US 20030105732A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2003/0105732 A1

(43) Pub. Date: Jun. 5, 2003

Kagalwala et al.

(54) DATABASE SCHEMA FOR STRUCTURE QUERY LANGUAGE (SQL) SERVER

(76) Inventors: **Raxit A. Kagalwala**, Issaquah, WA (US); **John Patrick Thompson**, Seattle, WA (US)

Correspondence Address:
LEE & HAYES PLLC
421 W RIVERSIDE AVENUE SUITE 500
SPOKANE, WA 99201

(21) Appl. No.: 09/789,804

(22) Filed: Feb. 20, 2001

Related U.S. Application Data

(60) Provisional application No. 60/249,528, filed on Nov. 17, 2000.

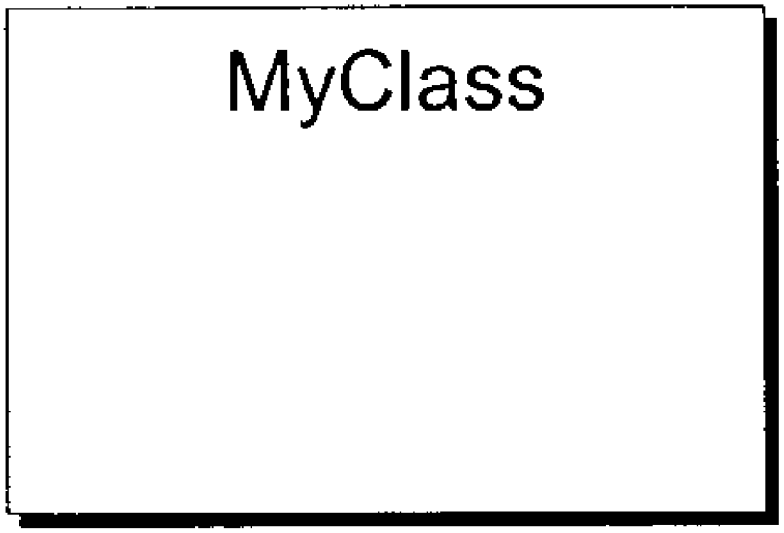
Publication Classification

(51) Int. Cl.[7] G06F 7/00

(52) U.S. Cl. 707/1

(57) ABSTRACT

A schema for a SQL (structured query language) database defines classes, properties, methods, and associations.



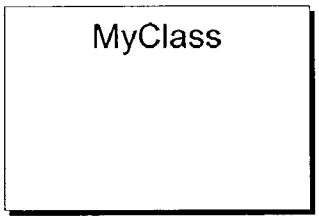


Fig. 1

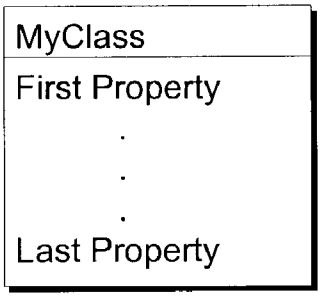


Fig. 2

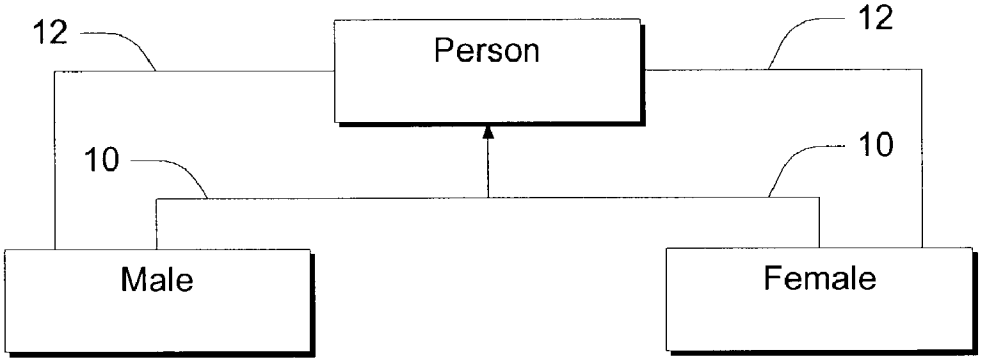


Fig. 3

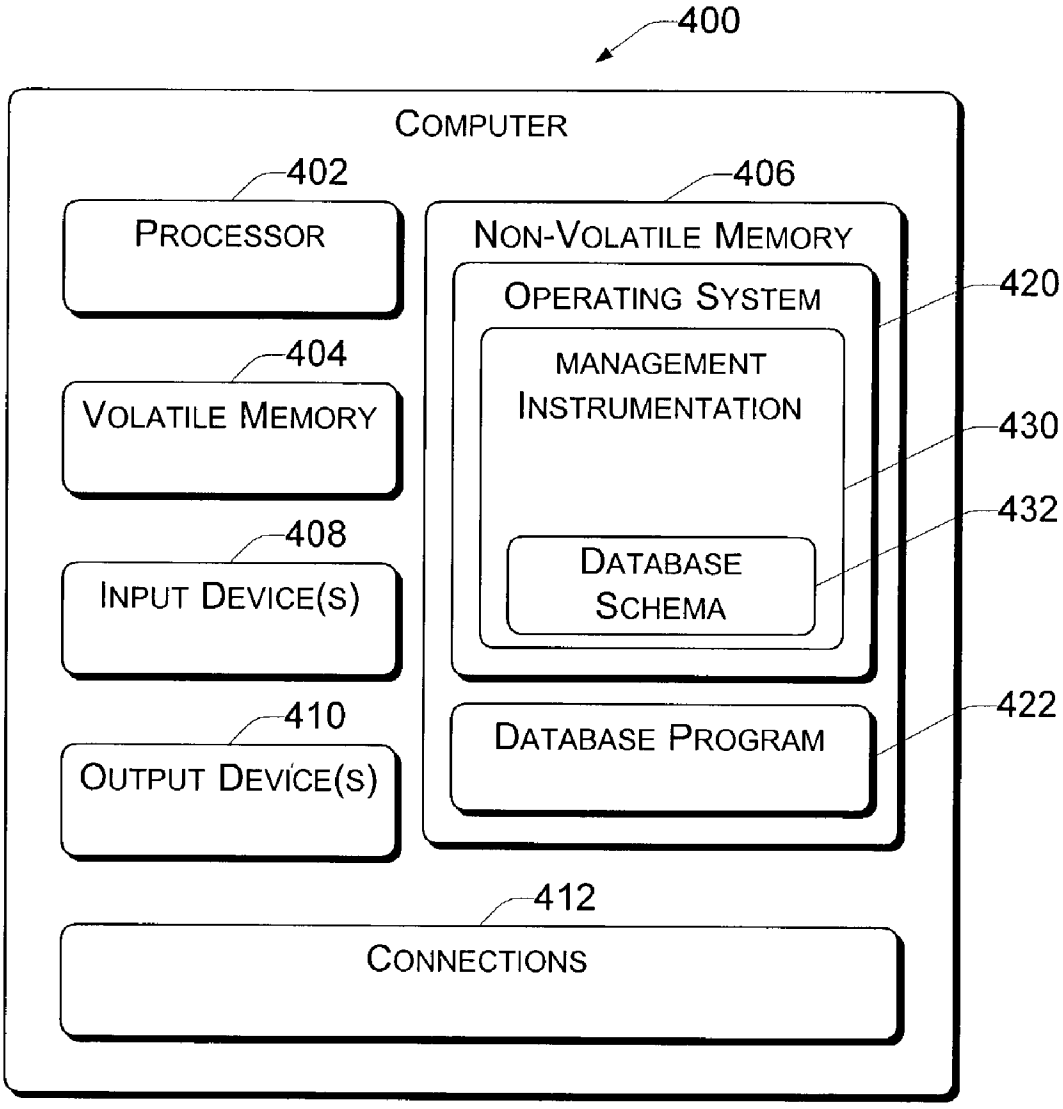

400
COMPUTER
402
PROCESSOR
404
VOLATILE MEMORY
408
INPUT DEVICE(S)
410
OUTPUT DEVICE(S)
406
NON-VOLATILE MEMORY
OPERATING SYSTEM
MANAGEMENT INSTRUMENTATION
DATABASE SCHEMA
DATABASE PROGRAM
420
430
432
422
412
CONNECTIONS

Fig. 4

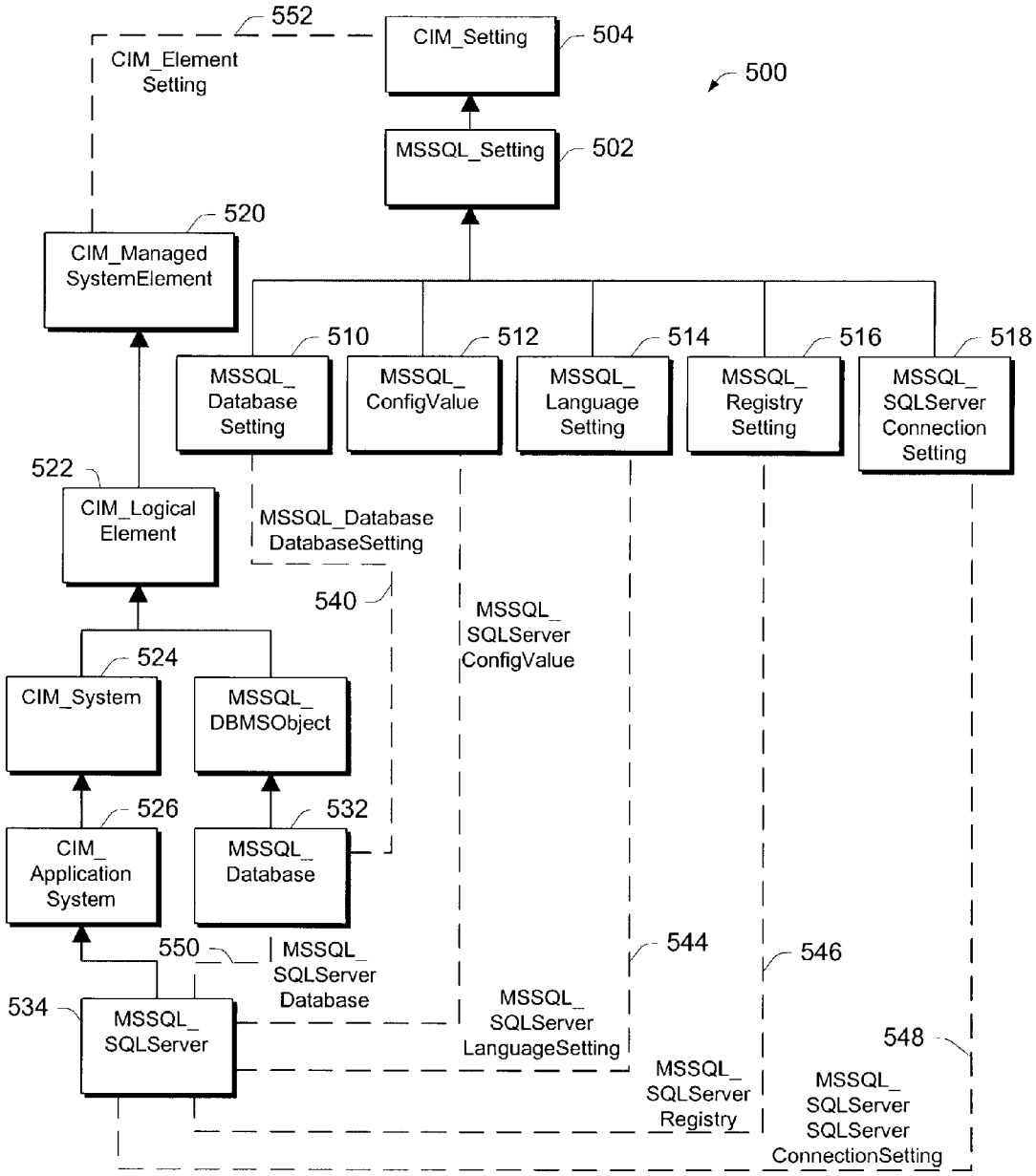

552
CIM_Element Setting
CIM_Setting
504
500
MSSQL_Setting
502
520
CIM_Managed SystemElement
510
512
514
516
518
MSSQL_ Database Setting
MSSQL_ ConfigValue
MSSQL_ Language Setting
MSSQL_ Registry Setting
MSSQL_ SQLServer Connection Setting
522
CIM_Logical Element
MSSQL_Database DatabaseSetting
540
MSSQL_ SQLServer ConfigValue
524
CIM_System
MSSQL_ DBMSObject
526
532
CIM_ Application System
MSSQL_ Database
550
MSSQL_ SQLServer Database
544
546
534
MSSQL_ SQLServer
MSSQL_ SQLServer LanguageSetting
548
MSSQL_ SQLServer Registry
MSSQL_ SQLServer SQLServer ConnectionSetting

Fig. 5

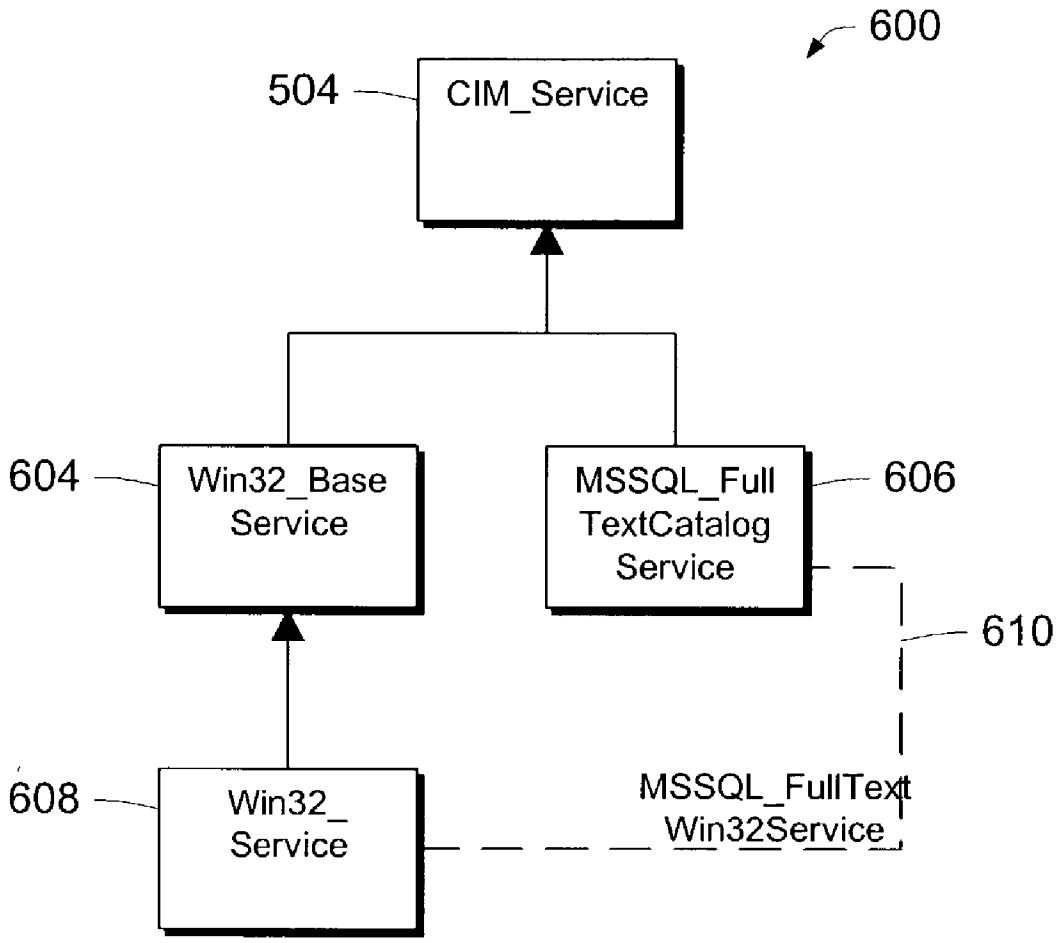

600
504
CIM_Service
604
Win32_Base Service
MSSQL_Full TextCatalog Service
606
610
608
Win32_ Service
MSSQL_FullText Win32Service

Fig. 6

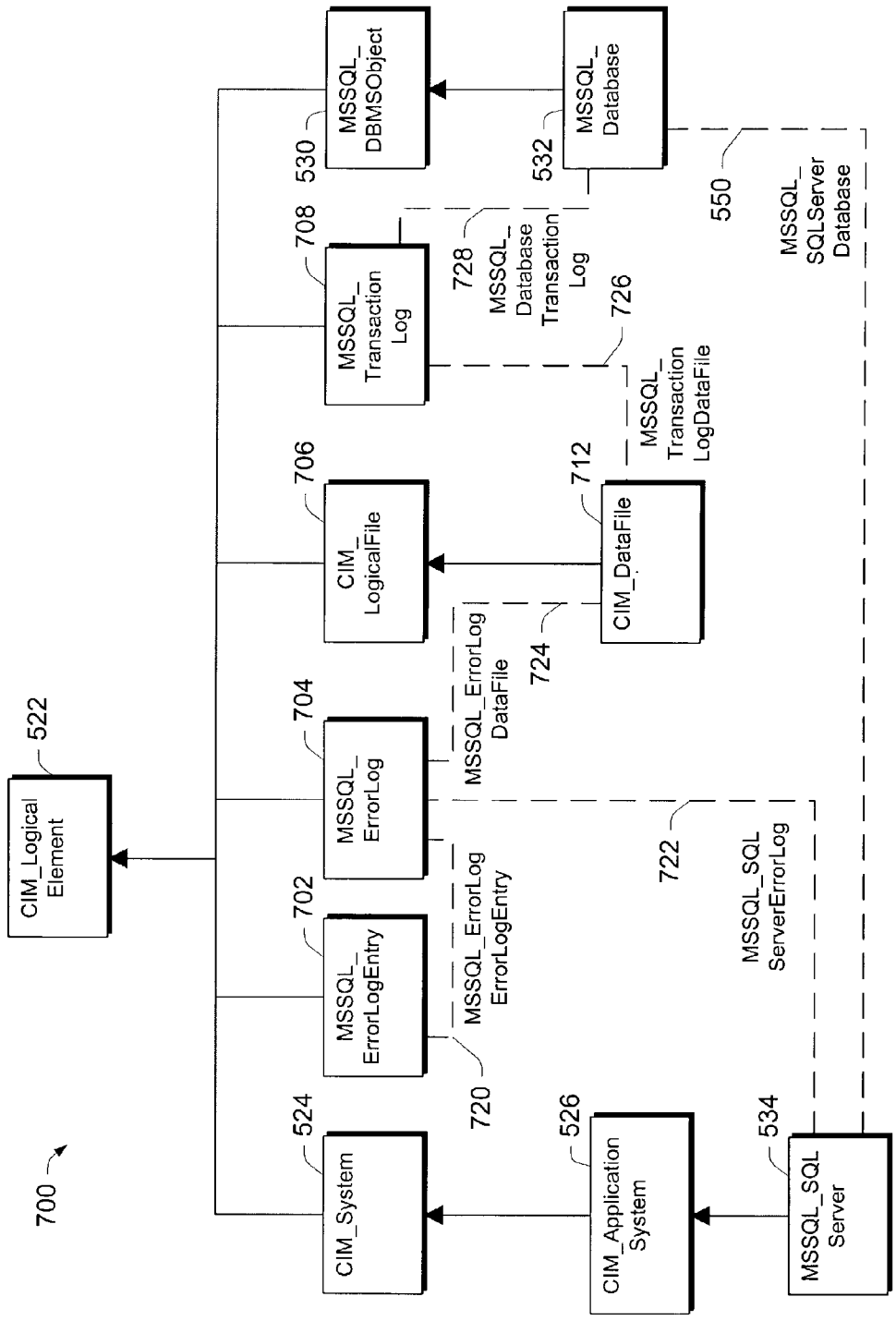

700
CIM_Logical Element
522
CIM_System
524
CIM_Application System
526
MSSQL_SQL Server
534
MSSQL_ErrorLogEntry
702
MSSQL_ErrorLog
704
CIM_LogicalFile
706
MSSQL_Transaction Log
708
MSSQL_DBMSObject
530
MSSQL_Database
532
CIM_DataFile
712
MSSQL_ErrorLog ErrorLogEntry
720
MSSQL_ErrorLog DataFile
724
MSSQL_SQL ServerErrorLog
722
MSSQL_Database Transaction Log
728
MSSQL_Transaction LogDataFile
726
MSSQL_SQLServer Database
550

Fig. 7

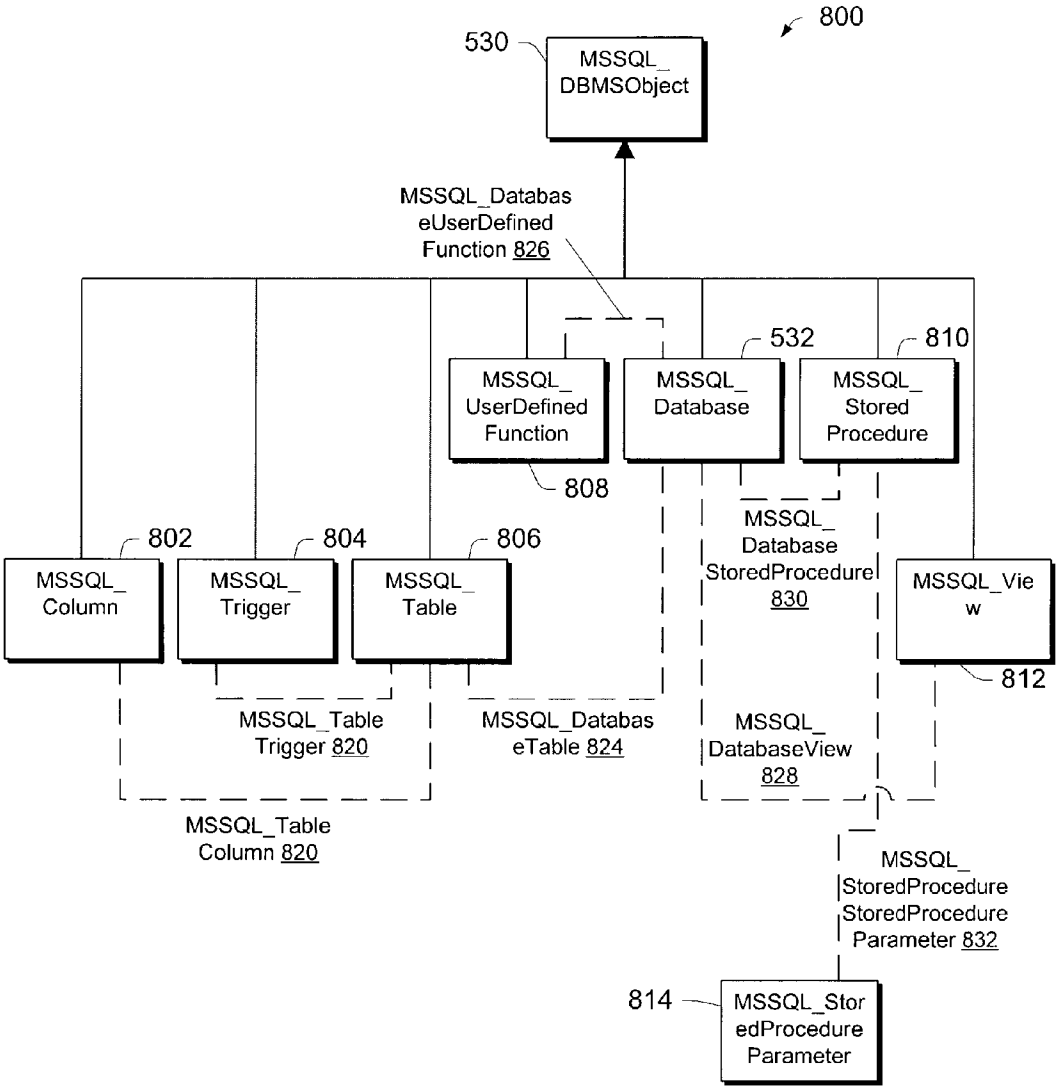

800
530
MSSQL_ DBMSObject
MSSQL_Databas eUserDefined Function 826
532
810
MSSQL_ UserDefined Function
MSSQL_ Database
MSSQL_ Stored Procedure
808
MSSQL_ Database StoredProcedure 830
802
804
806
MSSQL_ Column
MSSQL_ Trigger
MSSQL_ Table
MSSQL_Vie w
812
MSSQL_Table Trigger 820
MSSQL_Databas eTable 824
MSSQL_ DatabaseView 828
MSSQL_Table Column 820
MSSQL_ StoredProcedure StoredProcedure Parameter 832
814
MSSQL_Stor edProcedure Parameter

Fig. 8

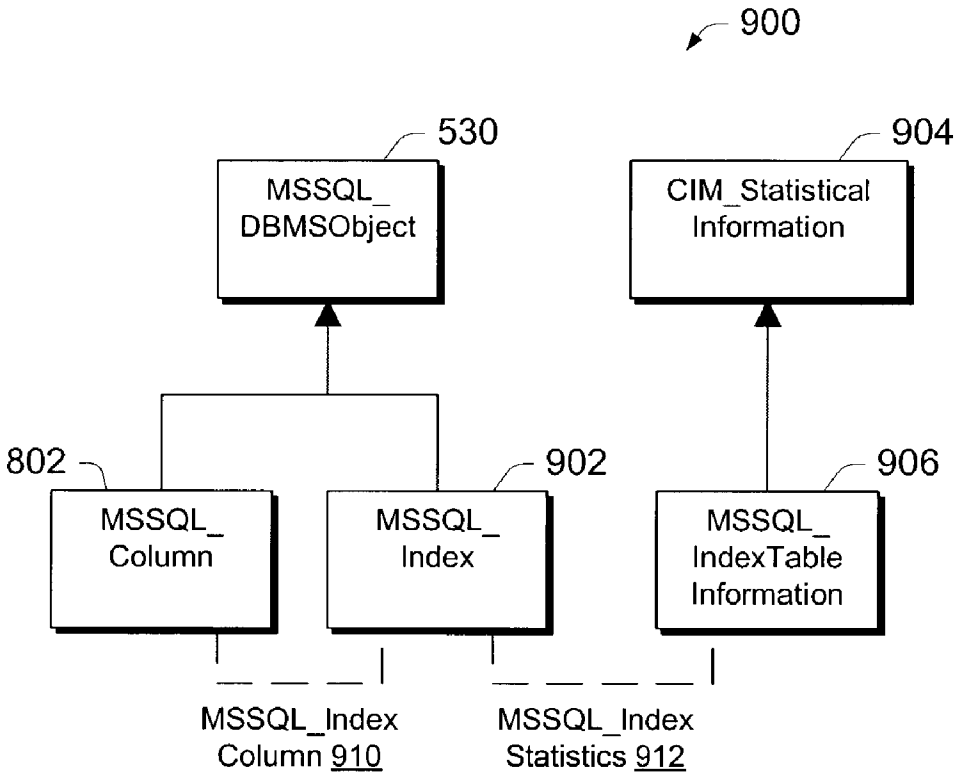

900
530
MSSQL_
DBMSObject
904
CIM_Statistical
Information
802
MSSQL_
Column
902
MSSQL_
Index
906
MSSQL_
IndexTable
Information
MSSQL_Index
Column 910
MSSQL_Index
Statistics 912

Fig. 9

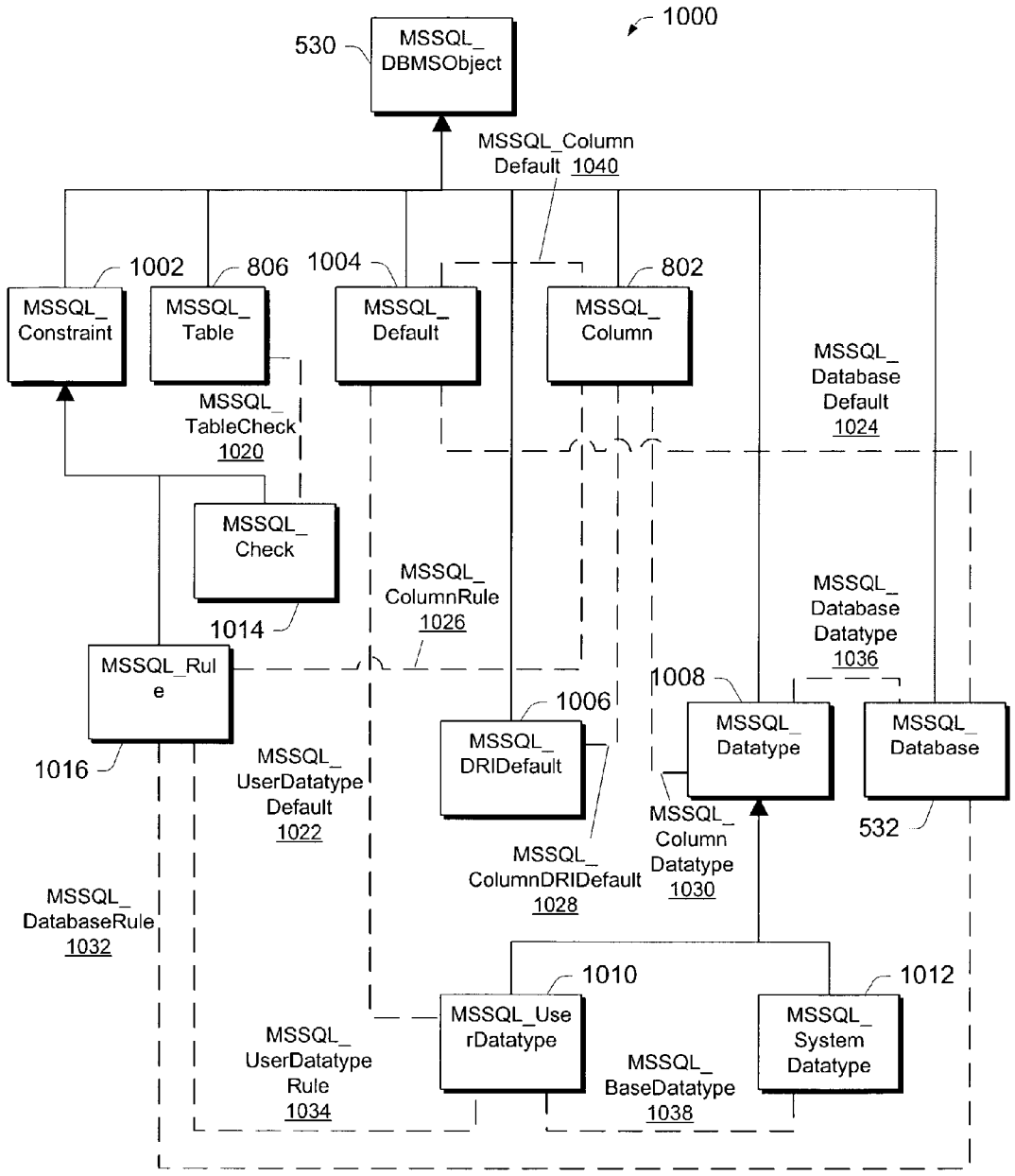

1000
530
MSSQL_ DBMSObject
MSSQL_Column Default 1040
1002
806
1004
802
MSSQL_ Constraint
MSSQL_ Table
MSSQL_ Default
MSSQL_ Column
MSSQL_ Database Default 1024
MSSQL_ TableCheck 1020
MSSQL_ Check
1014
MSSQL_ ColumnRule 1026
MSSQL_ Database Datatype 1036
MSSQL_Rul e
1008
1006
MSSQL_ Datatype
MSSQL_ Database
1016
MSSQL_ UserDatatype Default 1022
MSSQL_ DRIDefault
MSSQL_ Column Datatype 1030
532
MSSQL_ ColumnDRIDefault 1028
MSSQL_ DatabaseRule 1032
1010
1012
MSSQL_Use rDatatype
MSSQL_ System Datatype
MSSQL_ UserDatatype Rule 1034
MSSQL_ BaseDatatype 1038

Fig. 10

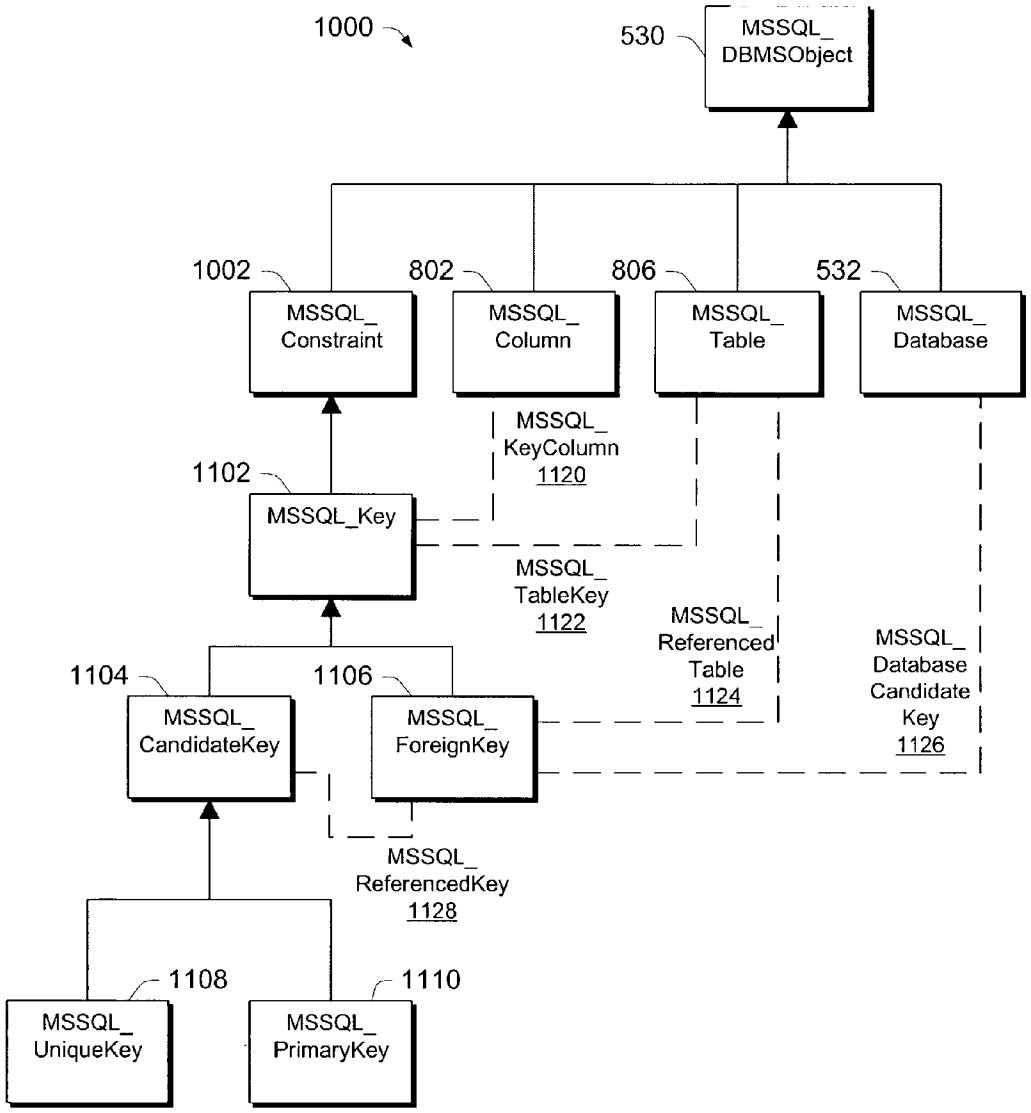

1000
530
MSSQL_ DBMSObject
1002
802
806
532
MSSQL_ Constraint
MSSQL_ Column
MSSQL_ Table
MSSQL_ Database
MSSQL_ KeyColumn 1120
1102
MSSQL_Key
MSSQL_ TableKey 1122
MSSQL_ Referenced Table 1124
MSSQL_ Database Candidate Key 1126
1104
1106
MSSQL_ CandidateKey
MSSQL_ ForeignKey
MSSQL_ ReferencedKey 1128
1108
1110
MSSQL_ UniqueKey
MSSQL_ PrimaryKey

Fig. 11

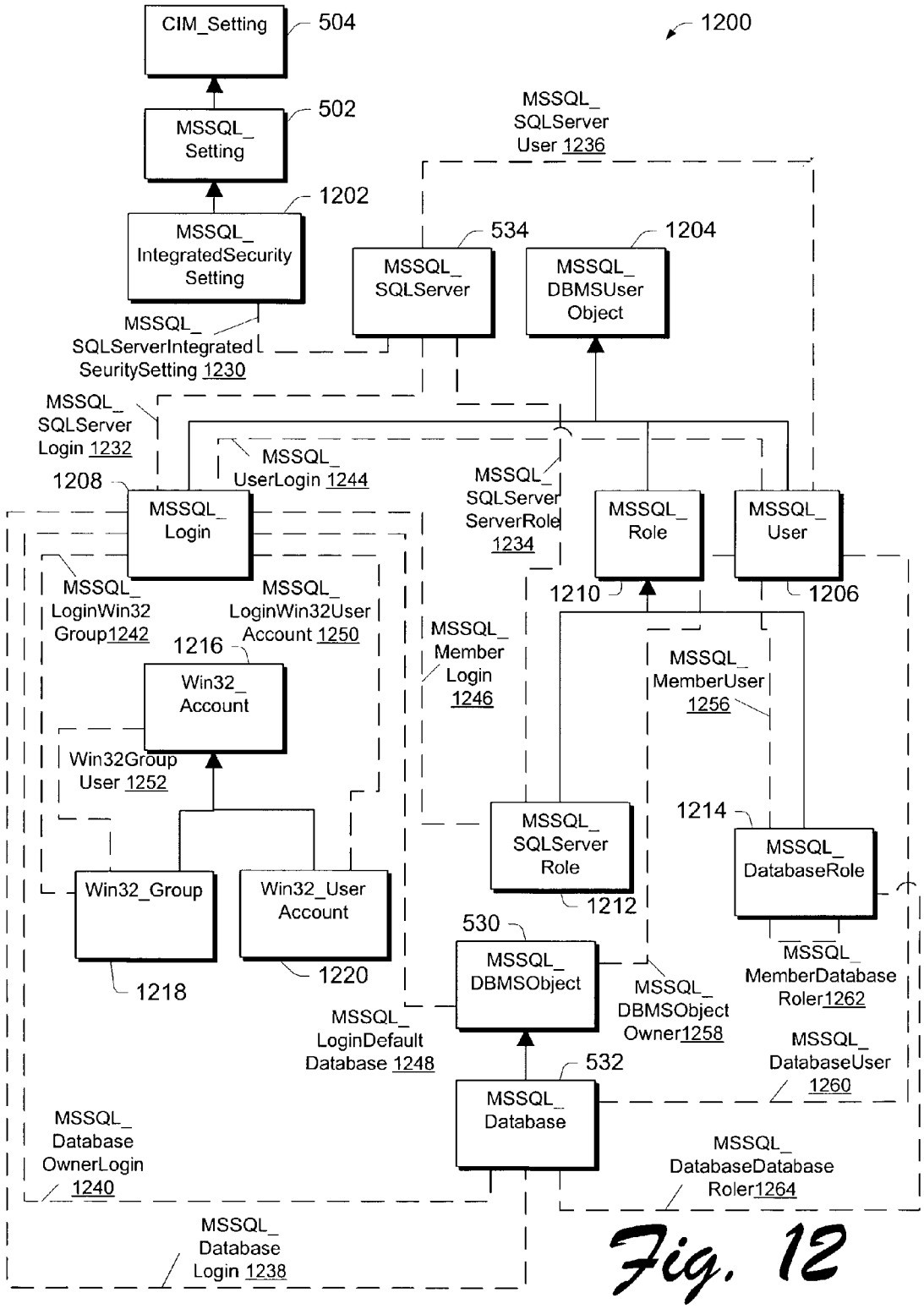

CIM_Setting
504
1200
MSSQL_ Setting
502
MSSQL_ SQLServer User 1236
1202
MSSQL_ IntegratedSecurity Setting
534
1204
MSSQL_ SQLServer
MSSQL_ DBMSUser Object
MSSQL_ SQLServerIntegrated SeuritySetting 1230
MSSQL_ SQLServer Login 1232
MSSQL_ UserLogin 1244
1208
MSSQL_ Login
MSSQL_ SQLServer ServerRole 1234
MSSQL_ Role
MSSQL_ User
1210
1206
MSSQL_ LoginWin32 Group1242
MSSQL_ LoginWin32User Account 1250
1216
Win32_ Account
MSSQL_ Member Login 1246
MSSQL_ MemberUser 1256
Win32Group User 1252
1214
MSSQL_ SQLServer Role
MSSQL_ DatabaseRole
1212
Win32_Group
Win32_User Account
530
MSSQL_ DBMSObject
MSSQL_ MemberDatabase Roler1262
1218
1220
MSSQL_ DBMSObject Owner1258
MSSQL_ LoginDefault Database 1248
532
MSSQL_ DatabaseUser 1260
MSSQL_ Database
MSSQL_ Database OwnerLogin 1240
MSSQL_ DatabaseDatabase Roler1264
MSSQL_ Database Login 1238

Fig. 12

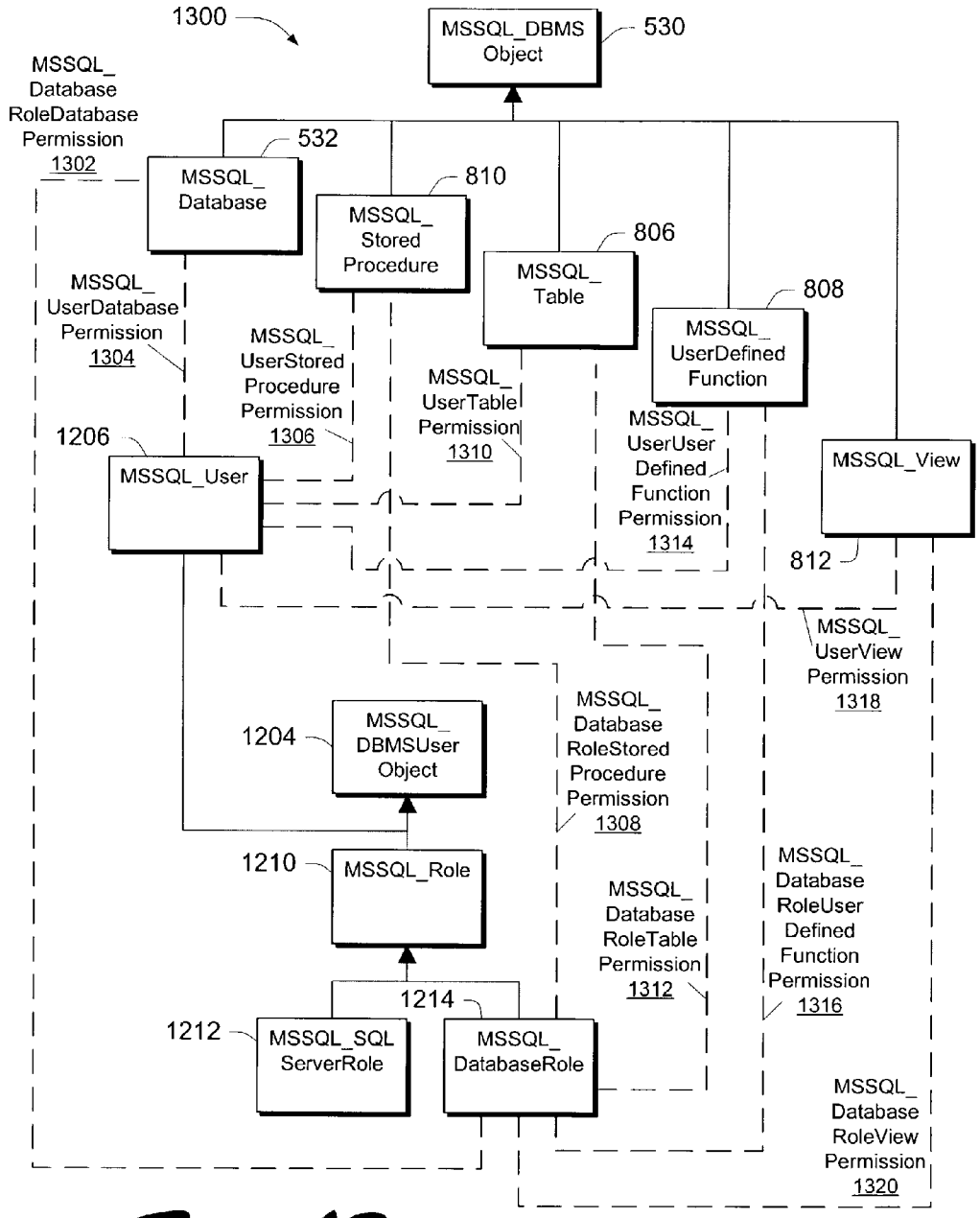

1300
MSSQL_DBMS Object
530
MSSQL_ Database RoleDatabase Permission 1302
532
MSSQL_ Database
810
MSSQL_ Stored Procedure
806
MSSQL_ Table
808
MSSQL_ UserDefined Function
MSSQL_ UserDatabase Permission 1304
MSSQL_ UserStored Procedure Permission 1306
MSSQL_ UserTable Permission 1310
MSSQL_ UserUser Defined Function Permission 1314
1206
MSSQL_User
MSSQL_View
812
MSSQL_ UserView Permission 1318
1204
MSSQL_ DBMSUser Object
MSSQL_ Database RoleStored Procedure Permission 1308
1210
MSSQL_Role
MSSQL_ Database RoleTable Permission 1312
MSSQL_ Database RoleUser Defined Function Permission 1316
1214
1212
MSSQL_SQL ServerRole
MSSQL_ DatabaseRole
MSSQL_ Database RoleView Permission 1320

Fig. 13

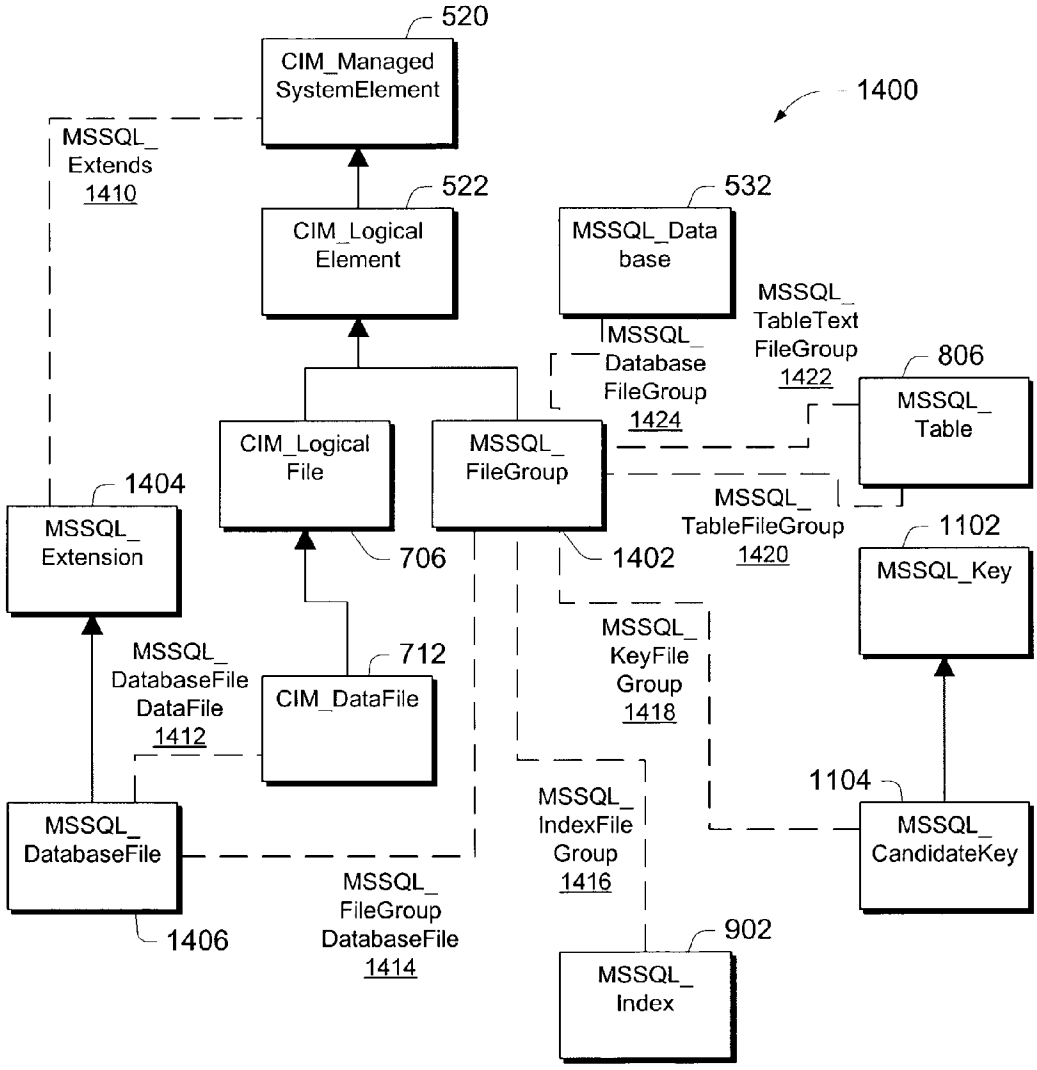

520
CIM_Managed SystemElement
1400
MSSQL_ Extends 1410
522
CIM_Logical Element
532
MSSQL_Data base
MSSQL_ TableText FileGroup 1422
MSSQL_ Database FileGroup 1424
806
MSSQL_ Table
CIM_Logical File
MSSQL_ FileGroup
1404
MSSQL_ Extension
706
1402
MSSQL_ TableFileGroup 1420
1102
MSSQL_Key
MSSQL_ KeyFile Group 1418
712
MSSQL_ DatabaseFile DataFile 1412
CIM_DataFile
1104
MSSQL_ IndexFile Group 1416
MSSQL_ DatabaseFile
MSSQL_ CandidateKey
1406
MSSQL_ FileGroup DatabaseFile 1414
902
MSSQL_ Index

Fig. 14

DATABASE SCHEMA FOR STRUCTURE QUERY LANGUAGE (SQL) SERVER

RELATED APPLICATIONS

[0001] This application claims priority to U.S. Provisional Application No. 60/249,528, filed Nov. 17, 2000, entitled "Database Schema for Structured Query Language (SQL) Server."

TECHNICAL FIELD

[0002] This invention relates to databases and to database management schemas.

BACKGROUND

[0003] Database management systems (DBMS) are core components of virtually every enterprise (e-business) application. The ability to effectively configure, monitor, and manage a DBMS is critical to the success of enterprise applications.

[0004] Most DBMSs are designed for compatibility with relational databases. A relational database comprises a plurality of tables. Each table has a plurality of data records (rows) and each table includes a definition of the fields (columns) that the records will contain. A relational database includes the specification of relationships between fields of different tables. A DBMS performs common management tasks such as creating databases, adding tables, replication management, data backup, etc.

[0005] The Desktop Management Task Force (DMTF) Common Information Model (CIM) is an approach to the management of systems, software, users, and networks that applies the basic structuring and conceptualization techniques of the object-oriented paradigm. More specifically, the purpose of CIM is to model various computer-related systems-both hardware and software. It is important to recognize that object-oriented modeling is different from object-oriented programming.

[0006] This type of modeling uses schemas to represent systems. A schema is an abstraction of something that exists in the real world. Generally, a schema comprises a collection of classes and associations.

[0007] A class models a set of objects that have similar properties and fulfill similar purposes. In a database management schema, for example, individual classes might define such things as files, users, tables, etc.

[0008] Classes follow a hierarchical structure. Classes can have subclasses, also referred to as specialization classes. The parent class of a subclass is referred to as a superclass or a generalization class. A class that does not have a superclass is referred to as a base class.

[0009] A typical schema might comprise a collection of different schemas, which in this case can also be referred to as subschemas. Such subschemas are often located in various different namespaces. A namespace is simply a way to logically group related data. Within a given namespace, all names are unique. Within the following disclosure, the terms "schema" and subschema are used interchangeably.

[0010] A subclass inherits properties of its superclass. All properties and methods of a superclass apply to the subclass.

[0011] It is conventional to represent a class by a rectangle containing the name of the class. **FIG. 1** shows an example. A class with properties is represented by a rectangle divided into two regions as in **FIG. 2**, one containing the name of the class and the other a list of properties. Inheritance, or a subclass/superclass relationship, is represented by a line drawn between the subclass and the superclass, with an arrow adjacent to the superclass indicating the superclass. Lines representing inheritance are shown in **FIG. 3**, indicated by reference numeral **10**.

[0012] Classes contain instances that are collections of values that conform to the type established by the class. Instances are identified by keys that are unique within the class. In other words, no two instances in the same class in the same namespace may have the same values for all of their key values. The term "object" may be used to refer to either an instance or a class.

[0013] An association represents a relationship between two or more objects. More specifically, an association is a mechanism for providing an explicit mapping between classes. Associations can be within a namespace or across namespaces. Associations are conventionally shown as a line between two classes, as indicated by reference number **12** in **FIG. 3**.

[0014] CIM schemas describe the gamut of managed elements: servers and desktops (operating systems, components, peripherals, and applications, all layers of the network (from Ethernet switches to IP and HTTP connections), and even end-users. Schema properties model the attributes that apply to objects, such as the type of printer or storage medium, RAM and CPU capacity, storage capacity, etc.

[0015] The discussion above gives a general overview of object-oriented modeling and CIM. Please refer to Winston Vumpus, John W. Sweitzer, Patrick Thompson, Andrea R. Westerinin, and Raymond C. Williams; *Common Information Model,* John Wiley & Sons, Inc., New York (2000) for further information regarding CIM. Also refer to Common Information Model (CIM) Specification, V2.0, Mar. 3, 1998, available from the Distributed Management Taskforce. DMTF has a number of other resources on its Internet web site.

SUMMARY

[0016] A database schema described herein is an extension of the CIM core model. It defines classes, properties, methods, and associations for a SQL (structured query language) database. Although a specific embodiment is disclosed herein, it should be recognized that variations of the described embodiment are possible while still remaining within the scope of the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] FIGS. **1-3** illustrate CIM drawing conventions.

[0018] **FIG. 4** is a block diagram showing components of a typical computer.

[0019] FIGS. **5-14** illustrates a database schema for a SQL database.

DETAILED DESCRIPTION

[0020] This disclosure addresses a database schema for a SQL (structure query language) database, such as

Microsoft's SQL Server database system. Prior to describing the database schema, however, an exemplary computing environment is described to provide a context for implementing the schema.

[0021] Exemplary Computing Environment

[0022] **FIG. 4** shows an exemplary computer system **400** that implements the database schema. The computer **400** is representative of many different configurations, including personal computers, server computers, hand-held or laptop devices, multiprocessor systems, micro-processor systems, game consoles, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like. Computer **400** typically includes a variety of computer-readable media. Computer-readable media can be any available media that can be accessed by computer **400** and includes both volatile and nonvolatile media, removable and non-removable media. By way of example, and not limitation, computer-readable media may comprise computer storage media and communication media.

[0023] Communication media typically embodies computer-readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. By way of example, communication media includes wired media such as a wired network or direct-wired connection and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of any of the above should also be included within the scope of computer readable media.

[0024] In the **FIG. 4** illustration, the computer **400** has a processor unit **402** with one or more processors, volatile memory **404** (e.g., RAM), and non-volatile memory **406** (e.g., ROM, Flash, hard disk, optical, RAID memory, etc.). The computer **400** also includes one or more input devices **408** (e.g., keyboard, mouse, stylus, touch screen, microphone, etc.) and one or more output devices **412** (e.g., display, speakers, printer, etc.) A set of connections **412** may also be provided to facilitate wireless or wire-based communication with other computers, peripherals, and the like.

[0025] The computer **400** runs an operating system **420**. The "Windows" brand of operating systems, available from Microsoft Corporation of Redmond, Wash., is one example of a suitable operating system. The computer **400** is also illustrated as running a database program **422**, although this may be executed on a separate computer. The "SQL Server" brand of database programs, also available from Microsoft Corporation, is an example of a suitable database program.

[0026] For illustration purposes, operating system **420** and database program **422** are shown as discrete blocks stored in the non-volatile memory **406**, although it is recognized that such programs and components reside at various times in different storage components of the computer **400** and are executed by the processor **402**. Generally, these software components are stored in non-volatile memory **406** and from there, are loaded at least partially into the volatile main memory **404** for execution on the processor **402**.

[0027] The "Windows 2000" operating system includes a service known as "Windows Management Instrumentation" (WMI) **430**. WMI is an example of a management infrastructure for managing the operating system. It is Microsoft's implementation of the Web-Based Enterprise Management (WBEM) initiative, an industry standard administered by the Distributed Management Task Force (DMTF). WMI provides an object-oriented way of monitoring, configuring and controlling systems, services, and applications on the Windows platform. WMI objects are defined with logical classes derived from the Common Information Model (CIM). WMI provides services such as SQL query language support and programmable event notification.

[0028] To support CIM, WMI maintains a schema definition. The schema definition consists of one or more data structures that are typically stored on some form of non-volatile media. A schema can be installed in one or more namespaces. The schema contains a definition of all classes, their properties, and associations.

[0029] In particular, the computer **400** supports a database schema **432** for use in WMI to represent manageable components of the SQL server database. Such managed objects include tables, files, configuration data, and other logical components. The database schema is located in its own namespace (e.g., root\MicrosoftSQLServer).

[0030] Database Schema

[0031] The remaining discussion pertains to an exemplary management instrumentation schema for an SQL Server database software. The schema is depicted in its entirety in FIGS. **5-14**. Throughout the drawings, conventional CIM classes are designated by the prefix "CIM". Classes that are defined for the SQL Server database software are designated by the prefix "MSSQL". Additionally, another group of classes pertaining to an operating system are also defined. In this example, the operating system is a "Windows" brand operating system from Microsoft Corporation and the classes are designated by the prefix "Win32".

[0032] The schema covers the following five main categories:

[0033] Application System: This category includes SQL Server settings, services, and logs.

[0034] Database Objects: This category includes the components of a database system such as databases, tables, keys, and columns.

[0035] Security: This category includes users, roles, authentication login records, and permission settings for the user and roles.

[0036] Physical Storage: This category includes the physical files and file groups that are used to store the data.

[0037] Bulk Data Operations: This category includes operations such as backup, restore, and bulk copy of data.

[0038] Rather than try to present the entire schema in one drawing, which is impractical, the schema is illustrated over multiple drawings according to the above categories. This results in some classes being replicated in more than one drawing. However, at the risk of some redundancy, each drawing portrays that portion of the schema that covers the category or subcategory.

[0039] Additionally, the schema is generally described below with reference to the corresponding figures. One exemplary implementation of all schema classes, as well as their properties, methods, and associations, is provided following this general description beneath the heading "Exemplary Implementation of Database Schema".

[0040] A. Application System

[0041] The schema portions covering the application system of the SQL Server database can be separated into three subcategories: (1) settings, (2) services, and (3) transactions and error logs. The schema portions for these three subcategories are illustrated in FIGS. **5-7**.

[0042] **FIG. 5** shows a setting portion **500** of the SQL Server schema. Setting portion **500** has a superclass **502** entitled "MSSQL_Setting", which is a subclass of a CIM class **504** entitled "CIM_Setting". Beneath the MSSQL_Setting class **502** are five subclasses: MSSQL_DatabaseSetting **510** that represents operational settings for a database; MSSQL_ConfigValue **512** that represents configuration values; MSSQL_LanguageSetting **514** that exposes the properties of an installed language record so that error and status information can be presented as localized text; MSSQL_RegistrySetting **516** that represents the installation and run-time parameters stored in the registry; and MSSQL_SQLServerConnectionSetting **518** represents default connection settings used by a WMI provider to connect to SQL DMO (Distributed Management Objects).

[0043] **FIG. 5** also illustrates a separate set of CIM classes represented by base class **520** entitled "CIM_ManagedSystemElement", which is a base class for the system element hierarchy. Any distinguishable component of a system is a candidate for inclusion in this class. Examples include software components (e.g., files), devices (e.g., disk drives, controllers), and physical components (e.g., chips, cards). Beneath the base class is CIM_LogicalElement class **522**, which is the base class for all the components of the system that represent abstract system components (e.g., profiles, processes, system capabilities in the form of logical devices, etc.). A subclass of class **522** is CIM_System class **524**, which represents a logical element that aggregates an enumerable set of managed system elements. Beneath CIM_System class **524** is CIM_ApplicationSystem class **526**, which represents an application or a software system that supports a particular business function and that can be managed as independent units.

[0044] The schema **500** also defines three extensions to the conventional CIM classes, including MSSQL_DBMSObject class **530**, MSSQL_Database class **532**, and MSSQL_SQLServer class **534**. MSSQL_DBMSObject class **530** is a subclass of CIM_LogicalElement class **522** and represents objects in a database system, such as databases, tables, keys and constraints. MSSQL_Database class **532** represents instances of SQL Server databases and MSSQL_SQLServer class **532** represents instances of the SQL Server.

[0045] **FIG. 5** also illustrates associations between the MSSQL_Setting class **502** and MSSQL_Database class **532** and MSSQL_SQLServer class **534**. The associations are illustrated as broken lines to distinguish them from the solid inheritance lines. The associations are defined as association-type classes, which include properties and methods like normal classes. Consider an association **540** named "MSSQL_DatabaseDatabaseSetting" that is defined between the MSSQL DatabaseSetting class **510** and the MSSQL_Database class **532**. The MSSQL_DatabaseDatabaseSetting association class **540** associates a SQL Server database to an instance of the MSSQL_DatabaseSetting class **510** that contains the settings for the database.

[0046] Other association classes include the following:

[0047] MSSQL_SQLServerConfigValue class **542** represents an association between a SQL Server installation and the configured value settings for the installation.

[0048] MSSQL_SQLServerLanguageSetting class **544** represents an association between a SQL Server installation and its language settings.

[0049] MSSQL_SQLServerRegistrySetting class **546** represents an association between a SQL Server installation and its registry setting.

[0050] MSSQL_SQLServerSQLServerConnection class **548** represents an association between a SQL Server installation and settings used by the WMI SQL Server administration provider to connect to the SQL Server.

[0051] There is also an association class **550**, named "MSSQL_SQLServerDatabase", between the MSSQL_SQLServer class **534** and the MSSSQL_Database class **532** that represents an association between a SQL Server installation and a database that is part of the installation. Additionally, **FIG. 5** illustrates a CIM_ElementSetting class **552** that associates the CIM_ManagedSystemElement class **520** with the CIM_Setting class **504**.

[0052] **FIG. 6** shows a services portion **600** of the SQL Server schema. Services portion **600** includes the CIM_Setting class **504**, a Win32_BaseService class **604** that represents executable objects installed in a registry database and maintained by the Service Control Manager, a MSSQL_FullTextCatalogService class **606** that represents a single search persistent data store, and a Win32_Service class **608** that represents a service on a 32-bit "Windows"-brand operating system. There is one association class **610**, named "MSSQL_FullTextWin32Service", that associates the MSSQL_FullTextCatelogService class **606** with the Win32_Service class **608**.

[0053] **FIG. 7** shows a portion **700** of the SQL Server schema for the transaction and error logs. Similar to the setting schema portion **500** of **FIG. 5**, schema portion **700** includes the CIM_LogicalElement class **522**, the CIM_System class **524**, the CIM_ApplicationSystem class **526**, the MSSQL_SQLServer class **534**, the MSSQL_DBMSObject class **530**, and the MSSQL_Database class **532**. New to the schema **700** are a set of subclasses of the CIM_LogicalELement class **522**. These subclasses include MSSQL_ErrorLogEntry class **702** that represents entries in a SQL Service error log, MSSQL_ErrorLog **704** that represents the SQL Service error logs, CIM_LogicalFile class **706** that represents a named collection of data located in a file system on a storage extent, and MSSQL_TransactionLog **708** that represents the transaction log of the SQL Server database. A CIM_DataFile class **712** inherits from the CIM_Logical File

class **706** to represent a type of logical file that is a named collection of data or executable code.

[**0054**] The schema **700** defines several association classes, including:

[**0055**] MSSQL_ErrorLogErrorLogEntry class **720** represents an association between an error log and an entry in the error log.

[**0056**] MSSQL_SQLServerErrorLog class **722** represents an association between a SQL Server installation and the error log used by the installation.

[**0057**] MSSQL_ErrorLogDataFile class **724** represents an association between the error log and the operating system file used to store the error log.

[**0058**] MSSQL_TransactionLogDataFile class **726** represents an association between the transaction log and the operating system file that is used to store the log.

[**0059**] MSSQL_DatabaseTransactionLog class **728** represents an association between the database and the transaction log for the database.

[**0060**] MSSQL_SQLServerDatabase class **550**, described previously, represents an association between a SQL Server installation and a database that is part of the installation.

[**0061**] B. Database Objects

[**0062**] The portions of the schema covering the database objects of the SQL Server database can be divided into four subcategories: (1) objects (e.g., tables, views, stored procedures, etc.), (2) index, (3) constraints, and (4) keys. The schema portions for these four subcategories are illustrated in FIGS. **8-11**.

[**0063**] **FIG. 8** shows a portion **800** of the SQL Server schema for the database objects, such as tables, columns, views, databases, stored procedures, and so on. The base class for database objects schema **800** is the MSSQL_DBMSObject class **530**. Subclasses of this base class **530** are MSSQL_Column class **802** that represents a column in a table, MSSQL_Trigger class **804** that represents a trigger to be executed when a specified data modification, such as an attempt to delete a row, is attempted on the table, MSSQL_Table class **806** that represents a table in the SQL database, MSSQL_UserDefinedFunction class **808** that represents a user defined function in the SQL database, MSSQL_Database class **532**, MSSQL_StoredProcedure class **810** that represents standard as well as extended stored procedures defined in the SQL database, MSSQL_View class **812** that represents view tables in the database, and MSSQL_StoredProcedureParameter class **814** that represents the input and output parameters of a stored procedure.

[**0064**] The database objects schema **800** defines several association classes, including:

[**0065**] MSSQL_TableColumn class **820** represents an association between a table and a column contained in the table.

[**0066**] MSSQL_TableTrigger class **822** represents an association between a table and a trigger defined for the table.

[**0067**] MSSQL_DatabaseTable class **824** associates a database to all the tables contained in the database.

[**0068**] MSSQL_DatabaseUserDefinedFunction class **826** represents an association between a database and a user-defined function defined within the database.

[**0069**] MSSQL_DatabaseView class **828** associates a database to the view contained within the database.

[**0070**] MSSQL_DatabaseStoredProcedure class **830** represents an association between the database and a stored procedure defined within the database.

[**0071**] MSSQL_StoredProcedureStoredProcedureParameter class **832** associates a stored procedure to a parameter used in the stored procedure.

[**0072**] **FIG. 9** shows an index portion **900** of the SQL Server schema. Again, the base class is the MSSQL_DBMSObject class **530**. Here, two subclasses are defined: MSSQL_Column class **802** and MSSQL_Index class **902** that represents an index for a table. Additionally, the index portion of the schema shows a CIM class **904** named "CIM_StatisticalInformation", having a subclass **906** named "MSSQL_IndexTableInformation" that represents the information regarding the age and structure of the index statistical information.

[**0073**] Two association classes are also defined:

[**0074**] MSSQL_IndexColumn class **910** represents an association between an index and a column that participates in the index.

[**0075**] MSSQL_IndexStatistics class **912** represents an association between an index and the statistical information stored with the index.

[**0076**] **FIG. 10** shows a portion **1000** of the SQL Server schema pertaining to constraints, which are the rules and checks to create constraints on data. The base class is once again the MSSQL_DBMSObject class **530**. Beneath this base class **530** are the familiar MSSQ Table class **806**, the MSSQL_Column class **802**, and MSSQL_Database class **532**. Another subclass is MSSQL_Constraint class **1002** that represents constraints defined in the SQL Server database. There are three types of constraints that can be defined: checks, keys, and rules. MSSQL_Default class **1004** represents the attributes of a single SQL Server default. Such defaults provide data to columns and user-defined data types when no other data is available on an "INSERT" statement execution.

[**0077**] The remaining subclasses of base class **530** are MSSQL_DRIDefault class **1006** that represents the properties of a SQL Server column DEFAULT constraint and MSSQL Datatype class **1008** that represents all the datatypes defined in a SQL Server installation, including both user-defined datatypes, as well as system-defined datatypes. Beneath the MSSQL_Datatype class **1008** are two subclasses that represent the two types of datatypes: MSSQL_UserDatatype class **1010** and MSSQL_SystemDatatype class **1012**.

[**0078**] The remaining classes illustrated in constraint portion **1000** of the SQL schema are MSSQL_Check class **1014**

that represents the attributes of a SQL Server integrity constraint and MSSQL_Rule class **1016** that represents a single data-integrity rule.

[**0079**] There are a number of association classes defined in this portion of the schema. The association classes include:

[**0080**] MSSQL_TableCheck class **1020** represents an association between a table and the checks defined for the table.

[**0081**] MSSQL_UserDatatypeDefault class **1022** represents an association between a user-defined datatype and the rule bound to the column.

[**0082**] MSSQL_DatabaseDefault class **1024** associates a database to the defaults defined within the database.

[**0083**] MSSQL_ColumnRule class **1026** represents an association between a column and a rule bound to the column.

[**0084**] MSSQL_ColumnDRIDefault class **1028** associates a column to a DRI default.

[**0085**] MSSQL_ColumnDatatype class **1030** associates a column with its data type.

[**0086**] MSSQL_DatabaseRule class **1032** represents an association between a database and the rules defined within the database.

[**0087**] MSSQL_UserDatatypeRule class **1034** represents an association between a user defined datatype and the rule bound to the column.

[**0088**] MSSQL_DatabaseDatatype class **1036** associates a database to the datatypes defined within the database.

[**0089**] MSSQL_BaseDatatype class **1038** represents an association between a user-defined datatype and the system datatype from which it is derived.

[**0090**] MSSQL_ColumnDefault class associates a column to the default for the column.

[**0091**] **FIG. 11** shows a portion **1100** of the SQL Server schema pertaining to keys. The base class is once again the MSSQL_DBMSObject class **530**. Beneath this base class **530** are previously introduced classes including the MSSQL_Constrain class **1002**, the MSSQL_Column class **802**, MSSQ_Table class **806**, and the MSSQL_Database class **532**. Newly presented in this portion of the schema is the MSSQL_Key class **1102** that represents the keys defined for a SQL Server table.

[**0092**] One subclass of key class **1102** is the MSSQL-_CandidateKey class **1104** that represents a candidate key in a SQL Server table. It consists of a set of columns that can uniquely identify a row in a table. Another subclass is MSSQL_ForeignKey class **1106**, which represents the foreign keys defined for a SQL Server database table.

[**0093**] Beneath the candidate key class **1104** are two classes: MSSQL_UniqueKey class **1108** that represents a unique key in a database and MSSQL_PrimaryKey class **1110** that represents a primary key of a table.

[**0094**] Association classes defined in this portion of the schema include:

[**0095**] MSSQL_KeyColumn class **1120** represents an association between a key and a column that is part of the key.

[**0096**] MSSQL_TableKey class **1122** represents an association between a table and a key defined for the table.

[**0097**] MSSQL_ReferencedTable class **1124** represents an association between a foreign key and the table that contains the primary key referenced by the foreign key.

[**0098**] MSSQL_DatabaseCandidateKey **1126** represents an association between a database and a candidate key that is present in one of the tables in the database. This association allows an application to perform a single traversal to find the candidate keys in a database.

[**0099**] MSSQL_ReferencedKey class **1128** represents an association between a foreign key and the candidate key that the foreign key references.

[**0100**] C. Security

[**0101**] The schema portions covering the security features of the SQL Server database can be separated into two subcategories: (1) server login records and (2) permissions. The schema portions for these subcategories are illustrated in FIGS. **12-13**.

[**0102**] **FIG. 12** shows a security portion **1200** of the SQL Server schema pertaining to server login records. Familiar classes in the security portion **1200** include CIM_Setting class **504**, MSSQL_Setting class **502**, MSSQL_SQLServer **534**, MSSQL_DBMSObject **530**, and MSSQL_Database **532**. Newly presented classes include MSSQL_Integrated-SecuritySetting class **1202** that represents the integrated security settings when WMI interacts with SQL Server. MSSQL_DBMSUserObject class **1204** represents all objects related to user authentication, including objects such as users, logins, and roles defined for a SQL Server installation. The user, logins, and roles are represented by MSSQL_User class **1206**, the MSSQL_Login class **1208**, and the MSSQL_Role class **1210**, respectively, which are subclasses to the MSSQL_DBMSUserObject class **1204**.

[**0103**] With respect to the MSSQL_Role class **1210**, roles are used to establish groups of users with similar security attributes. Permissions can be granted by role, simplifying security planning and administration. Beneath the MSSQL-_Role class **1210** are MSSLQ_SQLServerRole class **1212** that represents a SQL Server security role not constrained to operation within a single database and MSSQL_Database-Role class **1214** that represents the properties of a SQL Server database role.

[**0104**] Also illustrated in security portion **1200** of the SQL Server schema is a Win32_Account class **1216** that contains information about user accounts and group accounts known to a "Win32" operating system. Descendents of this class are group names recognized by a "Windows NT" domain as represented by Win32_Group class **1218** and information about a user account on a "Win32" system as represented by Win32_UserAccount class **1220**.

[0105] Association classes defined in this portion of the schema include the following:

[0106] MSSQL_SQLServerintegratedSecuritySetting class **1230** represents an association between a SQL Server installation and its security settings.

[0107] MSSQL_SQLServerLogin class **1232** represents an association between a SQL server and a login defined within the SQL Server.

[0108] MSSQL_SQLServerServerRole class **1234** represents an association between a SQL Server and server roles defined within the SQL Server.

[0109] MSSQL_SQLServerUser class **1236** represents an association between a SQL Server and a database user. This association allows an application to perform a single traversal to find the database users in a SQL Server and the login to which they are mapped.

[0110] MSSQL_DatabaseLogin class **1238** represents an association between a database and a login that is mapped to a user defined in the database. This association allows an application to perform a single traversal to find the logins used by a database.

[0111] MSSQL_DatabaseOwnerLogin class **1240** represents an association to between a database and the login mapped to the user that owns the database.

[0112] MSSQL_LoginWin32Group class **1242** represents an association between a login and the "Win32" user group used for authentication by the login.

[0113] MSSQL_UserLogin class **1244** represents an association between a database user and the login used to authenticate the user.

[0114] MSSQL_MemberLogin class **1246** represents an association between a SQL Server role and a login that is a member of the role.

[0115] MSSQL_LoginDefaultDatabase class **1248** represents an association between a login and the default database for the login.

[0116] MSSQL_LoginWin32UserAccount class **1250** represents an association between a login and the "Win32" user account used for authentication by the login.

[0117] Win32_GroupUser class **1252** represents an association between a group and an account that is a member of that group.

[0118] MSSQL_MemberUser class **1256** represents an association between a database role and a user that is a member of the role.

[0119] MSSQL_BMSObjectOwner class **1258** represents an association between a SQL Server database object and the user who owns the object.

[0120] MSSQL_DatabaseUser class **1260** represents an association between a database and a user defined for the database.

[0121] MSSQL_MemberDatabaseRole class **1262** associates two database roles, one being a member of the other.

[0122] MSSQL_DatabaseDatabaseRole class **1264** associates database role to the database within which the role is defined.

[0123] **FIG. 13** shows a portion **1300** of the SQL Server schema that relates to permissions of users and roles. All of the classes illustrated in **FIG. 13** have been introduced above. New to this schema portion are the association-type class that associates a user or a role with a database object, such as a table, view, and database. In particular, the associations define permissions granted or denied to specific users/roles with regard to specific databases and/or portions of databases.

[0124] The association classes include:

[0125] MSSQL_DatabaseRoleDatabasePermission class **1302** represents the permissions that a database role has for the database in which it is defined.

[0126] MSSQL_UserDatabasePermission class **1304** represents the permissions granted to a user for a database.

[0127] MSSQL_UserStoredProcedurePermission class **1306** represents the permissions granted to a user for a stored procedure.

[0128] MSSQL_DatabaseRoleStoredProcedurePermission class **1308** represents the permissions that a database role has for a stored procedure.

[0129] MSSQL_UserTablePermission class **1310** represents the permissions granted to a user for a table.

[0130] MSSQL_DatabaseRoleTablePermission **1312** represents the permissions that a database role has for a table.

[0131] MSSQL_UserUserDefinedFunctionPermission **1314** represents the permissions granted to a user for a stored procedure.

[0132] MSSQL_DatabaseRoleUserDefinedFunctionPernission **1316** represents the permissions that a database role has for a table.

[0133] MSSQL_UserViewPermission **1318** represents the permissions granted to a user for a view.

[0134] D. Physical Storage

[0135] **FIG. 14** shows a portion **1400** of the SQL Server schema that relates physical storage. Many of the classes have been described previously. New to this schema portion are MSSQL_FileGroup class **1402** that exposes the attributes of a the SQL Server filegroup, MSSQL_Extension class **1404** that represents extensions made via associations to a managed system element, and MSSQL_DatabaseFile class **1406** that is an extension to the CIM_DataFile class **712**.

[0136] The association classes in portion **1400** include the following:

[0137] MSSQL_Extends class **1410** is an abstract association class that associates a class with another

class that extends the former class by defining some new properties and methods.

[0138] MSSQL_DatabaseFileDataFile class **1412** associates a CIM_Datafile class **712** to the MSSQL-_DatabaseFile class **1406** class that contains database file specific properties of an operating system file.

[0139] MSSQL_FileGroupDatabaseFile class **1414** associates a database file group to the operating system files that are part of the group.

[0140] MSSQL_IndexFileGroup class **1416** represents an association between an index and a file group that stores the index.

[0141] MSSQL_KeyFileGroup class **1418** represents an association between a key and the file group used to store the key.

[0142] MSSQL_TableFileGroup class **1420** represents an association between a table and the file groups used to store the table.

[0143] MSSQL_TableTextFileGroup class **1422** associates a table with the file group that is used to store the variable length data in the table.

[0144] MSSQL_DatabaseFileGroup class **1424** represents an association between a database and the file group that contains the operating system files that store the data for the database.

[0145] E. Bulk Data Operations

[0146] The operations related to data backup and restore are performed by executing certain methods on the MSSQL_SQLServer class **534**, MSSQL_Database class **532**, MSSQL_Table class **806**, and MSSQL_View class **812**. The settings for tehse operations are specified using instances of the following classes:

[0147] MSSQL_RestoreSetting class specifies the behavior of a restore operation for a SQL Server database or log. The class is also used to specify the behavior of the verify operation for a SQL Server backup.

[0148] MSSQL_BackupSetting class specifies the settings for a backup operation.

[0149] MSSQL_TransferSetting class represents the settings used to control the data elements moved from one SQL database to another.

[0150] MSSQL_BulkCopySetting class represents the settings for importing or exporting data from a table or a view.

[0151] Exemplary Implementation of Database Schema

[0152] The following provides one exemplary implementation of all schema classes introduced above, as well as their properties, methods, and associations. The various classes are grouped together as CIM classes, MSSQL classes, and Win32 classes.

[0153] A. CIM Classes

[0154] CIM_ApplicationSystem

[0155] The CIM_ApplicationSystem class is used to represent an application or a software system that supports a particular business function and that can be managed as independent units. Such a system can be decomposed into its functional components using the CIM_SoftwareFeature class. The software features for a particular application or software system are located using the CIM_ApplicationSystemSoftwareFeature association.

[0156] Properties

[0157] string Caption

[0158] Access Type: Read-only

[0159] The Caption property is a short textual description (one-line string) of the object.

[0160] Maximum Length: 64

[0161] string CreationClassName

[0162] Access Type: Read-only

[0163] The CreationClassName property indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.

[0164] string Description

[0165] Access Type: Read-only

[0166] The Description property provides a textual description of the object.

[0167] datetime InstallDate

[0168] Access Type: Read-only

[0169] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[0170] string Name

[0171] Access Type: Read-only

[0172] The inherited Name property serves as key of a CIM_System instance in an enterprise environment.

[0173] string NameFormat

[0174] Access Type: Read-only

[0175] The CIM_System object and its derivatives are top level objects of CIM. They provide the scope for numerous components. Having unique system keys is required. A heuristic can be defined in individual system subclasses to attempt to always generate the same system name key. The NameFormat property identifies how the system name was generated, using the subclass' heuristic.

[0176] string PrimaryOwnerContact

[0177] Access Type: Read-only

[0178] A string that provides information on how the primary system owner can be reached (e.g. phone number, email address, . . .).

[0179] string PrimaryOwnerName

[0180] Access Type: Read-only

[0181] The name of the primary system owner.

[0182] string Roles []

[0183] Access Type: Read/Write

[0184] A collection of strings that specify the roles this system plays in the IT-environment.

[0185] string Status

[0186] Access Type: Read-only

[0187] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[0188] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[0189] Maximum Length: 10

[0190] CIM_LogicalFile

[0191] Abstract Class

[0192] The CIM_LogicalFile class represents a named collection of data (this can be executable code) located in a file system on a storage extent.

[0193] Properties

[0194] uint32 AccessMask

[0195] Access Type: Read-only

[0196] The AccessMask property is a bit array representing the access rights to the given file or directory held by the user or group on whose behalf the instance is returned. This property is only supported under Windows NT and Windows 2000. On Windows 98 and on Windows NT/2000 FAT volumes, FULL_ACCESS is returned, indicating no security has been set on the object.

Bit Position	Description
0	FILE_READ_DATA (file) or FILE_LIST_DIRECTORY (directory)
1	FILE_WRITE_DATA (file) or FILE_ADD_FILE (directory)

-continued

Bit Position	Description
2	FILE_APPEND_DATA (file) or FILE_ADD_SUBDIRECTORY (directory)
3	FILE_READ_EA
4	FILE_WRITE_EA
5	FILE_EXECUTE (file) or FILE_TRAVERSE (directory)
6	FILE_DELETE_CHILD (directory)
7	FILE_READ_ATTRIBUTES
8	FILE_WRITE_ATTRIBUTES
16	DELETE
17	READ_CONTROL
18	WRITE_DAC
19	WRITE_OWNER
20	SYNCHRONIZE

[0197] boolean Archive

[0198] Access Type: Read-only

[0199] The Archive property is a boolean value indicating that the file should be archived.

[0200] string Caption

[0201] Access Type: Read-only

[0202] The Caption property is a short textual description (one-line string) of the object.

[0203] Maximum Length: 64

[0204] boolean Compressed

[0205] Access Type: Read-only

[0206] The Compressed property is a boolean value indicating that the file is compressed.

[0207] string CompressionMethod

[0208] Access Type: Read-only

[0209] The CompressionMethod property is a free form string indicating the algorithm or tool used to compress the logical file. If it is not possible (or not desired) to describe the compression scheme (perhaps because it is not known), use the following words: "Unknown" to represent that it is not known whether the logical file is compressed or not, "Compressed" to represent that the file is compressed but either its compression scheme is not known or not disclosed, and "Not Compressed" to represent that the logical file is not compressed.

[0210] string CreationClassName

[0211] Access Type: Read-only

[0212] The CreationClassName property is a string indicating the name of this class.

[0213] datetime CreationDate

[0214] Access Type: Read-only

[0215] The CreationDate property is a datetime value indicating the file's creation date.

[0216] string CSCreationClassName

[0217] Access Type: Read-only

[0218] The CSCreationClassName property is a string indicating the class of the computer system.

[0219] string CSName

[0220] Access Type: Read-only

[0221] The CSName property is a string indicating the name of the computer system.

[0222] string Description

[0223] Access Type: Read-only

[0224] The Description property provides a textual description of the object.

[0225] string Drive

[0226] Access Type: Read-only

[0227] The Drive property is a string representing the drive letter (including colon) of the file. Example: c:

[0228] string EightDotThreeFileName

[0229] Access Type: Read-only

[0230] The EightDotThreeFileName property is a string representing the DOS-compatible file name for this file. Example: c:\progra~1

[0231] boolean Encrypted

[0232] Access Type: Read-only

[0233] The Encrypted property is a boolean value indicating that the file is encrypted.

[0234] string EncryptionMethod

[0235] Access Type: Read-only

[0236] The EncryptionMethod property is a free form string indicating the algorithm or tool used to encrypt the logical file. If it is not possible (or not desired) to describe the encryption scheme (perhaps for security reasons), use the following words: "Unknown" to represent that it is not known whether the logical file is encrypted or not, "Encrypted" to represent that the file is encrypted but either its encryption scheme is not known or not disclosed, and "Not Encrypted" to represent that the logical file is not encrypted.

[0237] string Extension

[0238] Access Type: Read-only

[0239] The Extension property is a string representing the file's extension (without the dot). Example: txt, mof, mdb.

[0240] string FileName

[0241] Access Type: Read-only

[0242] The FileName property is a string representing the filename (without extension) of the file. Example: autoexec

[0243] uint64 FileSize

[0244] Access Type: Read-only

[0245] The FileSize property represents the size of the file (in bytes).

[0246] Units: Bytes

[0247] string FileType

[0248] Access Type: Read-only

[0249] The FileType property is a string descriptor representing the file type (indicated by the Extension property).

[0250] string FSCreationClassName

[0251] Access Type: Read-only

[0252] The FSCreationClassName property is a string indicating the class of the file system.

[0253] string FSName

[0254] Access Type: Read-only

[0255] The FSName property is string indicating the name of the file system.

[0256] boolean Hidden

[0257] Access Type: Read-only

[0258] The Hidden property is a boolean value indicating if the file is hidden.

[0259] datetime InstallDate

[0260] Access Type: Read-only

[0261] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[0262] uint64 InUseCount

[0263] Access Type: Read-only

[0264] The InUseCount property is an integer indicating the number of 'file opens' that are currently active against the file.

[0265] datetime LastAccessed

[0266] Access Type: Read-only

[0267] The LastAccessed property is a datetime value indicating the time the file was last accessed.

[0268] datetime LastModified

[0269] Access Type: Read-only

[0270] The LastModified property is a datetime value indicating the time the file was last modified.

[0271] [key] string Name

[0272] Access Type: Read-only

[0273] The Name property is a string representing the inherited name that serves as a key of a logical file instance within a file system. Full path names should be provided. Example: c:\winnt\system\win.ini

[0274] string Path

[0275] Access Type: Read-only

[0276] The Path property is a string representing the path of the file. This includes leading and trailing backslashes. Example:\windows\system\

[0277] boolean Readable

[0278] Access Type: Read-only

[0279] The Readable property is a boolean value indicating if the file can be read.

[0280] string Status

[0281] Access Type: Read-only

[0282] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror-resilvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[0283] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[0284] Maximum Length: 10

[0285] boolean System

[0286] Access Type: Read-only

[0287] The system property is a boolean value indicating if the file is a system file.

[0288] boolean Writeable

[0289] Access Type: Read-only

[0290] The Writeable property is a boolean value indicating if the file can be written.

[0291] Methods

[0292] The CIM_LogicalFile class supports the following methods:

Method Name	Description
ChangeSecurityPermissions	The ChangeSecurityPermissions method changes the security permissions for the logical file.
ChangeSecurityPermissionsEx	The ChangeSecurityPermissionsEx method changes the security permissions for the logical file. The ChangeSecurityPermissionsEx method is an extended version of the ChangeSecurityPermissions method.

-continued

Method Name	Description
Copy	The Copy method copies the logical file or directory.
CopyEx	The CopyEx method copies the logical file or directory. CopyEx is an extended version of the Copy method.
Compress	The Compress method compresses the logical file (or directory).
CompressEx	The CompressEx method compresses the logical file (or directory). CompressEx is an extended version of the Compress method.
Delete	The Delete method will delete the logical file (or directory).
DeleteEx	The DeleteEx method will delete the logical file (or directory) specified in the object path. DeleteEx is an extended version of the Delete method.
GetEffectivePermission	The GetEffectivePermission method determines whether the caller has permissions for the file (or directory).
Rename	The Rename method renames the logical file (or directory).
TakeOwnerShip	The TakeOwnerShip method obtains ownership of the logical file.
TakeOwnerShipEx	The TakeOwnerShipEx method obtains ownership of the logical file. TakeOwnerShipEx is an extended version of the TakeOwnerShip method.
Uncompress	The Uncompress method uncompresses the logical file (or directory).
UncompressEx	The UncompressEx method uncompresses the logical file (or directory). UncompressEx is an extended version of the Uncompress method.

[0293] CIM_LogicalIdentity

[0294] Abstract Class

[0295] Association Class

[0296] CIM_LogicalIdentity is an abstract and generic association, indicating that two CIM_LogicalElements represent different aspects of the same underlying entity. This relationship conveys what could be defined with multiple inheritance. It is restricted to the 'logical' aspects of a CIM_ManagedSystemElement. In most scenarios, the Identity relationship is determined by the equivalence of Keys or some other identifying properties of the related Elements. The association should only be used in well-understood scenarios. This is why the association is abstract—allowing more concrete definition and clarification in subclasses. One of the scenarios where this relationship is reasonable is to represent that a Device is both a 'bus' entity and a 'functional' entity. For example, a Device could be both a USB (bus) and a Keyboard (functional) entity.

[0297] References

[0298] [key] CIM LogicalElement SystemElement

[0299] Access Type: Read-only

[0300] SystemElement represents one aspect of the LogicalElement.

[0301] [key] CIM_LogicalElement SameElement

[0302] Access Type: Read-only

[0303] SameElement represents an alternate aspect of the System entity.

[0304] CIM_ManagedSystemElement

[0305] Abstract Class

[0306] The CIM_ManagedSystemElement class is the base class for the system element hierarchy. Membership criteria: Any distinguishable component of a system is a candidate for inclusion in this class. Examples: software components, such as files; and devices, such as disk drives and controllers, and physical components such as chips and cards.

[0307] Properties

[0308] string Caption

[0309] Access Type: Read-only

[0310] The Caption property is a short textual description (one-line string) of the object.

[0311] Maximum Length: 64

[0312] string Description

[0313] Access Type: Read-only

[0314] The Description property provides a textual description of the object.

[0315] datetime InstallDate

[0316] Access Type: Read-only

[0317] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[0318] string Name

[0319] Access Type: Read-only

[0320] The Name property defines the label by which the object is known. When subclassed, the Name property can be overridden to be a Key property.

[0321] string Status

[0322] Access Type: Read-only

[0323] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror-resilvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[0324] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[0325] Maximum Length: 10

[0326] Associations

[0327] CIM_ManagedSystemElement is associated to

[0328] CIM_ManagedSystemElement as the GroupComponent property of the CIM_Component association.

[0329] CIM_ManagedSystemElement is associated to CIM_ManagedSystemElement as the PartComponent property of the CIM_Component association.

[0330] CIM_ManagedSystemElement is associated to CIM_Setting as the Element property of the CIM_ElementSetting association.

[0331] CIM_ManagedSystemElement is associated to CIM_ManagedSystemElement as the Antecedent property of the CIM_Dependency association.

[0332] CIM_ManagedSystemElement is associated to CIM_ManagedSystemElement as the Dependent property of the CIM_Dependency association.

[0333] CIM_ManagedSystemElement is associated to CIM_StatisticalInformation as the Element property of the CIM_Statistics association.

[0334] CIM_ManagedSystemElement is associated to MSSQL_Extension as the ExtendedElement property of the MSSQL_Extends association.

[0335] CIM_ManagedSystemElement is associated to CIM_ManagedSystemElement as the Containee property of the MSSQL_Containment association.

[0336] CIM_ManagedSystemElement is associated to CIM_ManagedSystemElement as the Container property of the MSSQL_Containment association.

[0337] CIM_Process: CIM_LogicalElement

[0338] Abstract Class

[0339] The CIM_Process class is derived from CIM_LogicalElement. It is intended to represent a program in execution, running under an operating system. Processes are also known as tasks.

[0340] Properties

[0341] string Caption

[0342] Access Type: Read-only

[0343] The Caption property is a short textual description (one-line string) of the object.

[0344] Maximum Length: 64

[0345] string CreationClassName

[0346] Access Type: Read-only

[0347] The inherited CreationClassName property indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.

[0348] datetime CreationDate

[0349] Access Type: Read-only

[0350] Time that the process began executing

[0351] string CSCreationClassName

[0352] Access Type: Read-only

[0353] The inherited CSCreationClassName property is a string indicating the class of the computer system.

[0354] string CSName

[0355] Access Type: Read-only

[0356] The inherited CSName property is a string indicating the name of the computer system.

[0357] string Description

[0358] Access Type: Read-only

[0359] The Description property provides a textual description of the object.

[0360] uint**16** ExecutionState

[0361] Access Type: Read-only

[0362] Indicates the current operating condition of the process.

Value	Description
0	Unknown
1	Other
2	Ready
3	Running
4	Blocked
5	Suspended Blocked
6	Suspended Ready

[0363] string Handle

[0364] Access Type: Read-only

[0365] A string used to identify the process. A process ID is a process handle.

[0366] datetime InstallDate

[0367] Access Type: Read-only

[0368] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[0369] uint**64** KernelModeTime

[0370] Access Type: Read-only

[0371] Time in kernel mode, in milliseconds. If this information is not available, a value of 0 should be used.

[0372] Units: Milliseconds (ms)

[0373] string Name

[0374] Access Type: Read-only

[0375] The Name property defines the label by which the object is known. When subclassed, the Name property can be overridden to be a Key property.

[0376] string OSCreationClassName

[0377] Access Type: Read-only

[0378] The inherited OSCreationClassName property indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.

[0379] string OSName

[0380] Access Type: Read-only

[0381] The inherited OSName property serves as key of an operating system instance within a computer system.

[0382] uint**32** Priority

[0383] Access Type: Read-only

[0384] Priority indicates the urgency or importance of execution of a process. If a priority is not defined for a process, a value of **0** should be used.

[0385] string Status

[0386] Access Type: Read-only

[0387] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[0388] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[0389] Maximum Length: 10

[0390] datetime TerminationDate

[0391] Access Type: Read-only

[0392] Time that the process was stopped or terminated.

[0393] uint**64** UserModeTime

[0394] Access Type: Read-only

[0395] Time in user mode, in milliseconds. If this information is not available, a value of 0 should be used.

[0396] Units: Milliseconds (ms)

[0397] uint64 WorkingSetSize

[0398] Access Type: Read-only

[0399] The amount of memory in bytes that a process needs to execute efficiently, for an operating system that uses page-based memory management. If an insufficient amount of memory is available (<working set size), thrashing will occur. If this information is not known, NULL or 0 should be entered. If this data is provided, it could be monitored to understand a process' changing memory requirements as execution proceeds.

[0400] Units: Bytes

[0401] CIM_Service: CIM_LogicalElement

[0402] Abstract Class

[0403] The CIM_Service class represents a logical element that contains the information necessary to represent and manage the functionality provided by a device and/or software feature. A service is a general-purpose object to configure and manage the implementation of functionality. It is not the functionality itself.

[0404] Properties

[0405] string Caption

[0406] Access Type: Read-only

[0407] The Caption property is a short textual description (one-line string) of the object.

[0408] Maximum Length: 64

[0409] string CreationClassName

[0410] Access Type: Read-only

[0411] CreationClassName indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.

[0412] string Description

[0413] Access Type: Read-only

[0414] The Description property provides a textual description of the object.

[0415] datetime InstallDate

[0416] Access Type: Read-only

[0417] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[0418] [key] string Name

[0419] Access Type: Read-only

[0420] The Name property uniquely identifies the service and provides an indication of the functionality that is managed. This functionality is described in more detail in the object's Description property.

[0421] boolean Started

[0422] Access Type: Read-only

[0423] Started is a boolean indicating whether the service has been started (TRUE), or stopped (FALSE).

[0424] string StartMode

[0425] Access Type: Read-only

[0426] StartMode is a string value indicating whether the service is automatically started by a operating system, or only started upon request.

[0427] Values are: "Automatic", "Manual"

[0428] string Status

[0429] Access Type: Read-only

[0430] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror-resilvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[0431] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[0432] Maximum Length: 10

[0433] string SystemCreationClassName

[0434] Access Type: Read-only

[0435] The type name of the system that hosts this service

[0436] string SystemName

[0437] Access Type: Read-only

[0438] The name of the system that hosts this service

[0439] Methods

[0440] The CIM_Service class supports the following methods:

Method Name	Description
StartService	The StartService method places the service in the started state.
StopService	The StopService method places the service in the stopped state.

[0441] CIM_Setting

[0442] Abstract Class

[0443] The CIM_Setting class represents configuration-related and operational parameters for one or more CIM_ManagedSystemElement(s). A CIM_ManagedSystemElement may have multiple Setting objects associated with it. The current operational values for an Element's parameters are reflected by properties in the Element itself or by properties in its associations. These properties do not have to be the same values present in the CIM_Setting object. For example, a modem may have a CIM_Setting baud rate of 56 Kb/sec but be operating at 19.2 Kb/sec.

[0444] Properties

[0445] string Caption

[0446] Access Type: Read-only

[0447] A short textual description (one-line string) of the CIM_Setting object.

[0448] Maximum Length: 64

[0449] string Description

[0450] Access Type: Read-only

[0451] A textual description of the CIM_Setting object.

[0452] tring SettingID

[0453] Access Type: Read-only

[0454] The identifier by which the CIM_Setting object is known.

[0455] Maximum Length: 256

[0456] Associations

[0457] CIM_Setting is associated to CIM_ManagedSystemElement as the Setting property of the CIM_ElementSetting association.

[0458] CIM_StatisticalInformation

[0459] Abstract Class

[0460] CIM_StatisticalInformation is a root class for any arbitrary collection of statistical data and/or metrics applicable to one or more managed system elements.

[0461] Properties

[0462] string Caption

[0463] Access Type: Read-only

[0464] A short textual description (one-line string) for the statistic or metric.

[0465] Maximum Length: 64

[0466] string Description

[0467] Access Type: Read-only

[0468] A textual description of the statistic or metric.

[0469] string Name

[0470] Access Type: Read-only

[0471] The Name property defines the label by which the statistic or metric is known. When subclassed, the property can be overridden to be a Key property.

[0472] Maximum Length: 256

[0473] Associations

[0474] CIM_StatisticalInformation is associated to CIM_ManagedSystemElement as the Stats property of the CIM_Statistics association.

[0475] CIM_Statistics

[0476] Abstract Class

[0477] Association Class

[0478] CIM_Statistics is an association that relates managed system elements to the statistical group(s) that apply to them.

[0479] References

[0480] CIM_ManagedSystemElement Element

[0481] Access Type: Read-only

[0482] The CIM_ManagedSystemElement for which statistical or metric data is defined.

[0483] CIM_StatisticalInformation Stats

[0484] Access Type: Read-only

[0485] The statistical information/object.

[0486] CIM_StorageExtent: CIM_LogicalDevice

[0487] Abstract Class

[0488] The CIM_StorageExtent class describes the capabilities and management of the various media that exist to store data and allow data retrieval.

[0489] Properties

[0490] uint16 Access

[0491] Access Type: Read-only

[0492] The Access property describes whether the media is readable, writeable, or both. Unknown can also be returned.

Value	Description
0	Unknown
1	Readable
2	Writeable
3	Read/Write Supported

[0493] uint16 Availability

[0494] Access Type: Read-only

[0495] The availability and status of the device. For example, the Availability property indicates that the device is running and has full power, or is in a warning, test, degraded or power save state. Regarding the power saving states, these are defined as follows: Power Save—Unknown indicates that the device is known to be in a power

save mode, but its exact status in this mode is unknown; Power Save—Low Power Mode indicates that the device is in a power save state but still functioning, and may exhibit degraded performance; Power Save—Standby describes that the device is not functioning but could be brought to full power 'quickly'; and Power Save—Warning indicates that the device is in a warning state, though also in a power save mode.

Value	Description
1	Other
2	Unknown
3	Running/Full Power
4	Warning
5	In Test
6	Not Applicable
7	Power Off
8	Off Line
9	Off Duty
10	Degraded
11	Not Installed
12	Install Error
13	Power Save - Unknown
14	Power Save - Low Power Mode
15	Power Save - Standby
16	Power Cycle
17	Power Save - Warning

[0496] uint**64** BlockSize

[0497] Access Type: Read-only

[0498] Size in bytes of the blocks that form this storage extent. If unknown or if a block concept is not valid (for example, for aggregate extents, memory or logical disks), enter a 1.

[0499] Units: Bytes

[0500] string Caption

[0501] Access Type: Read-only

[0502] The Caption property is a short textual description (one-line string) of the object.

[0503] Maximum Length: 64

[0504] uint**32** ConfigManagerErrorCode

[0505] Access Type: Read-only

[0506] Indicates the Win32 Configuration Manager error code.

Value	Description
0	This device is working properly.
1	This device is not configured correctly.
2	Windows cannot load the driver for this device.
3	The driver for this device might be corrupted, or your system may be running low on memory or other resources.
4	This device is not working properly. One of its drivers or your registry might be corrupted.
5	The driver for this device needs a resource that Windows cannot manage.

-continued

Value	Description
6	The boot configuration for this device conflicts with other devices.
7	Cannot filter.
8	The driver loader for the device is missing.
9	This device is not working properly because the controlling firmware is reporting the resources for the device incorrectly.
10	This device cannot start.
11	This device failed.
12	This device cannot find enough free resources that it can use.
13	This device cannot find enough free resources that it can use.
14	Windows cannot verify this device's resources.
15	This device cannot work properly until you restart your computer.
16	This device is not working properly because there is probably a re-enumeration problem.
17	Windows cannot identity all the resources this device uses.
18	This device is asking for an unknown resource type.
19	Reinstall the drivers for this device.
20	Your registry might be corrupted.
21	System failure: Try changing the driver for this device. If that does not work, see your hardware documentation. Windows is removing this device.
22	This device is disabled.
23	System failure: Try changing the driver for this device. If that doesn't work, see your hardware documentation.
24	This device is not present, is not working properly, or does not have all its drivers installed.
25	Windows is still setting up this device.
26	Windows is still setting up this device.
27	This device does not have valid log configuration.
28	The drivers for this device are not installed.
29	This device is disabled because the firmware of the device did not give it the required resources.
30	This device is using an Interrupt Request (IRQ) resource that another device is using.
31	This device is not working properly because Windows cannot load the drivers required for this device.

[0507] boolean ConfigManagerUserConfig

[0508] Access Type: Read-only

[0509] Indicates whether the device is using a user-defined configuration.

[0510] string CreationClassName

[0511] Access Type: Read-only

[0512] CreationClassName indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.

[0513] string Description

[0514] Access Type: Read-only

[0515] The Description property provides a textual description of the object.

[0516] string DeviceID

[0517] Access Type: Read-only

[0518] DeviceID is an address or other identifying information to uniquely name the logical device.

[0519] boolean ErrorCleared

[0520] Access Type: Read-only

[0521] ErrorCleared is a boolean property indicating that the error reported in LastErrorCode property is now cleared.

[0522] string ErrorDescription

[0523] Access Type: Read-only

[0524] ErrorDescription is a free-form string supplying more information about the error recorded in LastErrorCode property, and information on any corrective actions that may be taken.

[0525] string ErrorMethodology

[0526] Access Type: Read-only

[0527] ErrorMethodology is a free-form string describing the type of error detection and correction supported by this storage extent.

[0528] datetime InstallDate

[0529] Access Type: Read-only

[0530] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[0531] uint32 LastErrorCode

[0532] Access Type: Read-only

[0533] LastErrorCode captures the last error code reported by the logical device.

[0534] [key] string Name

[0535] Access Type: Read-only

[0536] The Name property defines the label by which the object is known.

[0537] uint64 NumberOfBIocks

[0538] Total number of consecutive blocks, each block the size of the value contained in the BlockSize property, which form this storage extent. Total size of the storage extent can be calculated by multiplying the value of the BlockSize property by the value of this property. If the value of BlockSize is 1, this property is the total size of the storage extent.

[0539] string PNPDeviceID

[0540] Access Type: Read-only

[0541] Indicates the Win32 Plug and Play device ID of the logical device. Example: *PNP03

[0542] uint16 PowerManagementCapabilities []

[0543] Access Type: Read-only

[0544] Indicates the specific power-related capabilities of the logical device. The array values, 0="Unknown", 1="Not Supported" and 2="Disabled" are self-explanatory. The value, 3="Enabled" indicates that the power management features are currently enabled but the exact feature set is unknown or the information is unavailable. "Power Saving Modes Entered Automatically" (4) describes that a device can change its power state based on usage or other criteria. "Power State Settable" (5) indicates that the SetPowerState method is supported. "Power Cycling Supported" (6) indicates that the SetPowerState method can be invoked with the PowerState input variable set to 5 ("Power Cycle"). "Timed Power On Supported" (7) indicates that the SetPowerState method can be invoked with the PowerState input variable set to 5 ("Power Cycle") and the Time parameter set to a specific date and time, or interval, for power-on.

[0545] Values are: "Unknown", "Not Supported", "Disabled", "Enabled", "Power Saving Modes Entered Automatically", "Power State Settable", "Power Cycling Supported", "Timed Power On Supported"

[0546] boolean PowerManagementSupported

[0547] Access Type: Read-only

[0548] Indicates that the device can be power managed—i.e. can be put into suspend mode, etc. This boolean does not indicate that power management features are currently enabled, only that the logical device is capable of power management.

[0549] string Purpose

[0550] Access Type: Read-only

[0551] A free form string describing the media and/or its use.

[0552] string Status

[0553] Access Type: Read-only

[0554] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror-resilvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[0555] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[0556] Maximum Length: 10

[0557] uint16 StatusInfo

[0558] Access Type: Read-only

[0559] StatusInfo is a string indicating whether the logical device is in an enabled (value=3), disabled (value=4) or some other (1) or unknown (2) state. If this property does not apply to the logical device, the value, 5 ("Not Applicable"), should be used.

Value	Description
1	Other
2	Unknown
3	Enabled
4	Disabled
5	Not Applicable

[0560] string SystemCreationClassName

[0561] Access Type: Read-only

[0562] The scoping system's CreationClassName.

[0563] string SystemName

[0564] Access Type: Read-only

[0565] The scoping system's Name.

[0566] Methods

[0567] The CIM_StorageExtent class supports the following methods:

Method Name	Description
Reset	Requests a reset of the logical device.
SetPowerState	SetPowerState defines the desired power state for a logical device and when a device should be put into that state.

[0568] CIM_System: CIM_LogicalElement

[0569] Abstract Class

[0570] The CIM_System class represents a logical element that aggregates an enumerable set of managed system elements. The aggregation operates as a functional whole. Within any particular subclass of CIM_System, there is a well-defined list of CIM_ManagedSystemElement classes whose instances must be aggregated.

[0571] Properties

[0572] string Caption

[0573] Access Type: Read-only

[0574] The Caption property is a short textual description (one-line string) of the object.

[0575] Maximum Length: 64

[0576] string CreationClassName

[0577] Access Type: Read-only

[0578] The CreationClassName property indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.

[0579] string Description

[0580] Access Type: Read-only

[0581] The Description property provides a textual description of the object.

[0582] datetime InstallDate

[0583] Access Type: Read-only

[0584] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[0585] string Name

[0586] Access Type: Read-only

[0587] The inherited Name property serves as key of a CIM_System instance in an enterprise environment.

[0588] string NameFormat

[0589] Access Type: Read-only

[0590] The CIM_System object and its derivatives are top level objects of CIM. They provide the scope for numerous components. Having unique system keys is required. A heuristic can be defined in individual system subclasses to attempt to always generate the same system name key. The NameFormat property identifies how the system name was generated, using the subclass' heuristic.

[0591] string PrimaryOwnerContact

[0592] Access Type: Read-only

[0593] A string that provides information on how the primary system owner can be reached (e.g. phone number, email address, . . .).

[0594] string PrimaryOwnerName

[0595] Access Type: Read-only

[0596] The name of the primary system owner.

[0597] string Roles []

[0598] Access Type: Read/Write

[0599] A collection of strings that specify the roles this system plays in the IT-environment.

[0600] string Status

[0601] Access Type: Read-only

[0602] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter,

"Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[0603] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[0604] Maximum Length: 10

[0605] B. MSSQL Classes

[0606] MSSQL_BackupDevice: CIM_StorageExtent

[0607] The MSSQL_BackupDevice class represents backup devices known to the SQL Server™ installation.

[0608] Properties

[0609] uint16 Access

[0610] Access Type: Read-only

[0611] The Access property describes whether the media is readable, writeable, or both. Unknown can also be returned.

Value	Description
0	Unknown
1	Readable
2	Writeable
3	Read/Write Supported

[0612] uint16 Availability

[0613] Access Type: Read-only

[0614] The availability and status of the device. For example, the Availability property indicates that the device is running and has full power, or is in a warning, test, degraded or power save state. Regarding the power saving states, these are defined as follows: Power Save—Unknown indicates that the device is known to be in a power save mode, but its exact status in this mode is unknown; Power Save—Low Power Mode indicates that the device is in a power save state but still functioning, and may exhibit degraded performance; Power Save—Standby describes that the device is not functioning but could be brought to full power 'quickly'; and Power Save—Warning indicates that the device is in a warning state, though also in a power save mode.

Value	Description
1	Other
2	Unknown
3	Running/Full Power
4	Warning
5	In Test
6	Not Applicable
7	Power Off
8	Off Line
9	Off Duty
10	Degraded

-continued

Value	Description
11	Not Installed
12	Install Error
13	Power Save - Unknown
14	Power Save - Low Power Mode
15	Power Save - Standby
16	Power Cycle
17	Power Save - Warning

[0615] uint64 BlockSize

[0616] Access Type: Read-only

[0617] Size in bytes of the blocks that form this storage extent. If unknown or if a block concept is not valid (for example, for aggregate extents, memory or logical disks), enter a 1.

[0618] Units: Bytes

[0619] string Caption

[0620] Access Type: Read-only

[0621] The Caption property is a short textual description (one-line string) of the object. Maximum Length: 64

[0622] uint32 ConfigManagerErrorCode

[0623] Access Type: Read-only

[0624] Indicates the Win32 Configuration Manager error code.

Value	Description
0	This device is working properly.
1	This device is not configured correctly.
2	Windows cannot load the driver for this device.
3	The driver for this device might be corrupted, or your system may be running low on memory or other resources.
4	This device is not working properly. One of its drivers or your registry might be corrupted.
5	The driver for this device needs a resource that Windows cannot manage.
6	The boot configuration for this device conflicts with other devices.
7	Cannot filter.
8	The driver loader for the device is missing.
9	This device is not working properly because the controlling firmware is reporting the resources for the device incorrectly.
10	This device cannot start.
11	This device failed.
12	This device cannot find enough free resources that it can use.
13	This device cannot find enough free resources that it can use.
14	Windows cannot verify this device's resources.
15	This device cannot work properly until you restart your computer.
16	This device is not working properly because there is probably a re-enumeration problem.
17	Windows cannot identify all the resources this device uses.
18	This device is asking for an unknown resource type.
19	Reinstall the drivers for this device.
20	Your registry might be corrupted.
21	System failure: Try changing the driver for this device. If that does not work, see your hardware documentation. Windows is removing this device.
22	This device is disabled.

-continued

Value	Description
23	System failure: Try changing the driver for this device. If that doesn't work, see your hardware documentation.
24	This device is not present, is not working properly, or does not have all its drivers installed.
25	Windows is still setting up this device.
26	Windows is still setting up this device.
27	This device does not have valid log configuration.
28	The drivers for this device are not installed.
29	This device is disabled because the firmware of the device did not give it the required resources.
30	This device is using an Interrupt Request (IRQ) resource that another device is using.
31	This device is not working properly because Windows cannot load the drivers required for this device.

[0625] boolean ConfigManagerUserConfig

[0626] Access Type: Read-only

[0627] Indicates whether the device is using a user-defined configuration.

[0628] string CreationClassName

[0629] Access Type: Read-only

[0630] CreationClassName indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.

[0631] string Description

[0632] Access Type: Read-only

[0633] The Description property provides a textual description of the object.

[0634] string DeviceID

[0635] Access Type: Read-only

[0636] DeviceID is an address or other identifying information to uniquely name the logical device.

[0637] boolean ErrorCleared

[0638] Access Type: Read-only

[0639] ErrorCleared is a boolean property indicating that the error reported in LastErrorCode property is now cleared.

[0640] string ErrorDescription

[0641] Access Type: Read-only

[0642] ErrorDescription is a free-form string supplying more information about the error recorded in LastErrorCode property, and information on any corrective actions that may be taken.

[0643] string ErrorMethodology

[0644] Access Type: Read-only

[0645] ErrorMethodology is a free-form string describing the type of error detection and correction supported by this storage extent.

[0646] datetime InstallDate

[0647] Access Type: Read-only

[0648] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[0649] uint32 LastErrorCode

[0650] Access Type: Read-only

[0651] LastErrorCode captures the last error code reported by the logical device.

[0652] [key] string Name

[0653] Access Type: Read-only

[0654] The Name property defines the label by which the object is known.

[0655] uint64 NumberOfBlocks

[0656] Total number of consecutive blocks, each block the size of the value contained in the BlockSize property, which form this storage extent. Total size of the storage extent can be calculated by multiplying the value of the BlockSize property by the value of this property. If the value of BlockSize is 1, this property is the total size of the storage extent.

[0657] string PhysicalLocation

[0658] Access Type: Read/Write

[0659] The PhysicalLocation property indicates the name of the backup device as known to the operating system For example, the string \\Seattle1\Backups\Northwind.bak specifies a server name, directory, and file name for a backup device. The string \\.\TAPE0 specifies a server and a file device, most likely a tape, as a backup device.

[0660] string PNPDeviceID

[0661] Access Type: Read-only

[0662] Indicates the Win32 Plug and Play device ID of the logical device. Example: *PNP03

[0663] uint16 PowerManagementCapabilities []

[0664] Access Type: Read-only

[0665] Indicates the specific power-related capabilities of the logical device. The array values, 0="Unknown", 1="" and 2="Disabled" are self-explanatory. The value, 3="Enabled" indicates that the power management features are currently enabled but the exact feature set is unknown or the information is unavailable. "Power Saving Modes Entered Automatically" (4) describes that a device can change its power state based on usage or other criteria. "Power State Settable" (5) indicates that the SetPowerState method is supported. "Power Cycling Supported" (6) indicates that the SetPowerState method can be invoked with the PowerState input variable set to 5 ("Power Cycle"). "Timed Power On Supported" (7) indicates that the SetPowerState method can be invoked with

the PowerState input variable set to 5 ("Power Cycle") and the Time parameter set to a specific date and time, or interval, for power-on.

Value	Description
0	Unknown
1	Not Supported
2	Disabled
3	Enabled
4	Power Saving Modes Entered Automatically
5	Power State Settable
6	Power Cycling Supported
7	Timed Power On Supported

[0666] boolean PowerManagementSupported

[0667] Access Type: Read-only

[0668] Indicates that the device can be power managed—i.e. can be put into suspend mode, etc. This boolean does not indicate that power management features are currently enabled, only that the logical device is capable of power management.

[0669] string Purpose

[0670] Access Type: Read-only

[0671] A free form string describing the media and/or its use.

[0672] boolean SkipTapeLabel

[0673] Access Type: Read/Write

[0674] The SkipTapeLabel property indicates whether the verification to check that correct media is loaded is performed. If True, media headers are not written. An existing media header is ignored. If False, media headers are honored.

[0675] [key] string SQLServerName

[0676] Access Type: Read-only

[0677] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[0678] Maximum Length: 128

[0679] string Status

[0680] Access Type: Read-only

[0681] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[0682] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[0683] Maximum Length: 10

[0684] uint16 StatusInfo

[0685] Access Type: Read-only

[0686] StatusInfo is an integer value indicating the status of the logical device. If this property does not apply to the logical device, the value, 5 ("Not Applicable"), should be used.

Value	Description
1	Other
2	Unknown
3	Enabled
4	Disabled
5	Not Applicable

[0687] string SystemCreationClassName

[0688] Access Type: Read-only

[0689] The scoping system's CreationClassName.

[0690] string SystemName

[0691] Access Type: Read-only

[0692] The scoping system's name.

[0693] boolean SystemObject

[0694] Access Type: Read-only

[0695] The SystemObject property indicates whether the object is owned by

[0696] Microsoft®. A value of True indicates that the object implementation is owned by Microsoft®.

[0697] uint32 Type

[0698] Access Type: Read-only

[0699] The Type property indicates the type of the backup media.

Value	Description
2	Disk File
3	File On Removable Media In A: Drive
4	File On Removable Media In B: Drive
5	Tape
6	Named Pipe
7	CD-ROM
100	Unknown

[0700] Methods

[0701] The MSSQL_BackupDevice class supports the following methods:

Method Name	Description
ReadBackupHeader	The ReadBackupHeader method returns the header information for the data backed up on the backup device.
ReadMediaHeader	The ReadMediaHeader method returns an object enumerating the media header information for the backup device.
Reset	Requests a reset of the logical device. This method is not implemented for this class.
SetPowerState	SetPowerState defines the desired power state for a logical device and when a device should be put into that state. This method is not implemented for this class.

[0702] Associations

[0703] MSSQL_BackupDevice is associated to MSSQL_SQLServer as the Antecedent property of the MSSQL_SQLServerBackupDevice association.

[0704] MSSQL_BackupHeader

[0705] Abstract Class

[0706] The MSSQL_BackupHeader class represents the contents of the header record for backup content. The instances of this class are returned as results of the ReadBackupHeader method on the MSSQL_BackupDevice class.

[0707] Properties

[0708] string BackupDescription

[0709] The BackupDescription property describes the contents of the backup.

[0710] datetime BackupFinishDate

[0711] The BackupFinishDate property specifies the date and time that the backup operation finished.

[0712] string BackupName

[0713] The BackupName property indicates the name of the backup set.

[0714] Maximum Length: 130

[0715] uint32 BackupSize

[0716] The BackupSize property specifies the size of the backup in bytes.

[0717] Units: Bytes

[0718] datetime BackupStartDate

[0719] The BackupStartDate property specifies the date and time that the backup operation began.

[0720] uint32 BackupType

[0721] The BackupType property indicates the type of the backup content. The content could be a database, transaction log, file or differential database backup.

[0722] Maximum Length: 256

Value	Description
1	Database
2	Transaction Log
4	File
5	Differential Database

[0723] uint32 CheckpointLsn

[0724] The CheckpointLsn property specifies the log sequence number of the most recent checkpoint at the time the backup was created.

[0725] uint32 CodePage

[0726] The CodePage property specifies the server code page or character set used by the server.

[0727] uint32 CompatibilityLevel

[0728] The CompatibilityLevel property specifies the compatibility level setting of the database from which the backup was created.

Value	Description
0	Unknown
60	SQL Server 6.0
65	SQL Server 6.5
70	SQL Server 7.0
80	SQL Server 2000

[0729] boolean Compressed

[0730] The Compressed property indicates whether the backed up data is compressed. SQL Server™ does not currently support software compression; hence the value of this property is always FALSE.

[0731] uint32 DatabaseBackupLsn

[0732] The DatabaseBackupLsn property specifies the log sequence number of the most recent full database backup.

[0733] datetime DatabaseCreationDate

[0734] The DatabaseCreationDate property specifies the date and time the database was created.

[0735] string DatabaseName

[0736] The DatabaseName property specifies the name of the database that was backed up.

[0737] Maximum Length: 130

[0738] uint32 DatabaseVersion

[0739] The DatabaseVersion property specifies the version of the database from which the backup was created.

[0740] uint32 DeviceType

[0741] The DeviceType property specifies the type of device used for the backup operation.

Value	Description
2	Temporary disk device
5	Temporary tape device
6	Temporary named pipe device
7	Temporary virtual device
102	Permanent disk device
105	Permanent tape device
106	Permanent named pipe device
107	Permanent virtual device

[0742] datetime ExpirationDate

[0743] The ExpirationDate property indicates the expiration date for the backup set.

[0744] uint32 FirstLsn

[0745] The firstLsn property specifies the log sequence number of the first transaction in the backup set. This property is NULL for file backups.

[0746] uint32 LastLsn

[0747] The lastLsn property specifies the log sequence number of the last transaction in the backup set. This property is NULL for file backups.

[0748] string MachineName

[0749] The machineName property specifies the name of the computer that performed the backup operation.

[0750] Maximun Length: 130

[0751] uint32 Position

[0752] The Position property indicates the position of the backup set relative to the volume.

[0753] string ServerName

[0754] The ServerName property specifies the name of the server that wrote the backup set.

[0755] Maximum Length: 130

[0756] uint32 SoftwareVendorId

[0757] The SoftwareVendorId property specifies the software vendor identification number. For SQL Server™, this number is 4608.

[0758] uint32 SoftwareVersionBuild

[0759] The SoftwareVersionBuild property specifies the build number of the server that created the backup set.

[0760] uint32 SoftwareVersionMajor

[0761] The SoftwareVersionMajor property specifies the major version number of the server that created the backup set.

[0762] uint32 SoftwareVersionMinor

[0763] The SoftwareVersionMinor property specifies the minor version number of the server that created the backup set.

[0764] uint32 SortOrder

[0765] The SortOrder property specifies the server sort order. This property is only valid for database backups.

[0766] string UserName

[0767] The UserName property specifies the name of the user that performed the backup operation.

[0768] Maximum Length: 130

[0769] MSSQL_BackupSetting: MSSQL_Setting

[0770] The MSSQL_BackupSetting class is used to specify the settings for a backup operation. An instance of this class is passed as an argument to the SQLBackup method on the MSSQL_SQLServer class.

[0771] Properties

[0772] string BackupSetDescription

[0773] Access Type: Read/Write

[0774] The BackupSetDescription property provides descriptive or identifying text for the result of a backup operation. The BackupSetDescription property value is limited to 255 characters. There is no default value.

[0775] Maximum Length: 255

[0776] string BackupSetName

[0777] Access Type: Read/Write

[0778] The BackupSetName property identifies a unit of backup work. The BackupSetName property value is limited to 128 characters.

[0779] Maximum Length: 128

[0780] sint32 BlockSize

[0781] Access Type: Read/Write

[0782] The BlockSize property specifies the formatting size unit for tapes, in bytes, formatted as part of a backup.

[0783] Units: Bytes

[0784] string Caption

[0785] Access Type: Read-only

[0786] A short textual description (one-line string) of the object.

[0787] Maximum Length: 64

[0788] string Database

[0789] Access Type: Read/Write

[0790] The Database identifies the source database for a for backup. The property is a required element and must be set prior to calling the SQLBackup method of the MSSQL_SQLServer class.

[0791] string DatabaseFileGroups[]

[0792] Access Type: Read/Write

[0793] The DatabaseFileGroups property identifies filegroups targeted by a backup or restore

operation. Set to an empty string to reset processing and target the entire database.

[0794] string DatabaseFiles[]

[0795] Access Type: Read/Write

[0796] The DatabaseFiles property identifies operating system files storing table or index data as targets of a backup or restore operation. Set to an empty string to reset processing and target the entire database.

[0797] string Description

[0798] Access Type: Read-only

[0799] A textual description of the object.

[0800] string Device []

[0801] Access Type: Read/Write

[0802] The Device property specifies one or more backup devices used as a database backup target source. Only one medium type can be specified for any backup or restore operation, but multiple media may be specified. Set the Devices property to specify one or more SQL Server backup devices as the backup medium. Specify more than a single database file to stripe the backup operation.

[0803] datetime ExpirationDate

[0804] Access Type: Read/Write

[0805] The ExpirationDate property specifies the last valid date for the backup data.

[0806] boolean FormatMedia

[0807] Access Type: Read/Write

[0808] The FormatMedia property controls tape formatting on a backup operation. The FormatMedia property applies only when the backup medium is tape. The property has no meaning for file or pipe media. If TRUE, the SQL Server™ backup operation attempts to format the tape as an initial step. If FALSE, the SQL Server™ backup operation does not attempt to format the tape.

[0809] boolean Initialize

[0810] Access Type: Read/Write

[0811] The Initialize property controls backup device append/overwrite behavior for a backup to one or more specified devices. If TRUE, the backup specified becomes the first backup set on the media, overwriting any existing backup sets on the media. The backup media is not overwritten if either of the following conditions is met: All backup sets on the media have not yet expired. The optionally specified backup set name does not match the name on the backup media. Specify backup set name with the BackupSetName property. If FALSE, the backup specified creates a new backup set appended as the last backup set on the media.

[0812] string MediaDescription

[0813] Access Type: Read/Write

[0814] The MediaDescription property provides informative text to aid in identification of a backup set. The MediaDescription property is written to a tape media when the media is initialized.

[0815] string MediaName

[0816] Access Type: Read/Write

[0817] The MediaName property provides informative text to aid in identification of a backup set. The MediaName property is are written to a tape media when the media is initialized.

[0818] boolean Restart

[0819] Access Type: Read/Write

[0820] The Restart property controls Backup object behavior when the backup operation was started and interrupted. If TRUE, SQL Server™ attempts to continue processing on a partial backup operation. If FALSE, SQL Server™ restarts an interrupted backup operation at the beginning of the backup set. Set the Restart property only when a user action or system error interrupts backup processing. When using the Restart property, the backup operation specified by the setting must match the originally specified setting in all particulars. Do not set any other properties for the object when setting the Restart property.

[0821] sint32 RetainDays

[0822] Access Type: Read/Write

[0823] The RetainDays property specifies the number of days that must elapse before a backup set can be overwritten.

[0824] [key] string SettingID

[0825] Access Type: Read/Write

[0826] The identifier by which the setting object is known.

[0827] Maximum Length: 256

[0828] boolean SkipTapeHeader

[0829] Access Type: Read/Write

[0830] The SkipTapeHeader property enables or disables backup operation logic that verifies that correct media is loaded. If TRUE, a media name recorded on the media is not checked. The backup set is appended to the media. If FALSE (default), a recorded media name is checked.

[0831] [key] string SQLServerName

[0832] Access Type: Read/Write

[0833] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[0834] Maximum Length: 128

[0835] uint32 TargetType

[0836] Access Type: Read/Write

[0837] The TargetType property controls the type of backup performed against a Microsoft® SQL Server™ database. SQL Server™ can back up an entire database, that portion of a database changed after the last backup, one or more operating system files containing table or index data, or the transaction log of a database. The value of the TargetType property determines applicability and interpretation of related MSSQL_BackupSetting object properties. For example, when TargetType is Files, either the DatabaseFileGroups or DatabaseFiles property must specify filegroups or files to be backed up.

Value	Description	Explanation
0	Database	Backup the entire database.
1	Differential	Back up rows changed after the most recent full database or differential backup.
2	Files	Back up only specified files.
3	Log	Back up only the database transaction log.

[0838] uint32 TruncateLog

[0839] Access Type: Read/Write

[0840] The TruncateLog property controls log file processing during backup operations.

Value	Description	Explanation
0	Truncate	Default. Transaction log is backed up. Records referencing committed transactions are removed.
1	No truncate	Transaction log is backed up. Records referencing committed transactions are not removed, providing a point-in-time image of the log.
2	No Log	Records referencing committed transactions are removed. Transaction log is not backed up.

[0841] boolean UnloadTapeAfter

[0842] Access Type: Read/Write

[0843] The UnloadTapeAfter property controls tape media handling on completion of a backup or restore operation. If TRUE, the tape media in the tape device(s) is rewound and unloaded when the operation completes. If FALSE (default), no attempt is made to rewind and unload the tape media.

[0844] MSSQL_BaseDatatype: CIM_Dependency

[0845] Association Class

[0846] The MSSQL_BaseDatatype class represents an association between a user-defined datatype and the system datatype from which it is derived.

[0847] References

[0848] [key]MSSQL_SystemDatatype Antecedent

[0849] Access Type: Read-only

[0850] The Antecedent property references the system datatype from which the user defined datatype is derived.

[0851] [key]MSSQL_UserDatatype Dependent

[0852] Access Type: Read-only

[0853] The Dependent property references the user-defined datatype that is derived from the base system data type.

[0854] MSSQL_BulkCopySetting: MSSQL_Setting

[0855] The MSSQL_BulkCopySetting class represents the settings for importing or exporting data from a table or a view. It is used as a parameter in the ImportData and ExportData methods in the MSSQL_Table class, and ExportData in the MSSQL_View class.

[0856] Properties

[0857] string Caption

[0858] Access Type: Read-only

[0859] A short textual description (one-line string) of the setting object.

[0860] Maximum Length: 64

[0861] sint32 CodePage

[0862] Access Type: Read/Write

[0863] The CodePage property returns the identifier of the character set used by the referenced Microsoft® SQL Server™ installation or is used to interpret data for a bulk-copy operation.A character set (code page) is used to interpret multibyte character data, determining character value, and therefore sort order. Code page settings apply only to multibyte character data, not to Unicode character data. A code page is chosen for a SQL Server™ installation during setup. By default, bulk-copy operations interpret character data assuming the code page used by the SQL Server™ installation that is either the source or the destination for the copied data.

[0864] string ColumnDelimiter

[0865] Access Type: Read/Write

[0866] The ColumnDelimiter property specifies one or more characters used to delimit a row of data in a bulk copy data file. The ColumnDelimiter property has meaning only when the DataFileType property is set to Special Delimted Char.

[0867] string DataFilePath

[0868] Access Type: Read/Write

[0869] The DataFilePath property indicates the target or source for the bulk copy operation. The operating system file specified by the DataFilePath property is the destination for the data copy performed by the ExportData method of MSSQL_Table and MSSQL_View objects. It is the source file for the ImportData method of MSSQL_Table objects.

[0870] uint32 DataFileType

[0871] Access Type: Read/Write

[0872] Microsoft® SQL Server™ bulk copy operations can copy to or read from files containing data in a number of formats. Use the DataFileType property to indicate the format type of the file desired or in use.

Value	Description	Explanation
1	Comma Delimited Char	Columns are delimited using a comma character.
2	Tab Delimited Char	Columns are delimited using a tab character. Each data row is delimited by a carriage return/linefeed character pair.
3	Special Delimited Char	User-defined by the ColumnDelimiter and RowDelimiter properties.
4	Native Format	SQL Server ™ bulk copy native format. When DataFileType property is Native Format, use the Use6xCompatible property to specify SQL Server ™ version compatibility.
5	Use Format File	Bulk copy uses the file identified in the FormatFilePath property.

[0873] string Description

[0874] Access Type: Read-only

[0875] A textual description of the setting object.

[0876] string ErrorFilePath

[0877] Access Type: Read/Write

[0878] The ErrorFilePath property specifies the full path and full file name of a bulk copy operation error log file. SQL Server™ bulk copy operation logs errors to a file when an error file is provided at time of bulk copy initialization. When errors occur, the bulk copy operation continues to process rows until a maximum number of allowed errors is reached. If that maximum is reached, the error logging file is closed and the bulk copy operation stops. Set the MaximumErrorsBeforeAbort property to set the limiting number of allowed errors in a bulk copy operation.

[0879] boolean ExportWideChar

[0880] Access Type: Read/Write

[0881] The ExportWideChar property controls character set used in the data file when creating a data file by using the ExportData method of the MSSQL_Table and MSSQL_View class. If TRUE, the data file is created as a Unicode text file. If FALSE, the data file is created as a multibyte character text file. The ExportWideChar property is evaluated only when the MSSQL_BulkCopySetting object is used as an argument to the ExportData method, and the bulk-copy operation specifies a character format target file, i.e. the DataFileType property of the MSSQL BulkCopySetting object is 'Comma Delimited Char', 'Special Delimited Char', or 'Tab Delimited Char'.

[0882] sint32 FirstRow

[0883] Access Type: Read/Write

[0884] The FirstRow property is an ordinal value defining the starting point for a bulk data copy When data is copied from SQL Server™ by using the ExportData method of a MSSQL_Table or MSSQL_View object, the FirstRow property indicates the starting row position in the SQL Server™ table. When data is copied to SQL Server™ by using the ImportData method of a MSSQL Table object, the FirstRow property indicates the starting row position in the source data file.

[0885] string FormatFilePath

[0886] Access Type: Read/Write

[0887] The FormatFilePath property indicates the path and file name of a bulk-copy format file. SQL Server™ bulk copy operations can use a user-specified data format stored in a text file. The property has meaning only when the DataFileType property is set to 'Use Format File'.

[0888] sint32 ImportRowsPerBatch

[0889] Access Type: Read/Write

[0890] The lmportRowsPerBatch property specifies the number of rows contained in a bulk copy transaction. The SQL Server™ bulk copy process can copy large amounts of data from an external data file to a SQL Server™ table. By default, a bulk copy data-import operation inserts all rows in the data file in a single transaction. SQL Server™ does not guarantee data integrity until and unless a bulk copy transaction is committed. Use ImportRowsPerBatch to adjust the size of the bulk copy transaction.

[0891] boolean IncludeIdentityValues

[0892] Access Type: Read/Write

[0893] The IncludeIdentityValues property controls the handling of existing values for a column with the SQL Server identity property when data is copied to the SQL Server™ table. When TRUE, the SET IDENTITY_INSERT ON statement is executed when the ImportData method of a MSSQL_Table object is called. When FALSE, any data values present for a column with the identity property are ignored. SQL Server™ generates data values for the column by using the column's setting for identity seed and increment. The default is FALSE.

[0894] sint32 LastRow

[0895] Access Type: Read/Write

[0896] The LastRow property is an ordinal value defining the end point for a bulk data copy.When data is copied from SQL Server™ by using the ExportData method of a MSSQL_Table or

MSSQL_View object, the Is property indicates the end row position in the SQL Server™ table. When data is copied to SQL Server™ by using the ImportData method of a MSSQL Table object, the property indicates the end row position in the source data file. The row will be the last one copied to the SQL Server™ table.

[0897] string LogFilePath

[0898] Access Type: Read/Write

[0899] The LogFilePath property specifies the full operating system path and file name for a bulk copy log file. A bulk copy log file contains statistics describing the number of rows copied and the processing time. It may also contain any non-bulk copy messages received from SQL Server™ during the bulk copy.

[0900] sint32 MaximumErrorsBeforeAbort

[0901] Access Type: Read/Write

[0902] The MaximumErrorsBeforeAbort property specifies the error limit for a bulk copy operation. The default is 10, and a bulk copy operation will stop when ten errors occur. Setting the property to a value greater than 65,535 results in use of the maximum, 65,535. An attempt to set the MaximumErrorsBeforeAbort property to a value less than 1 causes use of the default.

[0903] string RowDelimiter

[0904] Access Type: Read/Write

[0905] The RowDelimiter property specifies a character or character sequence that marks the end of a row in a Microsoft® SQL Server™ bulk copy data file. The RowDelimiter property has meaning only when the DataFileType property is 'Special Delimted Char'.

[0906] uint32 ServerBCPDataFileType

[0907] Access Type: Read/Write

[0908] The ServerBCPDataFileType property specifies the format for an imported data file. The ServerBCPDataFileType property is interpreted only when importing data and when the UseServerSideBCP property of the MSSQL_BulkCopySetting object is TRUE. When ServerBCPDataFileType is set to 'Char', specify a character set by using the CodePage property.

Value	Description	Explanation
1	Char	Read a data file as character data. Interpret the data file using the character set specified. This is the default.
2	Native	Assume bulk copy native data format when reading the data file.
4	Wide Char	Read a data file as Unicode character data.
8	Wide Native	Assume bulk copy wide native data format when reading the data file.

[0909] boolean ServerBCPKeepIdentity

[0910] Access Type: Read/Write

[0911] The ServerBCPKeepIdentity property controls the handling of existing values for a column with the identity property when importing data into the column. When TRUE, NULL is inserted when missing values are encountered in the data file. The default constraint does not supply a value for the column. When FALSE (default), the default constraint provides a value for any missing values encountered in the data file. The ServerBCPKeepNulls property is interpreted only when importing data and when the UseServerSideBCP property of the MSSQL_BulkCopySetting object is TRUE.

[0912] boolean ServerBCPKeepNulls

[0913] Access Type: Read/Write

[0914] The ServerBCPKeepNulls property controls the handling of missing values for all columns accepting NULL and possessing a default value constraint when importing data. When TRUE, NULL is inserted when missing values are encountered in the data file. The default constraint does not supply a value for the column. When FALSE (default), the default constraint provides a value for any missing values encountered in the data file. The ServerBCPKeepNulls property is interpreted only when importing data and when the UseServerSideBCP property of is set to TRUE.

[0915] [key] string SettingID

[0916] Access Type: Read/Write

[0917] The identifier by which the setting object is known.

[0918] Maximum Length: 256

[0919] [key] string SQLServerName

[0920] Access Type: Read/Write

[0921] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[0922] Maximum Length: 128

[0923] boolean SuspendIndexing

[0924] Access Type: Read/Write

[0925] The SuspendIndexing property controls index update when the ImportData method of the MSSQL_Table class is used to copy data to SQL Server™. If TRUE, indexes are dropped before the bulk copy operation is started and re-created after the bulk copy operation is completed. If FALSE, no changes are made to indexing. Note: Indexes that enforce referential or data integrity constraints, such as those implemented by SQL Server™ PRIMARY KEY or UNIQUE key constraints, are not dropped even when SuspendIndexing is TRUE.

[0926] boolean TableLock

[0927] Access Type: Read/Write

[0928] The TableLock property specifies whether to set table-level locking during the execution of a bulk copy import command. If TRUE, the table-level locking is used during the bulk copy import operation. The default is FALSE.

[0929] boolean TruncateLog

[0930] Access Type: Read/Write

[0931] The TruncateLog property controls log file processing upon completion of the ImportData method. If TRUE, the log file is truncated on successful completion of the lmportData method. If FALSE, the log file is not truncated regardless of the completion status of the ImportData method.

[0932] boolean Use6xCompatible

[0933] Access Type: Read/Write

[0934] The Use6xCompatible property controls interpretation of Microsoft® SQL Server™ bulk copy native format data files. A SQL Server™ bulk copy operation either creates or reads from a data file. SQL Server™ bulk copy data files are created in either native (proprietary) or Scharacter format. SQL Server™ bulk copy native data file format has changed for SQL Server™ version 7.0. The user must direct version-dependent handling of source files when processing native format files created by an earlier version of SQL Server™. If TRUE, a bulk copy operation interprets file data based on the pre-SQL Server™ 7.0 format for native data files. If FALSE, default, a bulk copy operation interprets file data based on the SQL Server™ 7.0 format.

[0935] boolean UseBulkCopyOption

[0936] Access Type: Read/Write

[0937] The UseBulkCopyoption property determines whether the select into/bulkcopy option is turned on automatically when the lmportData method of the MSSQL_Table object is executed. If TRUE, and the select into/bulkcopy database option is off in the target database, the option is turned on before an ImportData bulk copy is started and is turned off after the bulk copy is complete. If FALSE, no adjustments to the database options are made. Important: The select into/bulkcopy database option allows non-logged alteration to the target database. A target database should be backed up after any non-logged actions against it.

[0938] boolean UseServerSideBCP

[0939] Access Type: Read/Write

[0940] The UseServerSideBCP property controls the behavior of the bulk copy operation. The operation can be performed using either the bulk copy extensions to ODBC or the Transact-SQL BULK INSERT statement. When TRUE, the row import operation will be implemented by using the BULK INSERT statement. When FALSE (default), the row import or export operation will be implemented using extensions to the SQL Server™ ODBC driver.

[0941] MSSQL_CandidateKey: MSSQL_Key

[0942] Abstract Class

[0943] The MSSQL_CandidateKey class represents a candidate key in a SQL Server table. It consists of a set of columns that can uniquely identify a row in a table.

[0944] Properties

[0945] string Caption

[0946] Access Type: Read-only

[0947] The Caption property is a short textual description (one-line string) of the object.

[0948] Maximum Length: 64

[0949] [key] string DatabaseName

[0950] Access Type: Read-only

[0951] The DatabaseName property indicates the name of the database that the key is a part of.

[0952] Maximum Length: 128

[0953] string Description

[0954] Access Type: Read-only

[0955] The Description property provides a textual description of the object.

[0956] datetime InstallDate

[0957] Access Type: Read-only

[0958] The InstallDate property is a datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[0959] [key] string Name

[0960] Access Type: Read-only

[0961] The Name property defines the label by which the object is known. The name of a key is unique within a database.

[0962] [key] string SQLServerName

[0963] Access Type: Read-only

[0964] The SQLServerName property indicates the name of the SQL Server installation that the key is a part of.

[0965] Maximum Length: 128

[0966] string Status

[0967] Access Type: Read-only

[0968] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational sta-

tuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[0969] Values are: "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[0970] Maximum Length: 10

[0971] [key] string TableName

[0972] Access Type: Read-only

[0973] The TableName property indicates the name of the table that the key is defined in.

[0974] Maximum Length: 128

[0975] Associations

[0976] MSSQL_CandidateKey is associated to MSSQL-_FileGroup as the Dependent property of the MSSQL_Key-FileGroup association.

[0977] MSSQL_CandidateKey is associated to MSSQL-_ForeignKey as the Antecedent property of the MSSQL-_ReferencedKey association.

[0978] MSSQL_CandidateKey is associated to MSSQL-_Database as the Containee property of the MSSQL_DatabaseCandidateKey association.

[0979] MSSQL_Check: MSSQL_Constraint

[0980] The MSSQL_Check class represents the attributes of a SQL Server™ integrity constraint.

[0981] Properties

[0982] string Caption

[0983] Access Type: Read-only

[0984] The Caption property is a short textual description (one-line string) of the object.

[0985] Maximum Length: 64

[0986] boolean Checked

[0987] Access Type: Read-only

[0988] The Checked property enables or disables integrity constraint evaluation for an existing integrity constraint. If TRUE, an attempt is made to enforce an integrity constraint when rows are added to the table on which the constraint is defined. If FALSE, no attempt is made to enforce the integrity constraint when rows are added to the table on which the constraint is defined.

[0989] [key] string DatabaseName

[0990] Access Type: Read-only

[0991] The DatabaseName property indicates the name of the database that the check is a part of.

[0992] Maximum Length: 128

[0993] string Description

[0994] Access Type: Read-only

[0995] The Description property provides a textual description of the object.

[0996] datetime InstallDate

[0997] Access Type: Read-only

[0998] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[0999] [key] string Name

[1000] Access Type: Read-only

[1001] The Name property defines the label by which the object is known.

[1002] [key] string SQLServerName

[1003] Access Type: Read-only

[1004] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[1005] Maximum Length: 128

[1006] string Status

[1007] Access Type: Read-only

[1008] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[1009] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[1010] Maximum Length: 10

[1011] [key] string TableName

[1012] Access Type: Read-only

[1013] The TableName property indicates the name of the table that the check is defined in.

[1014] Maximum Length: 128

[1015] string Text

[1016] Access Type: Read/Write

[1017] The Text property indicates the Transact-SQL or other script that defines the object. Note that there is a special requirement for the name of the object is specified in the CREATE statement. The name of the object has to be in the form that includes the name of the owner. For example, in

order to create an object named "Some Object", one would need to specify it as [dbo].[Some Object].

[1018] Methods

[1019] The MSSQL_Check class supports the following methods:

Method Name	Description
Rename	The Rename method is used to rename a check instance.

[1020] Associations

[1021] MSSQL_Check is associated to MSSQL_Table as the Antecedent property of the MSSQL_TableCheck association.

[1022] MSSQL_Column: MSSQL_DBMSObject

[1023] The MSSQL_Column class represents columns in a SQL Server™ table.

[1024] Properties

[1025] boolean AllowNulls

[1026] Access Type: Read/Write

[1027] The AllowNulls property indicates the ability of a column to accept NULL as a value. If TRUE, the column can accept NULL as a value. If FALSE, NULL is not allowed.

[1028] boolean AnsiPaddingStatus

[1029] Access Type: Read-only

[1030] The AnsiPaddingStatus property returns TRUE if the column is defined to exhibit SQL-92 character padding behavior.

[1031] string Caption

[1032] Access Type: Read-only

[1033] The Caption property is a short textual description (one-line string) of the object.

[1034] Maximum Length: 64

[1035] string Collation

[1036] Access Type: Read/Write

[1037] The Collation property indicates the current collation of a string data type.

[1038] boolean Computed

[1039] Access Type: Read-only

[1040] The Computed property indicates whether the column is computed based on other values in the database.

[1041] string ComputedText

[1042] Access Type: Read/Write

[1043] The ComputedText property indicates Transact-SQL expression used to generate the value of a computed column.

[1044] [key] string DatabaseName

[1045] Access Type: Read-only

[1046] The DatabaseName property indicates the name of the database that the object is a part of.

[1047] Maximum Length: 128

[1048] string Datatype

[1049] Access Type: Read/Write

[1050] The Datatype property indicates the datatype for the column. It has to be the name of a valid system datatype or user defined datatype.

[1051] string Description

[1052] Access Type: Read-only

[1053] The Description property provides a textual description of the object.

[1054] boolean FullTextIndex

[1055] Access Type: Read/Write

[1056] The FullTextIndex property indicates whether the column is participating in Microsoft Search full-text queries. If TRUE, then the column is participating in full-text queries. FullTextIndex must be TRUE in a Table object before any Column object in the Columns collection can be set to TRUE.

[1057] boolean Identity

[1058] Access Type: Read-only

[1059] The Identity property indicates whether the column is the identity column for the table. SQL Server™ allows the row identity property on a single column within a table. Identity, like a primary key, identifies a row uniquely. SQL Server™ implements row identification using a numeric value. As rows are inserted, SQL Server™ generates the row value for an identity column by adding an increment to the existing maximum value.

[1060] sint32 IdentityIncrement

[1061] Access Type: Read-only

[1062] The IdentityIncrement property indicates the value by which the row identity value is incremented when SQL Server™ generates a new identity value. Identity, like a primary key, identifies a row uniquely. SQL Server™ implements row identification using a numeric value. As rows are inserted, SQL Server™ generates the row value for an identity column by adding an increment to the existing maximum value.

[1063] sint32 IdentitySeed

[1064] Access Type: Read-only

[1065] The IdentitySeed property exposes the initial row value for an identity column. Identity, like a primary key, identifies a row uniquely. SQL Server™ implements row identification using a numeric value. As rows are inserted, SQL

Server™ generates the row value for an identity column by adding an increment to the existing maximum value.

[1066] datetime InstallDate

[1067] Access Type: Read-only

[1068] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[1069] sint32 Length

[1070] Access Type: Read/Write

[1071] The Length property indicates the maximum number of characters or bytes accepted by the column. For columns containing character data types, such as char and nchar, interpret the Length property as a number of characters. For columns containing binary data types, such as varbinary, interpret the Length property as a number of bytes.

[1072] [key] string Name

[1073] Access Type: Read-only

[1074] The Name property defines the label by which the object is known.

[1075] boolean NotForRepl

[1076] Access Type: Read-only

[1077] The NotForRepl property enables or disables an identity constraint for data inserted by a replication process. If TRUE, the identity constraint is not enforced when data is added to the table by a known replication login. The replication process provides identity values. If FALSE, the identity constraint is enforced regardless of the source of the data.

[1078] sint32 NumericPrecision

[1079] Access Type: Read/Write

[1080] The NumericPrecision property specifies the maximum number of digits in a fixed-precision, numeric data type.

[1081] sint32 NumericScale

[1082] Access Type: Read/Write

[1083] The NumericScale property specifies the number of digits to the right of the decimal point in a fixed-precision, numeric data type.

[1084] boolean RowGuidCol

[1085] Access Type: Read/Write

[1086] The RowGuidCol indicates whether the column contains the globally unique identifier GUID for rows in table

[1087] [key] string SQLServerName

[1088] Access Type: Read-only

[1089] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[1090] Maximum Length: 128

[1091] string Status

[1092] Access Type: Read-only

[1093] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[1094] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[1095] Maximum Length: 10

[1096] [key] string TableName

[1097] Access Type: Read-only

[1098] The TableName property indicates the name of the table that the key is defined in.

[1099] Maximum Length: 128

[1100] Methods

[1101] The MSSQL_Column class supports the following methods:

Method Name	Description
Rename	The Rename method is used to rename a column instance.
UpdateStatisticsWith	The UpdateStatisticsWith method forces data distribution statistics update for a hypothetical index used to support data distribution statistics for the column.

[1102] Associations

[1103] MSSQL_Column is associated to MSSQL_Table as the PartComponent property of the MSSQL TableColumn association.

[1104] MSSQL_Column is associated to MSSQL_DRIDefault as the Dependent property of the MSSQL_ColumnDRIDefault association.

[1105] MSSQL_Column is associated to MSSQL_Datatype as the Dependent property of the MSSQL_ColumnDatatype association.

[1106] MSSQL_Column is associated to MSSQL_Index as the Antecedent property of the MSSQL IndexColumn association.

[1107] MSSQL_Column is associated to MSSQL_Key as the PartComponent property of the MSSQL_KeyColumn association.

[1108] MSSQL_Column is associated to MSSQL_Default as the Dependent property of the MSSQL_ColumnDefault association.

[1109] MSSQL_Column is associated to MSSQL_Rule as the Dependent property of the MSSQL_ColumnRule association.

[1110] MSSQL_ColumnDefault: CIM_Dependency

[1111] Association Class

[1112] The MSSQL_ColumnDefault class associates a column to the default for the column.

[1113] References

[1114] [key]MSSQL_Default Antecedent

[1115] Access Type: Read-only

[1116] The Antecedent property references the default that applies to a column in the database.

[1117] [key]MSSQL_Column Dependent

[1118] Access Type: Read-only

[1119] The Dependent property references a column in the database.

[1120] MSSQL_ColumnDRIDefault: CIM_Dependency

[1121] Association Class

[1122] The MSSQL_ColumnDRIDefault class associates a column to a DRI default.

[1123] References

[1124] [key]MSSQL_DRIDefault Antecedent

[1125] Access Type: Read-only

[1126] The Antecedent property references a DRI Default defined for the column.

[1127] [key]MSSQL_Column Dependent

[1128] Access Type: Read-only

[1129] The Dependent property references a column in the database.

[1130] MSSQL_ColumnRule: CIM_Dependency

[1131] Association Class

[1132] The MSSQL_ColumnRule class represents an association between a column and a rule bound to the column.

[1133] References

[1134] [key]MSSQL_Rule Antecedent

[1135] Access Type: Read-only

[1136] The Antecedent property references a rule constraint defined in the database.

[1137] [key]MSSQL_Column Dependent

[1138] Access Type: Read-only

[1139] The Dependent property references a column in the database.

[1140] MSSQL_ColumnDatatype: CIM_Dependency

[1141] Association Class

[1142] The MSSQL_ColumnDatatype class associates a column its data type.

[1143] References

[1144] [key]MSSQL_Datatype Antecedent

[1145] Access Type: Read-only

[1146] The Antecedent property references the data type of the column.

[1147] [key]MSSQL_Column Dependent

[1148] Access Type: Read-only

[1149] The Dependent property references a column in the database.

[1150] MSSQL_ConrigValue: MSSQL_Setting

[1151] The MSSQL_ConfigValue class represents the SQL Server™ configuration values. Some SQL Server™ configuration options do not take effect until the SQL Server™ service has been stopped and restarted. You can force the server to immediately accept changes in some options by using the ReconfigureWithOverride method. The DynamicReconfigure property indicates whether the ConfigValue object requires a restart.

[1152] Properties

[1153] string Caption

[1154] Access Type: Read-only

[1155] A short textual description (one-line string) of the object.

[1156] Maximum Length: 64

[1157] sint32 CurrentValue

[1158] Access Type: Read/Write

[1159] The CurrentValue property specifies the current configuration parameter value. The MinimumValue and MaximumValue properties provide the range of values acceptable for the CurrentValue property.

[1160] string Description

[1161] Access Type: Read-only

[1162] The Description property returns a text description of the configuration value.

[1163] boolean DynamicReconfigure

[1164] Access Type: Read-only

[1165] The DynamicReconfigure property indicates modifiability of the configuration value. If TRUE, a modification to the value is effective immediately. If FALSE, modifications are visible only after the SQL Server™ service has been stopped and restarted.

[1166] uint32 ID

[1167] Access Type: Read-only

[1168] The Id property returns a unique identifier for the configuration value. The ID is assigned by SQL Server™.

Value	Description
101	Recovery Interval
102	Allow Updates
103	User Connections
106	Locks
107	Open Objects
109	Fill Factor
115	Nested Triggers
117	Remote Access
124	Default Language
125	Language In Cache
502	Max Async I/O
503	Max Worker Threads
505	Network Packet Size
518	Show Advanced Option
542	Remote Proc Trans
543	Remote Conn Timeout
1110	Time Slice
1123	Default Sort order Id
1124	Unicode Local ID
1125	Unicode Comparison Style
1126	Language Neutral
1127	Two Digit Year Cutoff
1505	Index Create Mem
1514	Spin Counter
1517	Priority Boost
1519	Remote Login Timeout
1520	Remote Query Timeout
1531	Cursor Threshold
1532	Set Working Set Size
1533	Resource Timeout
1534	User Options
1535	Processor Affinity Mask
1536	Max Text Repl Size
1537	Media Retention
1538	Cost Threshold For Parallelism
1539	Max Degree Of Parallelism
1540	Min Memory Per Query
1541	Query Wait
1542	VLM Size
1543	Min Memory
1544	Max Memory
1545	Query Max Time
1546	Lightweight Pooling

[1169] sint32 MaximumValue

[1170] Access Type: Read-only

[1171] The MaximumValue property specifies an upper bound for a configuration value.

[1172] sint32 MinimumValue

[1173] Access Type: Read-only

[1174] The MinimumValue property specifies a lower bound for a configuration value.

[1175] sint32 RunningValue

[1176] Access Type: Read-only

[1177] The RunningValue property returns the setting used by SQL Server™ for the configuration option. Prior to changing a configurable SQL Server™ operating setting, the RunningValue and CurrentValue properties are identical for the ConfigValue object referencing that setting. A change is made to the setting by using the CurrentValue property, and the values will vary as changes are applied.

[1178] [key] string SettingID

[1179] Access Type: Read-only

[1180] The identifier by which the setting object is known.

[1181] Maximum Length: 256

[1182] [key] string SQLServerName

[1183] Access Type: Read-only

[1184] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[1185] Maximum Length: 128

[1186] Associations

[1187] MSSQL_ConfigValue is associated to MSSQL_SQLServer as the Setting property of the MSSQL_SQLServerConfigValue association.

[1188] MSSQL_Constraint: MSSQL_DBMSObject

[1189] Abstract Class

[1190] The MSSQL_Constraint class represents constraints defined in the Microsoft SQL Server database. There are three types of constraints that can be defined—checks, keys and rules.

[1191] Properties

[1192] string Caption

[1193] Access Type: Read-only

[1194] The Caption property is a short textual description (one-line string) of the object.

[1195] Maximum Length: 64

[1196] string Description

[1197] Access Type: Read-only

[1198] datetime InstallDate

[1199] Access Type: Read-only

[1200] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[1201] string Name

[1202] Access Type: Read-only

[1203] The Name property defines the label by which the object is known. When subclassed, the Name property can be overridden to be a Key property.

[1204] string Status

[1205] Access Type: Read-only

[1206] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined.

Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[1207] Values are: "OK","Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[1208] Maximum Length: 10

[1209] MSSQL_Containment

[1210] Abstract Class

[1211] Association Class

[1212] The MSSQL_Containment class represents an association between a container and the contained object.

[1213] References

[1214] CIM_ManagedSystemElement Containee

[1215] Access Type: Read-only

[1216] The Containee property references a managed system element that is contained within another managed system element.

[1217] CIM_ManagedSystemElement Container

[1218] Access Type: Read-only

[1219] The Container property references a managed system element that contains one or more other managed system elements.

[1220] MSSQL_Database: MSSQL_DBMSObject

[1221] The MSSQL_Database class represents instances of SQL Server™ databases.

[1222] Properties

[1223] string Caption

[1224] Access Type: Read-only

[1225] The Caption property is a short textual description (one-line string) of the object.

[1226] Maximum Length: 64

[1227] string Collation

[1228] Access Type: Read/Write

[1229] The Collation property specifies the column-level collation of a string datatype in the database. A Collation setting for a Database object overrides the default collation specified in the model database. All tables in the database then inherit the Collation setting.

[1230] uint32 CompatibilityLevel

[1231] Access Type: Read/Write

[1232] The CompatibilityLevel property controls Microsoft® SQL Server™ behavior, setting behavior to match either the current or earlier version (default is 70).

Value	Description
0	Unknown
60	SQL Server 6.0
65	SQL Server 6.5
70	SQL Server 7.0
80	SQL Server 2000

[1233] datetime CreateDate

[1234] Access Type: Read-only

[1235] The CreateDate property indicates the time and date on which the database was created. Note that creation date may be different from the install date in cases where the object is created in one place and then installed elsewhere.

[1236] boolean CreateForAttach

[1237] Access Type: Read-only

[1238] The CreateForAttach property controls database file creation and specifies whether a database is attached from an existing set of operating system files.

[1239] uint32 DatabaseStatus

[1240] Access Type: Read-only

[1241] The DatabaseStatus property reflects the current operational status on the database. The database is inaccessible when the status is Loading, Offline, Recovering or Suspect.

Value	Description
0	Normal
32	Loading
192	Recovering
256	Suspect
512	Offline
1024	Standby
32768	Emergency Mode

[1242] string Description

[1243] Access Type: Read-only

[1244] The Description property provides a textual description of the object.

[1245] boolean FullTextEnabled

[1246] Access Type: Read-only

[1247] The FullTextEnabled property is TRUE when the referenced database has been selected for participation in Microsoft® Search full-text queries.

[1248] datetime InstallDate

[1249] Access Type: Read-only

[1250] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[1251] [key] string Name

[1252] Access Type: Read-only

[1253] The Name property defines the label by which the object is known.

[1254] string PrimaryFilePath

[1255] Access Type: Read-only

[1256] The PrimaryFilePath property returns the path and name of the operating system directory containing the primary file for the database.

[1257] sint32 Size

[1258] Access Type: Read-only

[1259] The Size property exposes the total size, in megabytes, of the database.

[1260] Units: MegaBytes

[1261] sint32 SpaceAvailable

[1262] Access Type: Read-only

[1263] The SpaceAvailable property returns the amount of disk resource allocated in kilobytes and unused in operating system files implementing Microsoft® SQL Server™ database storage.

[1264] [key] string SQLServerName

[1265] Access Type: Read-only

[1266] The SQLServerName property indicates the name of the SQL Server™ installation that the database is a part of.

[1267] Maximum Length: 128

[1268] string Status

[1269] Access Type: Read-only

[1270] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[1271] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[1272] Maximum Length: 10

[1273] boolean SystemObject

[1274] Access Type: Read-only

[1275] The SystemObject property indicates whether the object is owned by Microsoft®. A value of TRUE indicates that the object implementation is owned by Microsoft®.

[1276] sint32 Version

[1277] Access Type: Read-only

[1278] The Version property returns a system-specified integer identifying the version of Microsoft® SQL Server™ used to create the referenced database.

[1279] Methods

[1280] The MSSQL_Database class supports the following methods:

Method Name	Description
CheckAllocations	The CheckAllocations method scans all pages of the referenced Microsoft ® SQL Server ™ database, testing pages to ensure integrity.
CheckCatalog	The CheckCatalog method tests the integrity of the catalog of the referenced database.
CheckIdentityValues	The CheckIdentityValues method verifies the integrity of all identity columns in tables of the referenced database.
Checkpoint	The Checkpoint method forces a write of dirty database pages.
CheckTables	The CheckTables method tests the integrity of database pages implementing storage for all tables and indexes defined on the tables of the database.
CheckTablesDataOnly	The CheckTablesDataOnly method tests the integrity of database pages implementing storage for all tables in the referenced database.
Create	The Create method is used to create a new database.
DisableFullTextCatalogs	The DisableFullTextCatalogs method suspends Microsoft ® Search full-text catalog maintenance on the database.
EnableFullTextCatalogs	The EnableFullTextCatalogs method enables Microsoft ® Search full-text indexing on the referenced Microsoft ® SQL Server ™ database.
EnumerateStoredProcedures	The EnumerateStoredProcedures method searches stored procedures and returns those that contain a specified string.
ExecuteImmediate	The ExecuteImmediate method allows any SQL Server ™ command to be executed, as long as the command doesn't return result sets.
FullTextIndexScript	The FullTextIndexScript method returns a Transact-SQL command batch enabling Microsoft ® Search full-text indexing on a database or table.
IsValidKeyDatatype	The IsValidKeyDatatype method returns TRUE when the data type specified can participate in a PRIMARY KEY or FOREIGN KEY constraint.

-continued

Method Name	Description
RecalcSpaceUsage	The RecalcSpaceUsage method forces the update of data reporting the disk resource usage of the referenced Microsoft ® SQL Server ™ database.
RemoveFullTextCatalogs	The RemoveFullTextCatalogs method drops all Microsoft ® Search full-text catalogs supporting full-text query on a Microsoft ® SQL Server ™ database.
Rename	The Rename method is used to rename the database instance.
Shrink	The Shrink method attempts to reduce the size of all operating system files maintaining the database.
Transfer	The Transfer method copies database schema and/or data from one Microsoft ® SQL Server ™ database to another.
UpdateIndexStatistics	The UpdateIndexStatistics method forces data distribution statistics update for all indexes on user-defined tables in the referenced Microsoft ® SQL Server ™ database.

[1281] Associations

[1282] MSSQL_Database is associated to MSSQL_SQLServer as the PartComponent property of the MSSQL_SQLServerDatabase association.

[1283] MSSQL_Database is associated to MSSQL_DatabaseSetting as the Element property of the MSSQL_DatabaseDatabaseSetting association.

[1284] MSSQL_Database is associated to MSSQL_FileGroup as the Dependent property of the MSSQL_DatabaseFileGroup association.

[1285] MSSQL_Database is associated to MSSQL_DatabaseRole as the ScopingElement property of the MSSQL_DatabaseDatabaseRole association.

[1286] MSSQL_Database is associated to MSSQL_Table as the GroupComponent property of the MSSQL_DatabaseTable association.

[1287] MSSQL_Database is associated to MSSQL_StoredProcedure as the ScopingElement property of the MSSQL_DatabaseStoredProcedure association.

[1288] MSSQL_Database is associated to MSSQL_User as the ScopingElement property of the MSSQL_DatabaseUser association.

[1289] MSSQL_Database is associated to MSSQL_View as the GroupComponent property of the MSSQL_DatabaseView association.

[1290] MSSQL_Database is associated to MSSQL_Datatype as the ScopingElement property of the MSSQL_DatabaseDatatype association.

[1291] MSSQL_Database is associated to MSSQL_Login as the Dependent property of the MSSQL_LoginDefaultDatabase association.

[1292] MSSQL_Database is associated to MSSQL_Login as the Dependent property of the MSSQL_DatabaseOwnerLogin association.

[1293] MSSQL_Database is associated to MSSQL_User as the Element property of the MSSQL_UserDatabasePermission association.

[1294] MSSQL_Database is associated to MSSQL_DatabaseRole as the Element property of the MSSQL_DatabaseRoleDatabasePermission association.

[1295] MSSQL_Database is associated to MSSQL_UserDefinedFunction as the ScopingElement property of the MSSQL_DatabaseUserDefinedFunction association.

[1296] MSSQL_Database is associated to MSSQL_Rule as the ScopingElement property of the MSSQL_DatabaseRule association.

[1297] MSSQL_Database is associated to MSSQL_Default as the ScopingElement property of the MSSQL_DatabaseDefault association.

[1298] MSSQL_Database is associated to MSSQLCandidateKey as the Container property of the MSSQL_DatabaseCandidateKey association.

[1299] MSSQL_Database is associated to MSSQL_Login as the Container property of the MSSQL_DatabaseLogin association.

[1300] MSSQL_Database is associated to MSSQL_TransactionLog as the Dependent property of the MSSQL_DatabaseTransactionLog association.

[1301] MSSQL_DatabaseCandidateKey: MSSQL_Containment

[1302] Association Class

[1303] The MSSQL_DatabaseCandidateKey class represents an association between a database and a candidate key that is present in one of the tables in the database. This association allows an application to perform a single traversal to find the candidate keys in a database.

[1304] References

[1305] [key]MSSQL_CandidateKey Containee

[1306] Access Type: Read-only

[1307] The Containee property references a candidate key that is contained within the database.

[1308] [key]MSSQL_Database Container

[1309] Access Type: Read-only

[1310] The Container property references a database that contains the candidate key referenced by the Containee property.

[1311] MSSQL_DatabaseDatabaseRole: MSSQL_Scope

[1312] Association Class

[1313] The MSSQL_DatabaseDatabaseRole class associates database role to the database within which the role is defined.

[1314] References

[1315] [key]MSSQL_DatabaseRole ScopedElement

[1316] Access Type: Read-only

[1317] The ScopedElement property references a database role that is defined within the scope of the database referenced by the ScopingElement property.

[1318] [key]MSSQL_Database ScopingElement

[1319] Access Type: Read-only

[1320] The ScopingElement property references a database in SQL.

[1321] MSSQL_DatabaseDatabaseSetting: CIM_ElementSetting

[1322] Association Class

[1323] The MSSQL_DatabaseDatabaseSetting class associates a SQL Server database to an instance of the MSSQL_DatabaseSetting class that contains the settings for the database.

[1324] References

[1325] [key]MSSQL_Database Element

[1326] Access Type: Read-only

[1327] The Element property references a SQL Server database.

[1328] [key]MSSQL_DatabaseSetting Setting

[1329] Access Type: Read-only

[1330] The Setting property references a class that represents the settings for a database.

[1331] MSSQL_DatabaseDatatype: MSSQL_Scope

[1332] Association Class

[1333] The MSSQL_DatabaseDatatype class associates a database to the datatypes defined within the database.

[1334] References

[1335] [key]MSSQL_Datatype ScopedElement

[1336] Access Type: Read-only

[1337] The ScopedElement property references the data type that has been defined within the database.

[1338] [key]MSSQL_Database ScopingElement

[1339] Access Type: Read-only

[1340] The ScopingElement property references the database within which the data type has been defined.

[1341] MSSQL_DatabaseDefault: MSSQL_Scope

[1342] Association Class

[1343] The MSSQL_DatabaseDefault association associates a database to the defaults defined within the database.

[1344] References

[1345] [key]MSSQL_Default ScopedElement

[1346] Access Type: Read-only

[1347] The ScopedElement property references a default constraint defined within the database.

[1348] [key]MSSQL_Database ScopingElement

[1349] Access Type: Read-only

[1350] The ScopingElement property references a SQL Server™ database.

[1351] MSSQL_DatabaseFile: MSSQL_Extension

[1352] The MSSQL_DatabaseFile class is an extension to the CIM_DataFile class. It contains properties that are relevant to an operating system file that is also a file storing SQL Server™ database data.

[1353] Properties

[1354] [key] string DatabaseName

[1355] Access Type: Read-only

[1356] The DatabaseName property indicates the name of the database that the object is a part of.

[1357] Maximum Length: 128

[1358] [key] string FileGroupName

[1359] Access Type: Read-only

[1360] The FileGroupName property indicates the name of the database file group that the database file is a part of.

[1361] Maximum Length: 128

[1362] sint32 FileGrowth

[1363] Access Type: Read/Write

[1364] The FileGrowth property indicates the growth increment of the operating system file used to store table, index, or log data. When FileGrowthType is Megabytes, the FileGrowth value represents the number of megabytes of disk space to allocate for incremental file growth. When FileGrowthType is Percent, the value represents a percentage and must be in the range from 1 through 100.

[1365] real32 FileGrowthInKB

[1366] Access Type: Read-only

[1367] The FileGrowthInKB property reports the number of kilobytes of disk space allocated when an incremental increase occurs on an operating system file.

[1368] uint32 FileGrowthType

[1369] Access Type: Read/Write

[1370] The FileGrowthType property indicates the method of incremental allocation applied when an operating system file is extended.

Value	Description
0	Megabyte
1	Percent
99	Invalid

[1371] sint32 MaximumSize

[1372] Access Type: Read/Nrite

[1373] The MaximumSize property indicates an upper limit for the size of an operating system file containing table and index data, or maintaining a database transaction log.

[1374] [key] string Name

[1375] Access Type: Read-only

[1376] The Name property defines the label by which the object is known.

[1377] Maximum Length: 128

[1378] string PhysicalName

[1379] Access Type: Read-only

[1380] The PhysicalName property specifies the path and file name of the operating system file storing Microsoft® SQL Server™ database or transaction log data.

[1381] boolean PrimaryFile

[1382] Access Type: Read-only

[1383] The PrimaryFile property indicates whether the database file is the one that maintains the database-specific system tables. A SQL Server™ database can have at most one primary file.

[1384] sint32 SpaceAvailableInMB

[1385] Access Type: Read-only

[1386] The SpaceAvailableInMB property returns the amount of disk resource, in megabytes, allocated and unused in operating system files implementing Microsoft® SQL Server™ database and database transaction log storage.

[1387] Units: Megabytes

[1388] [key] string SQLServerName

[1389] Access Type: Read-only

[1390] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[1391] Maximum Length: 128

[1392] Methods

[1393] The MSSQL_DatabaseFile class supports the following methods:

Method Name	Description
Shrink	The Shrink method attempts to reduce the size of the database file.

[1394] MSSQL_DatabaseFileDataFile: MSSQL_Extends

[1395] Association Class

[1396] The MSSQL_DatabaseFileDataFile class associates a CIM_DataFile class to the MSSQL_DatabaseFile class that contains database file specific properties of an operating system file.

[1397] [key]CIM_DataFile ExtendedElement

[1398] Access Type: Read-only

[1399] The ExtendedElement property references an operating system file.

[1400] [key]MSSQL_DatabaseFile Extension

[1401] Access Type: Read-only

[1402] The Extension property references an instance that contains database file specific extensions to an operating system file.

[1403] MSSQL_DatabaseFileGroup: CIM_Dependency

[1404] Association Class

[1405] The MSSQL_DatabaseFileGroup class represents an association between a database and the file group that contains the operating system files that store the data for the database.

[1406] References

[1407] [key]MSSQL_FileGroup Antecedent

[1408] Access Type: Read-only

[1409] The Antecedent property references the file group that contains the operating system files of the database.

[1410] [key]MSSQL_Database Dependent

[1411] Access Type: Read-only

[1412] The Dependent property references a SQL Server™ database.

[1413] MSSQL_DatabaseFullTextCatalog: CIM_Dependency

[1414] Association Class

[1415] The MSSQL_DatabaseFullTextCatalog class represents an association between a database and a full-text catalog that stores index data used for full-text queries against the database.

[1416] References

[1417] [key]MSSQL_FullTextCatalog Antecedent

[1418] Access Type: Read-only

[1419] The Antecedent property references a full-text catalog that stores index data used for full-text queries against the database referenced by the Dependent property.

[1420] [key]MSSQL_Database Dependent

[1421] Access Type: Read-only

[1422] The Dependent property references a Microsoft® SQL Server™ database.

[1423] MSSQL_DatabaseLogin: MSSQL_Containment

[1424] Association Class

[1425] The MSSQL_Containment class represents an association between a database and a login that is mapped to a user defined in the database. This association allows an application to perform a single traversal to find the logins used by a database.

[1426] Properties

[1427] [key] string UserName

[1428] Access Type: Read-only

[1429] This property indicates the database user that is mapped to the login.

[1430] Maximum Length: 128

[1431] References

[1432] [key]MSSQL_Login Containee

[1433] Access Type: Read-only

[1434] The Containee property references a managed system element that is contained within another managed system element.

[1435] [key]MSSQL_Database Container

[1436] Access Type: Read-only

[1437] The Container property references a managed system element that contains one or more other managed system elements.

[1438] MSSQL_DatabaseOwnerLogin: CIM_Dependency

[1439] Association Class

[1440] The MSSQL_DatabaseOwnerLogin class represents an association between a database and the login mapped to the user that owns the database.

[1441] References

[1442] [key]MSSQL_Login Antecedent

[1443] Access Type: Read-only

[1444] The Antecedent property references the login mapped to the user that owns the database referenced by the Dependent property.

[1445] [key]MSSQL_Database Dependent

[1446] Access Type: Read-only

[1447] The Dependent property references a SQL Server database.

[1448] MSSQL_DatabaseRole: MSSQL_Role

[1449] The DatabaseRole object represents the properties of a SQL Server™ database role. SQL Server™ database roles establish groups of users with similar security attributes. Database permissions can be granted by role, simplifying database security planning and administration.

[1450] Properties

[1451] boolean AppRole

[1452] Access Type: Read-only

[1453] The AppRole property indicates the whether the database role has been defined specifically for use by client applications.

[1454] string Caption

[1455] Access Type: Read-only

[1456] The Caption property is a short textual description (one-line string) of the object.

[1457] Maximum Length: 64

[1458] [key] string DatabaseName

[1459] Access Type: Read-only

[1460] The DatabaseName property indicates the name of the database that the object is a part of.

[1461] Maximum Length: 128

[1462] string Description

[1463] Access Type: Read-only

[1464] The Description property provides a textual description of the object.

[1465] boolean FixedRole

[1466] Access Type: Read-only

[1467] The FixedRole property returns True when the database role referenced is system-defined.

[1468] datetime InstallDate

[1469] Access Type: Read-only

[1470] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[1471] [key] string Name

[1472] Access Type: Read-only

[1473] The Name property indicates a label by which the role can be identified.

[1474] string Password

[1475] Access Type: Write-only

[1476] The Password property is used to set the password for an application role. The property cannot be read. It can be written at the time the database role object is created.

[1477] [key] string SQLServerName

[1478] Access Type: Read-only

[1479] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[1480] Maximum Length: 128

[1481] string Status

[1482] Access Type: Read-only

[1483] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational sta-

tuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror-resilvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[1484] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[1485] Maximum Length: 10

[1486] Associations

[1487] MSSQL_DatabaseRole is associated to MSSQL_Database as the ScopedElement property of the MSSQL_DatabaseDatabaseRole association.

[1488] MSSQL_DatabaseRole is associated to MSSQL_User as the Antecedent property of the MSSQL_MemberUser association.

[1489] MSSQL_DatabaseRole is associated to MSSQL_UserDefinedFunction as the Grantee property of the MSSQL_DatabaseRoleUserDefinedFunctionPermission association.

[1490] MSSQL_DatabaseRole is associated to MSSQL_Database as the Grantee property of the MSSQL_DatabaseRoleDatabasePermission association.

[1491] MSSQL_DatabaseRole is associated to MSSQL_StoredProcedure as the Grantee property of the MSSQL_DatabaseRoleStoredProcedurePermission association.

[1492] MSSQL_DatabaseRole is associated to MSSQL_View as the Grantee property of the MSSQL_DatabaseRoleViewPermission association.

[1493] MSSQL_DatabaseRole is associated to MSSQL_Table as the Grantee property of the MSSQL_DatabaseRoleTablePermission association.

[1494] MSSQL_DatabaseRole is associated to MSSQL_DatabaseRole as the Antecedent property of the MSSQL_MemberDatabaseRole association.

[1495] MSSQL_DatabaseRole is associated to MSSQL_DatabaseRole as the Dependent property of the MSSQL_MemberDatabaseRole association. MSSQL_DatabaseRoleDatabasePermission: MSSQL_Permission

[1496] Association Class

[1497] The MSSQL_DatabaseRoleDatabasePermission class represents the permissions that a database role has for the database in which it is defined. The instances of this class represent only the permission that has been explicitly granted or denied to the user object. For example, if a database role has permissions to access a database by virtue of being a member of another database role, then there will not be a permission association instance between the role and the database.

[1498] Properties

[1499] boolean Granted

[1500] Access Type: Read/Write

[1501] The Granted property indicates whether the permission has been granted or denied. A value of True indicates that the permission has been granted. A value of False indicates that permission has been denied.

[1502] [key] uint32 PrivilegeType

[1503] Access Type: Read/Write

[1504] The PrivilegeType property indicates the type of privilege that has been granted or denied.

Value	Description
0	Unknown
1	Permission to query a table
2	Permission to add rows to a table
4	Permission to update rows of a table
8	Permission to delete rows of a table
16	Permission to execute a stored procedure
32	Permission to grant DRI on a table
63	All privileges applicable to the database object
128	Permission to create and own a table
256	Permission to create and own a database
512	Permission to create and own a view
1024	Permission to create and own a stored procedure
2048	Permission to backup a database
4096	Permission to create a default
8192	Permission to backup a database transaction log
16384	Permission to create a rule
32768	Permission to backup to a table
65366	Permission to create a user defined function
130944	All privileges applicable to the database

[1505] References

[1506] [key]MSSQL_Database Element

[1507] Access Type: Read/Write

[1508] The Element property references a SQL Server™ database.

[1509] [key]MSSQL_DatabaseRole Grantee

[1510] Access Type: Read/Write

[1511] The Grantee property references a database role that has been granted or denied permission to access the database.

[1512] MSSQL_DatabaseRoleStoredProcedurePermission: MSSQL_Permission

[1513] Association Class

[1514] The MSSQL_DatabaseRoleStoredProcedurePermission class represents the permissions that a database role has for a stored procedure. The instances of this class represent only the permission that has been explicitly granted or denied to the user object. For example, if a database role has permissions to access the stored procedure by virtue of being a member of another database role, then there will not be a permission association instance between the role and the stored procedure.

[1515] Properties

[1516] boolean Granted

[1517] Access Type: Read/Write

[1518] The Granted property indicates whether the permission has been granted or denied. A value of True indicates that the permission has been granted. A value of False indicates that permission has been denied.

[1519] [key] uint32 PrivilegeType

[1520] Access Type: Read/Write

[1521] The PrivilegeType property indicates the type of privilege that has been granted or denied.

Value	Description
0	Unknown
1	Permission to query a table
2	Permission to add rows to a table
4	Permission to update rows of a table
8	Permission to delete rows of a table
16	Permission to execute a stored procedure
32	Permission to grant DRI on a table
63	All privileges applicable to the database object
128	Permission to create and own a table
256	Permission to create and own a database
512	Permission to create and own a view
1024	Permission to create and own a stored procedure
2048	Permission to backup a database
4096	Permission to create a default
8192	Permission to backup a database transaction log
16384	Permission to create a rule
32768	Permission to backup to a table
65366	Permission to create a user defined function
130944	All privileges applicable to the database

[1522] References

[1523] [key]MSSQL_StoredProcedure Element

[1524] Access Type: Read/Write

[1525] The Element property references a stored procedure in the database.

[1526] [key]MSSQL_DatabaseRole Grantee

[1527] Access Type: Read/Write

[1528] The Grantee property references a database role for which the permissions have been defined.

[1529] MSSQL_DatabaseRoleTablePermission: MSSQL_Permission

[1530] Association Class

[1531] The MSSQL_DatabaseRoleTablePermission class represents the permissions that a database role has for a table. The instances of this class represent only the permissions that have been explicitly granted or denied to the user object. For example, if a database role has permissions to access the table by virtue of being a member of another database role, then there will not be a permission association instance between the role and the table.

[1532] Properties

[1533] string ColumnName []

[1534] Access Type: Read/Write

[1535] The ColumnName property specifies the columns within the table for which the permission is specified. If this property is null, then the permission applies to all columns in the table, otherwise it applies only to the columns indicated in this property.

[1536] boolean Granted

[1537] Access Type: Read/Write

[1538] The Granted property indicates whether the permission has been granted or denied. A value of True indicates that the permission has been granted. A value of False indicates that permission has been denied.

[1539] [key] uint32 PrivilegeType

[1540] Access Type: Read/Write

[1541] The PrivilegeType property indicates the type of privilege that has been granted or denied.

Value	Description
0	Unknown
1	Permission to query a table
2	Permission to add rows to a table
4	Permission to update rows of a table
8	Permission to delete rows of a table
16	Permission to execute a stored procedure
32	Permission to grant DRI on a table
63	All privileges applicable to the database object
128	Permission to create and own a table
256	Permission to create and own a database
512	Permission to create and own a view
1024	Permission to create and own a stored procedure
2048	Permission to backup a database
4096	Permission to create a default
8192	Permission to backup a database transaction log
16384	Permission to create a rule
32768	Permission to backup to a table
65366	Permission to create a user defined function
130944	All privileges applicable to the database

[1542] References

[1543] [key]MSSQL_Table Element

[1544] Access Type: Read-only

[1545] The Element property references a SQL Server™ table.

[1546] [key]MSSQL_DatabaseRole Grantee

[1547] Access Type: Read-only

[1548] The Grantee property references a data bas e role for which the permissions have been defined.

[1549] MSSQL_DatabaseRoleUserDefined-FunctionPermission: MSSQL_Permission

[1550] Association Class

[1551] The MSSQL_DatabaseRoleUserDefined-FunctionPermission class represents the permissions that a database role has for a table. The instances of this class represent only the permissions that have been explicitly granted or denied to the user object. For example, if a database role has permissions to access the user defined function by virtue of being a member of another database

role, then there will not be a permission association instance between the role and the user defined function.

[1552] Properties

[1553] boolean Granted

[1554] Access Type: Read/Write

[1555] The Granted property indicates whether the permission has been granted or denied. A value of True indicates that the permission has been granted. A value of False indicates that permission has been denied.

[1556] [key] uint32 PrivilegeType

[1557] Access Type: Read/Write

[1558] The PrivilegeType property indicates the type of privilege that has been granted or denied.

Value	Description
0	Unknown
1	Permission to query a table
2	Permission to add rows to a table
4	Permission to update rows of a table
8	Permission to delete rows of a table
16	Permission to execute a stored procedure
32	Permission to grant DRI on a table
63	All privileges applicable to the database object
128	Permission to create and own a table
256	Permission to create and own a database
512	Permission to create and own a view
1024	Permission to create and own a stored procedure
2048	Permission to backup a database
4096	Permission to create a default
8192	Permission to backup a database transaction log
16384	Permission to create a rule
32768	Permission to backup to a table
65366	Permission to create a user defined function
130944	All privileges applicable to the database

[1559] References

[1560] [key]MSSQL_UserDefinedFunction Element

[1561] Access Type: Read/Write

[1562] The Element property references a user-defined function.

[1563] [key]MSSQL_DatabaseRole Grantee

[1564] Access Type: Read/Write

[1565] The Grantee property references a database role for which the permissions have been defined.

[1566] MSSQL_DatabaseRoleViewPermission: MSSQL_Permission

[1567] Association Class

[1568] The MSSQL_DatabaseRoleViewPermission class represents the permissions that a database role has for a view. The instances of this class represent only the permissions that have been explicitly granted or denied to the user object.

[1569] For example, if a database role has permissions to access the view by virtue of being a member of another database role, then there will not be a permission association instance between the role and the view.

[1570] Properties

[1571] string ColumnName []

[1572] Access Type: Read/Write

[1573] The ColumnName property specifies the columns within the view for which the permission is specified. If this property is null, then the permission applies to all columns in the view, otherwise it applies only to the columns indicated in this property.

[1574] boolean Granted

[1575] Access Type: Read/Write

[1576] The Granted property indicates whether the permission has been granted or denied. A value of True indicates that the permission has been granted. A value of False indicates that permission has been denied.

[1577] [key] uint32 PrivilegeType

[1578] Access Type: Read/Write

[1579] The PrivilegeType property indicates the type of privilege that has been granted or denied.

Value	Description
0	Unknown
1	Permission to query a table
2	Permission to add rows to a table
4	Permission to update rows of a table
8	Permission to delete rows of a table
16	Permission to execute a stored procedure
32	Permission to grant DRI on a table
63	All privileges applicable to the database object
128	Permission to create and own a table
256	Permission to create and own a database
512	Permission to create and own a view
1024	Permission to create and own a stored procedure
2048	Permission to backup a database
4096	Permission to create a default
8192	Permission to backup a database transaction log
16384	Permission to create a rule
32768	Permission to backup to a table
65366	Permission to create a user defined function
130944	All privileges applicable to the database

[1580] [key]MSSQL_View Element

[1581] Access Type: Read-only

[1582] The Element property references a SQL Server™ view.

[1583] [key]MSSQL_DatabaseRole Grantee

[1584] Access Type: Read-only

[1585] The Grantee property references a database role for which the permissions have been defined.

[1586] MSSQL_DatabaseRule: MSSQL_Scope

[1587] Association Class

[1588] The MSSQL_DatabaseRule class represents an association between a database and the rules defined within the database.

[1589] References

[1590] [key]MSSQL_Rule ScopedElement

[1591] Access Type: Read-only

[1592] The ScopedElement property references a rule defined in the database referenced by the ScopingElement property.

[1593] [key]MSSQL_Database ScopingElement

[1594] Access Type: Read-only

[1595] The ScopingElement property references a SQL Server™ database.

[1596] MSSQL_DatabaseSetting: MSSQL_Setting

[1597] The MSSQL_DatabaseSetting class represents operational settings for a database.

[1598] Properties

[1599] boolean AssignmentDiag

[1600] Access Type: Read/Write

[1601] The AssignmentDiag property enables SQL-92 standard behavior for NULL in aggregate, data truncation, divide-by-zero, and arithmetic overflow errors.

[1602] boolean AutoClose

[1603] Access Type: Read/Write

[1604] The AutoClose property exposes server behavior for databases not accessed by a user. If TRUE, the database is closed and its resources are freed when no user connection accesses the database. If FALSE, the server maintains the database in an open and ready state regardless of user activity.

[1605] boolean AutoCreateStat

[1606] Access Type: Read/Write

[1607] The AutoCreateStat property exposes Microsoft® SQL Server™ data distribution statistics creation behavior. If TRUE, the optimizer directs automatic creation of supporting data distribution statistics as required. If FALSE, the optimizer does not direct statistics creation.

[1608] boolean AutoShrink

[1609] Access Type: Read/Write

[1610] The AutoShrink property exposes Microsoft® SQL Server™ sizing behavior for operating system files maintaining table and index data. If TRUE, operating system files maintaining table and index data are evaluated for downward resizing when the server periodically checks for unused space. If FALSE, the operating system files storing the database are not evaluated during periodic checks for unused space.

[1611] boolean AutoUpdateStat

[1612] Access Type: Read/Write

[1613] The AutoUpdateStat property exposes Microsoft® SQL Server™ data distribution statistics creation behavior. If TRUE, the optimizer directs automatic rebuild of supporting data distribution statistics as required. If FALSE, the optimizer does not direct statistics rebuild.

[1614] string Caption

[1615] Access Type: Read-only

[1616] A short textual description (one-line string) of the MSSQL_DatabaseSetting object.

[1617] Maximum Length: 64

[1618] boolean ColumnsNullByDefault

[1619] Access Type: Read/Write

[1620] The ColumnsNullByDefault property controls column default value behavior when a table is created in the Microsoft® SQL Server™ database. If TRUE, columns in new tables allow NULL. If FALSE, columns in new tables do not allow NULL.

[1621] boolean CompareNull

[1622] Access Type: Read/Write

[1623] The CompareNull property controls evaluation of NULL for equality. If TRUE, the expression NULL=NULL evaluates as NULL. If FALSE, the expression NULL=NULL evaluates as TRUE.

[1624] boolean ContactNull

[1625] Access Type: Read/Write

[1626] The ContactNull property specifies NULL value handling for catenation. If TRUE, A+NULL, where A is a string, yields NULL. If FALSE, A+NULL, where A is a string, yields A.

[1627] boolean CursorCloseOnCommit

[1628] Access Type: Read/Write

[1629] The CursorCloseOnCommit property specifies cursor behavior when modifications made within a transaction are committed or rolled back. If TRUE (the default) Microsoft® SQL Server™ cursors are closed when an action ends a transaction. If FALSE, cursors remain open after a transaction-ending action.

[1630] real32 DataSpaceUsage

[1631] Access Type: Read-only

[1632] The DataSpaceUsage property indicates the physical disk resource used to maintain the data of a database. The value reflects the amount of space in use and reserved for use. The actual data space used by any given table is reported by the DataSpaceUsed property of the Table object. The value represents an amount in megabytes and is accurate to two decimal places.

[1633] Units: Megabytes

[1634] boolean DBOUseOnly

[1635] Access Type: Read/Write

[1636] The DBOUseOnly property toggles access rights to a Microsoft® SQL Server™ database. If TRUE, only users with database ownership privilege can access the database. If FALSE, any authorized user can access the database.

[1637] boolean DefaultCursor

[1638] Access Type: Read/Write

[1639] The DefaultCursor property controls the visibility of cursors created in Transact-SQL batches. If TRUE, cursors declared in a batch are created with local scope. If FALSE, cursors declared in a batch are created with global scope.

[1640] string Description

[1641] Access Type: Read-only

[1642] A textual description of the MSSQL_DatabaseSetting object.

[1643] real32 IndexSpaceUsage

[1644] Access Type: Read-only

[1645] The IndexSpaceUsage returns the number of kilobytes assigned to index storage within all operating system files maintaining indexes for the referenced database.

[1646] Units: Kilobytes

[1647] boolean Offline

[1648] Access Type: Read/Write

[1649] The Offline property controls Microsoft® SQL Server™ database availability. When TRUE, the database is unavailable, or is being made unavailable, for use by authorized users. When FALSE, the database is online, or is being brought online, for use by authorized users.

[1650] boolean QuoteDelimiter

[1651] Access Type: Read/Write

[1652] The QuoteDelimiter property controls Microsoft® SQL Server™ interpretation of identifier strings in statements submitted for execution. When TRUE, identifiers can be delimited by double quotation marks and character literal values must be delimited by single quotation marks. When FALSE, identifiers cannot be quoted and must follow all Transact-SQL rules for identifiers. For example, character literal values can be delimited by either single or double quotation marks.

[1653] boolean ReadOnly

[1654] Access Type: Read/Write

[1655] The ReadOnly property controls the ability to update a Microsoft® SQL Server™ database. If TRUE, data in the database cannot be changed. If FALSE, updates are allowed to data in the database.

[1656] uint32 RecoveryType

[1657] Access Type: Read/Write

[1658] The RecoveryType property specifies the type of recovery model that a database will use. Simple is the default setting for SQL Server Desktop Edition and the data engine, and 'Full' is the default for all other editions.

Value	Description	Explanation
0	Simple	The database can be recovered only to the last full database backup or last differential backup.
1	Bulk Logged	Logging for all SELECT INTO, CREATE INDEX, and bulk loading data operations is minimal and therefore requires less log space. In exchange for better performance and less log space usage, the risk of exposure to loss is greater than with full recovery.
2	Full	Database backups and transaction log backups are used to provide full recoverability from media failure. All operations, including bulk operations such as SELECT INTO, CREATE INDEX, and bulk loading data, are fully logged.
3	Unknown	The recovery type is not known.

[1659] boolean RecursiveTriggers

[1660] Access Type: Read/Write

[1661] The RecursiveTriggers property controls nested call behavior for Microsoft® SQL Server™ triggers. When TRUE, a trigger may fire more than once when statement execution directs more than a single trigger execution. When FALSE, a trigger will execute only once regardless of the actions of itself or other triggers enabled on other tables.

[1662] boolean SelectIntoBulkCopy

[1663] Access Type: Read/Write

[1664] The SelectIntoBulkCopy property enables non-logged operation on a Microsoft® SQL Server™ database. If TRUE, non-logged operations are allowed. If FALSE, non-logged operations are not allowed.

[1665] [key] string SettingID

[1666] Access Type: Read-only

[1667] The identifier by which the MSSQL_DatabaseSetting object is known.

[1668] Maximum Length: 256

[1669] booleanSingleUser

[1670] Access Type: Read/Write

[1671] The SingleUser property exposes one method of constraining user access to a Microsoft® SQL Server™ database. If TRUE, only one user can access the database at any one time. If FALSE, multiple users can access the database at one time.

[1672] [key] string SQLServerName

[1673] Access Type: Read-only

[1674] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[1675] Maximum Length: 128

[1676] boolean TornPageDetection

[1677] Access Type: Read/Write

[1678] The TornPageDetection property enables Microsoft® SQL Server™ logic-enhancing data security in the event of certain types of system failure. If TRUE, SQL Server marks units of a database page prior to attempting a write and checks page marking on every read. If FALSE, database pages are not marked or evaluated.

[1679] boolean TruncateLogOnCheckpoint

[1680] Access Type: Read/Write

[1681] The TruncateLogOnCheckpoint property configures automatic transaction log maintenance activity. If TRUE, SQL Server removes log entries referencing committed transactions when activity on the database forces a dirty page write. If FALSE, the forced dirty page writes have no effect on the database transaction log.

[1682] Associations

[1683] MSSQL_DatabaseSetting is associated to MSSQL_Database as the Setting property of the MSSQL_DatabaseDatabaseSetting association.

[1684] MSSQL_DatabaseStoredProcedure: MSSQL_Scope

[1685] Association Class

[1686] The MSSQL_DatabaseStoredProcedure class represents an association between the database and a stored procedure defined within the database.

[1687] References

[1688] [key] MSSQL_StoredProcedure ScopedElement

[1689] Access Type: Read-only

[1690] The ScopedElement property references a stored procedure defined within the database referenced by the ScopingElement property.

[1691] [key] MSSQL_Database ScopingElement

[1692] Access Type: Read-only

[1693] The ScopingElement property references a SQL Server™ database.

[1694] MSSQL_DatabaseTable: CIM_Component

[1695] Association Class

[1696] The MSSQL_DatabaseTable class associates a database to all the tables contained in the database.

[1697] References

[1698] [key]MSSQL_Database GroupComponent

[1699] Access Type: Read-only

[1700] The parent element in the association

[1701] [key]MSSQL_Table PartComponent

[1702] Access Type: Read-only

[1703] The child element in the association

[1704] MSSQL_DatabaseTransactionLog: CIM_Dependency

[1705] Association Class

[1706] The MSSQL_DatabaseTransactionLog class represents an association between the database and the transaction log for the database.

[1707] References

[1708] [key]MSSQL_TransactionLog Antecedent

[1709] Access Type: Read-only

[1710] Antecedent represents the independent object in this association.

[1711] [key]MSSQL_Database Dependent

[1712] Access Type: Read-only

[1713] Dependent represents the object dependent on the Antecedent.

[1714] MSSQL_DatabaseUser: MSSQL_Scope

[1715] Association Class

[1716] The MSSQL_DatabaseUser class represents an association between a database and a user defined for the database.

[1717] References

[1718] [key]MSSQL_User ScopedElement

[1719] Access Type: Read-only

[1720] The ScopedElement property references a user defined within the database referenced by the ScopingElement property.

[1721] [key]MSSQL_Database ScopingElement

[1722] Access Type: Read-only

[1723] The ScopingElement property references a database in SQL Server™.

[1724] MSSQL_DatabaseUserDefinedFunction MSSQL_Scope

[1725] Association Class

[1726] The MSSQL_DatabaseUserDefinedFunction class represents an association between a database and a user-defined function defined within the database.

[1727] References

[1728] [key]MSSQL_UserDefinedFunction ScopedElement

[1729] Access Type: Read-only

[1730] The ScopedElement property references a user-defined function defined within the database referenced by the ScopingElement property.

[1731] [key]MSSQL_Database ScopingElement

[1732] Access Type: Read-only

[1733] The ScopingElement property references a database in SQL Server™.

[1734] MSSQL_DatabaseView: CIM_Component

[1735] Association Class

[1736] The MSSQL_DatabaseView class associates a database to the view contained within the database.

[1737] References

[1738] [key]MSSQL_Database GroupComponent

[1739] Access Type: Read-only

[1740] The parent element in the association

[1741] [key]MSSQL_View PartComponent

[1742] Access Type: Read-only

[1743] The child element in the association

[1744] MSSQL_Datatype: MSSQL_DBMSObject

[1745] Abstract Class

[1746] The MSSQL_Datatype class represents all the datatypes defined in a SQL Server™ installation. This includes both the user-defined datatypes, as well as the system-defined datatypes.

[1747] Properties

[1748] boolean AllowIdentity

[1749] Access Type: Read-only

[1750] The AllowIdentity property indicates the ability of a data type to participate in a column defined with the identity property. The SQL Server™ identity property is defined for data types that can accept numeric values. A column defined with the identity property is defined with a starting value and a step value. SQL Server™ generates values for the column by querying the last applicable value and adding the step value.

[1751] boolean AllowNulls

[1752] Access Type: Read-only

[1753] The AllowNulls property indicates whether the data type has the ability to accept NULL as a value.

[1754] string Caption

[1755] Access Type: Read-only

[1756] The Caption property is a short textual description (one-line string) of the object.

[1757] Maximum Length: 64

[1758] [key] string DatabaseName

[1759] Access Type: Read-only

[1760] The DatabaseName property indicates the name of the database that the object is a part of.

[1761] Maximum Length: 128

[1762] string Description

[1763] Access Type: Read-only

[1764] The Description property provides a textual description of the object.

[1765] datetime InstallDate

[1766] Access Type: Read-only

[1767] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[1768] [key] string Name

[1769] Access Type: Read-only

[1770] The Name property defines the label by which the object is known.

[1771] [key] string SQLServerName

[1772] Access Type: Read-only

[1773] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[1774] Maximum Length: 128

[1775] string Status

[1776] Access Type: Read-only

[1777] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[1778] Values are: "OK", "Error", "Degraded", "Unknown","Pred Fail", "Starting", "Stopping", "Service"

[1779] Maximum Length: 10

[1780] Associations

[1781] MSSQL_Datatype is associated to MSSQL_Database as the ScopedElement property of the MSSQL_DatabaseDatatype association.

[1782] MSSQL_Datatype is associated to MSSQL_Column as the Antecedent property of the MSSQL_ColumnDatatype association.

[1783] MSSQL_DBMSObject: CIM_LogicalElement

[1784] Abstract Class

[1785] The MSSQL_DBMSObject class represents objects in a database system. These objects include storage objects such as databases, tables, keys and constraints.

[1786] Properties

[1787] string Caption

[1788] Access Type: Read-only

[1789] The Caption property is a short textual description (one-line string) of the object.

[1790] Maximum Length: 64

[1791] string Description

[1792] Access Type: Read-only

[1793] The Description property provides a textual description of the object.

[1794] datetime InstallDate

[1795] Access Type: Read-only

[1796] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[1797] string Name

[1798] Access Type: Read-only

[1799] The Name property defines the label by which the object is known. When subclassed, the Name property can be overridden to be a Key property.

[1800] string Status

[1801] Access Type: Read-only

[1802] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[1803] Values are: "OK","Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[1804] Maximum Length: 10

[1805] Associations

[1806] MSSQL_DBMSObject is associated to MSSQL_DBMSUserObject as the Element property of the MSSQL_Permission association.

[1807] MSSQL_DBMSObject is associated to MSSQL_User as the Dependent property of the MSSQL_DBMSObjectOwner association.

[1808] MSSQL_DBMSObjectOwner: CIM_Dependency

[1809] Association Class

[1810] The MSSQL_DBMSObjectOwner class represents an association between a SQL Server™ m database object and the user who owns the object.

[1811] References

[1812] [key]MSSQL_User Antecedent

[1813] Access Type: Read-only

[1814] The Antecedent property references the user who owns the database object referenced by the Dependent property.

[1815] [key]MSSQL_DBMSObject Dependent

[1816] Access Type: Read-only

[1817] The Dependent property references a SQL Server™ database object.

[1818] MSSQL_Default: MSSQL_DBMSObject

[1819] The MSSQL_Default class represents the attributes of a single Microsoft® SQL Server™ default. SQL Server™ defaults provide data to columns and user-defined data types when no other data is available on an INSERT statement execution.

[1820] Properties

[1821] string Caption

[1822] Access Type: Read-only

[1823] The Caption property is a short textual description (one-line string) of the object.

[1824] Maximum Length: 64

[1825] datetime CreateDate

[1826] Access Type: Read-only

[1827] The CreateDate property indicates the time and date on which the default was created. Note that creation date may be different from the install date in cases where the object is created in one place and then installed elsewhere.

[1828] [key] string DatabaseName

[1829] Access Type: Read-only

[1830] The DatabaseName property indicates the name of the database that the object is a part of.

[1831] Maximum Length: 128

[1832] string Description

[1833] Access Type: Read-only

[1834] The Description property provides a textual description of the object.

[1835] datetime InstallDate

[1836] Access Type: Read-only

[1837] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[1838] [key] string Name

[1839] Access Type: Read-only

[1840] The Name property defines the label by which the object is known.

[1841] [key] string SQLServerName

[1842] Access Type: Read-only

[1843] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[1844] Maximum Length: 128

[1845] string Status

[1846] Access Type: Read-only

[1847] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[1848] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", Stopping", "Service"

[1849] Maximum Length: 10

[1850] string Text

[1851] Access Type: Read/Write

[1852] The Text property indicates the Transact-SQL script that defines the object. Note that there is a special requirement for the name of the object is specified in the CREATE statement. The name of the object has to be in the form that includes the name of the owner. For example, in order to create an object named "Some Object", one would need to specify it as [dbo].[Some Object].

[1853] Methods

[1854] The MSSQL_Default class supports the following methods:

Method Name	Description
Rename	The Rename method is used to rename a default instance.

[1855] Associations

[1856] MSSQL_Default is associated to MSSQL_Database as the ScopedElement property of the MSSQL_DatabaseDefault association.

[1857] MSSQL_Default is associated to MSSQL_Column as the Antecedent property of the MSSQL_ColumnDefault association.

[1858] MSSQL_Default is associated to MSSQLUserDatatype as the Antecedent property of the MSSQL_UserDatatypeDefault association.

[1859] MSSQL_DRIDefault: MSSQL_DBMSObject

[1860] The MSSQL_DRIDefault class represents the properties of a Microsoft® SQL Server™ column DEFAULT constraint.

[1861] Properties

[1862] string Caption

[1863] Access Type: Read-only

[1864] The Caption property is a short textual description (one-line string) of the object.

[1865] Maximum Length: 64

[1866] [key] string ColumnName

[1867] Access Type: Read-only

[1868] The ColumnName property indicates the name of the column that the DRI default is defined in.

[1869] Maximum Length: 128

[1870] [key] string DatabaseName

[1871] Access Type: Read-only

[1872] The DatabaseName property indicates the name of the database that the object is a part of.

[1873] Maximum Length: 128

[1874] string Description

[1875] Access Type: Read-only

[1876] The Description property provides a textual description of the object.

[1877] datetime InstallDate

[1878] Access Type: Read-only

[1879] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[1880] [key] string Name

[1881] Access Type: Read-only

[1882] The Name property defines the label by which the object is known.

[1883] [key] string SQLServerName

[1884] Access Type: Read-only

[1885] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[1886] Maximum Length: 128

[1887] string Status

[1888] Access Type: Read-only

[1889] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a

failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[1890] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[1891] Maximum Length: 10

[1892] [key] string TableName

[1893] Access Type: Read-only

[1894] The TableName property indicates the name of the table that the DRI default is defined in.

[1895] Maximum Length: 128

[1896] string Text

[1897] Access Type: Read/Write

[1898] The Text property indicates the Transact-SQL or other script that defines the object. Note that there is a special requirement for the name of the object is specified in the CREATE statement. The name of the object has to be in the form that includes the name of the owner. For example, in order to create an object named "Some Object", one would need to specify it as [dbo].[Some Object]. Associations

[1899] MSSQL_DRIDefault is associated to MSSQL_Column as the Antecedent property of the MSSQL_ColumnDRIDefault association.

[1900] MSSQL_ErrorLog: CIM_LogicalElement

[1901] The MSSQL_ErrorLog class represents the SQL Service error logs.

[1902] Properties

[1903] [key] string ArchiveID

[1904] Access Type: Read-only

[1905] ArchiveNumber identifies the number of the log. The active log has number 0.

[1906] Maximum Length: 128

[1907] string Caption

[1908] Access Type: Read-only

[1909] The Caption property is a short textual description (one-line string) of the object.

[1910] Maximum Length: 64

[1911] string Description

[1912] Access Type: Read-only

[1913] The Description property provides a textual description of the object.

[1914] datetime InstallDate

[1915] Access Type: Read-only

[1916] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[1917] datetime LastModified

[1918] Access Type: Read-only

[1919] LastModified indicates the time and date the log was last modified.

[1920] [key] string Name

[1921] Access Type: Read-only

[1922] The Name property defines the label by which the object is known. When subclassed, the Name property can be overridden to be a Key property.

[1923] string Status

[1924] Access Type: Read-only

[1925] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[1926] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[1927] Maximum Length: 10

[1928] Associations

[1929] MSSQL_ErrorLog is associated to MSSQL_SQLServer as the Antecedent property of the MSSQL_SQLServerErrorLog association.

[1930] MSSQL_ErrorLog is associated to MSSQL ErrorLogEntry as the GroupComponent property of the MSSQL_ErrorLogErrorLogEntry association.

[1931] MSSQL_ErrorLog is associated to CIM_DataFile as the Dependent property of the MSSQL_ErrorLogDataFile association.

[1932] MSSQL ErrorLogDataFile: CIM_Dependency

[1933] Association Class

[1934] The MSSQL_ErrorLogDataFile class represents an association between a SQL Server™ error log, and the operating system file used to store the error log.

[1935] References

[1936] [key]CIM_DataFile Antecedent

[1937] Access Type: Read-only

[1938] The Antecedent property references an operating system file used to store the error log referenced by the Dependent property.

[1939] [key]MSSQL_ErrorLog Dependent

[1940] Access Type: Read-only

[1941] The Dependent property references a SQL Server™ error log.

[1942] MSSQL_ErrorLogEntry: CIM_LogicalElement

[1943] The MSSQL_ErrorLogEntry class represents the entries in a SQL Service error log.

[1944] Properties

[1945] [key] string ArchiveID

[1946] Access Type: Read-only

[1947] The ArchiveNumber property identifies the archive number of the log the entry is stored in. The active log has number 0.

[1948] Maximum Length: 128

[1949] string Caption

[1950] Access Type: Read-only

[1951] The Caption property is a short textual description (one-line string) of the object.

[1952] Maximum Length: 64

[1953] string Description

[1954] Access Type: Read-only

[1955] The Description property provides a textual description of the object.

[1956] [key] string EntryID

[1957] Access Type: Read-only

[1958] EntryNumber identifies the number of the entry within a log. Each entry in a log is successively numbered.

[1959] Maximum Length: 128

[1960] datetime InstallDate

[1961] Access Type: Read-only

[1962] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[1963] string Name

[1964] Access Type: Read-only

[1965] The Name property defines the label by which the object is known. When subclassed, the Name property can be overridden to be a Key property.

[1966] [key] string SQLServerName

[1967] Access Type: Read-only

[1968] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[1969] Maximum Length: 128

[1970] string Status

[1971] Access Type: Read-only

[1972] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[1973] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[1974] Maximum Length: 10

[1975] string Text

[1976] Access Type: Read-only

[1977] Text is the descriptive text of the error log entry.

[1978] Associations

[1979] MSSQL_ErrorLogEntry is associated to MSSQL_ErrorLog as the PartComponent property of the MSSQL_ErrorLogErrorLogEntry association.

[1980] MSSQL_ErrorLogErrorLogEntry: CIM_Component

[1981] Association Class

[1982] The MSSQL_ErrorLogErrorLogEntry class represents an association between an error log and an entry in the error log.

[1983] References

[1984] [key]MSSQL_ErrorLog GroupComponent

[1985] Access Type: Read-only

[1986] The GroupComponent property references a SQL Server™ error log.

[1987] [key]MSSQL_ErrorLogEntry PartComponent

[1988] Access Type: Read-only

[1989] The PartComponent property references an entry in the error log referenced by the GroupComponent property.

[1990] MSSQL_Extends

[1991] Abstract Class

[1992] Association Class

[1993] The MSSQL_Extends class is an abstract association class. It associates a class with another class that extends the former class by defining some new properties and methods.

[1994] References

[1995] CIM_ManagedSystemElement ExtendedElement

[1996] Access Type: Read-only

[1997] The ExtendedElement property references a managed system element that is being extended.

[1998] MSSQL_Extension Extension

[1999] Access Type: Read-only

[2000] The Extension property references a extension to the managed system element referenced by the ExtendedElement property.

[2001] MSSQL_Extension

[2002] Abstract Class

[2003] The MSSQL_Extension class represents extensions made via associations to a managed system element. Extensions are made via associations when it is not possible or desirable to extend a class by subclassing.

[2004] Associations

[2005] MSSQL_Extension is associated to CIM_ManagedSystemElement as the Extension property of the MSSQL_Extends association.

[2006] MSSQL_FileGroup: CIM_LogicalElement

[2007] The MSSQL_FileGroup class exposes the attributes of a Microsoft® SQL Server™ filegroup.

[2008] Properties

[2009] string Caption

[2010] Access Type: Read-only

[2011] The Caption property is a short textual description (one-line string) of the object.

[2012] Maximum Length: 64

[2013] [key] string DatabaseName

[2014] Access Type: Read-only

[2015] The DatabaseName property indicates the name of the database that the object is a part of.

[2016] Maximum Length: 128

[2017] boolean Default

[2018] Access Type: Read/Write

[2019] The Default property indicates the filegroup used when no filegroup is specified as part of table or index creation. If TRUE, the referenced filegroup is used to implement table or index data storage when a table or index is created and no filegroup is specified. If FALSE, the referenced filegroup is not used as the default in table and index creation. The filegroup may be specified by name to direct creation.

[2020] string Description

[2021] Access Type: Read-only

[2022] The Description property provides a textual description of the object.

[2023] datetime InstallDate

[2024] Access Type: Read-only

[2025] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[2026] [key] string Name

[2027] Access Type: Read-only

[2028] The Name property defines the label by which the object is known. When subclassed, the Name property can be overridden to be a Key property.

[2029] boolean ReadOnly

[2030] Access Type: Read/Write

[2031] The ReadOnly property controls the ability to update a Microsoft® SQL Server™ database or database filegroup. If TRUE, data in the database or database filegroup cannot be changed. If FALSE, updates are allowed to data in the database or database filegroup.

[2032] sint32 Size

[2033] Access Type: Read-only

[2034] The Size property exposes the total size, in megabytes, of the filegroup.

[2035] [key] string SQLServerName

[2036] Access Type: Read-only

[2037] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[2038] Maximum Length: 128

[2039] string Status

[2040] Access Type: Read-only

[2041] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[2042] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[2043] Maximum Length: 10

[2044] uint32 Type

[2045] This property indicates the type of the file group. The file group can be of three types. It is either a user defined file group, a primary file group, or a filegroup defined on files maintained on read-only media. When a database is created, it is created on exactly one filegroup named PRIMARY. This is the primary file group. After database creation, filegroups can be added to the database. These are the user defined file groups.

Value	Description
0	User Defined
8	On Read-Only Media
16	Primary

[2046] Methods

[2047] The MSSQL_FileGroup class supports the following methods:

Method Name	Description
CheckFilegroup	The CheckFilegroup method scans and tests the integrity of database pages maintained in operating system files implementing the referenced filegroup.
CheckFilegroupDataOnly	The CheckFilegroupDataOnly method scans and tests the integrity of database pages that are used to maintain table data in the operating system files implementing the referenced filegroup.
EnumStoredObiects	The EnumStoredObjects method enumerates the names of the indexes, tables and statistical mechanisms stored in the filegroup.

[2048] Associations

[2049] MSSQL_FileGroup is associated to MSSQL_Database as the Antecedent property of the MSSQL_DatabaseFileGroup association.

[2050] MSSQL_FileGroup is associated to MSSQL_Table as the Antecedent property of the MSSQL_TableFileGroup association.

[2051] MSSQL_FileGroup is associated to MSSQL_Table as the Antecedent property of the MSSQL_TableTextFileGroup association.

[2052] MSSQL_FileGroup is associated to MSSQL_Index as the Antecedent property of the MSSQL_indexFileGroup association.

[2053] MSSQL_FileGroup is associated to MSSQL_CandidateKey as the

[2054] Antecedent property of the MSSQL_KeyFileGroup association.

[2055] MSSQL_FileGroup is associated to MSSQL_DatabaseFile as the GroupComponent property of the MSSQL_FileGroupDatabaseFile association.

[2056] MSSQL_FileGroupDatabaseFile: CIM_Component Association Class

[2057] The MSSQL_FileGroupDatabaseFile class associates a database file group to the operating system files that are part of the group.

[2058] References

[2059] [key]MSSQL_FileGroup GroupComponent

[2060] Access Type: Read-only

[2061] The parent element in the association

[2062] [key]MSSQL_DatabaseFile PartComponent

[2063] Access Type: Read-only

[2064] The child element in the association

[2065] MSSQL_ForeignKey: MSSQL_Key

[2066] The MSSQL_ForeignKey class represents the foreign keys defined for a SQL Server™ database table.

[2067] Properties

[2068] string Caption

[2069] Access Type: Read-only

[2070] The Caption property is a short textual description (one-line string) of the object.

[2071] Maximum Length: 64

[2072] boolean Checked

[2073] Access Type: Read/Write

[2074] The Checked property enables or disables foreign constraint evaluation for an existing foreign key constraint. If True, an attempt is made to enforce a foreign key constraint when rows are added to the table on which the constraint is defined. If False, no attempt is made to enforce the foreign constraint when rows are added to the table on which the constraint is defined.

[2075] [key] string DatabaseName

[2076] Access Type: Read-only

[2077] The DatabaseName property indicates the name of the database that the key is a part of.

[2078] Maximum Length: 128

[2079] string Description

[2080] Access Type: Read-only

[2081] The Description property provides a textual description of the object.

[2082] boolean ExcludeReplication

[2083] Access Type: Read-only

[2084] The ExcludeReplication property controls foreign key constraint enforcement when replicated data is inserted into the columns on which the foreign key constraint is defined

[2085] datetime InstallDate

[2086] Access Type: Read-only

[2087] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[2088] [key] string Name

[2089] Access Type: Read-only

[2090] The Name property defines the label by which the object is known. The name of a key is unique within a database.

[2091] [key] string SQLServerName

[2092] Access Type: Read-only

[2093] The SQLServerName property indicates the name of the SQL Server™ installation that the key is a part of.

[2094] Maximum Length: 128

[2095] string Status

[2096] Access Type: Read-only

[2097] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror-resilvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[2098] Values are: "OK", Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[2099] Maximum Length: 10

[2100] [key] string TableName

[2101] Access Type: Read-only

[2102] The TableName property indicates the name of the table that the key is defined in.

[2103] Maximum Length: 128

[2104] Methods

[2105] The MSSQL_ForeignKey class supports the following methods:

Method Name	Description
Create	The Create method is used to create a new foreign key.
Rename	The Rename method is used to rename the foreign key instance.

[2106] Associations

[2107] MSSQL_ForeignKey is associated to MSSQL-_CandidateKey as the Dependent property of the MSSQL-_ReferencedKey association.

[2108] MSSQL_ForeignKey is associated to MSSQL-_Table as the Dependent property of the MSSQL_ReferencedTable association.

[2109] MSSQL_FullTextCatalog: CIM_LogicalElement

[2110] The MSSQL_FullTextCatalog class represents a single Microsoft® Search persistent data store. Microsoft® Search enables full-text queries on data maintained by Microsoft® SQL Server™. The service both builds the indexes providing full-text query capability and participates in query resolution by providing result data during a full-text query. Index data is maintained within a full-text catalog.

[2111] Properties

[2112] string Caption

[2113] Access Type: Read-only

[2114] The Caption property is a short textual description (one-line string) of the object.

[2115] Maximum Length: 64

[2116] [key] string DatabaseName

[2117] Access Type: Read-only

[2118] The DatabaseName property indicates the name of the database that the full text catalog is a part of.

[2119] Maximum Length: 128

[2120] string Description

[2121] Access Type: Read-only

[2122] The Description property provides a textual description of the object.

[2123] uint32 ErrorLogSize

[2124] Access Type: Read-only

[2125] The ErrorLogSize property returns the size, in bytes, of a Microsoft® Search full-text catalog error log.

[2126] Units: Bytes

[2127] boolean FullTextIndexedTables

[2128] Access Type: Read-only

[2129] The FullTextIndexedTables property reports Microsoft® Search full-text catalog use. When TRUE, at least one table uses the full-text catalog for index data storage. When FALSE, the full-text catalog is not currently used for index data storage.

[2130] uint32 FullTextIndex

[2131] Access Type: Read-only

[2132] The FullTextIndex property returns the size, in megabytes, of the full-text catalog.

[2133] Units: Megabytes

[2134] datetime InstallDate

[2135] Access Type: Read-only

[2136] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[2137] uint32 ItemCount

[2138] Access Type: Read-only

[2139] The ItemCount property returns the number of entries contained in a Microsoft® Search full-text catalog. For each table indexed in the full-text catalog, an entry is made for the table and an entry is made for each row in the table.

[2140] [key] string Name

[2141] Access Type: Read/Write

[2142] The Name property is a label used to uniquely identify the object.

[2143] uint32 PopulateCompletionAge

[2144] Access Type: Read-only

[2145] The PopulateCompletionAge property returns the number of seconds between the time of the most recent, successful Microsoft® Search full-text catalog population and a system-defined date and time. A value of zero represents the base date and time, 12:00:00 AM, Jan. 1, 1990.

[2146] Units: Seconds

[2147] datetime PopulateCompletionDate

[2148] Access Type: Read-only

[2149] The PopulateCompletionDate property returns the most recent date land time at which an update was made to the full-text catalog.

[2150] uint32 PopulateStatus

[2151] Access Type: Read-only

[2152] The PopulateStatus property returns the population state of a Microsoft® Search full-text catalog. The property can have one of the following values: Idle—No action is being performed against the referenced full-text catalog. In Progress—Full index population is in progress for the referenced full-text catalog. Paused—Lack of available resource, such as disk space, has caused an interruption. Throttled—Microsoft® Search service has paused the referenced full-text index population. Recovering—Interrupted population on the referenced full-text catalog is resuming. Shutdown—Referenced full-text catalog is being deleted or is otherwise not accessible. Incremental—Incremental index population is in progress for the referenced full-text catalog. Updating Index—Referenced full-text catalog is being assembled by the Microsoft® Search service. Assemblage is the final step in full-text catalog population.

[2153] Values are: "Idle", "In Progress", "Paused", "Throttled", "Recovering", "Shutdown", "Incremental", "Updating Index"

[2154] string RootPath

[2155] Access Type: Read/Write

[2156] The RootPath property specifies an operating system directory used as the primary path for Microsoft® Search full-text catalog storage. If the RootPath property is an empty string when creating a Microsoft® Search full-text catalog, the default data path, set for the Microsoft® Search service, is used.

[2157] [key] string SQLServerName

[2158] Access Type: Read-only

[2159] The SQLServerName property indicates the name of the SQL Server™ installation that the full-text catalog is a part of.

[2160] Maximum Length: 128

[2161] string Status

[2162] Access Type: Read-only

[2163] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[2164] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[2165] Maximum Length: 10

[2166] uint32 UniqueKeyCount

[2167] Access Type: Read-only

[2168] The UniqueKeyCount property returns an approximate number of words uniquely addressable in a Microsoft® Search full-text catalog.

[2169] Methods

[2170] The MSSQL_FullTextCatalog class supports the following methods:

Method Name	Description
Rebuild	The Rebuild method re-creates the Microsoft Search full-text catalog.
StartFullTextPopulation	The StartFullTextPopulation method starts Microsoft Search full-text table population.

-continued

Method Name	Description
StopFullTextPopulation	The StopFullTextPopulation method stops full-text population.

[2171] Associations

[2172] MSSQL_FullTextCatalog is associated to MSSQL_Database as the Antecedent property of the MSSQL_DatabaseFullTextCatalog association.

[2173] MSSQL_FullTextCatalogService: CIM_Service

[2174] The MSSQL_FullTextCatalogService class represents the Microsoft® Search full-text indexing service. The Microsoft® Search full-text indexing service enables full-text queries on data maintained by SQL Server. Microsoft® Search both builds the indexes providing full-text query capability and participates in query resolution by providing result data during a full-text query.

[2175] Properties

[2176] string Caption

[2177] Access Type: Read-only

[2178] The Caption property is a short textual description (one-line string) of the object.

[2179] Maximum Length: 64

[2180] sint32 ConnectTimeout

[2181] Access Type: Read/Write

[2182] The ConnectTimeout property specifies a time interval used by the Microsoft® Search service when attempting a connection to a Microsoft® SQL Server™ installation enabled for full-text search.

[2183] string CreationClassName

[2184] Access Type: Read-only

[2185] CreationClassName indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.

[2186] string DefaultPath

[2187] Access Type: Read-only

[2188] The DefaultPath property returns the operating system path naming a directory used as a root for Microsoft® Search full-text catalog implementation if no user-specified path is supplied during full-text catalog creation.

[2189] string Description

[2190] Access Type: Read-only

[2191] The Description property provides a textual description of the object.

[2192] datetime InstallDate

[2193] Access Type: Read-only

[2194] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[2195] boolean IsFullTextInstalled

[2196] Access Type: Read-only

[2197] The IsFullTextInstalled property returns True when the Microsoft® Search service has been successfully installed on a server running Microsoft® SQL Server™.

[2198] [key] string Name

[2199] Access Type: Read-only

[2200] The Name property uniquely identifies the service and provides an indication of the functionality that is managed. This functionality is described in more detail in the object's Description property. The default value of this property is "MSSearch"

[2201] sint32 ResourceUsage

[2202] Access Type: Read/Write

[2203] The ResourceUsage property specifies a relative operating system execution priority setting for the Microsoft® Search service. Use the ResourceUsage property to raise or lower execution priority for a running Microsoft® Search service. By default, ResourceUsage is 3, interpreted as normal priority for the service. Set ResourceUsage to 2 or 1 to lower the execution priority for the Microsoft® Search service. Set ResourceUsage to 4 or 5 to raise the execution priority. Note that a value of 5 represents dedicated priority for the Microsoft® Search service. Setting the ResourceUsage property to a value higher than 3 can have unintended consequences and should be considered only after evaluating the possible effects on other services running on the computer.

[2204] uints32 ServiceStatus

[2205] Access Type: Read-only

[2206] The ServiceStatus property reports the execution state of the service.

Value	Description
0	Unknown
1	Running
2	Paused
3	Stopped
4	Starting
5	Stopping
6	Pausing

[2207] boolean Started

[2208] Access Type: Read-only

[2209] Started is a boolean indicating whether the service has been started (TRUE), or stopped (FALSE).

[2210] string StartMode

[2211] Access Type: Read-only

[2212] StartMode is a string value indicating whether the service is automatically started by a operating system, or only started upon request.

[2213] Values are: "Automatic", "Manual"

[2214] string Status

[2215] Access Type: Read-only

[2216] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[2217] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[2218] Maximum Length: 10

[2219] string SystemCreationClassName

[2220] Access Type: Read-only

[2221] The type name of the system that hosts this service.

[2222] string SystemName

[2223] Access Type: Read-only

[2224] The name of the system that hosts this service.

[2225] Methods

[2226] The MSSQL_FullTextCatalogService class supports the following methods:

Method Name	Description
CleanUp	The CleanUp method directs the Microsoft ® Search service to locate and remove full-text catalog resources in the file system.
StartService	The StartService method starts the Search service.
StopService	The StopService method stops the Search service.

[2227] Associations

[2228] MSSQL_FullTextCatalogService is associated to Win32_Service as the SameElement property of the MSSQL_FullTextWin32Service association.

[2229] MSSQL_FullTextWin32Service: CIM_LogicalIdentity

[2230] Association Class

[2231] The MSSQL_FullTextWin32Service represents an association between an instance of MSSQL_FullTextCatalogService and the corresponding instance of the Win32_Service.

[2232] References

[2233] [key] Win32_Service SystemElement

[2234] Access Type: Read-only

[2235] The SystemElement property references a Win32 service.

[2236] [key]MSSQL_FullTextCatalogService SameElement

[2237] Access Type: Read-only

[2238] The SameElement property references the full text catalog service.

[2239] MSSQL_Index: MSSQL_DBMSObject

[2240] The MSSQL_Index class represents an index for a SQL Server™ table. A SQL Server™ index optimizes access to data in SQL Server™ tables. Indexes are also used to enforce some constraints, such as UNIQUE and PRIMARY KEY constraints.

[2241] Properties

[2242] string Caption

[2243] Access Type: Read-only

[2244] The Caption property is a short textual description (one-line string) of the object.

[2245] Maximum Length: 64

[2246] [key] string DatabaseName

[2247] Access Type: Read-only

[2248] The DatabaseName property indicates the name of the database that the object is a part of.

[2249] Maximum Length: 128

[2250] string Description

[2251] Access Type: Read-only

[2252] The Description property provides a textual description of the object.

[2253] sint32 FillFactor

[2254] Access Type: Read-only

[2255] The FillFactor property indicates the percent of each page used to store index data when the index is created.

[2256] datetime InstallDate

[2257] Access Type: Read-only

[2258] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[2259] boolean IsFullTextKey

[2260] Access Type: Read-only

[2261] The IsFullTextKey property identifies the index used by Microsoft® Search to support row identification. When TRUE, the referenced index is used by Microsoft® Search for row identification. When FALSE, the referenced index is not used by Microsoft® Search. Microsoft® Search requires that a single column identify rows participating in an index supporting full-text query. The column designated must contain unique, non-NULL values and must participate in a PRIMARY KEY or UNIQUE key constraint. A table containing a PRIMARY KEY constraint does not require a separate unique index for Microsoft® Search configuration. Use UniqueIndexForFullText in the MSSQL_Table class to configure Microsoft® Search full-text index key column use.

[2262] [key] string Name

[2263] Access Type: Read-only

[2264] The Name property defines the label by which the object is known.

[2265] boolean NoRecompute

[2266] Access Type: Read-only

[2267] The NoRecompute property controls statistics generation when the MSSQL_Index class is used to create a Microsoft® SQL Server™ index. When TRUE, SQL Server™ does not perform automatic data-distribution statistics update on the created index. When FALSE (default), automatic data-distribution statistics update is performed. Use the UpdateIndexStatistics, UpdateStatistics, or UpdateStatisticsWith methods to force an update of index statistics for SQL Server™ indexes not configured for automatic update. Use the UpdateStatisticsWith method of the MSSQL_Table object to enable or disable automatic update of data-distribution statistics for an existing index.

[2268] sint32 SpaceUsed

[2269] Access Type: Read-only

[2270] The SpaceUsed property returns the amount of disk resource used, in kilobytes, to store data that implements the index.

[2271] Units: Kilobytes

[2272] [key] string SQLServerName

[2273] Access Type: Read-only

[2274] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[2275] Maximum Length: 128

[2276] boolean StatisticsIndex

[2277] Access Type: Read-only

[2278] The StatisticsIndex property has a value of TRUE when the index maintains data distribution statistics. SQL Server™ query optimization relies, in part, on data distribution statistics maintained on indexes. To aid query optimization, SQL Server™ can generate data distribution statistics for one or more columns in a table without imposing overhead associated with index maintenance.

[2279] string Status

[2280] Access Type: Read-only

[2281] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[2282] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[2283] Maximum Length: 10

[2284] [key] string TableName

[2285] Access Type: Read-only

[2286] The TableName property indicates the name of the table that the key is defined in.

[2287] Maximum Length: 128

[2288] uint32 Type

[2289] Access Type: Read-only

[2290] The Type property specifies the type of the index. If none of the bits are set, it implies that it is a non-clustered index. This is the default type for the index. If DRI Primary Key or DRI Unique Key is set, it implies that that the index is used to maintain a DRI constraint. The bits for Default, Ignore Duplicate Key, Clustered, Pad Index, Drop Exist are bits that can be used at time of index creation.

Bit Position	Description	Explanation
0	Ignore Duplicate Key	Controls error generation when an INSERT or UPDATE operation could cause a constraint violation and the index implements a PRIMARY KEY or UNIQUE constraint.
1	Unique	Index implements a UNIQUE constraint.
4	Clustered	The index is clustered. SQL Server ™ supports a single clustered index on any table.
5	Hypothetical	Redirects index creation, mapping index object manipulation to CREATE STATISTICS and DROP STATISTICS statements.
8	Pad Index	Pad the index nodes using fill factor.

-continued

Bit Position	Description	Explanation
11	DRI Primary Key	The index implements a PRIMARY KEY constraint.
12	DRI Unique Key	The index implements a UNIQUE constraint on a table not constrained by primary key.
15	Drop Exist	Optimizes index creation when an existing index is rebuilt.
24	No Recompute	Index created with statistics computation off. For more information, see NoRecompute property.

[2291] Methods

[2292] The MSSQL_Index class supports the following methods:

Method Name	Description
CheckIndex	The CheckIndex method tests the integrity of database pages implementing storage for the referenced index.
Create	The Create method is used to create a new instance of an index.
GetIndexedColumnDESC	The GetIndexedColumnDESC method specifies whether the sort order of a column in an index is descending.
Rebuild	The Rebuild method re-creates the index.
RecalcSpaceUsage	The RecalcSpaceUsage method forces the update of data reporting the disk resource usage of the index.
Rename	The Rename method is used to rename an instance of an index.
SetIndexedColumnDESC	The SetIndexedColumnDESC method specifies a column to sort in descending order as part of an index.
UpdateStatistics	The UpdateStatistics method forces data distribution statistics update for the index.
UpdateStatisticsWith	The UpdateStatisticsWith method forces data distribution statistics update for a hypothetical index used to support data distribution statistics for the index.

[2293] Associations

[2294] MSSQL_Index is associated to MSSQL_FileGroup as the Dependent property of the MSSQL_IndexFileGroup association.

[2295] MSSQL_Index is associated to MSSQL_Table as the Antecedent property of the MSSQL_TableIndex association.

[2296] MSSQL_Index is associated to MSSQL_Column as the Dependent property of the MSSQL_IndexColumn association.

[2297] MSSQL_Index is associated to MSSQL_IndexTableInformation as the Element property of the MSSQL_IndexStatistics association.

[2298] MSSQL_IndexColumn: CIM_Dependency

[2299] Association Class

[2300] The MSSQL_IndexColumn class represents an association between an index and a column that participates in the index.

[2301] Properties

[2302] boolean DescendingSortOrder

[2303] Access Type: Read-only

[2304] This property indicates whether the sort order of a column in an index is descending. A value of TRUE implies that it is descending.

[2305] References

[2306] [key]MSSQL_Column Antecedent

[2307] Access Type: Read-only

[2308] The Antecedent property references a column that participates in the index referenced by the Dependent property.

[2309] [key]MSSQL_Index Dependent

[2310] Access Type: Read-only

[2311] The Dependent property references an index in a SQL Server™ table.

[2312] MSSQL_IndexFileGroup: CIM_Dependency

[2313] Association Class

[2314] The MSSQL_IndexFileGroup class represents an association between an index and a file group that stores the index.

[2315] References

[2316] [key]MSSQL_FileGroup Antecedent

[2317] Access Type: Read-only

[2318] The Antecedent property references the file group that stores the index referenced by the Dependent property.

[2319] [key]MSSQL_Index Dependent

[2320] Access Type: Read-only

[2321] The Dependent property references an index in SQL Server™.

[2322] MSSQL_IndexStatistics: CIM_Statistics

[2323] Association Class

[2324] The MSSQL_IndexStatistics class represents an association between an index and the statistical information stored with the index.

[2325] References

[2326] [key]MSSQL_Index Element

[2327] Access Type: Read-only

[2328] The Element property references a SQL Server index.

[2329] [key]MSSQL_IndexTableInformation Stats

[2330] Access Type: Read-only

[2331] The Stats property references the statistical information stored with the index referenced by the Element property.

[2332] MSSQL_IndexTableInformation: CIM_StatisticalInformation

[2333] The MSSQL_IndexTableInformation class represents the information regarding the age and structure of the index statistical information.

[2334] Properties

[2335] real32 AverageKeyLength

[2336] Access Type: Read-only

[2337] The AverageKeyLength property represents the average length of an index row.

[2338] string Caption

[2339] Access Type: Read-only

[2340] A short textual description (one-line string) for the statistic or metric.

[2341] Maximum Length: 64

[2342] [key] string DatabaseName

[2343] Access Type: Read-only

[2344] The DatabaseName property indicates the name of the database that the object is a part of.

[2345] Maximum Length: 128

[2346] real32 Density

[2347] Access Type: Read-only

[2348] The Density property indicates the selectivity of the index. All indexes have distribution statistics that describe the selectivity and distribution of the key values in the index. Selectivity is a property that relates to how many rows are typically identified by a key value.

[2349] string Description

[2350] Access Type: Read-only

[2351] A textual description of the statistic or metric.

[2352] datetime LastUpdate

[2353] Access Type: Read-only

[2354] The LastUpdate property indicates the date and time of most recent update of the statistical information.

[2355] [key] string Name

[2356] Access Type: Read-only

[2357] The Name property defines the label by which the object is known.

[2358] Maximum Length: 256

[2359] uint64 Rows

[2360] Access Type: Read-only

[2361] The Rows property indicates the number of rows in the table.

[2362] uint64 RowsSampled

[2363] Access Type: Read-only

[2364] The RowsSampled property indicates the number of rows sampled for statistics data.

[2365] [key] string SQLServerName

[2366] Access Type: Read-only

[2367] The SQLServerName property indicates the name of the SQL Server™ installation that the index is a part of.

[2368] Maximum Length: 128

[2369] uint32 Steps

[2370] Access Type: Read-only

[2371] The Steps property indicates the number of histogram values in the current distribution statistics.

[2372] [key] string TableName

[2373] Access Type: Read-only

[2374] The TableName property indicates the name of the table in which the index is defined.

[2375] Maximum Length: 128

[2376] Associations

[2377] MSSQL_IndexTableInformation is associated to MSSQL_Index as the Stats property of the MSSQL_IndexStatistics association.

[2378] MSSQL_IntegratedSecuritySetting: MSSQL_Setting

[2379] The MSSQL_IntegratedSecuritySetting class represents the integrated security settings when WMI interacts with SQL Server?.

[2380] Properties

[2381] uint32 AuditLevel

[2382] Access Type: Read/Write

[2383] The AuditLevel property exposes SQL Server™ Authentication logging behavior.

Value	Description	Explanation
0	None	Do not log authentication attempts.
1	Audit Login Success	Log successful authentication.
2	Audit Login Failure	Log failed authentication.
3	Audit All	Log all authentication attempts regardless of success or failure.

[2384] string Caption

[2385] Access Type: Read-only

[2386] A short textual description (one-line string) of the MSSQL_IntegratedSecuritySetting object.

[2387] Maximum Length: 64

[2388] string Description

[2389] Access Type: Read-only

[2390] A textual description of the MSSQL_IntegratedSecuritySetting object.

[2391] boolean ImpersonateClient

[2392] Access Type: Read/Write

[2393] The ImpersonateClient property exposes the security context for non-administrative users executing xp_cmdshell. If TRUE, xp_cmdshell runs in the security context of the client connection. If FALSE, xp_cmdshell runs in the security context of SQL Server Agent. The default is False.

[2394] uint32 SecurityMode

[2395] Access Type: Read/Write

[2396] The SecurityMode property directs the authentication mode used by a Microsoft® SQL Server™ installation.

Value	Description	Explanation
0	Normal	Allow SQL Server Authentication only
1	Integrated	Allow Windows NT Authentication only
2	Mixed	Allow Windows NT or SQL Server Authentication

[2397] [key] string SettingID

[2398] Access Type: Read-only

[2399] The identifier by which the MSSQL_IntegratedSecuritySetting object is known.

[2400] Maximum Length: 256

[2401] Associations

[2402] MSSQL_IntegratedSecuritySetting is associated to MSSQL_SQLServer as the Setting property of the MSSQL_SQLServerIntegratedSecuritySetting association.

[2403] MSSQL_Key: MSSQL_Constraint

[2404] Abstract Class

[2405] The MSSQL_Key class represents the keys defined for a SQL Server™ table.

[2406] Properties

[2407] string Caption

[2408] Access Type: Read-only

[2409] The Caption property is a short textual description (one-line string) of the object.

[2410] Maximum Length: 64

[2411] [key] string DatabaseName

[2412] Access Type: Read-only

[2413] The DatabaseName property indicates the name of the database that the key is a part of.

[2414] Maximum Length: 128

[2415] string Description

[2416] Access Type: Read-only

[2417] The Description property provides a textual description of the object.

[2418] datetime InstallDate

[2419] Access Type: Read-only

[2420] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[2421] [key] string Name

[2422] Access Type: Read-only

[2423] The Name property defines the label by which the object is known. The name of a key is unique within a database.

[2424] [key] string SQLServerName

[2425] Access Type: Read-only

[2426] The SQLServerName property indicates the name of the SQL Server™ installation that the key is a part of.

[2427] Maximum Length: 128

[2428] string Status

[2429] Access Type: Read-only

[2430] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[2431] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[2432] Maximum Length: 10

[2433] [key] string TableName

[2434] Access Type: Read-only

[2435] The TableName property indicates the name of the table that the key is defined in.

[2436] Maximum Length: 128

[2437] Associations

[2438] MSSQL_Key is associated to MSSQL_Table as the Antecedent property of the MSSQL_TableKey association.

[2439] MSSQL-Key is associated to MSSQL_Column as the GroupComponent property of the MSSQL_KeyColumn association.

[2440] MSSQL_KeyColumn: CIM_Component

[2441] Association Class

[2442] The MSSQL_KeyColumn class represents an association between a key and a column that is part of the key.

[2443] References

[2444] [key]MSSQL_Key GroupComponent

[2445] Access Type: Read-only

[2446] The GroupComponent property references a key in a SQL Server™ database.

[2447] [key]MSSQL_Column PartComponent

[2448] Access Type: Read-only

[2449] The PartComponent property references a column that is part of the key referenced by the GroupComponent property.

[2450] MSSQL_KeyFileGroup: CIM_Dependency

[2451] Association Class

[2452] The MSSQL_KeyFileGroup class represents an association between a key and the file group used to store the key.

[2453] References

[2454] [key]MSSQL_FileGroup Antecedent

[2455] Access Type: Read-only

[2456] The Antecedent property references the file group that stores the key referenced by the Dependent property.

[2457] [key]MSSQL_CandidateKey Dependent

[2458] Access Type: Read-only

[2459] The Dependent property references a candidate key in SQL Server™.

[2460] MSSQL_LanguageSetting: MSSQL_Setting

[2461] The MSSQL_LanguageSetting class exposes the properties of an installed Microsoft® SQL Server™ language record. SQL Server™ language record identifiers categorize system messages so that error and status information can be presented as localized text.

[2462] Properties

[2463] string Alias

[2464] Access Type: Read-only

[2465] The Alias property identifies an alternate name for a SQL Server™ language. For localized versions of SQL Server™, the Alias property is an English name for the language record. For all other versions, Alias is the localized language name.

[2466] string Caption

[2467] Access Type: Read-only

[2468] A short textual description (one-line string) of the MSSQL_LanguageSetting object.

[2469] Maximum Length: 64

[2470] string Description

[2471] Access Type: Read-only

[2472] A textual description of the MSSQL_LanguageSetting object.

[2473] uint32 FirstDayOfWeek

[2474] Access Type: Read-only

[2475] The FirstDayOfWeek property returns the calendar start day of the week for a language record. A SQL Server™ language record records the names of the days of the week localized to the language. To enable system selection of the correct day name, the day name string is stored so that the localized name for Monday appears first. For some locales, Monday is not the starting calendar weekday.

[2476] string LangDateFormat

[2477] Access Type: Read-only

[2478] The LangDateFormat property is a three-character string describing the position of the day, month, and year members of a date. The LangDateFormat property reports day, month, and year positions using the characters d, m, and y respectively. For example, a SQL Server™ language displaying dates in month/day/year order reports "mdy" in the LangDateFormat property.

[2479] [key] string SettingID

[2480] Access Type: Read-only

[2481] The identifier by which the MSSQL_LanguageSetting object is known.

[2482] Maximum Length: 256

[2483] [key] string SQLServerName

[2484] Access Type: Read-only

[2485] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[2486] Maximum Length: 128

[2487] string Days []

[2488] Access Type: Read-only

[2489] The Days property identifies the names of the days of the week for a SQL Server™ language record. The Days property string array contains seven members. The first member is the day name for Monday. The locale start of the calendar week is set using the FirstDayOfWeek property. For example, the string array "Montag,Dienstag,Mittwoch,Donnerstag,Freitag,Samstag,Sonntag" is the Days property for the German (Deutsch) language record. For the language record, the FirstDayOfWeek property is 1, indicating that Monday (Montag) is the start of the calendar week.

[2490] string Months []

[2491] Access Type: Read-only

[2492] The Months property returns an array containing unabbreviated month names. The month names are ordered, beginning with January and ending with December.

[2493] string ShortMonths []

[2494] Access Type: Read-only

[2495] The ShortMonths property returns an array containing a list of month name abbreviations for a language. The list is ordered from month 1 (January) through month 12 (December). Month names are represented as a three-character abbreviation.

[2496] Associations

[2497] MSSQL-LanguageSetting is associated to MSSQL_SQLServer as the Setting property of the MSSQL_SQLServerLanguageSetting association.

[2498] MSSQL_Login: MSSQL_DBMSUserObject

[2499] The MSSQL_Login class represents the login authentication records present in a SQL Server™ installation.

[2500] Properties

[2501] string Caption

[2502] Access Type: Read-only

[2503] The Caption property is a short textual description (one-line string) of the object.

[2504] Maximum Length: 64

[2505] boolean DenyNTLogin

[2506] Access Type: Read/Write

[2507] The DenyNTLogin property indicates the ability to access to a SQL Server™ installation for login records identifying Windows NT users or groups. When True, any Windows NT authenticated connection attempt specifying the user or group name fails authentication. When False, the Windows NT user or group is allowed access to the SQL Server™ installation on which the login is defined. Use DenyNTLogin to specifically deny access to Windows NT users and groups.

[2508] string Description

[2509] Access Type: Read-only

[2510] The Description property provides a textual description of the object.

[2511] datetime InstallDate

[2512] Access Type: Read-only

[2513] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[2514] string Language

[2515] Access Type: Read/Write

[2516] The Language property indicates the language used for a client connection using the login.

[2517] [key] string Name

[2518] Access Type: Read-only

[2519] The Name property defines the label by which the object is known. In order to use Windows NT authentication, the name of the login must be a valid NT account name in the form \\ServerName\UserName.

[2520] [key] string SQLServerName

[2521] Access Type: Read-only

[2522] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[2523] Maximum Length: 128

[2524] string Status

[2525] Access Type: Read-only

[2526] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror-resilvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[2527] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[2528] Maximum Length: 10

[2529] boolean SystemObject

[2530] Access Type: Read-only

[2531] The SystemObject property indicates whether the object is owned by Microsoft®. A value of True indicates that the object implementation is owned by Microsoft®.

[2532] uint32 Type

[2533] Access Type: Read/Write

[2534] The Type property indicates the type of authentication used. The authentication can be NT authentication, or SQL Server™ authentication. For NT authentication, the login can use the name of a user or a group.

Value	Description
0	Other NT User Authentication
1	NT Group Authentication
2	SQL Server ™ Authentication

[2535] Methods

[2536] The MSSQL_Login class supports the following methods:

Method Name	Description
GetUserName	The GetUserName method returns the database user used by the referenced login, when a connection using that login accesses the specified database.
SetPassword	The SetPassword method is used to set the password for a login that uses SQL Server ™ authentication.

[2537] Associations

[2538] MSSQL_Login is associated to MSSQL_SQLServer as the ScopedElement property of the MSSQL_SQLServerLogin association.

[2539] MSSQL_Login is associated to MSSQL_SQLServerRole as the Dependent property of the MSSQL_MemberLogin association.

[2540] MSSQL_Login is associated to MSSQL_Database as the Antecedent property of the MSSQL_LoginDefaultDatabase association.

[2541] MSSQL_Login is associated to Win32 UserAccount as the Dependent property of the MSSQL_LoginWin32UserAccount association.

[2542] MSSQL_Login is associated to MSSQL_Database as the Antecedent property of the MSSQL_DatabaseOwnerLogin association.

[2543] MSSQL_Login is associated to MSSQL_User as the Antecedent property of the MSSQL_UserLogin association.

[2544] MSSQL_Login is associated to Win32_Group as the Dependent property of the MSSQL_LoginWin32Group association.

[2545] MSSQL_Login is associated to MSSQL_Database as the Containee property of the MSSQL_DatabaseLogin association.

[2546] MSSQL_LoginDefaultDatabase: CIM_Dependency

[2547] Association Class

[2548] The MSSQL_LoginDefaultDatabase class represents an association between a login and the default database for the login.

[2549] References

[2550] [key]MSSQL_Login Antecedent

[2551] Access Type: Read-only

[2552] The Antecedent property references a SQL Server™ login record.

[2553] [key]MSSQL_Database Dependent

[2554] Access Type: Read-only

[2555] The Dependent property references the default database to connect to for the login referenced by the Antecedent property.

[2556] MSSQL_LoginWin32Group: CIM_Dependency

[2557] Association Class

[2558] The MSSQL_LoginWin32Group class represents an association between a login and the Win32 user group used for authentication by the login.

[2559] References

[2560] [key]Win32_Group Antecedent

[2561] Access Type: Read-only

[2562] The Antecedent property references the Win32 user group used for authenticating the login referenced by the Dependent property.

[2563] [key]MSSQL_Login Dependent

[2564] Access Type: Read-only

[2565] The Dependent property references a SQL Server™ login record.

[2566] MSSQL_LoginWin32UserAccount: CIM_Dependency

[2567] Association Class

[2568] The MSSQL_LoginWin32UserAccount class represents an association between a login and the Win32 user account used for authentication by the login.

[2569] References

[2570] [key]Win32_UserAccount Antecedent

[2571] Access Type: Read-only

[2572] The Antecedent property references the Win32 user account used for authenticating the login referenced by the Dependent property.

[2573] [key]MSSQL_Login Dependent

[2574] Access Type: Read-only

[2575] The Dependent property references a SQL Server™ login record.

[2576] MSSQL_MediaHeader

[2577] Abstract Class

[2578] The MSSQL_MediaHeader class represents the contents of the header record on a media. The instances of this class are returned as results of the ReadMediaHeader method on the MSSQL_BackupDevice class. A database backup performed by Microsoft® SQL Server™ can target multiple devices of a single type and can span multiple media maintained by the device. To organize media used in backup, SQL Server™ defines the media set and media family. A media label, or header record, maintains data about a media's location within a media set and media family.

[2579] Properties

[2580] datetime CreateDate

[2581] Access Type: Read-only

[2582] The CreateDate property indicates the time and date on which the media header was created. Note that creation date may be different from the

install date in cases where the object is created in one place and then installed elsewhere.

[2583] uint32 FamilyCount

[2584] Access Type: Read-only

[2585] The FamilyCount property indicates the number of families within the media set.

[2586] uint32 FamilySequenceNumber

[2587] Access Type: Read-only

[2588] The FamilySequenceNumber property indicates the ordinal position of the family within the entire media set.

[2589] string MediaDescription

[2590] Access Type: Read-only

[2591] The MediaDescription gives a text description of the media.

[2592] string MediaFamilyId

[2593] Access Type: Read-only

[2594] The MediaFamilyId indicates the system-generated unique identifier for the media family.

[2595] boolean MediaLabelPresent

[2596] Access Type: Read-only

[2597] The MediaDescription property indicates whether the media has a label present. If TRUE, the media has a label. If FALSE, the media has no label.

[2598] string MediaName

[2599] Access Type: Read-only

[2600] The MediaName property indicates the name of the media.

[2601] uint32 MediaSequenceNumber

[2602] Access Type: Read-only

[2603] The MediaSequenceNumber property indicates the ordinal position of the media within its family.

[2604] string MediaSetId

[2605] Access Type: Read-only

[2606] The MediaSetId property indicates a system-generated unique identifier for the media set. NULL when the media contains only a single media set.

[2607] string SoftwareName

[2608] Access Type: Read-only

[2609] The SoftwareName property indicates the name of the product creating the media header.

[2610] string SoftwareVendorId

[2611] Access Type: Read-only

[2612] The SoftwareVendorId property indicates the unique identifier of the manufacturer of the product creating the media header.

[2613] MSSQL_MemberDatabaseRole: CIM_Dependency

[2614] Association Class

[2615] The MSSQL_MemberDatabaseRole class associates two database roles, one being a member of the other.

[2616] References

[2617] [key]MSSQL_DatabaseRole Antecedent

[2618] Access Type: Read/Write

[2619] This property is a reference to the database role that contains the member.

[2620] [key]MSSQL_DatabaseRole Dependent

[2621] Access Type: Read/Write

[2622] This property is a reference to the database role that is the member.

[2623] MSSQL_MemberLogin: CIM_Dependency

[2624] Association Class

[2625] The MSSQL_MemberLogin class represents an association between a SQL Server™ role and a login that is a member of the role.

[2626] References

[2627] [key]MSSQL_SQLServerRole Antecedent

[2628] Access Type: Read/Write

[2629] The Antecedent property references a SQL Server™ role.

[2630] [key]MSSQL_Login Dependent

[2631] Access Type: Read/Write

[2632] The Dependent property references a login record that is a member of the SQL Server™ role referenced by the Antecedent property.

[2633] MSSQL_MemberUser: CIM_Dependency

[2634] Association Class

[2635] The MSSQL_MemberUser class represents an association between a database role and a user that is a member of the role.

[2636] References

[2637] [key]MSSQL_DatabaseRole Antecedent

[2638] Access Type: Read-only

[2639] The Antecedent property references a database role.

[2640] [key] MSSQL_User Dependent

[2641] Access Type: Read-only

[2642] The Antecedent property references the Win32 user that is a member of the database role referenced by the Dependent property.

[2643] MSSQL_MethodRtnVal

[2644] Abstract Class

[2645] The MSSQL_MethodRtnVal class is an object returned by many methods available with the WMI SQL

Server™ Administration provider. The object contains information about the success or failure of the API the provider called within SQL Server™ to execute the method. This information is separate from the success or failure results of the WMI API called by the client application.

[2646] Given this information, it is possible for the calling application to determine, if a method call fails, whether the failure occurred in the provider or in SQL Server™ and the reason for the failure. The calling application should always check the ReturnValue property of this object after executing a method to verify success.

[2647] Properties

[2648] string Description

[2649] Access Type: Read-only

[2650] The Description property contains a textual description of the error code reported in the ReturnValue property.

[2651] uint32 ReturnValue

[2652] Access Type: Read-only

[2653] The ReturnValue property reports the success or failure of the method execution within SQL Server™. ReturnValue will contain 0 if the method was executed successfully. If an error occurred, the error reported by the SQL Server™ API that was called will be returned.

[2654] string Source

[2655] Access Type: Read-only

[2656] The Source property contains a string reporting which subsystem within SQL Server™ returned the error.

[2657] MSSQL_Permission

[2658] Abstract Class

[2659] Association Class

[2660] The MSSQL_Permission class represents the permissions granted or denied to a user object, for access to a database object. The instances of this class represent only the permission that has been explicitly granted or denied to the user object. For example, if a user has permissions to access a table by virtue of being a member of a certain database role, then there will not be a permission association instance between the user and the table.

[2661] Properties

[2662] boolean Granted

[2663] Access Type: Read-only

[2664] The Granted property indicates whether the permission has been granted or denied. A value of True indicates that the permission has been granted. A value of False indicates that permission has been denied.

[2665] uint32 PrivilegeType

[2666] Access Type: Read-only

[2667] The PrivilegeType property indicates the type of privilege that has been granted or denied.

Value	Description
0	Unknown
1	Permission to query a table
2	Permission to add rows to a table
4	Permission to update rows of a table
8	Permission to delete rows of a table
16	Permission to execute a stored procedure
32	Permission to grant DRI on a table
63	All privileges applicable to the database object
128	Permission to create and own a table
256	Permission to create and own a database
512	Permission to create and own a view
1024	Permission to create and own a stored procedure
2048	Permission to backup a database
4096	Permission to create a default
8192	Permission to backup a database transaction log
16384	Permission to create a rule
32768	Permission to backup to a table
65366	Permission to create a user defined function
130944	All privileges applicable to the database

[2668] References

[2669] MSSQL_DBMSObject Element

[2670] Access Type: Read-only

[2671] The Element property references a database object such as a database or a table for which the permission are defined.

[2672] MSSQL_DBMSUserObject Grantee

[2673] Access Type: Read-only

[2674] The Grantee property references a database user object such as a user or a login for whom the permissions are defined.

[2675] MSSQL_PrimaryKey: MSSQL_CandidateKey

[2676] The MSSQL_PrimaryKey class represents a primary key of a table. A primary key must also be a candidate key of the table.

[2677] Properties

[2678] string Caption

[2679] Access Type: Read-only

[2680] The Caption property is a short textual description (one-line string) of the object.

[2681] Maximum Length: 64

[2682] boolean Clustered

[2683] Access Type: Read-only

[2684] The Clustered property indicates whether a clustered index has been used for the primary key. Clustered indexes sort and store the data rows in the table based on their key values. Non-clustered indexes have a structure that is completely separate from the data rows.

[2685] [key] string DatabaseName

[2686] Access Type: Read-only

[2687] The DatabaseName property indicates the name of the database that the key is a part of.

[2688] Maximum Length: 128

[2689] string Description

[2690] Access Type: Read-only

[2691] The Description property provides a textual description of the object.

[2692] sint32 FillFactor

[2693] Access Type: Read-only

[2694] The FillFactor property indicates the percent of each page used to store index data when the index for the key is created.

[2695] datetime InstallDate

[2696] Access Type: Read-only

[2697] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[2698] [key] string Name

[2699] Access Type: Read-only

[2700] The Name property defines the label by which the object is known. The name of a key is unique within a database.

[2701] [key] string SQLServerName

[2702] Access Type: Read-only

[2703] The SQLServerName property indicates the name of the SQL Server™ installation that the key is a part of.

[2704] Maximum Length: 128

[2705] string Status

[2706] Access Type: Read-only

[2707] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror-resilvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[2708] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[2709] Maximum Length: 10

[2710] [key] string TableName

[2711] Access Type: Read-only

[2712] The TableName property indicates the name of the table that the key is defined in.

[2713] Maximum Length: 128

[2714] Methods

[2715] The MSSQL_PrimaryKey class supports the following methods:

Method Name	Description
Create	The Create method is used to create a new primary key instance.
RebuildIndex	The RebuildIndex method re-creates an index for a candidate key constraint.
Rename	The Rename method is used to rename a primary key instance.

[2716] MSSQL_Process: CIM_Process

[2717] The MSSQL_Process class represents SQL Server™ processes. Note that these are not the same as an operating system's notion of a process. These are the processes identified by the SQL Server™ and assigned a SQL Server™ process ID by SQL Server™.

[2718] Properties

[2719] uint32 BlockedProcessID

[2720] Access Type: Read-only

[2721] The BlockedProcessID property represents the id of a process that is being blocked by the process.

[2722] string Caption

[2723] Access Type: Read-only

[2724] The Caption property is a short textual description (one-line string) of the object.

[2725] Maximum Length: 64

[2726] string ClientName

[2727] Access Type: Read-only

[2728] The ClientName property indicates the name of the client application that started the SQL Server™ process.

[2729] string Command

[2730] Access Type: Read-only

[2731] The Command property indicates the abbreviated indicator of current command. When no command is current, it has a value of AWAITING COMMAND.

[2732] uint32 CPUTime

[2733] Access Type: Read-only

[2734] The CPUTime property indicates the cumulative CPU usage time of the process.

[2735] string CreationClassName

[2736] Access Type: Read-only

[2737] The inherited CreationClassName property indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property

allows all instances of this class and its subclasses to be uniquely identified.

[2738] datetime CreationDate

[2739] Access Type: Read-only

[2740] Time that the process began executing

[2741] string CSCreationClassName

[2742] Access Type: Read-only

[2743] The inherited CSCreationClassName property is a string indicating the class of the computer system.

[2744] string CSName

[2745] Access Type: Read-only

[2746] The inherited CSName property is a string indicating the name of the computer system.

[2747] string DatabaseName

[2748] Access Type: Read-only

[2749] The DatabaseName property represents the database that is currently being used by the process.

[2750] string Description

[2751] Access Type: Read-only

[2752] The Description property provides a textual description of the object.

[2753] uint16 ExecutionState

[2754] Access Type: Read-only

[2755] Indicates the current operating condition of the process.

Value	Description
0	Unknown
1	Other
2	Ready
3	Running
4	Blocked
5	Suspended Blocked
6	Suspended Ready

[2756] [key] string Handle

[2757] Access Type: Read-only

[2758] A string used to identify the process. A process ID is a process handle.

[2759] string HostName

[2760] Access Type: Read-only

[2761] The HostName property indicates the name of the client workstation that started the SQL Server™ process.

[2762] datetime InstallDate

[2763] Access Type: Read-only

[2764] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[2765] uint64 KernelModeTime

[2766] Access Type: Read-only

[2767] Time in kernel mode, in milliseconds. If this information is not available, a value of 0 should be used.

[2768] Units: Milliseconds

[2769] string Login

[2770] Access Type: Read-only

[2771] The Login property represents the login used by the process to connect to SQL Server™.

[2772] uint32 MemoryUsage

[2773] Access Type: Read-only

[2774] The MemoryUsage property indicates the number of pages in the procedure cache that are currently allocated to this process. A negative number indicates that the process is freeing memory allocated by another process.

[2775] string Name

[2776] Access Type: Read-only

[2777] The Name property defines the label by which the object is known.

[2778] string OSCreationClassName

[2779] Access Type: Read-only

[2780] The inherited OSCreationClassName property indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.

[2781] string OSName

[2782] Access Type: Read-only

[2783] The inherited OSName property serves as key of an operating system instance within a computer system.

[2784] uint32 Priority

[2785] Access Type: Read-only

[2786] Priority indicates the urgency or importance of execution of a process. If a priority is not defined for a process, a value of 0 should be used.

[2787] [key] string SQLServerName

[2788] Access Type: Read-only

[2789] The SQLServerName property indicates the name of the SQL Server™ installation in which the process exists.

[2790] Maximum Length: 128

[2791] string State

[2792] Access Type: Read-only

[2793] The State property indicates whether the process is running or sleeping.

[2794] string Status

[2795] Access Type: Read-only

[2796] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[2797] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[2798] Maximum Length: 10

[2799] datetime TerminationDate

[2800] Access Type: Read-only

[2801] Time that the process was stopped or terminated.

[2802] uint64 UserModeTime

[2803] Access Type: Read-only

[2804] Time in user mode, in milliseconds. If this information is not available, a value of 0 should be used.

[2805] Units: Milliseconds

[2806] uint64 WorkingSetSize

[2807] Access Type: Read-only

[2808] The amount of memory in bytes that a process needs to execute efficiently, for an operating system that uses page-based memory management. If an insufficient amount of memory is available (<working set size), thrashing will occur. If this information is not known, NULL or 0 should be entered. If this data is provided, it could be monitored to understand a process' changing memory requirements as execution proceeds.

[2809] Units: Bytes

[2810] Methods

[2811] The MSSQL_Process class supports the following methods:

Method Name	Description
ProcessInputBuffer	The ProcessInputBuffer method returns the contents of the memory used by a Microsoft ® SQL Server ™ process for input.
ProcessOutputBuffer	The ProcessOutputBuffer method returns the contents of the memory used by a Microsoft ® SQL Server ™ process for output.

[2812] MSSQL_ProviderStatus:_ExtendedStatus

[2813] Abstract Class

[2814] The MSSQL_ProviderStatus class represents the error status information returned by the WMI SQL Server™ Administration provider.

[2815] Properties

[2816] string Description

[2817] Access Type: Read-only

[2818] The Description property contains a textual description of the returned status information.

[2819] string Object

[2820] Access Type: Read-only

[2821] The Object property indicates the object that the provider was working on when the failure occurred.

[2822] string Operation

[2823] Access Type: Read-only

[2824] The Operation property indicates the operation that the provider was performing on when the failure occurred.

[2825] string ParameterInfo

[2826] Access Type: Read-only

[2827] The ParameterInfo property identifies one or more parameters that were involved in the error or status change.

[2828] string Property

[2829] Access Type: Read-only

[2830] The Property property indicates the property that the provider was working on when the failure occurred.

[2831] string ProviderName

[2832] Access Type: Read-only

[2833] The ProviderName property identifies the provider that caused or reported the error or status change. If a provider was not involved, this string is set to "Windows Management".

[2834] string Routine

[2835] Access Type: Read-only

[2836] The Routine property indicates the routine in which the failure occurred.

[2837] string Source

[2838] Access Type: Read-only

[2839] The Source property indicates the COM source for the error.

[2840] uint32 StatusCode

[2841] Access Type: Read-only

[2842] The StatusCode property contains an error or information code for an operation. This can be any user-defined code, but the value 0 is usually reserved to indicate success.

[2843] MSSQL_ReferencedKey: CIM_Dependency

[2844] Association Class

[2845] The MSSQL_ReferencedKey class represents an association between a foreign key and the candidate key that the foreign key references.

[2846] References

[2847] [key]MSSQL_CandidateKey Antecedent

[2848] Access Type: Read-only

[2849] The Antecedent property references a candidate key in the SQL Server™ database.

[2850] [key]MSSQL_ForeignKey Dependent

[2851] Access Type: Read-only

[2852] The Dependent property references a foreign key that references the candidate key referenced by the Antecedent property.

[2853] MSSQL_ReferencedTable: CIM_Dependency

[2854] Association Class

[2855] The MSSQL_ReferencedTable class represents an association between a foreign key and the table that contains the primary key referenced by the foreign key.

[2856] References

[2857] [key]MSSQL_Table Antecedent

[2858] Access Type: Read-only

[2859] The Antecedent property references a table in the SQL Server™ database.

[2860] [key]MSSQL_ForeignKey Dependent

[2861] Access Type: Read-only

[2862] The Dependent property references a foreign key that references a candidate key contained in the table referenced by the Antecedent property.

[2863] MSSQL_RegistrySetting MSSQL_Setting

[2864] The MSSQL_RegistrySetting class represents the installation and run-time parameters of SQL Server™ stored in the registry.

[2865] Properties

[2866] string ADSP

[2867] Access Type: Read-only

[2868] The ADSP property specifies an AppleTalk (ADSP) service object name on a computer running Microsoft® SQL Server™.

[2869] string AgentLogFile

[2870] Access Type: Read/Write

[2871] The AgentLogFile property specifies the path and file name for the SQL Server™ Agent log.

[2872] boolean AutostartDTC

[2873] Access Type: Read/Write

[2874] The AutostartDTC property controls Microsoft® Distributed Transaction Coordinator service (MSDTC) behavior on computer start. If TRUE, the MSDTC service is started when the computer starts. If FALSE, the MSDTC service must be started manually.

[2875] boolean AutostartLicensing

[2876] Access Type: Read/Write

[2877] The AutostartLicensing property indicates the license logging service behavior for Microsoft® SQL Server™. If TRUE, the license logging service is started when SQL Server™ starts. If FALSE, license logging must be started manually.

[2878] boolean AutoStartMail

[2879] Access Type: Read/Write

[2880] The AutoStartMail property indicates whether the post office is started automatically when SQL Server™ starts. If TRUE, an attempt to start the SQL Server™ workgroup post office is made when SQL Server™ starts. If FALSE, no attempt is made to start the post office when SQL Server™ starts. SQL Server™ mail has to be started manually.

[2881] string BackupDirectory

[2882] Access Type: Read/Write

[2883] The BackupDirectory property specifies the backup directory.

[2884] string Caption

[2885] Access Type: Read-only

[2886] A short textual description (one-line string) of the MSSQL_RegistrySetting object.

[2887] Maximum Length: 64

[2888] boolean CaseSensitive

[2889] Access Type: Read-only

[2890] The CaseSensitive property indicates the comparison method for multibyte character data is case sensitive or not. If TRUE, a character comparison for equality and order is case-sensitive. For example, A is less than a. If FALSE, character comparison for equality and order is not case-sensitive.

[2891] string CharacterSet

[2892] Access Type: Read-only

[2893] The CharacterSet property identifies the code page used by the Microsoft® SQL Server™ installation to interpret multibyte character data.

[2894] string Description

[2895] Access Type: Read-only

[2896] A textual description of the MSSQL_RegistrySetting object.

[2897] string ErrorLogPath

[2898] Access Type: Read/Write

[2899] The ErrorLogPath property specifies the operating system path and file name to be used for the Microsoft® SQL Server™ error log. The value of ErrorLogPath by itself is not the name of the operating system file. SQL Server™ appends an integer to the value specified by ErrorLogPath, as an extension to indicate the current error log file.

[2900] string MailAccountName

[2901] Access Type: Read/Write

[2902] The MailAccountName property specifies the Microsoft® Exchange client account used by SQL Mail.

[2903] string MailPassword

[2904] Access Type: Read/Write

[2905] The MailPassword property specifies the Microsoft® Exchange client account password for SQL Mail.

[2906] string MasterDBPath

[2907] Access Type: Read/Write

[2908] The MasterDBPath property specifies the full path and file name of the operating system file containing the master database.

[2909] string NP

[2910] Access Type: Read-only

[2911] The NP property specifies the pipe name when using named pipe protocol on an instance of Microsoft® SQL Server™.

[2912] boolean NTEventLogging

[2913] Access Type: Read-only

[2914] The NTEventLogging property specifies whether Microsoft® SQL Server ™ uses the Microsoft® Windows NT application log. If TRUE, SQL Server™ sends all events to the Windows NT application log and the SQL Server™ error log. If FALSE, SQL Server ™ sends events only to the SQL Server™ error log.

[2915] sint32 NumberOfProcessors

[2916] Access Type: Read-only

[2917] The NumberOfProcessors property returns the number of computer processing units (CPUs) available to Microsoft® SQL Server™ on the server.

[2918] uint32 PerfMonMode

[2919] Access Type: Read/Write

[2920] The PerfMonMode property controls Windows NT Performance Monitor polling behavior when the monitor is launched. A value of Continuous indicates that performance monitor polls for statistics using the operating system default time slice. A value of On Demand indicates that performance monitor polls for statistics when directed to do so by the user.

Value	Description
0	Continuous
1	On Demand

[2921] string RegisteredOrganization

[2922] Access Type: Read-only

[2923] The RegisteredOrganization property returns the company name supplied by the installer during Microsoft® SQL Server™ installation.

[2924] string RegisteredOwner

[2925] Access Type: Read-only

[2926] The RegisteredOwner property returns the installer name supplied during Microsoft® SQL Server™ installation.

[2927] boolean ReplicationInstalled

[2928] Access Type: Read-only

[2929] The ReplicationInstalled property returns TRUE when components supporting replication are installed on a server running Microsoft® SQL Server™.

[2930] boolean RPCencrypt

[2931] Access Type: Read-only

[2932] The RpcEncrypt property specifies whether Microsoft®® Windows NT® RPC encryption is enabled (using the Multiprotocol Net-Library) on a computer running Microsoft® SQL Server™.

[2933] string RPClist

[2934] Access Type: Read-only

[2935] The RpcList property returns a Microsoft® Windows NT RPC protocol list.

[2936] uint32 RPCmaxCalls

[2937] Access Type: Read-only

[2938] The RpcMaxCalls property specifies the maximum number of Microsoft® Windows NT RPC connections that can be active.

[2939] uint32 RPCminCalls

[2940] Access Type: Read-only

[2941] The RpcMinCalls property specifies the maximum number of Microsoft® Windows NT RPC connections that can be active.

[2942] [key] string SettingID

[2943] Access Type: Read-only

[2944] The identifier by which the object is known.

[2945] Maximum Length: 256

[2946] boolean SNMP

[2947] Access Type: Read-only

[2948] The SNMP property indicates Whether Simple Network Management Protocol (SNMP) is installed on an instance of Microsoft® SQL Server™.

[2949] string SNMPCurrentVersion

[2950] Access Type: Read-only

[2951] The SNMPCurrentVersion property specifies the version of Simple Network Management Protocol (SNMP) currently installed on an instance of Microsoft® SQL Server™.

[2952] string SNMPExtensionAgentsData

[2953] Access Type: Read-only

[2954] The SNMPExtensionAgentsData property retrieves or sets the value of the SNMPExtensionAgents property. To set the SNMPExtensionAgentsData property, you must be a member of the sysadmin fixed server role.

[2955] string SortOrder

[2956] Access Type: Read-only

[2957] The SortOrder property returns a string describing the character set used and ordering applied for a Microsoft® SQL Server™ installation.

[2958] boolean SpxFlag

[2959] Access Type: Read-only

[2960] The SpxFlag property indicates whether an NWLink IPX/SPX flag is set on an instance of Microsoft® SQL Server™.

[2961] uint32 SpxPort

[2962] Access Type: Read-only

[2963] The SpxPort property specifies the NWLink IPX/SPX port number on an instance of Microsoft® SQL Server™.

[2964] string SpxServiceName

[2965] Access Type: Read-only

[2966] The SpxServiceName property specifies the name of the NWLink IPX/SPX service on an instance of Microsoft® SQL Server™.

[2967] string SQLDataRoot

[2968] Access Type: Read/Write

[2969] The SQLDataRoot property identifies the default operating-system directory implementing storage for Microsoft® SQL Server™ system user-defined databases.

[2970] string SQLRootPath

[2971] Access Type: Read/Write

[2972] The SQLRootPath property identifies the operating-system directory specified as the root directory for Microsoft® SQL Server™ installation.

[2973] boolean SuperSocketEncrypt

[2974] Access Type: Read-only

[2975] The SuperSocketEncrypt property specifies whether Super Sockets Net-Library encryption is enabled on an instance of Microsoft® SQL Server™.

[2976] SQL Server™ 2000 only

[2977] string SuperSocketList []

[2978] Access Type: Read-only

[2979] The SuperSocketList property returns a super socket protocol list.

[2980] SQL Server™ 2000 only

[2981] sint32 TapeLoadWaitTime

[2982] Access Type: Read/Write

[2983] The TapeLoadWaitTime property specifies a number of minutes a Microsoft® SQL Server™ backup or restore operation will wait, before timing out, when trying to write to or read from an indicated tape media. A value of −1 indicates that the backup or restore operation will not time out. A value of 0 indicates that the backup or restore operation will attempt to access the tape device exactly one time. Any positive integer indicates the number of minutes for which the backup or restore operation will attempt to access the tape device.

[2984] boolean TcpFlag

[2985] Access Type: Read-only

[2986] The TcpFlag property specifies whether the TCP/IP Sockets Net-Libraries hide flag is set on a computer running an instance of Microsoft® SQL Server™.

[2987] SQL Server™ 2000 only

[2988] string TcpPort

[2989] Access Type: Read/Write

[2990] The TcpPort property specifies the TCP/IP Sockets Net-Libraries port number on an instance of Microsoft® SQL Server™.

[2991] SQL Server™ 2000 only

[2992] string VinesGroupName

[2993] Access Type: Read-only

[2994] The VinesGroupName property specifies the Banyan Vines Net-Library group name on a computer running Microsoft® SQL Server™.

[2995] SQL Server ™ 2000 only

[2996] string VinesItemName

[2997] Access Type: Read-only

[2998] The VinesItemName property specifies the Banyan Vines Net-Library item name on a computer running Microsoft® SQL Server™.

[2999] SQL Server ™ 2000 only

[3000] string VinesorgName

[3001] Access Type: Read-only

[3002] The VinesOrgName property specifies the Banyan Vines Net-Library organization name on a computer running Microsoft® SQL Server™.

[3003] SQL Server™ 2000 only

[3004] string WSProxyAddress

[3005] Access Type: Read-only

[3006] The WSProxyAddress property specifies the WinSock proxy server address on a computer running Microsoft® SQL Server™.

[3007] SQL Server ™ 2000 only

[3008] uint32 WSProxyPort

[3009] Access Type: Read-only

[3010] The WSProxyPort property specifies the WinSock proxy server port number on a computer running Microsoft® SQL Server™.

[3011] SQL Server ™ 2000 only

[3012] Associations

[3013] MSSQL_RegistrySetting is associated to MSSQL_SQLServer as the Setting property of the MSSQL_SQLServerRegistry association.

[3014] MSSQL_RestoreSetting: MSSQL_Setting

[3015] The MSSQL_RestoreSetting class is used to specify the behavior of a restore operation for a SQL Server database or log. The class is also used to specify the behavior of the verify operation for a SQL Server backup. An instance of this class is passed as an argument to the SQLRestore method and the SQLVerify method of the MSSQL_SQLServer class.

[3016] Properties

[3017] string BackupDevice []

[3018] Access Type: Read/Write

[3019] The BackupDevice property specifies one or more backup devices used as a database restore source. Only one medium type can be specified for any restore operation, but multiple media may be specified. Set the BackupDevice property to specify one or more SQL Server™ backup devices as the backup medium.

[3020] string BackupSetName

[3021] Access Type: Read/Write

[3022] The BackupSetName property identifies a unit of backup work. The BackupSetName property value is limited to 128 characters.

[3023] Maximum Length: 128

[3024] string Caption

[3025] Access Type: Read-only

[3026] A short textual description (one-line string) of the setting object. Maximum Length: 64

[3027] string DatabaseFiles []

[3028] Access Type: Read/Write

[3029] The DatabaseFiles property identifies operating system files storing table or index data as targets of a restore operation. Setting DatabaseFiles directs restore processing to include only those operating system files listed. To specify an operating system file, use its logical name as visible to SQL Server™, not its physical or operating system name.

[3030] string DatabaseFileGroups []

[3031] Access Type: Read/Write

[3032] The DatabaseFileGroups property identifies filegroups targeted by a restore operation.

[3033] string DatabaseName

[3034] Access Type: Read/Write

[3035] The DatabaseName identifies the target database for a restore. The property is a required element and must be set prior to calling the SQLRestore method of the MSSQL_SQLServer class.

[3036] string Description

[3037] Access Type: Read-only

[3038] A textual description of the object.

[3039] uint32 FileNumber

[3040] Access Type: Read/Write

[3041] The FileNumber property identifies a backup set by ordinal location on the backup medium.

[3042] boolean LastRestore

[3043] Access Type: Read/Write

[3044] The LastRestore property identifies the last transaction log unit in a chain of log backups. When more than one log unit exists for restoration, it is imperative that the administrator specify that more than one log unit will be restored. After SQL Server™ processes the last log unit in the chain, no log backups made after that unit can be applied. Set the LastRestore property to False when restoring a backup unit that is not the last in a backup chain. Set the LastRestore property to TRUE when restoring a backup unit that is the last in the chain.

[3045] boolean LoadHistory

[3046] Access Type: Read/Write

[3047] The LoadHistory property controls the behavior of the operation that verifies the integrity of a SQL Server backup. If TRUE, msdb backup history tables are updated with backup set data when the SQLVerify method of the MSSQL_SQLServer object directs backup set verification. If FALSE, history tables are not altered when SQLVerify is used.

[3048] string MediaName

[3049] Access Type: Read/Write

[3050] The MediaName property provides informative text to aid in identification of a backup set. The MediaName property is written to a tape media when the media is initialized.

[3051] string RelocateFile []

[3052] Access Type: Read/Write

[3053] The RelocateFile property specifies database logical file names and operating system physical file names used to redirect database storage when a SQL Server™ database is restored to a new physical location.

[3054] boolean ReplaceDatabase

[3055] Access Type: Read/Write

[3056] The ReplaceDatabase property directs a restore operation when a new image of the restored database is needed. If True, a new image of the database is created. The image is created regardless of the presence of an existing database with the same name. If False (default), a new image of the database is not created by the restore operation. The database targeted by the restore operation must exist on the Microsoft® SQL Server™ installation.

[3057] boolean Restart

[3058] Access Type: Read/Write

[3059] The Restart property controls restore operation behavior when the restore operation specified by the object was started and interrupted. If TRUE, Microsoft® SQL Server™ attempts to continue processing on a partial backup or restore operation. If FALSE, SQL Server™ restarts an interrupted backup or restore operation at the beginning of the backup set.

[3060] datetime RestoreTillTime

[3061] Access Type: Read/Write

[3062] The RestoreTillTime property sets an endpoint for database log restoration.The RestoreTillTime setting is evaluated only when restoring to recover a database transaction log.

[3063] [key] string SettingID

[3064] Access Type: Read/Write

[3065] The identifier by which the object is known.

[3066] Maximum Length: 256

[3067] [key] string SQLServerName

[3068] Access Type: Read/Write

[3069] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[3070] Maximum Length: 128

[3071] string StandbyFile

[3072] Access Type: Read/Write

[3073] The StandbyFile property specifies the name of an undo file used as part of a SQL Server™ installation imaging strategy.

[3074] uint32 TargetType

[3075] Access Type: Read/Write

[3076] The TargetType property controls the type of restore operation to be performed. SQL Server can restore a database, one or more operating system files containing table or index data, or part or all of the transaction log of a database. The value of the TargetType property determines applicability and interpretation of related MSSQL_RestoreSetting object properties. For example, when TargetType is Files, either the DatabaseFileGroups or DatabaseFiles property must specify filegroups or files to be restored.

Value	Description	Explanation
0	Database	Restore the entire database.
1	Files	Restore only specified files.
2	Logs	Restore only the database transaction log.

[3077] boolean UnloadTapeAfter

[3078] Access Type: Read/Write

[3079] The UnloadTapeAfter property controls tape media handling on completion of a restore operation. If TRUE, the tape media in the tape device(s) is rewound and unloaded when the operation completes. If FALSE (default), no attempt is made to rewind and unload the tape media.

[3080] MSSQL_Role: MSSQL_DBMSUscrObject

[3081] Abstract Class

[3082] The MSSQL_Role class represents a database role or a SQL Server™ role. Roles are used to establish groups of users with similar security attributes. Permissions can be granted by role, simplifying security planning and administration.

[3083] Properties

[3084] string Caption

[3085] Access Type: Read-only

[3086] The Caption property is a short textual description (one-line string) of the object.

[3087] Maximum Length: 64

[3088] string Description

[3089] Access Type: Read-only

[3090] The Description property provides a textual description of the object.

[3091] datetime InstallDate

[3092] Access Type: Read-only

[3093] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[3094] string Name

[3095] Access Type: Read-only

[3096] The Name property defines the label by which the object is known. When subclassed, the Name property can be overridden to be a Key property.

[3097] string Status

[3098] Access Type: Read-only

[3099] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[3100] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[3101] Maximum Length: 10

[3102] MSSQL_Rule: MSSQL_Constraint

[3103] The MSSQL_Rule class represents a single Microsoft® SQL Server™ data-integrity rule. SQL Server™ offers several mechanisms for ensuring data integrity. A SQL Server™ rule is a Transact-SQL condition_expression syntax element that defines a data-integrity constraint. A rule can be bound to a column or user-defined data type.

[3104] Properties

[3105] string Caption

[3106] Access Type: Read-only

[3107] The Caption property is a short textual description (one-line string) of the object.

[3108] Maximum Length: 64

[3109] datetime CreateDate

[3110] Access Type: Read-only

[3111] The CreateDate property indicates the time and date on which the rule was created. Note that creation date may be different from the install date in cases where the object is created in one place and then installed elsewhere.

[3112] [key] string DatabaseName

[3113] Access Type: Read-only

[3114] The DatabaseName property indicates the name of the database that the object is a part of.

[3115] Maximum Length: 128

[3116] string Description

[3117] Access Type: Read-only

[3118] The Description property provides a textual description of the object.

[3119] datetime InstallDate

[3120] Access Type: Read-only

[3121] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[3122] [key] string Name

[3123] Access Type: Read-only

[3124] The Name property defines the label by which the object is known.

[3125] [key] string SQLServerName

[3126] Access Type: Read-only

[3127] The SQLServerName property indicates the name of the SQL Server™ installation that the rule is a part of.

[3128] Maximum Length: 128

[3129] string Status

[3130] Access Type: Read-only

[3131] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[3132] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[3133] Maximum Length: 10

[3134] string Text

[3135] Access Type: Read/Write

[3136] The Text property indicates the Transact-SQL script validating data integrity for the columns bound by the rule. Note that there is a special requirement for the name of the object is specified in the CREATE statement. The name of the object has to be in the form that includes the name of the owner. For example, in order to create an object named "Some Object", one would need to specify it as [dbo].[Some Object].

[3137] Methods

[3138] The MSSQL_Rule class supports the following methods:

Method Name	Description
Rename	The Rename method is used to rename a rule instance.

[3139] Associations

[3140] MSSQL_Rule is associated to MSSQL_Database as the ScopedElement property of the MSSQL_DatabaseRule association.

[3141] MSSQL_Rule is associated to MSSQL_Column as the Antecedent property of the MSSQL_ColumnRule association.

[3142] MSSQL_Rule is associated to MSSQL_UserDatatype as the Antecedent property of the MSSQL_UserDatatypeRule association.

[3143] MSSQL_Scope

[3144] Abstract Class

[3145] Association Class

[3146] The MSSQL_Scope class represents an association between a two logical elements, where one is scoped within the other. Scoped implies that the name of the scoped element has to be unique within the scope of the scoping element.

[3147] References

[3148] CIM_LogicalElement ScopedElement

[3149] Access Type: Read-only

[3150] The ScopedElement references the logical element that is scoped within the element referenced by the ScopingElement.

[3151] CIM_LogicalElement ScopingElement

[3152] Access Type: Read-only

[3153] The ScopingElement references the logical element that acts as the scope for the element referenced by the ScopedElement.

[3154] MSSQL_Setting: CIM_Setting

[3155] Abstract Class

[3156] The MSSQL_Setting class represents the settings that are used to configure a SQL Server™ installation.

[3157] Properties

[3158] string Caption

[3159] Access Type: Read-only

[3160] A short textual description (one-line string) of the setting object.

[3161] Maximum Length: 64

[3162] string Description

[3163] Access Type: Read-only

[3164] A textual description of the setting object.

[3165] string SettingID

[3166] Access Type: Read-only

[3167] The identifier by which the setting object is known.

[3168] Maximum Length: 256

[3169] MSSQL_SQLServer: CIM_LogicalElement

[3170] The MSSQL_SQLServer class represents instances of Microsoft® SQL Server™.

[3171] Properties

[3172] string Caption

[3173] Access Type: Read-only

[3174] The Caption property is a short textual description (one-line string) of the object.

[3175] Maximum Length: 64

[3176] boolean Clustered

[3177] Access Type: Read-only

[3178] The Clustered property indicates whether the SQL Server™ installation is part of a cluster of SQL Server™ instances.

[3179] sint32 CodePage

[3180] Access Type: Read-only

[3181] The CodePage property returns the identifier of the character set used by the referenced Microsoft® SQL Server™ installation. A character set (code page) is used to interpret multibyte character data, determining character value, and therefore sort order. Code page settings apply only to multibyte character data, not to Unicode character data. A code page is chosen for a SQL Server™ installation during setup.

[3182] string Collation

[3183] Access Type: Read-only

[3184] The Collation property specifies the column-level collation of a string data type in the database. Collation is a read-only property, and is used to retrieve the current collation for string data types.

[3185] string Description

[3186] Access Type: Read-only

[3187] The Description property provides a textual description of the object.

[3188] boolean FullTextInstalled

[3189] Access Type: Read-only

[3190] The IsFullTextInstalled property returns TRUE when the Microsoft Search service has been successfully installed on an instance of Microsoft® SQL Server™.

[3191] datetime InstallDate

[3192] Access Type: Read-only

[3193] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[3194] string Language

[3195] Access Type: Read/Write

[3196] The Language property indicates the language used by a server running Microsoft® SQL Server™. SQL Server™ language records direct display of error and status messages by choosing localized text for messages and localized formatting for date values. Set the Language property of the SQLServer object to alter the default language record used by all users on the referenced server.

[3197] [key] string Name

[3198] Access Type: Read-only

[3199] The Name property defines the label by which the object is known.

[3200] uint32 Package

[3201] Access Type: Read-only

[3202] The Package property is a long integer value identifying the Microsoft® SQL Server™ product installed on the referenced server running SQL Server™.

Value	Description
0	Unknown
1	Office
2	Standard
3	Enterprise
4	Microsoft ® Data Engine

[3203] string Status

[3204] Access Type: Read-only

[3205] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[3206] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[3207] Maximum Length: 10

[3208] string TrueLogin

[3209] Access Type: Read-only

[3210] The TrueLogin property indicates the name of the login used by the current connection. When a connection relies on Microsoft® Windows NT user or group name mapping for authentication, the TrueLogin property returns the SQL Server™ login used by the connection regardless of the login specified when the connection was established.

[3211] string TrueName

[3212] Access Type: Read-only

[3213] The TrueName property indicates the name of the SQL Server™ installation. By default, the SQL Server™ installation receives the network name of the server running SQL Server™.

[3214] uint32 UserProfile

[3215] Access Type: Read-only

[3216] The UserProfile property returns a high-level role description for the login used by the current connection.

Value	Description
0	No Privileges
1	System Administrator Privileges
2	Permission to Create Database
4	Permission to execute Extended Stored Procedures
7	All permissions

[3217] sint32 VersionMajor

[3218] Access Type: Read-only

[3219] The VersionMajor property returns the portion of a component version identifier to the left of the first decimal point in the identifier.

[3220] sint32 VersionMinor

[3221] Access Type: Read-only

[3222] The VersionMinor property returns the portion of a component version identifier to the right of the first decimal point in the identifier.

[3223] string VersionString

[3224] Access Type: Read-only

[3225] Version of the SQL Server™. This is the value returned by the Microsoft® SQL Server™ scalar function @@VERSION.

[3226] Methods

[3227] The MSSQL_SQLServer class supports the following methods:

Method Name	Description
AddStartParameter	The AddStartParameter method appends a Microsoft ® SQL Server ™ service startup option to those currently used by the service.
AttachDB	The AttachDB method makes a database visible to a Microsoft ® SQL Server ™ installation.
AttachDBWithSingleFile	The AttachDB WithSingleFile method makes a database visible to a Microsoft ® SQL Server ™ installation.
DetachDB	The DetachDB method makes a database invisible to a Microsoft ® SQL Server ™ installation.
ExecuteImmediate	The ExecuteImmediate method allows any SQL Server ™ command to be executed, as long as the command doesn't return result sets.
KillDatabase	The KillDatabase method drops a database from the Microsoft ® SQL Server ™ installation referenced, regardless of the status or availability of the database.
KillProcess	The KillProcess method terminates the Microsoft ® SQL Server ™ process identified.
ListCollations	The ListCollations method returns all valid Microsoft ® SQL Server ™ collation names.
ServerLoginMode	The ServerLoginMode method returns the default login mode for the specified server.
SQLBackup	The SQLBackup method starts the backup operation for the SQL Server ™ installation.
SQLRestore	The SQLRestore method starts the restore operation for the SQL Server ™ installation.
SQLVerifv	The SQL Verify method checks the backup media specified, ensuring that a backup set is readable and complete.
Start	The Start method starts the SQL Server ™ service.
Stop	The Stop method halts execution of the SQL Server ™ service immediately.
UnloadODSDLL	The UnloadODSDLL method frees a dynamic-link library (DLL) loaded into Microsoft ® SQL Server ™ memory.

[3228] Associations

[3229] MSSQL_SQLServer is associated to MSSQL_Database as the GroupComponent property of the MSSQL_SQLServerDatabase association.

[3230] MSSQL_SQLServer is associated to MSSQL_IntegratedSecuritySetting as the Element property of the MSSQL_SQLServerIntegratedSecuritySetting association.

[3231] MSSQL_SQLServer is associated to MSSQL_Login as the ScopingElement property of the MSSQL_SQLServerLogin association.

[3232] MSSQL_SQLServer is associated to MSSQL_ConfigValue as the Element property of the MSSQL_SQLServerConfigValue association.

[3233] MSSQL_SQLServer is associated to MSSQL_SQLServerRole as the ScopingElement property of the MSSQL_SQLServerServerRole association.

[3234] MSSQL_SQLServer is associated to MSSQL_RegistrySetting as the Element property of the MSSQL_SQLServerRegistry association.

[3235] MSSQL_SQLServer is associated to MSSQL_SQLServerConnectionSetting as the Element property of the MSSQL_SQLServerSQLServerConnectionSetting association.

[3236] MSSQL_SQLServer is associated to MSSQL_BackupDevice as the Dependent property of the MSSQL_SQLServerBackupDevice association.

[3237] MSSQL_SQLServer is associated to MSSQL_LanguageSetting as the Element property of the MSSQL_SQLServerLanguageSetting association.

[3238] MSSQL_SQLServer is associated to MSSQL_ErrorLog as the Dependent property of the MSSQL_SQLServerErrorLog association.

[3239] MSSQL_SQLServer is associated to MSSQL_User as the Container property of the MSSQL_SQLServerUser association.

[3240] MSSQL_SQLServerBackupDevice: CIM_Dependency

[3241] Association Class

[3242] The MSSQL_SQLServerBackupDevice class represents an association between a SQL Server™ installation and a backup device known to SQL Server™.

[3243] References

[3244] [key]MSSQL_BackupDevice Antecedent

[3245] Access Type: Read-only

[3246] The Antecedent property references a backup device known to SQL Server™.

[3247] [key]MSSQL_SQLServer Dependent

[3248] Access Type: Read-only

[3249] The Dependent property references a SQL Server™ installation.

[3250] MSSQL_SQLServerConfigValue: CIM_ElementSetting

[3251] Association Class

[3252] The MSSQL_SQLServerConfigValue class represents an association between a SQL Server™ installation and the configured value settings for the installation.

[3253] References

[3254] [key]MSSQL_SQLServer Element

[3255] Access Type: Read-only

[3256] The Element property references a SQL Server™ installation.

[3257] [key]MSSQL_ConfigValue Setting

[3258] Access Type: Read-only

[3259] The Setting property references configuration settings for the SQL Server™ installation.

[3260] MSSQL_SQLServerConnectionSetting: MSSQL_Setting

[3261] The MSSQL_SQLServerConnectionSetting class represents the default connection settings used by the WMI provider to connect to SQL DMO. These default settings are used by the WMI provider when the appropriate parameters in the WMI context object are not set. The settings are stored as static instances in the repository.

[3262] Properties

[3263] boolean AnsiNulls

[3264] Access Type: Read/Write

[3265] The AnsiNulls property reports the NULL acceptance behavior for new columns. By default, SQL Server™ creates columns that do not accept NULL when the user does not explicitly declare the ability to accept NULL. Further, SQL Server™ returns TRUE when evaluating the expression NULL=NULL. These default behaviors are nonstandard. When AnsiNulls is TRUE, new columns accept NULL by default and any comparison of NULL to any other value, including NULL, returns NULL. The AnsiNulls property affects NULL handling behaviors for the user's connection only and overrides any database specific settings for column creation and NULL comparison.

[3266] SQL Server™ Default value: FALSE

[3267] string ApplicationName

[3268] Access Type: Read/Write

[3269] The ApplicationName property identifies the client application connected to Microsoft® SQL Server™. The ApplicationName property is visible on the SQL Server™ installation when tools such as SQL Server™ Profiler are used to investigate server activity.

[3270] SQL Server™ Default value: A unique string generated by SQL Server™

[3271] boolean AutoReConnect

[3272] Access Type: Read/Write

[3273] The AutoReConnect property controls SQL Server™ behavior when the provider loses its connection to a SQL Server™ installation. If TRUE, the provider attempts to reconnect if it loses its connection at any time. If FALSE, the provider does not attempt to reconnect a lost connection.

[3274] SQL Server™ Default value: TRUE

[3275] sint32 BlockingTimeout

[3276] Access Type: Read/Write

[3277] The BlockingTimeout property specifies a timeout interval for resource requests that are blocked due to conflicting resource lock requests.

[3278] Units: Milliseconds

[3279] SQL Server™ Default value: 10000 (10 seconds)

[3280] string Caption

[3281] Access Type: Read-only

[3282] A short textual description (one-line string) of the object.

[3283] Maximum Length: 64

[3284] string CommandTerminator

[3285] Access Type: Read/Write

[3286] The CommandTerminator property specifies the Transact-SQL batch delimiter. The default batch delimiter is GO.

[3287] SQL Server™ Default value: "GO"

[3288] string Description

[3289] Access Type: Read-only

[3290] A textual description of the object.

[3291] boolean EnableBcp

[3292] Access Type: Read/Write

[3293] The EnableBcp property enables the use of bulk copy operations. If TRUE, bulk copy operations are available on the Microsoft® SQL Server™ connection. If FALSE, bulk copy operations are not available on the SQL Server™ connection.

[3294] SQL Server™ Default value: FALSE

[3295] string Login

[3296] Access Type: Read/Write

[3297] The Login property specifies a username for connecting when SQL Server™ Authentication is used to connect to a server running SQL Server™.

[3298] SQL Server™ Default value: none

[3299] boolean LoginSecure

[3300] Access Type: Read/Write

[3301] The LoginSecure property specifies the authentication mode to use when the provider attempts to connect to a SQL Server™. If TRUE, the provider uses Windows NT Authentication Mode. If FALSE, the provider uses SQL Server™ Authentication. For SQL Server™ authentication, the Login and Password properties are used to specify authentication information.

[3302] SQL Server™ Default value: TRUE

[3303] sint32 LoginTimeout

[3304] Access Type: Read/Write

[3305] The LoginTimeout property specifies the number of seconds to wait for a connection attempt to succeed. By default, the LoginTimeout property has a value of −1, which is interpreted currently as 60 seconds.

[3306] Units: Seconds

[3307] SQL Server™ Default value: −1 (60 seconds)

[3308] sint32 NetPacketSize

[3309] Access Type: Read/Write

[3310] The NetPacketSize property specifies the size of a network packet used to transmit a block of data between the provider and Microsoft® SQL Server™. The value of the property must have a value from 128 through 65535. SQL Server™ uses a default network packet size of 4096 bytes. Setting NetPacketSize to 0 enables the default size, 4096 bytes.

[3311] Units: Bytes

[3312] SQL Server™ Default value: 0 (4096 bytes)

[3313] boolean ODBCPrefix

[3314] Access Type: Read/Write

[3315] The ODBCPrefix property controls error and status message text formatting. When TRUE, descriptive error text is prefixed by the indicators of the error source. When FALSE, the error source indicators are stripped out and only the error message text is returned.

[3316] SQL Server™ Default value: TRUE

[3317] string Password

[3318] Access Type: Read/Write

[3319] The Password property indicates a password for a Microsoft® SQL Server™ login record. It is used for SQL Server™ authentication.

[3320] SQL Server™ Default value: none

[3321] sint32 QueryTimeout

[3322] Access Type: Read/Write

[3323] The QueryTimeout property specifies the number of seconds that must elapse before a time-out error is reported on an attempted statement execution.

[3324] Units: Seconds

[3325] SQL Server™ Default value: −1 (no timeout)

[3326] boolean QuotedIdentifier

[3327] Access Type: Read/Write

[3328] The QuotedIdentifier property controls Microsoft® SQL Server™ interpretation of identifier strings in statements submitted for execution. When TRUE, identifiers can be delimited by double quotation marks and character literal values must be delimited by single quotation marks. When FALSE, identifiers cannot be quoted and must follow all Transact-SQL rules for identifiers. For example, character literal values can be delimited by either single or double quotation marks.

[3329] SQL Server™ Default value: FALSE

[3330] [key] string SettingID

[3331] Access Type: Read/Write

[3332] The SettingId should contain the name of the instance of SQL Server™ these connection settings should apply to. This value should be identical to the Name property in the corresponding MSSQL_SQLServer instance.

[3333] Maximum Length: 256

[3334] boolean TranslateChar

[3335] Access Type: Read/Write

[3336] The TranslateChar property exposes the Microsoft® SQL Server™ ODBC driver statement attribute SQL_COPT_SS_TRANSLATE. SQL_COPT_SS_TRANSLATE causes the driver to translate characters between the client and server code pages as MBCS data is exchanged. The attribute affects only data stored in SQL Server™ char, varchar, and text columns. If TRUE, the connection behaves as defined for value SQL_XL_ON. This is the default behavior. The driver translates characters from one code page to another in character data exchanged between the client and the server. The driver automatically configures the character translation, determining the code page installed on the server and that in use by the client. If FALSE, the connection behaves as defined for value SQL_XL_OFF. The driver does not translate characters from one code page to another in character data exchanged between the client and the server.

[3337] SQL Server™ Default value: TRUE

[3338] Associations

[3339] MSSQL_SQLServerConnectionSetting is associated to MSSQL_SQLServer as the Setting property of the MSSQL_SQLServerSQLServerConnectionSetting association.

[3340] MSSQL_SQLServerDatabase: CIM_Component

[3341] Association Class

[3342] The MSSQL_SQLServerDatabase class represents an association between a SQL Server™ installation and a database that is part of the installation.

[3343] References

[3344] [key] MSSQL_SQLServer GroupComponent

[3345] Access Type: Read-only

[3346] The GroupComponent property references a SQL Server™ installation.

[3347] [key]MSSQL_Database PartComponent

[3348] Access Type: Read-only

[3349] The PartComponent property references a database that is part of the SQL Server™ installation referenced by the GroupComponent property.

[3350] MSSQL_SQLServerErrorLog: CIM_Dependency

[3351] Association Class

[3352] The MSSQL_SQLServerErrorLog represents an association between a SQL Server™ installation and the error log used by the installation.

[3353] References

[3354] [key]MSSQL_ErrorLog Antecedent

[3355] Access Type: Read-only

[3356] The Antecedent property references an error log used by the SQL Server™ installation.

[3357] [key]MSSQL_SQLServer Dependent

[3358] Access Type: Read-only

[3359] The Dependent property references a SQL Server™ installation.

[3360] MSSQL_SQLServerIntegratedSecuritySetting: CIM_ElementSetting

[3361] Association Class

[3362] The MSSQL_SQLServerIntegratedSecuritySetting class represents an association between a SQL Server™ installation and its security settings.

[3363] References

[3364] [key]MSSQL_SQLServer Element

[3365] Access Type: Read-only

[3366] The Element property references a SQL Server™ installation.

[3367] [key]MSSQL_IntegratedSecuritySetting Setting

[3368] Access Type: Read-only

[3369] The Setting property references the security settings used by the SQL Server™ installation.

[3370] MSSQL_SQLServerLanguageSetting: CIM_ElementSetting

[3371] Association Class

[3372] The MSSQL_SQLServerLanguageSetting class represents an association between a SQL Server™ installation and its language settings.

[3373] References

[3374] [key]MSSQL_SQLServer Element

[3375] Access Type: Read-only

[3376] The Element property references a SQL Server™ installation.

[3377] [key]MSSQL_LanguageSetting Setting

[3378] Access Type: Read-only

[3379] The Setting property references the language settings used by the SQL Server™ installation.

[3380] MSSQL_SQLServerLogin: MSSQL_Scope

[3381] Association Class

[3382] The MSSQL_SQLServerLogin class represents an association between a SQL server™ and a login defined within the SQL Server™.

[3383] References

[3384] [key]MSSQL_Login ScopedElement

[3385] Access Type: Read-only

[3386] The ScopedElement property references a login defined within the SQL Server™ installation.

[3387] [key]MSSQL_SQLServer ScopingElement

[3388] Access Type: Read-only

[3389] The ScopingElement property references a SQL Server™ installation.

[3390] MSSQL_SQLServerRegistry: CIM_ElementSetting

[3391] Association Class

[3392] The MSSQL_SQLServerRegistry class represents an association between a SQL Server™ installation and its registry setting.

[3393] References

[3394] [key] MSSQL_SQLServer Element

[3395] Access Type: Read-only

[3396] The Element property references a SQL Server™ installation.

[3397] [key]MSSQL_RegistrySetting Setting

[3398] Access Type: Read-only

[3399] The Setting property references the registry settings of the SQL Server™ installation.

[3400] MSSQL_SQLServerRole: MSSQL_Role

[3401] The MSSQL_SQLServerRole class represents a SQL Server™ security role not constrained to operation within a single database. Roles are used to establish groups of users, in order to make it convenient to set permissions for a group of users.

[3402] Properties

[3403] string Caption

[3404] Access Type: Read-only

[3405] The Caption property is a short textual description (one-line string) of the object.

[3406] Maximum Length: 64

[3407] string Description

[3408] Access Type: Read-only

[3409] The Description property returns a text describing the role.

[3410] string FullName

[3411] Access Type: Read-only

[3412] The FullName property returns a descriptive title for the role.

[3413] datetime InstallDate

[3414] Access Type: Read-only

[3415] The InstallDate-property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[3416] [key] string Name

[3417] Access Type: Read-only

[3418] The Name property is a label that uniquely identifies the SQL Server™ role.

[3419] [key] string SQLServerName

[3420] Access Type: Read-only

[3421] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[3422] Maximum Length: 128

[3423] string Status

[3424] Access Type: Read-only

[3425] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[3426] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[3427] Maximum Length: 10

[3428] Associations

[3429] MSSQL_SQLServerRole is associated to MSSQL_SQLServer as the ScopedElement property of the MSSQL_SQLServerServerRole association.

[3430] MSSQL_SQLServerRole is associated to MSSQL_Login as the Antecedent property of the MSSQL_MemberLogin association.

[3431] MSSQL_SQLServerServerRole: MSSQL_Scope

[3432] Association Class

[3433] The MSSQL_SQLServerServerRole class represents an association between a SQL Server™ and a server roles defined within the SQL Server™.

[3434] References

[3435] [key]MSSQL_SQLServerRole ScopedElement

[3436] Access Type: Read-only

[3437] The ScopedElement property references a system defined server role defined within the SQL Server™ installation.

[3438] [key]MSSQL_SQLServer ScopingElement

[3439] Access Type: Read-only

[3440] The ScopingElement property references a SQL Server™ installation.

[3441] MSSQL_SQLServerSQLServerConnectionSetting: CIM_ElementSetting

[3442] Association Class

[3443] The MSSQL_SQLServerSQLServerConnectionSetting class represents an association between a SQL Server™ installation and the settings used by the WMI SQL Server™ Administration provider to connect to the SQL Server™.

[3444] References

[3445] [key]MSSQL_SQLServer Element

[3446] Access Type: Read-only

[3447] The Element property references a SQL Server™ installation.

[3448] [key]MSSQL_SQLServerConnectionSetting Setting

[3449] Access Type: Read-only

[3450] The Setting property references the settings used by the SQL Server™ provider to connect to the SQL Server™.

[3451] MSSQL_SQLServerUser: MSSQL_Containment

[3452] Association Class

[3453] The MSSQL_SQLServerUser class represents an association between a SQL Server™ and a database user. This association allows an application to perform a single traversal to find the database users in a SQL Server™ and the login that they are mapped to.

[3454] Properties

[3455] [key] string DatabaseName

[3456] Access Type: Read-only

[3457] The DatabaseName property indicates the database that the user is defined in.

[3458] Maximum Length: 128

[3459] [key] string LoginName

[3460] Access Type: Read-only

[3461] The LoginName property indicates the login that the user is mapped to.

[3462] Maximum Length: 128

[3463] References

[3464] [key]MSSQL_User Containee

[3465] Access Type: Read-only

[3466] The Containee property references a user defined within the databases in the SQL Server™ installation.

[3467] [key]MSSQL_SQLServer Container

[3468] Access Type: Read-only

[3469] The Container property references a SQL Server™ installation.

[3470] MSSQL_StoredProcedure: MSSQL_DBMSObject

[3471] The MSSQL_StoredProcedure class represents standard as well as extended stored procedure defined in a SQL Server™ database. SQL Server™ stored procedures can contain input and output parameters and can return the results of one or more SELECT statements or a single long integer. In order to create an instance of a new stored procedure, the Text properties need to be specified along with the key properties of the class. The Text property specifies the Transact-SQL script that defines the stored procedure.

[3472] Properties

[3473] boolean AnsiNullsStatus

[3474] Access Type: Read-only

[3475] The AnsiNullsStatus property returns TRUE when the stored procedure depends on a table exhibiting SQL-92 NULL handling behavior. By default, SQL Server™ creates columns that do not accept NULL when the user does not explicitly declare the ability to accept NULL. Further, SQL Server™ returns TRUE when evaluating the expression NULL=NULL. These default behaviors are nonstandard. Database and client connection options override default SQL Server™ behavior. When the default is overridden, tables created exhibit SQL-92 standard NULL handling and objects that depend upon those tables function as specified by SQL-92.

[3476] string Caption

[3477] Access Type: Read-only

[3478] The Caption property is a short textual description (one-line string) of the object.

[3479] Maximum Length: 64

[3480] datetime CreateDate

[3481] Access Type: Read-only

[3482] The CreateDate property indicates the time and date on which the stored procedure was created. Note that creation date may be different from the install date in cases where the object is created in one place and then installed elsewhere.

[3483] [key] string DatabaseName

[3484] Access Type: Read-only

[3485] The DatabaseName property indicates the name of the database that the object is a part of.

[3486] Maximum Length: 128

[3487] string Description

[3488] Access Type: Read-only

[3489] The Description property provides a textual description of the object.

[3490] datetime InstallDate

[3491] Access Type: Read-only

[3492] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[3493] [key] string Name

[3494] Access Type: Read/Write

[3495] The Name property defines the label by which the object is known.

[3496] boolean QuotedIdentifierStatus

[3497] Access Type: Read-only

[3498] The QuotedIdentifierStatus property controls Microsoft® SQL Server™ interpretation of identifier strings in statements submitted for execution. When TRUE, identifiers can be delimited by double quotation marks and character literal values must be delimited by single quotation marks. When FALSE, identifiers cannot be quoted and must follow all Transact-SQL rules for identifiers. For example, character literal values can be delimited by either single or double quotation marks.

[3499] [key] string SQLServerName

[3500] Access Type: Read-only

[3501] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[3502] Maximum Length: 128

[3503] boolean Startup

[3504] Access Type: Read/Write

[3505] The Startup property indicates whether the stored procedure is executed automatically when SQL Server™ service is started. If TRUE when the stored procedure is executed automatically when the SQL Server™ service starts.

[3506] string Status

[3507] Access Type: Read-only

[3508] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[3509] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[3510] Maximum Length: 10

[3511] boolean SystemObject

[3512] Access Type: Read-only

[3513] The SystemObject property indicates whether the object is owned by Microsoft®. A value of TRUE indicates that the object implementation is owned by Microsoft®.

[3514] string Text

[3515] Access Type: Read/Write

[3516] The Text property indicates the Transact-SQL or other script that defines the stored procedure. Note that there is a special requirement for the name of the object is specified in the CREATE statement. The name of the object has to be in the form that includes the name of the owner. For example, in order to create an object named "Some Object", one would need to specify it as [dbo].[Some Object].

[3517] uint32 Type

[3518] Access Type: Read/Write

[3519] The Type property indicates the configured attributes of the stored procedure.

Value	Description	Explanation
0	Unknown	The value is invalid and the type is unknown.
1	Standard	The object is a standard SQL Server ™ stored procedure.
2	Extended	The object is an extended stored procedure.
3	Macro	This value is reserved for future use.
4	Replication Filter	This value is reserved for future use.

[3520] Methods

[3521] The MSSQL_StoredProcedure class supports the following methods:

Method Name	Description
Rename	The Rename method is used to rename the stored procedure instance.

[3522] Associations

[3523] MSSQL_StoredProcedure is associated to MSSQL_Database as the ScopedElement property of the MSSQL_DatabaseStoredProcedure association.

[3524] MSSQL_StoredProcedure is associated to MSSQL_User as the Element property of the MSSQL_UserStoredProcedurePermission association.

[3525] MSSQL_StoredProcedure is associated to MSSQL_DatabaseRole as the Element property of the MSSQL_DatabaseRoleStoredProcedurePermission association.

[3526] MSSQL_StoredProcedure is associated to MSSQL_StoredProcedureParameter as the ScopingElement property of the MSSQL_StoredProcedureStoredProcedureParameter association.

[3527] MSSQL_StoredProcedureParameter

[3528] The MSSQL_StoredProcedureParameter class represents the input and output parameters of a SQL Server™ stored procedure.

[3529] Properties

[3530] uint32 ColId

[3531] Access Type: Read-only

[3532] The ColId property indicates the ordinal position of the parameter.

[3533] [key] string DatabaseName

[3534] Access Type: Read-only

[3535] The DatabaseName property indicates the name of the database that the object is a part of.

[3536] Maximum Length: 128

[3537] string Datatype

[3538] Access Type: Read-only

[3539] The Datatype property indicates the data type of the stored procedure parameter.

[3540] uint32 Length

[3541] Access Type: Read-only

[3542] The Length property indicates the length of the SQL Server™ parameter, in bytes.

[3543] Units: Bytes

[3544] [key] string Name

[3545] Access Type: Read-only

[3546] The Name property defines the label by which the object is known.

[3547] Maximum Length: 128

[3548] boolean Output

[3549] Access Type: Read-only

[3550] The output property indicates if the parameter is an input parameter, output parameter or both input and output. A value of TRUE indicates that the parameter is output parameter, or an input and output parameter. A value of FALSE indicates that the parameter is an input parameter.

[3551] [key] string SQLServerName

[3552] Access Type: Read-only

[3553] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[3554] Maximum Length: 128

[3555] [key] string StoredProcedureName

[3556] Access Type: Read-only

[3557] The StoredProcedureName property indicates the name of the SQL Server™ stored procedure.

[3558] Maximum Length: 128

[3559] Associations MSSQL_StoredProcedureParameter is associated to MSSQL_StoredProcedure as the ScopedElement property of the MSSQL_StoredProcedureStoredProcedureParameter association.

[3560] MSSQL_StoredProcedureStoredProcedureParameter: MSSQL_Scope

[3561] Association Class

[3562] The MSSQL_StoredProcedureStoredProcedureParameter class associates a stored procedure to a parameter used in the stored procedure.

[3563] References

[3564] [key]MSSQL_StoredProcedureParameter ScopedElement

[3565] Access Type: Read-only

[3566] The ScopedElement property references a parameter used by the stored procedure referenced by the ScopingElement property.

[3567] [key]MSSQL_StoredProcedure ScopingElement

[3568] Access Type: Read-only

[3569] The ScopingElement property references a stored procedure.

[3570] MSSQL_SystemDatatype: MSSQL_Datatype

[3571] The MSSQL_SystemDatatype class represents base data type defined in Microsoft® SQL Server™.

[3572] Properties

[3573] boolean AllowIdentity

[3574] Access Type: Read-only

[3575] The AllowIdentity property indicates the ability of a data type to participate in a column defined with the identity property. The SQL Server™ identity property is defined for data types that can accept numeric values. A column defined with the identity property is defined with a starting value and a step value. SQL Server™ generates values for the column by querying the last applicable value and adding the step value.

[3576] boolean AllowLength

[3577] Access Type: Read-only

[3578] The AllowLength property indicates the ability to qualify a data type using a length parameter. AllowLength is TRUE for data types that accept a length qualification. For example, the property is TRUE for the SystemDatatype object referencing the varchar data type.

[3579] boolean AllowNulls

[3580] Access Type: Read-only

[3581] The AllowNulls property indicates whether the data type has the ability to accept NULL as a value.

[3582] string Caption

[3583] Access Type: Read-only

[3584] The Caption property is a short textual description (one-line string) of the object.

[3585] Maximum Length: 64

[3586] string Collation

[3587] Access Type: Read-only

[3588] The Collation property indicates the current collation of a string data type.

[3589] [key] string DatabaseName

[3590] Access Type: Read-only

[3591] The DatabaseName property indicates the name of the database that the object is a part of.

[3592] Maximum Length: 128

[3593] string Description

[3594] Access Type: Read-only

[3595] The Description property provides a textual description of the object.

[3596] datetime InstallDate

[3597] Access Type: Read-only

[3598] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[3599] sint32 MaximumChar

[3600] Access Type: Read-only

[3601] The MaximumChar property indicates the maximum number of characters used when a value of the data type is converted to a character string. The MaximumChar property returns a character count, not the number of bytes required to store a string of that length.

[3602] sint32 MaximumLength

[3603] Access Type: Read-only

[3604] The MaximumLength property returns the greatest length of a data type in bytes, or the precision of the type. For binary and character data types, the MaximumLength property returns the greatest number of bytes required to store a string of the type. For the fixed-precision, numeric data types, the MaximumLength property returns the maximum precision of the type. For all other referenced data types, the MaxSize property returns the number of bytes required to store a value of the type in a structure representing the type.

[3605] [key] string Name

[3606] Access Type: Read-only

[3607] The Name property defines the label by which the object is known.

[3608] boolean Numeric

[3609] Access Type: Read-only

[3610] The Numeric property indicates whether the system data type referenced is an exact, numeric data type. Exact numeric data types are scaled integer values represented as strings. When defining a column using an exact numeric data type, precision and scale are specified.

[3611] [key] string SQLServerName

[3612] Access Type: Read-only

[3613] The SQLServerName property indicates the name of the SQL Server# installation that the object is a part of.

[3614] Maximum Length: 128

[3615] string Status

[3616] Access Type: Read-only

[3617] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[3618] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[3619] Maximum Length: 10

[3620] boolean VariableLength

[3621] Access Type: Read-only

[3622] The VariableLength property specifies data length handling for a data type. A value of True indicates that the data type supports variable length.

[3623] Associations

[3624] MSSQL_SystemDatatype is associated to MSSQL_UserDatatype as the Antecedent property of the MSSQL_BaseDatatype association.

[3625] MSSQL_Table: MSSQL_DBMSObject

[3626] The MSSQL_Table class represents a table in the SQL Server™ database.

[3627] Properties

[3628] boolean AnsiNullsStatus

[3629] Access Type: Read-only

[3630] The AnsiNullsStatus property returns TRUE when the table exhibits SQL-92 NULL handling behavior. By default, SQL Server™ creates columns that do not accept NULL when the user does not explicitly declare the ability to accept NULL. Further, SQL Server™ returns TRUE when evaluating the expression NULL= NULL. These default behaviors are nonstandard. Database and client connection options override default SQL Server™ behavior. When the default is overridden, tables created exhibit SQL-92 standard NULL handling and objects that depend upon those tables behave as specified by SQL-92.

[3631] uint32 Attributes

[3632] Access Type: Read-only

[3633] The Attributes property indicates the various aspects of the table. The property indicates that the table can be one or more of the following: Identity—table has a column exposing the identity property. System Object—table is a system object defined by SQL Server™. Foreign Key—table has at least one foreign key. Referenced—table is referenced by at least one other table's foreign key. Published—table is published for replication. Replicated—table is actively subscribed to a Publisher. Check—table has at least one integrity constraint. Replica —at least one Subscriber has referenced the table's publication. Primary Key—table has a primary key. Unique—table has at least one UNIQUE constraint. Default—table has at least one DRI default defined. Replication Check—table has at least one integrity constraint not fired when replicated data is inserted.

Bit Position	Description
0	Identity
1	System Object
2	Foreign Key
3	Referenced
5	Published
6	Replicated
7	Check
8	Replica
9	Primary Key
10	Unique
11	Default
12	Replication Check

NOTE:
If one or more of bit positions 2, 7, 10, 11, or 12 are set, the table has at least one DRI constraint defined.

[3634] string Caption

[3635] Access Type: Read-only

[3636] The Caption property is a short textual description (one-line string) of the object.

[3637] Maximum Length: 64

[3638] datetime CreateDate

[3639] Access Type: Read-only

[3640] The CreateDate property indicates the time and date on which the table was created. Note that

creation date may be different from the install date in cases where the object is created in one place and then installed elsewhere.

[3641] [key] string DatabaseName

[3642] Access Type: Read-only

[3643] The DatabaseName property indicates the name of the database that the object is a part of.

[3644] Maximum Length: 128

[3645] sint32 DataSpaceUsed

[3646] Access Type: Read-only

[3647] The DataSpaceUsed property reports the storage space, in kilobytes, used by the rows of the table.

[3648] Units: Kilobytes

[3649] string Description

[3650] Access Type: Read-only

[3651] The Description property provides a textual description of the object.

[3652] boolean FakeSystemTable

[3653] Access Type: Read-only

[3654] The FakeSystemTable property returns TRUE when the table is a SQL Server™ system-defined table not implemented as a base or view table.

[3655] string FullTextCatalogName

[3656] Access Type: Read/Write

[3657] The FullTextCatalogName property specifies the Microsoft® Search full-text catalog supporting full-text query for the table. FullTextCatalogName is an empty string for tables not participating in full-text indexing.

[3658] boolean FullTextIndexActive

[3659] Access Type: Read/Write

[3660] The FullTextIndexActive property controls Microsoft® Search service activity for a table. If TRUE, the table is configured for participation in

[3661] Microsoft® Search full-text indexing. The Microsoft® Search service will gather index data from the designated columns and populate the index as directed. If FALSE, Microsoft® Search will not gather index data from the table regardless of configuration for full-text indexing participation. Full-text indexing must be properly configured for the table prior to setting FullTextIndexActive. The proper configuration involves setting of the FullTextCatalogName property, FullTextIndex property, and UniqueIndexForFullText property. If FullTextIndexActive is TRUE, setting it to TRUE generates an error. An error is also generated on attempts to set FullTextIndexActive to TRUE when full-text indexing has not been properly configured. If FullTextIndexActive is TRUE, setting it to FALSE simply removes the referenced table from participation in full-text index build and query. Setting the property does not affect the established configuration.

[3662] sint32 FullTextKeyColumn

[3663] Access Type: Read-only

[3664] The FullTextKeyColumn property returns the identifier of the column selected for row identification for Microsoft® Search. Microsoft®Search requires that a single column identify rows participating in an index that supports full-text query. The column designated must contain unique, non-NULL values and must participate in a table's PRIMARY KEY constraint or UNIQUE index.

[3665] uint32 FullTextPopulateStatus

[3666] Access Type: Read-only

[3667] The FullTextPopulateStatus property returns the population state of a Microsoft® Search full-text table. The FullTextPopulateStatus property returns one of the following values: Full—Full population of the table index is in progress for the full-text catalog. Incremental—Incremental population of the table index is in progress for the full-text catalog. None—No population of the table index is in progress for the full-text catalog.

[3668] Values are: "None", "Full", "Incremental"

[3669] sint32 IndexSpaceUsed

[3670] Access Type: Read-only

[3671] The IndexSpaceUsed property returns the number of kilobytes of disk resource used to store indexes built on the referenced Microsoft® SQL Server™ table.

[3672] Units: Kilobytes

[3673] datetime InstallDate

[3674] Access Type: Read-only

[3675] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[3676] [key] string Name

[3677] Access Type: Read-only

[3678] The Name property defines the label by which the object is known.

[3679] boolean QuotedIdentifierStatus

[3680] Access Type: Read-only

[3681] The QuotedIdentifierStatus property controls Microsoft® SQL Server™ interpretation of identifier strings in statements submitted for execution. When TRUE, identifiers can be delimited by double quotation marks and character literal values must be delimited by single quotation marks. When FALSE, identifiers cannot be quoted and must follow all Transact-SQL rules for

identifiers. For example, character literal values can be delimited by either single or double quotation marks.

[3682] sint32 Rows

[3683] Access Type: Read-only

[3684] The Rows property returns the number of rows in the table.

[3685] [key] string SQLServerName

[3686] Access Type: Read-only

[3687] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of. Maximum Length: 128

[3688] string Status

[3689] Access Type: Read-only

[3690] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[3691] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[3692] Maximum Length: 10

[3693] boolean SystemObject

[3694] Access Type: Read-only

[3695] The SystemObject property indicates whether the object is owned by Microsoft®. A value of TRUE indicates that the object implementation is owned by Microsoft®.

[3696] boolean TableFullTextChangeTrackingOn

[3697] Access Type: Read/Write

[3698] The TableFullTextChangeTrackingOn property specifies whether to enable the tracking and propagation of changes to a table for a full-text image index. When set to TRUE, the TableFullTextChangeTrackingOn property begins an incremental tracking of changes to a full-text search index if the table has a timestamp column to support the full-text tracking process. When set to FALSE, TableFullTextChangeTrackingOn stops tracking changes to the table. Set TableFullTextChangeTrackingOn to TRUE to enable the tracking and propagation of changes to a table for a full-text image index referenced by the Microsoft® Search service. TableFullTextChangeTrackingOn must be set to TRUE before an application can set the TableFullTextUpdateIndexOn property or call the FullTextUpdateIndex method to propagate the changes. Changes can be propagated to the index on a scheduled basis using a SQL Server™ Agent, or as they occur, using the TableFullTextUpdateIndexOn property, or on demand, using the FullTextUpdateIndex method.

[3699] boolean TableFullTextUpdateIndexOn

[3700] Access Type: Read/Write

[3701] The TableFullTextUpdateIndexOn property specifies whether to start or stop propagating tracked changes to the Microsoft® Search service automatically. Set the TableFullTextUpdateIndexOn property to TRUE to track index changes to the Microsoft® Search service as an automatic background operation. A list of all changes to the indexed data is propagated to the index as the changes occur. If TableFullTextUpdateIndexOn is set to FALSE, an application must call the FullTextUpdateIndex method to propagate the changes. Note that using TableFullTextUpdateIndexOn can have a significant impact on server performance, and should be used in an environment that has a CPU and memory configuration that allows propagation to keep pace with the index change rate.

[3702] string UniqueIndexForFullText

[3703] Access Type: Read/Write

[3704] The UniqueIndexForFullText property specifies the index used by Microsoft® Search to identify rows uniquely in a full-text indexed table.

[3705] boolean UsesFullTextIndex

[3706] Access Type: Read/Write

[3707] The UsesFullTextIndex property indicates whether the table is participating in Microsoft Search full-text queries. If TRUE, the table participates in full-text queries. UsesFullTextIndex must be TRUE in MSSQL_Table before the FullTextIndex property for the contained MSSQL_Column objects can be set to TRUE. If FALSE, the table does not participate in full-text queries.

[3708] Methods

[3709] The MSSQL_Table class supports the following methods:

Method Name	Description
CheckIdentity Value	The CheckIdentityValue method verifies the integrity of an identity column in the table.
CheckTable	The CheckTable method tests the integrity of database pages implementing storage for the table and indexes defined on it.
CheckTableDataOnly	The CheckTableDataOnly method tests the integrity of database pages implementing storage for the table.
Create	The Create method is used to create a new table instance.
ExportData	The ExportData method is used to copy data to a data file.

-continued

Method Name	Description
FullTextIndexScript	The FullTextIndexScript method returns a Transact-SQL command batch enabling Microsoft ® Search full-text indexing on the table.
FullTextUpdateIndex	The FullTextUpdateIndex method propagates the current set of tracked changes to Microsoft ® Search.
ImportData	The ImportData method is used to copy bulk copy data from a data file.
RebuildIndexes	The RebuildIndexes method re-creates all indexes defined on the table.
RecalcSpaceUsage	The RecalcSpaceUsage method forces the update of data reporting the disk resource usage of the table.
ReCompileReferences	The ReCompileReferences method causes recompilation, prior to the next execution, of any stored procedure or trigger that depend on the table.
Rename	The Rename method is used to rename a table instance.
StartFullTextPopulation	The StartFullTextPopulation method starts Microsoft ® Search full-text table population.
StopFullTextPopulation	The StopFullTextPopulation method stops Microsoft ® Search full-text table population.
TruncateData	The TruncateData method deletes all rows from the table as a non-logged operation.
UpdateStatistics	The UpdateStatistics method forces data distribution statistics update for all indexes defined on the table.
UpdateStatisticsWith	The UpdateStatisticsWith method forces data distribution statistics update for a indexes defined on the table.

[3710] Associations

[3711] MSSQL_Table is associated to MSSQL_FileGroup as the Dependent property of the MSSQL_TableFileGroup association.

[3712] MSSQL_Table is associated to MSSQL_FileGroup as the Dependent property of the MSSQL_TableTextFileGroup association.

[3713] MSSQL_Table is associated to MSSQL_Database as the PartComponent property of the MSSQL_DatabaseTable association.

[3714] MSSQL_Table is associated to MSSQL_Column as the GroupComponent property of the MSSQL_TableColumn association.

[3715] MSSQL_Table is associated to MSSQL_Index as the Dependent property of the MSSQL_TableIndex association.

[3716] MSSQL_Table is associated to MSSQL_Key as the Dependent property of the MSSQL_TableKey association.

[3717] MSSQL_Table is associated to MSSQL_Trigger as the Dependent property of the MSSQL_TableTrigger association.

[3718] MSSQL_Table is associated to MSSQL_User as the Element property of the MSSQL_UserTablePermission association.

[3719] MSSQL_Table is associated to MSSQL_DatabaseRole as the Element property of the MSSQL_DatabaseRoleTablePermission association.

[3720] MSSQL_Table is associated to MSSQL_ForeignKey as the Antecedent property of the MSSQL_ReferencedTable association.

[3721] MSSQL_Table is associated to MSSQL_Check as the Dependent property of the MSSQL_TableCheck association.

[3722] MSSQL_TableCheck: CIM_Dependency

[3723] Association Class

[3724] The MSSQL_TableCheck class represents an association between a table and the checks defined for the table.

[3725] References

[3726] [key]MSSQL_Check Antecedent

[3727] Access Type: Read-only

[3728] The Antecedent property references a check defined for the table referenced by the Dependent property.

[3729] [key]MSSQL_Table Dependent

[3730] Access Type: Read-only

[3731] The Dependent property references a table in SQL Server™.

[3732] MSSQL_TableColumn: CIM_Component

[3733] Association Class

[3734] The MSSQL_TableColumn class represents an association between a table and a column contained in the table.

[3735] References

[3736] [key]MSSQL_Table GroupComponent

[3737] Access Type: Read-only

[3738] The GroupComponent property represents a table in a SQL Server™ database.

[3739] [key]MSSQL_Column PartComponent

[3740] Access Type: Read-only

[3741] The PartComponent property references a column that is part of the table referenced by the GroupComponent property.

[3742] MSSQL_TableFileGroup: CIM_Dependency

[3743] Association Class

[3744] The MSSQL_TableFileGroup class represents an association between a table and the file groups used to store the table.

[3745] References

[3746] [key] MSSQL_FileGroup Antecedent

[3747] Access Type: Read-only

[3748] The Antecedent property references a file group defined for the table referenced by the Dependent property.

[3749] [key]MSSQL_Table Dependent

[3750] Access Type: Read-only

[3751] The Dependent property references a table in SQL Server™.

[3752] MSSQL_TableIndex: CIM_Dependency

[3753] Association Class

[3754] The MSSQL_TableIndex class represents an association between a table and an index defined for the table.

[3755] References

[3756] [key]MSSQL_Index Antecedent

[3757] Access Type: Read-only

[3758] The Antecedent property references a index defined for the table referenced by the Dependent property.

[3759] [key]MSSQL_Table Dependent

[3760] Access Type: Read-only

[3761] The Dependent property references a table in SQL Server™.

[3762] MSSQL_TableKey: CIM_Dependency

[3763] Association Class

[3764] The MSSQL_TableKey class represents an association between a table and a key defined for the table.

[3765] References

[3766] [key]MSSQL_Key Antecedent

[3767] Access Type: Read-only

[3768] The Antecedent property references a key defined for the table referenced by the Dependent property.

[3769] [key]MSSQL_Table Dependent

[3770] Access Type: Read-only

[3771] The Dependent property references a table in SQL Server™.

[3772] MSSQL_TableTextFileGroup: CIM_Dependency

[3773] Association Class

[3774] The MSSQL_TableTextFileGroup class associates a table with the file group that is used to store the variable length data in the table.

[3775] References p1[key]MSSQL_FileGroup Antecedent

[3776] Access Type: Read-only

[3777] The Antecedent property references a file group defined to store text data for the table referenced by the Dependent property.

[3778] [key]MSSQL_Table Dependent

[3779] Access Type: Read-only

[3780] The Dependent property references a table in SQL Server™.

[3781] MSSQL_TableTrigger: CIM_Dependency

[3782] Association Class

[3783] The MSSQL_TableTrigger class represents an association between a table and a trigger defined for the table.

[3784] References

[3785] [key]MSSQL_Trigger Antecedent

[3786] Access Type: Read-only

[3787] The Antecedent property references a trigger defined for the table referenced by the Dependent property.

[3788] [key]MSSQL_Table Dependent

[3789] Access Type: Read-only

[3790] The Dependent property references a table in SQL Server™.

[3791] MSSQL_TransactionLog: CIM_LogicalElement

[3792] The MSSQL_TransactionLog class represents the transaction log of a Microsoft® SQL Server™ database. A SQL Server™ transaction log maintains a record of modifications to the operating system files containing the data of an SQL Server™ database. The transaction log provides data-recovery assistance in the event of system failure and an SQL Server™ database has at least one operating system file that stores transaction log records. A transaction log can be written to more than one operating system file. Each SQL Server™ database maintains its own transaction log and the operating system file or files that store log records cannot be shared with another database.

[3793] Properties

[3794] string Caption

[3795] Access Type: Read-only

[3796] The Caption property is a short textual description (one-line string) of the object.

[3797] Maximum Length: 64

[3798] datetime CreateDate

[3799] Access Type: Read-only

[3800] The CreateDate property indicates the time and date on which the transaction log was created. Note that creation date may be different from the install date in cases where the object is created in one place and then installed elsewhere.

[3801] string Description

[3802] Access Type: Read-only

[3803] The Description property provides a textual description of the object.

[3804] uint32 InitialSize

[3805] Access Type: Read-only

[3806] The InitialSize property returns the initial size of file for the transaction log. This property is used to specify the size of the file at time of creating the transaction log.

[3807] datetime InstallDate

[3808] Access Type: Read-only

[3809] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[3810] datetime LastBackup

[3811] Access Type: Read-only

[3812] The LastBackup property returns the most recent date and time at which a backup operation was performed against the transaction log.

[3813] [key] string Name

[3814] Access Type: Read-only

[3815] The Name property defines the label by which the object is known.

[3816] real32 SpaceAvailableInMB

[3817] Access Type: Read-only

[3818] The SpaceAvailableInMB property returns the amount of disk resource allocated, but unused, in operating system files implementing SQL Server™ database and database transaction log storage. The figure is accurate to two decimal places.

[3819] Units: Megabytes

[3820] [key] string SQLServerName

[3821] Access Type: Read-only

[3822] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[3823] Maximum Length: 128

[3824] string Status

[3825] Access Type: Read-only

[3826] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[3827] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[3828] Maximum Length: 10

[3829] Methods

[3830] The MSSQL_TransactionLog class supports the following methods:

Method Name	Description
Truncate	The Truncate method archive-marks transaction log records.

[3831] Associations

[3832] MSSQL_TransactionLog is associated to CIM_DataFile as the GroupComponent property of the MSSQL_TransactionLogDataFile association.

[3833] MSSQL_TransactionLog is associated to MSSQL_Database as the Antecedent property of the MSSQL_DatabaseTransactionLog association.

[3834] MSSQL_TransactionLogDataFile: CIM_Component

[3835] Association Class

[3836] The MSSQL_TransactionLogDataFile class represents an association between SQL Server™ transaction log and the operating system file that is used to store the log.

[3837] References

[3838] [key] MSSQL_TransactionLog GroupComponent

[3839] Access Type: Read-only

[3840] The GroupComponent property references a transaction log.

[3841] [key]CIM_DataFile PartComponent

[3842] Access Type: Read-only

[3843] The PartComponent property references an operating system file that is used to store the transaction log.

[3844] MSSQL_TransferSetting: MSSQL_Setting

[3845] The MSSQL_TransferSetting class represents the settings used to control the data elements moved from one Microsoft® SQL Server™ database to another. This class has only static instances. An instance of this class is used as a parameter for the Transfer method of the MSSQL_Database class.

[3846] Properties

[3847] string Caption

[3848] Access Type: Read-only

[3849] A short textual description (one-line string) of the object.

[3850] Maximum Length: 64

[3851] boolean CopyAllDefaults

[3852] Access Type: Read/Write

[3853] The CopyAllDefaults property controls the transfer of Microsoft® SQL Server™ default definitions from the source to the target database. If

TRUE, all SQL Server™ default definitions in the source database are copied to the target. If FALSE, only default definitions indicated by the DefaultName property are copied.

[3854] boolean CopyAllFunctions

[3855] Access Type: Read/Write

[3856] The CopyAllFunctions property controls the transfer of SQL Server™ user-defined functions from the source to the target database. If TRUE, all SQL Server™ user-defined function definitions in the source database are copied to the target.)]

[3857] boolean CopyAllObjects

[3858] Access Type: Read/Write

[3859] The CopyAllObjects property controls the transfer of defaults, rules, stored procedures, tables, triggers, user-defined data types, and views from the source to the target database. If TRUE, all SQL Server™ database objects in the source database are copied to the target. If FALSE, only database objects indicated by the properties such as CopyAllTables, TableName and DefaultName property are copied.

[3860] boolean CopyAllRules

[3861] Access Type: Read/Write

[3862] The CopyAllRules property controls the transfer of Microsoft® SQL Server™ rule definitions from the source to the target database. If TRUE, all SQL Server™ rule definitions in the source database are copied to the target. If FALSE, only rule definitions indicated by the RuleName property are copied.

[3863] boolean CopyAllStoredProcedures

[3864] Access Type: Read/Write

[3865] The CopyAllStoredProcedures property controls the transfer of Microsoft® SQL Server™ stored procedure definitions from the source to the target database. If TRUE, all SQL Server™ stored procedures definitions in the source database are copied to the target. If FALSE, only stored procedures definitions indicated by the StoredProcedureName property are copied.

[3866] boolean CopyAllTables

[3867] Access Type: Read/Write

[3868] The CopyAllTables property controls the transfer of Microsoft® SQL Server™ table definitions from the source to the target database. If TRUE, all SQL Server™ table definitions in the source database are copied to the target. If FALSE, only table definitions indicated by the TableName property are copied.

[3869] boolean CopyAllTriggers

[3870] Access Type: Read/Write

[3871] The CopyAllTriggers property controls the transfer of Microsoft® SQL Server™ trigger definitions from the source to the target database. If TRUE, all SQL Server™ trigger definitions in the source database are copied to the target. If FALSE, only trigger definitions indicated by the TriggerName property are copied.

[3872] boolean CopyAllUserDefinedDatatypes

[3873] Access Type: Read/Write

[3874] The CopyAllUserDefinedDatatypes property controls the transfer of Microsoft® SQL Server™ user defined datatype definitions from the source to the target database. If TRUE, all SQL Server™ user defined datatype definitions in the source database are copied to the target. If FALSE, only use defined datatype definitions indicated by the DatatypeName are copied.

[3875] boolean CopyAllViews

[3876] Access Type: Read/Write

[3877] The CopyAllViews property controls the transfer of Microsoft® SQL Server™ view definitions from the source to the target database. If TRUE, all SQL Server™ view definitions in the source database are copied to the target. If FALSE, only view definitions indicated by the ViewName property are copied.

[3878] uint32 CopyData

[3879] Access Type: Read/Write

[3880] The CopyData property controls data transfer from a source to a target database. Append—data copied will be appended to existing tables. FALSE—data will not be copied. Only schema will be copied. Replace—existing data will be replaced by data copied.

[3881] Values are: "False", "Replace", "Append"

[3882] boolean CopySchema

[3883] Access Type: Read/Write

[3884] The CopySchema property controls table creation on data transfer. If TRUE, transfer processing creates tables prior to attempting to copy data. If FALSE, tables are not created prior to data copying. All tables indicated in the transfer operation must exist in the target database.

[3885] [key] string DatabaseName

[3886] Access Type: Read/Write

[3887] The DatabaseName property identifies the name of the database that the transfer setting has been defined for.

[3888] Maximum Length: 128

[3889] string DatatypeName []

[3890] Access Type: Read/Write

[3891] The DatatypeName property indicates the names of the datatypes to be transferred.

[3892] string DefaultName []

[3893] Access Type: Read/Write

[3894] The DefaultName property indicates the names of the defaults to be transferred. Note that the name should be in the format of owner.name, since in SQL Server™ the owner of a database object forms part of the identifier for the object.

[3895] string Description

[3896] Access Type: Read-only

[3897] A textual description of the object.

[3898] string DestDatabase

[3899] Access Type: Read/Write

[3900] The DestDatabase property specifies the target database for the transfer.

[3901] string DestLogin

[3902] Access Type: Read/Write

[3903] The DestLogin property provides a login identifier used to connect to the target server for the transfer.

[3904] string DestPassword

[3905] Access Type: Read/Write

[3906] The DestPassword property provides a password used to connect to a transfer target server.

[3907] string DestServer

[3908] Access Type: Read/Write

[3909] The DestServer property identifies the Microsoft® SQL Server™ installation that contains the target database for a transfer operation.

[3910] boolean DestTranslateChar

[3911] Access Type: Read/Write

[3912] The DestTranslateChar property controls behavior of character data translation on a destination server during a transfer operation. Performing character data translation during a transfer operation can significantly impact server performance if a large amount of data must be translated. Set DestTranslateChar to TRUE to perform character translation on the destination server. Set SourceTranslateChar to TRUE to resume character translation on the source server. DestTranslateChar is set to FALSE by default.

[3913] boolean DestUseTrustedConnection

[3914] Access Type: Read/Write

[3915] The DestUseTrustedConnection property requests Windows NT Authentication for the connection of the Transfer object to the target server. If TRUE, Windows NT Authentication is used in an attempt to connect to the target server. If FALSE, SQL Server™ Authentication is used in the connection attempt. The DestLogin and DestPassword properties provide login authentication parameters.

[3916] boolean DropDestObjectsFirst

[3917] Access Type: Read/Write

[3918] The DropDestObjectsFirst property is used to control the data transfer operation. If TRUE, the transfer attempts to drop a database object from the target database before copying the object from the source database. If FALSE, the transfer copies database objects. Note that the value of the DropDestObjectsFirst property applies only when database objects are copied in the transfer. To copy database objects, the CopySchema property value must be TRUE.

[3919] boolean IncludeDB

[3920] Access Type: Read/Write

[3921] The IncludeDB property specifies whether to create a database on the destination server during a data transfer operation. With the IncludeDB property set to TRUE, a database need not already exist at a destination server before database objects can be copied during a transfer operation. The IncludeDB property generates a destination database creation statement at the beginning of script execution during a transfer operation. The default is FALSE.

[3922] boolean IncludeDependencies

[3923] Access Type: Read/Write

[3924] The IncludeDependencies property controls the addition of dependent database objects to a user-defined list of SQL Server™ database objects in a transfer operation. If TRUE, the transfer automatically copies the SQL Server™ database objects on which user-selected database objects depend. If FALSE, only the user-selected objects are copied.

[3925] boolean IncludeLogins

[3926] Access Type: Read/Write

[3927] The IncludeLogins property controls handling of system administrator-created logins in a transfer operation. If TRUE, all system administrator-created logins in the source server's master database are created in the target server's master database as part of the transfer. If FALSE, no logins are created on the transfer target server.

[3928] boolean IncludeUsers

[3929] Access Type: Read/Write

[3930] The IncludeUsers property controls handling of SQL Server™ database user records in a transfer operation. If TRUE, all users in the source database are created in the target database as part of the transfer operation. If FALSE, no users are created in the target database.

[3931] string RuleName []

[3932] Access Type: Read/Write

[3933] The RuleName property indicates the names of the rules to be transferred. Note that the name should be in the format of owner.name, since in SQL Server™ the owner of a database object forms part of the identifier for the object.

[3934] uint32 Script2Type

[3935] The Script2Type property configures the Transact-SQL script generated and used to copy database schema in a transfer of schema from one database to another. The following are the possible bits that can be set on the Script2Type property. By default, none of these bits are set.

Bit Position	Description	Explanation
0	ANSI Padding	Generate Transact-SQL SET ANSI_PADDING ON and SET ANSI_PADDDING OFF statements before and after CREATE TABLE statements in the generated script. Applies only when scripting references an SQL Server ™ table.
1	ANSI File	Generated script file uses multibyte characters. Code page 1252 is used to determine character meaning.
2	Unicode File	Generated script output file is a Unicode-character text file.
4	No FG	Generated script does not include "ON <filegroup>" clause directing filegroup use. Applies only when scripting references an SQL Server ™ table.
7	Encrypt PWD	Encrypt passwords with script. When specified, Unicode File must be specified as well.
9	No What If Indexes	Do not script hypothetical indexes used to implement the CREATE STATISTICS statement. Applies only when scripting references an SQL Server ™ table.
10	Agent Notify	When scripting an alert, generate script creating notifications for the alert.
11	Agent Alert Job	Generate Transact-SQL script creating SQL Server ™ Agent jobs and alerts.
19	Full Text Index	Generated script includes statements defining Microsoft ® Search full-text indexing. Applies only when scripting references an SQL Server ™ table.
20	Login SID	Include security identifiers for logons scripted.
21	Full Text Cat	Command batch includes Transact-SQL statements creating Microsoft ® Search full-text catalogs.
22	Extended Property	Include extended property scripting as part of object scripting. Note that this bit is used only in Microsoft SQL Server ™ 2000.
23	No Collation	Do not script the collation clause if source is an instance of SQL Server ™ version 7.0 or later. The default is to generate collation. Note that this bit is used only in Microsoft SQL Server ™ 2000.

[3936] uint32 ScriptType

[3937] Access Type: Read/Write

[3938] The ScriptType property configures the Transact-SQL script generated and used to copy database schema in a transfer of schema from one database to another. The Primary Object, Drops, Bindings, Clustered Indexes, Non Clustered Indexes, Triggers, To File Only, Object Permissions, Database Permissions, Include Headers, Aliases, Include If Not Exists and Owner Qualify bits are set by default. Set the Database Permissions and Object Permissions bits to include all permissions in the transfer. Set the Clustered Index, Non Clustered Index and DRI Index bits to include all the indexes in the transfer. Set the Primary Key, Foreign Key and Unique Key bits to include all keys in the transfer. Set the Checks, Defaults, Foreign Keys, Primary Keys and Unique Keys bits to include all constraints in the transfer. Set the Primary Key, Foreign Key, Unique Key, Checks, Defaults, Foreign Keys, Primary Keys and Unique Keys bits to include all constraints and keys in the transfer.

[3939] The following bits may be set for the ScriptType property:

Bit Position	Description	Explanation
0	Drops	Generate Transact-SQL to remove referenced component. Script tests for existence prior attempt to remove component.
1	Object Permissions	Include Transact-SQL privilege defining statements when scripting database objects.
2	Primary Object	Generate Transact-SQL creating the referenced component.
3	Clustered Indexes	Generate Transact-SQL defining clustered indexes. Applies only when scripting references an SQL Server ™ table.
4	Triggers	Generate Transact-SQL defining triggers. Applies only when scripting references an SQL Server ™ table.
5	Database Permissions	Generate Transact-SQL database privilege defining script. Database permissions grant or deny statement execution rights.
7	Bindings	Generate sp_bindefault and sp_bindrule statements. Applies only when scripting references a table.
10	UDDTs To Base Type	Convert specification of user-defined data types to the appropriate SQL Server ™ base data type. Applies only when scripting references an SQL Server ™ table.
12	Include If Not Exists	Transact-SQL creating a component is prefixed by a check for existence. When script is executed, component is created only when a copy of the named component does not exist.
13	Non Clustered Indexes	Generate Transact-SQL defining nonclustered indexes. Applies only when scripting references an SQL Server ™ table.
17	Include Headers	Generated script is prefixed with a header containing date and time of generation and other descriptive information.
18	Owner Qualify	Object names in Transact-SQL generated to remove an object are qualified by the owner of the referenced object. Transact-SQL generated to create the referenced object qualify the object name using the current object owner.
19	Timestamp To Binary	When scripting object creation for a table or user-defined data type, convert specification of timestamp data type to binary(8).

-continued

Bit Position	Description	Explanation
22	DRI Non-Clustered	Generated script creates nonclustered indexes. Directs scripting when declarative referential integrity establishes dependency relationships. Applies only when scripting references an SQL Server ™ table.
23	DRI Clustered	Generated script creates clustered indexes. Directs scripting when declarative referential integrity establishes dependency relationships. Applies only when scripting references a SQL Server ™ table.
24	DRI Checks	Generated script creates column-specified CHECK constraints. Directs scripting when declarative referential integrity establishes dependency relationships. Applies only when scripting references a SQL Server ™ table.
25	DRI Defaults	Generated script includes column-specified defaults. Directs scripting when declarative referential integrity establishes dependency relationships. Applies only when scripting references a SQL Server ™ table.
26	DRI Unique Keys	Generated script creates candidate keys defined using a unique index. Directs scripting when declarative referential integrity establishes dependency relationships. Applies only when scripting references an SQL Server ™ table.
27	DRI Foreign Keys	Generated script creates FOREIGN KEY constraints. Directs scripting when declarative referential integrity establishes dependency relationships. Applies only when scripting references an SQL Server ™ table.
28	DRI Primary Key	Generated script creates PRIMARY KEY constraints. Directs scripting when declarative referential integrity establishes dependency relationships.Applies only when scripting references an SQL Server ™ table.
29	DRI With No Check	When using DRI Checks, DRI Foreign Keys, generated script includes the WITH NOCHECK clause optimizing constraint creation. Applies only when scripting references an SQL Server ™ table.
30	No Identity	Generated Transact-SQL statements do not include definition of identity property, seed, and increment. Applies only when scripting references an SQL Server ™ table.
31	Use Quoted Identifiers	Use quote characters to delimit identifier parts when scripting object names.

[3940] [key] string SettingID

[3941] Access Type: Read/Write

[3942] The identifier by which the setting object is known.

[3943] Maximum Length: 256

[3944] boolean SourceTranslateChar

[3945] Access Type: Read/Write

[3946] The SourceTranslateChar property specifies whether to perform character data translation on the source server during a transfer operation. Performing character data translation during a transfer operation can significantly impact server performance if a large amount of data must be translated. The SourceTranslateChar property is set to TRUE by default. Set the DestTranslateChar property to TRUE to perform character translation on the destination server. Set SourceTranslateChar to TRUE to resume character translation on the source server.

[3947] [key] string SQLServerName

[3948] Access Type: Read/Write

[3949] The SQLServerName property identifies the SQL Server™ installation that the transfer setting has been defined for.

[3950] Maximum Length: 128

[3951] string StoredProcedureName []

[3952] Access Type: Read/Write

[3953] The StoredProcedureName property indicates the names of the stored procedures to be transferred. Note that the name should be in the format of owner. name, since in SQL Server™ the owner of a database object forms part of the identifier for the object.

[3954] string TableName []

[3955] Access Type: Read/Write

[3956] The TableName property indicates the names of the tables to be transferred. Note that the name should be in the format of owner.name, since in SQL Server™ the owner of a database object forms part of the identifier for the object.

[3957] string TriggerName []

[3958] Access Type: Read/Write

[3959] The TriggerName property indicates the names of the triggers to be transferred. Note that the name should be in the format of owner.name, since in SQL Server™ the owner of a database object forms part of the identifier for the object.

[3960] boolean UseCollation

[3961] Access Type: Read/Write

[3962] The UseCollation property controls column-level collation settings when transferring data between computers running an instance of Microsoft® SQL Server™ 2000. If UseCollation

is set to TRUE, column-level collation settings are maintained when transferring data between computers running an instance of SQL Server™ 2000 if the code pages are the same on both servers. When transferring data to a computer running an instance of SQL Server™ 2000 using a different code page, all collation settings at the source computer are automatically translated to the code page of the destination server if the code pages settings are different. When transferring data to a computer running an instance of SQL Server™ 7.0 or earlier, all collation settings at the source server are automatically translated to the code page of the destination server if the code pages settings are different. The source database column-level collation is translated accordingly. If UseCollation is set to FALSE, direct data transfer is performed if the code pages are the same on both servers. If the code pages are different, the data is translated from source code page to destination code page. If both computers are running an instance of SQL Server™ 2000 and the source and destination databases are using different code pages, data might be translated to the incorrect code page setting depending on whether the column is using the default or a non-default collation. Note that setting UseCollation to TRUE can result in a increase in performance overhead if the data contains non-Unicode data types such as text or varchar. Performance can also be affected by the number of tables, columns, and rows in the source database.

[3963] boolean UseDestTransaction

[3964] Access Type: Read/Write

[3965] The UseDestTransaction property controls inclusion of DROP statements in a transaction during a transfer operation. When UseDestTransaction is set to TRUE, the entire transfer operation (including DROP statements, CREATE SCHEMA statements, and data copying) is included in a transaction. If any of these operations fail, the transaction is rolled back. Statistics are updated after the transaction is committed. The default is FALSE. When UseDestTransaction is set to TRUE, the application cannot perform these operations within the transaction - dump the transaction log, change bcp settings, update statistics, and script a full-text catalog.

[3966] string ViewName []

[3967] Access Type: Read/Write

[3968] The ViewName property indicates the names of the views to be transferred. Note that the name should be in the format of owner.name, since in SQL Server™ the owner of a database object forms part of the identifier for the object.

[3969] MSSQL_Trigger: MSSQL_DBMSObject

[3970] The MSSQL_Trigger class represents a trigger. SQL Server™ supports using triggers as a kind of stored procedure. Triggers are executed when a specified data modification, such as an attempt to delete a row, is attempted on the table on which the trigger is defined.

[3971] Properties

[3972] boolean AfterTrigger

[3973] Access Type: Read-only

[3974] The AfterTrigger property indicates whether a trigger is an AFTER trigger. A value of TRUE indicates that the trigger is an AFTER trigger. AFTER triggers fire after the triggering action (INSERT, UPDATE, or DELETE) and after any constraints have been processed. AFTER triggers can only be created on tables. All triggers created using SQL Server™ version 7.0 or earlier are AFTER triggers.

[3975] boolean AnsiNullsStatus

[3976] Access Type: Read-only

[3977] The AnsiNullsStatus property returns TRUE when the trigger depends on a table exhibiting SQL-92 NULL handling behavior. By default, SQL Server™ creates columns that do not accept NULL when the user does not explicitly declare the ability to accept NULL. Further, SQL Server™ returns TRUE when evaluating the expression NULL=NULL. These default behaviors are nonstandard. Database and client connection options override default SQL Server™ behavior. When the default is overridden, tables created exhibit SQL-92 standard NULL handling and objects that depend upon those tables behave as specified by SQL-92.

[3978] string Caption

[3979] Access Type: Read-only

[3980] The Caption property is a short textual description (one-line string) of the object.

[3981] Maximum Length: 64

[3982] datetime CreateDate

[3983] Access Type: Read-only

[3984] The CreateDate property indicates the time and date on which the trigger was created. Note that creation date may be different from the install date in cases where the object is created in one place and then installed elsewhere.

[3985] [key] string DatabaseName

[3986] Access Type: Read-only

[3987] The DatabaseName property indicates the name of the database that the object is a part of.

[3988] Maximum Length: 128

[3989] string Description

[3990] Access Type: Read-only

[3991] The Description property provides a textual description of the object.

[3992] boolean Enabled

[3993] Access Type: Read/Write

[3994] The Enabled property indicates the state of the trigger. When TRUE, the trigger is enabled. When FALSE, the trigger is disabled.

[3995] datetime InstallDate

[3996] Access Type: Read-only

[3997] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[3998] boolean InsteadOfrigger

[3999] Access Type: Read-only

[4000] The lnsteadOfTrigger property indicates whether a trigger is an INSTEAD OF trigger. A value of TRUE indicates that the trigger is an INSTEAD OF trigger. INSTEAD OF triggers are executed instead of the triggering action. INSTEAD OF triggers can also be defined on views, in which case they greatly extend the types of updates a view can support. Each table or view can have one INSTEAD OF trigger for each triggering action (UPDATE, DELETE, and INSERT).

[4001] [key] string Name

[4002] Access Type: Read-only

[4003] The Name property defines the label by which the object is known.

[4004] boolean QuotedIdentifierStatus

[4005] Access Type: Read-only

[4006] The QuotedIdentifierStatus property returns TRUE when the trigger has been created with a dependency on quote characters for identifier determination. When TRUE, identifiers can be delimited by double quotation marks and character literal values must be delimited by single quotation marks. When FALSE, identifiers cannot be quoted and must follow all Transact-SQL rules for identifiers. For example, character literal values can be delimited by either single or double quotation marks.

[4007] [key] string SQLServerName

[4008] Access Type: Read-only

[4009] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[4010] Maximum Length: 128

[4011] string Status

[4012] Access Type: Read-only

[4013] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[4014] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[4015] Maximum Length: 10

[4016] boolean SystemObject

[4017] Access Type: Read-only

[4018] The SystemObject property indicates whether the object is owned by Microsoft®. A value of TRUE indicates that the object implementation is owned by Microsoft®.

[4019] [key] string TableName

[4020] Access Type: Read-only

[4021] The TableName property indicates the name of the table that the trigger is defined in.

[4022] Maximum Length: 128

[4023] string Text

[4024] Access Type: Read/Write

[4025] The Text property returns the Transact-SQL script that defines the trigger. Note that there is a special requirement for the name of the object is specified in the CREATE statement. The name of the object has to be in the form that includes the name of the owner. For example, in order to create an object named "Some Object", one would need to specify it as [dbo].[Some Object].

[4026] uint32 Type

[4027] Access Type: Read-only

[4028] The Type property indicates the type of the trigger. A trigger can fire when a Transact-SQL INSERT, UPDATE, or DELETE statement modifies data in the table on which the trigger is defined.

Value	Description	Explanation
0	Unknown	Bad or invalid value.
1	Insert	Fired by an INSERT statement.
2	Update	Fired by an UPDATE statement.
4	Delete	Fired by a DELETE statement.
7	All	Fired by any data modification statement.

[4029] Methods

[4030] The MSSQL_Trigger class supports the following methods:

Method Name	Description
Rename	The Rename method is used to rename the trigger instance.

[4031] Associations

[4032] MSSQL_Trigger is associated to MSSQL_Table as the Antecedent property of the MSSQL_TableTrigger association.

[4033] MSSQL_UniqueKey: MSSQL_CandidateKey

[4034] The MSSQL_UniqueKey object represents a unique key in a database. All candidate keys that are not the primary key are unique keys.

[4035] Properties

[4036] string Caption

[4037] Access Type: Read-only

[4038] The Caption property is a short textual description (one-line string) of the object.

[4039] Maximum Length: 64

[4040] [key] string DatabaseName

[4041] Access Type: Read-only

[4042] The DatabaseName property indicates the name of the database that the key is a part of.

[4043] Maximum Length: 128

[4044] string Description

[4045] Access Type: Read-only

[4046] The Description property provides a textual description of the object.

[4047] datetime InstallDate

[4048] Access Type: Read-only

[4049] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[4050] [key] string Name

[4051] Access Type: Read-only

[4052] The Name property defines the label by which the object is known. The name of a key is unique within a database.

[4053] [key] string SQLServerName

[4054] Access Type: Read-only

[4055] The SQLServerName property indicates the name of the SQL Server™ installation that the key is a part of.

[4056] Maximum Length: 128

[4057] string Status

[4058] Access Type: Read-only

[4059] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[4060] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[4061] Maximum Length: 10

[4062] [key] string TableName

[4063] Access Type: Read-only

[4064] The TableName property indicates the name of the table that the key is defined in.

[4065] Maximum Length: 128

[4066] Methods

[4067] The MSSQL_UniqueKey class supports the following methods:

Method Name	Description
Create	The Create method is used to create a new unique key instance.
RebuildIndex	The RebuildIndex method re-creates an index for a candidate key constraint.
Rename	The Rename method is used to rename a unique key instance.

[4068] MSSQL_User: MSSQL_DBMSUserObject

[4069] The MSSQL_User user exposes the attributes of a single Microsoft® SQL Server™ database user.

[4070] Properties

[4071] string Caption

[4072] Access Type: Read-only

[4073] The Caption property is a short textual description (one-line string) of the object.

[4074] Maximum Length: 64

[4075] [key] string DatabaseName

[4076] Access Type: Read-only

[4077] The DatabaseName property indicates the name of the database that the user is a part of.

[4078] Maximum Length: 128

[4079] string Description

[4080] Access Type: Read-only

[4081] The Description property provides a textual description of the object.

[4082] datetime InstallDate

[4083] Access Type: Read-only

[4084] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[4085] [key] string Name

[4086] Access Type: Read-only

[4087] The Name property defines the label by which the user is known.

[4088] [key] string SQLServerName

[4089] Access Type: Read-only

[4090] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[4091] Maximum Length: 128

[4092] string Status

[4093] Access Type: Read-only

[4094] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[4095] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[4096] Maximum Length: 10

[4097] boolean SystemObject

[4098] Access Type: Read-only

[4099] The SystemObject property indicates whether the object is owned by Microsoft®. A value of True indicates that the object implementation is owned by Microsoft®.

[4100] Methods

[4101] The MSSQL_User class supports the following methods:

Method Name	Description
Create	The Create method is used to create a new user.

[4102] Associations

[4103] MSSQL_User is associated to MSSQL_Database as the ScopedElement property of the MSSQL_DatabaseUser association.

[4104] MSSQL_User is associated to MSSQL_DatabaseRole as the Dependent property of the MSSQL_MemberUser association.

[4105] MSSQL_User is associated to MSSQL_Login as the Dependent property of the MSSQL_UserLogin association.

[4106] MSSQL_User is associated to MSSQL_Database as the Grantee property of the MSSQL_UserDatabasePermission association.

[4107] MSSQL_User is associated to MSSQL_StoredProcedure as the Grantee property of the MSSQL_UserStoredProcedurePermission association.

[4108] MSSQL_User is associated to MSSQL_View as the Grantee property of the MSSQL_UserViewPermission association.

[4109] MSSQL_User is associated to MSSQL_Table as the Grantee property of the MSSQL_UserTablePermission association.

[4110] MSSQL_User is associated to MSSQL_DBMSObject as the Antecedent to property of the MSSQL_DBMSObjectOwner association.

[4111] MSSQL_User is associated to MSSQL_UserDefinedFunction as the Grantee property of the MSSQL_UserUserDefinedFunctionPermission association.

[4112] MSSQL_User is associated to MSSQL_SQLServer as the Containee property of the MSSQL_SQLServerUser association.

[4113] MSSQL_UserDatabasePermission: MSSQL_Permission

[4114] Association Class

[4115] The MSSQL_UserDatabasePermission class represents the permissions granted to a user for a database. The instances of this class represent only the permission that has been explicitly granted or denied to the user object. For example, if a user has permissions to access a database by virtue of being a member of a certain database role, then there will not be a permission association instance between the user and the database.

[4116] Properties

[4117] boolean Granted

[4118] Access Type: Read/Write

[4119] The Granted property indicates whether the permission has been granted or denied. A value of True indicates that the permission has been granted. A value of False indicates that permission has been denied.

[4120] [Key] uint32 PrivilegeType

[4121] Access Type: Read/Write

[4122] The PrivilegeType property indicates the type of privilege that has been granted or denied.

Value	Description
0	Unknown
1	Permission to query a table
2	Permission to add rows to a table
4	Permission to update rows of a table
8	Permission to delete rows of a table
16	Permission to execute a stored procedure
32	Permission to grant DRI on a table
63	All privileges applicable to the database object
128	Permission to create and own a table
256	Permission to create and own a database
512	Permission to create and own a view
1024	Permission to create and own a stored procedure
2048	Permission to backup a database
4096	Permission to create a default
8192	Permission to backup a database transaction log
16384	Permission to create a rule
32768	Permission to backup to a table
65366	Permission to create a user defined function
130944	All privileges applicable to the database

[4123] References

[4124] [key]MSSQL_Database Element

[4125] Access Type: Read/Write

[4126] The Element property references a SQL Server™ database.

[4127] [key]MSSQL_User Grantee

[4128] Access Type: Read/Write

[4129] The Grantee property references a user for whom the permissions have been defined.

[4130] MSSQL_UserDatatype: MSSQL_Datatype

[4131] The MSSQL_UserDatatype class represents a data type defined by a user.

[4132] Properties

[4133] boolean AllowIdentity

[4134] Access Type: Read-only

[4135] The AllowIdentity property indicates the ability of a data type to participate in a column defined with the identity property. The SQL Server™ identity property is defined for data types that can accept numeric values. A column defined with the identity property is defined with a starting value and a step value. SQL Server™ generates values for the column by querying the last applicable value and adding the step value.

[4136] boolean AllowNulls

[4137] Access Type: Read-only

[4138] The AllowNulls property indicates whether the data type has the ability to accept NULL as a value.

[4139] string BaseDatatype

[4140] Access Type: Read-only

[4141] The BaseDatatype property indicates the system datatype from which the user datatype was derived.

[4142] string Caption

[4143] Access Type: Read-only

[4144] The Caption property is a short textual description (one-line string) of the object.

[4145] Maximum Length: 64

[4146] string Caption

[4147] Access Type: Read-only

[4148] The Caption property is a short textual description (one-line string) of the object.

[4149] Maximum Length: 64

[4150] string Collation

[4151] Access Type: Read-only

[4152] The Collation property indicates the current collation of a string data type.

[4153] [key] string DatabaseName

[4154] Access Type: Read-only

[4155] The DatabaseName property indicates the name of the database that the object is a part of.

[4156] Maximum Length: 128

[4157] string Description

[4158] Access Type: Read-only

[4159] The Description property provides a textual description of the object.

[4160] datetime InstallDate

[4161] Access Type: Read-only

[4162] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[4163] boolean IsVariableLength

[4164] Access Type: Read-only

[4165] The IsVariableLength property specifies data length handling for a data type. A value of True indicates that the data type supports variable length.

[4166] sint32 Length

[4167] Access Type: Read-only

[4168] The Length property specifies the maximum number of characters or bytes accepted user-defined data type.

[4169] sint32 MaxSize

[4170] Access Type: Read-only

[4171] The MaxSize property returns the greatest length of a data type in bytes, or the precision of the type. For binary and character data types, the MaxSize property returns the greatest number of bytes required to store a string of the type. For the fixed-precision, numeric data types, the MaxSize property returns the maximum precision of the

type. For all other referenced data types, the MaxSize property returns the number of bytes required to store a value of the type in a structure representing the type.

[4172] [key] string Name

[4173] Access Type: Read-only

[4174] The Name property defines the label by which the object is known.

[4175] sint32 NumericPrecision

[4176] Access Type: Read-only

[4177] The NumericPrecision property specifies the maximum number of digits in a fixed-precision, numeric data type.

[4178] sint32 NumericScale

[4179] Access Type: Read-only

[4180] The NumericScale property specifies the number of digits to the right of the decimal point in a fixed-precision, numeric data type.

[4181] [key] string SQLServerName

[4182] Access Type: Read-only

[4183] The SQLServerName property indicates the name of the SQL Server installation that the object is a part of.

[4184] Maximum Length: 128

[4185] string Status

[4186] Access Type: Read-only

[4187] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[4188] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[4189] Maximum Length: 10

[4190] Methods

[4191] The MSSQL_UserDatatype class supports the following methods:

Method Name	Description
Rename	The Rename method is used to rename the user datatype instance.

[4192] Associations

[4193] MSSQL_UserDatatype is associated to MSSQL_Rule as the Dependent property of the MSSQL_UserDatatypeRule association.

[4194] MSSQL_UserDatatype is associated to MSSQL_Default as the Dependent property of the MSSQL_UserDatatypeDefault association.

[4195] MSSQL_UserDatatype is associated to MSSQL_SystemDatatype as the Dependent property of the MSSQL_BaseDatatype association.

[4196] MSSQL_UserDatatypeDefault: CIM_Dependency

[4197] Association Class

[4198] The MSSQL_UserDatatypeDefault class represents an association between a user-defined datatype and the rule bound to the column.

[4199] References

[4200] [key]MSSQL_Default Antecedent

[4201] Access Type: Read-only

[4202] The Antecedent property references the rule bound to the user-defined datatype.

[4203] [key]MSSQL_UserDatatype Dependent

[4204] Access Type: Read-only

[4205] The Dependent property references a user-defined datatype.

[4206] Methods

[4207] The MSSQL_UserDatatypeDefault class supports the following methods:

Method Name	Description
Create	The Create method is used to create a new instance.

[4208] MSSQL_UserDatatypeRule: CIM_Dependency

[4209] Association Class

[4210] The MSSQL_UserDatatypeRule class represents an association between a user defined datatype and the rule bound to the column.

[4211] References

[4212] [key]MSSQL_Rule Antecedent

[4213] Access Type: Read-only

[4214] The Antecedent property references the rule bound to the user-defined datatype.

[4215] [key]MSSQL_UserDatatype Dependent

[4216] Access Type: Read-only

[4217] The Dependent property references a user-defined datatype.

[4218] Methods

[4219] The MSSQL_UserDatatypeDefault class supports the following methods:

Method Name	Description
Create	The Create method is used to create a new instance.

[4220] MSSQL_UserDefinedFunction: MSSQL_DBMSObject

[4221] MSSQL_UserDefinedFunction class represents a user-defined function in the SQL Server™ database.

[4222] Properties

[4223] boolean AnsiNullsStatus

[4224] Access Type: Read/Write

[4225] The AnsiNullsStatus property returns TRUE when the database object referenced depends on a table exhibiting SQL-92 NULL handling behavior.

[4226] string Caption

[4227] Access Type: Read-only

[4228] The Caption property is a short textual description (one-line string) of the object.

[4229] Maximum Length: 64

[4230] datetime CreateDate

[4231] Access Type: Read-only

[4232] The CreateDate property indicates the time and date on which the user-defined function was created. Note that creation date may be different from the install date in cases where the object is created in one place and then installed elsewhere.

[4233] [key] string DatabaseName

[4234] Access Type: Read-only

[4235] The DatabaseName property indicates the name of the database that the object is a part of.

[4236] Maximum Length: 128

[4237] string Description

[4238] Access Type: Read-only

[4239] The Description property provides a textual description of the object.

[4240] boolean Deterministic

[4241] Access Type: Read-only

[4242] The Deterministic property specifies whether a user-defined function is a deterministic function. If TRUE, the user-defined function is deterministic. If FALSE, the user-defined function is not deterministic.

[4243] datetime InstallDate

[4244] Access Type: Read-only

[4245] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[4246] [key] string Name

[4247] Access Type: Read-only

[4248] The Name property defines the label by which the object is known.

[4249] boolean QuotedIdentifierStatus

[4250] Access Type: Read-only

[4251] The QuotedIdentifierStatus property controls Microsoft® SQL Server™ interpretation of identifier strings in statements submitted for execution. When True, identifiers can be delimited by double quotation marks and character literal values must be delimited by single quotation marks. When False, identifiers cannot be quoted and must follow all Transact-SQL rules for identifiers. For example, character literal values can be delimited by either single or double quotation marks.

[4252] [key] string SQLServerName

[4253] Access Type: Read-only

[4254] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[4255] Maximum Length: 128

[4256] string Status

[4257] Access Type: Read-only

[4258] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[4259] Values are: "OK""Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[4260] Maximum Length: 10

[4261] boolean SystemObject

[4262] Access Type: Read-only

[4263] The SystemObject property indicates whether the object is owned by Microsoft®. A value of True indicates that the object implementation is owned by Microsoft®.

[4264] string Text

[4265] Access Type: Read/Write

[4266] The Text property indicates the Transact-SQL or other script that defines the object. Note that there is a special requirement for the name of the object is specified in the CREATE statement. The name of the object has to be in the form that includes the name of the owner. For example, in order to create an object named "Some Object", one would need to specify it as [dbo].[Some Object].

[4267] Associations

[4268] MSSQL_UserDefinedFunction is associated to MSSQL_DatabaseRole as the Element property of the MSSQL_DatabaseRoleUserDefinedFunctionPermission association.

[4269] MSSQL_UserDefinedFunction is associated to MSSQL_Database as the ScopedElement property of the MSSQL_DatabaseUserDefinedFunction association.

[4270] MSSQL_UserDefinedFunction is associated to MSSQL_User as the Element property of the MSSQL_UserUserDefinedFunctionPermission association.

[4271] MSSQL_UserLogin: CIM_Dependency

[4272] Association Class

[4273] The MSSQL_UserLogin class represents an association between a database user and the login used to authenticate the user.

[4274] References

[4275] [key]MSSQL_Login Antecedent

[4276] Access Type: Read-only

[4277] The Antecedent property references the login used to authenticate the user referenced by the Dependent property.

[4278] [key] MSSQL User Dependent Access Type: Read-only

[4279] The Dependent property references a database user.

[4280] MSSQL_UserStoredProcedurePermission:

[4281] Association Class

[4282] The MSSQL_UserStoredProcedurePermission class represents the permissions granted to a user for a stored procedure. The instances of this class represent only the permission that has been explicitly granted or denied to the user object. For example, if a user has permissions to access a stored procedure by virtue of being a member of a certain database role, then there will not be a permission association instance between the user and the stored procedure.

[4283] Properties

[4284] boolean Granted

[4285] Access Type: Read/Write

[4286] The Granted property indicates whether the permission has been granted or denied. A value of True indicates that the permission has been granted. A value of False indicates that permission has been denied.

[4287] [key] uint32 PrivilegeType

[4288] Access Type: Read/Write

[4289] The PrivilegeType property indicates the type of privilege that has been granted or denied.

Value	Description
0	Unknown
1	Permission to query a table
2	Permission to add rows to a table
4	Permission to update rows of a table
8	Permission to delete rows of a table
16	Permission to execute a stored procedure
32	Permission to grant DRI on a table
63	All privileges applicable to the database object
128	Permission to create and own a table
256	Permission to create and own a database
512	Permission to create and own a view
1024	Permission to create and own a stored procedure
2048	Permission to backup a database
4096	Permission to create a default
8192	Permission to backup a database transaction log
16384	Permission to create a rule
32768	Permission to backup to a table
65366	Permission to create a user defined function
130944	All privileges applicable to the database

[4290] References

[4291] [key] MSSQL_StoredProcedure Element

[4292] Access Type: Read-only

[4293] The Element property references a stored procedure.

[4294] [key] MSSQL User Grantee

[4295] Access Type: Read/Write

[4296] The Grantee property references a user for whom the permissions have been defined.

[4297] MSSQL_UserTablePermission: MSSQL_Permission

[4298] Association Class

[4299] The MSSQL_UserTablePermission class represents the permissions granted to a user for a table. The instances of this class represent only the permission that has been explicitly granted or denied to the user object. For example, if a user has permissions to access a table by virtue of being a member of a certain database role, then there will not be a permission association instance between the user and the table.

[4300] Properties

[4301] string ColumnName []

[4302] Access Type: Read/Write

[4303] The ColumnName property specifies the columns within the table for which the permission is specified. If this property is null, then the permission applies to all columns in the table, otherwise it applies only to the columns indicated in this property.

[4304] boolean Granted

[4305] Access Type: Read/Write

[4306] The Granted property indicates whether the permission has been granted or denied. A value of True indicates that the permission has been granted. A value of False indicates that permission has been denied.

[4307] [key] uint32 PrivilegeType

[4308] Access Type: Read/Write

[4309] The PrivilegeType property indicates the type of privilege that has been granted or denied.

Value	Description
0	Unknown
1	Permission to query a table
2	Permission to add rows to a table
4	Permission to update rows of a table
8	Permission to delete rows of a table
16	Permission to execute a stored procedure
32	Permission to grant DRI on a table
63	All privileges applicable to the database object
128	Permission to create and own a table
256	Permission to create and own a database
512	Permission to create and own a view
1024	Permission to create and own a stored procedure
2048	Permission to backup a database
4096	Permission to create a default
8192	Permission to backup a database transaction log
16384	Permission to create a rule
32768	Permission to backup to a table
65366	Permission to create a user defined function
130944	All privileges applicable to the database

[4310] References

[4311] [key]MSSQL_Table Element

[4312] Access Type: Read-only

[4313] The Element property references a table in SQL Server™.

[4314] [key]MSSQL_User Grantee

[4315] Access Type: Read-only

[4316] The Grantee property references a user for whom the permissions have been defined.

[4317] MSSQL_UserUserDefinedFunctionPermission: MSSQL_Permission

[4318] Association Class

[4319] The MSSQL_UserUserDefinedFunction-Permission class represents the permissions granted to a user for a stored procedure. The instances of this class represent only the permission that has been explicitly granted or denied to the user object. For example, if a user has permissions to access a user defined function by virtue of being a member of a certain database role, then there will not be a permission association instance between the user and the user defined function.

[4320] Properties

[4321] boolean Granted

[4322] Access Type: Read/Write

[4323] The Granted property indicates whether the permission has been granted or denied. A value of True indicates that the permission has been granted. A value of False indicates that permission has been denied.

[4324] [Key] unint32 PrivilegeType

[4325] Access Type: Read/Write

[4326] The PrivilegeType property indicates the type of privilege that has been granted or denied.

Value	Description
0	Unknown
1	Permission to query a table
2	Permission to add rows to a table
4	Permission to update rows of a table
8	Permission to delete rows of a table
16	Permission to execute a stored procedure
32	Permission to grant DRI on a table
63	All privileges applicable to the database object
128	Permission to create and own a table
256	Permission to create and own a database
512	Permission to create and own a view
1024	Permission to create and own a stored procedure
2048	Permission to backup a database
4096	Permission to create a default
8192	Permission to backup a database transaction log
16384	Permission to create a rule
32768	Permission to backup to a table
65366	Permission to create a user defined function
130944	All privileges applicable to the database

[4327] References

[4328] [key]MSSQL_UserDefinedFunction Element

[4329] Access Type: Read/Write

[4330] The Element property references a user-defined function.

[4331] [key]MSSQL_User Grantee

[4332] Access Type: Read/Write

[4333] The Grantee property references a user for whom the permissions have been defined.

[4334] MSSQL_UserViewPermission: MSSQL_Permission

[4335] Association Class

[4336] The MSSQL_UserViewPermission class represents the permissions granted to a user for a view. The instances of this class represent only the permission that has been explicitly granted or denied to the user object. For example, if a user has permissions to access a view by virtue of being a member of a certain database role, then there will not be a permission association instance between the user and the view.

[4337] Properties

[4338] string ColumnName []

[4339] Access Type: Read/Write

[4340] The ColumnName property specifies the columns within the view for which the permission is specified. If this property is null, then the permission applies to all columns in the view, otherwise it applies only to the columns indicated in this property.

[4341] boolean Granted

[4342] Access Type: Read/Write

[4343] The Granted property indicates whether the permission has been granted or denied. A value of True indicates that the permission has been granted. A value of False indicates that permission has been denied.

[4344] [key] uint32 PrivilegeType

[4345] Access Type: Read/Write

[4346] The PrivilegeType property indicates the type of privilege that has been granted or denied.

Value	Description
0	Unknown
1	Permission to query a table
2	Permission to add rows to a table
4	Permission to update rows of a table
8	Permission to delete rows of a table
16	Permission to execute a stored procedure
32	Permission to grant DRI on a table
63	All privileges applicable to the database object
128	Permission to create and own a table
256	Permission to create and own a database
512	Permission to create and own a view
1024	Permission to create and own a stored procedure
2048	Permission to backup a database
4096	Permission to create a default
8192	Permission to backup a database transaction log
16384	Permission to create a rule
32768	Permission to backup to a table
65366	Permission to create a user defined function
130944	All privileges applicable to the database

[4347] References

[4348] [key]MSSQL_View Element

[4349] Access Type: Read-only

[4350] The Element property references a SQL Server™ view.

[4351] [key]MSSQL_User Grantee

[4352] Access Type: Read-only

[4353] The Grantee property references a user for which the permissions have been defined.

[4354] MSSQL_View: MSSQL_DBMSObject

[4355] The MSSQL_View class represents view tables in the database.

[4356] Properties

[4357] boolean AnsiNullsStatus

[4358] Access Type: Read-only

[4359] The AnsiNullsStatus property returns TRUE when the database object referenced depends on a table exhibiting SQL-92 NULL handling behavior.

[4360] string Caption

[4361] Access Type: Read-only

[4362] The Caption property is a short textual description (one-line string) of the object.

[4363] Maximum Length: 64

[4364] datetime CreateDate

[4365] Access Type: Read-only

[4366] The CreateDate property indicates the time and date on which the view was created. Note that creation date may be different from the install date in cases where the object is created in one place and then installed elsewhere.

[4367] [key] string DatabaseName

[4368] Access Type: Read-only

[4369] The DatabaseName property indicates the name of the database that the object is a part of.

[4370] Maximum Length: 128

[4371] string Description

[4372] Access Type: Read-only

[4373] The Description property provides a textual description of the object.

[4374] datetime InstallDate

[4375] Access Type: Read-only

[4376] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[4377] [key] string Name

[4378] Access Type: Read-only

[4379] The Name property defines the label by which the object is known.

[4380] boolean QuotedIdentifierStatus

[4381] Access Type: Read-only

[4382] The QuotedIdentifierStatus property controls Microsoft® SQL Server™ interpretation of identifier strings in statements submitted for execution. When TRUE, identifiers can be delimited by double quotation marks and character literal values must be delimited by single quotation marks. When FALSE, identifiers cannot be quoted and must follow all Transact-SQL rules for identifiers. For example, character literal values can be delimited by either single or double quotation marks.

[4383] [key] string SQLServerName

[4384] Access Type: Read-only

[4385] The SQLServerName property indicates the name of the SQL Server™ installation that the object is a part of.

[4386] Maximum Length: 128

[4387] string Status

[4388] Access Type: Read-only

[4389] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[4390] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[4391] Maximum Length: 10

[4392] boolean SystemObject

[4393] Access Type: Read-only

[4394] The SystemObject property indicates whether the object is owned by Microsoft®. A value of True indicates that the object implementation is owned by Microsoft®.

[4395] string Text

[4396] Access Type: Read/Write

[4397] The Text property indicates the Transact-SQL or other script that defines the object. Note that there is a special requirement for the name of the object is specified in the CREATE statement. The name of the object has to be in the form that includes the name of the owner. For example, in order to create an object named "Some Object", one would need to specify it as [dbo].[Some Object].

[4398] Methods

[4399] The MSSQL_View class supports the following methods:

Method Name	Description
ExportData	The ExportData method is used to copy data from a Microsoft ® SQL Server ™ database to a data file.
Rename	The Rename method is used to rename an instance of a view.

[4400] Associations

[4401] MSSQL_View is associated to MSSQL_Database as the PartComponent

[4402] property of the MSSQL_DatabaseView association.

[4403] MSSQL_View is associated to MSSQL_User as the Element property of the MSSQL_UserViewPermission association.

[4404] MSSQL_View is associated to MSSQL_DatabaseRole as the Element property of the MSSQL_DatabaseRoleViewPermission association.

[4405] C. Win32 Classes

[4406] Win32_Account: CIM_LogicalElement

[4407] Abstract Class

[4408] The Win32_Account class contains information about user accounts and group accounts known to the Win32 system. User or group names recognized by a Windows NT domain are descendents (or members) of this class. The Win32_Account class is not included in a default hardware inventory operation.

[4409] Properties

[4410] string Caption

[4411] Access Type: Read-only

[4412] The Caption property is a short textual description (one-line string) of the object.

[4413] Maximum Length: 64

[4414] string Description

[4415] Access Type: Read-only

[4416] The Description property provides a textual description of the object.

[4417] [key] string Domain

[4418] Access Type: Read-only

[4419] The Domain property indicates the name of the Windows domain to which a group or user belongs. Example: NA-SALES

[4420] datetime InstallDate

[4421] Access Type: Read-only

[4422] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[4423] [key] string Name

[4424] Access Type: Read-only

[4425] The Name property indicates the name of the Win32 system account on the domain specified by the Domain member of this class.

[4426] string SID

[4427] Access Type: Read-only

[4428] The SID property contains the security identifier (SID) for this account. a SID is a string value of variable length used to identify a trustee. Each account has a unique SID issued by an

authority (such as a Windows domain), stored in a security database. When a user logs on, the system retrieves the user's SID from the database and places it in the user's access token. The system uses the SID in the user's access token to identify the user in all subsequent interactions with Windows security. When a SID has been used as the unique identifier for a user or group, it cannot be used again to identify another user or group.

[4429] uint8 SIDType

[4430] Access Type: Read-only

[4431] The SIDType property contains enumerated values that specify the type of security identifier (SID).

Value	Description	Explanation
1	SidTypeUser	Indicates a user SID.
2	SidTypeGroup	Indicates a group SID.
3	SidTypeDomain	Indicates a domain SID.
4	SidTypeAlias	Indicates an alias SID.
5	SidTypeWellKnownGroup	Indicates a SID for a well-known group.
6	SidTypeDeletedAccount	Indicates a SID for a deleted account.
7	SidTypeInvalid	Indicates an invalid SID.
8	SidTypeUnknown	Indicates an unknown SID type.
9	SidTypeComputer	Indicates a SID for a computer.

[4432] string Status

[4433] Access Type: Read-only

[4434] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror-resilvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[4435] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[4436] Maximum Length: 10

[4437] Associations

[4438] Win32_Account is associated to Win32_Group as the PartComponent property of the Win32_GroupUser association.

[4439] Win32_BaseService: CIM_Service

[4440] Abstract Class

[4441] The Win32_BaseService class represents executable objects that are installed in a registry database maintained by the Service Control Manager. The executable file associated with a service can be started at boot time by a boot program or by the system. It can also be started on-demand by the Service Control Manager. Any service or process that is not owned by a specific user, and that provides an interface to some functionality supported by the computer system, is a descendent (or member) of this class. Example: The dynamic host configuration protocol (DHCP) client service on a Windows NT/Windows 2000 computer system.

[4442] Properties

[4443] boolean AcceptPause

[4444] Access Type: Read-only

[4445] The AcceptPause property indicates whether the service can be paused. Values: TRUE or FALSE. A value of TRUE indicates the service can be paused.

[4446] boolean AcceptStop

[4447] Access Type: Read-only

[4448] The AcceptStop property indicates whether the service can be stopped. Values: TRUE or FALSE. A value of TRUE indicates the service can be stopped.

[4449] string Caption

[4450] Access Type: Read-only

[4451] The Caption property is a short textual description (one-line string) of the object.

[4452] Maximum Length: 64

[4453] string CreationClassName

[4454] Access Type: Read-only

[4455] CreationClassName indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.

[4456] string Description

[4457] Access Type: Read-only

[4458] The Description property provides a textual description of the object.

[4459] boolean DesktopInteract

[4460] Access Type: Read-only

[4461] The DesktopInteract property indicates whether the service can create or communicate with windows on the desktop. Values: TRUE or FALSE. A value of TRUE indicates the service can create or communicate with windows on the desktop.

[4462] string DisplayName

[4463] Access Type: Read-only

[4464] The DisplayName property indicates the display name of the service. This string has a maximum length of 256 characters. The name is

case-preserved in the Service Control Manager. DisplayName comparisons are always case-insensitive. Constraints: Accepts the same value as the Name property. Example: Atdisk.

[4465] string ErrorControl

[4466] Access Type: Read-only

[4467] If this service fails to start during startup, the ErrorControl property specifies the severity of the error. The value indicates the action taken by the startup program if failure occurs. All errors are logged by the computer system. The computer system does not notify the user of "Ignore" errors. With "Normal" errorsthe user is notified. With "Severe" errors, the system is restarted with the last-known-good configuration. Finally, on"Critical" errors the system attempts to restart with a good configuration.

[4468] Values are: "Ignore", "Normal", "Severe", "Critical", "Unknown"

[4469] uint32 ExitCode

[4470] Access Type: Read-only

[4471] The ExitCode property specifies a Win32 error code defining any problems encountered in starting or stopping the service. This property is set to ERROR_SERVICE_SPECIFIC_ERROR (1066) when the error is unique to the service represented by this class, and information about the error is available in the ServiceSpecificExitCode member. The service sets this value to NO_ERROR when running, and again upon normal termination.

[4472] datetime InstallDate

[4473] Access Type: Read-only

[4474] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[4475] [key] string Name

[4476] Access Type: Read-only

[4477] The Name property uniquely identifies the service and provides an indication of the functionality that is managed. This functionality is described in more detail in the object's Description property.

[4478] string PathName

[4479] Access Type: Read-only

[4480] The PathName property contains the fully qualified path to the service binary file that implements the service. Example: \SystemRoot\System32\drivers\afd.sys

[4481] uint32 ServiceSpecificExitCode

[4482] Access Type: Read-only

[4483] The ServiceSpecificExitCode property specifies a service-specific error code for errors that occur while the service is either starting or stopping. The exit codes are defined by the service represented by this class. This value is only set when the ExitCodeproperty value is ERROR_SERVICE_SPECIFIC_ERROR, 1066.

[4484] string ServiceType

[4485] Access Type: Read-only

[4486] The ServiceType property supplies the type of service provided to calling processes.

[4487] Values are: "Kernel Driver", "File System Driver", "Adapter", "Recognizer Driver", "Own Process", "Share Process", "Interactive Process"

[4488] boolean Started

[4489] Access Type: Read-only

[4490] Started is a boolean indicating whether the service has been started (TRUE), or stopped (FALSE).

[4491] string StartMode

[4492] Access Type: Read-only

[4493] The StartMode property indicates the start mode of the Win32 base service. "Boot" specifies a device driver started by the operating system loader. This value is valid only for driver services. "System" specifies a device driver started by the lolnitSystem function. This value is valid only for driver services. "Automatic" specifies a service to be started automatically by the service control manager during system startup. "Manual" specifies a service to be started by the service control manager when a process calls the StartService function. "Disabled" specifies a service that can no longer be started.

[4494] Values are: "Boot""System", "Auto", "Manual", "Disabled"

[4495] string StartName

[4496] Access Type: Read-only

[4497] The StartName property indicates the account name under which the service runs. Depending on the service type, the account name may be in the form of "DomainName\Username". The service process will be logged using one of these two forms when it runs. If the account belongs to the built-in domain, ".\Username" can be specified. If NULL is specified, the service will be logged on as the LocalSystem account. For kernel or system level drivers, StartName contains the driver object name (that is, \FileSystem\Rdr or \Driver\Xns) which the input and output (I/O) system uses to load the device driver. Additionally, if NULL is specified, the driver runs with a default object name created by the I/O system based on the service name. Example: DWDOM\Admin.

[4498] string State

[4499] Access Type: Read-only

[4500] The State property indicates the current state of the base service.

[4501] Values are: "Stopped", "Start Pending", "Stop Pending", "Running", "Continue Pending", "Pause Pending", "Paused", "Unknown"

[4502] string Status

[4503] Access Type: Read-only

[4504] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror-resilvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[4505] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[4506] Maximum Length: 10

[4507] string SystemCreationClassName

[4508] Access Type: Read-only

[4509] The type name of the system that hosts this service.

[4510] string SystemName

[4511] Access Type: Read-only

[4512] The name of the system that hosts this service.

[4513] uint32 TagId

[4514] Access Type: Read-only

[4515] The TagId property specifies a unique tag value for this service in the group. A value of 0 indicates that the service has not been assigned a tag. A tag can be used for ordering service startup within a load order group by specifying a tag order vector in the registry located at: HKEY_LOCAL_MACHINE\SystemCurrent ControlSet\Control\GroupOr derList. Tags are only evaluated for Kernel Driver and File System Driver start type services that have "Boot" or "System" start modes.

[4516] Methods

[4517] The Win32_BaseService class supports the following methods:

Method Name	Description
Change	The Change method modifies a service.
ChangeStartMode	The ChangeStartMode method modifies the StartMode of a service.
Create	The Create method creates a new service.
Delete	The Delete method deletes an existing service.

-continued

Method Name	Description
InterrogateService	The InterrogateService method requests that the service update its state to the service manager.
PauseService	The PauseService method attempts to place the service in the paused state.
ResumeService	The ResumeService method attempts to place the service in the resumed state.
StartService	The StartService method places the service in the started state.
StopService	The StopService method places the service in the stopped state.
UserControlService	The UserControlService method attempts to send a user-defined control code to a service.

[4518] Win32_Group: Win32_Account

[4519] The Win32_Group class represents data about a group account. A group account allows access privileges to be changed for a list of users. Example:

[4520] Marketing2.

[4521] Properties

[4522] string Caption

[4523] Access Type: Read-only

[4524] The Caption property is a short textual description (one-line string) of the object.

[4525] Maximum Length: 64

[4526] string Description

[4527] Access Type: Read-only

[4528] The Description property provides a textual description of the object.

[4529] [key] string Domain

[4530] Access Type: Read-only

[4531] The Domain property indicates the name of the Windows domain to which the group account belongs. Example: NA-SALES

[4532] datetime InstallDate

[4533] Access Type: Read-only

[4534] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[4535] [key] string Name

[4536] Access Type: Read-only

[4537] The Name property indicates the name of the Win32 group account on the domain specified by the Domain member of this class.

[4538] string SID

[4539] Access Type: Read-only

[4540] The SID property contains the security identifier (SID) for this account. a SID is a string value of variable length used to identify a trustee. Each account has a unique SID issued by an

authority (such as a Windows domain), stored in a security database. When a user logs on, the system retrieves the user's SID from the database and places it in the user's access token. The system uses the SID in the user's access token to identify the user in all subsequent interactions with Windows security. When a SID has been used as the unique identifier for a user or group, it cannot be used again to identify another user or group.

[4541] uint8 SIDType

[4542] Access Type: Read-only

[4543] The SIDType property contains enumerated values that specify the type of security identifier (SID).

Value	Description	Explanation
1	SidTypeUser	Indicates a user SID.
2	SidTypeGroup	Indicates a group SID.
3	SidTypeDomain	Indicates a domain SID.
4	SidTypeAlias	Indicates an alias SID.
5	SidTypeWellKnownGroup	Indicates a SID for a well-known group.
6	SidTypeDeletedAccount	Indicates a SID for a deleted account.
7	SidTypeInvalid	Indicates an invalid SID.
8	SidTypeUnknown	Indicates an unknown SID type.
9	SidTypeComputer	Indicates a SID for a computer.

[4544] string Status

[4545] Access Type: Read-only

[4546] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror-resilvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[4547] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[4548] Maximum Length: 10

[4549] Associations

[4550] Win32_Group is associated to Win32_Account as the GroupComponent property of the Win32_GroupUser association.

[4551] Win32_Group is associated to MSSQL_Login as the Antecedent property of the MSSQL_LoginWin32Group association.

[4552] Win32_GroupUser: CIM_Component

[4553] Association Class

[4554] The Win32_GroupUser class represents an association between a group and an account that is a member of that group.

[4555] References

[4556] [key]Win32_Group GroupComponent

[4557] Access Type: Read-only

[4558] The GroupComponent reference represents a group that the account is a member of.

[4559] [key]Win32_Account PartComponent

[4560] Access Type: Read-only

[4561] The PartComponent reference represents a user or system account that is a part of a group of accounts.

[4562] Win32_Process: CIM_Process

[4563] The Win32_Process class represents a sequence of events on a Win32 system. Any sequence consisting of the interaction of one or more processors or interpreters, some executable code, and a set of inputs, is a descendent 22 (or member) of this class. Example: A client application running on a Win32 system.

[4564] Properties

[4565] string Caption

[4566] Access Type: Read-only

[4567] The Caption property is a short textual description (one-line string) of the object.

[4568] Maximum Length: 64

[4569] string CreationClassName

[4570] Access Type: Read-only

[4571] The inherited CreationClassName property indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.

[4572] datetime CreationDate

[4573] Access Type: Read-only

[4574] Time that the process began executing

[4575] string CSCreationClassName

[4576] Access Type: Read-only

[4577] The inherited CSCreationClassName property is a string indicating the class of the computer system.

[4578] string CSName

[4579] Access Type: Read-only

[4580] The inherited CSName property is a string indicating the name of the computer system.

[4581] string Description

[4582] Access Type: Read-only

[4583] The Description property provides a textual description of the object.

[4584] string ExecutablePath

[4585] Access Type: Read-only

[4586] The ExecutablePath property indicates the path to the executable file of the process. Example: C:\WINDOWS\EXPLORER.EXE

[4587] Privilleges Required: Debug (SeDebugPrivilege)

[4588] uint16 ExecutionState

[4589] Access Type: Read-only

[4590] Indicates the current operating condition of the process.

Value	Description
0	Unknown
1	Other
2	Ready
3	Running
4	Blocked
5	Suspended Blocked
6	Suspended Ready

[4591] [Key] string Handle

[4592] Access Type: Read-only

[4593] A strind used to identify the process. A process ID is a process handle.

[4594] uint32 HandleCount

[4595] Access Type: Read-only

[4596] The HandleCount property specifies the total number of handles currently open by this process. This number is the sum of the handles currently open by each thread in this process. A handle is used to examine or modify the system resources. Each handle has an entry in an internally maintained table. These entries contain the addresses of the resources and the means to identify the resource type.

[4597] datetime InstallDate

[4598] Access Type: Read-only

[4599] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[4600] uint64 KernelModeTime

[4601] Access Type: Read-only

[4602] Time in kernel mode, in milliseconds. If this information is not available, a value of 0 should be used.

[4603] Units: Milliseconds (ms)

[4604] uint32 MaximumWorkingSetSize

[4605] Access Type: Read-only

[4606] The MaximumWorkingSetSize property indicates the maximum working set size of the process. The working set of a process is the set of memory pages currently visible to the process in physical RAM. These pages are resident and available for an application to use without triggering a page fault. Example: 1413120.

[4607] Privileges Required: Debug (SeDebugPrivilege)

[4608] Units: Kilobytes

[4609] uint32 Minimum WorkingSetSize

[4610] Access Type: Read-only

[4611] The MinimumWorkingSetSize property indicates the minimum working set size of the process. The working set of a process is the set of memory pages currently visible to the process in physical RAM. These pages are resident and available for an application to use without triggering a page fault. Example: 20480.

[4612] Privileges Required: Debug (SeDebugPrivilege)

[4613] Units: Kilobytes

[4614] string Name

[4615] Access Type: Read-only

[4616] The Name property defines the label by which the object is known. When subclassed, the Name property can be overridden to be a Key property.

[4617] string OSCreationClassName

[4618] Access Type: Read-only

[4619] The inherited OSCreationClassName property indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.

[4620] string OSName

[4621] Access Type: Read-only

[4622] The inherited OSName property serves as key of an operating system instance within a computer system.

[4623] uint64 OtherOperationCount

[4624] Access Type: Read-only

[4625] The OtherOperationCount property specifies the number of I/O operations performed, other than read and write operations.

[4626] uint64 OtherTransferCount

[4627] Access Type: Read-only

[4628] The OtherTransferCount property specifies the amount of data transferred during operations other than read and write operations.

[4629] Units: Bytes

[4630] uint32 PageFaults

[4631] Access Type: Read-only

[4632] The PageFaults property indicates the number of page faults generated by the process. Example: 10

[4633] uint32 PageFileUsage

[4634] Access Type: Read-only

[4635] The PageFileUsage property indicates the amount of page file space currently being used by the process. Example: 102435

[4636] Units: Kilobytes

[4637] uint32 ParentProcessId

[4638] Access Type: Read-only

[4639] The ParentProcessId property specifies the unique identifier of the process that created this process. Process identifier numbers are reused, so they only identify a process for the lifetime of that process. It is possible that the process identified by ParentProcessId has terminated, so ParentProcessId may not refer to a running process. It is also possible that ParentProcessId incorrectly refers to a process which re-used that process identifier. The CreationDate property can be used to determine whether the specified parent was created after this process was created.

[4640] uint32 PeakPageFileUsage

[4641] Access Type: Read-only

[4642] The PeakPageFileUsage property indicates the maximum amount of page file space used during the life of the process. Example: 102367

[4643] Units: Kilobytes

[4644] uint64 PeakVirtualSize

[4645] Access Type: Read-only

[4646] The PeakVirtualSize property specifies the maximum virtual address space the process has used at any one time. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. However, virtual space is finite, and by using too much, the process might limit its ability to load libraries.

[4647] Units: Bytes

[4648] uint32 PeakWorkingSetSize

[4649] Access Type: Read-only

[4650] The PeakWorkingSetSize property indicates the peak working set size of the process. Example: 1413120

[4651] Units: Kilobytes

[4652] uint32 Priority

[4653] Access Type: Read-only

[4654] The Priority property indicates the scheduling priority of the process within the operating system. The higher the value, the higher priority the process receives. Priority values can range from 0 (lowest priority) to 31 (highest priority). Example: 7.

[4655] uint64 PrivatePageCount

[4656] Access Type: Read-only

[4657] The PrivatePageCount property specifies the current number of pages allocated that are accessible only to this process.

[4658] uint32 processId

[4659] Access Type: Read-only

[4660] The ProcessId property contains the global process identifier that can be used to identify a process. The value is valid from the creation of the process until the process is terminated.

[4661] uint32 QuotaNonPagedPoolUsage

[4662] Access Type: Read-only

[4663] The QuotaNonPagedPoolUsage property indicates the quota amount of non-paged pool usage for the process. Example: 15

[4664] uint32 QuotaPagedPoolUsage

[4665] Access Type: Read-only

[4666] The QuotaPagedPoolUsage property indicates the quota amount of paged pool usage for the process. Example: 22

[4667] uint32 QuotaPeakNonPagedPoolUsage

[4668] Access Type: Read-only

[4669] The QuotaPeakNonPagedPoolUsage property indicates the peak quota amount of non-paged pool usage for the process. Example: 31

[4670] uint32 QuotaPeakPagedPoolUsage

[4671] Access Type: Read-only

[4672] The QuotaPeakPagedPoolUsage property indicates the peak quota amount of paged pool usage for the process. Example: 31

[4673] uint64 ReadOperationCount

[4674] Access Type: Read-only

[4675] The ReadOperationCount property specifies the number of read operations performed.

[4676] uint64 ReadTransferCount

[4677] Access Type: Read-only

[4678] The ReadTransferCount property specifies the amount of data read.

[4679] Units: Bytes

[4680] uint32 SessionId

[4681] Access Type: Read-only

[4682] The SessionId property specifies the unique identifier that is generated by the operating system when the session is created. A session spans a period of time from log in to log out on a particular system.

[4683] string Status

[4684] Access Type: Read-only

[4685] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror re-silvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[4686] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[4687] Maximum Length: 10

[4688] datetime TerminationDate

[4689] Access Type: Read-only

[4690] Time that the process was stopped or terminated.

[4691] uint**32** ThreadCount

[4692] Access Type: Read-only

[4693] The ThreadCount property specifies the number of active threads in this process. An instruction is the basic unit of execution in a processor, and a thread is the object that executes instructions. Every running process has at least one thread. This property is for computers running Windows NT only.

[4694] uint**64** UserModeTime

[4695] Access Type: Read-only

[4696] Time in user mode, in milliseconds. If this information is not available, a value of 0 should be used.

[4697] Units: Milliseconds (ms)

[4698] uint**64** VirtualSize

[4699] Access Type: Read-only

[4700] The VirtualSize property specifies the current size in bytes of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and by using too much, the process can limit its ability to load libraries.

[4701] Units: Bytes

[4702] string WindowsVersion

[4703] Access Type: Read-only

[4704] The WindowsVersion property indicates the version of Windows in which the process is running. Example: 4.0

[4705] uint**64** WorkingSetSize

[4706] Access Type: Read-only

[4707] The amount of memory in bytes that a process needs to execute efficiently, for an operating system that uses page-based memory management. If an insufficient amount of memory is available (<working set size), thrashing will occur. If this information is not known, NULL or 0 should be entered. If this data is provided, it could be monitored to understand a process' changing memory requirements as execution proceeds.

[4708] Units: Bytes

[4709] uint**64** WriteOperationCount

[4710] Access Type: Read-only

[4711] The WriteOperationCount property specifies the number of write operations performed.

[4712] uint**64** WriteTransferCount

[4713] Access Type: Read-only

[4714] The WriteTransferCount property specifies the amount of data written.

[4715] Units: Bytes

[4716] Methods

[4717] The Win32_Process class supports the following methods:

Method Name	Description
Create	The Create method creates a new process.
GetOwner	The GetOwner method retrieves the user name and domain name under which the process is running.
GetOwnerSid	The GetOwnerSid method retrieves the security identifier (SID) for the owner of this process.
Terminate	The Terminate method terminates a process and all of its threads.

[4718] Win32_Service: Win32_BaseService

[4719] The Win32_Service class represents a service on a Win32 computer system. A service application conforms to the interface rules of the Service Control Manager (SCM) and can be started by a user automatically at system boot through the Services control panel utility, or by an application that uses the service functions included in the Win32 API. Services can execute even when no user is logged on to the system.

[4720] Properties

[4721] boolean AcceptPause

[4722] Access Type: Read-only

[4723] The AcceptPause property indicates whether the service can be paused. Values: TRUE or FALSE. A value of TRUE indicates the service can be paused.

[4724] boolean AcceptStop

[4725] Access Type: Read-only

[4726] The AcceptStop property indicates whether the service can be stopped. Values: TRUE or FALSE. A value of TRUE indicates the service can be stopped.

[4727] string Caption

[4728] Access Type: Read-only

[4729] The Caption property is a short textual description (one-line string) of the object.

[4730] Maximum Length: 64

[4731] uint32 Checkpoint

[4732] Access Type: Read-only

[4733] The CheckPoint property specifies a value that the service increments periodically to report its progress during a lengthy start, stop, pause, or continue operation. For example, the service should increment this value as it completes each step of its initialization when it is starting up. The user interface program that invoked the operation on the service uses this value to track the progress of the service during a lengthy operation. This value is not valid and should be zero when the service does not have a start, stop, pause, or continue operation pending.

[4734] string CreationClassName

[4735] Access Type: Read-only

[4736] CreationClassName indicates the name of the class or the subclass used in the creation of an instance. When used with the other key properties of this class, this property allows all instances of this class and its subclasses to be uniquely identified.

[4737] string Description

[4738] Access Type: Read-only

[4739] The Description property provides a textual description of the object.

[4740] boolean DesktopInteract

[4741] Access Type: Read-only

[4742] The Desktopinteract property indicates whether the service can create or communicate with windows on the desktop. Values: TRUE or FALSE. A value of TRUE indicates the service can create or communicate with windows on the desktop.

[4743] string DisplayName

[4744] Access Type: Read-only

[4745] The DisplayName property indicates the display name of the service. This string has a maximum length of 256 characters. The name is case-preserved in the Service Control Manager. DisplayName comparisons are always case-insensitive. Constraints: Accepts the same value as the Name property. Example: Atdisk.

[4746] string ErrorControl

[4747] Access Type: Read-only

[4748] If this service fails to start during startup, the ErrorControl property specifies the severity of the error. The value indicates the action taken by the startup program if failure occurs. All errors are logged by the computer system. The computer system does not notify the user of "Ignore" errors. With "Normal" errorsthe user is notified. With "Severe" errors, the system is restarted with the last-known-good configuration. Finally, on"Critical" errors the system attempts to restart with a good configuration.

[4749] Values are: "Ignore", "Normal", "Severe", "Critical", "Unknown"

[4750] uint32 ExitCode

[4751] Access Type: Read-only

[4752] The ExitCode property specifies a Win32 error code defining any problems encountered in starting or stopping the service. This property is set to ERROR_SERVICE_SPECIFIC_ERROR (1066) when the error is unique to the service represented by this class, and information about the error is available in the ServiceSpecificExitCode member. The service sets this value to NO_ERROR when running, and again upon normal termination.

[4753] datetime InstallDate

[4754] Access Type: Read-only

[4755] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[4756] [key] string Name

[4757] Access Type: Read-only

[4758] The Name property uniquely identifies the service and provides an indication of the functionality that is managed. This functionality is described in more detail in the object's Description property.

[4759] string PathName

[4760] Access Type: Read-only

[4761] The PathName property contains the fully qualified path to the service binary file that implements the service. Example: \System Root\System32\drivers\afd.sys

[4762] uint32 ProcessId

[4763] Access Type: Read-only

[4764] The ProcessId property specifies the process identifier of the service. Example: 324

[4765] uint32 ServiceSpecificExitCode

[4766] Access Type: Read-only

[4767] The ServiceSpecificExitCode property specifies a service-specific error code for errors that occur while the service is either starting or stopping. The exit codes are defined by the service represented by this class. This value is only set

when the ExitCodeproperty value is ERROR_SERVICE_SPECIFIC_ERROR, **1066**.

[**4768**] string ServiceType

[**4769**] Access Type: Read-only

[**4770**] The ServiceType property supplies the type of service provided to calling processes.

[**4771**] Values are: "Kernel Driver", "File System Driver", "Adapter", "Recognizer Driver", "Own Process", "Share Process", "Interactive Process"

[**4772**] boolean Started

[**4773**] Access Type: Read-only

[**4774**] Started is a boolean indicating whether the service has been started (TRUE), or stopped (FALSE).

[**4775**] string StartMode

[**4776**] Access Type: Read-only

[**4777**] The StartMode property indicates the start mode of the Win32 base service. "Boot" specifies a device driver started by the operating system loader. This value is valid only for driver services. "System" specifies a device driver started by the lolnitSystem function. This value is valid only for driver services. "Automatic" specifies a service to be started automatically by the service control manager during system startup. "Manual" specifies a service to be started by the service control manager when a process calls the StartService function. "Disabled" specifies a service that can no longer be started.

[**4778**] Values are: "Boot", "System", "Auto", "Manual", "Disabled"

[**4779**] string StartName

[**4780**] Access Type: Read-only

[**4781**] The StartName property indicates the account name under which the service runs. Depending on the service type, the account name may be in the form of "DomainName\Username". The service process will be logged using one of these two forms when it runs. If the account belongs to the built-in domain, "\Username" can be specified. If NULL is specified, the service will be logged on as the LocalSystem account. For kernel or system level drivers, StartName contains the driver object name (that is, \FileSystem\Rdr or \Driver\Xns) which the input and output (I/O) system uses to load the device driver. Additionally, if NULL is specified, the driver runs with a default object name created by the I/O system based on the service name. Example: DWDOM\Admin.

[**4782**] string State

[**4783**] Access Type: Read-only

[**4784**] The State property indicates the current state of the base service.

[**4785**] Values are: "Stopped", "Start Pending", "Stop Pending", "Running", "Continue Pending", "Pause Pending""Paused", "Unknown"

[**4786**] string Status

[**4787**] Access Type: Read-only

[**4788**] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror-resilvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[**4789**] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[**4790**] Maximum Length: 10

[**4791**] string SystemCreationClassName

[**4792**] Access Type: Read-only

[**4793**] The type name of the system that hosts this service.

[**4794**] string SystemName

[**4795**] Access Type: Read-only

[**4796**] The name of the system that hosts this service.

[**4797**] uint**32** TagId

[**4798**] Access Type: Read-only

[**4799**] The TagId property specifies a unique tag value for this service in the group. A value of 0 indicates that the service has not been assigned a tag. A tag can be used for ordering service startup within a load order group by specifying a tag order vector in the registry located at: HKEY_LOCAL_MACHINE\System\Current ControlSet\Control\GroupOr derList. Tags are only evaluated for Kernel Driver and File System Driver start type services that have "Boot" or "System" start modes.

[**4800**] uint**32** WaitHint

[**4801**] Access Type: Read-only

[**4802**] The WaitHint property specifies the estimated time required (in milliseconds) for a pending start, stop, pause, or continue operation.

[**4803**] After the specified amount of time has elapsed, the service makes its next call to the SetServiceStatus function with either an incremented CheckPoint value or a change in CurrentState. If the amount of time specified by WaitHint passes, and CheckPoint has not been incremented,

or the CurrentState has not changed, the service control manager or service control program assumes that an error has occurred.

[4804] Methods

[4805] The Win32_Service class supports the following methods:

Method Name	Description
Change	The Change method modifies a service.
ChangeStartMode	The ChangeStartMode method modifies the StartMode of a service.
Create	The Create method creates a new service.
Delete	The Delete method deletes an existing service.
InterrogateService	The InterrogateService method requests that the service update its state to the service manager.
PauseService	The PauseService method attempts to place the service in the paused state.
ResumeService	The ResumeService method attempts to place the service in the resumed state.
StartService	The StartService method places the service in the started state.
StopService	The StopService method places the service in the stopped state.
UserControlService	The UserControlService method attempts to send a user-defined control code to a service.

[4806] Associations

[4807] Win32_Service is associated to MSSQL_FullTextCatalogService as the SystemElement property of the MSSQL_FullTextWin32Service association.

[4808] Win32 UserAccount: Win32 Account

[4809] The Win32_UserAccount class contains information about a user account on a Win32 system.

[4810] Properties

[4811] uint32 AccountType

[4812] Access Type: Read-only

[4813] The AccountType property contains flags describing the characteristics of a Win32 user account

Bit Position	Description	Explanation
8	Temporary duplicate account	Local user account for users whose primary account is in another domain. This account provides user access to this domain, but not to any domain that trusts this domain.
9	Normal account	Default account type that representing a typical user.
11	Interdomain trust account	Account is for a system domain that trusts other domains.
12	Workstation trust account	This is a computer account for a Windows NT/Windows 2000 machine that is a member of this domain.
13	Server trust account	Account is for a system backup domain controller that is a member of this domain.

[4814] string Caption

[4815] Access Type: Read-only

[4816] The Caption property is a short textual description (one-line string) of the object.

[4817] Maximum Length: 64

[4818] string Description

[4819] Access Type: Read-only

[4820] The Description property provides a textual description of the object.

[4821] boolean Disabled

[4822] Access Type: Read-only

[4823] The Disabled property determines whether the Win32 user account is disabled. Values: TRUE or FALSE. If TRUE, the user account is disabled.

[4824] [key] string Domain

[4825] Access Type: Read-only

[4826] The Domain property indicates the name of the Windows domain to which the user account belongs. Example: NA-SALES

[4827] string FullName

[4828] Access Type: Read-only

[4829] The FullName property indicates the full name of the local user. Example: Thomas Williams

[4830] datetime InstallDate

[4831] Access Type: Read-only

[4832] The InstallDate property is datetime value indicating when the object was installed. A lack of a value does not indicate that the object is not installed.

[4833] boolean Lockout

[4834] Access Type: Read-only

[4835] The Lockout property determines whether the user account is locked out of the Win32 system. Values: TRUE or FALSE. If TRUE, the user account is locked out.

[4836] [key] string Name

[4837] Access Type: Read-only

[4838] The Name property indicates the name of the Win32 user account on the domain specified by the Domain member of this class. Example: thomasw

[4839] boolean PasswordChangeable

[4840] Access Type: Read-only

[4841] The PasswordChangeable property determines whether the password on the Win32 user account can be changed. Values: TRUE or FALSE. If TRUE, the password can be changed.

[4842] boolean PasswordExpires

[4843] Access Type: Read-only

[4844] The PasswordExpires property determines whether the password on the Win32 user account will expire. Values: TRUE or FALSE. If TRUE, the password will expire.

[4845] boolean PasswordRequired

[4846] Access Type: Read-only

[4847] The PasswordRequired property determines whether a password is required on the Win32 user account. Values: TRUE or FALSE. If TRUE, a password is required.

[4848] string SID

[4849] Access Type: Read-only

[4850] The SID property contains the security identifier (SID) for this account. a SID is a string value of variable length used to identify a trustee. Each account has a unique SID issued by an authority (such as a Windows domain), stored in a security database. When a user logs on, the system retrieves the user's SID from the database and places it in the user's access token. The system uses the SID in the user's access token to identify the user in all subsequent interactions with Windows security. When a SID has been used as the unique identifier for a user or group, it cannot be used again to identify another user or group.

[4851] uint8 SIDType

[4852] Access Type: Read-only

[4853] The SIDType property contains enumerated values that specify the type of security identifier (SID).

Value	Description	Explanation
1	SidTypeUser	Indicates a user SID.
2	SidTypeGroup	Indicates a group SID.
3	SidTypeDomain	Indicates a domain SID.
4	SidTypeAlias	Indicates an alias SID.
5	SidTypeWellKnownGroup	Indicates a SID for a well-known group.
6	SidTypeDeletedAccount	Indicates a SID for a deleted account.
7	SidTypeInvalid	Indicates an invalid SID.
8	SidTypeUnknown	Indicates an unknown SID type.
9	SidTypeComputer	Indicates a SID for a computer.

[4854] string Status

[4855] Access Type: Read-only

[4856] The Status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are "OK", "Degraded" and "Pred Fail". "Pred Fail" indicates that an element may be functioning properly but predicting a failure in the near future. An example is a SMART-enabled hard drive. Non-operational statuses can also be specified. These are "Error", "Starting", "Stopping" and "Service". The latter, "Service", could apply during mirror-resilvering of a disk, reload of a user permissions list, or other administrative work. Not all such work is on-line, yet the managed element is neither "OK" nor in one of the other states.

[4857] Values are: "OK", "Error", "Degraded", "Unknown", "Pred Fail", "Starting", "Stopping", "Service"

[4858] Maximum Length: 10

[4859] Associations

[4860] Win32_UserAccount is associated to MSSQL-Login as the Antecedent property of the MSSQL_LoginWin32UserAccount association.

[4861] Conclusion

[4862] Although details of specific implementations and embodiments are described above, such details are intended to satisfy statutory disclosure obligations rather than to limit the scope of the following claims. Thus, the invention as defined by the claims is not limited to the specific features described above. Rather, the invention is claimed in any of its forms or modifications that fall within the proper scope of the appended claims.

1. A data structure stored on one or more computer-readable media that is instantiated in accordance with a schema, the schema comprising:

at least one database objects class that represents components of a database system;

at least one application system class that represents installation parameters of the database system;

at least one security class that represents security features pertaining to use of the database system; and

at least one physical storage class that represents physical files and file groups used by the database system to store data.

2. A data structure as recited in claim 1, wherein the database system comprises a SQL (structured query language) database.

3. A data structure as recited in claim 1, wherein the database objects class comprises multiple classes representing tables, views, stored procedures, indexes, constraints, and keys.

4. A data structure as recited in claim 1, wherein the application system class comprises multiple classes representing settings, services, and transaction and error logs used by the database system.

5. A data structure as recited in claim 1, wherein the security class comprises multiple classes representing users, roles, authentication login, and permissions for the users and the roles.

6. A data structure as recited in claim 5, wherein permissions are modeled as association classes between database objects class and the users and roles.

7. A data structure stored on one or more computer-readable media that is instantiated in accordance with a schema, the schema comprising:

a SQL setting class to represent settings that are used to configure an installation of the database;

a SQL database setting class to represent operational settings for the database;

a SQL configuration value class to represent configuration values;

a SQL language setting class to expose properties of a language record;

a SQL registry setting class to represent the installation and run-time parameters stored in the registry;

a SQL server connection setting class to represent default connection settings;

a SQL objects class to represent objects in a SQL database;

a SQL database class to represent instances of the SQL database;

a SQL server class to represent instances of a SQL server;

a SQL server configuration value class to associate a SQL server installation and the configured value settings for the installation;

a SQL server language setting class to associate a SQL server installation and its language settings;

a SQL server registry setting class to associate a SQL server installation and its registry setting; and

a SQL server database class to associate a SQL server installation and a database that is part of the installation.

8. A data structure as recited in claim 7, wherein the SQL objects class comprises at least one of the following properties:

a name property that defines a label by which an object in the objects class is known; and

a status property to indicate a current status of the object.

9. A data structure as recited in claim 7, wherein the SQL database class comprises at least one of the following properties:

a collation property to specify a column-level collation of a string datatype in the database;

a create-for-attach property to control database file creation;

a database status property to indicate a current operational status on the database;

a name property to define a label by which an object is known;

a primary file path property to return a path and name of an operating system directory containing a primary file for the database;

a size property to expose a total size of the database;

a space available property that returns an amount of disk resource allocated and unused; and

a status property to indicate a current status of the object.

10. A data structure as recited in claim 7, wherein the a SQL server class comprises at least one of the following properties:

a collation property to specify a column-level collation of a string datatype in the database;

a name property to define a label by which an object is known;

a status property to indicate a current status of the object; and

a user profile property to return a high-level role description for a login used by a current connection.

11. A data structure stored on one or more computer-readable media that is instantiated in accordance with a schema, the schema comprising:

a SQL error log class to represent at least one error log;

a SQL error log entry class to represent entries in the error log;

a SQL transaction log to represent a transaction log in the SQL database;

a SQL objects class to represent objects in the SQL database;

a SQL database class to represent instances of the SQL database;

a SQL server class to represent instances of a SQL Server;

a SQL error log error log entry class to represent an association between the error log and an entry in the error log;

a SQL server error log class to represent an association between a server installation and the error log used by the installation;

a SQL error log data file class to represent an association between the error log and an operating system file used to store the error log;

a SQL transaction log data file class to represent an association between the transaction log and an operating system file that is used to store the log;

a SQL database transaction log class to represent an association between the database and the transaction log for the database; and

a SQL server database class to associate a SQL server installation and a database that is part of the installation.

12. A data structure as recited in claim 11, wherein the SQL error log class comprises at least one of the following properties:

a last modified property to indicate a time and date that the error log was last modified; and

a status property to indicate a current status of an object in the database.

13. A data structure as recited in claim 11, wherein the SQL objects class comprises at least one of the following properties:

a name property that defines a label by which an object in the objects class is known; and

a status property to indicate a current status of the object.

14. A data structure as recited in claim 11, wherein the SQL database class comprises at least one of the following properties:

a collation property to specify a column-level collation of a string datatype in the database;

a create-for-attach property to control database file creation;

a database status property to indicate a current operational status on the database;

a name property to define a label by which an object is known;

a primary file path property to return a path and name of an operating system directory containing a primary file for the database;

a size property to expose a total size of the database;

a space available property that returns an amount of disk resource allocated and unused; and

a status property to indicate a current status of the object.

15. A data structure stored on one or more computer-readable media that is instantiated in accordance with a schema, the schema comprising:

a SQL objects class to represent objects in a database;

a SQL database class to represent instances of a SQL database;

a SQL table class to represent at least one table in the SQL database;

a SQL column class to represent a column in the table;

a SQL trigger class to represent a trigger to be executed when a specified data modification is attempted on the table;

a SQL user defined function class to represent a user defined function in the SQL database

a SQL stored procedure class to represent stored procedures defined in the SQL database;

a SQL view class to represent view tables in the SQL database;

a SQL stored procedure parameter class **814** to represent input and output parameters of a stored procedure;

a SQL table column class to represent an association between the table and the column contained in the table;

a SQL table trigger class to represent an association between the table and the trigger defined for the table;

a SQL database table class to represent an association between the database and the table contained in the SQL database;

a SQL database user defined function class to represent an association between the database and the user-defined function defined within the SQL database;

a SQL database view class to represent an association between the SQL database and the view;

a SQL database stored procedure class to represent an association between the database and a stored procedure defined within the database; and

a SQL stored procedure stored procedure parameter class **832** associates a stored procedure to a parameter used in the stored procedure.

16. A data structure as recited in claim 15, wherein the SQL objects class comprises at least one of the following properties:

a name property that defines a label by which an object in the objects class is known; and

a status property to indicate a current status of the object.

17. A data structure as recited in claim 15, wherein the SQL database class comprises at least one of the following properties:

a collation property to specify a column-level collation of a string datatype in the database;

a create-for-attach property to control database file creation;

a database status property to indicate a current operational status on the database;

a name property to define a label by which an object is known;

a primary file path property to return a path and name of an operating system directory containing a primary file for the database;

a size property to expose a total size of the database;

a space available property that returns an amount of disk resource allocated and unused; and

a status property to indicate a current status of the object.

18. A data structure as recited in claim 15, wherein the SQL table class comprises at least one of the following properties:

an attributes property to indicate various aspects of the table;

a database name property to indicate a name of the database of which the table is part;

a data space used property to report storage space used by rows of the table;

an index space used property that returns a quantity of disk resource used to store indexes;

a rows property that returns a number of rows in the table; and

a status property to indicate a current status of the table.

19. A data structure as recited in claim 15, wherein the SQL column class comprises at least one of the following properties:

a computed property that indicates whether the column is computed based on other values in the database;

a datatype property that indicates a datatype for the column;

an identity property that indicates whether the column is an identity column for the table;

a length property that indicates a maximum number of characters or bytes accepted by the column;

a table name property that indicates a name of the table that a key is defined in; and

a status property to indicate a current status of the table.

20. A data structure stored on one or more computer-readable media that is instantiated in accordance with a schema, the schema comprising:

a SQL objects class to represent objects in a database;

a SQL column class to represent a column in the table;

a SQL index class to represent an index for a table; and

a SQL index column class to represent an association between the index and the column.

21. A data structure as recited in claim 20, wherein the SQL column class comprises at least one of the following properties:

a computed property that indicates whether the column is computed based on other values in the database;

a datatype property that indicates a datatype for the column;

an identity property that indicates whether the column is an identity column for the table;

a length property that indicates a maximum number of characters or bytes accepted by the column;

a table name property that indicates a name of the table that a key is defined in; and

a status property to indicate a current status of the table.

22. A data structure as recited in claim 20, wherein the SQL index class comprises at least one of the following properties:

a no recompute property that controls statistics generation;

a space used property that returns a quantity of disk resource used to store data that implements the index;

a statistics index property to specify when the index maintains data distribution statistics;

a type property that specifies a type of the index; and

a status property to indicate a current status of the index.

23. A data structure stored on one or more computer-readable media that is instantiated in accordance with a schema, the schema comprising:

a SQL objects class to represent objects in a database;

a SQL constraint class to represent constraints defined in the SQL database;

a SQL table class to represent at least one table in the SQL database;

a SQL default class to represent attributes of a single SQL server default;

a SQL column class to represent a column in the table;

a SQL database class to represent instances of a SQL database;

a SQL datatype class to represent datatypes defined in a SQL server installation, the SQL datatype class having two subclasses that represent user data and system data;

a SQL check class to represent check attributes of a SQL server integrity constraint;

a SQL rule class to represent a single data-integrity rule;

a SQL table check class to represent an association between the table and the check attributes defined for the table;

a SQL user datatype default class to represent an association between a user-defined datatype and a rule bound to the column;

a SQL database default class to represent an association between the database and the defaults defined within the database;

a SQL column rule class to represent an association between the column and the rule bound to the column;

a SQL column datatype class to associate the column with its data type;

a SQL database rule class to represent an association between the database and the rules defined within the database;

a SQL user datatype rule class represents an association between the user defined datatype and the rule bound to the column;

a SQL database datatype class associates the database to the datatypes defined within the database;

a SQL base datatype class represents an association between the user-defined datatype and the system datatype; and

a SQL column default class to associate the column to the default for the column.

24. A data structure stored on one or more computer-readable media that is instantiated in accordance with a schema, the schema comprising:

a SQL objects class to represent objects in a database;

a SQL constraint class to represent constraints defined in the SQL database;

a SQL table class to represent at least one table in the SQL database;

a SQL column class to represent a column in the table;

a SQL database class to represent instances of a SQL database;

a SQL key class to represent keys defined for the table, the SQL key class having two subclasses that represent candidate keys in the table and foreign keys in the table, the candidate key subclass having two subclasses that represent unique keys and primary keys;

a SQL key column class to represent an association between a key and the column that is part of the key;

a SQL table key class to represent an association between the table and the key defined for the table;

a SQL referenced table class to represent an association between the foreign key and the table that contains the primary key referenced by the foreign key;

a SQL database candidate key class to represent an association between the database and the candidate key that is present in one of the tables in the database; and

a SQL referenced key class to represent an association between the foreign key and the candidate key that the foreign key references.

25. A data structure stored on one or more computer-readable media that is instantiated in accordance with a schema, the schema comprising:

a SQL setting class to represent settings that are used to configure an installation of the database;

a SQL objects class to represent objects in a SQL database;

a SQL database class to represent instances of the SQL database;

a SQL server class to represent instances of a SQL server;

a SQL integrated security setting class to represent integrated security settings;

a SQL DBMS user object class to represent objects related to user authentication;

a SQL user class to represent users of the SQL database;

a SQL login class to represent logins of the SQL database;

a SQL role class to represent groups of users with similar security attributes of the SQL database, the SQL role class having two subclasses to represent a SQL server security role not constrained to operation within a single database and to represent properties of a SQL server database role;

a SQL server integrated security setting class to represent an association between a SQL server installation and its security settings;

a SQL server login class to represent an association between the SQL server and a login defined within the SQL Server;

a SQL server server role class to represent an association between the SQL server and server roles defined within the SQL Server;

a SQL server user class to represent an association between the SQL server and a database user;

a SQL database login class to represent an association between a database and a login that is mapped to a user defined in the database;

a SQL database owner login class to represent an association between a database and the login mapped to the user that owns the database;

a SQL user login class to represent an association between a database user and the login used to authenticate the user;

a SQL member login class to represent an association between a SQL Server role and a login that is a member of the role;

a SQL login default database class to represent an association between a login and the default database for the login;

a SQL member user class to represent an association between a database role and a user that is a member of the role;

a SQL DBMS object owner class to represent an association between a SQL server database object and the user who owns the object;

a SQL database user class to represent an association between a database and a user defined for the database;

a SQL member database role class to associate two database roles; and

a SQL database database role class to associate a database role to the database within which the role is defined.

26. A data structure stored on one or more computer-readable media that is instantiated in accordance with a schema, the schema comprising:

a SQL objects class to represent objects in a SQL database;

a SQL database class to represent instances of the SQL database;

a SQL stored procedure class to represent stored procedures defined in the SQL database;

a SQL table class to represent at least one table in the SQL database;

a SQL user defined function class to represent a user defined function in the SQL database

a SQL view class to represent view tables in the SQL database;

a SQL DBMS user object class to represent objects related to user authentication;

a SQL user class to represent users of the SQL database;

a SQL role class to represent groups of users with similar security attributes of the SQL database, the SQL role class having two subclasses to represent a SQL server security role not constrained to operation within a single database and to represent properties of a SQL server database role;

a SQL database role database permission class to represent permissions that a database role has for the database in which it is defined;

a SQL user database permission class to represent permissions granted to a user for a database;

a SQL user stored procedure permission class to represent permissions granted to a user for a stored procedure;

a SQL database role stored procedure permission class to represent permissions that a database role has for a stored procedure;

a SQL user table permission class to represent permissions granted to a user for a table;

a SQL database role table permission to represent the permissions that a database role has for a table;

a SQL user user defined function permission to represent permissions granted to a user for a stored procedure;

a SQL database role user defined function permission to represent permissions that a database role has for a table; and

a SQL user view permission to represent permissions granted to a user for a view.

27. A data structure stored on one or more computer-readable media that is instantiated in accordance with a schema, the schema comprising:

a SQL database class to represent instances of the SQL database;

a SQL table class to represent at least one table in the SQL database;

a SQL index class to represent an index for a table; and

a SQL key class to represent keys defined for the table, the SQL key class having a subclass that represents candidate keys in the table;

a SQL file group class to represent attributes of a SQL server file group;

a SQL extension class to represent extensions made via associations to a managed system element;

a SQL extends class to associate a first class with a second class that extends the first class by defining new properties and methods;

a SQL file group database file class to associate a database file group to operating system files that are part of the group;

a SQL index file group class to represent an association between an index and a file group that stores the index;

a SQL key file group class to represent an association between a key and a file group used to store the key;

a SQL table file group class to represent an association between a table and file groups used to store the table;

a SQL table text file group class to associate a table with a file group; and

a SQL database file group class to represent an association between a database and a file group that contains operating system files that store data for the database.

28. An operating system comprising:

an object-oriented management service to monitor, configure, and control systems, services, and applications;

a database schema for a SQL (structured query language) database maintained by the object-oriented service, the database schema comprising:

database objects classes that represent components of a SQL database;

application system classes that represent settings, services, and transaction and error logs used by the SQL database;

security classes that represent users, roles, authentication login, and permissions for the users and the roles to use the SQL database; and

physical storage classes that represent physical files and file groups used by the database system to store data.

29. An operating system as recited in claim 28, wherein the database objects class comprises multiple classes representing tables, views, stored procedures, indexes, constraints, and keys.

30. An operating system as recited in claim 28, wherein the application system class comprises multiple classes representing settings, services, and transaction and error logs used by the database system.

31. An operating system as recited in claim 28, wherein the security class comprises multiple classes representing users, roles, authentication login, and permissions for the users and the roles.

32. An operating system as recited in claim 31, wherein permissions are modeled as association classes between database objects class and the users and roles.

33. An operating system as recited in claim 28, wherein the database objects classes comprises at least one of the following classes:

an objects class to represent objects in the SQL database;

a database class to represent instances of the SQL database;

a server class to represent instances of a SQL server;

a table class to represent at least one table in the SQL database;

a column class to represent a column in the table;

a key class to represent keys defined for the table;

a user defined function class to represent a user defined function in the SQL database;

a stored procedure class to represent stored procedures defined in the SQL database; and

a view class to represent view tables in the SQL database.

34. An operating system as recited in claim 28, wherein the application system classes comprises at least one of the following classes:

a setting class to represent settings that are used to configure an installation of the SQL database;

a database setting class to represent operational settings for the SQL database;

a configuration value class to represent configuration values;

a language setting class to expose properties of a language record;

a registry setting class to represent the installation and run-time parameters stored in the registry; and

a server connection setting class to represent default connection settings.

35. An operating system as recited in claim 28, wherein the security classes comprises at least one of the following classes:

an integrated security setting class to represent integrated security settings;

a user class to represent users of the SQL database;

a login class to represent logins of the SQL database; and

a role class to represent groups of users with similar security attributes of the SQL database.

36. An operating system as recited in claim 28, wherein the physical storage classes comprises at least one of the following classes:

a file group class to represent attributes of a SQL server file group;

a extension class to represent extensions made via associations to a managed system element;

a extends class to associate a first class with a second class that extends the first class by defining new properties and methods;

a file group database file class to associate a database file group to operating system files that are part of the group;

a index file group class to represent an association between an index and a file group that stores the index;

a key file group class to represent an association between a key and a file group used to store the key;

a table file group class to represent an association between a table and file groups used to store the table;

a table text file group class to associate a table with a file group; and

a database file group class to represent an association between a database and a file group that contains operating system files that store data for the database.

37. A method comprising:

creating a data structure in accordance with a schema, the schema defining at least one database objects class that represents components of a database system, at least one application system class that represents installation parameters of the database system, at least one security class that represents security features pertaining to use of the database system, and at least one physical storage class that represents physical files and file groups used by the database system to store data; and

populating the data structure.

38. A method as recited in claim 37, wherein the the database objects class comprises multiple classes representing tables, views, stored procedures, indexes, constraints, and keys.

39. A method as recited in claim 37, wherein the application system class comprises multiple classes representing settings, services, and transaction and error logs used by the database system.

40. A method as recited in claim 37, wherein the security class comprises multiple classes representing users, roles, authentication login, and permissions for the users and the roles.

41. An operating system as recited in claim **40**, further comprising defining permissions as association classes between database objects class and the users and roles.

42. A data structure embodied on one or more computer-readable media that is produced as a result of the method as recited in claim 37.

43. A system comprising:

means for instantiating a data structure in accordance with a schema, the schema defining at least one database objects class that represents components of a database system, at least one application system class that represents installation parameters of the database system, at least one security class that represents security features pertaining to use of the database system, and at least one physical storage class that represents physical files and file groups used by the database system to store data; and

means for populating the data structure.

* * * * *