instanz ist mit einer IANet\_802dot1QVLANService-Instanz und dadurch mithilfe der Klasse IANet\_VLANFor mit einer Instanz von IANet\_EthernetAdapter assoziiert.

Jede Instanz kann mit mehreren IANet\_Configuration-Instanzen assoziiert sein, wodurch eine Gruppe von Einstellungen für das VLAN zusammengefasst werden kann. In dieser Provider-Ausgabe ist nur ein IANet\_Configuration-Objekt pro VLAN vorhanden.

Jede Instanz kann mit einem IANet\_IPProtocolEndpoint assoziiert werden, um die IP-Einstellungen für das VLAN mithilfe der Klasse IANet\_VLANProtocolDependency bereitzustellen.

### **Methoden**

Keine

---

Lesen Sie bitte alle [Einschränkungen und Haftungsausschlüsse](#).

---

[Zurück zur Inhaltsseite](#) [Zurück zum Seitenanfang](#)

## Ermitteln der aktuellen Konfiguration: Benutzerhandbuch für Intel® PRO Netzwerkkadappter – WMI und CDM Provider

- [Ermitteln der realen Adapter](#)
- [Ermitteln der PCI-Geräte](#)
- [Ermitteln der Adaptereinstellungen](#)
- [Ermitteln der Gruppenkonfiguration](#)
- [Ermitteln der Gruppeneinstellungen](#)
- [Ermitteln der VLAN-Konfiguration](#)
- [Ermitteln der VLAN-Einstellungen](#)
- [Ermitteln der IP-Protokollinformationen](#)
- [Ermitteln der Boot Agent-Informationen](#)
- [Ermitteln der Boot Agent-Einstellungen](#)

Der Client benötigt kein Sitzungshandle, um die aktuelle Konfiguration zu lesen. Clients können einen NULL-Kontext verwenden. Alle Fehlermeldungen werden jedoch in der Standardsprache des verwalteten Geräts ausgegeben. Die Einträge in { } in der folgenden Tabelle sind Objektpfade. Bei diesen Pfaden wird angenommen, dass sie bei vorherigen WQL-Abfragen ermittelt wurden. Für den Client sollte es nie notwendig sein, einen Objektpfad ohne Abfrage zu erstellen. Das **\_\_PATH**-Attribut für jedes Objekt enthält den Objektpfad für das jeweilige Objekt.

In allen folgenden Verwendungsfällen werden die Methoden **IWbemServices::ExecQuery** oder **IWbemServices::ExecQueryAsync** verwendet, um WQL-Abfragen auszuführen.

### Ermitteln der realen Adapter

Aufgabe	WQL-Abfrage	Ergebnisklasse	Kommentar
Auflisten aller Adapter	SELECT * FROM IANet_EthernetAdapter	IANet_EthernetAdapter	Gibt alle IANet_EthernetAdapters zurück.  Dies entspricht IWbemServices::CreateInstanceEnumAsync.
Bestimmen, ob es sich um virtuellen Adapter handelt	ASSOCIATORS OF {Adapterpfad} WHERE AssocClass = IANet_NetworkVirtualAdapter	IANet_TeamOfAdapters	Wenn bei der Abfrage keine Klassen ermittelt werden, handelt es sich bei dem Adapter um einen echten Adapter.
Bestimmen, ob es sich bei dem Adapter um einen Phantomadapter handelt	ASSOCIATORS OF {Adapterpfad} WHERE ResultClass = IANet_EthernetPCIDevice	IANet_EthernetPCIDevice	Falls es sich nicht um einen virtuellen Adapter handelt und bei dieser Abfrage keine Objekte zurückgegeben werden, ist der Adapter ein Phantomadapter.

Die Hauptklasse für die Adapter ist IANet\_EthernetAdapter. Diese Klasse wird sowohl für reale als auch virtuelle Adapter verwendet, und der Client muss darüber informiert werden, wie der Unterschied erkannt werden kann.

### Ermitteln der PCI-Geräte

Die Hauptklassen sind IANet\_EthernetPCIDevice, IANet\_PCIDevice und IANet\_AdapterDevice (eine Assoziationsklasse zum Assoziieren eines Adapters mit seinem Gerät).

In diesem Fall enthält die Assoziationsklasse keine Daten, d. h., sie selbst hat keinen Wert. IANet\_EthernetPCIDevice übernimmt Daten von IANet\_PCIDevice und enthält zusätzliche Attribute, die speziell für ein PCI-Gerät gelten, das ein Ethernet-Adapter ist.

Aufgabe	WQL-Abfrage	Ergebnisklasse	Kommentar
Auflisten von Netzwerk-PCI-Geräten	SELECT * FROM IANet_PCIDevice	IANet_PCIDevice	Bei der Abfrage können Modems und andere Geräte zurückgegeben werden, die Intel® PROSet nicht konfiguriert.  Dies entspricht IWbemServices::CreateInstanceEnumAsync.
Auflisten aller PCI-Geräte, die von mit Intel® PROSet konfigurierten Adaptern verwendet werden	SELECT * FROM IANet_EthernetPCIDevice	IANet_EthernetPCIDevice	Bei der Abfrage werden nur die PCI-Geräte zurückgegeben, die Intel® PROSet verwaltet.  Dies entspricht IWbemServices::CreateInstanceEnumAsync.
Bestimmen, ob der Adapter installiert ist	Nicht zutreffend	IANet_EthernetPCIDevice	Überprüfen Sie das "Verfügbarkeit"-Attribut von IANet_EthernetPCIDevice. Falls der Wert 10 (= "Nicht installiert") ist, ist das Gerät nicht installiert. Hinweis: Zu diesem Zeitpunkt liegen nur geringe Informationen zu dem Gerät vor.
Ermitteln des PCI-Geräts, das mit einem Adapter assoziiert ist	ASSOCIATORS OF { IANet_EthernetAdapter-Pfad} WHERE ResultClass = IANet_EthernetPCIDevice	IANet_EthernetPCIDevice	Falls bei dieser Abfrage keine Objekte zurückgegeben werden, handelt es sich bei dem Adapter entweder um einen virtuellen oder einen Phantomadapter.
Ermitteln des Adapters, der mit einem PCI-Gerät assoziiert ist	ASSOCIATORS OF {Ethernet PCI-Gerätepfad} WHERE ResultClass = IANet_EthernetAdapter	IANet_EthernetAdapter	Diese Abfrage ist für den Provider nicht ideal. Es ist zu bevorzugen, dass der Client mit dem Adapter beginnt.

[Zurück zum Seitenanfang](#)

## Ermitteln der Adaptereinstellungen

Die Einstellobjekte sind nicht direkt mit dem Adapter assoziiert. In Übereinstimmung mit dem CIM-Standard sind sie mit einem Konfigurationsobjekt assoziiert, das mit dem Adapter assoziiert ist.

Die bei diesem Teil des Schemas mitwirkenden Klassen sind IANet\_EthernetAdapter, IANet\_Configuration, IANet\_SettingInt, IANet\_SettingString, IANet\_SettingEnum, IANet\_SettingMultiSelection und IANetSettingSlider.

Die Assoziationsklassen IANet\_AdapterConfiguration und IANet\_SettingContext enthalten keine echten Daten. Sie wirken als Bindeglied zwischen den Einstellungen und ihrem übergeordneten Objekt.

Aufgabe	WQL-Abfragen	Ergebnisklasse	Kommentare
Ermitteln des Konfigurationsobjekts für den Adapter	ASSOCIATORS OF {IANet_EthernetAdapter-Pfad} WHERE ResultClass = IANet_Configuration	IANet_Configuration	Gibt exakt ein Objekt zurück; selbst wenn keine Einstellungen vorhanden sind, ist immer ein Konfigurationsobjekt vorhanden. Der Objektpfad von diesem Objekt wird in der nächsten Abfrage verwendet.

Ermitteln der mit einem Adapter assoziierten Einstellungen	ASSOCIATORS OF {IAnet_Configuration-Pfad} WHERE AssocClass = IAnet_SettingContext	Eine Kombination aus IAnet_SettingInt, IAnet_SettingString, IAnet_SettingEnum, IAnet_SettingMultiSelection, IAnet_SettingSlider	Gibt alle Einstellungsklassen zurück, die mit dem Adapter assoziiert sind. Der Client sollte das __CLASS-Attribut verwenden, um diesen Typ für jede Einstellung zu bestimmen.
--	---	---	---

[Zurück zum Seitenanfang](#)

## Ermitteln der Gruppenkonfiguration

Die Hauptklassen im Gruppenschema sind IAnet\_EthernetAdapter, IAnet\_TeamOfAdapters, IAnet\_NetworkVirtualAdapter und IAnet\_TeamedMemberAdapter.

Die besondere Schwierigkeit bei diesem Schema besteht darin, dass für jeden realen Adapter und für jeden virtuellen Adapter eine Instanz von IAnet\_EthernetAdapter existiert. Der Client muss fähig sein, zwischen dem virtuellen Adapter für eine Gruppe und den Adaptern zu unterscheiden, die Mitglieder der Gruppe sind.

Die Assoziationsklasse IAnet\_NetworkVirtualAdapter enthält keine brauchbaren Daten. Clients sind nur an den Endpunkten dieser Assoziation interessiert. IAnet\_TeamedMemberAdapter enthält brauchbare Daten zur Verwendung des Mitgliedsadapters innerhalb der Gruppe.

Aufgabe	WQL-Abfragen	Ergebnisklasse	Kommentare
Auflisten aller Gruppen	SELECT * FROM IAnet_TeamOfAdapters	IAnet_TeamOfAdapters	Für jede Gruppe ist eine Instanz von IAnet_TeamOfAdapters vorhanden.  Dies entspricht IWbemServices::CreateInstanceEnumAsync.
Ermitteln des virtuellen Adapters für eine Gruppe	ASSOCIATORS OF {IAnet_TeamOfAdapters-Pfad} WHERE AssocClass = IAnet_NetworkVirtualAdapter	IAnet_EthernetAdapter	Gibt nur das Adapterobjekt für den virtuellen Adapter in der Gruppe zurück. Dieser Adapter ist vorhanden, falls die Gruppe zwar erstellt wurde, aber noch nicht übernommen wurde. (Weiter unten finden Sie Informationen zum Aktualisieren der Konfiguration.)
Auflisten der Mitgliedsadapter einer Gruppe	ASSOCIATORS OF {IAnet_TeamOfAdapters-Pfad} WHERE AssocClass = IAnet_TeamedMemberAdapter	IAnet_EthernetAdapter	Gibt die Adapter zurück, die der Gruppe angehören, beschreibt jedoch nicht die Rolle der jeweiligen Adapter.
Bestimmen der Rolle eines Adapters in einer Gruppe	REFERENCES OF {IAnet_EthernetAdapter-Pfad} WHERE ResultClass = IAnet_TeamedMemberAdapter	IAnet_TeamedMemberAdapter	Die Klasse enthält Informationen dazu, welche Rolle der Mitgliedsadapter in der Gruppe einnimmt und welchen Status der Adapter in der Gruppe derzeit hat.

[Zurück zum Seitenanfang](#)

## Ermitteln der Gruppeneinstellungen

Die Einstellobjekte sind nicht direkt mit der Gruppe assoziiert. In Übereinstimmung mit dem CIM-Standard sind sie mit einem Konfigurationsobjekt assoziiert, das mit dem virtuellen IAnet\_EthernetAdapter für die Gruppe assoziiert ist. Dasselbe Konfigurationsobjekt ist auch mit dem IAnet\_TeamOfAdapters-Objekt für die Gruppe assoziiert.

Die bei diesem Teil des Schemas mitwirkenden Klassen sind IANet\_EthernetAdapter, IANet\_TeamOfAdapters, IANet\_Configuration, IANet\_SettingInt, IANet\_SettingString, IANet\_SettingEnum, IANet\_SettingMultiSelection und IANetSettingSlider.

Die Assoziationsklassen IANet\_AdapterConfiguration und IANet\_SettingContext enthalten keine echten Daten. Sie wirken als Bindeglied zwischen den Einstellungen und ihrem übergeordneten Objekt. Dies ist identisch mit der Situation hinsichtlich der Adaptereinstellungen.

Aufgabe	WQL-Abfragen	Ergebnisklasse	Kommentare
Ermitteln des Konfigurationsobjekts für die Gruppe, beginnend mit dem virtuellen Adapter	ASSOCIATORS OF {IANet_EthernetAdapter-Pfad} WHERE ResultClass = IANet_Configuration	IANet_Configuration	Gibt exakt ein Objekt zurück. Selbst wenn keine Einstellungen vorhanden sind, ist immer ein Konfigurationsobjekt vorhanden. Der Objektpfad von diesem Objekt wird in der nächsten Abfrage verwendet.
Ermitteln des Konfigurationsobjekts für die Gruppe, beginnend mit der Gruppe der Adapter	ASSOCIATORS OF {IANet_TeamOfAdapters-Pfad} WHERE ResultClass = IANet_Configuration		
Ermitteln der mit einem Adapter assoziierten Einstellungen	ASSOCIATORS OF {IANet_Configuration-Pfad} WHERE AssocClass = IANet_SettingContext	Eine Kombination aus IANet_SettingInt, IANet_SettingString, IANet_SettingEnum, IANet_SettingMultiSelection, IANet_SettingSlider	Gibt alle Einstellungsklassen zurück, die mit dem Adapter assoziiert sind. Der Client sollte das __CLASS-Attribut verwenden, um diesen Typ für jede Einstellung zu bestimmen.

[Zurück zum Seitenanfang](#)

## Ermitteln der VLAN-Konfiguration

Mit jedem Adapter, der VLANs unterstützt, ist ein IANet\_802dot1QVLANService mithilfe der Assoziationsklasse IANet\_Device802do1QVLANServiceImplementation assoziiert. Falls mit einem Adapter keine Instanz dieser Klasse assoziiert ist, unterstützt der Adapter VLANs nicht.

Jedes VLAN wird durch eine Instanz von IANet\_VLAN repräsentiert. Das VLAN ist nicht direkt mit dem Adapter, sondern mit dem IANet\_802dot1QVLANService für den Adapter assoziiert.

Die Assoziationsklasse IANet\_VLANFor wird verwendet, um jede VLAN-Instanz mit dem korrekten IANet\_802dot1QVLANService zu assoziieren. Diese Klasse enthält keine brauchbaren Daten für den Benutzer.

Aufgabe	WQL-Abfragen	Ergebnisklasse	Kommentare
Ermitteln des 802.1q VLAN-Serviceobjekts, das mit einem Adapter assoziiert ist	ASSOCIATORS OF {IANet_EthernetAdapter-Pfad} WHERE ResultClass = IANet_802dot1QVLANService	IANet_802dot1QVLANService	Gibt ein oder kein Objekt zurück.
Ermitteln der VLANs auf einem Adapter	ASSOCIATORS OF {IANet_802dot1QVLANService-Pfad} WHERE ResultClass = IANet_VLAN	IANet_VLAN	Dabei werden keine Objekte zurückgegeben, wenn keine VLANs installiert sind.

## Ermitteln der VLAN-Einstellungen

Die Einstellobjekte sind nicht direkt mit dem VLAN assoziiert. In Übereinstimmung mit dem CIM-Standard sind sie mit einem Konfigurationsobjekt assoziiert, das mit dem IANet\_VLAN-Objekt für das VLAN assoziiert ist.

Die bei diesem Teil des Schemas mitwirkenden Klassen sind IANet\_VLAN, IANet\_Configuration, IANet\_SettingInt, IANet\_SettingString, IANet\_SettingEnum, IANet\_SettingMultiSelection und IANetSettingSlider.

Die Assoziationsklassen IANet\_VLANConfiguration und IANet\_SettingContext enthalten keine echten Daten. Sie wirken als Bindeglied zwischen den Einstellungen und ihrem übergeordneten Objekt. Dies ist identisch mit der Situation hinsichtlich der Adaptereinstellungen.

Aufgabe	WQL-Abfragen	Ergebnisklasse	Kommentare
Ermitteln des Konfigurationsobjekts für das VLAN	ASSOCIATORS OF {IANet_VLAN-Pfad} WHERE ResultClass = IANet_Configuration	IANet_Configuration	Gibt exakt ein Objekt zurück; selbst wenn keine Einstellungen vorhanden sind, ist immer ein Konfigurationsobjekt vorhanden. Der Objektpfad von diesem Objekt wird in der nächsten Abfrage verwendet.
Ermitteln der mit einem VLAN assoziierten Einstellungen	ASSOCIATORS OF {IANet_Configuration-Pfad} WHERE AssocClass = IANet_SettingContext	Eine Kombination aus IANet_SettingInt, IANet_SettingString, IANet_SettingEnum, IANet_SettingMultiSelection, IANet_SettingSlider	Gibt alle Einstellungsklassen zurück, die mit dem VLAN assoziiert sind. Der Client sollte das __CLASS-Attribut verwenden, um den Typ für jede Einstellung zu bestimmen.

## Ermitteln der IP-Protokollinformationen

Der Provider teilt beschränkte Informationen zu den IP-Protokollendpunkten mit, die mit den Adaptern, VLANs und Gruppen assoziiert sind. Keine anderen Protokolle werden unterstützt.

Die Hauptklasse, die die Protokollinformationen enthält, ist IANet\_IPProtocolEndpoint. Es gibt zwei Assoziationsklassen: IANet\_VLANProtocolDependency und IANet\_AdapterProtocolImplementation. Um den IP-Endpunkt für eine Gruppe zu ermitteln, müssen Sie zuerst den virtuellen IANet\_EthernetAdapter für die Gruppe, d. h. den mit dieser Instanz assoziierten IP-Endpunkt, ermitteln.

Aufgabe	WQL-Abfragen	Ergebnisklasse	Kommentare
Ermitteln der IP-Protokollendpunkte, die mit einem Adapter assoziiert sind	ASSOCIATORS OF {Pfad für IANet_EthernetAdapter} WHERE ResultClass = IANet_IPProtocolEndpoint	IANet_IPProtocolEndpoint	Es kann zwar vorkommen, dass Adapter mehrere IP-Adressen haben, sie sind jedoch alle nur mit einer Instanz eines IP-Protokollendpunkts assoziiert.
Ermitteln der IP-Protokollendpunkte, die mit einem VLAN assoziiert sind	ASSOCIATORS OF {Pfad für IANet_VLAN} WHERE ResultClass = IANet_IPProtocolEndpoint	IANet_IPProtocolEndpoint	Es ist ein IP-Protokollendpunkt vorhanden, der mit dem VLAN assoziiert ist.

---

## Ermitteln der Boot Agent-Informationen

Jeder Adapter, der einen Boot Agent im Flash-ROM unterstützen kann, ist mithilfe der IANet\_DeviceBootServiceImplementation-Assoziationsklasse mit einer IANet\_BootAgent-Instanz assoziiert.

Aufgabe	WQL-Abfragen	Ergebnisklasse	Kommentare
Ermitteln des mit einem Adapter assoziierten Boot Agent	ASSOCIATORS OF {Pfad für IANet_EthernetAdapter} WHERE ResultClass = IANet_BootAgent	IANet_BootAgent	Die folgenden schreibgeschützten Attribute liefern Informationen über das Boot-ROM-Image für diesen Adapter: InvalidImageSignature, Version, UpdateAvailable, FlashImageType

[Zurück zum Seitenanfang](#)

---

## Ermitteln der Boot Agent-Einstellungen

Die Einstellobjekte sind nicht direkt mit dem Boot Agent assoziiert. In Übereinstimmung mit dem CIM-Standard sind sie mit einem Konfigurationsobjekt assoziiert, das mit dem Boot Agent assoziiert ist.

Die bei diesem Teil des Schemas mitwirkenden Klassen sind IANet\_BootAgent, IANet\_Configuration, IANet\_SettingInt, IANet\_SettingString, IANet\_SettingEnum, IANet\_SettingMultiSelection und IANetSettingSlider.

Die Assoziationsklassen IANet\_BootAgentConfiguration und IANet\_SettingContext enthalten keine echten Daten. Sie wirken als Bindeglied zwischen den Einstellungen und ihrem übergeordneten Objekt.

Aufgabe	WQL-Abfragen	Ergebnisklasse	Kommentare
Ermitteln des Konfigurationsobjekts für den Boot Agent	ASSOCIATORS OF { IANet_BootAgent-Pfad} WHERE ResultClass = IANet_Configuration	IANet_Configuration	Gibt exakt ein Objekt zurück. Selbst wenn keine Einstellungen vorhanden sind, ist immer ein Konfigurationsobjekt vorhanden. Der Objektpfad von diesem Objekt wird in der folgenden Abfrage verwendet.
Ermitteln der mit einem Boot Agent assoziierten Einstellungen	ASSOCIATORS OF {IANet_Configuration-Pfad} WHERE AssocClass = IANet_SettingContext	Eine Kombination aus IANet_SettingInt, IANet_SettingString, IANet_SettingEnum, IANet_SettingMultiSelection, IANet_SettingSlider	Gibt alle Einstellungsklassen zurück, die mit dem Adapter assoziiert sind. Der Client sollte das __CLASS-Attribut verwenden, um diesen Typ für jede Einstellung zu bestimmen.

---

Lesen Sie bitte alle [Einschränkungen und Haftungsausschlüsse](#).

---

[Zurück zur Inhaltsseite](#)   [Zurück zum Seitenanfang](#)

# Aktualisieren der Konfiguration: Benutzerhandbuch für Intel® PRO Netzwerkadapter – WMI und CDM Provider

## [Überblick](#)

[Ändern der Adapter-, Gruppen- oder VLAN-Einstellungen](#)

[Erstellen einer neuen \(leeren\) Gruppe](#)

[Hinzufügen eines Adapters zu einer Gruppe](#)

[Entfernen eines Adapters aus einer Gruppe](#)

[Löschen einer Gruppe](#)

[Ändern des Modus einer Gruppe](#)

[Ändern der Priorität eines Adapters in einer Gruppe](#)

[Deinstallieren eines Adapters](#)

[Erstellen eines VLAN](#)

[Ändern der Attribute eines VLAN](#)

[Löschen eines VLAN](#)

[Aktualisieren des Boot Agent](#)

---

## Überblick

In den meisten Fällen muss die Client-Anwendung zur Aktualisierung der Konfiguration ein Sitzungshandle von der IANet\_NetService-Klasse erhalten und dieses Handle in einem IWbemContext-Kontextobjekt speichern. Änderungen in der Konfiguration werden nur dann wirksam, wenn die **Apply**-Methode für IANet\_NetService aufgerufen wird. Es gibt einige Ausnahmen für diese Anforderung:

- Änderungen an der Boot Agent-Klasse werden sofort wirksam und erfordern kein Sitzungshandle.
- Bestimmte Methodenaufrufe (z. B. zum Identifizieren eines Adapters) bewirken, dass ein Vorgang ausgeführt wird, bevor **Apply** aufgerufen wird.

Bei manchen Vorgängen können Sie das PreCheck-Qualifizierungsmerkmal im Kontext verwenden, um festzustellen, ob ein Vorgang erlaubt ist. Auf diese Weise ist es über die Benutzeroberfläche möglich, bestimmte Steuerelemente oder Menüpunkte bei Bedarf zu deaktivieren.

[Zurück zum Seitenanfang](#)

---

## Ändern der Adapter-, Gruppen- oder VLAN-Einstellungen

Ändern der Adapter-, Gruppen- oder VLAN-Einstellungen:

- Ein Sitzungshandle ist erforderlich.
- PreCheck steht zur Verfügung.
- Erfordert einen **Apply**-Aufruf, bevor der Vorgang ausgeführt wird.

Zum Ändern einer Adapter-, VLAN- oder Gruppen-Einstellung muss der Client zuerst den Objektpfad für die Einstellung erhalten, die geändert wird. Die beste Methode besteht darin, die Einstellungen auf dem Objekt aufzulisten und das **\_\_PATH**-Attribut der Einstellung zu speichern (siehe oben).

Die Aktualisierung einer Einstellung ist am einfachsten auf folgende Weise möglich:

1. Rufen Sie eine Instanz des Einstellungsobjekts von WMI ab.
2. Ändern Sie das CurrentValue-Attribut (mithilfe von **IWbemClassObject::Put()**).
3. Rufen Sie **IWbemServices::PutInstance()** auf, um die geänderte Instanz zum WMI Provider zurückzuleiten. **PutInstance** muss mit dem Kennzeichen **WBEM\_FLAG\_UPDATE\_ONLY** aufgerufen werden.

Der WMI Provider validiert CurrentValue und gibt **WBEM\_E\_FAIL** zurück, falls die Validierung fehlschlägt. Der genaue Grund für

das Versagen wird im Beschreibungsattribut des `INet_ExtendedStatus`-Objekts angegeben.

Folgende einstellungsspezifischen Beschreibungen sind möglich:

- Der Ganzzahl-Einstellungswert war kleiner als das zulässige Minimum.
- Der Ganzzahl-Einstellungswert war größer als das zulässige Maximum.
- Der Ganzzahl-Einstellungswert entspricht keinem der zulässigen Schritte.
- Die Länge der Zeichenfolgeneinstellung übersteigt das zulässige Maximum.
- Der Einstellungswert entspricht keinem der zulässigen Werte.

Die letzte Beschreibung wird zurückgegeben, wenn der aktuelle Wert für `INet_SettingEnum`, `INet_SettingSlider` oder `INet_SettingMultiSelection` keinem der zulässigen Werte entspricht.

Das einzige Attribut für eine Einstellung, das der Client ändern kann, ist `CurrentValue`. Der WMI Provider ignoriert alle Änderungen an anderen Werten.

Für die Einstellungsklasse sind keine unterstützten Methoden vorhanden. Wenn Sie Änderungen an einer Einstellung durchführen möchten, ändern Sie die `CurrentValue`-Eigenschaft und rufen Sie **PutInstance** auf.

[Zurück zum Seitenanfang](#)

---

## Erstellen einer neuen (leeren) Gruppe

Erstellen einer neuen Gruppe:

- Ein `Sitzungshandle` ist erforderlich.
- `PreCheck` steht zur Verfügung.
- Erfordert einen **Apply**-Aufruf, bevor der Vorgang ausgeführt wird.

Wenn Sie eine neue Gruppe erstellen möchten, erstellen Sie eine Instanz von `INet_TeamOfAdapters` (verwenden Sie z. B. **`IWbemServices::GetObject()`**), um ein Klassenobjekt für `INet_TeamOfAdapters` zu erhalten, und verwenden Sie dann **`IWbemServices::SpawnInstance()`**, um eine Instanz dieses Objekts zu erstellen).

Verwenden Sie danach **`IWbemClassObject::Put`**, um das `TeamMode`-Attribut in der Instanz auf den gewünschten Gruppentyp (z. B. `AFT`) einzustellen. Rufen Sie schließlich **`IWbemServices::PutInstance()`** auf, um die Gruppe mit dem Kennzeichen `WBEM_FLAG_CREATE_ONLY` zu erstellen.

Der Objektpfad für die neue Gruppe wird in **`IWbemCallResultObject`** gespeichert. Dieses Objekt wird an Sie zurückgeleitet, wenn der Aufruf abgeschlossen ist. Mit der Methode **`IWbemCallResult::GetResultString`** wird der neue Objektpfad ermittelt.

Falls dieser Vorgang fehlschlägt, sollten Sie `INet_ExtendedStatus` überprüfen, um die Gründe für das Versagen zu ermitteln.

Die virtuellen `INet_EthernetAdapter`- und `INet_IPProtocolEndpoint`-Klassen für die Gruppe stehen erst zur Verfügung, wenn **Apply** aufgerufen wurde. Für den Zugriff auf die Einstellungen für die Gruppe kann das `INet_Configuration`-Objekt verwendet werden, das mit der neuen `INet_TeamOfAdapters`-Instanz assoziiert ist.

[Zurück zum Seitenanfang](#)

---

## Hinzufügen eines Adapters zu einer Gruppe

Hinzufügen eines Adapters zu einer Gruppe:

- Ein `Sitzungshandle` ist erforderlich.
- `PreCheck` steht zur Verfügung.
- Erfordert einen **Apply**-Aufruf, bevor der Vorgang ausgeführt wird.

Wenn Sie einer Gruppe einen Adapter hinzufügen möchten, erstellen Sie eine Instanz von `INet_TeamedMemberAdapter` (verwenden Sie z. B. **`IWbemServices::GetObject()`**), um ein Klassenobjekt für `INet_TeamedMemberAdapter` zu erhalten, und verwenden Sie dann **`IWbemServices::SpawnInstance()`**, um eine Instanz dieses Objekts zu erstellen).

Die folgenden Attribute im Objekt müssen mithilfe von `IWbemClassObject::Put()` eingestellt werden:

- `GroupComponent` muss auf den vollständigen Objektpfad der `IANet_TeamOfAdapters`-Instanz eingestellt werden, der der Adapter hinzugefügt werden soll.
- `PartComponent` muss auf den vollständigen Objektpfad der `IANet_EthernetAdapter`-Instanz eingestellt werden, die der Gruppe hinzugefügt werden soll.

Sie können auch die Priorität des Adapters in der Gruppe einstellen. Rufen Sie schließlich **`IWbemServices::PutInstance()`** auf, um den Adapter der Gruppe mit dem Kennzeichen `WBEM_FLAG_CREATE_ONLY` hinzuzufügen. Falls dieser Vorgang fehlschlägt, ermitteln Sie den Fehlercode in `IANet_ExtendedStatus`.

[Zurück zum Seitenanfang](#)

---

## Entfernen eines Adapters aus einer Gruppe

Entfernen eines Adapters aus einer Gruppe:

- Ein Sitzungshandle ist erforderlich.
- `PreCheck` steht zur Verfügung.
- Erfordert einen **Apply**-Aufruf, bevor der Vorgang ausgeführt wird.

Wenn Sie einen Adapter aus einer Gruppe entfernen möchten, löschen Sie die `IANet_TeamedMemberAdapter`-Instanz, die den Adapter mit der Gruppe assoziiert, mithilfe von **`IWbemServices::DeleteInstance()`**. Falls dieser Vorgang fehlschlägt, ermitteln Sie den Fehlercode in `IANet_ExtendedStatus`.

[Zurück zum Seitenanfang](#)

---

## Löschen einer Gruppe

Löschen einer Gruppe:

- Ein Sitzungshandle ist erforderlich.
- `PreCheck` steht zur Verfügung.
- Erfordert einen **Apply**-Aufruf, bevor der Vorgang ausgeführt wird.

Wenn Sie eine Gruppe löschen möchten, löschen Sie die `IANet_TeamOfAdapters`-Instanz mithilfe von **`IWbemServices::DeleteInstance()`**. Falls dieser Vorgang fehlschlägt, ermitteln Sie den Fehlercode in `IANet_ExtendedStatus`.

[Zurück zum Seitenanfang](#)

---

## Ändern des Modus einer Gruppe

Ändern des Modus einer Gruppe:

- Ein Sitzungshandle ist erforderlich.
- `PreCheck` steht zur Verfügung.
- Erfordert einen **Apply**-Aufruf, bevor der Vorgang ausgeführt wird.

Wenn Sie den Modus einer Gruppe ändern möchten, ermitteln Sie die Instanz von `IANet_TeamOfAdapters` für die Gruppe (verwenden Sie z. B. **`IWbemServices::GetObject`** mit dem Objektpfad der Gruppe). Verwenden Sie dann **`IWbemClassObject::Put`**, um das `TeamMode`-Attribut für die Gruppe zu ändern. Rufen Sie schließlich **`IWbemClassObject::PutInstance`** auf, um dem WMI Provider mit dem Kennzeichen `WBEM_FLAG_UPDATE_ONLY` den Befehl zur Aktualisierung des Gruppenmodus zu geben. Falls dieser Vorgang fehlschlägt, ermitteln Sie den Fehlercode in `IANet_ExtendedStatus`.

[Zurück zum Seitenanfang](#)

---

## Ändern der Priorität eines Adapters in einer Gruppe

Ändern der Priorität eines Adapters in einer Gruppe:

- Ein Sitzungshandle ist erforderlich.
- PreCheck steht zur Verfügung.
- Erfordert einen **Apply**-Aufruf, bevor der Vorgang ausgeführt wird.

Zum Ändern der Priorität eines Adapters sollte der Client zuerst die IANet\_TeamedMemberAdapter-Instanz für den Adapter ermitteln (verwenden Sie z. B. **IWbemServices::GetObject** mit dem Objektpfad). Der Client kann dann **IWbemClassObject::Put** verwenden, um das AdapterFunction-Attribut für den Adapter zu ändern. Schließlich muss der Client **IWbemClassObject::PutInstance** aufrufen, um dem WMI Provider den Befehl zur Aktualisierung der Priorität des Adapters zu geben. Falls dieser Vorgang fehlschlägt, sollte der Client den Fehlercode in IANet\_ExtendedStatus feststellen.

[Zurück zum Seitenanfang](#)

---

## Deinstallieren eines Adapters

Deinstallieren eines Adapters:

- Ein Sitzungshandle ist erforderlich.
- PreCheck steht zur Verfügung.
- Erfordert einen **Apply**-Aufruf, bevor der Vorgang ausgeführt wird.

Wenn Sie einen Adapter deinstallieren möchten, rufen Sie **IWbemServices::DeleteInstance** auf. Geben Sie dabei den Objektpfad des zu deinstallierenden Adapters an.

[Zurück zum Seitenanfang](#)

---

## Erstellen eines VLAN

Erstellen eines VLAN:

- Ein Sitzungshandle ist erforderlich.
- PreCheck steht zur Verfügung.
- Erfordert einen **Apply**-Aufruf, bevor der Vorgang ausgeführt wird.

Wenn Sie ein VLAN erstellen möchten, rufen Sie die **CreateVLAN**-Methode auf dem IANet\_802dot1QVLANService für den Adapter auf, dem das VLAN hinzugefügt werden soll. Die folgenden Argumente müssen an die Methode weitergeleitet werden:

- VLANNumber, d. h. die Nummer des VLAN (Bereich 1- 4094)
- Name, d. h. einen benutzerdefinierbaren Namen zur Identifikation des VLAN

Durch diese Funktion wird der Objektpfad des neu erstellten VLAN im Out-Parameter VLANpath zurückgegeben. Falls dieser Vorgang fehlschlägt, ermitteln Sie den Fehlercode in IANet\_ExtendedStatus.

[Zurück zum Seitenanfang](#)

---

## Ändern der Attribute eines VLAN

Ändern der Attribute eines VLAN:

- Ein Sitzungshandle ist erforderlich.
- PreCheck steht zur Verfügung.
- Erfordert einen **Apply**-Aufruf, bevor der Vorgang ausgeführt wird.

Der Client kann die VLANNumber- und VLANName-Attribute für ein VLAN ändern. Wenn Sie die Priorität eines Adapters ändern

möchten, ermitteln Sie zuerst die Instanz von IANet\_VLAN für den Adapter (verwenden Sie z. B. **IWbemServices::GetObject** mit dem Objektpfad der Gruppe).

Geben Sie dann für VLANNumber oder VLANName die gewünschten Werte ein. Rufen Sie schließlich **IWbemClassObject::PutInstance** auf, um dem WMI Provider mit dem Kennzeichen WBEM\_FLAG\_UPDATE\_ONLY den Befehl zur Aktualisierung der Attribute zu geben. Falls dieser Vorgang fehlschlägt, ermitteln Sie den Fehlercode in IANet\_ExtendedStatus.

[Zurück zum Seitenanfang](#)

## Löschen eines VLAN

Löschen eines VLAN:

- Ein Sitzungshandle ist erforderlich.
- PreCheck steht zur Verfügung.
- Erfordert einen **Apply**-Aufruf, bevor der Vorgang ausgeführt wird.

Wenn Sie ein VLAN deinstallieren möchten, rufen Sie **IWbemServices::DeleteInstance** auf. Geben Sie dabei den Objektpfad des zu löschenden VLAN an.

[Zurück zum Seitenanfang](#)

## Aktualisieren des Boot Agent

Aktualisieren des Boot Agent:

- Ein Sitzungshandle ist nicht erforderlich.
- PreCheck steht nicht zur Verfügung.
- Erfordert keinen **Apply**-Aufruf, bevor der Vorgang ausgeführt wird.

Der Client kann das Boot Agent-Image mithilfe von Methodenaufrufen aktualisieren. Ermitteln Sie zum Lesen/Schreiben eines Flash-Image zuerst die Instanz von IANet\_BootAgent für den Adapter (verwenden Sie z. B. **IWbemServices::GetObject** mit dem Objektpfad der Gruppe).

Führen Sie dann ReadFlash() aus, um das vorhandene Flash-Boot-ROM-Image zu lesen, oder ProgramFlash(), um das Flash-Boot-ROM-Image zu aktualisieren. Falls dieser Vorgang fehlschlägt, ermitteln Sie den Fehlercode in IANet\_ExtendedStatus.

Aufgabe	WMI-Methoden	Ergebnis	Kommentare
Aktualisieren oder Einfügen eines Boot-ROM-Image für den Adapter	uint32 ProgramFlash( [IN, ValueMap {"0","1"}, Values {"Check Version", "Write Flash"}: Amended  ]  uint32 Action,  [IN]  uint8	Wenn die Aktion "Check Version" (Version überprüfen) angegeben wird, wird bei dieser Methode eine Warnmeldung zurückgegeben, falls das zu aktualisierende Boot-ROM-Image in NewFlashData[] älter als das auf dem NIC vorhandene Image ist.  Wenn die Aktion "Write" (Schreiben) angegeben wird, wird der FLASH-ROM-Speicher auf dem NIC mit NewFlashData[] aktualisiert.	Diese Methode wird zur Aktualisierung des Flash-ROM-Speichers auf dem NIC verwendet. Während der Aktualisierung des Flash-Speichers unterbricht der NIC die Kommunikation mit dem Netzwerk.

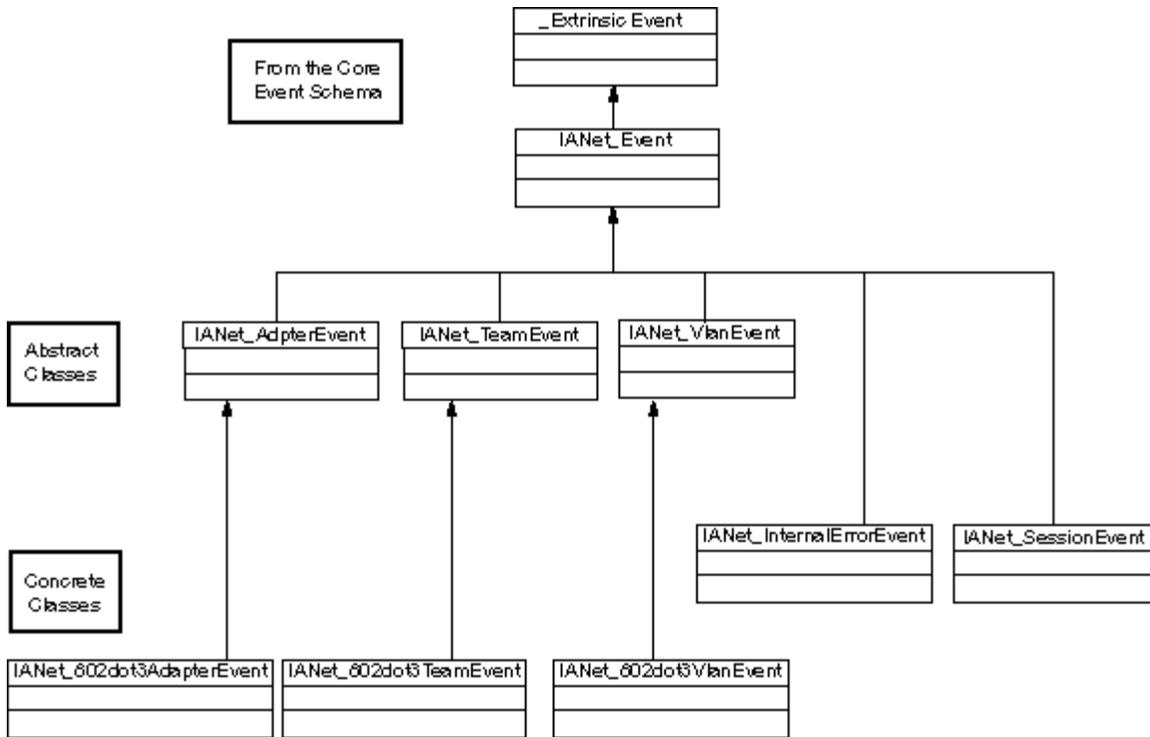
	<pre>NewFlashData[], [OUT] string strErrorMessage );</pre>		
Lesen des Boot-ROM-Image	<pre>uint32 ReadFlash( [OUT] uint8 FlashData[] );</pre>	FlashData[] enthält das Flash-ROM-Image auf dem NIC.	Diese Methode dient zum Lesen des Flash-ROM-Speichers auf dem NIC. Die Daten können in einer Datei gespeichert werden.

Lesen Sie bitte alle [Einschränkungen und Haftungsausschlüsse](#).

[Zurück zur Inhaltsseite](#)   [Zurück zum Seitenanfang](#)

# Ereignisbenachrichtigungen: Benutzerhandbuch für Intel® PRO Netzwerkkadpter – WMI und CDM Provider

[Konkrete Ereignisklassen](#)  
[Registrieren für Ereignisse](#)



[Zurück zum Seitenanfang](#)

## Konkrete Ereignisklassen

### IANet\_802dot3AdapterEvent

**Zweck**  
Bei diesem Ereignis wird der Client über eine Änderung des Status oder der Konfiguration eines Adapters informiert.

**Auslöser**  
Das Ereignis tritt ein, wenn ein Adapterstatus sich geändert hat oder wenn eine Adaptereinstellung geändert und **Apply** (Übernehmen) angewendet wird.

**Ereignisdaten**  
AdapterPath enthält den Objektpfad des Adapters, der das Ereignis ausgelöst hat.

### IANet\_802dot3TeamEvent

## Zweck

Bei diesem Ereignis wird der Client über eine Änderung des Status oder der Konfiguration einer Gruppe informiert.

## Auslöser

Dieses Ereignis tritt in folgenden Situationen ein:

- Wenn der Status eines Adapters sich ändert
- Wenn eine Gruppeneinstellung geändert und **Apply** (Übernehmen) angewendet wird
- Wenn die Gruppenkonfiguration sich geändert hat und **Apply** (Übernehmen) angewendet wird

## Ereignisdaten

TeamPath enthält den Objektpfad der Gruppe, die das Ereignis ausgelöst hat.

## IANet\_802dot3VlanEvent

### Zweck

Bei diesem Ereignis wird der Client über eine Änderung des Status oder der Konfiguration eines VLAN informiert.

### Auslöser

Dieses Ereignis tritt in folgenden Situationen ein:

- Wenn der Status eines VLAN sich ändert
- Wenn eine VLAN-Einstellung geändert und **Apply** (Übernehmen) angewendet wird
- Wenn die VLAN-Konfiguration sich geändert hat und **Apply** (Übernehmen) angewendet wird

### Ereignisdaten

VlanPath enthält den Objektpfad des VLAN, das das Ereignis ausgelöst hat.

[Zurück zum Seitenanfang](#)

---

## Registrieren für Ereignisse

Anwendungen sollten **IWbemServices:: ExecNotificationQuery** oder **IWbemServices:: ExecNotificationQueryAsync** verwenden, um Ereignisbenachrichtigungen anzufordern. Die folgenden Abfragen sind Beispiele für Ereignisbenachrichtigungsabfragen. Diese Liste ist nicht vollständig, da eine Vielzahl von Abfragen möglich ist.

- **SELECT \* FROM IANet\_Event** – Wird zum Abfragen aller Ereignisse verwendet.
  - **SELECT \* FROM IANet\_AdapterEvent** – Wird zum Abfragen aller Adapterereignisse verwendet.
  - **SELECT \* FROM IANet\_TeamEvent** – Wird zum Abfragen aller Gruppenereignisse verwendet.
  - **SELECT \* FROM IANet\_SessionEvent** – Wird zum Abfragen aller Sitzungsereignisse verwendet.
  - **SELECT \* FROM IANet\_VlanEvent** – Wird zum Abfragen aller VLAN-Ereignisse verwendet.
  - **SELECT \* FROM IANet\_InternalErrorEvent** – Wird zum Abfragen aller internen Ereignisse verwendet.
  - **SELECT \* FROM IANet\_AdapterEvent WHERE AdapterPath={IANet\_EthernetAdapter-Objektpfad}** – Wird zum Abfragen von Adapterereignissen eines bestimmten Adapters verwendet.
- 

Lesen Sie bitte alle [Einschränkungen und Haftungsausschlüsse](#).

---

[Zurück zur Inhaltsseite](#)   [Zurück zum Seitenanfang](#)

# Optimierte WQL-Abfragen: Benutzerhandbuch für Intel® PRO Netzwerkkadpter – WMI und CDM Provider

## [Überblick](#)

[Ermitteln der Einstellungen für bestimmte Adapter, VLANs oder Gruppen](#)

[Ermitteln einer einzelnen Einstellung](#)

---

## Überblick

Der WMI Provider ist optimal konfiguriert, um Anwendungen das Ermitteln von Einstellungen mithilfe von Abfragen zu erlauben. Der WMI Provider kann die folgenden Abfragen erkennen und gibt nur die übereinstimmenden Objekte zurück. Alle anderen Abfragen bewirken, dass der WMI Provider alle Einstellungen für alle Objekte ermittelt, und CIMOM filtert die Angaben, bevor sie die Anwendung erreichen. Bei mehreren Adaptern, Gruppen und VLANs kann beim Abrufen der erforderlichen Daten eine Verzögerung von mehreren Sekunden eintreten.

[Zurück zum Seitenanfang](#)

---

## Ermitteln der Einstellungen für bestimmte Adapter, VLANs oder Gruppen

Die folgende Abfrage ermittelt nur die Einstellungen für einen bestimmten Adapter, ein bestimmtes VLAN bzw. eine bestimmte Gruppe. WQL erlaubt keine zusätzlichen Sätze im WHERE-Satz.

```
ASSOCIATORS OF {IAnet_Configuration-Pfad} WHERE AssocClass = IAnet_SettingContext
```

[Zurück zum Seitenanfang](#)

---

## Ermitteln einer einzelnen Einstellung

Die folgende Abfrage kann verwendet werden, um eine einzelne Einstellung für ein Objekt abzufragen, ohne andere Einstellungen in die Abfrage einzubeziehen:

```
SELECT * FROM [EINSTELLUNGSKLASSE] WHERE ParentId="[Geräte-ID]" AND ParentType="[Typ]" AND Caption="[EINSTELLUNGSNAME]"
```

### Hinweise:

- Die exakte Einstellungsklasse muss eingegeben werden und nicht etwa eine Basisklasse (z. B. IAnet\_SettingInt).
- Als ParentTypes (übergeordnete Typen) kommen "NIC", "Team", "VLAN" oder "BootAgent" infrage.
- ParentId ist die GUID, die das Objekt mit der Einstellung eindeutig identifiziert (bei einem Adapter ist dies die DeviceId).
- Diese Methode wird nicht empfohlen, wenn die mit einem Objekt assoziierten Einstellungen ermittelt werden sollen; die bevorzugte Methode stützt sich auf die Verwendung von Assoziationen. WQL unterstützt jedoch nicht die erforderliche komplexe Abfrage (d. h., WQL unterstützt nicht ASSOCIATORS OF {IAnet\_Configuration-Pfad} WHERE AssocClass = IAnet\_SettingContext AND Caption="[EINSTELLUNGSNAME]").

---

Lesen Sie bitte alle [Einschränkungen und Haftungsausschlüsse](#).

---

[Zurück zur Inhaltsseite](#)   [Zurück zum Seitenanfang](#)

# Diagnose: Benutzerhandbuch für Intel® PRO Netzwerkadapter – WMI und CDM Provider

[Diagnose-Klassen](#)  
[Registrierungseinträge](#)  
[Protokollieren](#)  
[Assoziative Klassen](#)  
[Tests](#)

---

## Diagnose-Klassen

### IANet\_DiagTest

#### Zweck

IANet\_DiagTest gehört einer Unterklasse von CIM\_DiagnosticTest an. Die Klasse bietet eine generelle Plattform für das Ausführen und Steuern von Diagnosetests für einen von Intel® PROSet unterstützten Ethernet-Adapter. Die Überklasse CIM\_DiagnosticTest ist so gestaltet, dass sie das Testen beliebiger Computerhardware auf einem CIM-aktivierten System unterstützt. Die Eigenschaften der Klasse sind von deskriptiver Art und der Testmechanismus wird von den exponierten Methoden bereitgestellt.

#### Instanzen

Der Schlüssel ist der Name, und bei diesem Provider handelt es sich um die Konkatenation des numerischen Testindex an der GUID für den Referenzadapter (z. B. 1@{12345678-9ABC-DEF0-1234-123456789012}). Dieser Schlüsselwert stellt in gewissem Sinn eine redundante Information dar, da alle Informationen für das Referenzieren eines Adapters und Tests als Objektparameter an RunTest und andere Methoden übertragen werden. Die Instanz muss jedoch mit den Parametern konsistent sein, da der Provider den Befehl andernfalls ablehnt. In der Untertitel-Eigenschaft wird der Name des Tests angegeben, der von dieser Instanz ausgeführt wird. Andere Eigenschaftsfelder enthalten weitere Beschreibungen und Runtime-Informationen.

#### Erstellen von Instanzen

Sie können keine Instanzen von IANet\_DiagTest erstellen.

#### Löschen von Instanzen

Sie können keine Instanzen von IANet\_DiagTest löschen.

#### Ändern von Eigenschaften

Für diese Klasse sind keine durch den Benutzer veränderbaren Eigenschaften vorhanden.

#### Assoziationen

- Eine Instanz von IANet\_DiagTestForMSE assoziiert einen IANet\_DiagTest mit einem IANet\_ManagedSystemElement. Bei dem IANet\_ManagedSystemElement handelt es sich um eine IANet\_EthernetAdapter-Instanz.
- Eine Instanz von IANet\_DiagResultForTest assoziiert einen IANet\_DiagTest mit einer IANet\_DiagnosticResult-Instanz.
- Eine Instanz von IANetDiagSettingForTest assoziiert einen IANet\_DiagTest mit einer IANet\_DiagSetting-Instanz.

#### Nicht unterstützte Eigenschaften

Install Date, OtherCharacteristicDescription

#### Methoden

Diese Klasse unterstützt die folgenden Methoden:

- RunTest – Führt einen Test aus, der durch drei Parameter definiert wird:
  - SystemElement – definiert den Adapter, für den der Test mit Referenz auf eine Instanz von SystemElement ausgeführt wird; dabei handelt es sich immer um die Unterklasse IANet\_EthernetAdapter.
  - Setting – definiert den Test, der ausgeführt wird, und die Art und Weise, in der dieser Test mit Referenz auf eine Instanz von CIM\_DiagnosticSetting ausgeführt wird; dabei handelt es sich immer um die Unterklasse IANet\_DiagSetting.

- DiagnosticResult – definiert eine Instanz der Klasse CIM\_DiagnosticResult; dabei handelt es sich immer um die Klasse IANet\_DiagResult.
- DiscontinueTest – versucht, einen Diagnosetest, der gerade abläuft und durch die beiden Parameter mit Referenz auf CIM\_ManagedSystemElement und CIM\_DiagnosticResult definiert ist, anzuhalten. Diese Parameter üben dieselben Funktionen aus wie RunTest. Ein dritter Parameter mit der Bezeichnung TestingStopped gibt einen BOOLSCHEN Wert zurück, der anzeigt, ob der Test durch den Befehl gestoppt werden konnte.
- ClearResults – Löscht die Testergebnisse und verwendet die folgenden Parameter:
  - SystemElement
  - ResultsNotCleared

Der referenzierte Parameter ManagedSystemElement wirkt in Kombination mit dem Objektpfad dieses Objekts und referenziert Instanzen von DiagnosticResultForMSE; diese werden gelöscht. Alle Referenzen von DiagnosticResult-Objekten, auf die durch DiagnosticResultForMSE Bezug genommen wird, werden ebenfalls gelöscht. Außerdem werden alle Instanzen von DiagnosticResultForTest, die auf die gelöschten DiagnosticResult-Objekte verweisen, gelöscht. Das Zeichenfolge-Array für den Output-Parameter ResultsNotCleared listet die Schlüssel der DiagnosticResults auf, die nicht gelöscht werden konnten.

## Klassenhierarchie

Für CimV2. Unbenutzte Eigenschaften und Methoden werden nicht aufgeführt.

- CIM\_ManagedElement:
  - Caption
  - Description
- CIM\_ManagedSystemElement:
  - Install Date
  - Name
  - Status
- CIM\_LogicalElement
- CIM\_Service:
  - Key
    - Name (Zeichenfolge)
  - Properties
    - Caption (Zeichenfolge)
    - CreationClassName (Zeichenfolge)
    - Description (Zeichenfolge)
    - Started (Boolscher Wert)
    - StartMode (Zeichenfolge)
    - Status (Zeichenfolge)
    - SystemCreationClass (Zeichenfolge)
    - SystemName (Zeichenfolge)
- CIM\_DiagnosticTest:
  - Properties
    - Characteristics (uint16-Array)
    - IsInUse (Boolscher Wert)
    - ResourcesUsed (uint16-Array)
  - Methods
    - RunTest
    - ClearResults
    - DiscontinueTest

## Ausführen von RunTest und andere Methoden in WbemTest

Die RunTest-Methode aus der MOF-Datei sieht folgendermaßen aus:

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

Die ersten beiden Parameter kennzeichnen In-Parameter. Sie müssen den Objektpfad für beide referenzierten Objekte ermitteln. Außerdem benötigen Sie den Objektpfad für das IANet\_DiagTest-Objekt, das das RunTest-Objekt exportiert.

1. Klicken Sie im Hauptdialogfeld für den WBEM-Test auf **Connect** (Verbinden).
2. Geben Sie den zutreffenden Server/Namespaces ein. Die Namespaces IntelNCS und CimV2 werden unterstützt.
3. Klicken Sie auf die Schaltfläche **Enum Instances** (Instanzen aufzählen) des WBEM-Tests, und geben Sie `IANet_DiagTest` ein.
4. Doppelklicken Sie auf die Instanz von `IANet_DiagTest`. Der Name hat die Form `X@[AdapterGUID]`. X steht dabei für den Testnamen, und AdapterGUID kennzeichnet den Adapternamen, der dem Namensschlüssel von `IANet_EthernetAdapter` entspricht.
5. Das folgende Beispiel zeigt einen abgerufenen Objektpfad:  
`\\MYCOMPUTER\root\Cimv2:IANet_DiagTest.Name="1@{4A0CDABE-F6C3-45D0-B60D-F6E7BAFA2C2C}"`
6. Speichern Sie den Objektpfad.
7. Klicken Sie auf die Schaltfläche **Enum Instances** (Instanzen aufzählen) des WBEM-Tests, und geben Sie `IANet_EthernetAdapter` ein.
8. Doppelklicken Sie auf den zu testenden Adapter.
9. Das folgende Beispiel zeigt einen abgerufenen Objektpfad:  
`\\MYCOMPUTER\root\cimv2:IANet_EthernetAdapter.DeviceID="{4A0CDABE-F6C3-45D0-B60D-F6E7BAFA2C2C}"`
10. Speichern Sie den Objektpfad.
11. Klicken Sie auf die Schaltfläche **Enum Instances** (Instanzen aufzählen) des WBEM-Tests, und geben Sie `IANet_DiagSetting` ein.
12. Doppelklicken Sie auf die Einstellung, die der Adapter/Test-Kombination entspricht.
13. Das folgende Beispiel zeigt einen abgerufenen Objektpfad:  
`\\MYCOMPUTER\root\cimv2:IANet_DiagSetting.SettingID="1@{4A0CDABE-F6C3-45D0-B60D-F6E7BAFA2C2C}"`
14. Speichern Sie den Objektpfad.
15. Klicken Sie im Hauptdialogfeld für den WBEM-Test auf **Execute Method** (Methode ausführen).
16. Fügen Sie den `IANet_DiagTest`-Objektpfad in das Dialogfeld ein. Klicken Sie auf **OK**.
17. Wählen Sie den Test im Dropdown-Listefeld für die Methode aus.
18. Klicken Sie auf die Schaltfläche **Edit In Parameters** (In-Parameter bearbeiten).
19. Für `RunTest` sind `Setting` (Einstellung) und `SystemElement` die In-Parameter. Fügen Sie die zuvor gespeicherten Objektpfade für die Einstellung und den Adapter, und schließen Sie das Fenster.
20. Klicken Sie auf die Schaltfläche **Execute** (Ausführen).
21. Listen Sie die `IANet_DiagResult`-Klasse auf ähnliche Weise wie zuvor die In-Parameter auf.
22. Überprüfen Sie gegebenenfalls das ausgewählte Ergebnisobjekt.

## IANet\_DiagSetting

### Zweck

Instanzen von `IANet_DiagSetting` stellen spezifische Anweisungen für Runtime-Diagnosetests bereit. Die verwendeten Anweisungen gelten für alle Tests und sind an die Überklasse `CIM_DiagnosticSetting` gebunden. Dazu gehören Eigenschaften wie beispielsweise `ReportSoftErrors` und `HaltOnError`. Der Unterklasse `IANet_DiagSetting` werden keine zusätzlichen Eigenschaften hinzugefügt.

### Erstellen von Instanzen

Sie können keine Instanzen dieser Klasse erstellen.

### Löschen von Instanzen

Sie können keine Instanzen dieser Klasse löschen.

### Ändern von Eigenschaften

`UpdateInstanceAsync` wird implementiert und kann zum Einstellen von folgenden Testparametern verwendet werden: "Halt On Error" (Bei Fehler anhalten), "Report Soft Errors" (Soft-Fehler melden), "Report Status Messages" (Status-Nachrichten melden), "Quick Mode" (Schnellmodus), "Test Warning Level" (Testwarnstufe) und "Percent Of Test Coverage" (Testabdeckung in Prozent).

### Assoziationen

Eine Instanz von `IANetDiagSettingForTest` assoziiert einen `IANet_DiagTest` mit einer `IANet_DiagSetting`-Instanz.

### Nicht unterstützte Eigenschaften

Die folgenden Eigenschaften werden von NCS nicht unterstützt:

- Caption
- Description

## Methoden

Keine

## Klassenhierarchie

Für CimV2. Unbenutzte Eigenschaften und Methoden werden nicht aufgeführt.

- CIM\_ManagedElement
- CIM\_Setting:
  - Properties
  - SettingID
  - Methods (es werden keine unterstützt)
  - VerifyOKToApplyToMSE
  - ApplyToMSE
  - VerifyOKToApplyToCollection
  - ApplyToCollection
  - VerifyOKToApplyIncrementalChangeToMSE
  - ApplyIncrementalChangesToMSE
  - ApplyIncrementalChangeToCollection
- CIM\_DiagnosticSetting:
  - Key
    - SettingID (Zeichenfolge)
  - Properties
    - TestWarningLevel (uint16)
    - ReportSoftErrors (Boolscher Wert)
    - ReportStatusMessages (Boolscher Wert)
    - HaltOnError (Boolscher Wert)
    - QuickMode (Boolscher Wert)
    - PercentOfTestCoverage (uint8)

## IANet\_DiagResult

### Zweck

Instanzen von IANet\_DiagResult zeigen die Ergebnisdaten für einen bestimmten Testlauf auf einem bestimmten Adapter an. Instanzen dieser Klasse entsprechen auf identische Weise den Instanzen von IANet\_DiagTest und IANet\_DiagSetting.

### Instanzen

Instanzen von IANet\_DiagResult entsprechen den Ergebnissen für einen bestimmten Testlauf auf einem spezifischen Adapter. Für diesen Schlüssel wird dasselbe Format verwendet wie für IANet\_DiagTest und IANet\_DiagSetting. Die Instanz kann beliebige Testergebnisse speichern, und Daten, die nicht den definierten Eigenschaften entsprechen, können in der TestResults-Array-Eigenschaft untergebracht werden. Jedes Mal, wenn ein neuer Test für einen Adapter ausgeführt wird, überschreibt die neue Instanz die vorhandene Instanz der Testergebnisse für den Adapter und die Testkombination.

### Erstellen von Instanzen

Sie können keine Instanzen dieser Klasse erstellen.

### Löschen von Instanzen

Sie können keine Instanzen dieser Klasse löschen.

### Ändern von Eigenschaften

Sie können keine Instanzen dieser Klasse ändern.

### Assoziationen

Eine Instanz von IANet\_DiagResultForTest assoziiert einen IANet\_DiagTest mit einer IANet\_DiagnosticResult-Instanz.

### Nicht unterstützte Eigenschaften

Die folgenden Eigenschaften werden von NCS nicht unterstützt:

- EstimatedTimeOfPerforming
- HaltOnError
- OtherStateDescription

- ReportSoftErrors
- TestWarningLevel

## Methoden

Keine

## Klassenhierarchie

Für CimV2. Unbenutzte Eigenschaften und Methoden werden nicht aufgeführt.

- CIM\_DiagnosticResult:
  - Keys
    - DiagnosticCreationClassName (Zeichenfolge)
    - DiagnosticName (Zeichenfolge)
    - ExecutionID (Zeichenfolge)
    - DiagSystemCreationClassName (Zeichenfolge)
    - DiagSystemName (Zeichenfolge)
  - Properties
    - TimeStamp (Zeichenfolge)
    - IsPackage (Boolscher Wert)
    - TestStartTime (Datum/Zeit)
    - TestCompletionTime (Datum/Zeit)
    - TestState (uint16)
    - TestResults (Zeichenfolge)
    - PercentComplete (uint8)
- IANet\_DiagResult

[Zurück zum Seitenanfang](#)

## Registrierungseinträge

Die folgenden Einträge werden während der Installation unter **HKLM\Software\Intel\NETWORK\_SERVICES\NCS\NcsDiag** in die Registrierung aufgenommen. Diese weiter unten definierten Schlüssel und Werte steuern, wie die Diagnosetests ausgeführt werden.

Die folgende Tabelle zeigt die im Schlüssel angegebenen Werte, ihren Typ und eine kurze Erklärung bezüglich ihrer Verwendung:

Wert	Typ	Standard	Verwendung
Check Time	REG_DWORD	2 Sekunden	Zeitdauer zwischen den Überprüfungen auf Abschluss des Sender- oder Responder-Tests.
Enable	REG_DWORD	0	Aktiviert (1) oder deaktiviert (0) die Verwendung einer Ergebnis-Protokolldatei.
FileAppend	REG_DWORD	1	Ermöglicht das Anhängen (1) der Protokollergebnisdatei an die vorhandene Protokollergebnisdatei. Bei der Einstellung 0 wird die vorhandene Datei gelöscht.
LogFileName	REG_SZ	NcsDiag.log	Der Name der Protokollergebnisdatei.
MaxFileSize	REG_DWORD	0x10000	Die maximale Größe der Protokollergebnisdatei.
MaxPktsRcvd	REG_DWORD	200	Im Schnellmodus (Quick Mode in IANet_DiagSetting) wird das Senden/Empfangen eines Tests beendet, wenn die Anzahl der empfangenen Pakete den hier angegebenen Wert überschreitet.
TimeoutSndRsp	REG_DWORD	100	Der Test wird beendet, wenn die Testdauer (in Sekunden) diesen Wert überschreitet.

[Zurück zum Seitenanfang](#)

---

## Protokollieren

### Ergebnisprotokoll

Das Ergebnisprotokoll enthält hauptsächlich Informationen, die auch aus IANet\_DiagResult-Objekten ermittelt werden könnten. Der Unterschied besteht darin, dass die Informationen, die von einem CIM-Browser ermittelt werden, nur die unmittelbar letzten Ergebnisse bestimmter Tests für einen bestimmten Adapter enthalten. Nachfolgende Tests überschreiben frühere Testergebnisse. Das Ergebnisprotokoll bietet eine bequeme Lösung zum Einrichten und Anzeigen mehrfacher Durchläufe eines solchen Tests.

### Aktivieren des Ergebnisprotokolls

So wird das Ergebnisprotokoll aktiviert:

1. Stellen Sie den Wert für die Aktivierung im Registrierungsschlüssel **HKLM\Software\Intel\NETWORK\_SERVICES\NCS\NCSDiag** auf 1 ein.
2. Geben Sie den gewünschten Protokolldateinamen unter LogFileName ein, oder behalten Sie den Standardnamen **NcsDiag.log** bei.
3. Die Protokolldatei wird im Verzeichnis gespeichert, das im Wert InstalledDir angegeben ist.

[Zurück zum Seitenanfang](#)

---

## Assoziative Klassen

Assoziative Klasse	Referenz-Eigenschaft/Wert	Referenz-Eigenschaft/Wert
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

[Zurück zum Seitenanfang](#)

---

## Tests

Die implementierten Tests können auf einem einzelnen Gerät oder auf zwei Geräten ausgeführt werden. Eine detaillierte Beschreibung der Tests ist im Rahmen dieses Dokuments nicht möglich. Der CDM Provider ist als generelle Testplattform gedacht, die unabhängig von den genauen Details der Tests verwendet werden kann. Eine gewisse Abhängigkeit ist jedoch in den Code integriert. Diese Abhängigkeit wird in diesem Abschnitt erläutert.

### Tests für Einzeladapter

Die folgenden Tests werden auf einem Einzeladapter ausgeführt und erfordern keine Interaktion mit einem anderen Adapter:

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

Alle Fehlermeldungen dieser Tests resultieren aus HRESULT-Fehlercodes, die von Aufrufen an niedrigere Stapelschichten zurückgegeben werden. Die Fehlercodes werden intern als Fehlercodes gespeichert und werden erst dann in Fehlermeldungen übersetzt, wenn der Bezug auf IANet\_DiagResult-Objekte durch Auflistung aufgehoben wird oder wenn Objektaufrufe von einer Verwaltungsanwendung empfangen werden.

## Tests für zwei Adapter

Die Sender- und Responder-Tests setzen eine gegenseitige Abhängigkeit voraus. Ein Adapter (der Sender) sendet dabei Pakete an einen anderen Adapter (den Responder), der wiederum Pakete an den Sender zurücksendet und dadurch die Schleife vervollständigt. Es handelt sich dabei um dieselben Tests, die auch über Intel® PROSet ausgeführt werden können. Intel PROSet verwendet jedoch CDM nicht und lässt das gleichzeitige Ausführen von zwei Tests auf demselben Gerät nicht zu. Mit CDM sind gleichzeitige Tests unterschiedlicher Art auf demselben Gerät möglich.

### Sender/Responder-Tests

Für Sender/Responder-Tests sind zwei Intel Adapter erforderlich, wobei einer als Sender und der andere als Responder fungiert. Es handelt sich dabei um den einzigen Test, bei dem das Ausführen eines Tests auf einem zweiten Thread basiert, der so lange weiterläuft, bis der Test aufgrund bestimmter Abschlusskriterien beendet oder durch den primären Thread angehalten wird. Die Abschlusskriterien bewirken das Abschalten, sobald eine bestimmte Testzeit überschritten wird oder eine bestimmte Anzahl Pakete empfangen wurde. Beide Werte werden aus der Registrierung übernommen. Der Test kann nur dann aufgrund der Anzahl der empfangenen Pakete beendet werden, wenn Quick Mode (Schnellmodus) aktiviert wurde. Quick Mode ist eine Eigenschaft der IANet\_DiagSetting-Klasse und kann darum für jeden Adapter separat eingestellt werden. CDM-Responder antworten auf PROSet-Responder und umgekehrt.

Bei Sender/Responder-Tests werden zwei Arten von Fehlerwerten zurückgegeben. Erstens kann ein Fehlercode (HRESULT) von den unteren Schichten zurückgegeben werden. Während der Test ausgeführt wird und solange der Test nicht vorzeitig durch einen zurückgegebenen Fehlercode beendet wird, gibt der Test-Thread zweitens vorläufige und endgültige Teststatistiken der folgenden Art zurück:

- Link Status (Link-Status)
- Using Auto-Negotiation (Verwendung der automatischen Absprache)
- Collisions (Kollisionen)
- Packets Received (Empfangene Pakete)
- Packets Received Total (Empfangene Pakete insgesamt)
- Packets Sent (Gesendete Pakete)
- Transmit Oks (Übertragung OK)
- Receive Oks (Empfang OK)
- Transmit Errors (Übertragungsfehler)
- Receive Errors (Empfangsfehler)
- Collisions (Kollisionen)
- Diagnostic Phase (Diagnosephase)

---

Lesen Sie bitte alle [Einschränkungen und Haftungsausschlüsse](#).

---

[Zurück zur Inhaltsseite](#)   [Zurück zum Seitenanfang](#)

# Ausführen von Methoden in IANet\_DiagTest: Benutzerhandbuch für Intel® PRO Netzwerkkadpater – WMI und CDM Provider

## Ausführen von RunTest und andere Methoden in WbemTest

Die RunTest-Methode aus der MOF-Datei sieht folgendermaßen aus:

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

Die ersten beiden Parameter kennzeichnen In-Parameter. Sie müssen den Objektpfad für beide referenzierten Objekte ermitteln. Außerdem benötigen Sie den Objektpfad für das IANet\_DiagTest-Objekt, das das RunTest-Objekt exportiert.

1. Klicken Sie im Hauptdialogfeld für den WBEM-Test auf **Connect** (Verbinden).
2. Geben Sie den zutreffenden Server\Namespace ein. Die Namespaces IntelNCS und CimV2 werden unterstützt.
3. Klicken Sie auf die Schaltfläche **Enum Instances** (Instanzen aufzählen) des WBEM-Tests, und geben Sie IANet\_DiagTest ein.
4. Doppelklicken Sie auf die Instanz von IANet\_DiagTest. Der Name hat die Form X@[AdapterGUID]. X steht dabei für den Testnamen, und AdapterGUID kennzeichnet den Adapternamen, der dem Namensschlüssel von IANet\_EthernetAdapter entspricht.
5. Das folgende Beispiel zeigt einen abgerufenen Objektpfad:  
`\\MYCOMPUTER\root\Cimv2:IANet_DiagTest.Name="1@{4A0CDABE-F6C3-45D0-B60D-F6E7BAFA2C2C}"`
6. Speichern Sie den Objektpfad.
7. Klicken Sie auf die Schaltfläche **Enum Instances** (Instanzen aufzählen) des WBEM-Tests, und geben Sie IANet\_EthernetAdapter ein.
8. Doppelklicken Sie auf den zu testenden Adapter.
9. Das folgende Beispiel zeigt einen abgerufenen Objektpfad:  
`\\MYCOMPUTER\root\cimv2:IANet_EthernetAdapter.DeviceID="{4A0CDABE-F6C3-45D0-B60D-F6E7BAFA2C2C}"`
10. Speichern Sie den Objektpfad.
11. Klicken Sie auf die Schaltfläche **Enum Instances** (Instanzen aufzählen) des WBEM-Tests, und geben Sie IANet\_DiagSetting ein.
12. Doppelklicken Sie auf die Einstellung, die der Adapter/Test-Kombination entspricht.
13. Das folgende Beispiel zeigt einen abgerufenen Objektpfad:  
`\\MYCOMPUTER\root\cimv2:IANet_DiagSetting.SettingID="1@{4A0CDABE-F6C3-45D0-B60D-F6E7BAFA2C2C}"`
14. Speichern Sie den Objektpfad.
15. Klicken Sie im Hauptdialogfeld für den WBEM-Test auf **Execute Method** (Methode ausführen).
16. Fügen Sie den IANet\_DiagTest-Objektpfad in das Dialogfeld ein. Klicken Sie auf **OK**.
17. Wählen Sie den Test im Dropdown-Listefeld für die Methode aus.
18. Klicken Sie auf die Schaltfläche **Edit In Parameters** (In-Parameter bearbeiten).
19. Für RunTest sind Setting (Einstellung) und SystemElement die In-Parameter. Fügen Sie die zuvor gespeicherten Objektpfade für die Einstellung und den Adapter, und schließen Sie das Fenster.
20. Klicken Sie auf die Schaltfläche **Execute** (Ausführen).
21. Listen Sie die IANet\_DiagResult-Klasse auf ähnliche Weise wie zuvor die In-Parameter auf.
22. Überprüfen Sie gegebenenfalls das ausgewählte Ergebnisobjekt.

Lesen Sie bitte alle [Einschränkungen und Haftungsausschlüsse](#).

## Zusammenfassung der CIM-Klassen: Benutzerhandbuch für Intel® PRO Netzwerkkadpter – WMI und CDM Provider

Klasse	Kann erstellen?	Kann löschen?	Implementierte Methoden	Einstellbare Eigenschaften	Nicht unterstützte Eigenschaften	Instanzenzähler	Dazugehörige Assoziationsklassen
IANet_802dot1QVLANService	N	N	CreateVLAN	GRVPEEnabled, JoinTime	Description, Install Date, Started, StartMode, Status	Eine Instanz pro Gruppe oder Adapter, die bzw. der VLANs unterstützt	IANet_Device802dot1QVLANServiceImplementation, IANet_VLANFor
IANet_AdapterConfiguration	N	N	Keine	Keine	Keine	Eine Instanz pro Adapter	Diese Klasse assoziiert IANet_EthernetAdapter mit IANet_Configuration.
IANet_AdapterDevice	N	N	Keine	Keine	Keine	Eine Instanz pro Nicht-Phantomadapter	Diese Klasse assoziiert IANet_EthernetAdapter mit IANet_EthernetPCIDevice.
IANet_AdapterProtocolImplementation	N	N	Keine	Keine	Keine	Eine Instanz pro IP-Protokollendpunkt mit Anbindung an einen Adapter	Diese Klasse assoziiert IANet_EthernetAdapter mit IANet_IPProtocolEndpoint.
IANet_BootAgent	N	N	ProgramFlash ReadFlash	Keine	Caption, Description, InstallDate, Started, StartMode, Status	Eine Instanz pro Adapter, der die Boot Agent-Funktionen unterstützt	IANet_DeviceBootServiceImplementation, IANet_BootAgentConfiguration
IANet_BootAgentConfiguration	N	N	Keine	Keine	Keine	Eine Instanz pro Boot Agent	Diese Klasse assoziiert IANet_BootAgent mit IANet_Configuration.
IANet_Configuration	N	N	Keine	Keine	Keine	Eine Instanz pro Adapter, VLAN und Gruppe	IANet_AdapterConfiguration, IANet_VLANConfiguration, IANet_SettingContext
IANet_Device802dot1QVLANServiceImplementation	N	N	Keine	Keine	Keine	Eine Instanz pro Adapter oder Gruppe, der bzw. die VLANs unterstützt	Diese Klasse assoziiert IANet_EthernetAdapter mit IANet_802dot1QVLANService.
IANet_DiagTest	N	N	RunTest, DiscontinueTest, ClearResults	Keine	InstallDate, OtherCharacteristicsDescription	Eine Instanz pro Adapter/Test-Kombination	IANet_DiagTestForMSE, IANet_DiagResultForTest, IANet_DiagSettingForTest
IANet_DiagSetting	N	N	Keine	HaltOnError, ReportSoftErrors, ReportStatusMessages, QuickMode, PercentOfTestCoverage, TestWarningLevel	Caption, Description	Eine Instanz pro Adapter/Test-Kombination	IANet_DiagSettingForTest
IANet_DiagResult	N	N	Keine	Keine	EstimatedTimeOfPerforming, HaltOnError, OtherStateDescription, ReportSoftErrors, TestWarningLevel	Eine Instanz pro Adapter/Test-Kombination	IANet_DiagResultForTest, IANet_DiagResultForMSEM
IANet_EthernetAdapter	N	J	IdentifyAdapter HasVLANs IsPowerMgmtSupported GetPowerUsage SetPowerUsage GetPowerUsageOptions SetPowerUsageOptions TestCable AdvancedTestCable TestLinkSpeed	Keine	AutoSense – (dies ist als Einstellung erkennbar), ErrorCleared, OtherIdentifyingInfo, IdentifyingDescriptions, InstallDate, LastErrorCode, MaxDataSize, MaxQuiesceTime, PowerManagementCapabilities – (dies ist als Methode erkennbar), PowerManagementSupported – (dies ist als Methode	Eine Instanz pro installiertem Adapter, Phantomadapter bzw. installierter Gruppe mit Intel® PROSet-Unterstützung	IANet_AdapterProtocolImplementation, IANet_AdapterDevice, IANet_AdapterConfiguration, IANet_TeamedMemberAdapter, IANet_NetworkVirtualAdapter, IANet_Device802dot1QVLANServiceImplementation, IANet_DeviceBootServiceImplementation

					erkennbar), PowerOnHours , ShortFramesReceived, SymbolErrors, TotalPowerOnHours		
IANet_EthernetPCIDevice	N	N	Keine	Keine	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	Eine Instanz pro PCI-Karte, die ein von Intel PROSet unterstützter Ethernet-Adapter ist	IANet_AdapterDevice
IANet_IPProtocolEndpoint	N	N	Keine	Keine	Caption, Description, InstallDate, NameFormat, OtherTypeInfo, Status	Eine Instanz pro Bindung des IP-Protokollstapels an einen von Intel unterstützten Endpunkt	IANet_AdapterProtocolImplementation, IANet_VLANProtocolDependency
IANet_NetService	N	N	GetSessionHandle, Apply, ReleaseSessionHandle, Cancel	Keine	Caption, Description, Install Date, Started, Start Mode, Status	Genau eine Instanz	Keine
IANet_NetworkVirtualAdapter	N	N	Keine	Keine	Keine	Eine Instanz pro Gruppe	Diese Klasse assoziiert IANet_TeamOfAdapters mit IANet_EthernetAdapter.
IANet_PCIDevice	N	N	Keine	Keine	AdditionalAvailability, Capabilities, CapabilityDescriptions, Caption, DeviceSelectTiming, ErrorCleared, ErrorDescription, IdentifyingDescription, InstallDate, LastErrorCode, MaxNumberController, MaxQuiesceTime, Name, OtherIdentifyingInfo, PowerManagementCapabilities, PowerManagementSupported, PowerOnHours, ProtocolDescription, ProtocolSupported, SelfTestEnabled, TimeOfLastReset, TotalPowerOnHours	Eine Instanz pro PCI-Karte, die als Netzwerkgerät im System eingesetzt wird	
IANet_Setting	N	N	Keine	Keine	SettingID	Dies ist eine abstrakte Klasse.	IANet_SettingContext
IANet_SettingContext	N	N	Keine	Keine	Keine	Eine Instanz pro Einstellung	Diese Klasse assoziiert IANet_Setting mit IANet_Configuration.
IANet_SettingInt	N	N	Keine	CurrentValue	SettingID	Eine Instanz pro Ganzzahleinstellung	IANet_SettingContext
IANet_SettingMultiSelection	N	N	Keine	CurrentValue	SettingID	Eine Instanz pro Mehrfachauswahl-Einstellung	IANet_SettingContext
IANet_SettingSlider	N	N	Keine	CurrentValue	SettingID	Eine Instanz pro Reglereinstellung	IANet_SettingContext

IANet_SettingString	N	N	Keine	CurrentValue	SettingID	Eine Instanz pro Zeichenfolgeeinstellung	IANet_SettingContext
IANet_TeamedMemberAdapter	J	J	Keine	AdapterFunction	PrimaryAdapter, ScopeOfBalancing	Eine Instanz pro Adapter, der einer Gruppe angehört	Diese Klasse assoziiert IANet_TeamOfAdapters mit IANet_EthernetAdapter.
IANet_TeamOfAdapters	J	J	TestSwitchConfiguration	TeamingMode	Install Date, Status	Eine Instanz pro Gruppe	IANet_NetworkVirtualAdapter, IANet_TeamedMemberAdapter
IANet_VLAN	N	J	Keine	VLANNumber, Caption	Description, Install Date, StartMode, Status	Eine Instanz pro VLAN	IANet_VLANFor
IANet_VLANConfiguration	N	N	Keine	Keine	Keine	Eine Instanz pro VLAN	Diese Klasse assoziiert IANet_VLAN mit IANet_Configuration.
IANet_VLANFor	N	N	Keine	Keine	Keine	Eine Instanz pro VLAN	Diese Klasse assoziiert IANet_VLAN mit IANet_802dot1QVLANService.
IANet_VLANProtocolDependency	N	N	Keine	Keine	Keine	Eine Instanz pro VLAN	Diese Klasse assoziiert IANet_VLAN mit IANet_IPProtocolEndpoint.

Lesen Sie bitte alle [Einschränkungen und Haftungsausschlüsse](#).

[Zurück zur Inhaltsseite](#) [Zurück zum Seitenanfang](#)

# Intel Software-Lizenzvereinbarung (Endversion): Benutzerhandbuch für Intel® PRO Netzwerkadapter – WMI und CDM Provider

---

## **WICHTIG – BITTE LESEN SIE DIESE VEREINBARUNG VOR DEM KOPIEREN, INSTALLIEREN ODER VERWENDEN DER SOFTWARE.**

Verwenden oder laden Sie diese Software und alle dazugehörigen Elemente (nachfolgend als "Software" bezeichnet) erst dann, wenn Sie die folgenden Bestimmungen und Bedingungen durchgelesen haben. Durch das Laden oder Verwenden der Software erklären Sie sich mit den Bedingungen dieser Vereinbarung einverstanden. Wenn Sie mit den Bestimmungen und Bedingungen nicht einverstanden sind, dürfen Sie die Software nicht installieren oder verwenden.

### **LIZENZEN:**

- Wenn Sie ein Netzwerkadministrator sind, trifft die "Lizenzvereinbarung für Sites" auf Sie zu.
- Wenn Sie ein Endverbraucher sind, trifft die "Lizenzvereinbarung für Einzelbenutzer" auf Sie zu.
- Wenn Sie ein OEM-Hersteller sind, trifft die "Lizenzvereinbarung für OEM-Hersteller" auf Sie zu.

**Lizenzvereinbarung für Sites.** Sie dürfen die Software auf Computer Ihrer Organisation zum firmeninternen Gebrauch kopieren und von der Software unter folgenden Bedingungen eine angemessene Anzahl an Sicherungskopien erstellen:

1. Diese Software ist nur für den Gebrauch mit Intel Komponenten lizenziert. Der Gebrauch dieser Software mit nicht von Intel hergestellten Komponenten fällt nicht unter diese Lizenzvereinbarung.
2. Kein Teil der Software darf, außer wie in dieser Vereinbarung festgelegt, kopiert, modifiziert, vermietet, verkauft, verteilt oder übertragen werden, und Sie erklären sich damit einverstanden, ein unberechtigtes Kopieren dieser Software zu verhindern.
3. Sie dürfen die Software nicht dekompileieren, disassemblieren oder in sonstiger Weise zurückentwickeln.
4. Sie dürfen die Software keinem anderen Anwender im Rahmen einer Unterlizenz zur Verfügung stellen. Die Software darf jeweils nur von einem Anwender benutzt werden.
5. Die Software kann Teile umfassen, die zusätzlichen, hier nicht genannten Bedingungen unterliegen und in einer eigenen Lizenz ausgeführt sind, die den jeweiligen Teilen beiliegt.

**Lizenzvereinbarung für Einzelbenutzer.** Sie dürfen die Software auf einen einzelnen Computer für Ihren persönlichen, nicht kommerziellen Gebrauch kopieren und von der Software unter folgenden Bedingungen eine Sicherungskopie erstellen:

1. Diese Software ist nur für den Gebrauch mit Intel Komponenten lizenziert. Der Gebrauch dieser Software mit nicht von Intel hergestellten Komponenten fällt nicht unter diese Lizenzvereinbarung.
2. Kein Teil der Software darf, außer wie in dieser Vereinbarung festgelegt, kopiert, modifiziert, vermietet, verkauft, verteilt oder übertragen werden, und Sie erklären sich damit einverstanden, ein unberechtigtes Kopieren dieser Software zu verhindern.
3. Sie dürfen die Software nicht dekompileieren, disassemblieren oder in sonstiger Weise zurückentwickeln.
4. Sie dürfen die Software keinem anderen Anwender im Rahmen einer Unterlizenz zur Verfügung stellen. Die Software darf jeweils nur von einem Anwender benutzt werden.
5. Die Software kann Teile umfassen, die zusätzlichen, hier nicht genannten Bedingungen unterliegen und in einer eigenen Lizenz ausgeführt sind, die den jeweiligen Teilen beiliegt.

**Lizenzvereinbarung für OEM-Hersteller.** Sie dürfen die Software nur vervielfältigen und verteilen, sofern sie einen integralen Bestandteil oder eine essentielle Komponente Ihres Produkts darstellt bzw. als selbständige Software-Wartungsaktualisierung für die derzeitigen Endbenutzer ihres Produkts (ausgenommen sind andere selbständige Produkte) gedacht ist. Dabei gelten die folgenden Bedingungen:

1. Diese Software ist nur für den Gebrauch mit Intel Komponenten lizenziert. Der Gebrauch dieser Software mit nicht von Intel hergestellten Komponenten fällt nicht unter diese Lizenzvereinbarung.
2. Kein Teil der Software darf, außer wie in dieser Vereinbarung festgelegt, kopiert, modifiziert, vermietet, verkauft, verteilt oder übertragen werden, und Sie erklären sich damit einverstanden, ein unberechtigtes Kopieren dieser Software zu verhindern.
3. Sie dürfen die Software nicht dekompileieren, disassemblieren oder in sonstiger Weise zurückentwickeln.
4. Sie dürfen die Software nur aufgrund einer schriftlichen Lizenzvereinbarung an Ihre Kunden verteilen. Diese Lizenzvereinbarung kann mit Aufbrechen des Packungssiegels wirksam werden. Durch diese Lizenz müssen zumindest die Eigentumsrechte von Intel an der Software geschützt werden.
5. Die Software kann Teile umfassen, die zusätzlichen, hier nicht genannten Bedingungen unterliegen und in einer eigenen

Lizenz ausgeführt sind, die den jeweiligen Teilen beiliegt.

**KEINE ANDEREN RECHTE.** Intel überträgt weder explizit noch implizit irgendwelche Rechte oder Lizenzen an Sie hinsichtlich urheberrechtlich geschützter Informationen bzw. irgendwelcher Patent-, Urheber, Mask-Work-, Marken-, Branchengeheimnisse oder anderer Rechte an geistigem Eigentum, das Eigentum von Intel ist oder von Intel kontrolliert wird. Eventuelle Ausnahmen sind in dieser Vereinbarung aufgeführt.

**EIGENTUMSRECHTE BEZÜGLICH DER SOFTWARE UND URHEBERRECHTE.** Das Eigentum aller Kopien der Software verbleibt bei Intel oder seinen Zulieferern. Die Software ist urheberrechtlich und durch die Gesetze der Vereinigten Staaten von Amerika und anderer Länder sowie durch internationale Abkommen geschützt. Urheberrechtshinweise dürfen nicht aus der Software entfernt werden. Intel kann die Software oder darin erwähnte Elemente jederzeit und ohne Vorankündigung ändern. Intel ist jedoch nicht dazu verpflichtet, die Software zu unterstützen oder zu aktualisieren. Sofern nicht ausdrücklich erwähnt, gewährt Intel keine ausdrücklichen oder stillschweigenden Rechte an Intel Patenten, Urheberrechten, Marken oder andere Rechte an geistigem Eigentum. Sie dürfen die Software nur übertragen, wenn sich der Empfänger uneingeschränkt mit diesen Bedingungen einverstanden erklärt und Sie keine Kopien der Software behalten.

**BESCHRÄNKTE GARANTIE FÜR DATENTRÄGER.** Wenn die Software von Intel auf einem Datenträger geliefert wurde, garantiert Intel für einen Zeitraum von neunzig Tagen nach der Lieferung durch Intel, dass der Datenträger frei von Materialschäden ist. Falls der Datenträger beschädigt ist, können Sie ihn an Intel zurückgeben. Intel entscheidet in diesem Fall, ob der Datenträger ersetzt oder ob die Software auf einem anderen Weg an Sie ausgeliefert wird.

**AUSSCHLUSS ANDERER GARANTIE. MIT AUSNAHME DER OBEN GENANNTEN GARANTIE WIRD DIE SOFTWARE IM VORLIEGENDEN ZUSTAND UND OHNE AUSDRÜCKLICHE ODER STILLSCHWEIGENDE GARANTIE JEDER ART, EINSCHLIESSLICH UNTER ANDEREM DER EINHALTUNG DER RECHTE VON DRITTEN ODER DER EIGNUNG FÜR EINEN BESTIMMTEN ZWECK, GELIEFERT.** Intel übernimmt keine Garantie oder Verantwortung für die Richtigkeit oder Vollständigkeit der Informationen, des Texts, der Grafiken, Verknüpfungen oder anderer Elemente, die in der Software enthalten sind.

**HAFTUNGSBESCHRÄNKUNG. UNTER KEINEN UMSTÄNDEN HAFTEN INTEL ODER DRITTHERSTELLER FÜR IRGENDWELCHE SCHÄDEN (EINSCHLIESSLICH UNTER ANDEREM ALLER GEWINNVERLUSTE, BETRIEBSUNTERBRECHUNGEN ODER INFORMATIONSVERLUSTE), DIE DURCH DIE VERWENDUNG ODER DIE NICHTVERWENDBARKEIT DER SOFTWARE AUFTRETEN; DIES GILT AUCH FÜR FÄLLE, IN DENEN INTEL AUF DIE MÖGLICHKEIT SOLCHER SCHÄDEN HINGEWIESEN WURDE. IN EINIGEN GERICHTSBARKEITEN IST DER AUSSCHLUSS ODER DIE EINSCHRÄNKUNG DER HAFTUNG FÜR STILLSCHWEIGENDE GEWÄHRLEISTUNGEN ODER FÜR MITTELBARE BZW. BEILÄUFIG ENTSTANDENE SCHÄDEN NICHT ZULÄSSIG, SO DASS DIE OBEN AUFGEFÜHRTE EINSCHRÄNKUNG MÖGLICHERWEISE AUF SIE NICHT ZUTRIFFT. SIE HABEN UNTER UMSTÄNDEN NOCH ANDERE RECHTE, DIE JE NACH GERICHTSBARKEIT VARIIEREN KÖNNEN.**

**BEENDIGUNG DIESER VEREINBARUNG.** Diese Vereinbarung kann von Intel jederzeit beendet werden, wenn Sie die genannten Bestimmungen nicht einhalten. Wird die Vereinbarung beendet, sind Sie dazu verpflichtet, die Software entweder umgehend zu löschen oder alle Kopien der Software an Intel zurückzugeben.

**ANWENDBARES RECHT.** Ansprüche, die aufgrund dieser Vereinbarung geltend gemacht werden, unterliegen den Gesetzen von Kalifornien. Die bei Gesetzeskonflikten anwendbaren Prinzipien sowie die UN-Konvention über Verträge für den Warenverkauf bleiben unberücksichtigt. Sie dürfen die Software nicht exportieren, wenn der Export gegen geltende Exportgesetze und -bestimmungen verstößt. Intel ist an keine anderen Vereinbarungen gebunden, es sei denn, sie liegen in schriftlicher Form vor und wurden von einem autorisierten Repräsentanten von Intel unterzeichnet.

**EINGESCHRÄNKTE RECHTE DER REGIERUNG.** Die Software wird mit "EINGESCHRÄNKTE RECHTEN" geliefert. Die Verwendung, Vervielfältigung oder Veröffentlichung durch Regierungsbehörden unterliegt den Bestimmungen, die in FAR52.227-14 und DFAR252.227-7013ff oder einer entsprechenden Aktualisierung aufgeführt sind. Die Verwendung dieser Software durch US-Regierungsbehörden erfolgt unter Anerkennung der diesbezüglichen Eigentumsrechte von Intel. Auftraggeber oder Hersteller ist Intel.

---

Lesen Sie bitte alle [Einschränkungen und Haftungsausschlüsse](#).

---

[Zurück zur Inhaltsseite](#)   [Zurück zum Seitenanfang](#)

# Support: Benutzerhandbuch für Intel® PRO Netzwerkadapter – WMI und CDM Provider

---

## Web- und Internet-Sites

<http://www.dell.com>

---

## Kundensupport-Techniker

Setzen Sie sich bitte mit der Dell Computer Corporation in Verbindung, um technische Hilfe zu erhalten, wenn die Vorschläge zur Fehlerbehebung in diesem Dokument das Problem nicht lösen (siehe den Hilfeabschnitt in der Systemdokumentation).

### Bevor Sie anrufen...

Sie müssen an Ihrem Computer sitzen und die Software muss laufen. Legen Sie auch die Produktdokumentation bereit.

Der Techniker benötigt von Ihnen möglicherweise die folgenden Informationen:

- Ihre Adresse und Telefonnummer
  - Den Namen und die Modellnummer des Produkts, zu dem Sie Fragen haben
  - Die Seriennummer und die Servicekennzeichnung des Produkts
  - Die Namen und die Versionsnummern der Software, die Sie für den Betrieb des Produkts verwenden
  - Den Namen und die Versionsnummer des verwendeten Betriebssystems
  - Den Computertyp (Hersteller und Modellnummer)
  - Erweiterungsplatinen und zusätzlich im Computer eingebaute Karten
  - Die Speicherkapazität des Computers
- 

Lesen Sie bitte alle [Einschränkungen und Haftungsausschlüsse](#).

---