Windows Management Instrumentation (*WMI*)

Шестаков А.П.

Технология WMI

Технология WMI — это глобальная концепция настройки, управления и слежения за работой различных частей корпоративной компьютерной сети.

Задачи, решаемые с помощью WMI

Используя WMI, можно с помощью специальных утилит или сценариев Windows Script Host (WSH) решать следующие задачи.

- Управление различными версиями операционной системы Windows. С помощью сценариев WMI можно обращаться к системным счетчикам производительности, анализировать журналы событий (Event Logs), работать с файловой системой, установленными принтерами, управлять запущенными процессами и сервисами, просматривать и изменять настройки реестра, создавать и удалять совместно используемые ресурсы и т. д. При этом все операции можно выполнять одинаковым образом как на локальной, так и на удаленной машине.
- Управление ресурсами и службами сети. Сценарии WMI позволяют настраивать сетевые службы (DNS, DHCP и т. п.) и управлять сетевыми устройствами, поддерживающими технологию SNMP (Simple Network Management Protocol).

Задачи, решаемые с помощью WMI

- Мониторинг состояния системы в реальном времени. Можно создавать сценарии-обработчики событий WMI, которые позволяют отслеживать и нужным образом обрабатывать события, связанные с теми или иными изменениями в информационной системе (например, появление определенной записи в журнале событий на локальном или удаленном компьютере, заполнение жесткого диска сервера до определенного предела, изменение определенного ключа в системном реестре и т. п.).
- Управление серверными приложениями Windows. С помощью WMI можно управлять различными приложениями Microsoft: Application Center, *Operations Manager*, Systems Management Server, Internet Information Server, Exchange Server, SQL Server.

Архитектура WMI состоит из трех частей:

- Управляемые объекты/ресурсы (managed resources) — любые логические или физические компоненты информационной системы, доступ к которым может быть получен с помощью WMI. В качестве управляемых ресурсов могут выступать, например, файлы на жестком диске, запущенный экземпляр приложения, системное событие, предоставленный в общее пользование ресурс, сетевой пакет или установленный в компьютере процессор.
- Ядро WMI (WMI infrastructure). Это связующее звено архитектуры WMI, отвечающее за связь управляющих программ с управляемыми объектами. Ядро WMI, в свою очередь, можно разделить на три части: менеджер объектов *CIM* (Common Information Model Object Manager, CIMOM), репозиторий (хранилище классов и объектов) *CIM* и провайдеры WMI. Кроме этого, для доступа к WMI с помощью сценариев необходима специальная библиотека поддержки сценариев WMI (WMI scripting library), которая располагается в файле wbemdisp.dll в каталоге %SystemRoot%\System32*Wbem*.
- Управляющие программы (*management applications*), которые являются потребителями сервисов WMI. В качестве потребителей могут выступать полновесные Win-приложения, Web-приложения, сценарии WSH или другие инструменты администрирования, с помощью которых происходит доступ к управляемым объектам посредством WMI.

Провайдеры WMI

Провайдеры WMI обеспечивают связь между менеджером объектов *CIM* и управляемыми ресурсами: провайдеры предоставляют для CIMOM данные об управляемом объекте, обрабатывают запросы от управляющих программ и генерируют сообщения о наступлении определенных событий.

При этом провайдер WMI общается с управляемым объектом с помощью специфического API этого объекта, а с CIMOM — посредством стандартного интерфейса прикладного программирования WMI (WMI API). Таким образом, провайдеры скрывают детали внутренней реализации управляемых объектов, позволяя CIMOM обращаться к этим объектам единообразно, используя один и тот же WMI API.

Фактически провайдеры WMI являются серверами COM или DCOM, которые представлены динамическими библиотеками (DLL), находящимися чаще всего в каталоге %SystemRoot%\System32*Wbem*. WMI включает в себя множество встроенных (стандартных) провайдеров для операционных систем Windows.xx, которые предназначены для получения данных из известных системных источников таких, как подсистема Win32/64, журналы событий, системный реестр, системные счетчики производительности.

Провайдер	DLL-файл	Описание
Провайдер каталога Active Directory(Active Directory provider)	Dsprov.dll	Позволяет обращаться к объектам Active Directory как к объектам WMI
Провайдер журнала событий (Event Log provider)	Ntevt.dll	Обеспечивает управление журналом событий (выборка по определенному критерию записей для чтения, создание резервных копий и очистка *журнала, изменение* настроек и т. д.). Также этот провайдер позволяет обрабатывать события, генерируемые журналом (например, добавление в журнал записи определенного типа)
Провайдер системных счетчиков производительности (Perfomance Counter provider)	Wbemperf.dll	Обеспечивает доступ к счетчикам производительности, т. е. к данным, позволяющим численно оценивать производительность системы
Провайдер реестра (Registry provider)	Stdprov.dll	Позволяет читать данные из реестра, создавать и модифицировать там ключи и разделы. Кроме этого, провайдер обеспечивает генерацию события WMI при изменении определенного ключа или ветви реестра

Менеджер объектов CIM

- Регистрация провайдеров. Все провайдеры WMI должны быть зарегистрированы с помощью CIMOM; информация о провайдере (например, тип этого провайдера или путь к библиотеке DLL, которой он представлен) хранится в репозитории *CIM*.
- Переадресация запросов. Используя информацию о зарегистрированных провайдерах, CIMOM перенаправляет полученный от управляющего приложения запрос к нужному провайдеру.
- Доступ к удаленной машине с WMI. Управляющее приложение может обратиться с запросом к любой удаленной машине, на которой установлен WMI. При этом происходит соединение с CIMOM на удаленной машине, после чего все запросы здесь должны обрабатываться точно так же, как и на локальной машине.
- Обеспечение безопасности. Защита ресурсов WMI состоит в том, что CIMOM проверяет права пользователя, который пытается воспользоваться сервисами WMI на локальном или удаленном компьютере.

Менеджер объектов CIM

- Обработка запросов управляющих приложений. Потребители WMI обращаются к управляемым объектам с помощью специального языка запросов WMI Query Language (WQL). Если провайдер запрашиваемого объекта не поддерживает напрямую WQL, то CIMOM должен преобразовать этот запрос к тому виду, в котором он сможет быть обработан этим провайдером.
- Обработка событий WMI. Поддержка CIMOM этой функции позволяет потребителям WMI создавать обработчики событий, которые возникают при определенном изменении в управляемом объекте (примеры таких событий — снижение объема свободного пространства на жестком диске до заданного значения или запуск на компьютере определенного приложения). Для этого CIMOM периодически опрашивает нужный объект (интервал опроса задается в управляющем приложении) и генерирует событие как только обнаруживает, что заданное заранее условие возникновения события выполнено.

Репозиторий CIM

Основной идеей, на которой базируется WMI, является возможность представить информацию о состоянии любого управляемого объекта в виде стандартной схемы. В качестве такой схемы выступает информационная модель *CIM*, которая является репозиторием (хранилищем) объектов и классов, моделирующих различные компоненты компьютерной системы.

CIM можно считать хранилищем классов, где класс — это модель (шаблон) управляемого объекта (в качестве управляемых объектов могут выступать самые различные логические и физические компоненты компьютерной системы: жесткие диски, журналы событий, сетевые карты, файлы и папки, процессы, сервисы, процессоры и т. д.). С этой точки зрения *CIM* похожа на другие каталоги, которые используются в Windows (например, каталог файловой системы содержит объекты-файлы и объекты-папки, а каталог Active Directory — объекты-домены, объекты-пользователи, объекты-принтеры и т. д.)

Однако важной особенностью *CIM* является то, что хранящиеся в ней классы чаще всего соответствуют динамически изменяемым ресурсам, поэтому объекты-экземпляры таких классов не хранятся постоянно в *CIM*, а создаются провайдером по запросу потребителя WMI. Связано это с тем, что состояние большинства WMI-совместимых устройств меняется очень быстро и постоянное обновление информации в *CIM* может значительно снизить общую производительность системы.

Пространства имен

Классы, составляющие *CIM*, имеют свойства и методы и находятся в иерархической зависимости друг от друга — классы-потомки могут наследовать или переопределять свойства родительских классов, а также добавлять собственные свойства. Свойства описывают конфигурацию и текущее состояние управляемого ресурса, а методы позволяют выполнить над этим ресурсом определенные действия.

Классы *CIM* группируются в пространстве имен (namespaces), которые упорядочены иерархически (корневое пространство имен обозначается через Root). Пространство имен — это группа логически связанных друг с другом классов, которые относятся к какой-либо определенной технологии или области управления. Например, одно из наиболее часто используемых на практике пространств имен CIMV2 содержит классы, которые описывают компьютер и операционную систему.

Путь к классам и объектам CIM

Полный путь к хранящемуся в *CIM* классу или объекту-экземпляру класса (управляемому устройству) имеет следующую структуру:

[\\ComputerName][\Namespace][:ClassName][.KeyProperty1=Value1 [,KeyProperty2=Value2…]]

Здесь *ComputerName* — это сетевое имя компьютера, на котором расположен нужный класс или объект (для задания имени локального компьютера можно использовать символ "."), \Namespace — название пространства имен, в котором находится этот класс или объект, :ClassName — имя класса. Параметры KeyProperty1 и Value1, KeyProperty2 и Value2, …, задают список ключевых пар (свойство-значение) объекта.

Например, следующий путь

\\.\CIMV2:Win32_Process.Name="Notepad.exe"

определяет процесс (экземпляр класса Win32_Process из пространства имен CIMV2) с именем "Notepad.exe", который запущен на локальной машине.

Безопасность при работе с WMI

Технология WMI позволяет с помощью специальных утилит или сценариев производить различные потенциально опасные действия (например, остановку служб или перезагрузку компьютера). Причем на удаленной машине это выполнить так же просто, как и на локальной — достаточно написать имя нужной машины в пути к объекту WMI. Поэтому вопросы безопасности при работе с WMI имеют очень большое значение.

Безопасность основана на именах пользователей и их паролях. Когда заводится пользователь, то его учетной записи присваивается уникальный идентификатор безопасности (*Security IDentifier*, SID). На основе SID для пользователя формируется маркер доступа (*Access Token*), в который также добавляется список групп, членом которых является пользователь, и список привилегий, которыми он обладает (например, остановка служб или выключение компьютера). Этот маркер доступа присваивается и всем процессам, которые запускает пользователь. Далее каждый объект операционной системы, доступ к которому определяет система безопасности (это может быть файл, процесс, служба и т. д.) имеет дескриптор безопасности (*Security Descriptor*, SD), в котором хранится таблица контроля доступа (Access Control List, ACL) для этого объекта. При обращении пользователя или процесса, запущенного пользователем, к объекту происходит сравнение маркера доступа этого пользователя с таблицей контроля доступа и в зависимости от результатов выдается или отклоняется разрешение на выполнение запрашиваемых действий над объектом.

Основные типы классов CIM

В *CIM* существует три основных типа классов, различающихся между собой по способу хранения информации об управляемых ресурсах.

- Абстрактный класс (abstract class) — это шаблон, который служит исключительно для образования новых классов-потомков (абстрактных и неабстрактных). Абстрактный класс не может непосредственно использоваться для получения экземпляра управляемого ресурса.
- Статический класс (static class) определяет данные, которые физически хранятся в репозитории *CIM* (к такому типу относятся, например, данные о собственных настройках WMI). Вследствие этого для доступа к экземплярам статических классов не нужно прибегать к помощи каких-либо провайдеров.
- Динамический класс (dynamic class) моделирует управляемый ресурс, данные о котором соответствующий провайдер возвращает в динамическом режиме.

Деление классов по информационным моделям

- Системные классы. Системными называются те классы, которые служат для задания конфигурации и выполнения внутренних функций WMI (определение пространств имен, обеспечение безопасности при работе с пространствами имен, регистрация провайдеров, подписка на события WMI и формирование сообщений о наступлении таких событий). Системные классы могут быть абстрактными или статическими. Имена всех системных классов начинаются с символов "__" (двойное подчеркивание), например, __SystemClass, __NAMESPACE,__Provider или __Win32Provider.

- Классы модели ядра (основной модели) (core model). К этой модели относятся абстрактные классы, которые обеспечивают интерфейс со всеми областями управления. Названия таких классов начинаются с префикса "CIM_". Примерами классов модели ядра могут служить класс CIM_ManagedSystemElement (свойства этого класса идентифицируют управляемые компоненты системы) и его наследники CIM_LogicalElement (описание логического управляемого ресурса, например, файла или каталога) и CIM_PhysicalElement (описание физического управляемого ресурса, например, периферийного устройства).

Деление классов по информационным моделям

- Классы общей модели (common model). Общая модель является расширением основной модели — здесь представлены классы, которые являются специфическими для задач управления, но не зависят от конкретной технологии или реализации (другими словами, не зависят от типа операционной системы). Названия таких классов, как и классов модели ядра, начинаются с "CIM_". Класс CIM_LogicalFile (наследник класса CIM_LogicalElement), описывающий файл, является примером класса общей модели, т. к. файловая система присутствует практически в любой операционной системе.

- Классы модели расширения (extension model). Эта категория классов включает в себя специфические для каждой технологии или реализации дополнения к общей модели. В WMI определено большое количество классов, которые соответствуют ресурсам, специфическим для среды Win32 (имена этих классов начинаются с префикса "Win32_"). Например, классы Win32_PageFile и Win32_ShortCutFile, которые описывают соответственно файлы подкачки Windows и файлы-ярлыки, являются потомками класса CIM_LogicalFile из общей модели.

Методы классов WMI

Методы класса позволяют выполнять те или иные действия над управляемым ресурсом, которому соответствует этот класс (так как не над каждым ресурсом можно производить какие-либо операции, то не у всякого класса есть методы).

Тестер WMI

Тестер WMI (wbemtest.exe) — это графическая утилита, с помощью которой можно взаимодействовать с инфраструктурой WMI на локальном или удаленном компьютере. С помощью тестера WMI можно решать следующие задачи:

- подсоединяться к определенному пространству имен *CIM*;
- создавать и удалять классы и экземпляры классов;
- получать список имеющихся классов и экземпляров классов *CIM*;
- просматривать и изменять свойства и квалификаторы классов или экземпляров классов;
- выполнять методы классов и экземпляров классов;
- составлять и выполнять запросы на языке WQL;
- выводить код MOF для классов и экземпляров управляемых ресурсов.

Исполняемый файл wbemtest.exe является стандартным компонентом WMI в любой операционной системе; устанавливается он в каталог %SystemRoot%\System32\Wbem. После запуска этого файла появляется диалоговое окно Тестер инструментария управления Windows (Windows Management Instrumentation Tester), с помощью которого можно получить доступ ко всем функциям тестера WMI

Административные утилиты WMI

- WMI *CIM* Studio. Это наиболее универсальное приложение, которое может быть использовано для просмотра и редактирования в репозитории *CIM* классов и их экземпляров. С помощью WMI *CIM* Studio можно также выполнять методы классов и объектов, просматривать ассоциации между различными классами, выполнять запросы на языке WQL, генерировать и компилировать файлы MOF для классов и объектов. Утилита WMI *CIM* Studio обладает практически теми же возможностями, что и тестер WMI (wbemtest.exe), однако имеет гораздо более удобный *интуитивный интерфейс*. Как и работа с тестером WMI, использование WMI *CIM* Studio предполагает довольно хорошее знание структуры репозитория *CIM* и названий нужных классов.

- WMI Object Browser. Эта утилита предназначена для просмотра и редактирования объектов (экземпляров классов) в репозитории *CIM*, а также для вызовов их методов. Особенностью WMI Object Browser является то, что информация об объектах представлена в виде иерархического дерева, где в качестве корневого объекта может использоваться произвольный экземпляр выбранного нами класса. Само дерево объектов строится с помощью ассоциированных классов, что помогает извлекать информацию об управляемых ресурсах, не обладая глубокими знаниями о структуре репозитория *CIM* и используемых классах.

Административные утилиты WMI

- WMI Event Registration Tool. Данная утилита предоставляет графический интерфейс для регистрации и конфигурирования постоянных потребителей событий WMI. Здесь можно создавать или изменять фильтры событий, определять постоянных потребителей и устанавливать связи между ними и фильтрами событий.
- WMI Event Viewer. Это вспомогательное приложение является постоянным потребителем событий, позволяющим сортировать и просматривать подробную информацию о полученных событиях.