

# Guide de l'utilisateur des fournisseurs WMI et CDM pour cartes réseau Intel® PRO

---

Les informations qui suivent ont été communiquées par le fournisseur du périphérique mentionné. Elles n'ont pas été vérifiées par Dell et sont soumises aux [restrictions et clauses exonératoires de responsabilité](#) ci-dessous.

---

- [Introduction](#)
  - [WMI](#)
  - [Fonctionnalités principales](#)
  - [Fichiers installés](#)
  - [Sécurité](#)
  - [Espace de noms et contexte](#)
  - [Paramètres régionaux et localisation](#)
  - [Rapports d'erreurs](#)
  - [Schéma principal](#)
  - [Schéma de la carte Ethernet](#)
  - [Schéma des paramètres](#)
  - [Schéma de l'association](#)
  - [Schéma du réseau local virtuel](#)
  - [Obtention de la configuration actuelle](#)
  - [Mise à jour de la configuration](#)
  - [Notifications d'événements](#)
  - [Requêtes WQL optimisées](#)
  - [Diagnostics](#)
  - [Exécution de méthodes dans IANet DiagTest](#)
  - [Récapitulatif des classes CIM](#)
  - [Licence du logiciel](#)
  - [Assistance à la clientèle](#)
- 

**Les informations contenues dans ce document peuvent être modifiées sans préavis.**

**© 2003 Intel Corporation. Tous droits réservés.**

Marques commerciales utilisées dans ce texte : *Dell* et le logo *DELL* sont des marques commerciales de Dell Computer Corporation ; *Intel* est une marque commerciale ou déposée d'Intel Corporation ou de ses filiales aux États-Unis et dans d'autres pays.

\* Les autres marques et noms de produit utilisés dans ce document font référence aux entités revendiquant les marques et les noms de leurs produits. Intel renonce à tout intérêt propriétaire dans les marques et noms de produits qui ne lui appartiennent pas.

---

## **Restrictions et clauses exonératoires de responsabilité**

Les informations contenues dans le présent document, y compris l'ensemble des instructions, avertissements ainsi que les approbations et les homologations de réglementation, ont été soumises par le fournisseur et n'ont été ni vérifiées ni testées de façon indépendante par Dell. Dell décline toute responsabilité en cas de dommage résultant du respect ou du non respect de ces instructions.

Toutes les déclarations ou revendications relatives aux propriétés, capacités, vitesses ou qualifications du dispositif mentionné dans le présent document sont communiquées par le revendeur et non pas par Dell. Dell n'est pas en mesure de certifier l'exactitude de ces déclarations, d'en garantir l'exhaustivité ou de les justifier. Veuillez vous adresser à votre fournisseur pour toute question relative à ces déclarations.

---

# Introduction : Guide de l'utilisateur des fournisseurs WMI et CDM pour cartes réseau Intel® PRO

---

## Présentation

Bienvenue dans le *Guide de l'utilisateur des fournisseurs WMI et CDM pour cartes réseau Intel® PRO*. Ce document décrit l'affichage externe des fournisseurs WMI et CDM pour cartes réseau Intel PRO. Le fournisseur WMI (Windows Management Interface) est un bloc de configuration réseau des services de configuration réseau (NCS). Il permet de déployer et de gérer toutes les technologies réseau Intel pour station finale en utilisant des méthodes standard de l'industrie. Le fournisseur Intel PRO CDM (Common Diagnostic Model) est une interface API supérieure conforme aux normes CIM 2.5 et WMI. Au niveau de l'interface inférieure, le fournisseur CDM met en oeuvre une interface client dans les couches inférieures de la pile logicielle PROSet. Pour cette raison, tous les mécanismes PROSet pour l'intégrité des données sont conservés.

Les fournisseurs WMI et CDM sont des jeux de composants logiciels qui permettent de mettre en oeuvre des classes réseau Intel WMI. Ces classes sont basées sur le schéma CIM DTMF (Desktop Management Task Force) version 2.5.

Ce document ne reprend pas les informations contenues dans les fichiers MOF (Managed Object Format) fournis avec ce produit (p. ex., des détails sur la signification des attributs sont disponibles dans les descriptions des attributs MOF).

Ce document décrit comment une application WMI telle qu'Intel PROSet utilise les classes pour configurer le réseau d'un système et pour tester une carte d'interface réseau Intel. Le lecteur doit être familiarisé avec les API WMI et le kit de développement WMI SDK (disponible sur le site <http://www.microsoft.com/>).


[Retour au début de la page](#)

---

## Documents associés

Les documents suivants peuvent être utilisés pour mieux comprendre la technologie WMI :

- CIM Schema (Schéma CIM) version 2.0, 2.2, publié par Desktop Management Task Force (DMTF). Disponible à l'adresse <http://www.dmtf.org>.
- Microsoft Windows Management Interface (et autres informations de gestion). Disponible à l'adresse [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), initiative de DMTF. Disponible à l'adresse <http://www.dmtf.org/standards/wbem>.
- Kit de développement WMI (Implémentation Microsoft CIM). Disponible à l'adresse <http://msdn.microsoft.com/downloads/>.
- System Diagnostic Model (Modèle de diagnostic de système), fiche technique de DFTM. Disponible à l'adresse <http://www.dmtf.org/standards/documents/CIM/DSP0138.pdf>.

 **ATTENTION** : ce produit contient des informations qui pourraient être utilisées pour attaquer et/ou désactiver des ordinateurs ou des réseaux. Une connaissance approfondie des fonctionnalités de la sécurité des systèmes d'exploitation Microsoft est requise avant toute mise en oeuvre de ce produit. Les développeurs et les utilisateurs sont invités à contacter Microsoft pour toute question relative à la sécurité qu'ils pourraient avoir, avant de mettre ce produit en oeuvre dans un environnement de production.

---

Veillez lire toutes les [restrictions et clauses exonératoires de responsabilité](#).

---

[Retour à la table des matières](#)   [Retour au début de la page](#)

# WMI : Guide de l'utilisateur des fournisseurs WMI et CDM pour cartes réseau Intel® PRO

[Présentation](#)

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

---

## Présentation

Web-Based Enterprise Management (WBEM) est une initiative de DMTF (Desktop Management Task Force) prévue pour fournir une méthode économique et standardisée de gestion des stations finales aux gestionnaires de systèmes d'entreprise. L'initiative WBEM englobe une multitude de tâches, allant de la simple configuration d'une station de travail à la gestion complète de plates-formes multiples au sein d'une entreprise. Le modèle CIM (Common Information Model), un modèle de données extensible pour la représentation des objets existant dans des environnements de gestion typiques, et le langage MOF (Managed Object Format) pour la définition et le stockage des données modélisées, sont les éléments centraux de cette initiative.

Windows Management Instrumentation (WMI) est une mise en oeuvre de l'initiative WBEM pour les plates-formes Microsoft\* Windows\*.

WMI est constitué de trois composants principaux :

- Les composants principaux — Ces composants font partie du système d'exploitation. Ils sont requis pour le fonctionnement des applications compatibles WMI, et doivent être installés pour pouvoir utiliser le kit de développement logiciel SDK.
- SDK — Le kit de développement logiciel SDK contient les outils qui permettent de parcourir et d'étendre le schéma WMI, de créer les fournisseurs, et d'enregistrer et utiliser les événements WMI. Il fournit également de la documentation utile pour développer des applications qui utiliseront le WMI. Le SDK est installé au cours du processus d'installation Microsoft Platform SDK, et il est pris en charge sur Windows NT4 avec le Service Pack 4 ou 5, Windows 2000, Windows Me, Windows XP et Windows Server 2003.
- Outils — Les outils Microsoft WMI fournissent aux développeurs les outils nécessaires à la création d'une toute nouvelle génération d'applications et de solutions de gestion. Ce composant comporte une quantité de documents et d'outils qui vous guideront à travers le processus d'accès aux données de gestion à partir de WMI.

L'architecture WMI est constituée des composants suivants :

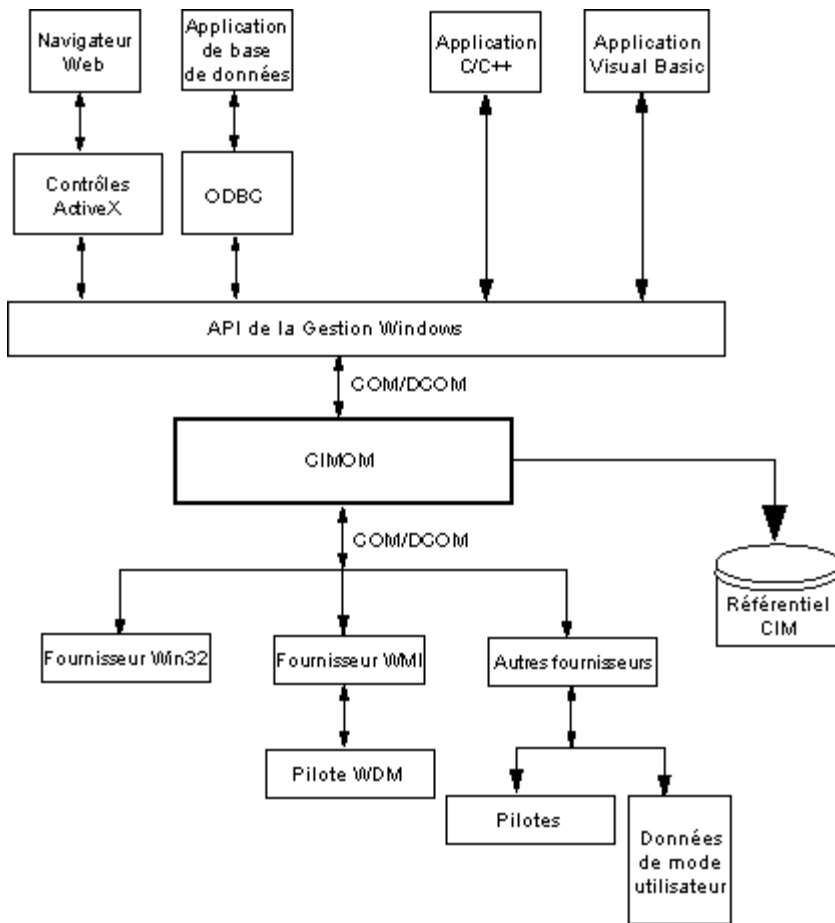
- Applications de gestion
- Objets gérés
- Fournisseurs
- Infrastructure de gestion (constituée de la Gestion Windows et de son référentiel)
- API de la Gestion Windows (qui utilise COM/DCOM pour permettre aux fournisseurs et aux applications de gestion de communiquer avec l'infrastructure de la Gestion Windows)

Les applications de gestion traitent ou affichent les données des objets gérés qui sont des composants logiques ou physiques de l'entreprise. Ces composants sont modélisés en utilisant CIM, et les applications y accèdent via la Gestion Windows. Les fournisseurs utilisent l'interface de programmation de la Gestion Windows pour lui fournir les données des objets gérés, traiter les requêtes des applications et générer des notifications d'événements.

L'infrastructure de gestion est constituée de la Gestion Windows (pour le traitement des communications entre les applications de gestion et les fournisseurs) et du référentiel de la Gestion Windows (pour le stockage des données). Le référentiel de la Gestion Windows contient les données de gestion statiques. Les données dynamiques sont générées à la demande par les fournisseurs. Les données sont placées dans le référentiel en utilisant le compilateur de langage MOF ou l'API de la Gestion Windows.

Les applications et les fournisseurs communiquent via la Gestion Windows en utilisant l'API de la Gestion Windows, qui offre des services tels que les notifications d'événements ou le traitement des requêtes.

Le diagramme suivant présente les relations et les interactions entre les composants de l'architecture WMI :



[Retour au début de la page](#)

## Schéma CIM (Common Information Model)

Le schéma CIM (Common Information Model) présente un affichage cohérent et unifié de tous les types d'objets logiques et physiques existant dans un environnement géré. Les objets gérés sont représentés à l'aide de constructions orientées objet, telles que des classes. Les classes comportent des propriétés qui décrivent les données et les méthodes décrivant un comportement. Le CIM a été conçu par le DMTF pour être indépendant des systèmes d'exploitation et des plates-formes. La technologie WBEM comprend une extension du CIM pour les plates-formes des systèmes d'exploitation Microsoft Windows. Reportez-vous au schéma CIM sur le site du DMTF pour obtenir davantage d'informations.

Le CIM définit trois niveaux de classes :

- Les classes représentant les objets gérés applicables à toutes les zones de gestion. Ces classes fournissent un vocabulaire de base pour analyser et décrire les systèmes gérés, et elles font partie de ce que l'on appelle le modèle principal.
- Les classes représentant les objets gérés applicables à une zone de gestion spécifique, mais indépendantes d'une technologie ou d'une mise en oeuvre particulière. Ces classes font partie de ce que l'on appelle le modèle commun.
- Les classes représentant les objets gérés et qui sont des ajouts technologiques au modèle commun. Ces classes s'appliquent généralement à des plates-formes spécifiques, comme UNIX ou l'environnement Microsoft Win32, et elles font partie de ce que l'on appelle le modèle étendu.

Toutes les classes peuvent être liées par héritage, où une classe enfant comporte des données et des méthodes de sa classe parent. Les relations de l'héritage ne sont généralement pas visibles par les applications de gestion qui les utilisent, et les applications ne doivent pas nécessairement connaître la hiérarchie de l'héritage. Les hiérarchies des classes peuvent être obtenues en utilisant les applications incluses dans les outils WMI (visitez le site <http://www.microsoft.com> pour davantage d'informations sur les outils WMI).

La Gestion Windows prend également en charge les classes d'association. Les classes d'association lient deux classes différentes pour modéliser une relation définie par l'utilisateur, et elles sont visibles par les applications de gestion. La Gestion Windows définit

les classes d'association pour prendre en charge les classes système. Les développeurs tiers peuvent également définir des classes d'association pour leur environnement de gestion.

WBEM prend en charge le concept des schémas pour grouper les classes et les instances utilisées dans un environnement de gestion particulier. Platform SDK comprend deux schémas : le schéma CIM et le schéma Microsoft Win32. Le schéma CIM contient les définitions des classes pour les deux premiers niveaux du CIM. Ces classes représentent des objets gérés faisant partie de tout environnement de gestion, indépendamment de la plate-forme. Le schéma Win32 contient les définitions des classes pour les objets gérés faisant partie d'un environnement Win32 typique.

Pour plus d'informations sur CIM, visitez le site <http://www.dmtf.org>.

---

Veillez lire toutes les [restrictions et clauses exonératoires de responsabilité](#).

---

[Retour au sommaire](#)   [Retour au début de la page](#)

# Fonctionnalités principales : Guide de l'utilisateur des fournisseurs WMI et CDM pour cartes réseau Intel® PRO

[Fonctionnalités du fournisseur WMI NCS](#)

[Fonctionnalités du fournisseur CDM](#)

---

## Fonctionnalités du fournisseur WMI NCS

Les fonctionnalités principales du fournisseur WMI sont les suivantes :

### Fonctionnalités de carte

- Énumérer toutes les cartes physiques prises en charge par Intel® PROSet.
- Énumérer les paramètres d'une carte installée.
- Ajouter/supprimer/mettre à jour les paramètres d'une carte installée.
- Obtenir les informations du périphérique physique d'une carte.
- Obtenir les informations du connecteur système d'une carte.
- Obtenir les paramètres de protocole IPv4 pour une carte.
- Mettre à jour et modifier l'agent d'amorçage d'une carte et les paramètres associés.
- Désinstaller une carte.

### Fonctionnalités d'association

- Énumérer toutes les associations prises en charge par Intel® PROSet.
- Créer/supprimer une association de cartes.
- Ajouter/supprimer/mettre à jour les paramètres d'une association.
- Ajouter/supprimer des cartes membres d'une association.
- Obtenir les paramètres de protocole IPv4 pour une association.

### Fonctionnalités de réseau local virtuel

- Énumérer les réseaux locaux virtuels d'une carte ou d'une association.
- Créer/supprimer des réseaux locaux virtuels sur une carte physique ou une association de cartes.
- Ajouter/supprimer/mettre à jour les paramètres d'un réseau local virtuel.
- Obtenir les paramètres de protocole IPv4 pour une association.

### Fonctionnalités de notifications d'événements

- Permet au client d'enregistrer les événements suivants :
  - Événements d'état d'une carte.
  - Événements de configuration d'une carte.
  - Événements de session.
  - Événements d'état d'une association.
  - Événements de configuration d'une association.
  - Événements de configuration d'un réseau local virtuel.

[Retour au début de la page](#)

---

## Fonctionnalités du fournisseur CDM

Les fonctionnalités principales du fournisseur CDM sont les suivantes :

- Exécuter des tests, arrêter les tests et effacer les résultats sans dépendance au type de test de diagnostic.
- L'utilisation d'une classe de paramètre générique devrait permettre de contrôler les tests d'une manière non anticipée par le

logiciel CDM lui-même.

- Le fournisseur CDM est utilisé pour les cartes uniquement.
  - L'utilisation d'une classe de résultat générique sépare des messages de résultat spécifiques du code du fournisseur CDM.
  - Les entrées du registre contrôlent l'exécution du fournisseur.
  - Les résultats des tests sont enregistrés dans un fichier journal.
- 

Veillez lire toutes les [restrictions et clauses exonératoires de responsabilité](#).

---

[Retour à la table des matières](#)   [Retour au début de la page](#)

# Fichiers installés : Guide de l'utilisateur des fournisseurs WMI et CDM pour cartes réseau Intel® PRO

[Fichiers WMI](#)

[Fichiers du fournisseur CDM](#)

---

## Fichiers WMI

### Fichiers exécutables

Les fichiers exécutables du fournisseur WMI sont les suivants :

- **NcsWmiCo.exe** — Le fournisseur principal. Met en oeuvre IANet\_NetService et les classes d'événement principales.
- **NcsWmilm.exe** — Le fournisseur d'instances et de méthodes. Met en oeuvre le schéma de la carte Ethernet, le schéma de l'association, le schéma des paramètres et le schéma du réseau local virtuel.
- **NcsWmiEv.exe** — Le fournisseur d'événements. Met en oeuvre les événements de carte, d'association et de réseau local virtuel.

### Fichiers MOF

Il y a des fichiers MOF séparés pour les données indépendantes et spécifiques à une langue. De même, il existe des fichiers MOF pour les espaces de noms IntelNCS et CIMV2. Reportez-vous aux sections [Paramètres régionaux et localisation](#) et [Rapports d'erreurs](#) pour davantage de détails.

Les fichiers MOF pour un espace de noms IntelNCS sont les suivants :

- **NcsCmLn.mof** — Classes de base CIM dont dépendent les classes NCS.
- **NcsCmEnu.mfl** — Version anglaise (États-Unis) des classes de base CIM.
- **NcsCoLn.mof** — Classes principales mises en oeuvre par le fournisseur principal.
- **NcsCoEnu.mfl** — Amendements textuels en anglais (États-Unis) aux classes principales.
- **NcslaLn.mof** — Classes pour les cartes, les associations et les réseaux locaux virtuels IEEE 802.3.
- **NcslaEnu.mfl** — Amendements textuels en anglais (États-Unis) aux classes principales 802.3.

Les fichiers MOF pour un espace de noms CIMV2 sont les suivants :

- **C2CmLn.mof** — Classes de base CIM dont dépendent les classes NCS.
- **C2CmEnu.mfl** — Version anglaise (États-Unis) des classes de base CIM.
- **C2CoLn.mof** — Classes principales mises en oeuvre par le fournisseur principal.
- **C2CoEnu.mfl** — Amendements textuels en anglais (États-Unis) aux classes principales.
- **C2laLn.mof** — Classes pour les cartes, les associations et les réseaux locaux virtuels IEEE 802.3.
- **C2laEnu.mfl** — Amendements textuels en anglais (États-Unis) aux classes principales 802.3.

### Fichiers de ressources

Les fichiers de ressources du fournisseur WMI sont les suivants :

- **ENU\_8023.dll** — Ressource 8023 en anglais (États-Unis).
- **ENU\_NWRC.dll** — Ressources WMI en anglais (États-Unis) pour les fournisseurs principaux.
- **ENU\_NWR.dll** — Ressources WMI en anglais (États-Unis) pour les fournisseurs 8023.

Les autres fichiers de ressources localisés peuvent être chargés à la demande. Le modèle général pour les noms des DLL de ressources localisées est "\_mwr.dll", représentant le code de langue pour la localisation (p. ex., FRA pour le français standard).



## Fichiers du fournisseur CDM

### Fichiers exécutables

Les fichiers exécutables du fournisseur CDM sont les suivants :

- **Ncsdiag.exe** est le fichier exécutable principal pour les diagnostics CDM. Il est conforme à la spécification de l'interface Microsoft\* WMI, et est accédé en tant que serveur COM extra-processus.
- Autres fichiers exécutables de la pile logicielle Intel® PROSet.

### Fichiers MOF

Les fichiers **.mof** principaux ne sont pas livrés avec le produit, mais ils sont compilés en composants dépendants ou indépendants de la langue, conformément au modèle de globalisation Microsoft\* Windows\* Management Instrumentation. Pour davantage d'informations, reportez-vous au chapitre du kit de développement WMI (composant du kit Platform SDK) de Microsoft\* portant sur la localisation WMI. Faites particulièrement attention à la section *Compiling Localized MOF Files* (Compilation des fichiers MOF localisés).

La suppression d'un fichier **.mof** (DNcsCdmN.mof) supprime les définitions de classes dérivées d'Intel, mais pas les classes définies par DMTF car cela risquerait d'endommager des applications existantes.

L'utilisation typique de cette mise en oeuvre CDM est basée sur l'espace de noms CIMV2. Les fichiers MOF pour un espace de noms IntelNCS sont les suivants :

Nom du fichier	Type de langue	Description
Cdla.mof	Principal	Définitions de classes pour la mise en oeuvre Intel CDM
CdlaLn.mof	Indépendant de la langue	Définitions de classes pour la mise en oeuvre Intel CDM
CdlaEnu.mfl	Dépendant de l'anglais	Extensions de langue pour la mise en oeuvre Intel CDM
CdCm.mof	Principal	Définitions de classes CDM du surensemble principal
CdCmLn.mof	Indépendant de la langue	Définitions de classes CDM du surensemble principal
CdCmEnu.mfl	Dépendant de l'anglais	Extensions de langue pour les définitions de classes CDM du surensemble principal
DNcsCdmN.mof	S/O	Supprime les classes Intel CDM

Les fichiers MOF pour un espace de noms CIMV2 sont les suivants :

Nom du fichier	Type de langue	Description
C2lcd.mof	Principal	Définitions de classes pour la mise en oeuvre Intel CDM
C2lcdLn.mof	Indépendant de la langue	Définitions de classes pour la mise en oeuvre Intel CDM
C2lcdEnu.mfl	Dépendant de l'anglais	Extensions de langue pour la mise en oeuvre Intel CDM
C2Cd.mof	Principal	Définitions de classes CDM du surensemble principal

C2CdLn.mof	Indépendant de la langue	Définitions de classes CDM du surensemble principal
C2CdEnu.mfl	Dépendant de l'anglais	Extensions de langue pour les définitions de classes CDM du surensemble principal
DNcsCdm2.mof	S/O	Supprime les classes Intel CDM

**Remarque :** la localisation requiert l'ajout du bon fichier **.mof** dépendant de la langue.

## Fichiers de ressources

Le fichier de ressources du fournisseur CDM est le suivant :

- **ENU\_Diag.dll** — Ressources WMI en anglais (États-Unis) pour le fournisseur de diagnostics.

---

Veuillez lire toutes les [restrictions et clauses exonératoires de responsabilité](#).

---

[Retour à la table des matières](#)   [Retour au début de la page](#)

[Retour à la table des matières](#)

## **Sécurité : Guide de l'utilisateur des fournisseurs WMI et CDM pour cartes réseau Intel® PRO**

---

Les fournisseurs WMI et CDM utilisent la représentation de client pour gérer la sécurité. Chaque appel au sein des fournisseurs sera fait dans le propre contexte de sécurité du client, qui sera transmis dans les couches inférieures. Une ou toutes les opérations peuvent échouer si vous ne disposez pas de droits d'administrateur sur l'ordinateur cible.

---

Veuillez lire toutes les [restrictions et clauses exonératoires de responsabilité](#).

---

[Retour à la table des matières](#)   [Retour au début de la page](#)

# Espace de noms et contexte : Guide de l'utilisateur des fournisseurs WMI et CDM pour cartes réseau Intel® PRO

Les classes CIM résident dans un espace de noms. L'espace de noms standard de Microsoft\* est appelé **root/cimv2** et est basé sur CIM v2.2 ou sur **root/défaut**. Les classes des fournisseurs WMI et CDM peuvent être ajoutées à cet espace de noms. Les fournisseurs sont basés sur CIM v2.5. En raison de cela et des différences utilisées dans les clés des objets, les classes des fournisseurs sont situées dans un espace de noms séparé, **root/IntelINCS**.

## Contexte WBEM

Les objets de contexte offrent des informations supplémentaires aux fournisseurs qui ne peuvent être transmises en tant que paramètre à une méthode API WMI. Pour enregistrer les qualificateurs de contexte, utilisez **IWbemContext**. L'indicateur d'interface pour l'objet de contexte est transmis en tant que dernier paramètre d'une méthode **IWbemServices**.

La table suivante contient les qualificateurs de contexte (valeurs nommées) utilisés par les fournisseurs. La plupart des qualificateurs, comme SessionHandle par exemple, ne sont utilisés qu'en conjonction avec des zones fonctionnelles spécifiques des fournisseurs, alors que LocaleID, MachineName et ApplicationName peuvent être définis pour tous les appels **IWbemServices**.

Si aucun contexte n'est passé aux fournisseurs, ils utilisent le LocaleID passé lors de l'appel **Initialize** aux fournisseurs. Toute lecture effectuée dans un contexte lit la configuration actuelle jusqu'à ce qu'une opération d'écriture soit réalisée. Les lectures suivantes affichent le système tel qu'il serait une fois l'opération d'écriture réussie. Un contexte NULL peut être utilisé pour les lectures.

Qualificateur de contexte	Type de variante	Description
SessionHandle	VT_BSTR	Identifie la copie des classes réseau IANet de l'application. L'application ne peut modifier les classes ou leurs attributs sans d'abord établir un descripteur de session. Reportez-vous à la section relative à la classe IANet_NetService pour savoir comment établir et utiliser un descripteur de session. Ce qualificateur n'est pas requis si l'application ne fait que lire les données des classes. Le descripteur de session permet au logiciel NCS de gérer plusieurs accès simultanés à la configuration, empêchant ainsi à la session de bloquer l'accès à tous les autres. Chaque session possède une mémoire cache séparée qui stocke toutes les modifications effectuées. Si plusieurs utilisateurs apportent des modifications simultanément, le premier utilisateur qui applique ses modifications prime. Les mémoires cache de tous les autres utilisateurs seront annulées.
LocaleID	VT_BSTR	ID de Microsoft pour des paramètres régionaux. Ce qualificateur est requis si l'application requiert des chaînes de texte localisées des fournisseurs. Tous les messages d'erreur et les avertissements seront en anglais, sauf si le LocaleID requis est utilisé.
ApplicationName	VT_BSTR	Le nom de l'application qui a lancé l'appel. Ce qualificateur est nécessaire pour la consignation.
MachineName	VT_BSTR	Le nom de l'ordinateur qui est connecté aux fournisseurs. Ce qualificateur est nécessaire pour la consignation.
PreCheck	VT_BOOL	Cette valeur booléenne est utilisée pour signaler aux fournisseurs que le client essaie de vérifier si oui ou non une opération est autorisée avant de l'exécuter. Par exemple, l'ajout d'une carte à une association.  Valeurs : <ul style="list-style-type: none"><li>• TRUE = Le fournisseur n'exécute pas l'opération mais renvoie un code d'erreur et un état détaillé si l'opération n'est pas autorisée.</li><li>• FALSE = Le fournisseur exécute l'opération.</li></ul>

		Si ce qualificateur est manquant, l'attribut est considéré comme étant FALSE.
WarningErrorCode	VT_I4	Certaines opérations requièrent l'envoi d'avertissements à l'utilisateur (p. ex., l'ajout d'une carte à une association peut nécessiter le rechargement de l'association dans certains cas). WMI ne fournit aucun mécanisme pour cela. Si ce qualificateur est présent et différent de zéro, le fournisseur renvoie E_FAIL si l'opération a réussi mais avec un avertissement. Le client doit consulter l'état détaillé pour obtenir le texte de l'avertissement.

---

Veillez lire toutes les [restrictions et clauses exonératoires de responsabilité](#).

---

[Retour à la table des matières](#)   [Retour au début de la page](#)

# Paramètres régionaux et localisation : Guide de l'utilisateur des fournisseurs WMI et CDM pour cartes réseau Intel® PRO

[Fichiers MOF localisés](#)

[Données d'attribut localisées](#)

---

Il y a deux aspects à la localisation des fournisseurs WMI et CDM : les fichiers MOF localisés et les données d'attribut localisées.

## Fichiers MOF localisés

Tous les fichiers MOF utilisés par les fournisseurs sont localisés conformément au modèle de localisation Microsoft Windows\* Management Instrumentation (WMI). Pour accomplir cela, chaque définition de classe est séparée comme suit :

- Une version indépendante de la langue ne contenant que la définition de base de la classe dans le fichier **.mof**.
- Une version dépendante de la langue contenant les informations de localisation, telles que les descriptions des propriétés spécifiques aux paramètres régionaux, dans le fichier **.mfl**.

## Langues prises en charge

Chinois (Taiwan)  
Chinois (RPC)  
Danois  
Néerlandais (Pays-Bas)  
Anglais (États-Unis)  
Finnois  
Français (France)  
Allemand (Allemagne)  
Italien (Italie)  
Japonais  
Norvégien (Bokmal)  
Portugais (Brésil)  
Espagnol (Espagne - moderne)  
Suédois

## Stockage des classes

Les définitions des classes spécifiques aux langues sont stockées dans un sous-espace de noms enfant sous l'espace de noms contenant les définitions de classe indépendantes aux langues. Par exemple, pour les fournisseurs WMI et CDM, un espace de noms enfant **ms\_409** se trouvera sous l'espace de noms **root/Intelncs** dans le cas des paramètres régionaux anglais. De même, il existe un sous-espace de noms enfant pour chaque langue prise en charge sous l'espace de noms **root/Intelncs**.

## Prise en charge des fichiers MOF localisés dans l'espace de noms CIMV2

Dans le cas des espaces de noms **root/cimv2**, les classes des fournisseurs (c.-à-d. les classes IANet\_) sont dérivées des classes de base ajoutées à ces espaces par WMI. Un sous-espace de noms avec les définitions des classes spécifiques à une langue pour les classes de base existe déjà sous l'espace de noms **root/cimv2**. Les définitions de classe IA\_Net spécifiques à une langue seront ajoutées à cet espace de noms enfant. En raison de cette dépendance des classes de base, la localisation des fichiers MOF est réalisée sous les paramètres régionaux par défaut du système.

## Exécution

Pour récupérer les données localisées, une application WMI peut spécifier les paramètres régionaux en utilisant le paramètre `strLocale` au cours des appels **SWbemLocator::ConnectServer** ou **IWbemLocator::ConnectServer**. Si les paramètres régionaux ne sont pas spécifiés, les paramètres par défaut du système sont utilisés (p. ex., MS\_409 pour l'anglais [États-Unis]). Ces paramètres régionaux sont utilisés pour sélectionner l'espace de noms approprié lors d'un ajout aux chaînes en anglais.

En outre, **IWbemServices::GetObject**, **SwbemServices.GetObject**, **IWbemServices::ExecQuery** et **SWbemServices.ExecQuery** doivent spécifier l'indicateur `WBEM_FLAG_USE_AMENDED_QUALIFIERS` pour demander des données localisées avec la définition de base. Ceci est requis pour toutes les fonctions qui génèrent des valeurs affichables en utilisant le mappage de valeurs, et affichent des descriptions ou d'autres qualificatifs modifiés des fichiers MOF.

[Retour au début de la page](#)

---

## Données d'attribut localisées

Pour obtenir les données d'attribut localisées (des messages d'erreur par exemple), les fournisseurs doivent connaître les paramètres régionaux de l'appelant pour chaque appel. Pour que cela fonctionne correctement, le client doit ajouter les paramètres régionaux dans l'objet de contexte passé pour chaque appel (voir la section Contexte WBEM sous [Espace de noms et contexte](#)). Si les fournisseurs doivent renvoyer une chaîne localisable, ils essaient de charger la DLL de ressources appropriée pour les paramètres régionaux du client. Si aucune DLL de ressources appropriée n'est trouvée, les fournisseurs renvoient les chaînes en anglais (États-Unis).

Veillez lire toutes les [restrictions et clauses exonératoires de responsabilité](#).

---

[Retour à la table des matières](#)   [Retour au début de la page](#)

# Rapports d'erreurs : Guide de l'utilisateur des fournisseurs WMI et CDM pour cartes réseau Intel® PRO

[Présentation](#)

[Codes d'erreur](#)

---

## Présentation

La section suivante, relative à `IANet_ExtendedStatus`, explique comment traiter les erreurs générées par les fournisseurs WMI et CDM. Le moment et la manière dont une erreur est renvoyée dépend de l'état synchrone, semi-synchrone ou asynchrone d'un appel. Dans la plupart des cas, `HRESULT` est défini sur `WBEM_E_FAILED` lorsqu'une erreur se produit. À ce stade cependant, il est impossible de savoir si l'erreur a été générée par WMI ou par les fournisseurs.

Pour obtenir l'objet de l'erreur dans le cas d'appels synchrones, utilisez `GetErrorInfo()` pour obtenir l'objet `IErrorInfo`. Utilisez `QueryInterface()` pour obtenir `IWbemClassObject` contenant les informations sur l'erreur.

Pour obtenir l'objet de l'erreur dans le cas d'appels asynchrones, `IWbemClassObject` est renvoyé en dernier lors du dernier appel `SetStatus()`. Après avoir obtenu l'instance de l'objet de l'erreur, vous pouvez vérifier la propriété `__Class` pour déterminer l'origine de l'erreur. WMI crée une instance de `__ExtendedStatus`, et les fournisseurs créent une instance de `IANet_ExtendedStatus` pour les erreurs associées à `IANet_` classes. `IANet_ExtendedStatus` est dérivé de `__ExtendedStatus` et contient les qualificatifs d'objet d'erreur suivants :

- Description – Description de l'erreur en fonction des paramètres régionaux actuels.
- File – Fichier code dans lequel l'erreur a été générée.
- Line – Numéro de ligne du fichier code dans laquelle l'erreur s'est produite.
- ParameterInfo – Classe ou attribut utilisé lorsque l'erreur s'est produite.
- Operation – Opération tentée lorsque l'erreur s'est produite.
- ProviderName – Nom du fournisseur ayant causé l'erreur.
- StatusCode – Code renvoyé depuis l'appel interne qui a échoué.
- SessionHandle – Descripteur de session utilisé pour l'opération.
- RuleFailureReasons – Raison de l'échec de l'opération. Une opération peut échouer suite à l'échec d'une règle technique. (P. ex., vous devez disposer d'une carte de gestion dans certaines associations.)

[Retour au début de la page](#)

---

## Codes d'erreur

Pour tous les codes d'erreur, les fournisseurs donnent une description personnalisée sur base des paramètres régionaux. Les codes d'erreur sont sous la forme de `HRESULT` avec une sévérité 1 et une facilité ITF. Une application peut utiliser les codes suivants comme base pour une action de récupération :

- 0x80040901 – "WMI: Put property failed" (Échec de l'application de la propriété)
- 0x80040902 – "WMI: No class object" (Pas d'objet de classe)
- 0x80040903 – "WMI: Failed to create class" (Impossible de créer la classe)
- 0x80040904 – "WMI: Failed to spawn instance of class" (Impossible de générer une instance de classe)
- 0x80040905 – "WMI: Failed to create safe array" (Impossible de créer une zone de sécurité)
- 0x80040906 – "WMI: Failed to put safe array" (Impossible de placer une zone de sécurité)
- 0x80040907 – "WMI: Failed to return object to WMI" (Impossible de retourner l'objet à WMI)
- 0x80040908 – "WMI: Get property failed" (Échec de l'obtention de la propriété)
- 0x80040909 – "WMI: Unexpected type while getting property" (Type inattendu lors de l'obtention de la propriété)
- 0x8004090A – "WMI: Class not implemented by this provider" (Classe non mise en oeuvre par ce fournisseur)
- 0x8004090B – "WMI: Unable to parse WQL statement" (Impossible d'analyser le langage de la requête WQL)
- 0x8004090C – "WMI: Providers only support WQL" (Le fournisseur ne prend en charge que le langage WQL)



- 0x8004090D – "WMI: Parameter in context has the wrong type" (Le type du paramètre n'est pas valide dans ce contexte)
- 0x8004090E – "WMI: Error formatting debug log" (Erreur de formatage du journal de débogage)
- 0x8004090F – "WMI: bad object path" (Chemin d'accès à l'objet non valide)
- 0x80040910 – "WMI: Failed to update setting" (Échec lors de la mise à jour du paramètre)
- 0x80040911 – "WMI: Null parameter passed to method" (Paramètre NULL transmis à la méthode)
- 0x80040912 – "Setting value too small." (La valeur du paramètre est trop petite)
- 0x80040913 – "Setting value too big." (La valeur du paramètre est trop grande)
- 0x80040914 – "Setting not in step" (Le paramètre ne correspond pas à un échelon)
- 0x80040915 – "String setting is too long" (Le paramètre de chaîne est trop long)
- 0x80040916 – "Setting is not one of the allowed values" (Le paramètre ne correspond pas à une des valeurs autorisées)
- 0x80040917 – "WMI: Qualifier not found" (Qualificateur introuvable)
- 0x80040918 – "WMI: Qualifer set not found" (Jeu de qualificateurs introuvable)
- 0x80040919 – "WMI: Safe array access failed" (Échec de l'accès à la zone de sécurité)
- 0x8004091A – "WMI: Unhandled exception" (Exception non gérée)
- 0x8004091B – "WMI: Operation is not supported for this class" (Opération non prise en charge par cette classe)
- 0x8004091C – "WMI: Unexpected event class" (Classe d'événement inattendue)
- 0x8004091D – "WMI: Bad event data" (Données d'événement non valides)
- 0x8004091E – "WMI: Operation succeeded with warnings" (Opération réussie avec des avertissements)
- 0x8004081F – "WMI: The NCS Service has been stopped." (Le service de configuration du réseau a été interrompu)
  
- 0x80040801 – "EAL: Internal exception" (Exception interne)
- 0x80040802 – "EAL: General failure" (Défaillance générale)
- 0x80040803 – "EAL: Not initialized" (Non initialisé)
- 0x80040804 – "EAL: Failed to initialize." (Échec d'initialisation)
- 0x80040805 – "EAL: Session limits exceeded" (Limites de session dépassées)
- 0x80040806 – "EAL: Out of memory" (Mémoire insuffisante)
- 0x80040807 – "EAL: Rule syntax error" (Erreur de syntaxe dans une règle)
- 0x80040808 – "EAL: Unexpected end of list" (Fin de liste inattendue)
- 0x80040809 – "EAL: Rule link error" (Erreur de liaison de règle)
- 0x8004080A – "EAL: Device Creation Failed" (Échec de création du périphérique)
- 0x8004080B – "EAL: Media service not found" (Service de liaison introuvable)
- 0x8004080C – "EAL: Device service not found" (Service de périphérique introuvable)
- 0x8004080D – "EAL: PCI bus module not found" (Module du bus PCI introuvable)
- 0x8004080E – "EAL: Adapter is a member of a team" (La carte est membre d'une association)
- 0x8004080F – "EAL: Rule Access Point creation error" (Erreur de création du point d'accès de règle)
- 0x80040810 – "EAL: Registry key error" (Erreur de clé de registre)
- 0x80040811 – "EAL: Registry XML file path error" (Erreur du chemin d'accès du fichier XML du registre)
- 0x80040812 – "EAL: Unknown event class" (Classe d'événement inconnue)
- 0x80040813 – "EAL: Unknown module id" (ID de module inconnu)
- 0x80040814 – "EAL: Rule service not found" (Service de règle introuvable)
- 0x80040815 – "EAL: NULL input pointer" (Indicateur d'entrée NULL)
- 0x80040816 – "EAL: Rule grammar error" (Erreur de grammaire de règle)
- 0x80040817 – "EAL: Rule failed" (Échec de règle)
- 0x80040818 – "EAL: Setting is already grouped" (Le paramètre est déjà groupé)
  
- 0x80040220 – "Sync Layer: Team removal failed." (Couche Sync : échec de la suppression de l'association)
- 0x80040221 – "Sync Layer: Vlan creation failed" (Couche Sync : échec de la création du réseau local virtuel)
- 0x80040222 – "Sync Layer: Vlan removal failed" (Couche Sync : échec de la suppression du réseau local virtuel)
- 0x80040223 – "Sync Layer: Adapter removal failed" (Couche Sync : échec de la suppression de la carte)
- 0x80040224 – "Sync Layer: Setting Change/Creation/Removal failed." (Couche Sync : échec de modification/création/suppression du réglage)
- 0x80040225 – "Sync Layer: Parameter Change/Removal failed." (Couche Sync : échec de modification/création/suppression du paramètre)
- 0x80040226 – "Sync Layer: NetConfig subsystem locked" (Couche Sync : sous-système NetConfig verrouillé) "
- 0x80040227 – "Sync Layer: System Update In Progress. Please try again later." (Couche Sync : mise à jour du système en cours. Veuillez essayer ultérieurement.)
- 0x80040228 – "Sync Layer: Adapter is Locked" (Couche Sync : la carte est verrouillée)
- 0x80040229 – "Sync Layer: Flash read failed." (Couche Sync : échec de lecture de la mémoire flash)
- 0x8004022A – "Sync Layer:"

- 0x80040210 – "Sync Layer: Invalid event offset" (Couche Sync : décalage d'événement non valide)
- 0x80040211 – "Sync Layer: Invalid input" (Couche Sync : entrée non valide)
- 0x80040212 – "Sync Layer: Invalid key" (Couche Sync : clé non valide)
- 0x80040213 – "Sync Layer: Adapter not team member" (Couche Sync : la carte n'est pas un membre de l'association)
- 0x80040214 – "Sync Layer: Driver not loaded" (Couche Sync : le pilote n'est pas chargé)
- 0x80040215 – "Sync Layer: Client impersonation failed" (Couche Sync : échec de représentation du client)
- 0x80040216 – "Sync Layer: Caught exception" (Couche Sync : exception interceptée)
- 0x80040217 – "Sync Layer: Session not locked" (Couche Sync : la session n'est pas verrouillée)
- 0x80040218 – "Sync Layer: Hardware access layer is not available" (Couche Sync : la couche d'accès au matériel n'est pas disponible)
- 0x80040219 – "Sync Layer: Flash not available" (Couche Sync : mémoire flash non disponible)
- 0x8004021A – "Sync Layer: Diagnostics not supported" (Couche Sync : diagnostics non pris en charge)
- 0x8004021B – "Sync Layer: Diagnostic test not running" (Le test de diagnostic n'est pas en cours d'exécution)
- 0x8004021C – "Sync Layer: Boot Agent update not available" (Couche Sync : mise à jour de l'agent d'amorçage Boot Agent non disponible)
- 0x8004021D – "Sync Layer: Boot Agent corrupted." (Couche Sync : agent d'amorçage Boot Agent endommagé)
- 0x8004021E – "Sync Layer: Flash write failed." (Couche Sync : échec d'écriture dans la mémoire flash)
- 0x8004021F – "Sync Layer: Team creation failed." (Couche Sync : échec de la création de l'association)
- 0x80040201 – "Sync Layer: Initialization failed" (Couche Sync : échec de l'initialisation)
- 0x80040202 – "Sync Layer: Invalid initialization handle" (Couche Sync : descripteur d'initialisation non valide)
- 0x80040203 – "Sync Layer: Session handle already exists" (Couche Sync : le descripteur de session existe déjà)
- 0x80040204 – "Sync Layer: Invalid session handle" (Couche Sync : descripteur de session non valide)
- 0x80040205 – "Sync Layer: The maximum number of sessions has been reached." (Couche Sync : le nombre maximal de sessions a été atteint)
- 0x80040206 – "Sync Layer: The session lock handle already exists" (Couche Sync : le descripteur de verrouillage de session existe déjà)
- 0x80040207 – "Sync Layer: Invalid session lock handle" (Couche Sync : descripteur de verrouillage de session non valide)
- 0x80040208 – "Sync Layer: Session already locked" (Couche Sync : la session est déjà verrouillée)
- 0x80040209 – "Sync Layer: Invalid media service module Id" (Couche Sync : ID de module de service de liaison non valide)
- 0x8004020A – "Sync Layer: Invalid Advanced Service Module Id" (Couche Sync : ID de module de service avancé non valide)
- 0x8004020B – "Sync Layer: Invalid device service module Id" (Couche Sync : ID de module de service de périphérique non valide)
- 0x8004020C – "Sync Layer: Invalid component type Id" (Couche Sync : ID de type de composant non valide)
- 0x8004020D – "Sync Layer: Invalid bus interface module Id" (Couche Sync : ID de module d'interface de bus non valide)
- 0x8004020E – "Sync Layer: Invalid sink window handle" (Couche Sync : descripteur de fenêtre de réception non valide)
- 0x8004020F – "Sync Layer: Invalid event Id" (Couche Sync : ID d'événement non valide)
- 0x80040401 – "HAM PCI: Invalid memory map address" (Adresse de mappage de mémoire non valide)
- 0x80040402 – "HAM PCI: Configuration driver failed to load" (Impossible de charger le pilote de configuration)
- 0x80040403 – "HAM PCI: Configuration driver version mismatch" (Discordance de version du pilote de configuration)
- 0x80040404 – "HAM PCI: Device slot not found" (Connecteur de périphérique introuvable)
- 0x80040405 – "HAM PCI: Diagnostic driver failed to load" (Impossible de charger le pilote de diagnostic)
- 0x80040406 – "HAM PCI: Diagnostic driver version mismatch" (Discordance de version du pilote de diagnostic)
- 0x80040407 – "HAM PCI: Diagnostic driver initialization failed" (Impossible d'initialiser le pilote de diagnostic)
- 0x80040408 – "HAM PCI: Diagnostics not initialized" (Diagnostics non initialisés)
- 0x80040409 – "HAM PCI: Diagnostics already initialized" (Diagnostics déjà initialisés)
- 0x8004040A – "HAM PCI: Diagnostic test already running" (Le test de diagnostic est déjà en cours d'exécution)
- 0x8004040B – "HAM PCI: Diagnostic test not running" (Le test de diagnostic n'est pas en cours d'exécution)
- 0x8004040C – "HAM PCI: Diagnostic test terminated" (Test de diagnostic terminé)
- 0x8004040D – "HAM PCI: Diagnostic Invalid test number" (Numéro de test de diagnostic non valide)
- 0x8004040E – "HAM PCI: Diagnostic hardware missing" (Matériel de diagnostic introuvable)
- 0x8004040F – "HAM PCI: Diagnostic send receive initialization failed" (Impossible d'initialiser l'envoi/la réception du diagnostic)
- 0x80040511 – "Media Service: NDIS IO call failed" (Service de liaison : échec de l'appel E/S NDIS)
- 0x80040512 – "Media Service: Miniport not loaded" (Service de liaison : miniport non chargé)
- 0x8004051B – "Media Service: Invalid device handle" (Service de liaison : descripteur de périphérique non valide)
- 0x8004051C – "Media Service: Invalid adapter handle" (Service de liaison : descripteur de carte non valide)
- 0x8004051D – "Media Service: Invalid team handle" (Service de liaison : descripteur d'association non valide)

- 0x8004051E – "Media Service: Invalid VLAN handle" (Service de liaison : descripteur de réseau local virtuel non valide)
  - 0x8004051F – "Media Service: Device missing" (Service de liaison : périphérique introuvable)
  - 0x80040520 – "Media Service: Invalid setting type" (Service de liaison : type de paramètre non valide)
  - 0x80040521 – "Media Service: Unknown invalid object" (Service de liaison : objet non valide inconnu)
  - 0x80040522 – "Media Service: Invalid Setting Handle" (Service de liaison : descripteur de paramètre non valide)
  - 0x80040523 – "Media Service: Invalid Team Mode" (Service de liaison : mode d'association non valide)
  - 0x80040525 – "Media Service: Setting Already Exists" (Service de liaison : le paramètre existe déjà)
  
  - 0x80042001 – "RAP: Already initialized" (Déjà initialisé)
  - 0x80042002 – "RAP: Invalid XML file" (Fichier XML non valide)
  - 0x80042003 – "RAP: XML load error" (Erreur de chargement du fichier XML)
  - 0x80042004 – "RAP: Not initialized" (Non initialisé)
  - 0x80042005 – "RAP: Rule not extracted before" (Règle non extraite précédemment)
  - 0x80042006 – "RAP: Conditions count mismatch" (Nombre de conditions incorrect)
  - 0x80042007 – "RAP: Results apply error" (Erreur d'application des résultats)
  - 0x80042008 – "RAP: Invalid rule" (Règle non valide)
  - 0x80042009 – "RAP: Node not found" (Noeud introuvable)
  - 0x8004200A – "RAP: Error no single node" (Erreur, pas de noeud unique)
  - 0x8004200B – "RAP: No action rule" (Pas de règle d'action)
  - 0x8004200C – "RAP: Zero condition" (Aucune condition)
  - 0x8004200D – "RAP: Zero action" (Aucune action)
  - 0x8004200E – "RAP: XML Decode error" (Erreur de décodage XML)
- 

Veillez lire toutes les [restrictions et clauses exonératoires de responsabilité](#).

---

[Retour à la table des matières](#)   [Retour au début de la page](#)

# Le schéma principal : Guide de l'utilisateur des fournisseurs WMI et CDM pour cartes réseau Intel® PRO

[Présentation](#)

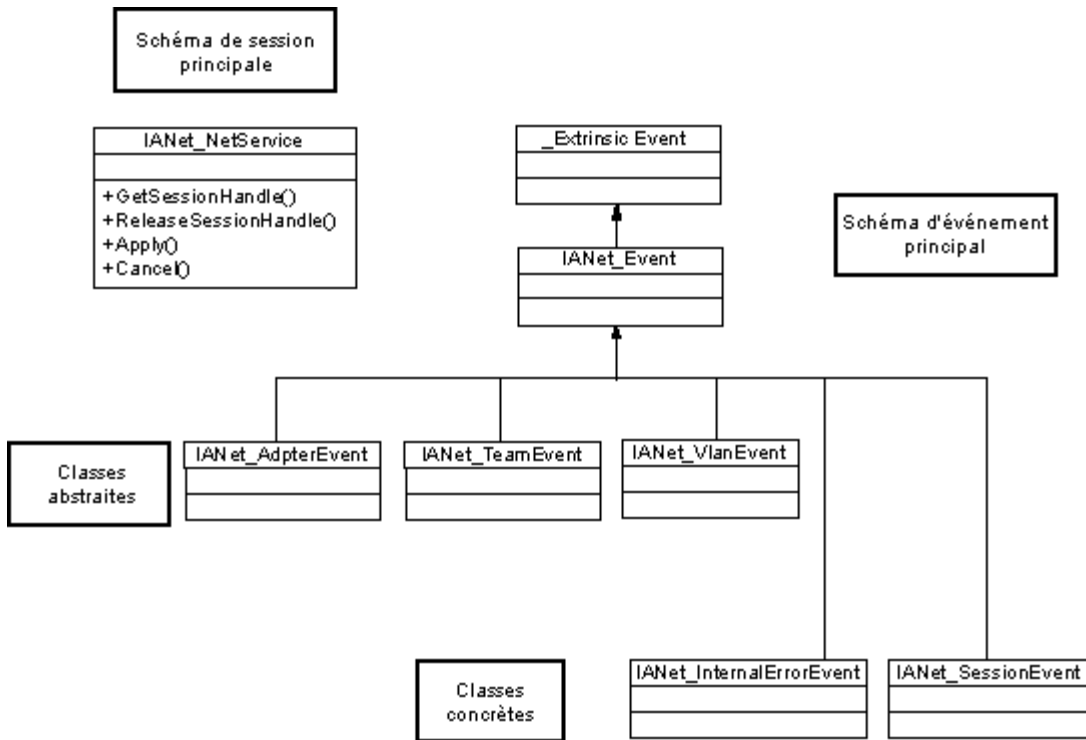
[IANet\\_NetService](#)

[Événements principaux](#)

[Cas d'emploi](#)

## Présentation

Le schéma principal consiste en la classe IANet\_NetService et les classes des événements principaux.



## IANet\_NetService

### Objectif

La classe IANet\_NetService est l'objet racine du schéma IANet\_ schema. Cette classe permet au client d'accéder aux sessions requises pour effectuer des opérations.

### Instances

Il y a une instance de cet objet. Le client ne doit pas se fier à la clé utilisée pour cette classe. Il doit plutôt obtenir l'instance de la classe en énumérant toutes les instances de IANet\_NetService.

### Création d'instances

Vous ne pouvez pas créer des instances de IANet\_NetService.

### Suppression d'instances

Vous ne pouvez pas supprimer des instances de IANet\_NetService.

### Modification des propriétés

Aucune propriété de cette classe ne peut être modifiée par l'utilisateur.

### Attributs pris en charge

Cette classe met en oeuvre deux attributs :

- Version — Contient la version actuelle du fournisseur principal.
- InstallDate — Contient la date d'installation des fournisseurs.

### Méthodes

Les méthodes suivantes peuvent être utilisées pour gérer des sessions :

- **void GetSessionHandle([([OUT] string SessionHandle, [out] uint32 ActiveSessions)** — Utilisé pour définir la chaîne du descripteur de session qui doit être placée dans l'objet de contexte du qualificateur SessionHandle. ActiveSessions renvoie le nombre de sessions actives pour ce système. Cela permet aux clients de signaler que d'autres peuvent être en train de modifier les paramètres réseau.
- **void Apply([IN] string sSessionHandle, [OUT] uint32 FollowupAction);** — Applique les modifications effectuées par un autre descripteur de session particulier. L'argument uint32 renvoyé est utilisé par les fournisseurs WMI et CDM pour indiquer à l'application que le serveur doit être redémarré avant que les modifications ne soient prises en compte. Ceci peut être réalisé en appelant la méthode **Reboot** sur la classe Win32\_OperatingSystem.

Valeurs :

1 = Redémarrage système requis  
0 = Redémarrage non requis

- **void ReleaseSessionHandle ([IN] string SessionHandle)** — Émet un descripteur de session après son utilisation. Toute modification de cette session est alors perdue. Le descripteur de session n'est plus valide après cet appel et il ne peut plus être utilisé.
- **void Cancel([IN] string SessionHandle);** — Annule une session. La mémoire cache interne est effacée et toutes les données lues après cet appel affichent la configuration actuelle.

[Retour au début de la page](#)

---

## Événements principaux

### IANet\_SessionEvent

#### Objectif

Cet événement est utilisé pour avertir le client de l'utilisation de l'API de la session NCS. Les clients peuvent utiliser cet événement pour recevoir une notification si d'autres clients créent ou utilisent des sessions.

#### Déclenchement

Cet événement est déclenché lorsqu'un client crée ou supprime une session, ou appelle la méthode **Apply** pour une session.

#### Données d'événement

**EventType** peut avoir l'une des valeurs suivantes :

- "New session" (Nouvelle session) indique qu'une nouvelle session a été créée par le client ou un autre client.
- "End session" (Session terminée) indique qu'un client a terminé une session. La session a été terminée par le client ou un autre client.
- "Cache invalidated" (Mémoire cache annulée) indique qu'un autre client a appelé la méthode **Apply** pour une session. Toutes les autres sessions sont annulées et les mémoires cache associées sont effacées.
- "Configuration changed" (Configuration modifiée) indique que la configuration de la session a changé.

**SessionHandle** contient le descripteur de session qui a déclenché l'événement.

**OpenSessions** contient le nombre de sessions ouvertes. Cet élément de données est NULL pour les événements "Cache invalidated" et "Configuration changed".

## IANet\_InternalErrorEvent

### Objectif

Cet événement est utilisé pour avertir le client qu'une erreur interne est survenue dans les fournisseurs d'événements. Dans certains cas, cela signifie que le fournisseur d'événements ne peut plus signaler les événements.

### Déclenchement

Cet événement se produit :

- lorsque le fournisseur d'événements reçoit un événement inconnu d'une source d'événements ;
- lorsque le logiciel qui fournit les événements est arrêté ;
- lorsque le fournisseur d'événements reçoit un événement, mais que la source d'événements ne peut obtenir des données supplémentaires sur celui-ci.

### Données d'événement

**EventType** peut avoir l'une des valeurs suivantes :

- "Could not get event data" (Impossible d'obtenir les données liées à l'événement). Un événement se produit, mais la source d'événements ne peut obtenir des données supplémentaires sur celui-ci.
- "Event source has shut down" (La source d'événements a été fermée). La source de données de l'événement a été fermée. Dans ce cas, le fournisseur d'événements est également fermé et aucun autre événement n'est généré tant que la source n'est pas redémarrée et que des nouvelles requêtes de notification sont faites.
- "Unexpected message" (Message inattendu). Le fournisseur d'événements a reçu un type d'événement inattendu.

[Retour au début de la page](#)

---

## Cas d'emploi

Un descripteur de session est requis pour modifier la configuration. Le descripteur de session permet au logiciel NCS de gérer plusieurs accès simultanés à la configuration, empêchant ainsi à la session de bloquer l'accès à tous les autres. Chaque session possède une mémoire cache séparée qui stocke toutes les modifications effectuées. Si plusieurs sessions effectuent des modifications simultanément, la première session qui applique ses modifications prime. Les mémoires cache de toutes les autres sessions sont annulées.

### Obtention d'un descripteur de session

Le client doit obtenir le chemin d'accès à l'objet d'une instance unique de `IANet_NetService` avant d'accéder au descripteur de session. Appelez `IWbemServices::CreateInstanceEnum` et passez le nom de la classe : `IANet_NetService`. Ceci revient à appeler `IWbemServices::ExecQuery` avec la requête `SELECT * FROM IANet_NetService`. Avant de modifier la configuration, le client doit obtenir un descripteur de session. Utilisez la méthode `GetSessionHandle` pour démarrer une nouvelle session.

Le client peut utiliser `IWbemServices::ExecMethod` pour exécuter une méthode sur un objet CIM, et doit obtenir le chemin de l'objet à partir de l'attribut `__PATH` de l'instance de `IANet_NetService`. Cette méthode renvoie également le nombre de sessions actives en cours. Le client peut émettre un avertissement s'il n'a pas un accès exclusif au service de configuration réseau (NCS).

### Utilisation d'un descripteur de session dans l'objet `IWbemContext`

Lorsque le client a obtenu un descripteur de session, il doit créer un objet `IWbemContext`. Stockez le descripteur de session dans le qualificateur `SessionHandle` de cet objet. Un indicateur de cet objet COM doit être passé à chaque appel dans `IWbemServices`. Le descripteur de session n'est pas requis pour les appels d'accès à l'objet `IANet_NetService` car il traite le descripteur comme un argument explicite.

### Lecture des modifications en attente en utilisant un descripteur de session

Si vous passez un descripteur de session dans le contexte lors de la lecture de la configuration, les fournisseurs renvoient la configuration telle qu'elle serait si les mises à jour avaient été appliquées (p. ex. les cartes désinstallées sont manquantes et les

paramètres modifiés renvoient leurs nouvelles valeurs). Cependant, certains objets n'apparaîtront pas tant que la méthode **Apply** n'aura pas été appelée (p. ex., `IANet_IPProtocolEndpoints` ne sera pas créé tant que le protocole n'aura pas été associé au miniport approprié).

## Terminer avec un descripteur de session

Après avoir modifié la configuration, appelez la méthode **Apply** pour valider les modifications. Cette action peut renvoyer un code d'action de suivi (p. ex., redémarrer le système pour que les modifications soient prises en compte).

Appelez toujours **ReleaseSessionHandle** une fois la session terminée, sinon les modifications seront perdues. L'appel de la méthode **Cancel** annule également les modifications apportées, mais le client peut continuer à utiliser le descripteur de session comme s'il venait d'être créé.

## Enregistrement des événements principaux

Les applications doivent utiliser **IWbemServices::ExecNotificationQuery** ou **IWbemServices::ExecNotificationQueryAsync** pour demander des notifications d'événements. Les requêtes suivantes sont des exemples de requêtes de notification (liste non exhaustive car de nombreuses requêtes sont possibles) :

- **SELECT \* FROM IANet\_Event** — Requête pour tous les événements.
- **SELECT \* FROM IANet\_SessionEvent** — Requête pour tous les événements de session.
- **SELECT \* FROM IANet\_InternalErrorEvent** — Requête pour tous les événements internes.

---

Veuillez lire toutes les [restrictions et clauses exonératoires de responsabilité](#).

---

[Retour à la table des matières](#)   [Retour au début de la page](#)

# Le schéma de la carte Ethernet : Guide de l'utilisateur des fournisseurs WMI et CDM pour cartes réseau Intel® PRO

[Présentation](#)

[IANet\\_EthernetAdapter](#)

[IANet\\_IPProtocolEndpoint](#)

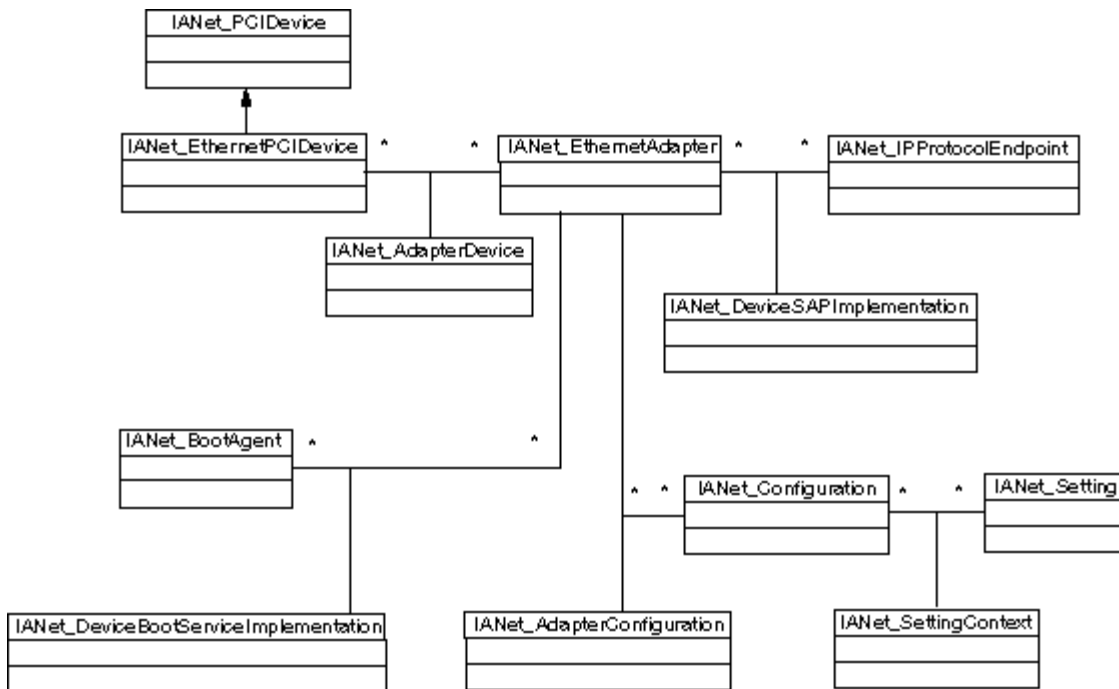
[IANet\\_BootAgent](#)

[IANet\\_PCIDevice](#)

[IANet\\_EthernetPCIDevice](#)

## Présentation

Le schéma de la carte est utilisé comme modèle pour les différentes cartes Ethernet Intel® PROSet configurables. Ce schéma est basé sur le schéma CIM v2.5.



[Retour au début de la page](#)

## IANet\_EthernetAdapter

### Objectif

IANet\_EthernetAdapter définit les capacités et l'état de toutes les cartes réseau Intel PRO installées, ainsi que de toute autre carte pouvant être ajoutée à une association en utilisant les pilotes intermédiaires d'Intel. La classe est dérivée de la superclasse CIM\_EthernetAdapter définie dans CIMv2.5. CIM\_EthernetAdapter est dérivé de CIM\_NetworkAdapter, une classe abstraite définissant les concepts matériels généraux de réseau comme l'adresse permanente, l'adresse actuelle, la vitesse d'opération, etc.

### Instances

Les instances de cette classe existent pour :

- les cartes d'interface réseau prises en charge et installées ;



- les cartes d'interface réseau non Intel pouvant faire partie d'une association multiconstructeur Intel ;
- les associations de cartes Intel créées.

### Création d'instances

Vous ne pouvez pas créer des instances de IANet\_EthernetAdapter.

### Suppression d'instances

La suppression d'une instance de IANet\_EthernetAdapter désinstalle les cartes physiques. Seules les cartes Intel non virtuelles peuvent être désinstallées de cette manière. Un descripteur de session est requis pour cette opération.

### Modification des propriétés

Aucune propriété de cette classe ne peut être modifiée par l'utilisateur.

### Attributs non pris en charge

Les attributs suivants ne sont pas requis pour Intel PROSet et ne sont par conséquent pas pris en charge :

- AutoSense (présenté comme un paramètre)
- ErrorCleared
- OtherIdentifyingInfo
- IdentifyingDescriptions
- InstallDate
- LastErrorCode
- MaxDataSize
- MaxQuiesceTime
- PowerManagementCapabilities (présenté comme une méthode)
- PowerManagementSupported (présenté comme une méthode)
- PowerOnHours
- ShortFramesReceived
- SymbolErrors
- TotalPowerOnHours

### Méthodes

Cette instance de classe prend en charge les méthodes suivantes :

- IdentifyAdapter** — Identifie la carte en faisant clignoter le témoin sur la carte pendant quelques secondes. Cette méthode ne fonctionne que pour les cartes physiques.
- HasVLANs** — Spécifie le nombre de réseaux locaux virtuels de cette carte.
- IsPowerMgmtSupported** — Indique si la gestion de la consommation électrique est prise en charge par la carte.
- GetPowerUsage** — Détection de la consommation électrique globale de la carte.  
0 = Consommation normale  
1 = Faible consommation
- SetPowerUsage** — Réduit la consommation électrique globale de la carte. La consommation électrique n'est pas conservée entre les redémarrages du système ou les rechargements du pilote. Lorsque le système redémarre ou que le pilote est rechargé, la consommation électrique normale de la carte est automatiquement restaurée.
- GetPowerUsageOptions** — Détection de toute option de consommation électrique de la carte (p. ex., consommation en mode de veille, en fonctionnement sur batterie, etc.).
- SetPowerUsageOptions** — Change les options de consommation électrique (p. ex., la méthode peut être utilisée pour réduire la consommation en mode de veille, en fonctionnement sur batterie, etc.).  
**Remarque** : les paramètres de consommation électrique sont enregistrés et utilisés lors de chaque démarrage.
- TestCable** — Exécute un test de diagnostic sur une carte particulière. En cas de panne, cette méthode renvoie les problèmes, les causes et les solutions possibles.
- AdvancedTestCable** — Exécute des tests de diagnostic avancés sur une carte particulière. Cette suite de tests est disponible pour les cartes 1 000 Mbit/s. La méthode renvoie les noms des tests et leurs résultats respectifs.  
**Remarque** : un échec de la liaison peut se produire si **SpeedDuplex** (Vitesse/Duplex) n'est pas défini sur **Auto Negotiate** (Négociation automatique). Dans cette instance, le paramètre de sortie **SpeedAndDuplexNotAutomatic** est TRUE.
- TestLinkSpeed** — Détermine si la carte opère à vitesse maximale. Si la carte affiche une vitesse inférieure à 1 gigabit, cette méthode énonce les raisons possibles. Par exemple, "Le partenaire de liaison ne peut opérer à une vitesse de 1 000 Mbit/s".

[Retour au début de la page](#)

# IANet\_IPProtocolEndpoint

## Objectif

Cette classe est utilisée pour décrire les paramètres IP pour un point de terminaison de protocole dans le système. WMI Provider ne fournit aucune information pour les autres types de protocole réseau. La classe est dérivée de la classe abstraite CIM\_IPProtocolEndpoint. WMI Provider fournit des informations sur le protocole lorsque celui-ci concerne une entité gérée par Intel PROSet uniquement.

## Instances

Une instance IANet\_IPProtocolEndpoint existera pour chaque liaison de la pile de protocole IP avec un point de terminaison pris en charge par Intel (p. ex., des cartes Intel, des cartes ou des réseaux locaux virtuels compatibles avec les associations Intel). Certaines cartes associées ne possèdent pas une adresse IP qui leur est propre, et n'ont donc aucun IANet\_IPProtocolEndpoint directement associé à leur instance de carte. IANet\_IPProtocolEndpoint n'existe que lorsque le système d'exploitation a lié le protocole à la carte ou au réseau local virtuel. Bien que certaines cartes puissent avoir plusieurs adresses IP, elles seront associées avec une seule instance de point de terminaison de protocole IP. Le fournisseur ne prend pas en charge cette utilisation avancée car elle n'est pas requise ni utilisée par Intel PROSet.

## Création d'instances

Vous ne pouvez pas créer des instances de IANet\_IPProtocolEndpoint. L'instance n'existe que lorsque le système d'exploitation a lié le protocole au point de terminaison.

## Suppression d'instances

Vous ne pouvez pas supprimer des instances de IANet\_IPProtocolEndpoint.

## Modification des propriétés

Aucune propriété de cette classe ne peut être modifiée par l'utilisateur.

## Associations

Une instance IANet\_AdapterProtocolImplementation est utilisée pour associer une carte IANet\_EthernetAdapter avec un IANet\_IPProtocolEndpoint. Une instance IANet\_VLANProtocolDependency est utilisée pour associer un réseau local virtuel IANet\_IPProtocolEndpoint.

**Remarque :** une association est associée au point de terminaison via la carte représentant la carte virtuelle de l'association.

## Attributs pris en charge

Les attributs en lecture seule suivants sont requis par Intel PROSet :

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

## Attributs non pris en charge

Les attributs suivants ne sont pas requis pour Intel PROSet et ne sont par conséquent pas pris en charge :

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

## Méthodes

Aucune.

[Retour au début de la page](#)

# IANet\_BootAgent

## Objectif

Cette classe est utilisée pour capturer des informations relatives aux capacités de démarrage réseau d'une carte (p. ex., les paramètres pour l'agent d'amorçage PXE pris en charge par certaines cartes Intel). Elle est dérivée de CIM\_BootService.

## Instances

Une instance IANet\_BootAgent existe pour chaque carte prenant en charge l'agent d'amorçage, même si celui-ci n'est pas installé.

## Création d'instances

Vous ne pouvez pas créer des instances de IANet\_BootAgent. Une instance n'existe que si la carte prend en charge la fonctionnalité de l'agent d'amorçage.

## Suppression d'instances

Vous ne pouvez pas supprimer des instances de IANet\_BootAgent.

## Modification des propriétés

Aucune propriété de cette classe ne peut être modifiée par l'utilisateur.

## Associations

Une instance IANet\_DeviceBootServiceImplementation est utilisée pour associer une carte IANet\_EthernetAdapter avec un IANet\_BootAgent si la carte le prend en charge.

## Attributs pris en charge

Les attributs en lecture seule suivants sont requis par Intel PROSet :

- InvalidImageSignature
- Version
- UpdateAvailable
- FlashImageType

## Attributs non pris en charge

Les attributs suivants ne sont pas requis pour Intel PROSet et ne sont par conséquent pas pris en charge :

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

## Méthodes

Les méthodes suivantes de cette classe peuvent être utilisées pour mettre à jour la ROM flash de la carte d'interface réseau :

<pre>uint32 ProgramFlash(     [IN,     ValueMap {"0","1"} ,      Values {"Check Version", "Write Flash"}: Amended     ]     uint32 Action,     [IN]     uint8 NewFlashData[],     [OUT]     string strErrorMessage</pre>	<p>Cette méthode est utilisée pour mettre à jour la ROM flash de la carte d'interface réseau. La liaison entre la carte d'interface réseau et le réseau est interrompue pendant la mise à jour de la mémoire flash.</p>
--	---

uint32 ReadFlash( [OUT] uint8 FlashData[] );	Cette méthode lit la ROM flash de la carte d'interface réseau.

[Retour au début de la page](#)

---

## IANet\_PCIDevice

### Objectif

Cette classe est utilisée pour décrire les propriétés d'un périphérique PCI pour un périphérique réseau du système. Elle est dérivée de CIM\_PCIDevice.

### Instances

Une instance de cette classe existe pour chaque carte PCI qui est un périphérique réseau du système. Pour IA64, seuls les périphériques PCI qui sont des cartes prises en charge par Intel PROSet auront des instances.

### Création d'instances

Vous ne pouvez pas créer des instances de IANet\_PCIDevice.

### Suppression d'instances

Vous ne pouvez pas supprimer des instances de IANet\_PCIDevice.

### Modification des propriétés

Aucune propriété de cette classe ne peut être modifiée par l'utilisateur.

### Associations

Voir IANet\_EthernetPCIDevice pour les associations de classes.

### Méthodes

Il n'y a aucune méthode prise en charge pour cette classe.

### Attributs non pris en charge

Les attributs suivants ne sont pas pris en charge par WMI Provider :

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

[Retour au début de la page](#)

---

# IANet\_EthernetPCIDevice

## Objectif

Cette classe est utilisée pour décrire les propriétés d'un périphérique PCI pour une carte Ethernet prise en charge par Intel PROSet. Il s'agit d'une sous-classe de IANet\_PCIDevice. La classe contient quelques attributs supplémentaires connus pour les périphériques PCI pris en charge par Intel PROSet uniquement.

## Instances

Une instance de cette classe existe pour chaque carte PCI qui est une carte Ethernet prise en charge par Intel PROSet.

## Création d'instances

Vous ne pouvez pas créer des instances de IANet\_EthernetPCIDevice.

## Suppression d'instances

Vous ne pouvez pas supprimer des instances de IANet\_EthernetPCIDevice.

## Modification des propriétés

Aucune propriété de cette classe ne peut être modifiée par l'utilisateur.

## Associations

Une instance IANet\_AdapterDevice est utilisée pour associer le périphérique IANet\_PCIDevice avec la carte IANet\_EthernetAdapter. Les cartes virtuelles (c.-à-d. les cartes créées pour représenter des associations) n'ont pas de IANet\_PCIDevice associé.

## Attributs non pris en charge

Les attributs suivants ne sont pas pris en charge par WMI Provider :

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

## Méthodes

Il n'y a aucune méthode prise en charge pour cette classe.

---

Veillez lire toutes les [restrictions et clauses exonératoires de responsabilité](#).

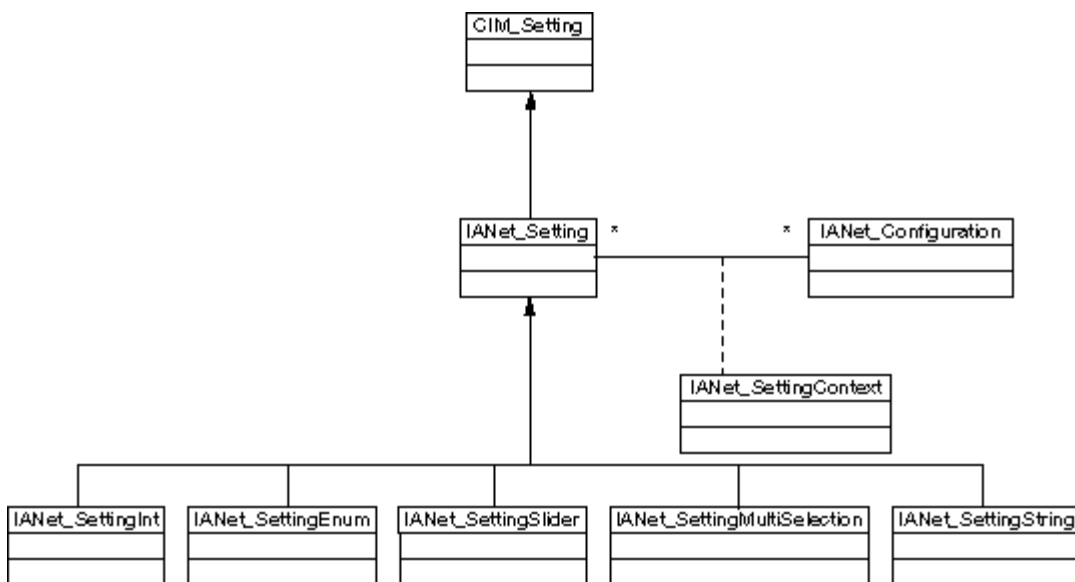
---

[Retour à la table des matières](#)   [Retour au début de la page](#)

# Le schéma des paramètres : Guide de l'utilisateur des fournisseurs WMI et CDM pour cartes réseau Intel® PRO

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

## Présentation



## IANet\_Configuration

**Objectif**  
Cette classe est utilisée pour grouper une collection d'instances `IANet_Setting`. Elle est dérivée de `CIM_Configuration`.

**Instances**  
Chaque carte, réseau local virtuel ou association peut avoir plusieurs instances `IANet_Configuration` qui lui sont associées (chaque configuration correspond à un scénario d'utilisation différent pour la carte).

Pour cette version des fournisseurs WMI et CDM, il n'y aura qu'une seule instance de `IANet_Configuration` par carte, réseau local virtuel ou association.

**Création d'instances**  
Vous ne pouvez pas créer des instances de `IANet_Configuration`.

**Suppression d'instances**

Vous ne pouvez pas supprimer des instances de IANet\_Configuration.

### **Modification des propriétés**

Aucune propriété de cette classe ne peut être modifiée par l'utilisateur.

### **Associations**

Une instance IANet\_AdapterConfiguration existera pour associer chaque carte (IANet\_EthernetAdapter) à cette configuration. Une instance IANet\_VLANConfiguration existera pour associer chaque réseau local virtuel (IANet\_VLAN) à cette configuration. Une instance IANet\_BootAgentConfiguration existera pour associer chaque agent d'amorçage (IANet\_BootAgent) à cette configuration.

### **Méthodes**

Il n'y a aucune méthode prise en charge pour cette classe.

### **Attributs non pris en charge**

Aucun.

[Retour au début de la page](#)

---

## **IANet\_Setting**

### **Objectif**

Cette classe abstraite est utilisée pour décrire une propriété paramétrable dans une configuration. Elle est dérivée de CIM\_Setting.

### **Instances**

Une instance séparée de cette classe existe pour chaque paramètre de chaque carte, réseau local virtuel ou association. Les paramètres ne sont pas partagés entre les configurations.

Il existe plusieurs sous-classes pour IANet\_Setting. Les sous-classes correspondent aux différents types et plages de valeurs que les paramètres peuvent accepter. Chaque sous-classe correspond à un style de GUI différent pouvant être utilisé pour afficher ou modifier les paramètres.

### **Création d'instances**

Vous ne pouvez pas créer des instances de IANet\_Setting.

### **Suppression d'instances**

Vous ne pouvez pas supprimer des instances de IANet\_Setting.

### **Modification des propriétés**

Cette classe abstraite ne possède aucune propriété modifiable. Les classes enfant, cependant, en possèdent (voir ci-dessous).

### **Associations**

Chaque instance IANet\_Setting est associée à une instance IANet\_Configuration en utilisant une instance de IANet\_SettingContext.

### **Méthodes**

Il n'y a aucune méthode prise en charge pour cette classe. Pour modifier un paramètre, modifiez la propriété requise et appelez **PutInstance**.

### **Attributs non pris en charge**

SettingID n'est pas utilisé.

[Retour au début de la page](#)

---

## **IANet\_SettingInt**

### **Objectif**

Cette classe modélise un paramètre qui accepte une valeur entière. Plusieurs classes de paramètre IANet sont utilisées pour modéliser des nombres entiers. La différence entre ces classes est liée à la façon dont le GUI affiche et modifie le nombre entier, ainsi qu'à la méthode de validation par le fournisseur. Pour IANet\_SettingInt, le GUI affichera une zone d'édition dotée d'un bouton toupie.

## Instances

Une instance de cette classe existe pour chaque paramètre qui doit être affiché dans une zone d'édition de nombre entier.

## Création d'instances

Vous ne pouvez pas créer des instances de cette classe.

## Suppression d'instances

Vous ne pouvez pas supprimer des instances de cette classe.

## Modification des propriétés

L'attribut "CurrentValue" est la seule propriété modifiable de cette classe. Vous pouvez modifier cette propriété en utilisant **IWbemClassObject::Put()** pour changer la valeur, et en appelant ensuite **IWbemServices::PutInstance()** pour mettre le paramètre à jour. Les fournisseurs vérifieront que :

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

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

(**CurrentValue** - **min**) est un multiple de **Step**

où **max**, **min**, **CurrentValue** et **Step** sont tous des attributs de **IANet\_SettingInt**.

## Associations

Chaque instance **IANet\_SettingInt** est associée à une instance **IANet\_Configuration** en utilisant une instance de **IANet\_SettingContext**.

## Attributs non pris en charge

SettingID n'est pas utilisé.

## Méthodes

Il n'y a aucune méthode prise en charge pour cette classe. Pour modifier un paramètre, modifiez la propriété requise et appelez **PutInstance**.

[Retour au début de la page](#)

---

# IANet\_SettingEnum

## Objectif

Cette classe modélise un paramètre qui accepte une valeur entière. Plusieurs classes de paramètre **IANet** sont utilisées pour modéliser des nombres entiers. La différence entre ces classes est liée à la façon dont le GUI affiche et modifie le nombre entier, ainsi qu'à la méthode de validation par le fournisseur. Pour **IANet\_SettingEnum**, le GUI affichera une liste de chaînes mappées sur quelques valeurs énumérées (p. ex., une zone de liste déroulante).

## Instances

Une instance de cette classe existe pour chaque paramètre qui sera affiché sous la forme d'une énumération.

## Création d'instances

Vous ne pouvez pas créer des instances de cette classe.

## Suppression d'instances

Vous ne pouvez pas supprimer des instances de cette classe.

## Modification des propriétés

L'attribut **CurrentValue** est la seule propriété modifiable de cette classe. Vous pouvez modifier cette propriété en utilisant **Put()** pour changer la valeur, et en appelant ensuite **PutInstance()** pour mettre le paramètre à jour. Les fournisseurs vérifieront que **CurrentValue** ? **PossibleValues[]**

## Associations

Chaque instance **IANet\_SettingEnum** est associée à une instance **IANet\_Configuration** en utilisant une instance de **IANet\_SettingContext**.

## Attributs non pris en charge

SettingID n'est pas utilisé.



## Méthodes

Il n'y a aucune méthode prise en charge pour cette classe. Pour modifier un paramètre, modifiez la propriété requise et appelez **PutInstance**.

[Retour au début de la page](#)

---

## IANet\_SettingSlider

### Objectif

Cette classe modélise un paramètre qui accepte une valeur entière. Plusieurs classes de paramètre IANet sont utilisées pour modéliser des nombres entiers. La différence entre ces classes est liée à la façon dont le GUI affiche et modifie le nombre entier, ainsi qu'à la méthode de validation par le fournisseur. Pour IANet\_SettingSlider, le GUI affichera un curseur qui permettra de choisir la valeur d'une manière graphique - la valeur réelle ne doit pas être affichée.

### Instances

Une instance de cette classe existe pour chaque paramètre qui sera affiché sous la forme d'un curseur.

### Création d'instances

Vous ne pouvez pas créer des instances de cette classe.

### Suppression d'instances

Vous ne pouvez pas supprimer des instances de cette classe.

### Modification des propriétés

L'attribut **CurrentValue** est la seule propriété modifiable de cette classe. Vous pouvez modifier cette propriété en utilisant **Put()** pour changer la valeur, et en appelant ensuite **PutInstance()** pour mettre le paramètre à jour. Les fournisseurs vérifieront que **CurrentValue ? PossibleValues[]**

### Associations

Chaque instance IANet\_SettingSlider est associée à une instance IANet\_Configuration en utilisant une instance de IANet\_SettingContext.

### Attributs non pris en charge

SettingID n'est pas utilisé.

## Méthodes

Il n'y a aucune méthode prise en charge pour cette classe. Pour modifier un paramètre, modifiez la propriété requise et appelez **PutInstance**.

[Retour au début de la page](#)

---

## IANet\_SettingMultiSelection

### Objectif

Cette classe modélise un paramètre permettant de sélectionner différentes options sur une liste. Pour IANet\_SettingMultiSelection, le GUI affichera une zone de liste multi-sélection qui permettra de choisir des options.

### Instances

Une instance de cette classe existe pour chaque paramètre qui sera affiché sous la forme d'une multi-sélection.

### Création d'instances

Vous ne pouvez pas créer des instances de cette classe.

### Suppression d'instances

Vous ne pouvez pas supprimer des instances de cette classe.

### Modification des propriétés

L'attribut **CurrentValue** est la seule propriété modifiable de cette classe. Vous pouvez modifier cette propriété en utilisant **Put()** pour changer la valeur, et en appelant ensuite **PutInstance()** pour mettre le paramètre à jour. Les fournisseurs vérifieront que

## CurrentValue ? PossibleValues[]

### Associations

Chaque instance IANet\_SettingMultiSelection est associée à une instance IANet\_Configuration en utilisant une instance de IANet\_SettingContext.

### Attributs non pris en charge

SettingID n'est pas utilisé.

### Méthodes

Il n'y a aucune méthode prise en charge pour cette classe. Pour modifier un paramètre, modifiez la propriété requise et appelez **PutInstance**.

[Retour au début de la page](#)

---

## IANet\_SettingString

### Objectif

Cette classe modélise un paramètre permettant d'entrer une valeur de chaîne ouverte. Pour IANet\_SettingMultiSelection, le GUI affichera une zone d'édition.

### Instances

Une instance de cette classe existe pour chaque paramètre qui sera affiché sous la forme d'une zone d'édition.

### Création d'instances

Vous ne pouvez pas créer des instances de cette classe.

### Suppression d'instances

Vous ne pouvez pas supprimer des instances de cette classe.

### Modification des propriétés

L'attribut **CurrentValue** est la seule propriété modifiable de cette classe. Vous pouvez modifier cette propriété en utilisant **Put()** pour changer la valeur, et en appelant ensuite **PutInstance()** pour mettre le paramètre à jour.

### Associations

Chaque instance IANet\_SettingMultiSelection est associée à une instance IANet\_ElementConfiguration en utilisant une instance de IANet\_SettingString.

### Méthodes

Il n'y a aucune méthode prise en charge pour cette classe.

### Attributs non pris en charge

SettingID n'est pas utilisé.

### Méthodes

Il n'y a aucune méthode prise en charge pour cette classe. Pour modifier un paramètre, modifiez la propriété requise et appelez **PutInstance**.

---

Veillez lire toutes les [restrictions et clauses exonératoires de responsabilité](#).

---

[Retour à la table des matières](#)   [Retour au début de la page](#)

# Le schéma de l'association : Guide de l'utilisateur des fournisseurs WMI et CDM pour cartes réseau Intel® PRO

## Présentation

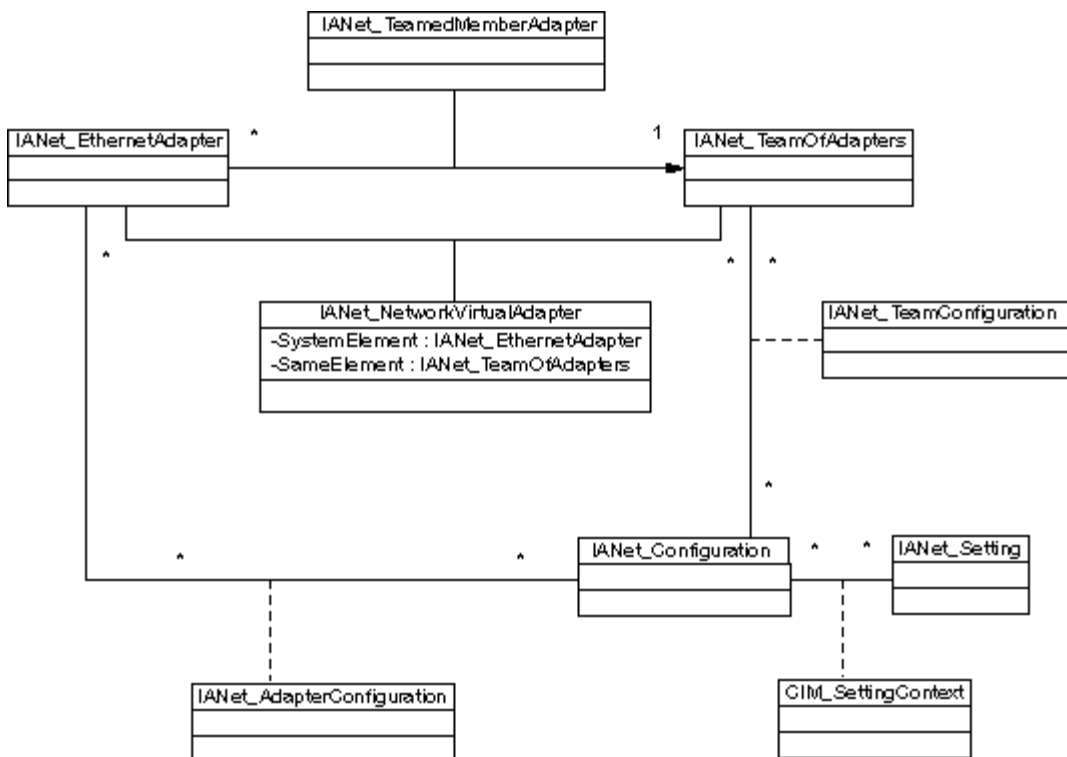
[IANet\\_TeamOfAdapters](#)

[IANet\\_TeamedMemberAdapter](#)

[IANet\\_NetworkVirtualAdapter](#)

## Présentation

Le schéma de l'association décrit comment les cartes Ethernet sont groupées en associations.



## IANet\_TeamOfAdapters

### Objectif

Cette classe met en oeuvre la classe CIM\_RedundancyGroup. Elle possède des membres décrivant le type et le nombre de cartes de l'association, et le nombre maximal de cartes pouvant faire partie de l'association.

### Instances

Il y a une instance de cette classe pour chaque association Intel.

### Création d'instances

Pour créer une association vide, créez une instance de IANet\_TeamOfAdapters. Vous devez définir un **TeamingMode** approprié avant d'appeler **IWbemServices::PutInstance()** pour créer l'objet dans les fournisseurs. Les fournisseurs renverront une chaîne contenant le chemin d'accès du nouvel objet.

### Suppression d'instances

Similairement, supprimez l'instance `IANet_TeamOfAdapters` pour supprimer une association. Les fournisseurs supprimeront l'association, les membres, la carte virtuelle et les paramètres de l'association.

### Modification des propriétés

Vous pouvez modifier cette propriété en utilisant **Put()** pour changer la valeur de **TeamingMode**, et en appelant ensuite **PutInstance()** pour mettre l'association à jour.

### Associations

Chaque carte de l'association est associée avec l'instance de `IANet_TeamOfAdapters` en utilisant l'instance de `IANet_TeamMemberAdapter`. La carte virtuelle de l'association est associée avec cette classe en utilisant une instance de `IA_NetNetworkVirtualAdapter`.

### Méthodes

Cette instance de classe prend en charge la méthode suivante :

**TestSwitchConfiguration** — Teste la configuration du commutateur pour vérifier que l'association fonctionne correctement avec celui-ci. Ce test peut être utilisé pour s'assurer que les partenaires de liaison (c.-à-d. un périphérique auquel la carte est liée, tel qu'une autre carte, un concentrateur, un commutateur, etc.) prennent en charge le mode d'agrégation de cartes sélectionné. Par exemple, si une carte est membre d'une association de type Agrégation de liens, ce test peut vérifier que les partenaires de liaison connectés prennent en charge le mode Agrégation de liens.

[Retour au début de la page](#)

---

## IANet\_TeamedMemberAdapter

### Objectif

Cette classe est utilisée pour associer la carte à l'association, déterminer la fonction de la carte au sein de l'association, et pour déterminer si la carte est actuellement active au sein de l'association. Cette classe met en oeuvre la classe CIM `CIM_NetworkAdapterRedundancyComponent`.

### Instances

Une instance de cette classe existe pour chaque carte membre d'une association.

### Création d'instances

Pour ajouter une carte à une association, créez une instance de `IANet_TeamedMemberAdapter`.

### Suppression d'instances

Pour supprimer une carte d'une association, supprimez l'instance `IANet_TeamedMemberAdapter`. La carte ne fera plus partie de l'association mais peut être liée à un point de terminaison de protocole IP une fois la fonction **Apply()** appelée.

### Modification des propriétés

La propriété **AdapterFunction** de cette classe peut être modifiée pour décrire comment la carte est utilisée au sein de l'association.

### Associations

Il s'agit d'une classe d'association.

### Méthodes

Il n'y a aucune méthode prise en charge pour cette classe.

[Retour au début de la page](#)

---

## IANet\_NetworkVirtualAdapter

### Objectif

Cette classe est utilisée pour associer l'instance de `IANet_TeamOfAdapters` avec l'instance de `IANet_EthernetAdapter` qui représente la carte virtuelle de l'association. Cette classe met en oeuvre la classe CIM `CIM_NetworkVirtualAdapter`.

## Instances

Une instance de cette classe existe pour chaque association Intel qui a été liée à une carte virtuelle.

## Création d'instances

Vous ne pouvez pas créer des instances de cette classe. Pour créer une association, créez une instance de IANet\_TeamOfAdapters. Cette classe n'existera pas tant que **IANet\_NetService.Apply()** n'aura pas été appelé dans le contexte d'une session valide et que l'instance IANet\_EthernetAdapter n'aura pas été créée.

## Suppression d'instances

Vous ne pouvez pas supprimer des instances de cette classe.

## Associations

Il s'agit d'une classe d'association.

## Méthodes

Il n'y a aucune méthode prise en charge pour cette classe.

---

Veuillez lire toutes les [restrictions et clauses exonératoires de responsabilité](#).

---

[Retour à la table des matières](#)   [Retour au début de la page](#)

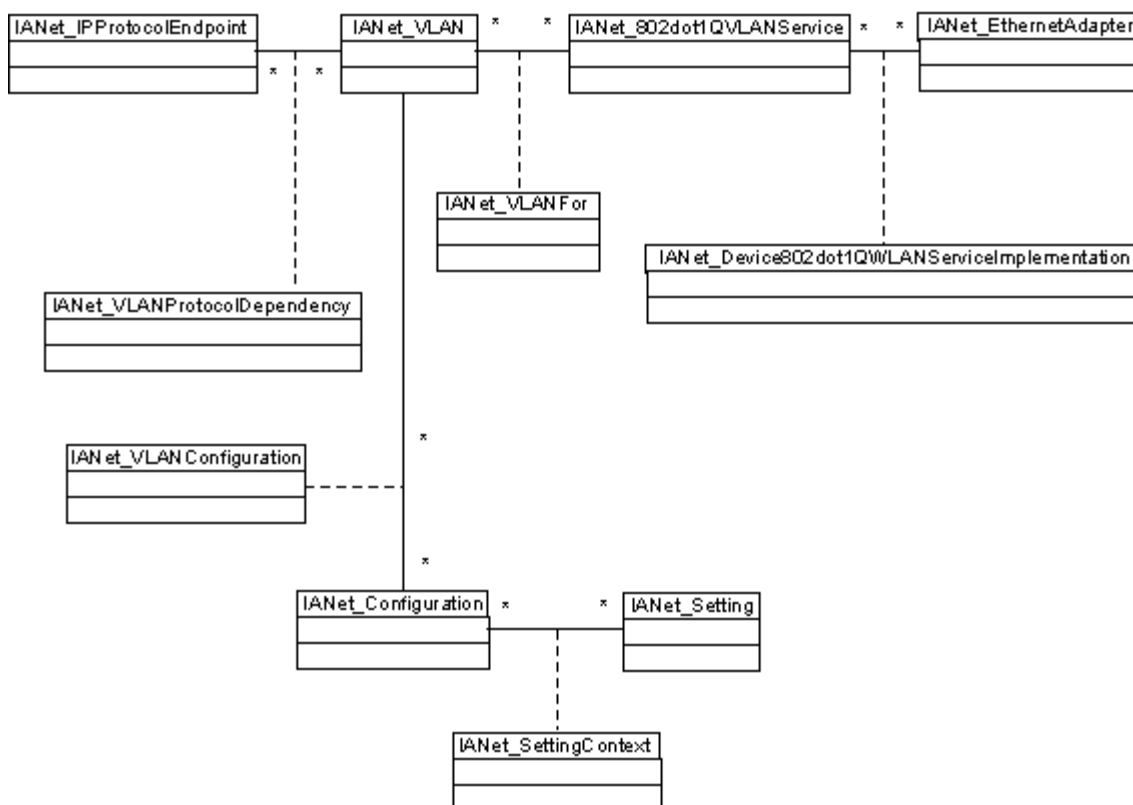
# Le schéma du réseau local virtuel : Guide de l'utilisateur des fournisseurs WMI et CDM pour cartes réseau Intel® PRO

[Présentation](#)

[IANet\\_802dot1QVLANService](#)

[IANet\\_VLAN](#)

## Présentation



## IANet\_802dot1QVLANService

### Objectif

Cette classe est utilisée pour conserver les propriétés IEEE 802.1Q d'une carte réseau. Elle met en oeuvre la classe CIM CIM\_802dot1QVLANService.

### Instances

Une instance de cette classe existe pour chaque carte ou association prenant en charge IEEE 802.1Q. Chaque carte peut avoir un seul IANet\_802dot1QVLANService uniquement. Certaines associations, telles que les associations multiconstructeur avec tolérance aux pannes, ne prennent pas ce service en charge.

### Anomalie

Une association sans réseau local virtuel ne possède pas de service VLAN, sauf si vous l'énumérez au sein du contexte d'une

session valide. Pour les associations, l'instance 802.3QvlanService n'apparaît que dans les situations suivantes :

- L'association a déjà des réseaux locaux virtuels.
- L'association n'a pas de réseau local virtuel et vous utilisez un descripteur de session dans le contexte en énumérant cette classe.

### Création d'instances

Vous ne pouvez pas créer des instances de cette classe. Si la classe n'a pas d'instance qui lui est associée, elle ne prend pas en charge ce service.

### Suppression d'instances

Vous ne pouvez pas supprimer des instances de cette classe.

### Modification des propriétés

Aucune propriété de cette classe ne peut être modifiée.

### Associations

Chaque instance de cette classe sera associée avec une instance IANet\_EthernetAdapter en utilisant une instance de IANet\_DeviceServiceImplementation.

Chaque instance de IANet\_802dot1QVLANService peut prendre en charge plusieurs réseaux locaux virtuels ; chaque réseau local virtuel sera associé avec l'instance en utilisant l'association IANet\_VLANFor.

### Méthodes

**uint16 CreateVLAN( [in] uint32 VLANNumber, [in] string Name, [out] IANet\_VLAN REF VLANpath );** — Utilisé pour créer un réseau local virtuel sur la carte ou sur l'association. Le client doit fournir le numéro et le nom du réseau local virtuel. Il recevra ensuite le chemin d'accès à l'objet du nouveau réseau local virtuel.

[Retour au début de la page](#)

---

## IANet\_VLAN

### Objectif

Cette classe contient les informations pour chaque réseau local virtuel Intel. Elle met en oeuvre CIM\_VLAN.

### Instances

Une instance de cette classe existera pour chaque réseau local virtuel Intel.

### Création d'instances

Pour créer un réseau local virtuel, appelez **CreateVLAN** sur l'instance appropriée de IANet\_802dot1QVLANService.

### Suppression d'instances

Vous pouvez supprimer une instance de cette classe pour supprimer le réseau local virtuel correspondant.

### Modification de propriétés

Vous pouvez modifier les attributs VLANNumber et Caption.

### Associations

Chaque instance est associée avec une instance de IANet\_802dot1QVLANService, et donc avec une instance de IANet\_EthernetAdapter, en utilisant la classe IANet\_VLANFor.

Chaque instance peut être associée avec plusieurs instances IANet\_Configuration pour grouper un ensemble de paramètres pour le réseau local virtuel. Pour cette version du fournisseur, il n'y a qu'un seul objet IANet\_Configuration par réseau local virtuel.

Chaque instance peut être associée avec un IANet\_IPProtocolEndpoint pour fournir les paramètres IP au réseau local virtuel en utilisant la classe IANet\_VLANProtocolDependency.

### Méthodes

Aucune

---

Veuillez lire toutes les [restrictions et clauses exonératoires de responsabilité](#).





# Obtention de la configuration actuelle : Guide de l'utilisateur des fournisseurs WMI et CDM pour cartes réseau Intel® PRO

- [Obtention des cartes physiques](#)
- [Obtention des périphériques PCI](#)
- [Obtention des paramètres de la carte](#)
- [Obtention de la configuration de l'association](#)
- [Obtention des paramètres de l'association](#)
- [Obtention de la configuration du réseau local virtuel](#)
- [Obtention des paramètres du réseau local virtuel](#)
- [Obtention des informations de protocole IP](#)
- [Obtention des informations de l'agent d'amorçage](#)
- [Obtention des paramètres de l'agent d'amorçage](#)

Le client n'a pas besoin d'un descripteur de session pour pouvoir lire la configuration actuelle. Les clients peuvent utiliser un contexte NULL. Cependant, tout message d'erreur sera renvoyé dans la langue par défaut de l'ordinateur géré. Les objets de la table suivante entourés de { }, sont des chemins d'accès aux objets. On suppose que ces chemins d'accès ont été obtenus au cours de précédentes requêtes WQL. Le client ne doit jamais créer le chemin d'accès d'un objet sans effectuer une requête. L'attribut **\_\_PATH** de chaque objet contient le chemin d'accès de cet objet.

Les méthodes **IWbemServices::ExecQuery** ou **IWbemServices::ExecQueryAsync** sont utilisées pour exécuter les requêtes WQL dans tous les cas d'emploi suivants.

## Obtention des cartes physiques

Tâche	Requête WQL	Classe de résultat	Commentaire
Lister toutes les cartes	SELECT * FROM IANet_EthernetAdapter	IANet_EthernetAdapter	Renvoie toutes les cartes IANet_EthernetAdapters. Équivalent à IWbemServices::CreateInstanceEnumAsync.
Déterminer si la carte est virtuelle	ASSOCIATORS OF {chemin de la carte} WHERE AssocClass = IANet_NetworkVirtualAdapter	IANet_TeamOfAdapters	Si la requête ne renvoie aucun résultat dans aucune classe, la carte est une carte réelle.
Déterminer si la carte est une carte fantôme	ASSOCIATORS OF {chemin de la carte} WHERE ResultClass = IANet_EthernetPCIDevice	IANet_EthernetPCIDevice	Si la carte n'est pas virtuelle et que cette requête ne renvoie aucun objet, la carte est une carte fantôme.

La classe principale des cartes est IANet\_EthernetAdapter. Cette classe est utilisée à la fois pour les cartes physiques et les cartes virtuelles, et le client doit pouvoir faire la distinction entre les deux.

## Obtention des périphériques PCI

Les classes principales sont IANet\_EthernetPCIDevice, IANet\_PCIDevice et IANet\_AdapterDevice (une classe d'association pour lier une carte à son périphérique).

Dans ce cas, la classe d'association ne contient aucune donnée, c'est-à-dire, elle n'a aucune valeur. IANet\_EthernetPCIDevice

hérite de IANet\_PCIDevice et contient des attributs supplémentaires spécifiques à un périphérique PCI (une carte Ethernet).

Tâche	Requête WQL	Classe de résultat	Commentaire
Énumérer les périphériques PCI du réseau	SELECT * FROM IANet_PCIDevice	IANet_PCIDevice	La requête peut renvoyer des modems et d'autres périphériques non configurés par Intel® PROSet.  Équivalent à IWbemServices::CreateInstanceEnumAsync.
Énumérer tous les périphériques PCI utilisés par les cartes configurées par Intel® PROSet	SELECT * FROM IANet_EthernetPCIDevice	IANet_EthernetPCIDevice	La requête renvoie juste les périphériques PCI gérés par Intel® PROSet.  Équivalent à IWbemServices::CreateInstanceEnumAsync.
Déterminer si une carte est installée.	S/O	IANet_EthernetPCIDevice	Vérifier l'attribut 'Availability' de IANet_EthernetPCIDevice. S'il est égal à 10 (Non installé), le périphérique n'est pas installé. Remarque : peu d'informations sont connues au sujet du périphérique à ce stade.
Obtenir le périphérique PCI associé à une carte	ASSOCIATORS OF {chemin de IANet_EthernetAdapter} WHERE ResultClass = IANet_EthernetPCIDevice	IANet_EthernetPCIDevice	Si cette requête ne renvoie aucun objet, la carte est soit une carte virtuelle, soit une carte fantôme.
Obtenir la carte associée à un périphérique PCI	ASSOCIATORS OF {chemin du périphérique PCI Ethernet} WHERE ResultClass = IANet_EthernetAdapter	IANet_EthernetAdapter	Cette requête n'est pas optimale pour le fournisseur. Il est préférable que le client commence par la carte.

[Retour au début de la page](#)

## Obtention des paramètres de la carte

Les objets Setting ne sont pas associés directement à la carte. Conformément à la norme CIM, ils sont associés à un objet Configuration qui lui-même est associé à la carte.

Les classes impliquées dans cette partie du schéma sont IANet\_EthernetAdapter, IANet\_Configuration, IANet\_SettingInt, IANet\_SettingString, IANet\_SettingEnum, IANet\_SettingMultiSelection et IANetSettingSlider.

Les classes d'association IANet\_AdapterConfiguration et IANet\_SettingContext ne contiennent aucune donnée réelle. Elles servent de contact entre les paramètres et leur objet parent.

Tâche	Requête WQL	Classe de résultat	Commentaire
Obtenir l'objet Configuration de la carte	ASSOCIATORS OF {chemin de IANet_EthernetAdapter} WHERE ResultClass = IANet_Configuration	IANet_Configuration	Renvoie exactement un objet ; même s'il n'y a pas de paramètre, il y aura toujours un objet Configuration. Le chemin de cet objet est utilisé pour la requête suivante.
Obtenir les paramètres associés à	ASSOCIATORS OF {chemin de IANet_Configuration} WHERE AssocClass =	Un mélange de IANet_SettingInt, IANet_SettingString,	Renvoie toutes les classes Setting associées à la carte. Le client doit utiliser l'attribut __CLASS pour déterminer le type de chaque paramètre.

une carte	IANet_SettingContext	IANet_SettingEnum, IANet_SettingMultiSelection, IANet_SettingSlider.
-----------	----------------------	--

[Retour au début de la page](#)

## Obtention de la configuration de l'association

Les classes principales du schéma d'association sont IANet\_EthernetAdapter, IANet\_TeamOfAdapters, IANet\_NetworkVirtualAdapter et IANet\_TeamedMemberAdapter.

Le défi dans ce schéma est qu'une instance de IANet\_EthernetAdapter existe pour chaque carte physique et chaque carte virtuelle. Le client doit pouvoir faire la distinction entre la carte virtuelle d'une association et les cartes qui sont des membres de l'association.

La classe d'association IANet\_NetworkVirtualAdapter ne contient aucune donnée utile - les clients ne sont intéressés que par les points de terminaison de cette association. IANet\_TeamedMemberAdapter ne contient aucune donnée utile quant à la méthode d'utilisation de la carte membre au sein de l'association.

Tâche	Requête WQL	Classe de résultat	Commentaire
Lister toutes les associations	SELECT * FROM IANet_TeamOfAdapters	IANet_TeamOfAdapters	Il y a une instance de IANet_TeamOfAdapters pour chaque association.  Équivalent à IWbemServices::CreateInstanceEnumAsync.
Obtenir la carte virtuelle d'une association	ASSOCIATORS OF {chemin de IANet_TeamOfAdapters} WHERE AssocClass = IANet_NetworkVirtualAdapter	IANet_EthernetAdapter	Ne renvoie que l'objet de la carte pour la carte virtuelle de l'association. Cette carte sera inexistante si l'association a été créée mais que l'option Apply n'a pas été appelée. (Voir ci-dessous pour la mise à jour de la configuration.)
Lister les cartes membres de l'association	ASSOCIATORS OF {chemin de IANet_TeamOfAdapters} WHERE AssocClass = IANet_TeamedMemberAdapter	IANet_EthernetAdapter	Renvoie les cartes qui font partie de l'association, mais ne décrit pas le rôle des cartes.
Déterminer le rôle d'une carte dans une association	REFERENCES OF {chemin de IANet_EthernetAdapter} WHERE ResultClass = IANet_TeamedMemberAdapter	IANet_TeamedMemberAdapter	La classe contient des informations décrivant comment la carte membre est associée à l'association et son état actuel au sein de celle-ci.

[Retour au début de la page](#)

## Obtention des paramètres de l'association

Les objets Setting ne sont pas associés directement à l'association. Conformément à la norme CIM, ils sont associés à un objet Configuration qui lui-même est associé à la carte virtuelle IANet\_EthernetAdapter de l'association. Le même objet Configuration est également associé à l'objet IANet\_TeamOfAdapters de l'association.

Les classes impliquées dans cette partie du schéma sont IANet\_EthernetAdapter, IANet\_TeamOfAdapters, IANet\_Configuration, IANet\_SettingInt, IANet\_SettingString, IANet\_SettingEnum, IANet\_SettingMultiSelection et IANetSettingSlider.

Les classes d'association IANet\_AdapterConfiguration et IANet\_SettingContext ne contiennent aucune donnée réelle. Elles servent de contact entre les paramètres et leur objet parent. Cela correspond exactement au cas des paramètres de la carte.

Tâche	Requête WQL	Classe de résultat	Commentaire
Obtenir l'objet Configuration de l'association, en commençant par la carte virtuelle	ASSOCIATORS OF {chemin de IANet_EthernetAdapter} WHERE ResultClass = IANet_Configuration	IANet_Configuration	Renvoie exactement un objet. Même s'il n'y a pas de paramètre, il y aura toujours un objet Configuration. Le chemin de cet objet est utilisé pour la requête suivante.
Obtenir l'objet Configuration de l'association, en commençant par l'association de cartes	ASSOCIATORS OF {chemin de IANet_TeamOfAdapters} WHERE ResultClass = IANet_Configuration		
Obtenir les paramètres associés à une carte	ASSOCIATORS OF {chemin de IANet_Configuration} WHERE AssocClass = IANet_SettingContext	Un mélange de IANet_SettingInt, IANet_SettingString, IANet_SettingEnum, IANet_SettingMultiSelection, IANet_SettingSlider.	Renvoie toutes les classes Setting associées à la carte. Le client doit utiliser l'attribut __CLASS pour déterminer le type de chaque paramètre.

[Retour au début de la page](#)

## Obtention de la configuration du réseau local virtuel

Chaque carte prenant en charge un réseau local virtuel a un IANet\_802dot1QVLANSservice qui lui est associé avec la classe d'association IANet\_Device802do1QVVLANSserviceImplementation. Si la carte n'a pas d'instance de cette classe qui lui est associée, elle ne prend pas en charge les réseaux locaux virtuels.

Chaque réseau local virtuel est représenté par une instance de IANet\_VLAN. Le réseau local virtuel n'est pas directement associé à la carte. Il est associé au service IANet\_802dot1QVLANSservice de la carte.

La classe d'association IANet\_VLANFor est utilisée pour associer chaque instance de réseau local virtuel avec le service IANet\_802dot1QVLANSservice approprié. Cette classe ne contient aucune donnée utile à l'utilisateur.

Tâche	Requête WQL	Classe de résultat	Commentaire
Obtenir l'objet du service 802.1q VLAN associé à une carte	ASSOCIATORS OF {chemin de IANet_EthernetAdapter} WHERE ResultClass = IANet_802dot1QVLANSservice	IANet_802dot1QVLANSservice	Renvoie zéro ou un objet.
Obtenir les réseaux locaux virtuels d'une carte	ASSOCIATORS OF {chemin de IANet_802dot1QVLANSservice} WHERE ResultClass = IANet_VLAN	IANet_VLAN	Peut ne renvoyer aucun objet si aucun réseau local virtuel n'est installé.

[Retour au début de la page](#)

## Obtention des paramètres du réseau local virtuel

Les objets Setting ne sont pas associés directement au réseau local virtuel. Conformément à la norme CIM, ils sont associés à un objet Configuration qui lui-même est associé à l'objet IANet\_VLAN du réseau local virtuel.

Les classes impliquées dans cette partie du schéma sont IANet\_VLAN, IANet\_Configuration, IANet\_SettingInt, IANet\_SettingString, IANet\_SettingEnum, IANet\_SettingMultiSelection et IANetSettingSlider.

Les classes d'association IANet\_VLANConfiguration et IANet\_SettingContext ne contiennent aucune donnée réelle. Elles servent de contact entre les paramètres et leur objet parent. Cela correspond exactement au cas des paramètres de la carte.

Tâche	Requête WQL	Classe de résultat	Commentaire
Obtenir l'objet Configuration du réseau local virtuel	ASSOCIATORS OF {chemin de IANet_VLAN} WHERE ResultClass = IANet_Configuration	IANet_Configuration	Renvoie exactement un objet ; même s'il n'y a pas de paramètre, il y aura toujours un objet Configuration. Le chemin de cet objet est utilisé pour la requête suivante.
Obtenir les paramètres associés à un réseau local virtuel	ASSOCIATORS OF {chemin de IANet_Configuration} WHERE AssocClass = IANet_SettingContext	Un mélange de IANet_SettingInt, IANet_SettingString, IANet_SettingEnum, IANet_SettingMultiSelection, IANet_SettingSlider.	Renvoie toutes les classes Setting associées au réseau local virtuel. Le client doit utiliser l'attribut __CLASS pour déterminer le type de chaque paramètre.

[Retour au début de la page](#)

## Obtention des informations de protocole IP

Le fournisseur donne une quantité limitée d'informations relatives aux points de terminaison de protocole IP, associés aux cartes, aux réseaux locaux virtuels et aux associations. Aucun autre protocole n'est pris en charge.

La classe principale contenant les informations de protocole est IANet\_IPProtocolEndpoint. Il existe deux classes d'association : IANet\_VLANProtocolDependency et IANet\_AdapterProtocolImplementation. Pour obtenir le point de terminaison IP d'une association, obtenir tout d'abord la carte virtuelle IANet\_EthernetAdapter de l'association (c.-à-d. le point de terminaison IP associé à cette instance).

Tâche	Requête WQL	Classe de résultat	Commentaire
Obtenir les points de terminaison de protocole IP associés à une carte	ASSOCIATORS OF {chemin de IANet_EthernetAdapter} WHERE ResultClass = IANet_IPProtocolEndpoint	IANet_IPProtocolEndpoint	Bien que certaines cartes puissent avoir plusieurs adresses IP, elles seront associées avec une seule instance de point de terminaison de protocole IP.
Obtenir les points de terminaison de protocole IP associés à un réseau local virtuel	ASSOCIATORS OF {chemin de IANet_VLAN} WHERE ResultClass = IANet_IPProtocolEndpoint	IANet_IPProtocolEndpoint	Obtenir les points de terminaison de protocole IP associés à un réseau local virtuel

[Retour au début de la page](#)

## Obtenir les informations de l'agent d'amorçage

Chaque carte pouvant prendre en charge un agent d'amorçage dans la mémoire ROM flash aura une instance IANet\_BootAgent qui lui sera associée en utilisant la classe d'association IANet\_DeviceBootServiceImplementation.

Tâche	Requête WQL	Classe de résultat	Commentaire
Obtenir l'agent d'amorçage associé à une carte	ASSOCIATORS OF {chemin de IANet_EthernetAdapter} WHERE ResultClass = IANet_BootAgent	IANet_BootAgent	Les attributs en lecture seule suivants fournissent des informations sur l'image de la ROM de démarrage pour cette carte : InvalidImageSignature, Version, UpdateAvailable, FlashImageType

[Retour au début de la page](#)

## Obtenir les paramètres de l'agent d'amorçage

Les objets Setting ne sont pas associés directement à l'agent d'amorçage. Conformément à la norme CIM, ils sont associés à un objet Configuration qui lui-même est associé à l'agent d'amorçage.

Les classes impliquées dans cette partie du schéma sont IANet\_BootAgent, IANet\_Configuration, IANet\_SettingInt, IANet\_SettingString, IANet\_SettingEnum, IANet\_SettingMultiSelection et IANetSettingSlider.

Les classes d'association IANet\_BootAgentConfiguration et IANet\_SettingContext ne contiennent aucune donnée réelle. Elles servent de contact entre les paramètres et leur objet parent.

Tâche	Requête WQL	Classe de résultat	Commentaire
Obtenir l'objet Configuration de l'agent d'amorçage	ASSOCIATORS OF {chemin de IANet_BootAgent} WHERE ResultClass = IANet_Configuration	IANet_Configuration	Renvoie exactement un objet. Même s'il n'y a pas de paramètre, il y aura toujours un objet Configuration. Le chemin de cet objet est utilisé pour la requête suivante.
Obtenir les paramètres associés à un agent d'amorçage	ASSOCIATORS OF {chemin de IANet_Configuration} WHERE AssocClass = IANet_SettingContext	Un mélange de IANet_SettingInt, IANet_SettingString, IANet_SettingEnum, IANet_SettingMultiSelection, IANet_SettingSlider.	Renvoie toutes les classes Setting associées à la carte. Le client doit utiliser l'attribut __CLASS pour déterminer le type de chaque paramètre.

Veillez lire toutes les [restrictions et clauses exonératoires de responsabilité](#).

[Retour à la table des matières](#)   [Retour au début de la page](#)

# Mise à jour de la configuration : Guide de l'utilisateur des fournisseurs WMI et CDM pour cartes réseau Intel® PRO

## [Présentation](#)

[Modification des paramètres d'une carte, d'une association ou d'un réseau local virtuel](#)

[Création d'une association \(vide\)](#)

[Ajout d'une carte à une association](#)

[Suppression d'une carte d'une association](#)

[Suppression d'une association](#)

[Modification du mode d'une association](#)

[Modification de la priorité d'une carte au sein d'une association](#)

[Désinstallation d'une carte](#)

[Création d'un réseau local virtuel \(VLAN\)](#)

[Modification des attributs d'un réseau local virtuel](#)

[Suppression d'un réseau local virtuel](#)

[Mise à jour de Boot Agent](#)

---

## Présentation

Dans la plupart des cas, l'application cliente doit obtenir un descripteur de session de la classe IANet\_NetService, et le stocker dans un objet de contexte IWbemContext pour mettre à jour la configuration. Les modifications de la configuration ne sont prises en compte que lorsque la méthode **Apply** est appelée sur IANet\_NetService. Cette condition comporte quelques exceptions :

- Les modifications de la classe de l'agent d'amorçage sont prises en compte immédiatement et ne nécessitent pas de descripteur de session.
- Certains appels de méthode (p. ex. identifier la carte) entraînent l'exécution d'une opération avant que la méthode **Apply** ne soit appelée.

Pour certaines opérations, vous pouvez utiliser le qualificateur PreCheck (Pré-contrôle) dans le contexte pour vérifier si une opération est autorisée. Cela permet à l'interface utilisateur de désactiver certaines commandes ou certains éléments de menu, le cas échéant.

[Retour au début de la page](#)

---

## Modification des paramètres d'une carte, d'une association ou d'un réseau local virtuel

La modification d'une carte, d'une association ou d'un réseau local virtuel :

- Nécessite un descripteur de session.
- Le PreCheck (Pré-contrôle) est disponible.
- Requiert l'appel de la méthode **Apply** avant l'exécution d'une opération.

Pour modifier un paramètre de carte, d'association ou de réseau local virtuel, le client doit tout d'abord obtenir le chemin d'accès à l'objet du paramètre à modifier. Pour ce faire, la meilleure solution est d'énumérer les paramètres pour l'objet et de stocker l'attribut **\_\_PATH** du paramètre (voir ci-dessus).

Le moyen le plus facile pour le client de mettre un paramètre à jour est :

1. d'obtenir une instance de l'objet Setting de WMI ;
2. de modifier l'attribut CurrentValue (en utilisant **IWbemClassObject::Put()**) ;
3. d'appeler **IWbemServices::PutInstance()** pour renvoyer l'instance modifiée au fournisseur WMI. **PutInstance** doit être appelé avec l'indicateur WBEM\_FLAG\_UPDATE\_ONLY.

Le fournisseur WMI va valider la valeur actuelle CurrentValue et renvoyer WBEM\_E\_FAIL si la validation échoue. La raison exacte de l'échec sera renvoyée dans l'attribut Description de l'objet IANet\_ExtendedStatus.

Les descriptions spécifiques aux paramètres incluent :

- La valeur du nombre entier est inférieure au minimum autorisé.
- La valeur du nombre entier est supérieure au maximum autorisé.
- La valeur du nombre entier ne correspond pas à un échelon autorisé.
- La longueur de la chaîne est supérieure à la longueur maximale autorisée.
- La valeur du paramètre n'est pas une des valeurs autorisées.

La dernière description est renvoyée lorsque la valeur actuelle pour IANet\_SettingEnum, IANet\_SettingSlider ou IANet\_SettingMultiSelection n'est pas une valeur autorisée.

Le seul attribut que le client puisse modifier pour un paramètre est CurrentValue. Le fournisseur WMI ignore toute modification apportée aux autres valeurs.

Il n'y a aucune méthode prise en charge pour la classe Setting. Pour modifier un paramètre, modifiez la propriété CurrentValue et appelez **PutInstance**.

[Retour au début de la page](#)

---

## Création d'une association (vide)

La création d'une association :

- Nécessite un descripteur de session.
- Le PreCheck (Pré-contrôle) est disponible.
- Requiert l'appel de la méthode **Apply** avant l'exécution d'une opération.

Pour créer une association, créez une instance de IANet\_TeamOfAdapters (c.-à-d., utilisez **IWbemServices::GetObject()** pour obtenir un objet de classe pour IANet\_TeamOfAdapters, et utilisez ensuite **IWbemServices::SpawnInstance()** pour créer une instance de cet objet).

Utilisez ensuite **IWbemClassObject::Put** pour définir l'attribut TeamMode dans l'instance et lui attribuer le type d'association désiré (p. ex. AFT). Enfin, appelez **IWbemServices::PutInstance()** pour créer l'association en passant l'indicateur WBEM\_FLAG\_CREATE\_ONLY.

Le chemin d'accès à l'objet pour la nouvelle association est stocké dans l'objet **IWbemCallResultObject** qui est renvoyé lorsque l'appel est terminé. La méthode **IWbemCallResult::GetResultString** obtiendra le nouveau chemin d'accès à l'objet.

Si cette action échoue, le client doit vérifier l'état IANet\_ExtendedStatus pour obtenir les raisons de l'échec.

Les classes virtuelles IANet\_EthernetAdapter et IANet\_IPProtocolEndpoint de l'association ne sont disponibles qu'après l'appel de la méthode **Apply**. Les paramètres de l'association sont accessibles en utilisant l'objet IANet\_Configuration associé avec le nouveau IANet\_TeamOfAdapters.

[Retour au début de la page](#)

---

## Ajout d'une carte à une association

L'ajout d'une carte à une association :

- Nécessite un descripteur de session.
- Le PreCheck (Pré-contrôle) est disponible.
- Requiert l'appel de la méthode **Apply** avant l'exécution d'une opération.

Pour ajouter une carte à une association, créez une instance de IANet\_TeamedMemberAdapter (c.-à-d., utilisez **IWbemServices::GetObject()** pour obtenir un objet de classe pour IANet\_TeamedMemberAdapter, et utilisez ensuite **IWbemServices::SpawnInstance()** pour créer une instance de cet objet).



Les attributs suivants de l'objet doivent être définis en utilisant `IWbemClassObject::Put()` :

- `GroupComponent` doit être défini comme étant le chemin d'accès complet à l'objet de l'association `IANet_TeamOfAdapters` à laquelle la carte doit être ajoutée.
- `PartComponent` doit être défini comme étant le chemin d'accès complet à l'objet de la carte `IANet_EthernetAdapter` qui doit être ajoutée à l'association.

Vous pouvez également définir la priorité de la carte au sein de l'association. Enfin, appelez **`IWbemServices::PutInstance()`** pour ajouter la carte à l'association en passant l'indicateur `WBEM_FLAG_CREATE_ONLY`. Si cette action échoue, vérifiez `IANet_ExtendedStatus` pour obtenir le code d'erreur.

[Retour au début de la page](#)

---

## Suppression d'une carte d'une association

La suppression d'une carte d'une association :

- Nécessite un descripteur de session.
- Le `PreCheck` (Pré-contrôle) est disponible.
- Requiert l'appel de la méthode **`Apply`** avant l'exécution d'une opération.

Pour supprimer une carte d'une association, supprimez l'instance `IANet_TeamedMemberAdapter` qui associe la carte à l'association en utilisant **`IWbemServices::DeleteInstance()`**. Si cette action échoue, vérifiez `IANet_ExtendedStatus` pour obtenir le code d'erreur.

[Retour au début de la page](#)

---

## Suppression d'une association

La suppression d'une association :

- Nécessite un descripteur de session.
- Le `PreCheck` (Pré-contrôle) est disponible.
- Requiert l'appel de la méthode **`Apply`** avant l'exécution d'une opération.

Pour supprimer une association, supprimez l'instance `IANet_TeamOfAdapters` en utilisant **`IWbemServices::DeleteInstance()`**. Si cette action échoue, vérifiez `IANet_ExtendedStatus` pour obtenir le code d'erreur.

[Retour au début de la page](#)

---

## Modification du mode d'une association

La modification du mode d'une association :

- Nécessite un descripteur de session.
- Le `PreCheck` (Pré-contrôle) est disponible.
- Requiert l'appel de la méthode **`Apply`** avant l'exécution d'une opération.

Pour modifier le mode d'une association, obtenez l'instance de `IANet_TeamOfAdapters` pour l'association (p. ex., utilisez **`IWbemServices::GetObject`** avec le chemin d'accès à l'objet de l'association). Utilisez ensuite **`IWbemClassObject::Put`** pour modifier l'attribut `TeamMode` pour l'association. Enfin, appelez **`IWbemClassObject::PutInstance`** pour demander au fournisseur WMI de mettre à jour le mode de l'association en passant l'indicateur `WBEM_FLAG_UPDATE_ONLY`. Si cette action échoue, vérifiez `IANet_ExtendedStatus` pour obtenir le code d'erreur.

[Retour au début de la page](#)

---

## Modification de la priorité d'une carte au sein d'une association

La modification de la priorité d'une carte au sein d'une association :

- Nécessite un descripteur de session.
- Le PreCheck (Pré-contrôle) est disponible.
- Requiert l'appel de la méthode **Apply** avant l'exécution d'une opération.

Pour modifier la priorité d'une carte au sein d'une association, le client doit d'abord obtenir l'instance de `IANet_TeamedMemberAdapter` pour la carte (p. ex., utilisez `IWbemServices::GetObject` avec le chemin d'accès à l'objet). Le client peut ensuite utiliser `IWbemClassObject::Put` pour modifier l'attribut `AdapterFunction` pour la carte. Enfin, le client doit appeler `IWbemClassObject::PutInstance` pour demander au fournisseur WMI de mettre à jour la priorité de la carte. Si cette action échoue, le client doit vérifier `IANet_ExtendedStatus` pour obtenir le code d'erreur.

[Retour au début de la page](#)

---

## Désinstallation d'une carte

La désinstallation d'une carte :

- Nécessite un descripteur de session.
- Le PreCheck (Pré-contrôle) est disponible.
- Requiert l'appel de la méthode **Apply** avant l'exécution d'une opération.

Pour désinstaller une carte, appelez `IWbemServices::DeleteInstance` en passant le chemin d'accès à l'objet de la carte à désinstaller.

[Retour au début de la page](#)

---

## Création d'un réseau local virtuel (VLAN)

La création d'un réseau local virtuel :

- Nécessite un descripteur de session.
- Le PreCheck (Pré-contrôle) est disponible.
- Requiert l'appel de la méthode **Apply** avant l'exécution d'une opération.

Pour créer un réseau local virtuel, appelez la méthode `CreateVLAN` sur le service `IANet_802dot1QVLANService` pour la carte sur laquelle le réseau local virtuel doit être ajouté. Les arguments suivants doivent être passés à la méthode :

- `VLANNumber`, qui est le numéro du réseau local virtuel (1 à 4094).
- `Name`, qui est un nom que l'utilisateur peut définir pour identifier le réseau local virtuel.

La fonction renvoie le chemin d'accès à l'objet du nouveau réseau local virtuel dans le paramètre de sortie `VLANpath`. Si cette action échoue, vérifiez `IANet_ExtendedStatus` pour obtenir le code d'erreur.

[Retour au début de la page](#)

---

## Modification des attributs d'un réseau local virtuel

La modification des attributs d'un réseau local virtuel :

- Nécessite un descripteur de session.
- Le PreCheck (Pré-contrôle) est disponible.
- Requiert l'appel de la méthode **Apply** avant l'exécution d'une opération.

Le client peut modifier les attributs `VLANNumber` et `VLANName` d'un réseau local virtuel. Pour modifier la priorité d'une carte,

obtenez tout d'abord l'instance de IANet\_VLAN pour la carte (p. ex., utilisez **IWbemServices::GetObject** avec le chemin d'accès à l'objet).

Modifiez ensuite VLANNumber ou VLANName en leur attribuant la valeur désirée. Enfin, appelez **IWbemClassObject::PutInstance** pour demander au fournisseur WMI de mettre à jour les attributs en passant l'indicateur WBEM\_FLAG\_UPDATE\_ONLY. Si cette action échoue, vérifiez IANet\_ExtendedStatus pour obtenir le code d'erreur.

[Retour au début de la page](#)

---

## Suppression d'un réseau local virtuel

La suppression d'un réseau local virtuel :

- Nécessite un descripteur de session.
- Le PreCheck (Pré-contrôle) est disponible.
- Requiert l'appel de la méthode **Apply** avant l'exécution d'une opération.

Pour supprimer un réseau local virtuel, appelez **IWbemServices::DeleteInstance** en passant le chemin d'accès à l'objet du réseau local virtuel à supprimer.

[Retour au début de la page](#)

---

## Mise à jour de Boot Agent

La mise à jour de Boot Agent :

- Ne nécessite pas de descripteur de session.
- Le PreCheck (Pré-contrôle) n'est pas disponible.
- Ne requiert pas l'appel de la méthode **Apply** avant l'exécution d'une opération.

Le client peut mettre à jour l'image de Boot Agent en utilisant des appels de méthode. Pour lire/écrire dans l'image de la mémoire flash, obtenez tout d'abord l'instance de IANet\_BootAgent pour la carte (p. ex., utilisez **IWbemServices::GetObject** avec le chemin d'accès à l'objet).

Exécutez ensuite ReadFlash() pour lire l'image de la ROM flash de démarrage existante, ou ProgramFlash() pour mettre à jour l'image de la ROM flash de démarrage. Si cette action échoue, vérifiez IANet\_ExtendedStatus pour obtenir le code d'erreur.

Tâche	Méthodes WMI	Résultat	Commentaire
Mettre à jour ou insérer une image de ROM de démarrage pour la carte	uint32 ProgramFlash( [IN, ValueMap {"0","1"}, Values {"Check Version", "Write Flash"}: Amended  ]  uint32 Action,  [IN]  uint8	Si l'action "Check Version" (Vérifier la version) est spécifiée, cette méthode renvoie un message d'avertissement si l'image de la ROM de démarrage mise à jour comme dans NewFlashData[] est plus ancienne que l'image existante sur la carte d'interface réseau.  Si l'action "Write" (Écrire) est spécifiée, la ROM flash de la carte d'interface réseau est mise à jour avec NewFlashData[].	Cette méthode est utilisée pour mettre à jour la ROM flash de la carte d'interface réseau. La liaison entre la carte d'interface réseau et le réseau est interrompue pendant la mise à jour de la mémoire flash.

	<pre>NewFlashData[], [OUT] string strErrorMessage );</pre>		
Lire l'image de la ROM de démarrage	<pre>uint32 ReadFlash( [OUT] uint8 FlashData[] );</pre>	FlashData[] contient la ROM flash de la carte d'interface réseau.	Cette méthode lit la ROM flash de la carte d'interface réseau qui peut être enregistrée dans un fichier.

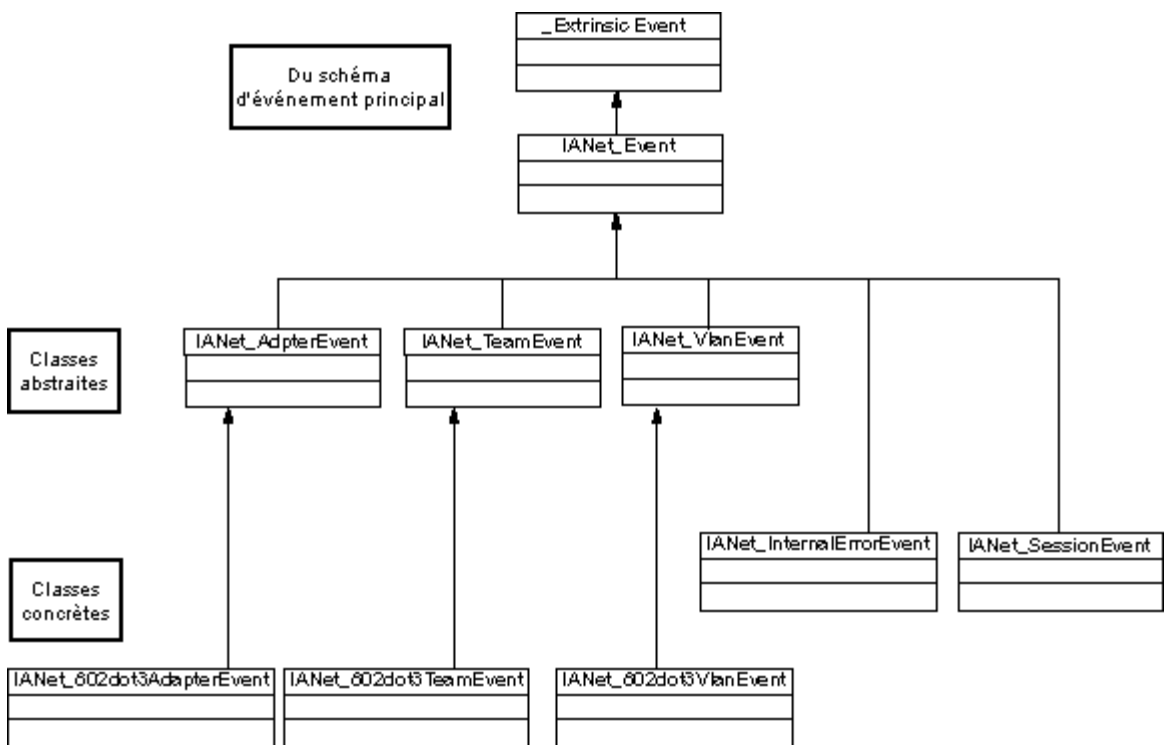
Veuillez lire toutes les [restrictions et clauses exonératoires de responsabilité](#).

[Retour à la table des matières](#)   [Retour au début de la page](#)

# Notifications d'événements : Guide de l'utilisateur des fournisseurs WMI et CDM pour cartes réseau Intel® PRO

[Classes d'événement concrètes](#)

[Enregistrement pour les événements](#)



## Classes d'événement concrètes

### IANet\_802dot3AdapterEvent

#### Objectif

Cet événement avertit le client de la modification de l'état ou de la configuration d'une carte.

#### Déclenchement

Cet événement se produit après un changement d'état de la carte ou après avoir modifié un paramètre d'une carte et appelé la méthode **Apply**.

#### Données d'événement

AdapterPath contient le chemin d'accès à l'objet de la carte ayant provoqué l'événement.

### IANet\_802dot3TeamEvent

## Objectif

Cet événement avertit le client de la modification de l'état ou de la configuration d'une association.

## Déclenchement

Cet événement se produit :

- après un changement de l'état d'une carte ;
- après avoir modifié un paramètre d'une association et appelé la méthode **Apply** ;
- après avoir modifié la configuration d'une association et appelé la méthode **Apply**.

## Données d'événement

TeamPath contient le chemin d'accès à l'objet de l'association ayant provoqué l'événement.

## IANet\_802dot3VlanEvent

### Objectif

Cet événement avertit le client de la modification de l'état ou de la configuration d'un réseau local virtuel.

### Déclenchement

Cet événement se produit :

- après un changement de l'état d'un réseau local virtuel ;
- après avoir modifié un paramètre d'un réseau local virtuel et appelé la méthode **Apply** ;
- après avoir modifié la configuration d'un réseau local virtuel et appelé la méthode **Apply**.

### Données d'événement

VlanPath contient le chemin d'accès à l'objet du réseau local virtuel ayant provoqué l'événement.

[Retour au début de la page](#)

---

## Enregistrement pour les événements

Les applications doivent utiliser **IWbemServices:: ExecNotificationQuery** ou **IWbemServices:: ExecNotificationQueryAsync** pour demander des notifications d'événements. Les requêtes suivantes sont des exemples de requêtes de notification. Cette liste est non exhaustive car de nombreuses requêtes sont possibles.

- **SELECT \* FROM IANet\_Event** — Requête pour tous les événements.
  - **SELECT \* FROM IANet\_AdapterEvent** — Requête pour tous les événements de carte.
  - **SELECT \* FROM IANet\_TeamEvent** — Requête pour tous les événements d'association.
  - **SELECT \* FROM IANet\_SessionEvent** — Requête pour tous les événements de session.
  - **SELECT \* FROM IANet\_VlanEvent** — Requête pour tous les événements de réseau local virtuel.
  - **SELECT \* FROM IANet\_InternalErrorEvent** — Requête pour tous les événements internes.
  - **SELECT \* FROM IANet\_AdapterEvent WHERE AdapterPath={IANet\_EthernetAdapter object path}** — Requête pour les événements de carte d'une carte particulière.
- 

Veillez lire toutes les [restrictions et clauses exonératoires de responsabilité](#).

---

[Retour à la table des matières](#)   [Retour au début de la page](#)

# Requêtes WQL optimisées : Guide de l'utilisateur des fournisseurs WMI et CDM pour cartes réseau Intel® PRO

## [Présentation](#)

[Obtention des paramètres d'une carte, d'une association ou d'un réseau local virtuel particulier](#)

[Obtention d'un seul paramètre](#)

---

## Présentation

Le fournisseur WMI est optimisé pour permettre aux applications d'obtenir les paramètres en utilisant des requêtes. Il peut reconnaître les requêtes suivantes et renvoie uniquement les objets correspondants. Le fournisseur WMI obtiendra tous les paramètres de tous les objets en utilisant toutes les autres requêtes, et CIMOM les filtrera avant qu'ils n'atteignent l'application. La récupération des données requises peut prendre quelques secondes si plusieurs cartes, associations ou réseaux locaux virtuels sont présents.

[Retour au début de la page](#)

---

## Obtention des paramètres d'une carte, d'une association ou d'un réseau local virtuel particulier

La requête suivante obtient uniquement les paramètres d'une carte, d'une association ou d'un réseau local virtuel particulier. WQL n'autorise pas de clauses supplémentaires dans la clause WHERE.

```
ASSOCIATORS OF {chemin de IANet_Configuration} WHERE AssocClass = IANet_SettingContext
```

[Retour au début de la page](#)

---

## Obtention d'un seul paramètre

La requête suivante peut être utilisée pour obtenir le paramètre unique d'un objet sans les interroger tous :

```
SELECT * FROM [CLASSE DE PARAMÈTRE] WHERE ParentId="[ID du périphérique]" AND ParentType="[type]" AND Caption="[NOM DU PARAMÈTRE]"
```

### Remarques :

- la classe doit être la classe de paramètre exacte et non une classe de base (p. ex., IANet\_SettingInt).
  - Les ParentTypes acceptables sont "NIC", "Team", "VLAN" ou "BootAgent".
  - ParentId est le GUID qui définit de manière unique l'objet avec le paramètre (pour une carte, il s'agit du DeviceId).
  - Cette méthode n'est pas recommandée pour obtenir les paramètres associés pour un objet ; la méthode conseillée est d'utiliser les associations. Cependant, WQL ne prend pas en charge la requête complexe requise (c.-à-d. WQL ne prend pas en charge ASSOCIATORS OF {chemin de IANet\_Configuration} WHERE AssocClass = IANet\_SettingContext AND Caption="[NOM DU PARAMÈTRE]").
- 

Veuillez lire toutes les [restrictions et clauses exonératoires de responsabilité](#).

---

[Retour à la table des matières](#)   [Retour au début de la page](#)

# Diagnostics : Guide de l'utilisateur des fournisseurs WMI et CDM pour cartes réseau Intel® PRO

[Classes de diagnostic](#)

[Entrées du registre](#)

[Consignation au journal](#)

[Classes d'association](#)

[Tests](#)

---

## Classes de diagnostic

### IANet\_DiagTest

#### Objectif

IANet\_DiagTest est une sous-classe de CIM\_DiagnosticTest. La classe offre une plate-forme générique pour exécuter et contrôler les tests de diagnostic d'une carte Ethernet prise en charge par Intel® PROSet. La superclasse, CIM\_DiagnosticTest, est conçue pour prendre en charge de manière générale les tests de tout matériel informatique sur un système compatible CIM. Les propriétés de la classe sont descriptives par nature et les systèmes de test sont fournis par les méthodes exposées.

#### Instances

La clé est Name, et dans ce fournisseur, il s'agit de la concaténation d'un index numérique du test du GUID de la carte référencée (p. ex., 1@{12345678-9ABC-DEF0-1234-123456789012}). La valeur de la clé est d'une certaine manière une information redondante, car toutes les informations de référence de carte et de test sont passées en tant que paramètres d'objet dans RunTest, ainsi que dans d'autres méthodes. Malgré cela, l'instance doit être cohérente avec les paramètres de la méthode sinon le fournisseur rejette la commande. La propriété Caption fournit le nom du test que cette instance va exécuter. D'autres propriétés offrent une description et des informations sur l'exécution.

#### Création d'instances

Vous ne pouvez pas créer des instances de IANet\_DiagTest.

#### Suppression d'instances

Vous ne pouvez pas supprimer des instances de IANet\_DiagTest.

#### Modification des propriétés

Aucune propriété de cette classe ne peut être modifiée par l'utilisateur.

#### Associations

- Une instance de IANet\_DiagTestForMSE associe un IANet\_DiagTest avec un IANet\_ManagedSystemElement. L'élément IANet\_ManagedSystemElement sera une instance de IANet\_EthernetAdapter.
- Une instance de IANet\_DiagResultForTest associe un IANet\_DiagTest avec une instance IANet\_DiagnosticResult.
- Une instance de IANetDiagSettingForTest associe un IANet\_DiagTest avec un IANet\_DiagSetting.

#### Propriétés non prises en charge

Install Date, OtherCharacteristicDescription

#### Méthodes

Cette classe prend en charge les méthodes suivantes :

- RunTest — Exécute un test tel que défini par les trois paramètres référençant :
  - SystemElement — Définit la carte sur laquelle le test doit être exécuté en faisant référence à une instance de SystemElement qui sera toujours la sous-classe de IANet\_EthernetAdapter.
  - Setting — Définit le test qui doit être exécuté et la manière selon laquelle il doit être exécuté, en faisant référence à une instance de CIM\_DiagnosticSetting qui sera toujours la sous-classe de IANet\_DiagSetting.
  - DiagnosticResult — Définit une instance de la classe CIM\_DiagnosticResult, qui sera toujours la classe de



IANet\_DiagResult.

- DiscontinueTest — Tente d'arrêter un test de diagnostic en cours, comme défini par deux paramètres référençant CIM\_ManagedSystemElement et CIM\_DiagnosticResult. Ces paramètres fonctionnent de la même manière que RunTest. Un troisième paramètre, TestingStopped, renvoie une valeur booléenne qui indique si la commande est parvenue à arrêter le test.
- ClearResults — Efface les résultats du test en utilisant les paramètres :
  - SystemElement
  - ResultsNotClearedLe paramètre ManagedSystemElement référencé combiné avec le chemin de cet objet est combiné avec des instances de référence de DiagnosticResultForMSE qui seront supprimées. De même, toutes les références des objets DiagnosticResult référencés par DiagnosticResultForMSE seront supprimées. Toutes les instances de DiagnosticResultForTest faisant référence aux objets DiagnosticResult supprimés seront également supprimées. Enfin, le paramètre de sortie ResultsNotCleared du tableau de chaînes listera les clés de DiagnosticResults qui n'ont pu être effacées.

## Hiérarchie des classes

Pour CimV2. Les propriétés et les méthodes inutilisées ne sont pas affichées.

- CIM\_ManagedElement :
  - Caption
  - Description
- CIM\_ManagedSystemElement :
  - Install Date
  - Name
  - Status
- CIM\_LogicalElement
- CIM\_Service :
  - Clé
    - Name (chaîne)
  - Propriétés
    - Caption (chaîne)
    - CreationClassName (chaîne)
    - Description (chaîne)
    - Started (booléen)
    - StartMode (chaîne)
    - Status (chaîne)
    - SystemCreationClass (chaîne)
    - SystemName (chaîne)
- CIM\_DiagnosticTest :
  - Propriétés
    - Characteristics (tableau uint16)
    - IsInUse (booléen)
    - ResourcesUsed (tableau uint16)
  - Méthodes
    - RunTest
    - ClearResults
    - DiscontinueTest

## Exécution de RunTest et d'autres méthodes dans WbemTest

La méthode RunTest du fichier MOF se présente comme suit :

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

Les deux premiers paramètres sont des paramètres d'entrée. Vous devez obtenir le chemin des deux objets référencés. Vous devez également obtenir le chemin de l'objet IANet\_DiagTest qui exporte l'objet RunTest.

1. À partir de la boîte de dialogue de test WBEM principale, cliquez sur **Connect**.

2. Entrez la combinaison serveur/espace de noms appropriée. Les espaces de noms IntelNCS et CimV2 sont pris en charge.
3. Cliquez sur le bouton **Enum instances** du test WBEM et entrez `IANet_DiagTest`.
4. Double-cliquez sur l'instance de `IANet_DiagTest`. Le nom sera au format `X@[AdapterGUID]`, `X` est le nom du test et `AdapterGUID` sera le nom de la carte, comme pour la clé `Name` de `IANet_EthernetAdapter`.
5. Ci-dessous se trouve un exemple de chemin d'accès à l'objet récupéré :  
`\\MYCOMPUTER\root\Cimv2:IANet_DiagTest.Name="1@{4A0CDABE-F6C3-45D0-B60D-F6E7BAFA2C2C}"`
6. Enregistrez le chemin d'accès à l'objet.
7. Cliquez sur le bouton **Enum Instances** du test WBEM et entrez `IANet_EthernetAdapter`.
8. Double-cliquez sur la carte à tester.
9. Ci-dessous se trouve un exemple de chemin d'accès à l'objet récupéré :  
`\\MYCOMPUTER\root\cimv2:IANet_EthernetAdapter.DeviceID="{4A0CDABE-F6C3-45D0-B60D-F6E7BAFA2C2C}"`
10. Enregistrez le chemin d'accès à l'objet.
11. Cliquez sur le bouton **Enum Instances** du test WBEM et entrez `IANet_DiagSetting`.
12. Double-cliquez sur le paramètre représentant la combinaison carte/test.
13. Ci-dessous se trouve un exemple de chemin d'accès à l'objet récupéré :  
`\\MYCOMPUTER\root\cimv2:IANet_DiagSetting.SettingID="1@{4A0CDABE-F6C3-45D0-B60D-F6E7BAFA2C2C}"`
14. Enregistrez le chemin d'accès à l'objet.
15. À partir de la boîte de dialogue de test WBEM principale, cliquez sur **Execute Method**.
16. Collez le chemin de l'objet `IANet_DiagTest` dans la boîte de dialogue. Cliquez sur **OK**.
17. Sélectionnez le test sur la liste déroulante sous la méthode.
18. Cliquez sur le bouton **Edit In Parameters**.
19. Pour `RunTest`, les paramètres d'entrée sont `Setting` et `SystemElement`. Collez les chemins des objets `Setting` et `Adapter` enregistrés précédemment et fermez la boîte de dialogue.
20. Cliquez sur le bouton **Execute**.
21. Énumérez la classe `IANet_DiagResult` de la même manière que les paramètres d'entrée.
22. Examinez l'objet du résultat sélectionné si nécessaire.

## IANet\_DiagSetting

### Objectif

Les instances de `IANet_DiagSetting` fournissent des directives d'exécution spécifiques des tests de diagnostic. Les directives utilisées sont communes à tous les tests et sont liées à la superclasse `CIM_DiagnosticSetting`. Elles comprennent des propriétés comme `ReportSoftErrors` et `HaltOnError`. Aucune autre propriété n'est ajoutée à la sous-classe `IANet_DiagSetting`.

### Création d'instances

Vous ne pouvez pas créer des instances de cette classe.

### Suppression d'instances

Vous ne pouvez pas supprimer des instances de cette classe.

### Modification des propriétés

`UpdateInstanceAsync` est mis en oeuvre et peut être utilisé pour définir les paramètres de test sur "Halt On Error", "Report Soft Errors", "Report Status Messages", "Quick Mode", "Test Warning Level" et "Percent Of Test Coverage".

### Associations

Une instance de `IANetDiagSettingForTest` associe un `IANet_DiagTest` avec un `IANet_DiagSetting`.

### Propriétés non prises en charge

Les propriétés suivantes ne sont pas prises en charge par NCS :

- o Caption
- o Description

### Méthodes

Aucune

### Hiérarchie des classes

Pour CimV2. Les propriétés et les méthodes inutilisées ne sont pas affichées.

- CIM\_ManagedElement
- o CIM\_Setting :
  - Propriétés
  - SettingID
  - Méthodes (aucune n'est prise en charge)
  - VerifyOKToApplyToMSE
  - ApplyToMSE
  - VerifyOKToApplyToCollection
  - ApplyToCollection
  - VerifyOKToApplyIncrementalChangeToMSE
  - ApplyIncrementalChangesToMSE
  - ApplyIncrementalChangeToCollection
- o CIM\_DiagnosticSetting :
  - Clé
    - SettingID (chaîne)
  - Propriétés
    - TestWarningLevel (uint16)
    - ReportSoftErrors (booléen)
    - ReportStatusMessages (booléen)
    - HaltOnError (booléen)
    - QuickMode (booléen)
    - PercentOfTestCoverage (uint8)

## IANet\_DiagResult

### Objectif

Les instances de IANet\_DiagResult affichent les données des résultats d'une exécution de test particulière sur une carte spécifique. Les instances de cette classe correspondent exactement aux instances de IANet\_DiagTest et de IANet\_DiagSetting.

### Instances

Les instances de IANet\_DiagResult correspondent aux résultats d'une exécution de test particulière sur une carte spécifique. Le format de la clé est identique à IANet\_DiagTest et IANet\_DiagSetting. L'instance peut stocker tout résultat de test arbitraire, et toute donnée ne correspondant pas aux propriétés définies peut être placée dans la propriété du tableau TestResults. Chaque fois qu'un test est exécuté sur une carte, la nouvelle instance remplace l'instance existante des résultats du test correspondant à cette combinaison carte/test.

### Création d'instances

Vous ne pouvez pas créer des instances de cette classe.

### Suppression d'instances

Vous ne pouvez pas supprimer des instances de cette classe.

### Modification d'instances

Vous ne pouvez pas modifier les instances de cette classe.

### Associations

Une instance de IANet\_DiagResultForTest associe un IANet\_DiagTest avec une instance IANet\_DiagnosticResult.

### Propriétés non prises en charge

Les propriétés suivantes ne sont pas prises en charge par NCS :

- o EstimatedTimeOfPerforming
- o HaltOnError
- o OtherStateDescription
- o ReportSoftErrors
- o TestWarningLevel

### Méthodes

Aucune

### Hiérarchie des classes

Pour CimV2. Les propriétés et les méthodes inutilisées ne sont pas affichées.

- o CIM\_DiagnosticResult :
  - Clés
    - DiagnosticCreationClassName (chaîne)
    - DiagnosticName (chaîne)
    - ExecutionID (chaîne)
    - DiagSystemCreationClassName (chaîne)
    - DiagSystemName (chaîne)
  - Propriétés
    - TimeStamp (chaîne)
    - IsPackage (booléen)
    - TestStartTime (DateHeure)
    - TestCompletionTime (DateHeure)
    - TestState (uint16)
    - TestResults (chaîne)
    - PercentComplete (uint8)
- o IANet\_DiagResult

[Retour au début de la page](#)

---

## Entrées du registre

Les entrées suivantes sont entrées dans le registre pendant l'installation sous **HKLM\Software\Intel\NETWORK\_SERVICES\NCS\NcsDiag**. Ces clés et valeurs contrôlent l'exécution des tests de diagnostic et sont définies ci-dessous.

La table suivante contient les valeurs et les types de clés, ainsi qu'une brève explication sur leur utilisation :

Valeur	Type	Valeur par défaut	Utilisation
Check Time	REG_DWORD	2 secondes	Délai entre les vérifications de l'exécution du test envoyeur/répondeur
Enable	REG_DWORD	0	Activer (1) ou Désactiver (0) l'utilisation d'un fichier journal de résultats.
FileAppend	REG_DWORD	1	Annexer (1) le fichier journal des résultats à un fichier journal existant. Si 0, le fichier existant est remplacé.
LogFileName	REG_SZ	NcsDiag.log	Le nom du fichier journal de résultats.
MaxFileSize	REG_DWORD	0x10000	La taille maximale du fichier journal de résultats.
MaxPktsRcvd	REG_DWORD	200	En mode rapide (Quick Mode de IANet_DiagSetting), le test envoyer/recevoir s'arrête lorsque le nombre de paquets reçus est supérieur à cette valeur.
TimeoutSndRsp	REG_DWORD	100	Le test s'arrête lorsque sa durée (en secondes) dépasse cette valeur.

[Retour au début de la page](#)

---

## Consignation au journal

### Journal des résultats

Le journal des résultats affiche principalement des informations qui peuvent être également obtenues à partir des objets IANet\_DiagResult. La différence est que les informations obtenues d'un navigateur CIM n'affichent que les derniers résultats immédiats de tests particuliers sur une carte particulière. Les tests suivants remplacent les résultats des tests précédents. Le

journal des résultats est plus pratique pour configurer et afficher plusieurs exécutions de tels tests.

## Activation du journal des résultats

Pour activer le journal des résultats :

1. Dans la clé de registre **HKLM\Software\Intel\NETWORK\_SERVICES\NCS\NCS Diag**, affectez la valeur 1 à Enable.
2. Définissez un nom de fichier préféré pour LogFileName ou conservez le nom par défaut **NcsDiag.log**.
3. Le fichier journal se trouve dans le répertoire indiqué par la valeur InstalledDir.

[Retour au début de la page](#)

---

## Classes d'association

Classe d'association	Propriété/valeur de référence	Propriété/valeur de référence
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

[Retour au début de la page](#)

---

## Tests

Les tests mis en oeuvre peuvent être exécutés sur un ou deux ordinateurs. Les descriptions détaillées des tests ne sont pas incluses dans ce document, car le fournisseur CDM est prévu pour être une plate-forme de test générique non dépendante des caractéristiques des tests. Cependant, une certaine dépendance expliquée dans cette section existe dans le code.

### Tests d'une carte unique

Les tests suivants sont exécutés sur une carte unique et ne nécessitent aucune interaction avec une autre carte :

- o Mémoire EEPROM
- o Registres de contrôle
- o Bouclage MAC
- o Bouclage PHY
- o Liaison

Tous les messages d'erreur de ces tests sont le résultat des codes d'erreur HRESULT renvoyés des appels vers les couches de piles inférieures. Les codes d'erreur sont stockés intérieurement sous forme de codes d'erreur, et ils ne sont pas traduits en messages d'erreur jusqu'à ce que les objets IANet\_DiagResult soient déréférencés par énumération, ou que des appels d'objet soient réceptionnés par une application de gestion.

### Tests nécessitant deux cartes

Les tests envoyeur/répondeur sont des tests interdépendants au cours desquels une carte (envoyeur) envoie des paquets à une autre carte (répondeur) qui, à son tour, renvoie les paquets à l'envoyeur pour fermer la boucle. Ces tests sont identiques et peuvent être exécutés à partir d'Intel® PROSet. Cependant, Intel PROSet n'utilise pas CDM et ne permet pas d'exécuter deux tests simultanément sur le même ordinateur. CDM permet l'exécution simultanée de différents tests sur un même ordinateur.

### Tests Envoyeur/Répondeur

Le test envoyeur/répondeur requiert deux cartes Intel (1 envoyeur et 1 répondeur). Ce test est le seul qui soit exécuté sur la base d'un second thread qui est actif jusqu'à ce que le test s'arrête suivant des critères définis ou que le test soit arrêté par le premier thread. Les critères d'achèvement du test sont des temporisations basées sur la durée du test ou sur le nombre de paquets reçus. Ces deux valeurs sont obtenues à partir du registre. Le test ne peut se terminer en fonction du nombre de paquets reçus que si le mode rapide (Quick Mode) a été activé. Ce mode est une propriété de la classe

IANet\_DiagSetting et peut donc être défini pour chaque carte. Les répondeurs CDM répondent aux répondeurs PROSet et vice versa.

Deux types de valeurs d'erreur sont renvoyés par les tests envoyeur/répondeur. Tout d'abord, un code d'erreur (HRESULT) peut être renvoyé par les couches inférieures. Ensuite, le thread du test renvoie des statistiques intermédiaires et finales pendant l'exécution du test, sauf si le test est terminé prématurément en raison d'un code d'erreur renvoyé. Les statistiques comprennent :

- o l'état de la liaison ;
- o l'utilisation de la négociation automatique ;
- o les collisions ;
- o les paquets reçus ;
- o le nombre total de paquets reçus ;
- o les paquets envoyés ;
- o les transmissions réussies ;
- o les réceptions réussies ;
- o les erreurs de transmission ;
- o les erreurs de réception ;
- o les collisions ;
- o les phases de diagnostic.

---

Veuillez lire toutes les [restrictions et clauses exonératoires de responsabilité](#).

---

[Retour à la table des matières](#)   [Retour au début de la page](#)

# Exécution de méthodes dans IANet\_DiagTest : Guide de l'utilisateur des fournisseurs WMI et CDM pour cartes réseau Intel® PRO

---

## Exécution de RunTest et d'autres méthodes dans WbemTest

La méthode RunTest du fichier MOF se présente comme suit :

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

Les deux premiers paramètres sont des paramètres d'entrée. Vous devez obtenir le chemin des deux objets référencés. Vous devez également obtenir le chemin de l'objet IANet\_DiagTest qui exporte l'objet RunTest.

1. À partir de la boîte de dialogue de test WBEM principale, cliquez sur **Connect**.
2. Entrez la combinaison serveur/espace de noms appropriée. Les espaces de noms IntelNCS et CimV2 sont pris en charge.
3. Cliquez sur le bouton **Enum instances** du test WBEM et entrez IANet\_DiagTest.
4. Double-cliquez sur l'instance de IANet\_DiagTest. Le nom sera au format X@[AdapterGUID], X est le nom du test et AdapterGUID sera le nom de la carte, comme pour la clé Name de IANet\_EthernetAdapter.
5. Ci-dessous se trouve un exemple de chemin d'accès à l'objet récupéré :  
`\\MYCOMPUTER\root\Cimv2:IANet_DiagTest.Name="1@{4A0CDABE-F6C3-45D0-B60D-F6E7BAFA2C2C}"`
6. Enregistrez le chemin d'accès à l'objet.
7. Cliquez sur le bouton **Enum instances** du test WBEM et entrez IANet\_EthernetAdapter.
8. Double-cliquez sur la carte à tester.
9. Ci-dessous se trouve un exemple de chemin d'accès à l'objet récupéré :  
`\\MYCOMPUTER\root\cimv2:IANet_EthernetAdapter.DeviceID="{4A0CDABE-F6C3-45D0-B60D-F6E7BAFA2C2C}"`
10. Enregistrez le chemin d'accès à l'objet.
11. Cliquez sur le bouton **Enum Instances** du test WBEM et entrez IANet\_DiagSetting.
12. Double-cliquez sur le paramètre représentant la combinaison carte/test.
13. Ci-dessous se trouve un exemple de chemin d'accès à l'objet récupéré :  
`\\MYCOMPUTER\root\cimv2:IANet_DiagSetting.SettingID="1@{4A0CDABE-F6C3-45D0-B60D-F6E7BAFA2C2C}"`
14. Enregistrez le chemin d'accès à l'objet.
15. À partir de la boîte de dialogue de test WBEM principale, cliquez sur **Execute Method**.
16. Collez le chemin de l'objet IANet\_DiagTest dans la boîte de dialogue. Cliquez sur **OK**.
17. Sélectionnez le test sur la liste déroulante sous la méthode.
18. Cliquez sur le bouton **Edit In Parameters**.
19. Pour RunTest, les paramètres d'entrée sont Setting et SystemElement. Collez les chemins des objets Setting et Adapter enregistrés précédemment et fermez la boîte de dialogue.
20. Cliquez sur le bouton **Execute**.
21. Énumérez la classe IANet\_DiagResult de la même manière que les paramètres d'entrée.
22. Examinez l'objet du résultat sélectionné si nécessaire.

---

Veuillez lire toutes les [restrictions et clauses exonératoires de responsabilité](#).

---

## Récapitulatif des classes CIM : Guide de l'utilisateur des fournisseurs WMI et CDM pour cartes réseau Intel® PRO

Classe	Création possible ?	Suppression possible ?	Méthodes de mise en oeuvre	Propriétés configurables	Propriétés non prises en charge	Nombre d'instances	Classes d'association liées
IANet_802dot1QVLANService	N	N	CreateVLAN	GRVPEEnabled, JoinTime	Description, Install Date, Started, StartMode, Status	Une instance pour chaque association ou carte prenant en charge les réseaux locaux virtuels	IANet_Device802dot1QVLANServiceImplementation, IANet_VLANFor
IANet_AdapterConfiguration	N	N	Aucune	Aucune	Aucune	Une instance pour chaque carte	Cette classe associe IANet_EthernetAdapter avec IANet_Configuration.
IANet_AdapterDevice	N	N	Aucune	Aucune	Aucune	Une instance pour chaque carte non fantôme	Cette classe associe IANet_EthernetAdapter avec IANet_EthernetPCIDevice.
IANet_AdapterProtocolImplementation	N	N	Aucune	Aucune	Aucune	Une instance pour chaque point de terminaison de protocole IP lié à une carte	Cette classe associe IANet_EthernetAdapter avec IANet_IPProtocolEndpoint.
IANet_BootAgent	N	N	ProgramFlash ReadFlash	Aucune	Caption, Description, InstallDate, Started, StartMode, Status	Une instance pour chaque carte prenant en charge l'agent d'amorçage	IANet_DeviceBootServiceImplementation, IANet_BootAgentConfiguration.
IANet_BootAgentConfiguration	N	N	Aucune	Aucune	Aucune	Une instance pour chaque agent d'amorçage.	Cette classe associe IANet_BootAgent avec IANet_Configuration.
IANet_Configuration	N	N	Aucune	Aucune	Aucune	Une instance pour chaque carte, réseau local virtuel et	IANet_AdapterConfiguration, IANet_VLANConfiguration, IANet_SettingContext.



						association	
IANet_Device802dot1QVLANSERVICEImplementation	N	N	Aucune	Aucune	Aucune	Une instance pour chaque association ou carte prenant en charge les réseaux locaux virtuels	Cette classe associe IANet_EthernetAdapter avec IANet_802dot1QVLANSERVICE.
IANet_DiagTest	N	N	RunTest, DiscontinueTest, ClearResults	Aucune	InstallDate, OtherCharacteristicsDescription	Une instance pour chaque combinaison carte/test	IANet_DiagTestForMSE, IANet_DiagResultForTest, IANet_DiagSettingForTest
IANet_DiagSetting	N	N	Aucune	HaltOnError, ReportSoftErrors, ReportStatusMessages, QuickMode, PercentOfTestCoverage, TestWarningLevel	Caption, Description	Une instance pour chaque combinaison carte/test	IANet_DiagSettingForTest
IANet_DiagResult	N	N	Aucune	Aucune	EstimatedTimeOfPerforming, HaltOnError, OtherStateDescription, ReportSoftErrors, TestWarningLevel	Une instance pour chaque combinaison carte/test	IANet_DiagResultForTest, IANet_DiagResultForMSEM
IANet_EthernetAdapter	N	O	IdentifyAdapter HasVLANs IsPowerMgmtSupported GetPowerUsage SetPowerUsage GetPowerUsageOptions SetPowerUsageOptions TestCable AdvancedTestCable TestLinkSpeed	Aucune	AutoSense – (présenté comme un paramètre), ErrorCleared, OtherIdentifyingInfo, IdentifyingDescriptions, InstallDate, LastErrorCode, MaxDataSize, MaxQuiesceTime, PowerManagementCapabilities– (présenté comme une méthode), PowerManagementSupported– (présenté comme une méthode), PowerOnHours , ShortFramesReceived, SymbolErrors, TotalPowerOnHours	Une instance pour chaque carte Intel® PROSet, carte fantôme et association installées et prises en charge	IANet_AdapterProtocolImplementation, IANet_AdapterDevice, IANet_AdapterConfiguration, IANet_TeamedMemberAdapter, IANet_NetworkVirtualAdapter, IANet_Device802dot1QVLANSERVICEImplementation, IANet_DeviceBootSERVICEImplementation
IANet_EthernetPCIDevice	N	N	Aucune	Aucune	AdditionalAvailability, Capabilities, CapabilityDescriptions, Caption, Description, DeviceSelectTiming, ErrorCleared, ErrorDescription, IdentifyingDescription, InstallDate, LastErrorCode, MaxNumberController, MaxQuiesceTime, Name, OtherIdentifyingInfo, PowerManagementCapabilities, PowerManagementSupported, PowerOnHours, ProtocolDescription, ProtocolSupported, SelfTestEnabled, Status, StatusInfo, TimeOfLastReset,	Une instance pour chaque carte PCI qui est une carte Ethernet prise en charge par Intel PROSet	IANet_AdapterDevice

					TotalPowerOnHours		
IANet_IPProtocolEndpoint	N	N	Aucune	Aucune	Caption, Description, InstallDate, NameFormat, OtherTypeInformation, Status	Une instance pour chaque liaison de pile de protocole IP avec un point de terminaison Intel pris en charge	IANet_AdapterProtocolImplementation, IANet_VLANProtocolDependency
IANet_NetService	N	N	GetSessionHandle, Apply, ReleaseSessionHandle, Cancel	Aucune	Caption, Description, Install Date, Started, Start Mode, Status	Exactement une instance.	Aucune
IANet_NetworkVirtualAdapter	N	N	Aucune	Aucune	Aucune	Une instance pour chaque association	Cette classe associe IANet_TeamOfAdapters avec an IANet_EthernetAdapter.
IANet_PCIDevice	N	N	Aucune	Aucune	AdditionalAvailability, Capabilities, CapabilityDescriptions, Caption, DeviceSelectTiming, ErrorCleared, ErrorDescription, IdentifyingDescription, InstallDate, LastErrorCode, MaxNumberController, MaxQuiesceTime, Name, OtherIdentifyingInfo, PowerManagementCapabilities, PowerManagementSupported, PowerOnHours, ProtocolDescription, ProtocolSupported, SelfTestEnabled, TimeOfLastReset, TotalPowerOnHours	Une instance pour chaque carte PCI qui est un périphérique réseau du système	
IANet_Setting	N	N	Aucune	Aucune	SettingID	Il s'agit d'une classe abstraite.	IANet_SettingContext
IANet_SettingContext	N	N	Aucune	Aucune	Aucune	Une instance pour chaque paramètre	Cette classe associe IANet_Setting avec IANet_Configuration.
IANet_SettingInt	N	N	Aucune	CurrentValue	SettingID	Une instance pour chaque paramètre de nombre entier.	IANet_SettingContext
IANet_SettingMultiSelection	N	N	Aucune	CurrentValue	SettingID	Une instance pour chaque paramètre multi-sélection.	IANet_SettingContext

IANet_SettingSlider	N	N	Aucune	CurrentValue	SettingID	Une instance pour chaque paramètre de curseur	IANet_SettingContext
IANet_SettingString	N	N	Aucune	CurrentValue	SettingID	Une instance pour chaque paramètre de chaîne	IANet_SettingContext
IANet_TeamedMemberAdapter	O	O	Aucune	AdapterFunction	PrimaryAdapter, ScopeOfBalancing	Une instance pour chaque carte se trouvant dans une association	Cette classe associe IANet_TeamOfAdapters avec IANet_EthernetAdapter.
IANet_TeamOfAdapters	O	O	TestSwitchConfiguration	TeamingMode	Install Date, Status	Une instance pour chaque association.	IANet_NetworkVirtualAdapter, IANet_TeamedMemberAdapter
IANet_VLAN	N	O	Aucune	VLANNumber, Caption	Description, Install Date, StartMode, Status	Une instance pour chaque réseau local virtuel.	IANet_VLANFor
IANet_VLANConfiguration	N	N	Aucune	Aucune	Aucune	Une instance pour chaque réseau local virtuel.	Cette classe associe IANet_VLAN avec IANet_Configuration.
IANet_VLANFor	N	N	Aucune	Aucune	Aucune	Une instance pour chaque réseau local virtuel.	Cette classe associe IANet_VLAN avec IANet_802dot1QVLANService.
IANet_VLANProtocolDependency	N	N	Aucune	Aucune	Aucune	Une instance pour chaque réseau local virtuel.	Cette classe associe IANet_VLAN avec IANet_IPProtocolEndpoint.

Veillez lire toutes les [restrictions et clauses exonératoires de responsabilité](#).

[Retour à la table des matières](#) [Retour au début de la page](#)

# Contrat de licence du logiciel Intel (Licence finale) : Guide de l'utilisateur des fournisseurs WMI et CDM pour cartes réseau Intel® PRO

## **IMPORTANT - À LIRE AVANT TOUTE COPIE, INSTALLATION OU UTILISATION**

**Veillez ne pas utiliser ou charger ce logiciel et tout programme associé (regroupés sous la désignation « Logiciel ») tant que vous n'avez pas lu attentivement les conditions suivantes. En chargeant ou en utilisant le Logiciel, vous acceptez les termes de ce contrat. Si vous êtes en désaccord avec les termes de ce contrat, veuillez ne pas installer ni utiliser le Logiciel.**

### **CONTRATS DE LICENCE :**

- Si vous êtes un administrateur réseau, reportez-vous à la « Licence de site » ci-dessous.
- Si vous êtes un utilisateur final, reportez-vous à la « Licence utilisateur unique » ci-dessous.
- Si vous êtes un fabricant OEM, reportez-vous à la « Licence OEM ».

**Licence de site.** Vous pouvez copier le Logiciel sur les ordinateurs de votre organisation afin que les employés puissent l'utiliser et vous pouvez sauvegarder un nombre raisonnable de copies du Logiciel, conformément à ces conditions :

1. Ce Logiciel est concédé sous licence pour une utilisation exclusive avec les produits composants Intel. L'utilisation du Logiciel avec des produits composants autres que Intel n'est pas couverte par cette licence.
2. Vous n'êtes pas autorisé à copier, à modifier, à louer, à vendre, à distribuer ou à transférer toute partie du Logiciel (sauf dans les conditions prévues par ce contrat) et vous acceptez d'interdire toute procédure visant à copier le Logiciel de manière non autorisée.
3. Vous n'êtes pas autorisé à reconstituer la logique du Logiciel, à le décompiler ou à le désassembler.
4. Vous n'êtes pas autorisé à concéder une sous-licence du Logiciel ou à permettre à plusieurs utilisateurs de l'utiliser simultanément.
5. Le Logiciel peut inclure des portions sujettes à des termes complémentaires aux présents, définis dans une licence accompagnant ces portions.

**Licence utilisateur unique.** Vous pouvez copier le Logiciel sur un seul ordinateur pour une utilisation personnelle et non commerciale, et vous pouvez effectuer une copie de sauvegarde du Logiciel, conformément à ces conditions :

1. Ce Logiciel est concédé sous licence pour une utilisation exclusive avec les produits composants Intel. L'utilisation du Logiciel avec des produits composants autres que Intel n'est pas couverte par cette licence.
2. Vous n'êtes pas autorisé à copier, à modifier, à louer, à vendre, à distribuer ou à transférer toute partie du Logiciel (sauf dans les conditions prévues par ce contrat) et vous acceptez d'interdire toute procédure visant à copier le Logiciel de manière non autorisée.
3. Vous n'êtes pas autorisé à reconstituer la logique du Logiciel, à le décompiler ou à le désassembler.
4. Vous n'êtes pas autorisé à concéder une sous-licence du Logiciel ou à permettre à plusieurs utilisateurs de l'utiliser simultanément.
5. Le Logiciel peut inclure des portions sujettes à des termes complémentaires aux présents, définis dans une licence accompagnant ces portions.

**Licence OEM.** Vous êtes autorisé à reproduire et à distribuer le Logiciel en tant que partie intégrale de votre produit ou en l'incorporant à celui-ci, ou en tant que mise à jour d'entretien autonome du Logiciel pour des utilisateurs existants de vos produits, à l'exception de tout autre produit autonome et sous réserve des conditions suivantes :

1. Ce Logiciel est concédé sous licence pour une utilisation exclusive avec les produits composants Intel. L'utilisation du Logiciel avec des produits composants autres que Intel n'est pas couverte par cette licence.
2. Vous n'êtes pas autorisé à copier, à modifier, à louer, à vendre, à distribuer ou à transférer toute partie du Logiciel (sauf dans les conditions prévues par ce contrat) et vous acceptez d'interdire toute procédure visant à copier le Logiciel de manière non autorisée.
3. Vous n'êtes pas autorisé à reconstituer la logique du Logiciel, à le décompiler ou à le désassembler.
4. Vous êtes autorisé à distribuer le Logiciel à vos clients conformément à un contrat de licence écrit. L'ouverture de

l'emballage scellé entraîne l'acceptation implicite des termes de l'accord de licence. Cette licence doit au minimum protéger les droits de propriété du Logiciel d'Intel.

5. Le Logiciel peut inclure des portions sujettes à des termes complémentaires aux présents, définis dans une licence accompagnant ces portions.

**EXCLUSION DES AUTRES DROITS.** Aucun droit ou licence ne vous est accordé par Intel, de manière explicite ou implicite, en ce qui concerne toute information ou tout brevet propriétaire, droit d'auteur, moyen de masquage, marque commerciale, secret industriel ou tout autre droit de propriété intellectuelle contrôlé par Intel, sauf stipulation contraire décrite dans ce Contrat.

**PROPRIÉTÉ DU LOGICIEL ET DROITS D'AUTEUR.** Intel ou ses fournisseurs demeurent les propriétaires de toutes les copies du Logiciel. Le logiciel est protégé par la loi sur le copyright, les lois en vigueur aux États-Unis et dans d'autres pays, ainsi que par les dispositions de traités internationaux. Vous n'êtes pas autorisé à retirer la mention de copyright du Logiciel. Intel peut modifier le Logiciel ou des éléments référencés dans ce document, à tout moment et sans préavis, mais n'est pas dans l'obligation de mettre à jour le Logiciel ni d'offrir une assistance. Sauf stipulation expresse contraire, Intel exclut tout droit, explicite ou implicite, vis à vis des brevets, copyrights et marques Intel, et tout autre droit de propriété intellectuelle. Vous pouvez transférer le Logiciel uniquement si le bénéficiaire accepte d'être lié par ces termes et si vous ne conservez aucune copie du Logiciel.

**GARANTIE LIMITÉE DU SUPPORT.** Si le logiciel a été remis par Intel sur un support physique, Intel garantit que celui-ci est exempt de défauts matériels pendant une période de quatre-vingt-dix jours à compter de la date de livraison par Intel. Si un tel défaut est détecté, renvoyez le support à Intel qui pourra choisir de le remplacer ou d'employer une autre méthode de livraison.

**EXCLUSION DE TOUTE AUTRE GARANTIE. À L'EXCEPTION DES MODALITÉS STIPULÉES CI-DESSUS, LE LOGICIEL EST FOURNI "TEL QUEL" SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE DE TOUTE NATURE, Y COMPRIS DES GARANTIES DE COMMERCIALISATION, NON CONTREFAÇON OU ADAPTATION À UNE UTILISATION PARTICULIÈRE.** Intel ne garantit pas et n'assume pas la précision ni l'exhaustivité des informations, des textes, des graphiques, des liens ou d'autres éléments contenus dans le Logiciel et ne peut en être tenu responsable.

**LIMITATION DE RESPONSABILITÉ. EN AUCUN CAS, INTEL OU SES FOURNISSEURS NE POURRONT ÊTRE TENUS POUR RESPONSABLES DE TOUT DOMMAGE, DE QUELQUE NATURE QUE CE SOIT (Y COMPRIS, MAIS SANS SE LIMITER À CELLES-CI, TOUTE PERTE DE MANQUE À GAGNER, INTERRUPTION D'ACTIVITÉ OU PERTE D'INFORMATIONS), RÉSULTANT DE L'UTILISATION OU DE L'IMPOSSIBILITÉ D'UTILISER LE LOGICIEL, ET CE MÊME SI INTEL A ÉTÉ PRÉVENU DE L'ÉVENTUALITÉ DE TELS DOMMAGES. CERTAINES JURIDICTIONS N'AUTORISENT PAS L'EXCLUSION OU LA LIMITATION DE RESPONSABILITÉ POUR LES GARANTIES IMPLICITES OU LES DOMMAGES INDIRECTS OU ACCESSOIRES, DE SORTE QUE LA LIMITATION CI-DESSUS PEUT NE PAS VOUS ÊTRE APPLICABLE. VOUS POUVEZ ÉGALEMENT DISPOSER D'AUTRES DROITS QUI VARIENT DE JURIDICTION EN JURIDICTION.**

**RÉSILIATION DE CE CONTRAT.** Intel peut résilier ce contrat à tout moment si vous n'en respectez pas les termes. Dans ce cas, vous devez immédiatement détruire le Logiciel et retourner toutes les copies dudit Logiciel à Intel.

**LOIS APPLICABLES.** Toute réclamation établie dans le cadre de ce contrat sera régie par les lois de l'État de Californie (États-Unis), à l'exception de ses principes de conflit des lois et de la convention des Nations Unies sur les contrats pour la vente de biens. Vous n'êtes pas autorisé à exporter le Logiciel en infraction des lois et des réglementations d'exportation applicables. Intel n'est contraint par aucun autre contrat à moins que celui-ci ne soit établi par écrit et signé par un représentant autorisé d'Intel.

**LIMITATION DES DROITS DU GOUVERNEMENT.** Le Logiciel est fourni avec des « DROITS LIMITÉS ». L'utilisation, la duplication ou la diffusion par le gouvernement est soumise à des restrictions comme indiqué dans les clauses FAR52.227-14 et DFAR252.227-7013 et seq. ou son successeur. L'utilisation du Logiciel par le gouvernement constitue la reconnaissance des droits de propriété de Intel mentionnés dans ce document. Le sous-traitant ou le fabricant est Intel.

---

Veillez lire toutes les [restrictions et clauses exonératoires de responsabilité](#).

---

[Retour à la table des matières](#)   [Retour au début de la page](#)

# Assistance : Guide de l'utilisateur des fournisseurs WMI et CDM pour cartes réseau Intel® PRO

---

## Sites Web et Internet

<http://www.dell.com>

---

## Techniciens d'assistance à la clientèle

Si les procédures de dépannage décrites dans ce document ne vous aident pas à résoudre le problème, contactez Dell Computer Corporation pour obtenir une aide technique (consultez la section « Obtenir de l'aide » de la documentation du système).

### Avant d'appeler...

Installez-vous devant votre ordinateur, exécutez le logiciel et munissez-vous de la documentation.

Le technicien peut vous demander les informations suivantes :

- Votre adresse et votre numéro de téléphone
  - Le nom et le numéro du modèle du produit qui fait l'objet de votre appel
  - Le numéro de série et l'étiquette de service du produit
  - Les noms et numéros de version des logiciels utilisés avec le produit
  - Le nom et le numéro de version du système d'exploitation utilisé
  - Le type de votre ordinateur (fabricant et numéro de modèle)
  - Les cartes d'extension ou additionnelles installées
  - La quantité de mémoire sur votre ordinateur
- 

Veuillez lire toutes les [restrictions et clauses exonératoires de responsabilité](#).

---