CA DataMinder

Database Schema and Views Reference Guide

Release 14.6

CA Technologies Product References

This document references the following CA Technologies products:

- CA DataMinder™

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Documentation Changes

The following documentation updates have been made since the last release of this documentation:

- Wgn3RelatedEvent (see page 55)—Added table.
- WGN_V_RELATEDEVENT_1 (see page 135)—Added view.

Contents

Chapter 3: Database Views 125

Chapter 1: About Database Schema and Views

This guide contains the database table specifications used by the CA DataMinder product. It is provided to allow customers to develop additional customized database queries. To make the best use of this document, you must be familiar with relational database design and have a thorough understanding of SQL.

Schema Version

This guide is based on the database schema version 3.90 used by CA DataMinder 14.5 (build 3200 and its derivatives).

Schema Diagram

The relationships between all documented tables are shown in a separate schema diagram. You can open .html and .pdf versions of the diagram from the 'D', 'Reference', and 'Database Administrator' pages of the bookshelf.

dlp_database_schema_diagram.html

DLP_DBSchema_Diagram_ENU.pdf

Note: The diagram is also available in the \Bookshelf Files\DBSchema folder of the CA DataMinder bookshelf.

This section contains the following topics:

Datatypes

The database types described in this document are CA DataMinder datatypes. This is because CA DataMinder can utilize many different DBMSs and each of these has its own variation of SQL datatypes. See CA DataMinder Datatype Mappings (see page 12) for mappings between CA DataMinder datatypes and your own DBMS datatypes.

VARCHAR

Because each DBMS supported by CA DataMinder can vary the amount of physical storage needed to store VARCHAR data, the column size indicated for VARCHAR columns in this schema document refers to the number of characters, **not** the number of bytes. Note that for SQL Server, each character requires 2 bytes of storage, but for Oracle we recommend using the UTF-8 character set, which uses between one and three bytes to store an individual character.

IDM Columns

Various tables in the database schema contain column names that end with 'IDM', such as SequenceIDM or LoginIDM. Such columns provide an implicit relationship to the machine which generated that row of data.

For example, you can determine from the SequenceIDM column of the Wgn3Event table the machine on which that event was captured. To identify the machine, search the MachineID column in the WgnMachine table to identify the MachineID that matches the entry in the SequenceIDM column. To identify the machine that generated a row with a SequenceIDM column of 1234, use the following query:

```
SELECT MachineName FROM WgnMachine WHERE MachineID=1234
```

CA DataMinder Datatype Mappings

This section defines the SQL type mappings used by CA DataMinder to map *CA DataMinder SQL Types* to a DBMS specific datatype.

CA DataMinder Type	MS SQL Type	Oracle Type
IDENTITYDEF	Decimal(28,0)identity(1,1)	NUMERIC
IDENTITYREF	Decimal(28,0)identity(1,1)	NUMERIC
INTEGER	INT	NUMBER(10)
VARCHAR	VARCHAR	VARCHAR2
DECIMAL	DECIMAL	NUMBER
TIMESTAMP	DATETIME	DATE
BIT	BIT	NUMBER(1,0)
VARBINARY	VARBINARY	RAW
LONGVARCHAR	TEXT	VARCHAR2(4000)
LONGVARBINARY	IMAGE	LONG RAW

Chapter 2: Database Tables

This section contains the following topics:

Wgn3Address

This table contains all the strings that CA DataMinder has identified as addresses. Each address is assigned a unique ID, which is used by other tables that need to hold a reference to an address. The contents of this table are typically email addresses, although other types of address (including IM participant data) can also be stored in here. It is mandatory that all addresses are stored using lower case letters, although this is not enforced in the schema.

Note from schema version 3.50 events that are captured that do not have a natural address such as Web page will have an address automatically created that can be used to define the relationship between the user and the event. These addresses have an address name value of a specific format which is:-

/user_auto_addr=<x,y> where x and y are the unique identifier of the user.

These addresses are purely internally generated and can be recognized by having a value of 1 for address type in the WGN3UserAddressEx (see page 70) table.

Column	Primary Key	Datatype	Length	Notes
AddressUID1 Part-key used to uniquely identify an email address or the address of an IM participant.	Yes	DECIMAL	13	
AddressUID2 Part-key used to uniquely identify an email address or the address of an IM participant.	Yes	DECIMAL	13	

Column	Primary Key	Datatype	Length	Notes
AddressName The actual e-mail address or IM participant.		VARCHAR	255	

Wgn3BLOBDeletion

This table identifies all of the captured data blobs (Binary Large Objects) that need to be removed from the CMS.

Note: Only blobs files and Centera blobs are purged by CA DataMinder; 'remote' blobs such as EAS blobs are purged by the associated third-party application.

Column	Primary Key	Datatype	Length	Notes
BlobType (see page 43) Identifies the type of the blob, for example, a file-based blob.	Yes	INTEGER	4	Follow the hyperlink for details.
BlobLocation The location of the blob data to be deleted. The format of this string depends on the **BlobType**.	Yes	VARCHAR	255	

Wgn3Checkpoint

This table holds the definitions of replication Checkpoints created on the CMS. The table is only populated on the CMS. The CheckpointID (the CPID column) is used as an unenforced Foreign Key by the Wgn3CheckpointACK (see page 17) table.

Column	Primary Key	Datatype	Length	Notes
CpID Unique ID of the checkpoint.	Yes	IDENTITYDEF	8	

CpTimestamp Time at which the checkpoint was generated.	TIMESTAMP	8	
CpLevel Currently unused	INTEGER	4	Currently always 0
CpDescription Description supplied when the checkpoint was generated	LONGVARCHAR		

Wgn3CheckpointACK

This table contains checkpoint acknowledgements from all child machines of the CMS. It is used to verify that machines have processed checkpoints created by the CMS and can be used to analyze the time taken to propagate the checkpoint through the machine hierarchy.

Column	Primary Key	Datatype	Length	Notes
MachineIDM Part-key identifying the machine that generated this checkpoint acknowledgement.	Yes	INTEGER	4	FK -> WgnMachine
MachineID Part-key identifying the machine that generated this checkpoint acknowledgement.	Yes	INTEGER	4	FK -> WgnMachine
CpID Unique identifier of the checkpoint being acknowledged.	Yes	IDENTITYREF	8	Unenforced FK to the Wgn3Checkpoint table.
ACKTimestamp Time at which the acknowledgement was generated.		TIMESTAMP	8	

Wgn3ClassificationBase

This is a generic reference table, used in conjunction with the Wgn3ClassificationNode (see page 18) table, to define two trees of classification values that are referenced by the Wgn3EventParticipant table. It also contains Policy class trees that are referenced by Wgn3Trigger (see page 60) and Wgn3Resource (see page 114) tables.

Column	Primary Key	Datatype	Length	Notes
ClassUID (see page 19) Unique ID of the classification	Yes	INTEGER	4	Follow the hyperlink for details about Participant Class Node tables.
i18NID Encoded ID used to reference the string value from table WGN3Stringi18n.		VARCHAR	255	Encoding of classuid WCN[classuid]

Wgn3ClassificationNode

This table contains nodes in a hierarchy, where each node represents an unambiguous classification that can be applied to any entity that requires hierarchical classification. Each node references its immediate parent node (another entry in this table) and the type of the node (a reference to the Wgn3Classification table). This table is populated once at installation and thereafter is used for reference only.

In the CA DataMinder 3.50 schema, there are two separate classification trees defined by this table: a participant classification tree and an intervention classification tree. For details, see the Available class node trees (see page 19).

Column	Primary Key	Datatype	Length	Notes
NodeUID (see page 20) Unique ID of the Node.	Yes	INTEGER		Follow the hyperlink for details about Participant Class Node tables.
ParentNodeUID Reference to the parent node.		INTEGER		Null value indicates root node.

ClassUID (see page 19) Reference to classification of this node.	Yes	INTEGER	Follow the hyperlink for details about Participant Class Node tables.

Available Class Node Trees

The tree views on the following pages are schematic representations of the two classification trees defined by the Wgn3Classification and Wgn3ClassificationNode (see page 18) tables.

- **Participant classification tree:** See Participant class nodes (see page 20) for details. NodeUID values starting with **1** refer to Participant Classifications, referenced by the ParticipantNodeUID column of the Wgn3EventParticipant table.
- **Intervention classification tree:** See Intervention nodes (see page 22) for details. NodeUID values starting with **2** refer to Intervention Classifications, referenced by the InterventionNodeUID column of the Wgn3EventParticipant table.
- **Policy classification tree:** See Policy classification nodes (see page 25) for details. NodeUID values starting with **3** refer to Policy Classifications, referenced by the PolicyID column of the Wgn3Event (see page 32) table.

The structure of each tree is derived directly from the entries in the Wgn3ClassificationNode table. Each node references an entry in the Wgn3Classification table to define its type; for clarity, the type of each node (its 'class name' and class UID) is shown alongside each entry in the tree.

Notes: Be aware of the following:

- The NodeUID uniquely identifies each node in the tree. However, the node 'type' is not necessarily unique, so many nodes may refer to the same ClassUID in the Wgn3Classification table.
- In the Wgn3Classification and Wgn3ClassificationNode tables, all NodeUID and ClassUID values are decimal, not hexadecimal.

Participant Class Nodes

NodeUID values are decimal and start with **1**. They refer to participant classifications, referenced by the ParticipantNodeUID column of the Wgn3EventParticipant table.

NodeUID	Tree	ClassUID	ClassName
1000000	[participant]	1000000	WGNPN_ROOT
	|		
1001000	+[web]	1001001	WGNPN_WEB
1001001	| +[viewer]	1002001	WGNPN_VIEWER
	|		
1002000	+[email]	1001002	WGNPN_EMAIL
	| |		
1002001	| +[sender]	1002002	WGNPN_SENDER
1002002	| | +[from]	1003001	WGNPN_FROM
1002003	| | | +[internal]	1004001	WGNPN_INTERNAL
1002004	| | | +[external]	1004002	WGNPN_EXTERNAL
1002005	| | +[sent on behalf of]	1003002	WGNPN_SOB
1002006	| | | +[internal]	1004001	WGNPN_INTERNAL
1002007	| | | +[external]	1004002	WGNPN_EXTERNAL
1002027	| | +[reply_to]	1003009	WGNPN_REPLY_TO
1002028	| | +[internal]	1004001	WGNPN_INTERNAL
1002029	| | +[external]	1004002	WGNPN_EXTERNAL
1002008	| +[recipient]	1002003	WGNPN_RECIPIENT
1002009	| +[received_by]	1003003	WGNPN_REC_BY
1002010	| | +[internal]	1004001	WGNPN_INTERNAL
1002011	| | +[external]	1004002	WGNPN_EXTERNAL
1002012	| +[to]	1003004	WGNPN_TO
1002013	| | +[internal]	1004001	WGNPN_INTERNAL
1002014	| | +[external]	1004002	WGNPN_EXTERNAL
1002015	| +[cc]	1003005	WGNPN_CC
1002016	| | +[internal]	1004001	WGNPN_INTERNAL
1002017	| | +[external]	1004002	WGNPN_EXTERNAL
1002018	| +[bcc]	1003006	WGNPN_BCC
1002019	| | +[internal]	1004001	WGNPN_INTERNAL
1002020	| | +[external]	1004002	WGNPN_EXTERNAL
1002021	| +[received on behalf of]	1003007	WGNPN_ROB
1002022	| | +[internal]	1004001	WGNPN_INTERNAL
1002023	| | +[external]	1004002	WGNPN_EXTERNAL
1002024	| +[transport only recipient]	1003008	WGNPN_TRANSPORT
1002025	| +[internal]	1004001	WGNPN_INTERNAL
1002026	| +[external]	1004002	WGNPN_EXTERNAL
	|		
1003000	+[appmon]	1001003	WGNPN_APPMON
1003001	| +[viewer]	1002001	WGNPN_VIEWER
	|		
1004000	+[IM]	1001004	WGNPN_IM
	| |		
1004001	| +[participant]	1002004	WGNPN_PARTICIPANT

1004002	| | +[internal]	1004001 WGNPN_INTERNAL
1004003	| | +[external]	1004002 WGNPN_EXTERNAL
	| |	
1004004	| +[sender]	1002002 WGNPN_SENDER
1004005	| | +[internal]	1004001 WGNPN_INTERNAL
1004006	| | +[external]	1004002 WGNPN_EXTERNAL
	| |	
1004007	| +[recipient]	1002003 WGNPN_RECIPIENT
1004008	| +[internal]	1004001 WGNPN_INTERNAL
1004009	| +[external]	1004002 WGNPN_EXTERNAL
	|	
	|	
1005000	+[file]	1001005 WGNPN_FILE
1005001	+[policy_user]	1005001 WGNPN_POLICY
1005002	+[owner]	1005002 WGNPN_OWNER
1005003	+[creator]	1005003 WGNPN_CREATOR
1005004	+[associated]	1005004 WGNPN_ASSOCIATED
1005005	+[host]	1005005 WGNPN_HOST
1005006	+[src]	1005006 WGNPN_SRC
1005007	+[dest]	1005007 WGNPN_DEST

Intervention Nodes

NodeUID values are decimal and start with **2**. They refer to Intervention Classifications, referenced by the InterventionNodeUID column of the Wgn3EventParticipant table.

NodeUID	Tree	ClassUID	ClassName
2000000	[intervention]	2000000	WGNIN_ROOT
	|		
2001000	+[web]	2001001	WGNIN_WEB
	| |		
2001001	| +[blocked on viewing]	2002001	WGNIN_VIEW_BLOCK
2001002	| | +[with notification]	2004001	WGNIN_NOTIFY
2001003	| | +[heeded warning]	2004003	WGNIN_HEED_WARN
2001004	| | +[heeded inform]	2004005	WGNIN_HEED_INFORM
2001005	| | +[silently]	2004007	WGNIN_SILENT
	| |		
2001006	| +[viewed]	2002002	WGNIN_VIEW
2001007	| +[ignored warning]	2004002	WGNIN_IGNORE_WARN
2001008	| | +[auto]	2005001	WGNIN_AUTO
2001009	| +[ignored informational]	2004004	WGNIN_IGNORE_INFORM
2001010	| | +[auto]	2005001	WGNIN_AUTO
2001011	| +[personal]	2004006	WGNIN_PERSONAL
	|		
2002000	+[email]	2001002	WGNIN_EMAIL
	| |		
2002001	| +[blocked on send]	2003002	WGNIN_SEND_BLOCK
2002002	| | +[with notification]	2004001	WGNIN_NOTIFY
2002003	| | +[heeded warning]	2004003	WGNIN_HEED_WARN
2002004	| | +[heeded inform]	2004005	WGNIN_HEED_INFORM
2002005	| | +[silently]	2004007	WGNIN_SILENT
	| |		
2002006	| +[send]	2003003	WGNIN_SEND
2002007	| | +[ignored warning]	2004002	WGNIN_IGNORE_WARN
2002019	| | | +[quarantine]	2006001	WGNIN_QUARANTINE
2002020	| | | +[silent]	2004007	WGNIN_SILENT
2002021	| | | +[with notification]	2004001	WGNIN_NOTIFY
2002008	| | +[ignored informational]	2004004	WGNIN_IGNORE_INFORM
2002022	| | | +[quarantine]	2006001	WGNIN_QUARANTINE
2002023	| | | +[silent]	2004007	WGNIN_SILENT
2002024	| | | +[with notification]	2004001	WGNIN_NOTIFY
2002025	| | +[quarantine]	2006001	WGNIN_QUARANTINE
2002026	| | | +[silent]	2004007	WGNIN_SILENT
2002027	| | | +[with notification]	2004001	WGNIN_NOTIFY
2002009	| | +[personal]	2004006	WGNIN_PERSONAL
	| |		
2002010	| +[blocked on receipt]	2003004	WGNIN_RECEIVE_BLOCK
2002011	| | +[with notification]	2004001	WGNIN_NOTIFY
2002012	| | +[heeded warning]	2004003	WGNIN_HEED_WARN
2002013	| | +[heeded inform]	2004005	WGNIN_HEED_INFORM

```
2002014    |  |  +[silently]                          2004007       WGNIN_SILENT
           |  |
2002028    |  +[blocked on receipt: legal hold]       2003006       WGNIN_RECEIVE_BLOCK_LH
2002029    |  |  +[accept]                            2004008       WGNIN_ACCEPT_LH
2002030    |  |  +[reject]                            2004009       WGNIN_REJECT_LH
2002015    |
2002015    |  +[receive]                              2003005       WGNIN_RECEIVE
2002016    |     +[ignored warning]                   2004002       WGNIN_IGNORE_WARN
2002017    |     +[ignored informational]             2004004       WGNIN_IGNORE_INFORM
2002018    |     +[personal]                          2004006       WGNIN_PERSONAL
           |
           |
2003000    +[appmon]                                 2001003       WGNIN_APPMON
           |  |
2003001    |  +[blocked on viewing]                   2002001       WGNIN_VIEW_BLOCK
2003002    |  |  +[with notification]                 2004001       WGNIN_NOTIFY
2003003    |  |  +[heeded warning]                    2004003       WGNIN_HEED_WARN
2003004    |  |  +[heeded inform]                     2004005       WGNIN_HEED_INFORM
2003005    |  |  +[silently]                          2004007       WGNIN_SILENT
           |  |
2003006    |  +[viewed]                               2002002       WGNIN_VIEW
2003007    |     +[ignored warning]                   2004002       WGNIN_IGNORE_WARN
2003008    |     |   +[auto]                          2005001       WGNIN_AUTO
2003009    |     +[ignored informational]             2004004       WGNIN_IGNORE_INFORM
2003010    |     |   +[auto]                          2005001       WGNIN_AUTO
2003011    |     +[personal]                          2004006       WGNIN_PERSONAL
           |
2004000    +[IM]                                     2001004       WGNIN_IM
           |  |
2004001    |  +[participated]                         2003001       WGNIN_PARTICIPATE
           |  |
2004002    |  +[blocked on send]                      2003002       WGNIN_SEND_BLOCK
2004003    |  |  +[with notification]                 2004001       WGNIN_NOTIFY
2004004    |  |  +[heeded warning]                    2004003       WGNIN_HEED_WARN
2004005    |  |  +[heeded inform]                     2004005       WGNIN_HEED_INFORM
2004006    |  |  +[silently]                          2004007       WGNIN_SILENT
           |  |
2004007    |  +[send]                                 2003003       WGNIN_SEND
2004008    |  |  +[ignored warning]                   2004002       WGNIN_IGNORE_WARN
2004009    |  |  +[ignored informational]             2004004       WGNIN_IGNORE_INFORM
2004010    |  |  +[personal]                          2004006       WGNIN_PERSONAL
           |  |
2004011    |  +[blocked on receipt]                   2003004       WGNIN_RECEIVE_BLOCK
2004012    |  |  +[with notification]                 2004001       WGNIN_NOTIFY
2004013    |  |  +[heeded warning]                    2004003       WGNIN_HEED_WARN
2004014    |  |  +[heeded inform]                     2004005       WGNIN_HEED_INFORM
2004015    |  |  +[silently]                          2004007       WGNIN_SILENT
           |  |
2004016    |  +[received]                             2003005       WGNIN_RECEIVE
```

```
          |
2005000   +[file]                                2001005      WGNIN_FILE
             |
2005001      +[deleted]                          2007001      WGNIN_DELETE
2005002      |  +[secure]                        2007004      WGNIN_SECURE
2005003      +[replaced]                         2007002      WGNIN_REPLACE
2005004      |  +[secure]                        2007004      WGNIN_SECURE
2005005      +[associated]                       2007003      WGNIN_ASSOCIATE
2005006      +[blocked in motion]                2008001      WGNIN_IN_MOTION_BLOCK
2005007      |  +[with notification]             2004001      WGNIN_NOTIFY
2005008      |  +[heeded warning]                2004003      WGNIN_HEED_WARN
2005009      |  +[heeded inform]                 2004005      WGNIN_HEED_INFORM
2005010      +[moved]                            2008002      WGNIN_IN_MOTION_MOVED
2005011         +[ignored warning]               2004002      WGNIN_IGNORE_WARN
2005012         |  +[auto]                       2005001      WGNIN_AUTO
2005013         +[ignored informational]         2004004      WGNIN_IGNORE_INFORM
2005014            +[auto]                       2005001      WGNIN_AUTO
```

Policy Classification Nodes

These values identify the type of policy that a trigger implements, within a classification hierarchy. These values are referenced by the PolicyID column of the Wgn3Trigger (see page 60) table. NodeUID values are decimal; values for predefined policies start with **3**; values for custom policies start with **4**.

```
NodeUID    Tree                                                    ClassUID
3000000  [Pre-defined]                                             3000000
          |
3010000   +[Data Loss Prevention]                                  3010000
3010100   |  +[Customer/Supplier Treatment]                        3010100
3010101   |  |  +[Customer Complaints Response]                    3010101
3010102   |  |  +[Gifts and Entertainment]                         3010102
3010103   |  |  +[Guarantees and Assurances]                       3010103
          |  |
3010200   |  +[Employee Behavior]                                  3010200
3010201   |  |  +[Coercive Behavior and Intimidation]              3010201
3010202   |  |  +[Communication with Competitors]                  3010202
3010203   |  |  +[Communication with the Press/News Organizations] 3010203
3010204   |  |  +[Fantasy League]                                  3010204
3010205   |  |  +[Foreign Language]                                3010205
3010206   |  |  +[Gambling Prohibition]                            3010206
3010207   |  |  +[Harassment]                                      3010207
3010208   |  |  +[Inappropriate, Offensive and Sexual Language]    3010208
3010209   |  |  +[Intent to Resign]                                3010209
3010210   |  |  +[Resumes]                                         3010210
3010211   |  |  +[Discrimination and Racism]                       3010211
3010212   |  |  +[Office Relationships: Romantic]                  3010212
3010213   |  |  +[Jokes]                                           3010213
3010214   |  |  +[Termination/Layoff Discussions]                  3010214
          |  |
3010300   |  +[Personally Identifiable Information]                3010300
3010301   |  |  +[Account Number]                                  3010301
3010302   |  |  +[Account Number - Threshold]                      3010302
3010303   |  |  +[Account Number with Additional PII]              3010303
3010304   |  |  +[Account Number and Routing Information]          3010304
3010305   |  |  +[Background Checks]                               3010305
3010306   |  |  +[Credit Card Information]                         3010306
3010307   |  |  +[Credit Card Information - Threshold]             3010307
3010308   |  |  +[Credit Report]                                   3010308
3010310   |  |  +[Employee Compensation Information]               3010310
3010311   |  |  +[Employee Evaluation Information]                 3010311
3010312   |  |  +[Social Security Number]                          3010312
3010313   |  |  +[Social Security Number - Threshold]              3010313
3010314   |  |  +[Social Security Number with Additional PII]      3010314
3010315   |  |  +[UK Drivers License]                              3010315
3010316   |  |  +[UK Drivers License - Threshold]                  3010316
3010317   |  |  +[UK National Insurance Number]                    3010317
3010318   |  |  +[UK National Insurance Number - Threshold]        3010318
```

```
3010319   |  |  +[UK National Insurance Number with Additional PII]  3010319
3010320   |  |  +[UK Tax Identification Number]                      3010320
3010321   |  |  +[UK Tax Identification Number - Threshold]          3010321
3010322   |  |  +[US Drivers License]                                3010322
3010323   |  |  +[US Drivers License - Threshold]                    3010323
3010324   |  |  +[US Individual Taxpayer Identification Number (ITIN)]      3010324
3010325   |  |  +[US Individual Taxpayer Ident. Number (ITIN) - Threshold]  3010325
3010326   |  |  +[US Passport Number]                                3010326
3010327   |  |  +[US Passport Number - Threshold]                    3010327
3010328   |  |  +[Unencrypted Wire Transfer Information]             3010328
3010329   |  |  +[Canadian Social Insurance Number]                  3010329
3010330   |  |  +[Canadian Social Insurance Number - Threshold]      3010330
3010331   |  |  +[Canadian Social Insurance Number with Additional PII]     3010331
          |  |
3010400   |  +[Personal Health Information]                          3010400
3010401   |  |  +[Benefits Enrollment Information]                   3010401
3010402   |  |  +[Diagnoses Information]                             3010402
3010403   |  |  +[Individually Identifiable Health Information (IIHI)]       3010403
3010404   |  |  +[Medical Billings and Claims]                       3010404
3010405   |  |  +[Medical History]                                   3010405
3010406   |  |  +[Medical Record Numbers]                            3010406
3010407   |  |  +[Medical Record Numbers - Threshold]                3010407
          |  |
3010500   |  +[Intellectual Property]                               3010500
3010501   |  |  +[Confidential Trade Data]                           3010501
3010502   |  |  +[Proprietary Software Code]                         3010502
3010503   |  |  +[Technical Specifications or Designs]               3010503
3010504   |  |  +[Patent Applications]                               3010504
3010505   |  |  +[Product and Design Specifications]                 3010505
          |  |
3010600   |  +[Non-Public Information]                              3010600
3010601   |  |  +[Board Minutes and Discussions]                     3010601
3010602   |  |  +[Customer Lists]                                    3010602
3010603   |  |  +[Financial Information - Balance Sheet]             3010603
3010604   |  |  +[Financial Information - Income Statement]          3010604
3010605   |  |  +[Financial Information - Projections]               3010605
3010606   |  |  +[Information Security Label Control]                3010606
3010607   |  |  +[Internal IT Support Documents]                     3010607
3010608   |  |  +[Licensing Agreements]                              3010608
3010609   |  |  +[Pricing List]                                      3010609
3010610   |  |  +[Mergers and Acquisitions]                          3010610
3010612   |  |  +[Project Information]                               3010612
3010613   |  |  +[Restricted List]                                   3010613
3010614   |  |  +[Sales Information]                                 3010614
3010615   |  |  +[Draft Documentation]                               3010615
3010616   |  |  +[Inside Information: Rumors and Secrets]            3010616
3010617   |  |  +[Inside Information: Non-public Financial Information Loss]  3010617
3010618   |  |  +[Inside Information: Non-public Company Information Loss]    3010618
          |  |
```

```
3010700   |  +[Security General]                                          3010700
3010701   |  |  +[Audio Files]                                           3010701
3010703   |  |  +[Forwarding Senior Management Email or Documents]       3010703
3010704   |  |  +[Graphic and Image Files]                               3010704
3010705   |  |  +[Large Message or File Size]                            3010705
3010706   |  |  +[Password Protection/Encryption: Prohibition]           3010706
3010707   |  |  +[Sharing of Usernames and Passwords]                    3010707
3010708   |  |  +[Video Files]                                           3010708
3010710   |  |  +[Network Security Threats]                              3010710
3010714   |  |  +[Restricted Websites]                                   3010714
3010715   |  |  +[Transfer of Personal Email File Folders]               3010715
3010716   |  |  +[Transfer of Attachments - Threshold]                   3010716
3010717   |  |  +[Suspicious Email Behavior]                             3010717
3010718   |  |  +[Email to Personal Addresses]                           3010718
          |  |
3010800   |  +[Web Communication Control]                                3010800
3010801   |     +[Blogging/Messaging Sites]                              3010801
3010802   |     +[Wiki Posting Control]                                  3010802
3010803   |     +[Social Networking]                                     3010803
          |
          |
3020000   +[Compliance]                                                 3020000
3020100   |   +[Corporate and Regulatory Compliance]                     3020100
3020101   |     +[Anti-Money Laundering - OFAC]                          3020101
3020102   |     +[Solicitations: Political]                              3020102
3020103   |     +[Solicitations: Religious]                              3020103
3020104   |     +[Solicitations: Charitable]                             3020104
3020105   |     +[Bribes/Kickbacks/Quid Pro Quos/Blackmail]              3020105
          |
3030000   +[Legal]                                                      3030000
3030100      +[Attorney Communications]                                  3030100
3030101      |  +[Attorney Client Privilege]                             3030101
             |
3030200      +[Litigation Risk]                                          3030200
3030201         +[Threats of Litigation]                                 3030201
3030202         +[Discussion of Legal Proceedings]                       3030202
3030203         +[Potential Ethical Issues]                              3030203
3030204         +[Potential Legal Issues]                                3030204

4000000 [Custom]                                                         4000000
```

Wgn3Diagnostic

This table holds machine diagnostics collected for each machine in the hierarchy. Each machine holds the diagnostics for all child machines within its own hierarchy.

Column	Primary Key	Datatype	Length	Notes
MachineIDM Part-key identifying the machine to which the diagnostic belongs.	Yes	INTEGER	4	FK -> WgnMachine
MachineID Part-key identifying the machine to which the diagnostic belongs.	Yes	INTEGER	4	FK -> WgnMachine
DiagID (see page 29) Identifies the type of diagnostic stored by this row.	Yes	INTEGER	4	Follow the hyperlink for details.
DiagValueType (see page 30) Identifies the type of data that has been stored.		INTEGER	4	Follow the hyperlink for details.
DiagValueStr For string type diagnostics, this column holds the string value.		VARCHAR	255	
DiagValueLong For numeric type diagnostics, this column holds the numeric value.		DECIMAL(28,0)	16	
DiagValueTimestamp For timestamp type diagnostics, this column holds the timestamp value.		TIMESTAMP	8	
DiagTimestamp This column stores the time that the diagnostic value was last updated.		TIMESTAMP	8	

DiagID

This column identifies the individual diagnostic data that has been collected.

Notes		
0	NUMERIC	**Infrastructure State** Indicates the state of machines infrastructure when the diagnostic was collected. This value can be updated dynamically (such as, if a machine is suspended) and not just when diagnostics are collected by the parent. **Note:** The individual state values are defined in the WgnWellKnownString table.
1	STRING	**Infrastructure Error**. The value of this diagnostic is only relevant if the 'Infrastructure State' is 3 (FAILED) or 4 (SUSPENDED). This value contains the error code and description associated with the current state.
2	TIMESTAMP	**Last Replication Time**. This indicates the time which the child machine last replicated infrastructure data down from its parent.
3	NUMERIC	**Outstanding Infrastructure Data Count**. This is the count of data objects (such as, policies, BLOBs, etc.) waiting to be replicated down to this machine from its parent.
4	NUMERIC	**Outstanding Application Data Count**. This is a count of captured data objects waiting to be replicated up from this machine to its parent.
5	NUMERIC	**Replication Subscriber State**. This indicates the current replication state of the machine. Please see SubscriberState (see page 88). **Note:** The individual state values are defined in the WgnWellKnownString table
6	NUMERIC	**Replication Synchronization Stage**. Indicates the current replication synchronization stage when the diagnostic was collected. **Note:** The individual state values are defined in the WgnWellKnownString table

DiagValueType

The DiagValueType column defines the type of data represented by the diagnostic, and is used to identify the column that actual data value is stored inside.

Notes	
0	**STRING**. The diagnostic data value is stored in the 'DiagValueStr' column.
1	**NUMERIC**. The diagnostic data value is stored in the 'DiagValueLong' column.
2	**TIMESTAMP**. The diagnostic data value is stored in the 'DiagValueTimestamp' column.

Wgn3EA

This table defines additional attributes of an event. Note that most event types do not require any definitions in this table.

Column	Primary Key	Datatype	Length	Notes
EventUID Key used to uniquely identify a captured or imported event.	Yes	IDENTITYREF	13	Foreign key: *Wgn3Event*
AttrIndex Part-key used to identify this event attribute.	Yes	INTEGER	4	Possible values: 0 upwards: Normal event attribute -1 Primary event ID -2 Transaction event -10000 downwards: Site attributes See AttrType for more details
EventTimeStamp The time at which the event occurred.	Yes	TIMESTAMP	8	Foreign key: *Wgn3Event*
AttrType (see page 31) The type of the attribute.		VARCHAR	255	Follow the hyperlink for details.
AttrValue The attribute value.		VARCHAR	255	

AttrType

AttrType denotes the type of value found in the AttrValue field. Some attributes apply to an entire sequence rather than a single event. These attributes are always attached to the first event in the sequence (EventIndex = 0), and are prefixed with "sq.". Possible values are:

Notes	
"wb.uf"	Name of file uploaded from web event. There may be multiple instances of this attribute per event.
"wb.ru"	URL that web event was redirected to.
"em.af"	Name of file attached to an email. There may be multiple instances of this attribute per event.
"em.ra"	Address that email was redirected to.
"em.qt"	Quarantine timeout. The presence of this value indicates that an event has been quarantined. Its value will always be zero, indicating that the default timeout applicable for the event will be applied.
"em.mc"	The 'message classification' of an email.
"am.kp"	Number of key presses captured in application monitoring event.
"am.mc"	Number of mouse clicks captured in application monitoring event.
"am.at"	Time in milliseconds that user was active in application monitoring event.
"am.dr"	Deactivation reason of application monitoring event. Possible values are 0 (ongoing event was updated), 1 (switched to another top level window), 2 (closed the top level window), 3 (top level window title changed) or 4 (application was shutdown).
"im.af"	Name of file attached to Instant Messaging chapter. There may be multiple instances of this attribute per event.
"im.nw"	Name of IM Network over which conversation is transported.
"fi.rl"	Location that file was copied to
"cl.fc"	Event classification
"tr.fc"	Trigger classification
"Extracted.<*name*>"	A value extracted from a web page or email by a classifier will start with "Extracted." followed by the name of the extracted value.
"SmartTag.<*name*>"	A SmartTag™ identified on a web page or email by a trigger will start with "SmartTag." followed by the name of the tag.
"sq.prim_ev"	Sequence attribute denoting the primary event (ie, the first event in the sequence with a capture module trigger). The value of this attribute takes the form "<*SequenceIDM*>.<*SequenceID*>:<*EventIndex*>" eg, "100.1:2". This attribute will always have an AttrIndex = -1.

Notes	
"sq.trans_ev"	Sequence attribute denoting the transaction event. This event is the last event in the sequence which has a transaction module trigger, and the one to which the WgnTransaction record points. The value of this attribute uses the same syntax as the "sq.prim_ev" attribute. This attribute will always have an AttrIndex = -2.

Wgn3Event

This table contains all captured events for all users. An event is a single object (eg, web page, email) captured from a monitored software package.

Column	Primary Key	Datatype	Length	Notes
EventUID Key used to uniquely identify a captured or imported event.	Yes	IDENTITYDEF	13	
EventTimestamp The time at which the event occurred.	Yes	TIMESTAMP	8	
SequenceIDM Part-key used to identify the owning sequence and also uniquely identify the event.		INTEGER	4	
SequenceID Part-key used to identify the owning sequence and also uniquely identify the event.		INTEGER	4	
EventIndex Part-key used to uniquely identify the event.		INTEGER	4	The index identifies the event order within its owning sequence.
EventMajorType (see page 35) Identifies the 'major' event type (eg, email or web page)		INTEGER	4	Follow the hyperlink for details.
EventMinorType (see page 36) Identifies the 'minor' event type (eg, email sent)		INTEGER	4	Follow the hyperlink for details.

Column	Primary Key	Datatype	Length	Notes
EventSubType (see page 37) Identifies the 'direction' of the event (eg, incoming or outgoing)		INTEGER	4	Follow the hyperlink for details.
AesMajorType (see page 39) Part-key that identifies the source object (software package) that captured the sequence.		INTEGER	4	Follow the hyperlink for details
AesMinorType (see page 39) Part-key that identifies the source object (software package) that captured the sequence.		INTEGER	4	Follow the hyperlink for details
AesSubType (see page 41) Part-key that identifies the source object (software package) that captured the sequence.		INTEGER	4	Follow the hyperlink for details
EventText1 (see page 37) Event specific string identifying the event.		VARCHAR	255	Follow the hyperlink for details.
EventText2 (see page 38) Event specific string identifying the event.		LONGVARCHAR		Follow the hyperlink for details.
EventAttributes (see page 41) Attributes of the event.		VARCHAR	255	Follow the hyperlink for details.
ExternalID This attribute contains an external (non-CA DataMinder) unique ID of an event to enable matching with duplicates of the same event captured elsewhere.		VARCHAR	255	
Duration The length of time for which the event occurred.		INTEGER	4	Set to 0 if the event has no duration.
ExpiryTimestamp Stores the time at which the event expires. Expired events are removed by the next scheduled database purge.		TIMESTAMP	8	

Column	Primary Key	Datatype	Length	Notes
IsPermanent If this column has a non-zero (true) value, then the event will never be deleted by a scheduled database purge.		BIT	1	
UpdateTimestamp The time that event was last written to the DBMS.		TIMESTAMP	8	This will be different on the CMS and client.
PurgeState Indicates whether the row can be purged if 'Purge On Replicate' is enabled. This column is ignored on a CMS.		INTEGER	4	Possible values: 0 - Cannot be purged 1 - Can be purged
GMTOffset Specifies the difference (in minutes) between UTC time and the time zone in which an event's source machine resides.		INTEGER	4	The source machine is an import machine or the machine on which an event was captured. Captured or imported events are time stamped in UTC time.
DSTOffset Specifies the length (in minutes) of the daylight saving offset for the source machine's timezone.		INTEGER	4	For source machines in timezones where DST is used, the DSTOffset is always set to 60 irrespectve of whether the event was captured in summer or winter.
BlobType (see page 43) Identifies the type of the blob (such as a file based blob).		INTEGER	4	Follow the hyperlink for details. NULL if the event has no BLOB.
BlobLocation The location of the blob data. The format of this string depends on the **BlobType**.		VARCHAR	255	NULL if the event has no BLOB.
BlobSize The size of the blob (in bytes) before encryption/compression.		DECIMAL	13	NULL if the event has no BLOB.

Column	Primary Key	Datatype	Length	Notes
BLOBPhysicalSize The size of the blob (in bytes) after encryption/compression.		DECIMAL	13	NULL if the event has no BLOB.
QueryFlags Denormalized column indicating if the event has associated Trigger and/or Audit Records		INTEGER	4	Possible values: 0 = None 1 = Trigger 2 = Issue 3 = Issue+Trigger 4 = Audit 5 = Audit+Trigger 6 = Audit+Issue 7 = Audit+Issue+Trigger

EventMajorType

EventMajorType identifies the basic type of the event. The value is closely aligned to the AesMajorType (see page 39). Possible values are:

Notes	
1	Web browser generated events.
2	Email client generated events.
3	CA DataMinder Application Monitoring events.
4	Instant Messaging events.
5	File events.
127	Events generated by CA DataMinder from extracted data.

EventMinorType

EventMinorType is used to qualify the specific type of event. The interpretation of this value depends on the EventMajorType.

Notes	
For an EventMajorType of '1', the possible values of EventMinorType are:	
6	Web page.
64	A file from the Content Classification Service (CCS).
For an EventMajorType of '2', the possible values of EventMinorType are:	
2	An email that the user has read (client integration).
3	An email captured manually by the user (client integration).
4	An email that was captured in transit (server integration).
16	An email that has been sent (client integration).
17	An email that has been received but was not read (i.e. blocked - client integration).
18	An email that was canceled after being sent (not transmitted - client integration).
32	An email that contains an embedded IM conversation
48	An email that contains an embedded FAX (usually as an image file attachment)
64	An email from a Bloomberg terminal
For an EventMajorType of '3', the possible values of EventMinorType are:	
1	An active application updated its monitored data.
16	The active application was de-activated.
For an EventMajorType of '4', the possible values of EventMinorType are:	
1	A chapter of an IM conversation.
For an EventMajorType of '5', the possible values of EventMinorType are:	
1	A file at rest on a local or network volume.
2	A file at rest in a Microsoft Exchange public folder
3	A file at rest in a Microsoft Sharepoint folder
4	Data at rest in a database
16	A file moving on the network containing web data (captured by the Network Boundary Agent or ICAP agent)
32	A file moving to a removable storage device
48	A file moving to a printer

EventSubType

The interpretation of the EventSubType field depends on the major type of the event.

Notes	
For Web and email events (EventMajorType 1 and 2) the EventSubType describes the direction of the event. Possible values are:	
0	The direction is not known or is not applicable.
1	The event is incoming (eg, web page or incoming email).
2	The event is outgoing (eg, submitted form data or outgoing email).
17	The event is via the webmail channel.
For email and file events (EventMajorType 1 and 5) captured at the network boundary the EventSubType describes the channel of communication:	
17	The event is via the webmail channel.
18	The event is via the web channel.
19	The event is via the FTP channel.
20	The event is via the NNTP channel
21	The event is via the IM channel
For Transaction events (EventMajorType 127), possible values of EventSubType are:	
1	The event is a "full" transaction and contains the merged results of one or more partial transactions
2	The event is a "partial" transaction and contains the transaction data corresponding to a single captured sequence.

EventText1

EventText1 is interpreted differently depending on the EventMajorType. The list below show the interpretation for each possible EventMajorType:

Notes	
EventMajorType=1	Web URL.
EventMajorType=2	Summary of sender/recipients of the mail (maximum 5 recipients). i.e. sender(s) **|** recipient(s)
EventMajorType=3	File system path to the application executable.

Notes	
EventMajorType=4	Summary of participants in IM conversation (maximum 5 active and 5 passive participants)
EventMajorType=5	The original location or the source of the file. The details of this field depend on the EventMinorType as follows … Host machine or IP address **|** file path Microsoft Exchange server name **|** Exchange public folder Microsoft Sharepoint server name **|** document path Data source name (from connection string) **|** SQL Query of source data source machine or IP addr **|** target machine or IP addr Host machine **|** file path print server name **|** printer location **** printer name
	Not Used.

Note: the | and \ characters are separators that exist in the database to separate the sub –elements of the field.

EventText2

EventText2 is interpreted differently depending on the EventMajorType. The list below show the interpretation for each possible EventMajorType:

Notes	
EventMajorType=1	Title of the web page (extracted from the HTML) if available.
EventMajorType=2	Subject line of the email address.
EventMajorType=3	Title of the top-level window of the application.
EventMajorType=4	Title of the conversation if available.
EventMajorType=5	The name of the file. The details of this field depend on the EventMinorType as follows : filename Microsoft Exchange public folder item name. *This could* be *a filename though other types (such as, message) are supported.* filename Subject – for example '*row 1 – 100*' protocol **:** filename Host machine **|** file path document title (if available)

Notes	
EventMajorType=127	Not used.

AesMajorType

AesMajorType identifies the type of client that captured the sequence. Possible values are:

Notes	
1	Web browser.
2	Email client.
3	CA DataMinder's Application Monitoring client.
4	Instant Messaging client.
5	File client

AesMinorType

AesMinorType is used to qualify the specific type of client that captured the sequence. The interpretation of this value depends on the AesMajorType.

For an AesMajorType of '1', the possible values of AesMinorType are:	
1	Internet Explorer (all versions) and Windows Explorer
2	All embedded web browsers (eg, within Outlook).
3	Generic browser client
For an AesMajorType of '2', the possible values of AesMinorType are:	
1	Microsoft Outlook
2	Lotus Notes
17	Microsoft Exchange Server (Live Mail)
18	Microsoft Exchange Server (Mailbox Import)
19	Domino Server (Lotus Notes)
33	Email File Importer
34	Symantec Enterprise Vault
49	Bloomberg Server (Live Mail)

65	External Agent API (email)
81	Network Boundary Agent (Email)
97	MTA Milter Agent
113	IIS SMTP Agent
For an AesMajorType of '3', the possible values of AesMinorType are:	
1	Application Monitoring client type 1 (basic)
For an AesMajorType of '4', the possible values of AesMinorType are:	
17	Instant Bloomberg Server
18	Mind Align Server
19	Bloomberg Server (IM)
20	Actiance Server (formerly Facetime)
21	IMlogic Server
22	Akonix Server
23	HubIM Server
24	Jabber Server
25	IM-Age Server
For an AesMajorType of '5', the possible values of AesMinorType are:	
1	Client File System Agent; scanned files (Data At Rest)
2	Client File System Agent; saved or copied files (Data In Motion)
3	Client Print System Agent
17	File Scanning Agent
33	File System Import
65	External Agent API (file)
81	Network Boundary Agent (file)
97	ICAP Agent
113	Content Classification Service (CCS)
129	Client Network Agent

AesSubType

AesSubType identifies whether the sequence was captured from the original application, or through the CA DataMinder Data Replay or Event Import utilities. Possible values are listed in the following table:

Notes	
1	The event was captured by the original client.
2	The event was captured by the CA DataMinder Data Replay utility. **Note:** The Data Replay utility is no longer included in the current CA DataMinder distribution.
3	The event was captured by the CA DataMinder Event Import utility
4	The event was captured by the CA DataMinder Event Import utility via the Policy Engine Connector (Import Policy)
5	The event was captured by an 'offline' client (such as, the Network Boundary Agent in 'passive' mode)

EventAttributes

EventAttributes contains a human readable list of attributes in the general form "/<flag>" or "/<name>=<value>". The first form is used to set named Boolean flags on the event, whereas the second form allows name/values pairs to be encoded. The EventAttributes string may comprise combinations of the following attributes:

Notes	
/Bn	Boolean: The event was blocked with a notification message.
/Bq	Boolean: The event was blocked quietly.
/Ws	Boolean: The event provoked a warning dialog, which the user heeded.
/Wps	Boolean: A server side event provoked a warning which the user heeded. But before the event could be reanalyzed against policy, policy changed.
/Wc	Boolean: The event provoked a warning dialog, which the user ignored.
/Wa	Boolean: The event provoked a warning dialog that the user had previously ignored on the same site, and this dialog was therefore not shown.
/Is	Boolean: The event provoked an inform dialog, which the user heeded.
/Ips	Boolean: A server side event provoked an inform dialog. But before the event could be reanalyzed against policy, policy changed.

Notes	
/Ic	Boolean: The event provoked an inform dialog, which the user ignored.
/Ia	Boolean: The event provoked an inform dialog that the user had previously ignored on the same site, and this dialog was therefore not shown.
/Qn	Boolean: The event was quarantined with a notification message.
/Qq	Boolean: The event was quarantined quietly.
/PE	Boolean: The event was designated 'Personal' by the user.
/CS	Boolean: The event provoked a silent control (for example, user input was not requested and the event was not blocked).
/NA	Boolean: Action processing was halted by a 'No further actions' action
/RD	Boolean: The event was redirected.
/RP	Boolean: The event generated an automatic reply.
/SC	Boolean: The event used security (digitally signed or encrypted).
/SG	Boolean: The event was digitally signed.
/EN	Boolean: The event was encrypted.
/DR	Boolean: The event was protected by Digital Rights Management.
/TR	Boolean: The event has an associated partial transaction record.
/NT	Boolean: There are no triggers on this event.
/SPi	Boolean: The event has internal scope - ie, *exclusively* sent to, or received from locations matching the policy definition of 'Internal Email' (email events) or 'Intranet Sites' (web events).
/SPe	Boolean: The event has external scope - ie, *exclusively* sent to, or received from locations NOT matching the policy definition of 'Internal Email' (email events) or 'Intranet Sites' (web events).
/SPm	Boolean: The event has mixed scope - ie, NOT *exclusively* sent to, or received from locations matching the policy definition of 'Internal Email' (email events) or 'Intranet Sites' (web events).
/SId	String: Security ID. Used enforce custom row-level-security.
/ST	Boolean: The event was ultimately blocked, either directly by policy, or because the user heeded a warning.
/HRe	Boolean: All *external* recipients in the original TO/CC lists were moved to the BCC list.
/HRa	Boolean: *All* recipients in the original TO/CC lists were moved to the BCC list.
/DS	Boolean: The file was deleted quietly.
/RS	Boolean: The file's contents were replaced quietly.
/DSs	Boolean: The file's contents were securely obliterated and deleted quietly
/RSs	Boolean: The file's contents were securely obliterated and replaced quietly.
/HT	String: The type of the file hash (stored in the ExternalID field)

Notes	
/LHa	Boolean: The email contained a legal hold notification, which the user accepted
/LHr	Boolean: The email contained a legal hold notification, which the user rejected
/CA	Boolean: The event was categorized automatically
/CM	Boolean: The event was categorized manually by the user
/SV	32 bit Integer: The severity score of the trigger

Note: some attributes are mutually exclusive and will never appear in combination.

BlobType

- The BlobType column determines the type of the blob, that is, how the actual blob data is stored.

Notes	
0	**File blob**. The blob is stored on the host machines file system. For these blobs, the *BlobLocation* column defines the blob file location relative to the CA DataMinder 'Data' directory specified at install time.
1	**Remote (RDM) blob**. The blob data is held using a remote storage mechanism that is independent of CA DataMinder. For these blobs, the *BlobLocation* column contains a unique identifier used by CA DataMinder's Remote Data Manager (RDM) to retrieve the blob from the remote store.
2	**Centera blob**. The blob is stored on a Centera system. For these blobs, the *BlobLocation* column contains the Centera hash code that identifies the blob.
3	**IBM Content Manager blob**. The blob is stored on an IBM Content Manager (IBMCM) system. For these blobs, the *BlobLocation* column contains the IBMCM Resource Manager location identifier.
4	**SnapLock blobs**. The blob is stored on a NetApp SnapLock volume. For these blobs, the *BlobLocation* column defines the blob file location relative to the SnapLock volume.

Wgn3EventAudit

This table defines the audit trail for all events. The audit trail captures event viewings, audit status changes and auditor comments.

Column	Primary Key	Datatype	Length	Notes
AuditUID Key used to uniquely identify an entry in an event's audit trail.	Yes	IDENTITYDEF	13	
EventUID Key used to uniquely identify a captured or imported event.	Yes	IDENTITYREF	13	Foreign key: *Wgn3Event*
EventTimestamp The time at which the event occurred.	Yes	TIMESTAMP	8	Foreign key: *Wgn3Event*
IssueUID Key used to uniquely identify an event issue.		IDENTITYREF	13	A null value means that the entry relates to the entire event; otherwise it must reference a valid issue
NextAuditUID Reference to the next audit entry		IDENTITYREF	13	Foreign key: *Wgn3EventAudit*
BulkAudit Indicates if the audit entry was created from a bulk review.		INTEGER	4	A value of 1 means the entry was created from a bulk review
AuditTimestamp The time that the audit trail entry was generated.	Yes	TIMESTAMP	8	
LoginIDM Part-key that indicates the user that generated the transaction.		INTEGER	4	Foreign key: *WgnUserLogin*
LoginID Part-key that indicates the user that generated the transaction.		INTEGER	4	Foreign key: *WgnUserLogin*
AuditType (see page 45) The type of audit trail entry.	Yes	INTEGER	4	Follow the hyperlink for details.

Column	Primary Key	Datatype	Length	Notes
ApplicationID (see page 45) Identifies the application that triggered the audit trail entry		INTEGER	4	Follow the hyperlink for details.
AuditDetail The detail of the audit trail entry. Its content is specific to the **AuditType**.		LONGVARCHAR		For audit trail comments this will contain the comment, and for status changes this indicates the new status value.

ApplicationID

The ApplicationID defines the application that triggered the audit event trail entry. It could be one of the following values:

Notes	
-1	**Unknown**. An unknown application triggered the event audit entry.
0	**Data Management Console**.
1	**iConsole.**
2	**Quarantine Manager Service.**

AuditType

AuditType defines the type of the audit trail entry. Possible values are:

Notes	
-1	**Unknown**. An unknown audit event occurred.
0	**Event Viewed**. The 'AuditDetail' column is empty for automatically audited events. The 'AuditDetail' column may also contain a string representation of the duration (in seconds) that an event was viewed in the iConsole Quick View pane.
1	**Field 1 Changed**. The audit field 1 value for an issue was changed by an auditor. The 'AuditDetail' column defines the new field 1 index value (which can be used as an index into the WgnWellKnownString table for WKType 2 entries, or may be -1 to indicate "unset").
2	**Comment.** An auditor added a comment. The 'AuditDetail' column contains the actual comment.

Notes	
3	**Expiry.** The expiry date for an event was changed. The 'AuditDetail' column contains the new expiry date (decimal representation of a WGNTIMESTAMP).
4	**Permanent.** The 'Do not delete' setting for an event was changed. The 'AuditDetail' column contains the new setting (0 or 1).
5	**Mailed Event.** An audit notification mail was sent for an event/issue. The 'AuditDetail' column contains the summary details of the mail that was sent.
6	**Field 2 Changed.** The audit field 2 value for an issue was changed by an auditor. The 'AuditDetail' column defines the new field 2 index value (which can be used as an index into the WgnWellKnownString table for WKType 5 entries, or may be -1 to indicate "unset").
7	**Field 3 Changed.** The audit field 3 value for an issue was changed by an auditor. The 'AuditDetail' column defines the new field 2 index value (which can be used as an index into the WgnWellKnownString table for WKType 6 entries, or may be -1 to indicate "unset"). Note that for audit field 3 only, the auditdetail column may also contain one or more values, enclosed in square brackets, such as, [0][3][7].
8	**Quarantine.** The quarantine state for an event was changed. The 'AuditDetail' column contains one of the following values: 1 = Released from quarantine 2 = Rejected from quarantine 3 = Successfully sent by quarantine manager service after release from quarantine. 4 = Failed to be sent by quarantine manager service after release from quarantine (awaiting retry). 5 = Failed to be sent by quarantine manager service after release from quarantine (blocked).
9	**Issue Participant.** The (single) associated participant for an audit issue was changed. The 'AuditDetail' column contains the participant index (or -1 for "all participants"). This audit type has been superseded by types 14 and 15 as we now support multiple participants per issue, but will still exist in pre 4.7 DB versions.
10	**Issue Name.** The name of an audit issue was changed. The 'AuditDetail' column contains the new issue name.
11	**Issue Created.** An audit issue was created for an event. The 'AuditDetail' column contains the issue name.
12	**Event Exported.** The event was exported from the Data Management Console. The 'AuditDetail' column contains the export format ("Event File", "Microsoft Outlook Message", "Microsoft Personal Folder" or "Virtual Web Site").
13	**Event Printed**. The 'AuditDetail' column is empty for these events.
14	**Issue Participant Added.** A participant was added to the list of participants associated with an audit issue. The 'AuditDetail' column contains the participant index. This type has been added to support multiple participants per issue.

Notes	
15	**Issue Participant Removed.** A participant was removed from the list of participants associated with an audit issue. The 'AuditDetail' column contains the participant index. This type has been added to support multiple participants per issue.

Wgn3EventIssue

This table identifies all the audit issues that have been created against captured data. The details of each issue are contained in the Wgn3EventAudit table.

Column	Primary Key	Datatype	Length	Notes
IssueUID Key used to uniquely identify an event issue.	Yes	IDENTITYDEF	13	
IssueName The name of the issue, as specified in the iConsole or Data Management console.		VARCHAR	255	
EventUID Key used to uniquely identify a captured or imported event.	Yes	IDENTITYREF	13	Foreign key: *Wgn3Event*
EventTimestamp The time at which the event occurred.	Yes	TIMESTAMP	8	Foreign key: *Wgn3Event*

Wgn3EventParticipant

This table defines the relationship between an event and any users associated with the event. The user association may be via the Wgn3Address table and/or the WgnUserLogin table. Note that a user association via the WgnUserLogin table is true only for 'client-captured' events.

From schema version 3.50, the columns addressuid1 and addressuid2 are mandatory. You can therefore assume that all wgn3eventparticipant rows have a corresponding wgn3address row, irrespective of the type of event. Events that would not naturally have an associated address such as, a web event will have an address generated based on the user information.

Column	Primary Key	Datatype	Length	Notes
EventUID Key used to uniquely identify a captured or imported event.	Yes	IDENTITYDEF	13	Foreign key: *Wgn3Event*
ParticipantIndex Part-key used to identify an event participant.	Yes	INTEGER	4	
EventTimestamp The time at which the event occurred.	Yes	TIMESTAMP	8	Foreign key: *Wgn3Event*
AddressUID1 Part-key used to uniquely identify an email address or the address of an IM participant.		DECIMAL	13	Foreign key: *Wgn3Address*
AddressUID2 Part-key used to uniquely identify an email address or the address of an IM participant.		DECIMAL	13	Foreign key: *Wgn3Address*
ParticipantNodeUID The classification of the participant.		INTEGER	4	Foreign key: *Wgn3ClassificationNode*
InterventionNodeUID The classification that CA DataMinder assigns to the participant.		INTEGER	4	Foreign key: *Wgn3ClassificationNode*

Column	Primary Key	Datatype	Length	Notes
LoginIDM Part-key that indicates the participant who generated the event.		INTEGER	4	Foreign key: *WgnUserLogin*
LoginID Part-key that indicates the participant who generated the event.		INTEGER	4	Foreign key: *WgnUserLogin*
QueryFlags Denormalized column indicating if the event has associated Trigger and/or Audit Records		INTEGER	4	Possible values: 0 = None 1 = Trigger 2 = Issue 3 = Issue+Trigger 4 = Audit 5 = Audit+Trigger 6 = Audit+Issue 7 = Audit+Issue+Trigger

Wgn3EventQueue

This table contains a list of events whose blobs are currently stored as 'File' blobs but which need to be converted into other blob types (such as Centera blobs).

Column	Primary Key	Datatype	Length	Notes
EventUID Key used to uniquely identify a captured or imported event.	Yes	IDENTITYREF	13	Foreign key: *Wgn3Event*
EventTimestamp The time at which the event occurred.	Yes	TIMESTAMP	8	Foreign key: *Wgn3Event*
StoreType The type of storage that the BLOB should be converted to.		INTEGER	4	Contains a value of 1 or 2: 1 = Centera 2 = Centera (Requeued)

Location The current physical location/reference of blob file.	VARCHAR	255

Wgn3IssueParticipant

This table forms the intersection between Wgn3EventIssue and Wgn3EventParticipant, allowing a Wgn3EventIssue record to be associated with multiple records in Wgn3EventParticipants.

Column	Primary Key	Datatype	Length	Notes
IssueUID Key used to identify the Issue to which this participant issue is associated.	Yes	IDENTITYREF	13	Foreign Key to *Wgn3EventIssue*
EventUID Key used to uniquely identify a captured or imported event.	Yes	IDENTITYREF	13	Foreign Key to *Wgn3EventParticipant*
EventTimestamp The time at which the event occurred.	Yes	TIMESTAMP		Foreign Key to *Wgn3EventParticipant*
ParticipantIndex Key used to identify the participant to which this participant issue is associated.	Yes	INTEGER	4	Foreign Key to Wgn3EventParticipant

Wgn3IssueTrigger

This table forms the intersection between Wgn3EventIssue and Wgn3Trigger, allowing a Wgn3EventIssue record to be associated with multiple records in Wgn3Trigger.

Column	Primary Key	Datatype	Length	Notes
IssueUID Key used to identify the Issue to which this participant issue is associated.	Yes	IDENTITYREF	13	Foreign Key to Wgn3EventIssue
EventUID Key used to uniquely identify a captured or imported event.	Yes	IDENTITYREF	13	Foreign Key to Wgn3Trigger
EventTimestamp The time at which the event occurred.	Yes	TIMESTAMP		Foreign Key to Wgn3trigger
TriggerIndex Key used to identify the Trigger to which this issue Trigger is associated.	Yes	INTEGER	4	Foreign Key to Wgn3EventParticipant

Wgn3JobHistory

Maintains a history of various utilities that run on Oracle RDBMS.

The jobs whose run history is tracked via this table are:

- Partitioning utility
- Review Queue

In schema versions prior to 3.50 the partitioning utility and review queue recorded history in tables WGN3PurgeHistory and WGN3queueHistory respectively, this single table replaces these older tables.

Column	Primary Key	Datatype	Length	Notes
Jobtype The type of job		INTEGER	10	0 Partition Utility 1- Review Queue 2 – wgn_stats
Process The identifier of the process being run		INTEGER	10	The process that is being run, this is specific to the job type.
Run Unique identifier of a particular job.	Yes	INTEGER	10	
Step The last step executed		INTEGER	5	Internal, used to determine where a failed purge will restart from.
StartTime The time when the step started.		TIMESTAMP		
EndTime The time when the step completed.		TIMESTAMP		
D1 -5 5 date columns for recording job specific date values		TIMESTAMP		.
N1-10 10 numeric columns for recording job specific numeric values		INTEGER	10	

Column	Primary Key	Datatype	Length	Notes
C1-5 5 character columns for recording job specific character values		VARCHAR	30	
Status The status of the job		VARCHAR	30	COMPLETED, RESTARTED, STARTED or FAILED

To maintain compatibility with older releases views WGN3PURGEHISTORY and WGN3QUEUEHISTORY are introduced in schema version 3.50 to mimic the tables from previous release. These views only return rows of the appropriate type.

Wgn3JobState

Records the detailed status of the various utility jobs on Oracle RDBMS .

The jobs whose run history is tracked via this table are:

- Partitioning utility
- Review Queue
- Wgn_stats

In schema versions prior to 3.50 the partitioning utility and review queue recorded detail status in tables WGN3PurgeState and WGN3queueState respectively, this single table replaces these older tables. Note also in previous releases that these tables only held status information from the most recent run, the new table holds information from multiple runs and individual runs can be identified by the log number column value.

Column	Primary Key	Datatype	Length	Notes
Run Identifies a specific instance of a job that has run		INTEGER		
Jobtype Identifies the type of job.		INTEGER		0 Partition Utility 1- Review Queue 2 – wgn_stats

Column	Primary Key	Datatype	Length	Notes
LogNum Identifies the specific log enabling entries to be retained over time.		INTEGER		Each job type defines how many logs to retain.
Step Identifies the particular step performed.		INTEGER	5	
STMT Uniquely identifies each statement.		INTEGER	5	A step can comprise multiple statements.
Created Date and time the operation occurred		TIMESTAMP		
Statement What the actual SQL DDL performed is.		VARCHAR	2000	Contains error message if the statement fails.
Status Status of current statement.		VARCHAR	2000	Values are STARTED, COMPLETED, FAILED.

To maintain compatibility with older releases views WGN3PURGESTATE and WGN3QUEUESTATE are introduced in schema version 3.50 to mimic the tables from previous release. These views only return rows of the appropriate type.

Wgn3MgmtGroup

Defines the groups that a user is allowed to manage or review. This is an extension to the previous schema, which limited users to only a management group and its subgroups.

Column	Primary Key	Datatype	Length	Notes
UserIDM Part-key that uniquely identifies a CA DataMinder user.	Yes	INTEGER	4	Foreign key: *WgnUser*

Column	Primary Key	Datatype	Length	Notes
UserID Part-key that uniquely identifies a CA DataMinder user.	Yes	INTEGER	4	Foreign key: *Wgn3User*
GroupIDM Part-key identifying the group that the user can manage.		INTEGER	4	Foreign key: *WgnGroup*
GroupID Part-key identifying the group that the user can manage.		INTEGER	4	Foreign key: *WgnGroup*
IncludeChildren Currently unused.		BIT	1	

Wgn3RelatedEvent

This table contains custom information that "relates" or groups events together. This table contains custom information that "relates" or groups events together. It is used by the standard search when the “Pre-defined” rollup key is selected.

Column	Primary Key	Datatype	Length	Notes
EventUID	Yes	IDENTITYREF	13	Foreign Key to Wgn3Event
EventTimestamp	Yes	TIMESTAMP		The time at which the event occurred.
RelationType	Yes	INTEGER	4	Type of relationship. Arbitrary Integer to allow multiple rows of the same "type" of relationship to exist
RelationIndex	Yes	INTEGER	4	Relationship increment.
RelatedEventUID		IDENTITYREF	13	Related Wgn3Event.EventUID record (Not constrained).
RelatedEventTimeStamp		TIMESTAMP		Related Wgn3Event.EventTimeStamp record (Not constrained).
ApplicationID		INTEGER	4	Identifier of the application that created the record

Wgn3Transaction (Deprecated)

This table is deprecated and will be removed in a future release.

This table contains all transactions detected for all users.

Column	Primary Key	Datatype	Length	Notes
EventUID Key used to uniquely identify a captured or imported event.	Yes	IDENTITYREF	13	Foreign key: *Wgn3Event*
EventTimestamp The time at which the event occurred.	Yes	TIMESTAMP	8	Foreign key: *Wgn3Event*
TransType		INTEGER	4	Not used.
TransState (see page 58) The transactions current state.		INTEGER	4	Follow the hyperlink for details.
TransTimestamp The time at which the transaction occurred.		TIMESTAMP	8	
TransCurrency The currency in which the transaction occurred.		VARCHAR	4	Should be the ISO currency value.
TransTotal The total amount of the transaction.		DECIMAL	13	TransTotal is a fixed point decimal and must be divided by 1000 to get the actual value in the specified currency units
TransTax The amount of tax paid on the transaction.		DECIMAL	13	TransTax is a fixed point decimal and must be divided by 1000 to get the actual value in the specified currency units
TransDelivery The delivery charge associated with the transaction.		DECIMAL	13	TransDelivery is a fixed point decimal and must be divided by 1000 to get the actual value in the specified currency units

Column	Primary Key	Datatype	Length	Notes
PaymentMethod How the transaction was paid for.		VARCHAR	255	This is a simple string which the user must supply. It provides more information about the payment method (for example, 'Mastercard')
PaymentData Payment method specific information, such as a credit card number.		LONGVARCHAR		
TransSiteName The web site or email address associated with the transaction.		VARCHAR	255	
TransDescription A user supplied description of the transaction.		LONGVARCHAR		
TransComment A user supplied comment about the transaction.		LONGVARCHAR		
SupplierReference The suppliers reference number for the transaction.		VARCHAR	255	
BuyerReference The users reference number for the transaction.		VARCHAR	255	
TransactionIDM Part-key that uniquely identifies this transaction.		INTEGER	4	
TransactionID Part-key that uniquely identifies this transaction.		INTEGER	4	
ConfidenceLevels (see page 59) Indicates the degree of confidence with which the values of other columns in this table were detected.		VARCHAR	255	Follow the hyperlink for details.

Column	Primary Key	Datatype	Length	Notes
ParentIDM Part-key that uniquely identifies the full transaction associated with a partial transaction.		INTEGER	4	Only valid for partial transactions (see TransState). Foreign key: *TransactionIDM/ID*
ParentID Part-key that uniquely identifies the full transaction associated with a partial transaction.		INTEGER	4	Only valid for partial transactions (see TransState). Foreign key: *TransactionIDM/ID*

TransState

TransState describes the current state of the transaction. Possible values are:

Notes	
0	The transaction is 'full' – it contains the combined data of one or more partial transactions.
1	The transaction was extracted to the required level of confidence, but the user designated it as 'Not a Transaction'.
2	The transaction was *not* extracted to the required level of confidence, and the user designated it as 'Not a Transaction'.
3	The transaction was *not* extracted to the required level of confidence and the dialog was not shown.
4	The transaction is 'partial' – it contains transaction information extracted from a single sequence.
5	The transaction is 'full' and 'closed' – The transaction timeout has elapsed, and so no further changes can be made to it. This means that new partial transactions will NOT be matched with it.

ConfidenceLevels

ConfidenceLevels contains a human readable list of attributes in the general form "/<flag>" or "/<name>=<value>". The first form is used to set named Boolean flags on the event, whereas the second form allows name/values pairs to be encoded. The EventAttributes string may comprise combinations of the attributes in the following table.

Many of these values represent a 'level of confidence' in a transaction data field. These values range from 1 to 5, where 1=low confidence, 4 is the highest level of confidence of an automatically extracted transaction field and 5 means that the user entered the datum, or confirmed the auto-extracted value

Notes	
/TVu	Boolean: The 'TransTotal' field is required
/TVo	Boolean: The 'TransTotal' field can be overridden by the user
/TVa=<num>	32 bit Integer: Minimum acceptable level of confidence in the 'TransTotal'.
/TVc=<num>	32 bit Integer: Level of confidence of the current 'TransTotal'.
/TCu	Boolean: The 'TransCurrency' field is required.
/TCo	Boolean: The 'TransCurrency' field can be overridden by the user.
/TCa=<num>	32 bit Integer: Minimum acceptable level of confidence in the 'TransCurrency'.
/TCc=<num>	32 bit Integer: Level of confidence of the current 'TransCurrency'.
/TTu	Boolean: The 'TransTax' field is required.
/TTo	Boolean: The 'TransTax' field can be overridden by the user.
/TTa=<num>	32 bit Integer: Minimum acceptable level of confidence in the 'TransTax'.
/TTc=<num>	32 bit Integer: Level of confidence of the current 'TransTax'.
/SCu	Boolean: The 'TransDelivery' field is required.
/SCo	Boolean: The 'TransDelivery' field can be overridden by the user.
/SCa=<num>	32 bit Integer: Minimum acceptable level of confidence in the 'TransDelivery'.
/SCc=<num>	32 bit Integer: Level of confidence of the current 'TransDelivery'.
/PTu	Boolean: The 'PaymentMethod' field is required.
/PTo	Boolean: The 'PaymentMethod' field can be overridden by the user.
/PTa=<num>	32 bit Integer: Minimum acceptable level of confidence in the 'PaymentMethod'.
/PTc=<num>	32 bit Integer: Level of confidence of the current 'PaymentMethod'.
/PDu	Boolean: The 'PaymentData' field is required.

Notes	
/PDo	Boolean: The 'PaymentData' field can be overridden by the user.
/PDa=<num>	32 bit Integer: Minimum acceptable level of confidence in the 'PaymentData'.
/PDc=<num>	32 bit Integer: Level of confidence of the current 'PaymentData'.
/SRu	Boolean: The 'SupplierReference' field is required.
/SRo	Boolean: The 'SupplierReference' field can be overridden by the user.
/SRa=<num>	32 bit Integer: Minimum acceptable level of confidence in the 'SupplierReference'.
/SRc=<num>	32 bit Integer: Level of confidence of the current 'SupplierReference'.
/BRu	Boolean: The 'BuyerReference' field is required.
/BRo	Boolean: The 'BuyerReference' field can be overridden by the user.
/BRa=<num>	32 bit Integer: Minimum acceptable level of confidence in the 'BuyerReference'.
/BRc=<num>	32 bit Integer: Level of confidence of the current 'BuyerReference'.
/TDu	Boolean: The 'TransDescription' field is required.
/TDo	Boolean: The 'TransDescription' field can be overridden by the user.
/TDa=<num>	32 bit Integer: Minimum acceptable level of confidence in the 'TransDescription'.
/TDc=<num>	32 bit Integer: Level of confidence of the current 'TransDescription'.
/UCu	Boolean: The 'TransComment' field is required.
/UCo	Boolean: The 'TransComment' field can be overridden by the user.
/UCa=<num>	32 bit Integer: Minimum acceptable level of confidence in the 'TransComment'.
/UCc=<num>	32 bit Integer: Level of confidence of the current 'TransComment'.

Wgn3Trigger

This table contains all event triggers (and their associated actions) for all users. A trigger is something that caused one or more actions to be taken as a result of analyzing the event.

Column	Primary Key	Datatype	Length	Notes
EventUID Key used to uniquely identify a captured or imported event.	Yes	IDENTITYDEF	13	Foreign key: *Wgn3Event*

Column	Primary Key	Datatype	Length	Notes
TriggerIndex Part-key used to uniquely identify this trigger.	Yes	INTEGER	4	
EventTimestamp The time at which the event occurred.	Yes	TIMESTAMP	8	Foreign key: *Wgn3Event*
TriggerType (see page 62) The type of trigger.		INTEGER	4	Follow the hyperlink for details.
TriggerName The name of the trigger.		VARCHAR	255	
TriggerAttributes (see page 63) Attributes of the trigger.		VARCHAR	255	Follow the hyperlink for details.
TriggerText1 Trigger type specific text. Contains the most significant definition from policy that matched		LONGVARCHAR		
TriggerText2 Trigger type specific text. Contains the extracted data from the event that matched the policy definition of the trigger		LONGVARCHAR		
ActionName Name of the action that the trigger fired.		VARCHAR	255	
ActionAttributes (see page 64) Attributes of the action fired by the trigger.		VARCHAR	255	Follow the hyperlink for details.
ActionType (see page 66) Type of action fired by the trigger.		VARCHAR	255	Follow the hyperlink for details.

Column	Primary Key	Datatype	Length	Notes
PolicyID Identifies the policy classification associated with the trigger.		INTEGER		May be NULL if no policy classification is associated with the trigger. For a list of pre-configured policy classifications that may be defined in this column, please see the Wgn3ClassificationNode (see page 18) table.
Severity Indicates the severity value of the trigger.		INTEGER		A value of NULL indicates that the trigger has no severity value.

TriggerType

TriggerType describes the type of trigger condition detected. Possible values are:

Notes	
0x00003000	The user manually identified a transaction
0x01001000	Manual web recording was switched on.
0x01010000	Navigated to a specified web site.
0x01012000	Navigated to a specified secure web site (HTTPS).
0x01020000	Submitted a specified credit card number.
0x01022000	Submitted a specified keystring.
0x01023000	Uploaded a specified file.
0x01030000	Matched search text on a web page.
0x01040000	Matched a content agent on a web page.
0x01050000	Transaction detected (heuristic) on a web page.
0x01090000	Web page matched a classification.
0x02002001	Manually captured a read mail.
0x02002002	Manually captured a sent mail.
0x02010001	Received email from a specified sender.
0x02010002	Sent email to a specified recipient.

Notes	
0x02011001	Received encrypted email.
0x02011002	Sent encrypted email.
0x02012001	Received digitally signed mail.
0x02012002	Sent a digitally signed mail.
0x02020001	Received an email containing a specified credit card number.
0x02020002	Sent an email containing a specified credit card number.
0x02030001	Received an email containing a specified keystring.
0x02030002	Sent an email containing a specified keystring.
0x02040001	Received an email matching a content agent.
0x02040002	Sent an email matching a content agent.
0x02050001	Received an email containing a transaction (heuristic).
0x02080001	Received an email containing a specified attachment and/or an attachment containing a specified keystring.
0x02080002	Sent an email containing a specified attachment and/or an attachment containing a specified keystring.
0x02090001	Received a mail matching a classification.
0x02090002	Sent a mail matching a classification.
0x03010001	Activated a top-level window with a specified title belonging to a specified executable.
0x05090001	File matches a classification.
0x05091001	File in motion matches a classification.

TriggerAttributes

TriggerAttributes contains a human readable list of attributes in the general form "/<flag>" or "/<name>=<value>". The first form is used to set named Boolean flags on the event, whereas the second form allows name/values pairs to be encoded. The TriggerAttributes string may comprise combinations of the following attributes.

Notes	
/Tv=<num>	64 bit Integer - The value of a transaction.
/Tl=<num>	64 bit Integer - The transaction value limit above which the trigger may fire.
/CL=<string>	String – Name of the classification associated with the trigger.

Notes	
/GH	Boolean: The trigger is a 'ghost' copy of the trigger that caused the sequence to be captured, but is attached to an event in an earlier sequence (eg, manual recording).

ActionAttributes

ActionAttributes contains a human readable list of attributes in the general form "/<flag>" or "/<name>=<value>". The first form is used to set named Boolean flags on the event, whereas the second form allows name/values pairs to be encoded. The ActionAttributes string may comprise combinations of the following attributes. (Note: Some attributes are mutually exclusive and will never appear in combination).

Notes	
/PF	Boolean – Indicates that the action associated with the trigger was actually performed (and by implication its absence indicates that an action was not performed).
/AP	Boolean – An action was automatically performed on the user's behalf because a second instance of an identical action was automatically performed by CA DataMinder in the same manner that the original action was performed. For example if a warning was not heeded, subsequent examples of the same warning may be automatically not heeded by CA DataMinder.
/NP=<num>	32 bit integer – The presence of this attribute indicates that the action was not performed; its value is an enumeration that indicates the reason. Possible values are: 0=Action processing terminated in a higher priority action (eg, a previous warning was heeded by the user). 1=Client is not live (Import Policy or Data Replay. **Note:** The Data Replay utility is no longer included in the current CA DataMinder distribution.) 2=Event is being processed offline after-the-fact 3=Event is being resubmitted offline after having previously failed 4=Event was blocked by an older policy 5=Action was skipped because previous action indicated 'no further actions' 6=Action was not performed because the client does not support it 7=Part of an action was not performed because the client does not support it 8=Part of the action was not performed because it was superseded by a higher priority action.
/RD=<string>	String – Indicates a URL/email address that the web/email event will be redirected to if the action is performed.

Notes	
/UI=<num>	32 bit integer – The value is an enumeration that indicates the type of User Intervention that is specified in the action. Possible values are: 1=Block Quietly 2=Block with Notification 3=Warn 4=Warn but allow user to designate 'Personal' 5=None 6=Inform 7=Notify 8=Quarantine Quietly 9=Quarantine with Notification 10=Legal Hold Notification 11=Categorize single category only 12=Categorize multiple categories allowed 13=No further actions.
/RA=<num>	32 bit integer – The value is an enumeration that indicates how an incoming email that was blocked is handled by the action. Possible values are: 1=Delete 2=Replace the Content 3=None.
/IG	Boolean – The user did not heed the warning (that is, the user clicked 'Continue').
/IP	Boolean – The user indicated that the event was personal.
/FI	32 bit integer – The value is an enumeration that indicates the type of File Intervention that is specified in the action. Possible values are: 1=None 2=Delete quietly 3=Replace quietly 4=Securely delete quietly 5=Securely replace quietly 6=Categorize quietly 7=No further actions.
/DB	String – The label of the button the user selected in the legal hold dialog

ActionType

ActionType describes the type of action specified by the trigger. Possible values are:

Notes	
0x01003001	Capture the web page
0x01010000	Perform a web control action
0x02003001	Capture an incoming mail
0x02003002	Capture an outgoing mail
0x02010001	Perform an incoming mail control action
0x02010002	Perform an outgoing mail control action
0x03003001	Capture details of application usage
0x03010000	Perform an application usage control action
0x05003001	Capture file details
0x05004001	Capture file in motion details
0x05010000	Perform a file control action
0x05011000	Perform a file in motion control action

Wgn3User

This table contains CA DataMinder user accounts.

Column	Primary Key	Datatype	Length	Notes
UserIDM Part-key that uniquely identifies a CA DataMinder user.	Yes	INTEGER	4	None
UserID Part-key that uniquely identifies a CA DataMinder user.	Yes	INTEGER	4	

Column	Primary Key	Datatype	Length	Notes
UserName The name of the CA DataMinder user. The naming convention (if any) depends on the user creation policy settings.		VARCHAR	64	There may be only one active (non-deleted) user with this user name at any one time.
PolicyIDM Part-key identifying the policy document for this user.		INTEGER	4	Foreign key: *WgnPolicy*
PolicyID Part-key identifying the policy document for this user.		INTEGER	4	Foreign key: *WgnPolicy*
UserCreateTime The time that the CA DataMinder user was created.		TIMESTAMP	8	
UserRole (see page 68) The CA DataMinder role assigned to the user. This is used to determine the user's initial privileges.		INTEGER	4	Follow the hyperlink for details.
DeletedFlag Indicates whether the user account is deleted or not.		BIT	1	
UserData Salt used to hash the password.		VARBINARY	255	
PasswordHash A hashed copy of the users password.		VARBINARY	255	
UserPrivileges (see page 68) A bitmap of CA DataMinder privileges the user is allowed to perform. The initial value is taken from the 'privilege profile' associated with the user's role.		DECIMAL	13	Follow the hyperlink for details.

UserRole

Each user is assigned a role. The value for each role and its meaning is defined as follows:

Notes	
1	**Administrator**. Anyone granted this role has access to all features defined by the administrator privilege profile (at the time the role is granted).
2	**Manager**. Anyone granted this role has access to all features defined by the manager privilege profile (at the time the role is granted).
3	**User**. Anyone granted this role has access to all features defined by the user privilege profile (at the time the role is granted).

UserPrivileges

UserPrivileges is a bitmap of permissions that can be granted to CA DataMinder users in the administrator console. The bit values and their meanings are given below (the values given are bit positions):

Notes	
0	Force Infrastructure Replication.
1	Edit CMS Policy.
2	Edit Machine Hierarchy.
3	Edit User Hierarchy.
4	***DEPRECATED***
5	Adjust Privilege Profiles.
6	Change any User Password.
7	View Log files.
8	Modify Data.
9	View Captured Data.
10	Run the Executive Console.
11	Install License File.
12	***DEPRECATED***
13	Access Statistics.
14	Run Searches.

Notes	
15	***DEPRECATED***
16	Disable a Statistic.
17	View Policy.
18	Edit Policy.
19	View Machine Hierarchy.
20	View User Hierarchy.
21	Change Database Passwords.
22	Edit Agents.
23	View Agents.
24	Event Import.
25	View Audit Trail.
26	Update Audit Trail.
27	Change System Defaults.
28	Unrestricted SQL.
29	Perform Content Searches.
30	Perform Administrative Searches.
31	Bulk Session Login.
32	Always suppress automatic auditing.
33	Choose to suppress automatic auditing.
34	Allow auditing without viewing the event.
35	Allow export to virtual web site.
36	Web Console Search Management.
37	Control Quarantine.
38	Single Sign On Allowed.

Wgn3UserAddressEx

Defines an email or IM address for a user. A user can have multiple addresses; the actual address is stored separately in the Wgn3Address table, allowing multiple addresses to be referenced by the same user. Note from schema version 3.50 it is possible for this table to contain associations with system generated addresses derived from the user details. These type of associated address can be distinguished by using the address type value.

Column	Primary Key	Datatype	Length	Notes
UserIDM Part-key that uniquely identifies a CA DataMinder user.	Yes	INTEGER	4	None Foreign key: *Wgn3User*
UserID Part-key that uniquely identifies a CA DataMinder user.	Yes	INTEGER	4	Foreign key: *Wgn3User*
AddressUID1 Part-key used to uniquely identify an email address or the address of an IM participant.		DECIMAL	13	Foreign key: *Wgn3Address*
AddressUID2 Part-key used to uniquely identify an email address or the address of an IM participant.		DECIMAL	13	Foreign key: *Wgn3Address*
AddressType (see page 70) Defines the 'type' of address		INTEGER	4	Follow the hyperlink.

AddressType

Defines the type of association between the user and the address.

Notes	
0	The associated address is a 'normal' address, and email or IM address.
1	The associated address is an automatically system generated address that is assigned to the user for events captured that do not have a natural associated address, such as, a Web page.

WgnBLOB

This table contains references to Binary Large Objects (blobs) used by CA DataMinder. This table does not contain the BLOB data itself. Instead, it defines the type and location of the blob data.

Column	Primary Key	Datatype	Length	Notes
BLOBIDM Part-key identifying the blob.	Yes	INTEGER	4	
BLOBID Part-key identifying the blob.	Yes	INTEGER	4	
BlobType (see page 43) Identifies the type of the blob, such as a file-based blob.		INTEGER	4	Follow the hyperlink for details.
BlobTag (see page 73) Identifies the type of content of the blob, such as a policy blob.		INTEGER	4	Follow the hyperlink for details.
Location The location of the blob data. The format of this string depends on the **BlobType**.		VARCHAR	255	
BlobSize The real unaltered size of the blob data, independent of factors such as compression and encryption.		DECIMAL	13	The physical blob size (for example, compressed) is stored in the column **BLOBPhysicalSize**.
VersionIDM Used to detect and resolve replication update collisions.		INTEGER	4	
VersionID Used to detect and resolve replication update collisions.		INTEGER	4	
VersionRanking Used to detect and resolve replication update collisions.		INTEGER	4	

Column	Primary Key	Datatype	Length	Notes
PurgeState Indicates whether the row can be purged if 'Purge On Replicate' is enabled. This column is ignored on a CMS.		INTEGER	4	Possible values: 0 - Cannot be purged 1 - Can be purged
BLOBPhysicalSize The physical storage size occupied by the blob. The value depends factors such as the **BlobType** and whether the blob is compressed.		DECIMAL	13	For file blobs, this represents the disk space used by the blob rather than its actual data size (the file may be compressed).
BLOBState The current state of the blob.		INTEGER	4	Currently unused.
BLOBAttributes (see page 73) Attributes of the blob.		VARCHAR	255	Follow the hyperlink for details.

BlobType

- The BlobType column determines the type of the blob, that is, how the actual blob data is stored.

Notes	
0	**File blob**. The blob is stored on the host machines file system. For these blobs, the *BlobLocation* column defines the blob file location relative to the CA DataMinder 'Data' directory specified at install time.
1	**Remote (RDM) blob**. The blob data is held using a remote storage mechanism that is independent of CA DataMinder. For these blobs, the *BlobLocation* column contains a unique identifier used by CA DataMinder's Remote Data Manager (RDM) to retrieve the blob from the remote store.
2	**Centera blob**. The blob is stored on a Centera system. For these blobs, the *BlobLocation* column contains the Centera hash code that identifies the blob.
3	**IBM Content Manager blob**. The blob is stored on an IBM Content Manager (IBMCM) system. For these blobs, the *BlobLocation* column contains the IBMCM Resource Manager location identifier.
4	**SnapLock blobs**. The blob is stored on a NetApp SnapLock volume. For these blobs, the *BlobLocation* column defines the blob file location relative to the SnapLock volume.

BlobTag

The BlobTag column defines the type of data held within the blob. The Tag is defined as one of the following values:

Notes	
0	**Policy Blob**. These blobs contain policy data.
1	**Event (Captured Data) Blob**. These blobs contain captured data such as email or web page content.
2	**Privilege Blob**. These blobs contain CA DataMinder user privilege profiles.
3	**Restriction Policy Blob**. These blobs contain restriction policies.
4	**File Blob**. These blobs contain file data associated with CA DataMinder's own internal file system.

BLOBAttributes

The BLOBAttributes column allows string attributes to be defined for individual blob entries. This is currently only used by 'Remote' blobs.

Notes	
/TMP=*relativeFil*e	For 'Remote' blobs, this attribute indicates the location where the temporary copy of the remote blob resides when it has been retrieved from the remote store

Wgn3UserGroup

This table stores a user's association with groups in the CA DataMinder user hierarchy. Inevitably, users will move between different groups in an organization. When they change groups, they are subjected to different CA DataMinder policies and different levels of review. This table retains a user's group context as it applied when data was captured, rather than applying the context of the group that the user currently belongs to.

Note: For the current release, this table enforces a constraint that a user can only have a single 'current' group. Potentially, in future releases this table may permit a user to belong to multiple groups concurrently.

Column	Primary Key	Datatype	Length	Notes
UserGroupUID System-generated key that uniquely identifies a user group	Yes	INTEGER	4	
UserIDM Part-key that uniquely identifies a CA DataMinder user.		INTEGER	4	Foreign key: *Wgn3User*
UserID Part-key that uniquely identifies a CA DataMinder user.		INTEGER	4	Foreign key: *Wgn3User*
GroupIDM Part-key identifying the group that the user belongs to.		INTEGER	4	Foreign key: *WgnGroup*
GroupID Part-key identifying the group that the user belongs to.		INTEGER	4	Foreign key: *WgnGroup*
StartDate The date at which the user became a member of the group.		TIMESTAMP		
NextGroupHistoryUID Used to explicitly link the group history in chronological order, so that an 'end date' can be derived from the start date of the next entry.		INTEGER		This column is optional. A null value indicates a current group context for the specified user.

Column	Primary Key	Datatype	Length	Notes
EffectiveStartDate A denormalized value so that the date at which a user started in a group does not need to be re computed continually.		TIMESTAMP		If this is the first row for a User then this takes the value of 01-Jan-1753 , which is the earliest possible date that oracle and sqlserver support. For all other entries effective startdate will equal startdate.
EffectiveEndDate A denormalised value so that the date at which a user ended in a group does not need to be re computed continually.		TIMESTAMP		If this is the last row for a User then this takes the value of 31-Dec-9999 , which is the highest possible date that oracle and sqlserver support. For all other entries effective enddate will equal startdate of the next row in the sequence for that user.

Wgn3UserPropertyValue

Table entries represent an instance of a value of a specific property for a specific user. If the current display name is updated, an entry is created in the Wgn3UserPropertyValue table containing the previous value. It is held against a predefined property of 'Previous display name'.

Column	Primary Key	Datatype	Length	Notes
UserIDM Part-key that uniquely identifies a CA DataMinder user.	Yes	INTEGER	4	None Foreign key: *Wgn3User*
UserID Part-key that uniquely identifies a CA DataMinder user.	Yes	INTEGER	4	Foreign key: *Wgn3User*
UserPropIndex Part-key that uniquely identifies the user with an attribute.	Yes	INTEGER		Collation sequence, plus permits multiple values for the same property

Column	Primary Key	Datatype	Length	Notes
UserPropValue The actual property value.		VARCHAR	255	The actual value
PropertyID Identifies the property type.		INTEGER		Identifies the type of the property. This value corresponds to the 'WellKnownIndex' in the WgnWellKnownString table. There are also 'special' values which have native ID values: -10 specifies a user's 'Full Name' -30 specifies a BusinessObjects user account that is mapped to this CA DataMinder user account. Foreign key: *WgnWellKnownString*
DataModified The date at which the property was last modified.		TIMESTAMP		Date of creation or last modification

WgnFile

This table is used to store details about the files within the CA DataMinder distributed file system.

Column	Primary Key	Datatype	Length	Notes
FileIDM Part-key identifying a file entry within the CA DataMinder file system.	Yes	INTEGER	4	
FileID Part-key identifying a file entry within the CA DataMinder file system.	Yes	INTEGER	4	

Column	Primary Key	Datatype	Length	Notes
FileVersion The version of the file within the file system.	Yes	INTEGER	4	The file system allows multiple concurrent versions of the same file.
FileType (see page 78) The type of the file (eg, file or directory).		INTEGER	4	Follow the hyperlink for details.
FileName The name of the file.		VARCHAR	255	
FileClass (see page 79) User defined class of the file, provided for file grouping.		INTEGER	4	Follow the hyperlink for details.
FileDescription A use specified description of the file.		VARCHAR	255	
FileAttributes Attributes of the file.		VARCHAR	255	Currently undefined
CreateTimestamp The time that the file was created.		TIMESTAMP	8	
ModifyTimestamp The time that the file was modified.		TIMESTAMP	8	
OwnerType Indicates whether the owner type of the file (eg, user or group).		INTEGER	4	
OwnerIDM OwnerType specific part-key identifying the file owner.		INTEGER	4	
OwnerID OwnerType specific part-key identifying the file owner.		INTEGER	4	
ParentIDM Part-key identifying the files parent.		INTEGER	4	Foreign key: *WgnFile*

Column	Primary Key	Datatype	Length	Notes
ParentID Part-key identifying the files parent.		INTEGER	4	Foreign key: *WgnFile*
ParentVersion Part-key identifying the files parent.		INTEGER	4	Foreign key: *WgnFile*
DataIDM Part-key identifying the BLOB containing the file content.		INTEGER	4	Directories have no associated BLOB Foreign key: *WgnBLOB*
DataID Part-key identifying the BLOB containing the file content.		INTEGER	4	Directories have no associated BLOB Foreign key: *WgnBLOB*
VersionIDM Used to detect and resolve replication update collisions.		INTEGER	4	
VersionID Used to detect and resolve replication update collisions.		INTEGER	4	
VersionRanking Used to detect and resolve replication update collisions.		INTEGER	4	

FileType

FileType defines the type of file. The following types are currently defined:

Notes	
0	**File**. Denotes a 'regular' file containing data.
1	**Folder/Directory**. A 'special' file used as container for other files.
2	**Link**. Defines a file entry that shares its file content with another file in the file system (currently unsupported).

FileClass

The file class column is used to classify files according to their use. The following classes are currently defined by CA DataMinder:

Notes	
0	**Normal**. Files/Directories which do not require classification
1	**Script**.
2	**Classifier**.
3	**Executive Console (ECON)**. Files used by ECON to store settings and so on.
5	**Search.** iConsole search.
6	**Saved Search**. iConsole saved search.

WgnGroup

The WgnGroup table holds information relating to CA DataMinder Groups. The relationships between groups is maintained in a separate table (see the WgnPolicyRelation (see page 91) table).

Column	Primary Key	Datatype	Length	Notes
GroupIDM Part-key used to uniquely identify a group within the installation.	Yes	INTEGER	4	
GroupID Part-key used to uniquely identify a group within the installation.	Yes	INTEGER	4	
GroupName The group name.		VARCHAR	64	Group names do not need to be unique, provided that they have different parent groups.
PolicyIDM Part key identifying the group's policy document.		INTEGER	4	*FK -> WgnPolicy*

Column	Primary Key	Datatype	Length	Notes
PolicyID Part key identifying the group's policy document.		INTEGER	4	*FK -> WgnPolicy*
DeletedFlag Indicate whether the group has been deleted.		BIT	1	
GroupCreateTime		TIMESTAMP	8	

WgnGroupRelation

This table defines the hierarchy of CA DataMinder groups.

Column	Primary Key	Datatype	Length	Notes
ChildGroupIDM Part-key identifying a child group.	Yes	INTEGER	4	Foreign key: *WgnGroup*
ChildGroupID Part-key identifying a child group.	Yes	INTEGER	4	Foreign key: *WgnGroup*
ParentGroupIDM Part-key identifying the child group's parent		INTEGER	4	Foreign key: *WgnGroup*
ParentGroupID Part-key identifying the child group's parent		INTEGER	4	Foreign key: *WgnGroup*

WgnID

This table is used to control the IDs generated by the machine.

Column	Primary Key	Datatype	Length	Notes
IDType (see page 81) Identifies the type of ID.	Yes	INTEGER	4	Follow the hyperlink for details.
NextID Indicates the next available ID to be allocated.		INTEGER	4	Each object type will allocate these in batches for performance reasons.
PrevID The previous ID allocated.		INTEGER	4	Used internally to determine if unused batch IDs can be returned on service shutdown.

IDType

IDType indicates the type of ID:

Notes	
0	Machine
1	User
2	BLOB
3	Policy
4	Collection
5	Sequence
6	Group
7	Cache
8	Version
9	Transaction
10	Login
11	Statistic
12	Statistic event list
13	File

Notes	
14	User-Group

WgnMachine

This table contains all details of machines accounts for machines in the CMS machine hierarchy.

Column	Primary Key	Datatype	Length	Notes
MachineIDM Part-key that uniquely identifies a machine.	Yes	INTEGER	4	None
MachineID Part-key that uniquely identifies a machine.	Yes	INTEGER	4	
MachineName The name of the machine.		VARCHAR	64	There may be only one active (non-deleted) machine with this name at any one time.
PolicyIDM Part-key identifying the machines policy document.		INTEGER	4	Foreign key: *WgnPolicy*
PolicyID Part-key identifying the machines policy document.		INTEGER	4	Foreign key: *WgnPolicy*
MachineCreateTime The time at which the machine account was created.		TIMESTAMP	8	
MachineRole (see page 83) The role of the machine (eg, CMS).		INTEGER	4	Follow the hyperlink for details.
MachineAttributes Bitmap of machine attributes.		DECIMAL	13	Currently unused.
MachineIPAddress The IP address of the machine.		VARCHAR	64	Currently unused.

Column	Primary Key	Datatype	Length	Notes
DeletedFlag Identifies whether this machine account deleted.		BIT	1	
VersionIDM Used to detect and resolve replication update collisions.		INTEGER	4	
VersionID Used to detect and resolve replication update collisions.		INTEGER	4	
VersionRanking Used to detect and resolve replication update collisions.		INTEGER	4	

MachineRole

MachineRole can be one of the following values:

Notes	
1	CMS.
2	Gateway.
3	Client.
4	Utility.

WgnMachineLogin

This table contains information about machine logins. Each machine must login to the machine hierarchy (preferably to its parent machine) before any user logins can take place.

Column	Primary Key	Datatype	Length	Notes
MachineIDM Part-key identifying the machine associated with the login record, and the login record itself.	Yes	INTEGER	4	Foreign key: *WgnMachine*
MachineID Part-key identifying the machine associated with the login record, and the login record itself.	Yes	INTEGER	4	Foreign key: *WgnMachine*
LoginState (see page 84) The state of the machine login, that is, logged in or out.		INTEGER	4	Follow the hyperlink for details.
LastLogonTime The time that the machine last logged in/out.		TIMESTAMP	8	
SoftwareVersion The version of the software running on the machine when it last logged in.		VARCHAR	64	
PolicyVersion The version of the machine policy last reported by the machine.		VARCHAR	255	

LoginState

LoginState may be set to any of the following values:

Notes	
0	Logged Off.
1	Logged On.

WgnMachineRelation

This table defines the machine hierarchy.

Column	Primary Key	Datatype	Length	Notes
ChildMachineIDM Part-key identifying a child machine.	Yes	INTEGER	4	Foreign key: WgnMachine
ChildMachineID Part-key identifying a child machine.	Yes	INTEGER	4	Foreign key: WgnMachine
ParentMachineIDM Part-key identifying the child machine's parent.		INTEGER	4	Foreign key: WgnMachine
ParentMachineID Part-key identifying the child machine's parent.		INTEGER	4	Foreign key: WgnMachine

WgnMonitorCache

This table is used to store objects changes that need to be replicated to other machines.

Column	Primary Key	Datatype	Length	Notes
CacheID Unique ID for the cache entry.	Yes	INTEGER	4	Cache entries should be ordered by ID to ensure they are read in the correct order.
MonitorID (see page 86) The replication monitor that owns the cache entry.		INTEGER	4	Follow the hyperlink for details.
CacheType (see page 87) The type of cache entry		INTEGER	4	Follow the hyperlink for details.
CacheData The update data associated with the cache entry.		LONGVARBINARY		May be NULL, depending on the CacheType.

Column	Primary Key	Datatype	Length	Notes
DBOperation (see page 87) The database operation associated with the cache type.		INTEGER	4	Follow the hyperlink for details.
TransIDM Part-key identifying the replication transaction ID.		INTEGER	4	Unused
TransID Part-key identifying the replication transaction ID.		INTEGER	4	Unused
TransCommit Indicates whether the remote machine has committed the transaction.		INTEGER	4	Unused
MachineIDM Part-key identifying the machine which sourced the cache entry		INTEGER	4	Foreign key: *WgnMachine* 'Sourced' refers to immediate parent/child. NULL if locally sourced.
MachineID Part-key identifying the machine which sourced the cache entry		INTEGER	4	Foreign key: *WgnMachine* 'Sourced' refers to immediate parent/child. NULL if locally sourced.
RetryCount Indicates the number of attempts to replicate the row to the parent machine.				Used for Application Data only.

MonitorID

MonitorID identifies the replication monitor that 'owns' the cache entry. Possible values are:

Notes	
1	Application Data Monitor
2	Infrastructure Data Monitor

CacheType

CacheType identifies the type of the cache entry.

Notes	
0	Object Key
1	Object
3	Replicate Onwards

DBOperation

DBOperation is only valid for CacheType 0/1 entries. Possible values are:

Notes	
1	Write
2	Delete

WgnMonSubscriber

This table is used to persist information about replication subscribers to the current machine (ie, the machines that a replication update must be posted to).

Column	Primary Key	Datatype	Length	Notes
MonSubscriber Name of a subscribing machine.	Yes	INTEGER	4	
MonitorID The monitor associated with the subscription.	Yes	INTEGER	4	See WgnMonitorCache (see page 85) for a list of valid Monitor IDs.
CacheID The last Cache ID read by the subscriber (for this monitor).		INTEGER	4	A value of 0 means all relevant cache entries have been read.
SubscriberState (see page 88) Current Subscription state.		INTEGER	4	Follow the hyperlink for details.

Column	Primary Key	Datatype	Length	Notes
IsCaching Must this machine cache updates for this subscriber?		BIT	1	
LastCacheReadTime Time the subscriber last read cache entries.		TIMESTAMP	8	
IsValidSubscriber Is the subscriber in a valid state?		BIT	1	

SubscriberState

SubscriberState indicates the current state of the subscribing machine for the particular monitor:

Notes	
1	**New Subscriber**. *Only applies to 'child' Infrastructure Data Monitor subscribers* Machines in this state have yet to make contact their parent. The monitor uses this state to send 'New Parent Notification' messages to its children during replication notifications. Subscribers move to state '2' when they make contact with the monitor.
2	**Subscriber out of Synchronization**. *Only applies to 'child' Infrastructure Data Monitor subscribers* Machines in this state need to resynchronize their infrastructure data (user, machine, policy data etc.) with their parent because it cannot be guaranteed to be consistent with the parent machine. When resynchronization completes subscribers move to state '3'.
3	**Synchronized**. Machines is this state are guaranteed to be synchronized with this machine **except** for any outstanding table updates for the machine in the WgnMonitorCache (see page 85) table. This state does not indicate that all outstanding data has been sent to the machine, only that it is capable of receiving object changes. Machines will move back to state '2' if they contain outstanding data that has not been retrieved within the policy controlled 'CacheTimeout' value.

WgnPolicy

This table defines all of the policy documents contained within the CA DataMinder installation.

Column	Primary Key	Datatype	Length	Notes
PolicyIDM Part-key that uniquely identifies the policy within CA DataMinder.	Yes	INTEGER	4	
PolicyID Part-key that uniquely identifies the policy within CA DataMinder.	Yes	INTEGER	4	
PolicyName Name of the policy. This usually reflects the name of the user, group or machine that the policy belongs to.		VARCHAR	64	
PolicyCategory (see page 90) Indicates the category of the policy document. Categories are common to each type of policy.		INTEGER	4	Follow the hyperlink for details.
PolicyType (see page 91) The type of the policy document (ie, machine, group or user).		INTEGER	4	Follow the hyperlink for details.
PolicyVersion The version of the policy document.		INTEGER	4	Each time the policy document is updated, this value is incremented.
Encrypted		BIT	1	Unused
Compressed		BIT	1	Unused
PolicyBLOBIDM Part-key identifying the BLOB that contains the policy document data.		INTEGER	4	Foreign key: *WgnBLOB*

Column	Primary Key	Datatype	Length	Notes
PolicyBLOBID Part-key identifying the BLOB that contains the policy document data.		INTEGER	4	Foreign key: *WgnBLOB*
PolicyCreateTime The time at which the policy document was created.		TIMESTAMP	8	
VersionIDM Used to detect and resolve replication update collisions.		INTEGER	4	
VersionID Used to detect and resolve replication update collisions.		INTEGER	4	
VersionRanking Used to detect and resolve replication update collisions.		INTEGER	4	

PolicyCategory

PolicyCategory defines the category of the policy document. The following categories are supported:

Notes	
1	**System Master.** The top-level policy that cannot be edited.
2	**Master.** The top level editable policy
3	**Default**.
4	**Generic**.
5	**Local**.
6	**Editor.** Used by the Administrator and User consoles to provide details about policy settings.
7	**Blank.** Empty policy of the relevant policy type.
8	**Schema**.
9	**Common Client**. This policy is the policy from which all client machine policies are derived.
10	**Restriction**. These restrict policy access and content.

Notes	
11	**Common Gateway**. This is the policy from which all gateway machine policies are derived.

PolicyType

PolicyType indicates the type of the policy. The types are defined in the following table:

Notes	
1	**User**. User policy documents are assigned to all CA DataMinder users and groups.
2	**Machine**. These policies are assigned to all CA DataMinder machines.

WgnPolicyRelation

This table stores the relationships between all policy documents defined in the WgnPolicy (see page 89) table.

Column	Primary Key	Datatype	Length	Notes
ChildPolicyIDM Part-key defining the ID of a child policy.	Yes	INTEGER	4	None
ChildPolicyID Part-key defining the ID of a child policy.	Yes	INTEGER	4	
ParentPolicyIDM Part-key defining the ID of a parent policy.		INTEGER	4	
ParentPolicyID Part-key defining the ID of a parent policy.		INTEGER	4	

WgnStatDefinition (Deprecated)

This table is deprecated and will be removed in a future release.

This table contains a list of all defined statistics maintained by the CMS.

Column	Primary Key	Datatype	Length	Notes
StatisticID Unique identifier for the statistic definition.	Yes	INTEGER	4	
LongName 'Friendly' name of the statistic.		VARCHAR	255	
ShortName Unique name used in statistic calculations etc.		VARCHAR	255	
Description A description of the statistic.		LONGVARCHAR		
StatType (see page 93) The statistic type.		INTEGER	4	Follow the hyperlink for details.
StatAttributes (see page 94) Statistic attributes.		INTEGER	4	Follow the hyperlink for details.
ListLength Defines the maximum number of events to be retained.		INTEGER	4	Event list statistics only
ForecastLength Defines the number of 'future' values supported by the statistic.		INTEGER	4	Static statistics only
HistoryLength Defines the how many previous time period values should be maintained.		INTEGER	4	Value statistics only
TimePeriod (see page 94) Indicates time period for which statistics should be gathered.		INTEGER	4	Follow the hyperlink for details.

Column	Primary Key	Datatype	Length	Notes
TimePeriodOffset (see page 95) The 'offset' into the time period from which statistics should be calculated.		DECIMAL	13	Follow the hyperlink for details.
FormatString Defines the parameter values for parameter aware statistics.		VARCHAR	255	The format of this string is statistic specific.
CurrencyString The currency associated with the statistic.		VARCHAR	255	Currency statistics only.
CurrentPhysicalSlotIndex The physical (real) statistic event index for the current time period.		INTEGER	4	Value statistics only. Physical slots are 'wrapped' to maintain linear progression.
CurrentVirtualSlotIndex The virtual statistic event index for the current time period.		INTEGER	4	Value statistics only. Physical slots are 'wrapped' to maintain linear progression.
TimePeriodStartTime The actual time when the time period for the current index started.		TIMESTAMP	8	
StatisticEnabled Is the statistic enabled?		BIT	1	
ParentID Identifies the parent statistic of this statistic.		INTEGER	4	Foreign key: *WgnStatDefinition*

StatType

The following StatType values are valid:

Notes	
0	Node
1	Value
2	Event List

StatAttributes

StatAttributes is made up of a bitmap of the following values (bit position is indicated):

Notes	
0	Static
1	Currency
3	Insert List
4	Has History
5	Has Forecast
6	Is Node
7	Transient. A dynamic attribute to indicate the statistic is being updated and should be ignored.
8	Reset. A dynamic attribute indicating the statistic was recently reset.
9	Uses Parameters
10	Parameter Change causes reset

TimePeriod

TimePeriod indicates the period over which a statistic may be collected.

Notes	
0	Year
1	Quarter
2	Month
3	Week
4	Day
5	Hour
6	Minute

TimePeriodOffset

TimePeriodOffset unit is determined by the TimePeriod value. Below is a list of time periods and their associated offset units:

Notes	
Year	Offset in Days from Jan 1
Quarter	Offset in Days from beginning of Quarter
Month	Offset in Days from start of Month
Week	Offset in days since start of Week
Day	Offset in Hours since midnight
Hours	Offset in Minutes since start of Hour
Minutes	Offset in Seconds since start of Minute

WgnStatEvent (Deprecated)

This table is deprecated and will be removed in a future release.

This table defines all event list values for all event lists.

Column	Primary Key	Datatype	Length	Notes
EventListID Identifies the event list that the event belongs to.	Yes	INTEGER	4	Foreign key: *WgnStatEventList*
PhysicalIndex The physical event list slot entry storing the event.	Yes	INTEGER		
VirtualIndex The virtual index of the event list entry.		INTEGER		

Column	Primary Key	Datatype	Length	Notes
EventTimestamp The time that the underlying captured data event was generated.		TIMESTAMP	8	Foreign key: *Wgn3Event* Events may appear out of order against the relevant index value. This is because events from different client machines are processed by the CMS at different times.
StringValue A statistic specific description of the event.		VARCHAR	255	
SequenceIDM Part-key identifying the associated capture event.		INTEGER		Foreign key: *Wgn3Event*
SequenceID Part-key identifying the associated capture event.		INTEGER		Foreign key: *Wgn3Event*
EventIndex Part-key identifying the associated capture event.		INTEGER		Foreign key: *Wgn3Event*
StatisticID The statistic to which the value belongs.	Yes	INTEGER	4	Foreign key: *WgnStatDefinition*
OwnerType (see page 98) The type of statistic owner.	Yes	INTEGER	4	Follow the hyperlink for details.
OwnerIDM Part-key identifying the statistic owner.	Yes	INTEGER	4	
OwnerID Part-key identifying the statistic owner.	Yes	INTEGER	4	
PhysicalTimePeriodIndex The physical slot index that the value is occupying.	Yes	INTEGER	4	
VirtualTimePeriodIndex The virtual slot index that the value is occupying.		INTEGER	4	

Column	Primary Key	Datatype	Length	Notes
NumericValue The statistic value.		DECIMAL	13	

WgnStatEventList (Deprecated)

This table is deprecated and will be removed in a future release.

This table defines all event lists at 'owner' level for all event list statistics.

Column	Primary Key	Datatype	Length	Notes
StatisticID The statistic to which the list belongs.	Yes	INTEGER	4	Foreign key: *WgnStatDefinition*
OwnerType (see page 98) The type of statistic owner.	Yes	INTEGER	4	Follow the hyperlink for details.
OwnerIDM Part-key identifying the statistic owner.	Yes	INTEGER	4	
OwnerID Part-key identifying the statistic owner.	Yes	INTEGER	4	
PhysicalTimePeriodIndex The physical slot index that the value is occupying.	Yes	INTEGER	4	
VirtualTimePeriodIndex The virtual slot index that the value is occupying.		INTEGER	4	
EventListID A unique statistic/owner identifier for this list.		INTEGER	4	

WgnStatNumber (Deprecated)

This table is deprecated and will be removed in a future release.

This table stores all instances of 'Value' statistics, including all historic and future slot values.

Column	Primary Key	Datatype	Length	Notes
StatisticID The statistic to which the value belongs.	Yes	INTEGER	4	Foreign key: *WgnStatDefinition*
OwnerType (see page 98) The type of statistic owner.	Yes	INTEGER	4	Follow the hyperlink for details.
OwnerIDM Part-key identifying the statistic owner.	Yes	INTEGER	4	
OwnerID Part-key identifying the statistic owner.	Yes	INTEGER	4	
PhysicalTimePeriodIndex The physical slot index that the value is occupying.	Yes	INTEGER	4	
VirtualTimePeriodIndex The virtual slot index that the value is occupying.		INTEGER	4	
NumericValue The statistic value.		DECIMAL	13	

OwnerType

This describes the possible type of statistic instance owner that the Owner IDM/ID relates to.

Notes	
0	Group.
1	User.

WgnUserLogin

This table stores login records used to determine which machines a CA DataMinder user logs on from, and which OS user they logged on as. This table is referenced by the captured data, allowing the data to be coupled to a particular user login location.

Column	Primary Key	Datatype	Length	Notes
LoginIDM Part-key identifying a unique login instance. A user will have one of these for each machine they have logged on to.	Yes	INTEGER	4	
LoginID Part-key identifying a unique login instance.	Yes	INTEGER	4	
NativeUser Name of the OS user that the CA DataMinder is logged on as.		VARCHAR	128	
UserIDM Part-key identifying the CA DataMinder user associated with the login.		INTEGER	4	Foreign key: *Wgn3User*
UserID Part-key identifying the CA DataMinder user associated with the login.		INTEGER	4	Foreign key: *Wgn3User*
LoginState (see page 100) The state of the user login, that is, logged in or out.		INTEGER	4	Follow the hyperlink for details.
LastLogonTime The time the user last logged in or out.		TIMESTAMP	8	
PolicyVersion The version of user policy that is being used.		VARCHAR	255	

Column	Primary Key	Datatype	Length	Notes
SourceMachine The name of the machine on which the user logged in.		VARCHAR	255	For administrator console logins to the CMS, the source machine may not belong to the local CMS machine hierarchy.
DefaultLoginFlag This is the 'default' login record for the OS user. This is used to control automatic logins.		BIT	1	

LoginState

LoginState can be set to any of the following values:

Notes	
-1	Login Invalid. This login belongs to a user who has either been deleted or who was created as part of a bulk (server-side) session.
0	Logged Off.
1	Logged On.

WgnVersion

This table contains version information.

Column	Primary Key	Datatype	Length	Notes
WgnVersionID (see page 101) The unique ID for the version.	Yes	INTEGER	4	Follow the hyperlink for details.
WgnCurrentVersion The current version number.		INTEGER	4	

WgnVersionID

WgnVersionID identifies the version. Possible values are:

Notes	
0	DBMS Schema Major Version
1	DBMS Schema Minor Version
2	DBMS Schema Sub Version

WgnWellKnownID

This table is used to store the IDs of 'well known' objects. The actual referenced object depends on the object type associated with the well known ID.

Column	Primary Key	Datatype	Length	Notes
WellKnownID (see page 102) The Object Type specific ID used to obtain the well-known object's ID.	Yes	INTEGER	4	Follow the hyperlink for details.
ObjectType (see page 103) The type of Object whose ID is to be retrieved.	Yes	INTEGER	4	Follow the hyperlink for details.
ObjectIDM Part-Key identifying the well known object.		INTEGER	4	
ObjectID Part-Key identifying the well known object.		INTEGER	4	

WellKnownID

The value of the WellKnownID column has a different meaning for each ObjectType. A table is given for each object type:

User/Machine Policy	
User and Machine policy well known IDs return Policy IDs. The well known ID values below indicate the well known ID required:	
1	System Master
2	Master
3	Default
6	Editor
7	Blank
9	Common Client (Machine only)
10	Restriction
11	Common Gateway (Machine only)
Group	
Group types return Group IDs. The defined values are below:	
1	Master
2	Default
Machine	
Machine types return Machine IDs. The defined values are below:	
1	CMS
Privilege	
Privilege types return BLOB IDs (the blobs contain profiles). The defined values are below:	
1	Administrator
2	Manager
3	User
Statistic	
Statistic types return Statistic Definition IDs. The defined values are below:	
1	Root Group

ObjectType

ObjectType will be set to one of the following values:

Notes	
1	User Policy
2	Machine Policy
3	Group
4	Machine
5	Privilege
6	Statistic

WgnWellKnownStringBase

This table defines changeable or configurable strings. This table defines the unique identifier of each string, and a reference to the table WG3Stringi18n (see page 110) which holds the actual strings for that identifier.

Column	Primary Key	Datatype	Length	Notes
WKType (see page 104) Identifies the type of well known string to return.	Yes	INTEGER	4	Follow the hyperlink for details.
WKIndex (see page 109) The type specific index of the well known string.	Yes	INTEGER	4	Follow the hyperlink for details.
i18NID Encoded ID used to reference the string value from table WGN3Stringi18n.		VARCHAR	255	Encoding of WKType and WKIndex WKS[WkType|WKIndex]

WKType

WKType identifies the category of a particular string.

Notes	
0	Global.
1	User Property. The index corresponds to the property 'type' in the PropertyID column of the Wgn3UserPropertyValue table.
2	Audit Field 1. Each index corresponds to the field 1 status value and defines the status string.
3	Audit Comments. Each index corresponds to the predefined comment value and defines the comment string.
4	Audit Mail Subjects. Each index corresponds to the predefined mail subject value and defines the subject string.
5	Audit Field 2. Each index corresponds to the field 2 status value and defines the status string.
6	Audit Field 3. Each index corresponds to the field 3 status value and defines the status string.
7	Audit Field Names. Each index (0-2) corresponds to the audit field name value and defines the name string.
8	Audit Toolbar Button Configuration. Each index (0-4) corresponds to the audit button number and defines the configuration options for that button.
9	Quarantine Button Configuration. The index is always 0, and the WKString value defines the configuration options for both buttons.
10	Default Mail Template. The index is always 0, and the WKString value is the name of the default template used to populate the "compose audit mail" window.
11	Audit Configuration. Stores information regarding the auditing of Issues.
12	User-defined role names.
13	User-defined role privileges.
14	Machine state.
15	Replication status.
16	Replication synchronization stages.
17	Audit Field display order. Each index (0-2) corresponds to the audit field index value and defines the display order of the values for that field. The string is a (colon) delimited list of position values (0-39), such as, 0:3:1:4:2…, etc. means that the order of the audit strings would be s0, s2, s4, s1, s3, etc.
18	Audit Field 1 "available values" filter. Each index (0-39) corresponds to the audit field 1 index value and defines the available audit field 2 indexes that can be selected if a particular field 1 value is selected. The string is a (colon) delimited list of indexes, such as, 0:1:2:4 means that the only field 2 values available to be selected would be index 0, 1, 2 and 4.

Notes	
19	Audit Field 2 "available values" filter. Each index (0-39) corresponds to the audit field 2 index value and defines the available audit field 3 indexes that can be selected if a particular field 2 value is selected. The string is a (colon) delimited list of indexes, such as, 0:1:3:5 means that the only field 3 values available to be selected would be index 0, 1, 3 and 5.
20	Descriptions of Low/Medium/High severity
21	Resource Classifications held in wgn3resourcetype
22	Resource Values held in wgn3resourcetype
23	Security Model types
24	Security Models Ids
25	Security Model abbreviations

WKType Detail

This section provides full details about the following WKType values.

WKType	
8	Audit Toolbar Button Configuration. Each index (0-4) corresponds to the audit button number and defines the configuration options for that button. The configuration string is a comma-delimited string containing eleven values in the following order: **Version** – used internally to identify the configuration version (to allow support of older configuration strings). Current value for r14.0 is 2. **Enable Type** – Can be a value between 0 and 3. 0 = disable button, 1 = show button for single event audit only, 2 = show button for bulk audit only, 3 = show buttons for single and bulk audit. **Affected Field** – Can be a value between 0 and 2. Indicates which audit field is used as the basis of the audit operation. 0 = field 1, 1 = field 2, 2 = field 3. For example, if 0, then the audit button operation checks the current state of audit field 1 to determine whether the operation is valid.. **Index From** – Can be a value between -1 and 39. Indicates which audit field is used for validation of the operation. If -1, the button operation is valid for any event issue. If 0-39, the operation is valid for events with an issue where the audit field specified by the affected field (above) has the same index set currently. **Index To 1** – Indicates which index to set audit field 1 to as a result of the operation. Can be a value between -1 and 39. If -1, the audit field 1 setting is not changed by the button operation. If 0-39, the audit field 1 setting is changed to that index. **Index To 2** – Indicates which index to set audit field 2 to as a result of the operation. Can be a value between -1 and 39. If -1, the audit field 2 setting is not changed by the button operation. If 0-39, the audit field 2 setting is changed to that index. **Index To 3** – Indicates which index to set audit field 3 to as a result of the operation. Can be a value between -1 and 39. If -1, the audit field 3 setting is not changed by the button operation. If 0-39, the audit field 3 setting is changed to that index. **Invalid Action** – Indicates the action to take during a bulk audit if an invalid event is found, i.e. an event that has no issues matching the required "from" state. This is no longer used as bulk auditing has become an asynchronous operation in r14.0, so we leave it to the user to decide if they want to cancel at any point, based on a log of the audits so far. For reference (for earlier CMSes), the previous values were: 0 = abort the entire operation without updating any events (the entire list of events is validated up front), 1 = ignore invalid events, 2 = ignore invalid events but report them at the end of the operation. **Comment** – An optional comment added to the issue audit history when creating a new issue as a result of the button operation. **Issue Name** – The issue name to use when creating a new issue as a result of the button operation. **Tooltip** – An optional string to use as a tooltip for the button. If this is left empty, the tooltip is formed dynamically based on the configuration of the button.

WKType	
9	Quarantine Button Configuration. The index is always 0, and the WKString value defines the configuration options for both buttons. The configuration string is a comma-delimited string containing three values in the following order: **Version** – used internally to identify the configuration version (to allow support of older configuration strings). Current value for r14.0 is 1. **Release** from quarantine option – Specifies the action to take when the user releases a mail from quarantine. If 0, an accompanying comment is not required. If 1, the user is prompted to enter a comment, but it is not mandatory. If 2, the user must enter a comment. **Reject from quarantine option** – Specifies the action to take when the user rejects a mail from quarantine. If 0, an accompanying comment is not required. If 1, the user is prompted to enter a comment, but it is not mandatory. If 2, the user must enter a comment.
10	Default Mail Template. The index is always 0, and the WKString value is the name of the default template used to populate the "compose mail" window. This is no longer used as there is no longer a "Send Mail" option in the iConsole (Note that the default reply template is still used – see below).

WKType	
11	Audit Configuration. The index is always 0, and the WKString value defines the configuration options for the auditing of Issues in the iConsole. The configuration string is a comma-delimited string containing twelve values in the following order: **Version** – used internally to identify the configuration version (to allow support of older configuration strings). Current value for r14.0 is 5. **Allow multi-select audit field 3** – Can be 0 or 1 for off or on. If on, users can select multiple values from the audit field 3 control to assign to an issue. **Default audit field 1** – Can be a value between 0 – 39. Indicates the index of default value to display in audit field 1 when creating a new issue. **Default audit field 2** – Can be a value between 0 – 39. Indicates the index of default value to display in audit field 2 when creating a new issue. **Default audit field 3** – Can be a value between 0 – 39. Indicates the index of default value to display in audit field 3 when creating a new issue. **Default associated user** – Can be -3 (sender), -4 (all managed users) or -5 (none). This default is used when creating a new issue either manually or using an audit button. Note that -4 is invalid if the "Allow multiple associated users" option is off. **Allow multiple associated users for an issue** – Can be 0 or 1 for off or on. If on, more than one user can be associated with the issue. **Prevent issues having no associated users** – Can be 0 or 1 for off or on. If on, all issues are validated during creation/editing to ensure that one or more users have been associated with that issue. **Default audit mail reply template** – the name of the default template used to populate the audit mail compose popup initially. **New issue popup default field visibility** – A bit field that specifies which fields are shown on the new issue popup. Note that all fields can be made visible using an "Advanced" link on the popup. Bit 1 = name, bit 2 = user, bit 3 = comment, bit 4 = field1, bit 5 = field 2, bit 6 = field 3, bit 7 = incidents, bit 8 = not used (always set). **Default issue name** – Can be 0 – 3. If 0, the Issue name is set using a custom name specified manually by the user. If 1, the current audit field 1 value is used to set the issue name. If 2, the current audit field 2 value is used to set the issue name. If 3, the issue name has no default value and must be manually entered. **Custom name** – A string specifying the Issue name to use if the above option is 0.

WKIndex

WKIndex defines the index specific to the well known type. Each table below defines the known type specific indexes.

Type	Notes
Global	These are common strings. 1 DBMS schema owner (that is, the owning machine). 2 Event Retention Period. 3 Default Purge Age 4 RLS Security Model 5 RLS Security Model Version 6 RLS Security Model Script
User Property	The index of each user property entry corresponds to the appropriate property defined in the Wgn3UserPropertyValue table.
Audit Field 1	The index of each string corresponds to the audit field 1 status. This index is the AuditDetail value stored in the Wgn3EventAudit (see page 44) table entries where the AuditType (see page 45) value is 2.
Audit Comments	The index of each string corresponds to the predefined comment value.
Audit Mail Subjects	The index of each string corresponds to the predefined audit mail subject value.
Audit Field 2	The index of each string corresponds to the audit field 1 status. This index is the AuditDetail value stored in the Wgn3EventAudit (see page 44) table entries where the AuditType (see page 45) value is 6.
Audit Field 3	The index of each string corresponds to the audit field 1 status. This index is the AuditDetail value stored in the Wgn3EventAudit (see page 44) table entries where the AuditType (see page 45) value is 7.
Audit Field Names	The index of each string corresponds to the audit field value (0, 1 or 2).
Audit Toolbar Button Configuration	The index (0-4) corresponds to the audit button number.
Quarantine Button Configuration	The index is always 0 (there is only ever one entry of this type in the table).
Default Mail Template	The index is always 0 (there is only ever one entry of this type in the table).
Audit Configuration	The index is always 0 (there is only ever one entry of this type in the table).
User-defined role names	The index of each string corresponds to the role value which can be assigned to a user (it also matches the index used to identify the roles privileges).
User-defined role privileges	The index of each string corresponds to the role value which can be assigned to a user (it also matches the index used to identify the roles name). The privileges are encoded in a hexadecimal string format beyond the scope of this document.

Type	Notes
Machine state	The index corresponds to the numeric state value recorded in the machine state entries in the Wgn3Diagnostics table.
Replication status	The index corresponds to the numeric status recorded in the replication status entries in the Wgn3Diagnostics table.
Replication synchronization stages	The index corresponds to the numeric status recorded in the replication synchronization stage entries in the Wgn3Diagnostics table.
Audit Field Display Order	The index of each string corresponds to the audit field value (0, 1 or 2).
Audit Field 1 "available values" filter	The index of each string corresponds to the index of the field 1 value (0-39).
Audit Field 2 "available values" filter	The index of each string corresponds to the index of the field 2 value (0-39).
Resource Classifications	1 Review (note it is envisaged this will expand in future releases)
Resource Values	1 View (note it is envisaged this will expand in future releases)
Security Model types	1 Management Group (Standard) 2 Management Group (Sender) 3 Policy (Standard) 11 Management Group (Standard, Self-Exclude) 12 Management Group (Sender, Self-Exclude) 13 Policy (Standard, Self-Exclude)
Security Models Ids	The index 1 represents the 'default' model, additional models are assigned a unique id > 1. Hybrid models are encoded as x,y where x and y represent the two base model types.
Security Model abbreviations	MD MS PD MDX MSX PDX These abbreviations correspond directly (by matching wkindex values) with the Security Model types.

Wgn3Stringi18n

This table contains miscellaneous strings used by many areas within the product. The table permits multiple translations to be defined for any particular string instance. The table itself is normally only exposed via a database view that allows a single occurrence of a specific string to be returned according to the currently active locale setting of the session.

Column	Primary Key	Datatype	Length	Notes
I18nID Identifier of the string instance.	Yes	VARCHAR	255	For strings referenced by wgnwellknowstringbase, the identifier is of the form WKS[wktype|wkindex]. For strings referenced by wgn3classificationbase, the identifier is of the form WCN[classuid]. For other miscellaneous strings, there is no rigorous structure to this column.
IETFLangCode ISO Language Identifier.	Yes	VARCHAR	16	The ISO language identifier for the string. For example 'en' or 'fr'.
I18NString The string value.		VARCHAR	255	
Description Additional descriptive text associated with string.		VARCHAR	255	

Wgn_UE_Export

This table contains data relating to events exported by the Universal Extractor.

Column	Primary Key	Datatype	Length	Notes
EventUID Part-key identifying an event that was exported.	Yes	IDENTITYREF	8	
EventTimestamp Part-key identifying an event that was exported.	Yes	TIMESTAMP	8	

Column	Primary Key	Datatype	Length	Notes
ExportTimestamp Time that the event was last exported.		TIMESTAMP	8	This field is updated if the event is re-exported such as, because the audit trail has been extended.
SuccessTimestamp Time that the event was successfully loaded into the target system.		TIMESTAMP	8	This field is set by acknowledgement messages from the target system. When Data Integrity is not required this will always be set equal to ExportTimestamp.
Received Flag indicating whether acknowledgement has been received for the event.		BIT		0 – Acknowledgement from the target system is required but has not yet been received. 1 – Acknowledgement has been received or data integrity was not required.
Committed Flag indicating whether the event was successfully exported.		BIT		0 – Event has not yet been sent to the target system. Events left with this status will be retried in the next run as it indicates that the previous run failed to complete. 1 – Event successfully sent to target system (such as, written to XML file)
Error Description of error that occurred when attempting to export the event.		VARCHAR	255	Not used currently.

Wgn3Role

This table defines the Roles that can exist in the system. A Role in its generic form is simply used to collect together one or more resources, and if that role is assigned to a user (or another Role) then implicitly that assigns all the underlying resources. For CA DataMinder r12.5 this will only be partially implemented in the software to permit Policy Classes to be assigned to Roles for the purposes of policy based security models, but the underlying concept is more generic than that single use case.

Column	Primary Key	Datatype	Length	Notes
RoleUID The unique ID of the Role	Yes	INTEGER	4	
RoleName The name of the Role		VARCHAR	255	
ParentRoleUID The identifier of the parent role		INTEGER	4	The concept of hierarchic roles is not implemented in CA DataMinder r12.5 , but the column is included for future use.

Wgn3Resource

This will contain one entry for every resource that can exist in the system. Its relationship with the resource type is mandatory, each resource must be of a defined resource type. Its relationship with policy class is optional, as we do not want to be constrained that we can only create resources for policy classes. It's relationship with resource level will also be optional, but is unused in CA DataMinder r12.5 and is for future use

For CA DataMinder r12.5, there will be 1 Wgn3Resource for every entry in Wgn_v_Policy_Class, with a reference to the WGN3ResourceType that corresponds to 'Review/ View'.

Column	Primary Key	Datatype	Length	Notes
ResourceUID Unique ID of the resource	Yes	INTEGER	4	
ResourceTypeUID The resource Type		INTEGER	4	Foreign key to wgn3resourcetype
PolicyNodeUID The Policy Class		INTEGER	4	Foreign key to wgn_v_policy_class
LevelID The associated resource level		INTEGER	4	Not used in CA DataMinder r12.5
Name The name of the resource		VARCHAR	255	
NumericValue Numeric value to associate with the resource		INTEGER	4	Not used in CA DataMinder r12.5
CharValue Character value to associate with the resource		VARCHAR	255	Not used in CA DataMinder r12.5
DateValue Date value to associate with the resource		TIMESTAMP		Not used in CA DataMinder r12.5

Wgn3ResourceRole

This is a simple intersect table and is required to implement the relationship that a resource can be assigned to many roles and a role can be assigned to many resources. It represents the instance of a specific resource being assigned to a specific role.

Column	Primary Key	Datatype	Length	Notes
ResourceUID ID of the resource	Yes	INTEGER	4	Foreign key to WGN3RESOURCE
RoleUID The ID of the role	Yes	INTEGER	4	Foreign Key to WGN3ROLE

Wgn3ResourceType

Reference table that defines the various types of resource that can exist. In CA DataMinder r12.5 there will be only one value which will be to represent the Resourcetype to 'Review', it is envisaged that this list will expand over time to encompass more resource types.

Column	Primary Key	Datatype	Length	Notes
ResourceTypeUID Unique ID of the resourcetype	Yes	INTEGER	4	
ResourceTypeClass The class of the resource type		INTEGER	4	In CA DataMinder r12.5 there will only be one value 'Review'
ResourceTypeValue The value of the resource type		INTEGER	4	In CA DataMinder r12.5 there will only be one value 'View'

The columns ResourceTypeClass and ResourceTypeValue are numeric Id's, in order to obtain the corresponding description they need to be joined to WgnWellKnownString with values of 21 and 22 respectively for WKType. A view wgn_v_resource_type is available for this purpose.

Wgn3UserRole

This is a simple intersect and is required to implement the relationship that a user can be assigned to many roles and a role can be assigned to many users. It represents the instance of a specific role being assigned to a specific user

Column	Primary Key	Datatype	Length	Notes
UserId Part ID of the user	Yes	INTEGER	4	Foreign key to WGN3User
UserIdM Part ID of the user	Yes	INTEGER	4	Foreign key to WGN3User
RoleUID The ID of the role	Yes	INTEGER	4	Foreign Key to WGN3ROLE

Wgn3ReviewQueue

This table contains references to event participants that are populated by the review queue population process. This table is the queue of events queued by the review queue process that are required to be reviewed.

Column	Primary Key	Datatype	Length	Notes
GroupIDM Part-key identifying the user's group associated with the review queue entry.		INTEGER	4	
GroupID Part-key identifying the user's group associated with the review queue entry.		INTEGER	4	

Column	Primary Key	Datatype	Length	Notes
EventUID Part-key identifying the event participant for the review queue entry.	Yes	IDENTIFYREF		
EventTimestamp Part-key identifying the event participant for the review queue entry.	Yes	TIMESTAMP		
ParticipantIndex Part-key identifying the event participant for the review queue entry.		INTEGER	4	
UserIDM Part-key identifying the user associated with the review queue entry.		INTEGER	4	
UserID Part-key identifying the user associated with the review queue entry.		INTEGER	4	
Category The category of review queue metric associated with the review queue entry.		INTEGER	4	Possible values: 0 = Main selection 1 = Under subscription 2 = Not used 3 = Mandatory selection
Run The run of the review queue process that created this review queue entry.		INTEGER	4	Used as a reference to Wgn3JobHistory table.
Precedence		INTEGER	4	Not used.

Wgn3ReviewMetrics

This table contains the definition of Review Queue metrics, each row defines the metric for the specified group, the review queue process interprets the metric defined for a group as applying to all subgroups of that group, unless specifically overridden at a lower level in the hierarchy.

Column	Primary Key	Datatype	Length	Notes
Process The Review Queue process that this metric is associated with.		INTEGER	4	
GroupIDM Part-key identifying the user's group associated with the review queue entry.		INTEGER	4	
GroupID Part-key identifying the user's group associated with the review queue entry.		INTEGER	4	
Category The category of review queue metric associated with the review queue entry.	Yes	INTEGER	4	Possible values: 0 = Main selection 1 = Under subscription 2 = Not used 3 = Mandatory selection
Value The value of the metric.	Yes	INTEGER		Interpreted using the Type column.
Type Flag to determine if the Value column should be interpreted as a percentage or absolute value.	Yes	INTEGER	4	Possible values: 0 = Absolute 1 = Percentage
MTarget Flag to indicate which data set is used to calculate the runtime metric value.	Yes	VARCHAR		Possible values: A = All P = Prime
DTarget Flag to indicate which data set is used to calculate the runtime metric value.	Yes	VARCHAR		Possible values: A = All P = Prime

TMP_Wgn3RLS

An internal table that is dynamically populated at runtime to provide security context for searches. On SQL Server this table is a normal table and contains the column SPID (as shown below) on Oracle it is implemented as a Global Temp Table and does not have column SPID.

This table should never be accessed directly, only via the view WGN_V_RLS_1, which has a consistent interface for both SQL Server and Oracle.

On Oracle CBO stats should never be manually collected for this table as the contents are volatile, optimizer dynamic sampling (level 3) should be used.

Column	Primary Key	Datatype	Length	Notes
SPID This identifies the DBMS connection associated with the groups to apply security against.	Yes	INTEGER		For SQL Server databases only.
GroupIDM Part-key identifying the group that the user belongs to.	Yes	INTEGER	4	
GroupID Part-key identifying the group that the user belongs to.	Yes	INTEGER	4	
ParentGroupIDM Part-key identifying the parent group that the user belongs to.		INTEGER	4	
ParentGroupID Part-key identifying the parent group that the user belongs to.		INTEGER	4	

TMP_WGN3ADDRRLS

An internal table used to provide optimized performance for RLS and queries against users/groups/addresses.

On SQL Server this table is a normal table and contains the column SPID (as shown below) on Oracle it is implemented as a Global Temp Table and does not have column SPID.

This table should never be accessed directly, only via the view WGN_V_RLS_ADDR_1, which has a consistent interface for both SQL Server and Oracle.

On Oracle CBO stats should never be manually collected for this table as the contents are volatile, optimizer dynamic sampling (level 3) should be used.

Column	Primary Key	Datatype	Length	Notes
SPID This identifies the DBMS connection associated with the groups to apply security against.		INTEGER		For SQL Server databases only.
EffectiveStartDate Date the user joined this group.		TIMESTAMP	8	None
EndDate Date the user was removed from the group		TIMESTAMP	8	
AddressUID1 Part-key identifying the users address associated with the entry.		INTEGER	4	Can be NULL
AddressUID2 Part-key identifying the users address associated with the entry.		INTEGER	4	Can be NULL
UserIDM Part-key identifying the user associated with the entry.		INTEGER	4	
UserID Part-key identifying the user associated with the entry.		INTEGER	4	

Column	Primary Key	Datatype	Length	Notes
GroupIDM Part-key identifying the user's group associated with the entry		INTEGER	4	
GroupID Part-key identifying the user's group associated with the entry		INTEGER	4	

TMP_WGN3EXADDRRLS

An internal table used to provide optimized performance for RLS models that are required to exclude the user (currently the logged on user). This table contains the UserID/IDM and associated addressUID1/2 of the user that is to be excluded from RLS search results.

Note on SQL Server this table is a normal table and contains the column SPID (as shown below) on Oracle it is implemented as a Global Temp Table and does not have column SPID.

This table should never be accessed directly, only via the view WGN_V_RLS_EX_ADDR_1, which has a consistent interface for both SQL Server and Oracle.

On Oracle CBO stats should never be manually collected for this table as the contents are volatile, optimizer dynamic sampling (level 3) should be used.

Column	Primary Key	Datatype	Length	Notes
SPID This identifies the DBMS connection associated user to exclude.		INTEGER		For SQL Server databases only.
AddressUID1 Part-key identifying the users address associated with the excluded user.		INTEGER	4	

Column	Primary Key	Datatype	Length	Notes
AddressUID2 Part-key identifying the users address associated with the excluded user.		INTEGER	4	
UserIDM Part-key identifying excluded user.		INTEGER	4	
UserID Part-key identifying the excluded user.		INTEGER	4	

TMP_Wgn3PolicyRLS

An internal table that is dynamically populated at runtime to provide security context for searches. This table is populated if the logged on user is subject to a Policy based security model, and in which case it contains all the policy ids that user is permitted to access.

Note on SQL Server this table is a normal table and contains the column SPID (as shown below) on Oracle it is implemented as a Global Temp Table and does not have column SPID.

This table should never be accessed directly, only via the view WGN_V_RLS_POLICY_1 , which has a consistent interface for both SQL Server and Oracle.

On Oracle CBO stats should never be manually collected for this table as the contents are volatile, optimizer dynamic sampling (level 4) should be used.

Column	Primary Key	Datatype	Length	Notes
SPID This identifies the DBMS connection associated with the groups to apply security against.	Yes	INTEGER		For SQL Server databases only.
PolicyID Key identifying the Policy class that the user can access	Yes	INTEGER	4	

PolicyName Name of the Policy class that the user can access	VARCHAR	255
ParentPolicyID Id of the parent Policy class	INTEGER	4
ParentPolicyName Name of the parent Policy class	VARCHAR	255

TMP_WgnAdminIDs

ORACLE ONLY

This is an internal table used to help populate certain machine administration views.

Column	Primary Key	Datatype	Length	Notes
ObjectIDM Party-key defining an object ID.		IDENTITYREF	8	None
ObjectID Party-key defining an object ID.		INTEGER	4	None

TMP_ WgnUEExportedEvents

Internal table used by the Universal Exporter. For scalability, the XML for the event is retrieved in chunks into this table. The full XML for the event is then reassembled when writing the XML to file. The table contents persist only for the duration of a run of the Universal Extractor.

In Oracle this table is created as a Global Temporary Table, in SQL Server a temporary table is created dynamically.

Column	Primary Key	Datatype	Length	Notes
EventUID Part-key identifying an event that was exported.	Yes	IDENTITYREF	8	
EventTimestamp Part-key identifying an event that was exported.	Yes	TIMESTAMP	8	
Part The sequence number of the XML chunk.	Yes	INTEGER		
EventXML A chunk of the XML for the event.		VARCHAR	4000	

Chapter 3: Database Views

This section contains the following topics:

Row Level Security

Row level security (RLS), prior to V12.5, ensures that reviewers (or any search user) cannot see events associated with users outside of their management groups when searching the CMS database for events. From V12.5 onwards it is also possible to apply RLS based on certain policies that the reviewer can see or it is possible to have a combination of both types of RLS (this is known as a hybrid model). RLS is primarily applied to events, users and groups for management group RLS with the underlying tables (Wgn3Event, Wgn3User and WgnGroup) inaccessible directly to search users. For policy based RLS this also applies to triggers (Wgn3Trigger). In order to access these underlying tables and to ensure that RLS is not bypassed, several views have been created, and you should use these views when writing reports.

Within the CA DataMinder infrastructure each user of the system has a username and an associated set of administrative privileges that give permissions to certain features of the software. These privileges are set to either on or off. When a user requires a connection to the database the infrastructure checks the CA DataMinder user privileges. One privilege is 'Admin: Disable Security Model Filtering' and this determines whether or not RLS is applied to a user. By default, it is disabled for all CA DataMinder users except administrators, but can be enabled if a particular user or group of users requires access to all events, thus bypassing RLS.

More information:

Database Users: Owner and Search User (see page 126)
Implementing RLS: Oracle versus SQL Server (see page 127)
Connection Pool (see page 128)

Database Users: Owner and Search User

For the database, RLS is applied using schema separation. In SQL Server two database users are set up: owner/WGNUser (dbo) and search user. In Oracle there is an option when installing to have two of three database users: owner (optional), WGNUser and search user. In the two user model, owner and WGNUser are the same. For the purpose of this document we will assume the 2 user model for simplicity. The deployment guide gives instructions on how to set up a 3 user model. The owner does not have RLS applied and the search user does. The owner owns all objects in the database and has full access to them. The search user is granted access to certain objects within the database. Views (and for Oracle Synonyms) are used to create a common interface to the database that both users can use. The owner will have a view such as WGN_V_EVENT_n. Typically this view may just point to an underlying table that the owner could access directly if required. The search user is not granted permission to access this view. However another view in SQL Server or synonym in Oracle is created that is given the same name which the search user can access. The owner also owns a number of other viewsthat these top views reference. However the search user is granted access to these views. The search user view or synonym then points to this view rather than directly to the underlying table. It is this view and possibly further underlying views that allow RLS to be maintained. The details of all these view are described below.

From r12.5 the above database users still exit. Prior to V12.5 it was only possible to have a single search user with a resulting single RLS model. Now it is possible to create multiple RLS models, with an associated user, and different reviewers can be configured to apply RLS in different ways depending on the RLS model they are set. An overview of these security models can be found in the Database guide and setting up these models can be done via the ACon (see Administration Console help for details). For each RLS model that exists, a top level set of views/synonyms are created. These views then point to their own underlying view, such as WGN_V_EVENT_n -> WGN_V_m_EVENT_n, where m is just an integer that is incremented for each additional RLS model.

Implementing RLS: Oracle versus SQL Server

If the 'Admin: Disable Security Model Filtering' privilege is checked, RLS is not applied. In this case the user will have full access to all data in the database. A JDBC connection is set up to the database and connects to the database as Owner. If the privilege is not checked, then RLS will be applied. The first time a user attempts to set up a JDBC connection to the database, the infrastructure needs to know which data the user can access. Oracle and SQL Server do this is slightly different ways:

Oracle uses a package called WGN_CLIENT_INFO. Functions within package populate between one and four (dependent on the RLS model the reviewer uses) global temporary tables (GTTs), TMP_WGN3RLS, TMP_WGN3ADDRRLS, TMP_WGN3EXADDRRLS and TMP_WGN3POLICYRLS, with all the information required to maintain RLS. These tables are described below and accessed by the views WGN_V_RLS_1, WGN_V_RLS_ADRR_1, WGN_V_RLS_EX_ADRR_1 and WGN_V_RLS_POLICY_1 respectively in all searches. The nature of GTTs means that only the current user will be able to see their own data although other users will be able to see the tables.

SQL Server uses a slightly different implementation as it is not possible to create a view on a temporary table within SQL Server. Instead, SQL Server uses permanent heap tables TMP_WGN3RLS, TMP_WGN3ADDRRLS, TMP_WGN3EXADDRRLS and TMP_WGN3POLICYRLS,. Here, stored procedures are used to populate these tables. There is also an additional column (SPID) used in both tables to be able to identify the specific users. This is hidden from the report writer by the views WGN_V_RLS_1, WGN_V_RLS_ADRR_1, WGN_V_RLS_EX_ADRR_1 and WGN_V_RLS_POLICY_1. Additionally because SQL Server uses permanent tables to store the RLS data there is an extra overhead required to remove the rows from the tables once the connection is no longer required.

Connection Pool

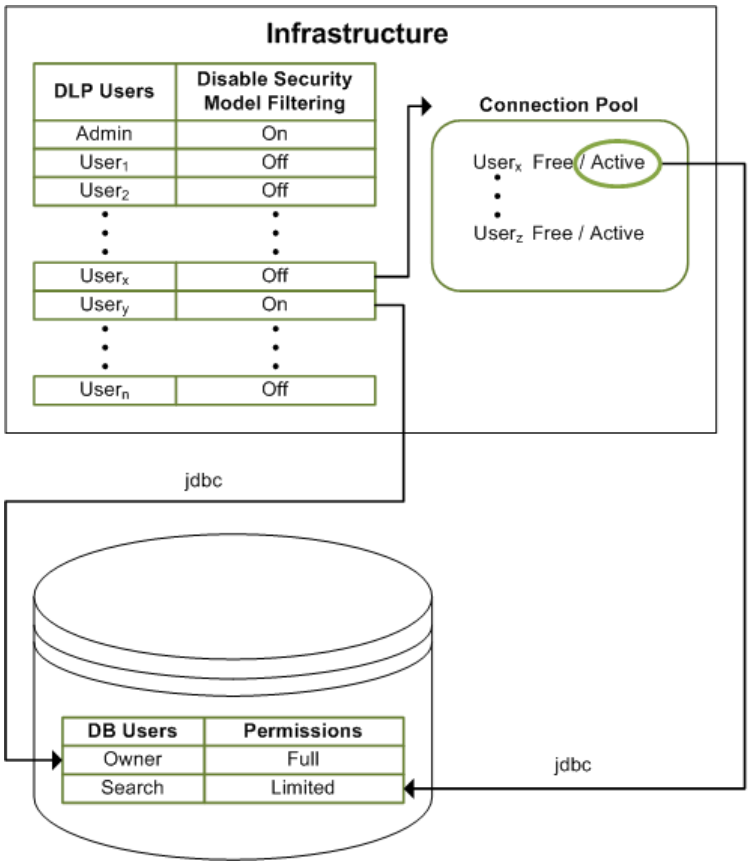
This initial setup of the RLS data can be quite expensive, so the CA DataMinder infrastructure maintains a connection pool to minimize unnecessary load on the database. Once the RLS data is loaded, an entry is put into the connection pool for that user and it is set to 'active' while connected to the database. Once the connection is no longer needed (for example, a search has finished running and returned the data to the iConsole), the entry in the connection pool is set to 'free'. The RLS data is still in the TMP tables.

When the user wants to run another search, the infrastructure checks the connection pool to see whether the user has a free connection. The infrastructure then connects to the database and uses the existing RLS data that is associated with this user, meaning that subsequent queries are less expensive.

If the user has an active connection and a second connection is required, a second connection is required in the connection pool, which incurs the expense of entering another set of rows into the TMP tables.

By default, entries in the connection pool are maintained for one hour. If the connection is left free for one hour, the connection is dropped from the pool and the RLS rows no longer exist in the database. Any subsequent searches require the RLS rows to be reinitialized.

The role of the connection pool is shown in the following diagram.



Role of Infrastructure and Connection Pool

Security Models

Security models ensure that reviewers can only see events they are permitted to see when searching the CMS database.

You can choose which security models are available on your CMS. You can also have multiple security models active at the same time, though each reviewer is linked to a single model.

For example, some reviewers may only be permitted to see events linked to users in their own management group. Other reviewers may only be permitted to see specific types or categories of events.

CA DataMinder supports the following security models:

Management Group (Standard)

This is the default model, optimized to allow fast searching. It is based on the CA DataMinder user hierarchy.

It uses e-mail addresses (including synthesized addresses for participants in Web and Application Monitor events) to map participants to CA DataMinder users. Under this model, reviewers can only view events where at least one participant was in their management group when the event was captured.

Management Group (Standard, Self-Exclude)

This model prevents reviewers from seeing their own events. As above, reviewers can only view events where at least one participant was in their management group. However, under this model the search results also exclude any events in which the 'logged-on user' (that is, the reviewer) was a participant.

Management Group (Sender)

Under this model, when a reviewer runs an e-mail search, they can only view events where the e-mail sender was in their management group when the event was captured.

Important! This sender-centric security model is only appropriate for e-mail searches. Searches for other event types will return zero results.

Management Group (Sender, Self-Exclude)

This model prevents reviewers from seeing their own e-mails (or any other events) when they run a search.

As above, reviewers can only view events where the e-mail sender was in their management group. However, under this model the search results also exclude any events in which the 'logged-on user' (that is, the reviewer) was a participant.

Policy (Standard)

This model ensures that reviewers can only see specific types of event. For example, this model can be used to ensure that HR reviewers only see events that relate to HR issues such as employee behavior, while Legal reviewers only see events that relate to legal issues such as litigation threats or a breach of attorney client privilege.

The model is based on *policy classes*. For categorization purposes, you can associate individual triggers with a *policy class*, such as 'Employee Behavior' or 'Legal'. When a trigger fires, the policy class is stored with the associated event.

Likewise, each reviewer has a policy role. A *policy role* links a user to a collection of policy classes. In effect, the policy role determines which policy classes a user is permitted to see. When the user runs a search, the results only include events associated with these policy classes.

Policy (Standard, Self-Exclude)

This variant of the Policy model prevents reviewers from seeing their own events. As above, reviewers can see only specific types of event. However, the search results also exclude any events in which the reviewer was a participant

Hybrid Model: Management Group and Policy

If required, you can add a hybrid model on your CMS. This combines the Management Group and Policy models. Its effect is to restrict reviewers so they can only see specific types of event associated with users in their management group. For example, under this model a reviewer in the Legal team can only review legal events associated with members of their management group.

Unrestricted

This model is not subject to row level security (RLS). It permits reviewers to see any database items (events, users, triggers, and so on) when they run a database query. For example, Search results or reports are not restricted by policy class or the reviewer's management group. This model is required by:

- CA DataMinder administrators. Paradoxically, this security model *restricts* the extent to which administrators can edit the CMS database. Specifically, it prevents administrators from inadvertently updating the CMS database when they exercise their 'Admin: Allow Unrestricted SQL Searches' privilege.
- CA DataMinder user accounts set up explicitly for use by external reporting tools *that require full access* when searching the Data Warehouse for events.

  Note: If the user of an external reporting tool is subject to row level security, CA DataMinder applies that user's security model (typically a Management Group model) when the user runs a report.

Note: You can only assign the Unrestricted security model to a CA DataMinder user if you have the 'Admin: Disable security model filtering' administrative privilege.

Important! Certain reports and the Review Queue are not designed for use with Policy security models. See the reference below for details.

Primary Views Definitions

This section describes the primary views a report writer should use when constructing queries. It provides details of what the view should be used for and when it should be used. Each view also contains a View stack and a table defining the columns in the view.

More information:

Naming Convention

The naming convention of all standard views starts with Wgn_V_ (views written for specific customers should not use this convention to avoid confusion). There is one exception to this with CA DataMinder r12.0 which will be discussed below. The rest of the view name contains an indication of what tables the view points to, followed by a digit. This digit does not guarantee anything other than, for example, that Wgn_V_Event_2 was developed after WGN_V_EVENT_1, but the view changed significantly enough to change the data returned thus potentially breaking existing searches. Details of whether these views have been deprecated are detailed below for each applicable view.

View Stacks

The view stacks detail what additional views and tables are referenced under the primary views. Each view stack looks similar to this:

Wgn_V_Event_n

→ wgn_v_m_event_n

→ WGN_V_MD_EVENT_n

→ *Table_1*

View_1

→ *Table_2*

This is common to most of the primary views. The Primary view points to another underlying view. This view points to a view, which in most case references a view with almost the same name but with an integer number (m) inserted, such as WGN_V_1_EVENT_2.

There is no significance to what RLS model the number m represents, although these details are stored internally, and is purely based on the order in which RLS model are created. The exception to this is for m=1. This is always the default model that has been set up.

In addition to these views will be a set of views that end in _RLS_n. These are identical to the m=1 views and were the view used by previous version of CA DataMinder. These views should be ignored as they will be deprecated at some point in the future. This view points to one of six views, usually given the same name as the top level view but with an additional _MD_, _MDX_, _MS_, _MSX_,_PD_, _PDX appended after the _V part of the name. Which one of these is dependent on which RLS model that is relevant.

For the purposes of this document the default model will be used _MD_ will be used as an example in most case, however for some views the _PD_ model will be used to demonstrate how this is applied particularly when the _MD_ simply points to the underlying table (such as no RLS is applied to the trigger view for any for the management group models. There are also 2 other sub-level view types: WGN_V_AD_... is the administrator view; and WGN_V_CM_... is an additional sub level that contains common views which do not apply RLS and are common across all models. These lower level views contain much of the detail of what and how data is returned. It contains tables, highlighted in italics, and additional views, which in turn reference additional tables.

It is important to note that these view stacks should not be taken as an exact reflection of what may be found in customer environments. This will change some of the underlying tables and views depending on which security model is used. Also from time to time the actual views may change, particularly the lower levels views, due to a schema change or a performance enhancement.

If searches are written using these primary views, any changes to underlying views should have no effect on the results returned. The only effect that would occur may be that the search may return results more quickly due to a performance improvement to an underlying view. If a schema change takes place it may be that additional views are included. Although a search should not fail using the old views it may be beneficial to change the search to use the new views, as it may improve performance and would guard any possible future deprecation of old views.

The table contains brief descriptions of the columns. For further details of see the Database Schema document.

Provided a user has the administrative privilege 'Admin: Disable Security Group Filtering' disabled, all views described below will have RLS applied with the exceptions of:

WGN_V_PARTPNTUSER_1

WGN_V_QUARANTINE_EVENT_1

WGN_V_GROUP_HIST_1

WGN_V_GROUP_HIST_2

WGN_V_CURR_PROP_VAL_1

WGN3USERADDRESS

WGN_V_USER_ADDRESS_1

WGN_V_ADDRESS_1

WGN_V_INTPARTPNTUSER_1

If using any of these views, RLS should be applied by an appropriate join to one of the other views.

WGN_V_RELATEDEVENT_1

This view provides an interface to the new table Wgn3RelatedEvent. RLS is not applied to this view.

Column	Primary Key	Datatype	Length	Notes
EventUID	Yes	IDENTITYREF	13	Foreign Key to Wgn3Event
EventTimestamp	Yes	TIMESTAMP		The time at which the event occurred.
RelationType	Yes	INTEGER	4	Type of relationship. Arbitrary Integer to allow multiple rows of the same "type" of relationship to exist
RelationIndex	Yes	INTEGER	4	Relationship increment.
RelatedEventUID		IDENTITYREF	13	Related Wgn3Event.EventUID record (Not constrained).
RelatedEventTimeStamp		TIMESTAMP		Related Wgn3Event.EventTimeStamp record (Not constrained).
ApplicationID		INTEGER	4	Identifier of the application that created the record

WGN_V_GROUP_1

This view returns all columns from the table WgnGroup along with the Parent Group ID's subject to RLS. This view should always be used when a search wants to retrieve information from WgnGroup. If a search attempts to retrieve rows directly from WgnGroup the search will fail as the search user will not have permission to select data from this table directly.

View Stack

wgn_v_group_1

→ wgn_v_mgroup_1

→ wgn_v_md_group_1

→ WGNGROUP

→ WGN_V_RLS_1

→ *TMP_WGN3RLS*

Column Details

Column	Data Type	Length	Description
GroupIDM	INTEGER	4	Part-key used to uniquely identify a group within the installation.
GroupID	INTEGER	4	Part-key used to uniquely identify a group within the installation.
GroupName	VARCHAR	64	The group name.
PolicyIDM	INTEGER	4	Part-key identifying the policy document for this user.
PolicyID	INTEGER	4	Part-key identifying the policy document for this user.
DeletedFlag	BIT	1	Indicates whether the user/group is deleted or not.
GroupCreateTime	TIMESTAMP	8	The time that the CA DataMinder group was created.
parentgroupidm	INTEGER	4	Part-key identifying the child groups parent
parentgroupid	INTEGER	4	Part-key identifying the child groups parent

WGN_V_USER_1

This view returns all columns from Wgn3User subject to RLS being applied. This view should always be used when a search wants to retrieve information from Wgn3User. If a search attempts to retrieve rows directly from Wgn3User the search will fail as the search user will not have permission to select data from this table directly.

View Stack

wgn_v_user_1
→ wgn_v_m_user_1
→ wgn_v_md_user_1
→ *WGN3USER*
WGN3USERGROUP
WGN_V_RLS_1
→ *TMP_WGN3RLS*

Column Details

Column	Data Type	Length	Description
UserIDM	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
UserID	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
UserName	VARCHAR	64	The name of the CA DataMinder user. The naming convention (if any) depends on the user creation policy settings.
PolicyIDM	INTEGER	4	Part-key identifying the policy document for this user.
PolicyID	INTEGER	4	Part-key identifying the policy document for this user.
UserCreateTime	TIMESTAMP	8	The time that the CA DataMinder user was created.
UserRole	INTEGER	4	The CA DataMinder role assigned to the user. This is used to determine the user's initial privileges.
DeletedFlag	BIT	1	Indicates whether the user/group is deleted or not.
UserData	VARBINARY	255	Salt used to hash the password.
PasswordHash	VARBINARY	255	A hashed copy of the user's password.
UserPrivileges	DECIMAL	13	A bitmap of CA DataMinder privileges the user is allowed to perform. The initial value is taken from the 'privilege profile' associated with the user's role.

WGN_V_USER_GROUP_1

This view returns all columns from WgnGroup, plus the parent group id's associated with each group, and all columns from Wgn3User subject to RLS being applied. This view should be used if you require both user and group information. It would be possible to get the same information from Wgn_V_User_1 and Wgn_V_Group_1, however you would apply RLS twice (as you are unable to reference the underlying table directly) if you choose this option. This view allows easy access to user and group information but only applies RLS once.

View Stack

wgn_v_user_group_1

→ wgn_v_m_user_group_1

→ wgn_v_md_user_group_1

→ wgn_v_md_group_1

→ *WGNGROUP*

WGN_V_RLS_1

→ *TMP_WGN3RLS*

WGN3USER

WGN3USERGROUP

Column Details

Column	Data Type	Length	Description
GroupIDM	INTEGER	4	Part-key used to uniquely identify a group within the installation.
GroupID	INTEGER	4	Part-key used to uniquely identify a group within the installation.
GroupName	VARCHAR	64	The group name.
GroupPolicyIDM	INTEGER	4	Part key identifying the groups policy document.
GroupPolicyID	INTEGER	4	Part key identifying the groups policy document.
GroupDeletedFlag	BIT	1	Indicates whether the group is deleted or not.
GroupCreateTime	TIMESTAMP	8	The time that the CA DataMinder group was created.
ParentGroupIDM	INTEGER	4	Part-key identifying the child group's parent
ParentGroupID	INTEGER	4	Part-key identifying the child group's parent
UserIDM	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
UserID	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
UserName	VARCHAR	64	The name of the CA DataMinder user. The naming convention (if any) depends on the user creation policy settings.
PolicyIDM	INTEGER	4	Part-key identifying the policy document for this user.
PolicyID	INTEGER	4	Part-key identifying the policy document for this user.
UserCreateTime	TIMESTAMP	8	The time that the CA DataMinder user was created.
UserRole	INTEGER	4	The CA DataMinder role assigned to the user. This is used to determine the user's initial privileges.
DeletedFlag	BIT	1	Indicates whether the user/group is deleted or not.
UserData	VARBINARY	255	Salt used to hash the password.
PasswordHash	VARBINARY	255	A hashed copy of the user's password.

Column	Data Type	Length	Description
UserPrivileges	DECIMAL	13	A bitmap of CA DataMinder privileges the user is allowed to perform. The initial value is taken from the 'privilege profile' associated with the user's role.

WGN_V_GROUP_HIST_1

This view returns all rows and most of the columns from Wgn3UserGroup. Additionally, this also returns the end date that this user is part of this group and the effective start date. This Effective start date is set to approximately the start date minus 100 years if this is the first group a user has belonged to or the actual start date otherwise.

It is possible on large databases, in both Oracle and SQL Server, that this view can produce poor query plans. In some of these cases it may be better to use WGN_V_RLS_ADDR_1 (see description of this view below) as it contains similar information and this is created once when the iConsole is started so it does not need to search through several tables. This view as now been superceded by the view WGN_V_GROUP_HIST_2 due to a schema change which as added the columns EffectStartDate and EffectEndDate to the underlying table to help improve performance. WGN_V_GROUP_HIST_2 should now be used.

View Stack

WGN_V_GROUP_HIST_1

→ WGN_V_CM_GROUP_HIST_1

→ WGN_V_GROUP_HIST_NULL_END_1

→ WGN_V_CM_GROUP_HIST_NULL_END_1

→ *WGN3USERGROUP*

Column Details

Column	Data Type	Length	Description
UserGroupUID	INTEGER	4	System-generated key that uniquely identifies a user group
StartDate	TIMESTAMP	8	The date at which the user became a member of the group.
UserIDM	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
UserID	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.

Column	Data Type	Length	Description
GroupIDM	INTEGER	4	Part-key used to uniquely identify a group within the installation.
GroupID	INTEGER	4	Part-key used to uniquely identify a group within the installation.
NextGroupHistoryUID	INTEGER	4	Used to explicitly link the group history in chronological order, so that an 'end date' can be derived from the start date of the next entry.
EndDate	TIMESTAMP	8	End date when a particular reviewer can view a particular user's CA DataMinder events.
EffectiveStartDate	TIMESTAMP	8	Start date when a particular reviewer can view a particular user's CA DataMinder events.

WGN_V_GROUP_HIST_2

This view returns all rows and columns from Wgn3UserGroup. Additionally, this also returns the end date that this user is part of this group.

View Stack

WGN_V_GROUP_HIST_2

→ WGN_V_CM_GROUP_HIST_2

→ WGN3USERGROUP

Column Details

Column	Data Type	Length	Description
UserGroupUID	INTEGER	4	System-generated key that uniquely identifies a user group
StartDate	TIMESTAMP	8	The date at which the user became a member of the group.
UserIDM	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
UserID	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
GroupIDM	INTEGER	4	Part-key used to uniquely identify a group within the installation.

Column	Data Type	Length	Description
UserGroupUID	INTEGER	4	System-generated key that uniquely identifies a user group
GroupID	INTEGER	4	Part-key used to uniquely identify a group within the installation.
NextGroupHistoryUID	INTEGER	4	Used to explicitly link the group history in chronological order, so that an 'end date' can be derived from the start date of the next entry.
EndDate	TIMESTAMP	8	End date when a particular reviewer can view a particular user's CA DataMinder events.
EffectiveStartDate	TIMESTAMP	8	Start date when a particular reviewer can view a particular user's CA DataMinder events.
EffectiveEndDate	TIMESTAMP	8	Effective end date when a particular reviewer can view a particular user's CA DataMinder events.

WGN_V_USER_HIST_1

This view returns all columns from Wgn3User similar to WGN_V_USER_1 subject to RLS being applied. WGN_V_USER_1 returns rows by strictly applying RLS so that only current users, subject to RLS, will be returned. WGN_V_USER_HIST_1 additionally returns records for users that were, historically, in the logged on user's management group but have since moved and would therefore not be seen via WGN_V_USER_1.

View Stack

WGN_V_USER_HIST_1

→ WGN_V_m_USER_HIST_1

→ wgn_v_md_user_hist_1

→ *WGN3USER*

WGN3USERGROUP

WGN_V_RLS_1

→ *TMP_WGN3RLS*

Column Details

Column	Data Type	Length	Description
UserIDM	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.

UserID	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
UserName	VARCHAR	64	The name of the CA DataMinder user. The naming convention (if any) depends on the user creation policy settings.
PolicyIDM	INTEGER	4	Part-key identifying the policy document for this user.
PolicyID	INTEGER	4	Part-key identifying the policy document for this user.
UserCreateTime	TIMESTAMP	8	The time that the CA DataMinder user was created.
UserRole	INTEGER	4	The CA DataMinder role assigned to the user. This is used to determine the user's initial privileges.
DeletedFlag	BIT	1	Indicates whether the user/group is deleted or not.
UserData	VARBINARY	255	Salt used to hash the password.
PasswordHash	VARBINARY	255	A hashed copy of the user's password.
UserPrivileges	DECIMAL	13	A bitmap of CA DataMinder privileges the user is allowed to perform. The initial value is taken from the 'privilege profile' associated with the user's role.

WGN_V_USER_GROUP_HIST_1

This view returns the same columns as WGN_V_USER_GROUP_1 subject to RLS being applied. WGN_V_USER_GROUP_1 returns rows by strictly applying RLS so that only current users, subject to RLS, will be returned. WGN_V_USER_GROUP_HIST_1 additionally returns records for users that were, historically, in the logged on user's management group but have since moved and would therefore not be seen via WGN_V_USER_GROUP_1.

View Stack

WGN_V_USER_GROUP_HIST_1

→ wgn_v_m_user_group_hist_1

→ wgn_v_md_user_group_hist_1

→ *WGN3USERGROUP*

WGN_V_RLS_1

→ *TMP_WGN3RLS*

WGN3USER

WGNGROUP

Column Details

Column	Data Type	Length	Description
GroupIDM	INTEGER	4	Part-key used to uniquely identify a group within the installation.
GroupID	INTEGER	4	Part-key used to uniquely identify a group within the installation.
GroupName	VARCHAR	64	The group name.
GroupPolicyIDM	INTEGER	4	Part-key identifying the policy document for this user.
GroupPolicyID	INTEGER	4	Part-key identifying the policy document for this user.
GroupDeletedFlag	BIT	1	Indicates whether the user/group is deleted or not.
GroupCreateTime	TIMESTAMP	8	The time that the CA DataMinder group was created.
parentgroupidm	INTEGER	4	Part-key identifying the child groups parent
parentgroupid	INTEGER	4	Part-key identifying the child groups parent
UserIDM	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
UserID	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
UserName	VARCHAR	64	The name of the CA DataMinder user. The naming convention (if any) depends on the user creation policy settings.
PolicyIDM	INTEGER	4	Part-key identifying the policy document for this user.
PolicyID	INTEGER	4	Part-key identifying the policy document for this user.
UserCreateTime	TIMESTAMP	8	The time that the CA DataMinder user was created.
UserRole	INTEGER	4	The CA DataMinder role assigned to the user. This is used to determine the user's initial privileges.
DeletedFlag	BIT	1	Indicates whether the user/group is deleted or not.
UserData	VARBINARY	255	Salt used to hash the password.
PasswordHash	VARBINARY	255	A hashed copy of the user's password.
UserPrivileges	DECIMAL	13	A bitmap of CA DataMinder privileges the user is allowed to perform. The initial value is taken from the 'privilege profile' associated with the user's role.

WGN_V_EVENT_1

This has mostly been superseded by WGN_V_EVENT_2. This has not been entirely deprecated yet but report writers are encouraged to use WGN_V_EVENT_2 as standard and only use this view in exceptional cases; see the description of this below.

View Stack

Wgn_V_Event_1

→ wgn_v_m_event_1

→ WGN_V_MD_EVENT_1

→ *wgn3event*

WGN_V_MD_PARTPNTUSER_1

→ WGN_V_CM_PARTPNTUSER_1

→ *Wgn3EventParticipant*

WGN_V_USER_ADDRESS_1

→ WGN_V_CM_USER_ADDRESS_1

→ *WGN3USERADDRESSEX*

WGN_V_RLS_1

→ *TMP_WGN3RLS*

WGN_V_GROUP_HIST_1

→ WGN_V_GROUP_HIST_NULL_END_1

→ *WGN3USERGROUP*

WGN_V_EVENT_2

This view and, with exceptions, WGN_V_EVENT_1, must always be used when a search wants to retrieve information from Wgn3Event. If a search attempts to retrieve rows directly from Wgn3Event, the search will fail because the search user does not have permission to select data from this table directly.

WGN_V_EVENT_1 removes duplicate rows from the view so that it only returns 1 row per event. WGN_V_EVENT_2 returns duplicate rows from the view so that it returns 1 row per participant. One row per event can then be achieved using some further SQL (such as DISTINCT). Here we have just moved the emphasis of where this de-duplication happens.

WGN_V_EVENT_2 is now the default view to use; it has essentially superseded WGN_V_EVENT_1. We strongly recommend that you use WGN_V_EVENT_2. This is primarily for performance reasons. While there is no current plan to deprecate WGN_V_EVENT_1, this may happen in the future. This is another reason for using WGN_V_EVENT_2. There may be some specific cases where WGN_V_EVENT_1 performs better, but this should only be tested if it is believed that WGN_V_EVENT_2 is causing a performance issue (for example, when retrieving events given the UID may perform better using WGN_V_EVENT_1).

The final column in either view, epeventtimestamp, was added because of performance issues, primarily on SQL Server 2000. This does not appear to be an issue in Oracle. The field reflects the same value as EventTimeStamp on Wgn3Event but on the table Wgn3EventParticipant. The problem is that SQL Server 2000 does not reliably perform predicate closure. It is highly likely that most searches will include a date range. These dates will be passed to the search as literal values. In these cases, it is important that these literal values are put against both EventTimeStamp and epEventTimeStamp, otherwise SQL Server does not recognize that the literal values apply to both columns from any join and this can produce a poor query plan.

Example

```
FROM WGN_V_EVENT_2 EV …
  WHERE EV.EventTimeStamp >= {start date} AND EV.EventTimeStamp <= {end date} …
  AND EV.epEventTimeStamp >= {start date} AND EV.epEventTimeStamp <= {end date} …
```

View Stack

Wgn_V_Event_2

→ wgn_v_m_event_2

→ WGN_V_MD_EVENT_2

→ *wgn3event*

wgn3eventparticipant

WGN_V_RLS_ADDR_1

→ *TMP_WGN3ADDRRLS*

Column Details

Column	Data Type	Length	Description
EventUID	IDENTITYDEF	13	Key used to uniquely identify a captured or imported event.
EventTimestamp	TIMESTAMP	8	The time at which the event occurred.
SequenceIDM	INTEGER	4	Part-key used to identify the owning sequence and also uniquely identify the event.
SequenceID	INTEGER	4	Part-key used to identify the owning sequence and also uniquely identify the event.
EventIndex	INTEGER	4	Part-key used to uniquely identify the event.
EventMajorType	INTEGER	4	Identifies the 'major' event type (such as, email or web page)
EventMinorType	INTEGER	4	Identifies the 'minor' event type (such as, email sent)
EventSubType	INTEGER	4	Identifies the 'direction' of the event (such as, incoming or outgoing)
AesMajorType	INTEGER	4	Part-key that identifies the source object (software package) that captured the sequence.
AesMinorType	INTEGER	4	Part-key that identifies the source object (software package) that captured the sequence.
AesSubType	INTEGER	4	Part-key that identifies the source object (software package) that captured the sequence.
EventText1	LONGVARCHAR	255	Event specific string identifying the event.
EventText2	LONGVARCHAR	4000	Event specific string identifying the event.
EventAttributes	VARCHAR	255	Attributes of the event.
ExternalID	VARCHAR	255	This attribute contains an external (non-CA DataMinder) unique ID of an event to enable matching with duplicates of the same event captured elsewhere.
Duration	INTEGER	4	The length of time for which the event occurred.
ExpiryTimestamp	TIMESTAMP	8	Stores the time at which the event expires. Expired events are removed by the next scheduled database purge.

Column	Data Type	Length	Description
IsPermanent	BIT	1	If this column has a non-zero (true) value, then the event will never be deleted by a scheduled database purge.
UpdateTimestamp	TIMESTAMP	8	The time that the event was last modified.
PurgeState	INTEGER	4	Indicates whether the row can be purged if 'Purge On Replicate' is enabled. This column is ignored on a CMS.
GMTOffset	INTEGER	4	Specifies the difference (in minutes) between UTC time and the time zone in which an events source machine resides.
DSTOffset	INTEGER	4	Specifies the size of the daylight saving offset (in minutes) for the source machine's timezone.
BlobType	INTEGER	4	Identifies the type of the blob (such as a file based blob).
BlobLocation	VARCHAR	255	The location of the blob data. The format of this string depends on the BlobType.
BlobSize	DECIMAL	13	The size of the blob (in bytes) before encryption/compression.
BLOBPhysicalSize	DECIMAL	13	The size of the blob (in bytes) after encryption/compression.
epeventtimestamp	TIMESTAMP	8	Same as EventTimeStamp but this is taken from Wgn3EventParticipants instead of Wgn3Event. Used to ensure predicates commute.

WGN_V_EVENTPARTPNTUSER_1

This view returns all columns from Wgn3Event along with participant, address and user information subject to RLS. Use this view when you require event and user information. This view allows easy access to event and user information and only applies RLS once.

Note: Because the underlying tables are not accessible, if this view were not available you would need to use the WGN_V_EVENT_2 and Wgn_V_User_1 views, which would mean that RLS was applied twice.

View Stack

WGN_V_EVENTPARTPNTUSER_1

→ wgn_v_m_eventpartpntuser_1

→ WGN_V_MD_EVENTPNTUSER_1

→ *WGN3Event*

Wgn3EventParticipant

WGN_V_RLS_ADDR_1

→ *TMP_WGN3ADDRRLS*

WGN3ADDRESS

Column Details

Column	Data Type	Length	Description
EventUID	IDENTITYDEF	13	Key used to uniquely identify a captured or imported event.
EventTimestamp	TIMESTAMP	8	The time at which the event occurred.
SequenceIDM	INTEGER	4	Part-key used to identify the owning sequence and also uniquely identify the event.
SequenceID	INTEGER	4	Part-key used to identify the owning sequence and also uniquely identify the event.
EventIndex	INTEGER	4	Part-key used to uniquely identify the event.
EventMajorType	INTEGER	4	Identifies the 'major' event type (such as, email or web page)
EventMinorType	INTEGER	4	Identifies the 'minor' event type (such as, email sent)
EventSubType	INTEGER	4	Identifies the 'direction' of the event (such as, incoming or outgoing)
AesMajorType	INTEGER	4	Part-key that identifies the source object (software package) that captured the sequence.
AesMinorType	INTEGER	4	Part-key that identifies the source object (software package) that captured the sequence.
AesSubType	INTEGER	4	Part-key that identifies the source object (software package) that captured the sequence.
EventText1	LONGVARCHAR	255	Event specific string identifying the event.
EventText2	LONGVARCHAR	4000	Event specific string identifying the event.
EventAttributes	VARCHAR	255	Attributes of the event.
ExternalID	VARCHAR	255	This attribute contains an external (non-CA DataMinder) unique ID of an event to enable matching with duplicates of the same event captured elsewhere.

Column	Data Type	Length	Description
Duration	INTEGER	4	The length of time for which the event occurred.
ExpiryTimestamp	TIMESTAMP	8	Stores the time at which the event expires. Expired events are removed by the next scheduled database purge.
IsPermanent	BIT	1	If this column has a non-zero (true) value, then the event will never be deleted by a scheduled database purge.
UpdateTimestamp	TIMESTAMP	8	The time that event was last written to the DBMS.
PurgeState	INTEGER	4	Indicates whether the row can be purged if 'Purge On Replicate' is enabled. This column is ignored on a CMS.
GMTOffset	INTEGER	4	Specifies the difference (in minutes) between UTC time and the time zone in which an events source machine resides.
DSTOffset	INTEGER	4	Specifies the size of the daylight saving offset (in minutes) for the source machine's timezone.
BlobType	INTEGER	4	Identifies the type of the blob (such as a file based blob).
BlobLocation	VARCHAR	255	The location of the blob data. The format of this string depends on the BlobType.
BlobSize	DECIMAL	13	The size of the blob (in bytes) before encryption/compression.
BLOBPhysicalSize	DECIMAL	13	The size of the blob (in bytes) after encryption/compression.
ParticipantIndex	INTEGER	4	Part-key used to identify an event participant.
AddressUID1	DECIMAL	13	Part-key used to uniquely identify an email address or the address of an IM participant.
AddressUID2	DECIMAL	13	Part-key used to uniquely identify an email address or the address of an IM participant.
ParticipantNodeUID	INTEGER	4	The classification of the participant.
InterventionNodeUID	INTEGER	4	The classification that CA DataMinder assigns to the participant.
LoginIDM	INTEGER	4	Part-key that indicates the participant who generated the event.
LoginID	INTEGER	4	Part-key that indicates the participant who generated the event.
AddressName	VARCHAR	255	The actual email address or IM participant.

Column	Data Type	Length	Description
NativeUser	VARCHAR	128	Name of the OS user that the CA DataMinder is logged on as.
LoginState	INTEGER	4	The state of the user login, that is, logged in or out.
LastLogonTime	TIMESTAMP	8	The time the user last logged in or out.
PolicyVersion	VARCHAR	255	The version of user policy that is being used.
SourceMachine	VARCHAR	255	The name of the machine on which the user logged in.
DefaultLoginFlag	BIT	1	This is the 'default' login record for the OS user. This is used to control automatic logins.
UserIDM	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
UserID	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
epeventtimestamp	TIMESTAMP	8	Same as EventTimeStamp but this is taken from Wgn3EventParticipants instead of Wgn3Event. Used to ensure predicates commute.

WGN_V_EVENTPARTUSERGRP_1

This view returns all columns from Wgn3Event along with participant, user and group information subject to RLS. Use this view when you require event, user and group information. This view allows easy access to event, user and group information but only applies RLS once.

Note: Because the underlying tables are not accessible, if this view were not available you would need to use the WGN_V_EVENT_2, Wgn_V_User_1 and Wgn_V_Group_1 views, which would mean that RLS was applied three times.

View Stack

WGN_V_EVENTPARTUSERGRP_1

→ WGN_V_m_EVENTPARTUSERGRP_1

→ WGN_V_MD_EVENTPNTUSRGR_1

→ *WGN3Event*

wgn3eventparticipant

WGN_V_RLS_ADDR_1

→ *TMP_WGN3ADDRRLS*

Column Details

Column	Data Type	Length	Description
EventUID	IDENTITYDEF	13	Key used to uniquely identify a captured or imported event.
EventTimestamp	TIMESTAMP	8	The time at which the event occurred.
SequenceIDM	INTEGER	4	Part-key used to identify the owning sequence and also uniquely identify the event.
SequenceID	INTEGER	4	Part-key used to identify the owning sequence and also uniquely identify the event.
EventIndex	INTEGER	4	Part-key used to uniquely identify the event.
EventMajorType	INTEGER	4	Identifies the 'major' event type (such as, email or web page)
EventMinorType	INTEGER	4	Identifies the 'minor' event type (such as, email sent)
EventSubType	INTEGER	4	Identifies the 'direction' of the event (such as, incoming or outgoing)
AesMajorType	INTEGER	4	Part-key that identifies the source object (software package) that captured the sequence.
AesMinorType	INTEGER	4	Part-key that identifies the source object (software package) that captured the sequence.
AesSubType	INTEGER	4	Part-key that identifies the source object (software package) that captured the sequence.
EventText1	LONGVARCHAR	255	Event specific string identifying the event.
EventText2	LONGVARCHAR	4000	Event specific string identifying the event.
EventAttributes	VARCHAR	255	Attributes of the event.
ExternalID	VARCHAR	255	This attribute contains an external (non-CA DataMinder) unique ID of an event to enable matching with duplicates of the same event captured elsewhere.
Duration	INTEGER	4	The length of time for which the event occurred.
ExpiryTimestamp	TIMESTAMP	8	Stores the time at which the event expires. Expired events are removed by the next scheduled database purge.
IsPermanent	BIT	1	If this column has a non-zero (true) value, then the event will never be deleted by a scheduled database purge.

Column	Data Type	Length	Description
EventUID	IDENTITYDEF	13	Key used to uniquely identify a captured or imported event.
UpdateTimestamp	TIMESTAMP	8	The time that event was last written to the DBMS.
PurgeState	INTEGER	4	Indicates whether the row can be purged if 'Purge On Replicate' is enabled. This column is ignored on a CMS.
GMTOffset	INTEGER	4	Specifies the difference (in minutes) between UTC time and the time zone in which an events source machine resides.
DSTOffset	INTEGER	4	Specifies the size of the daylight saving offset (in minutes) for the source machine's timezone.
BlobType	INTEGER	4	Identifies the type of the blob (such as a file based blob).
BlobLocation	VARCHAR	255	The location of the blob data. The format of this string depends on the BlobType.
BlobSize	DECIMAL	13	The size of the blob (in bytes) before encryption/compression.
BLOBPhysicalSize	DECIMAL	13	The size of the blob (in bytes) after encryption/compression.
ParticipantIndex	INTEGER	4	Part-key used to identify an event participant.
AddressUID1	DECIMAL	13	Part-key used to uniquely identify an email address or the address of an IM participant.
AddressUID2	DECIMAL	13	Part-key used to uniquely identify an email address or the address of an IM participant.
ParticipantNodeUID	INTEGER	4	The classification of the participant.
InterventionNodeUID	INTEGER	4	The classification that CA DataMinder assigns to the participant.
LoginIDM	INTEGER	4	Part-key that indicates the participant who generated the event.
LoginID	INTEGER	4	Part-key that indicates the participant who generated the event.
NativeUser	VARCHAR	128	Name of the OS user that the CA DataMinder is logged on as.
LoginState	INTEGER	4	The state of the user login, that is, logged in or out.
LastLogonTime	TIMESTAMP	8	The time the user last logged in or out.

Column	Data Type	Length	Description
EventUID	IDENTITYDEF	13	Key used to uniquely identify a captured or imported event.
PolicyVersion	VARCHAR	255	The version of user policy that is being used.
SourceMachine	VARCHAR	255	The name of the machine on which the user logged in.
DefaultLoginFlag	BIT	1	This is the 'default' login record for the OS user. This is used to control automatic logins.
UserIDM	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
UserID	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
GroupIDM	INTEGER	4	Part-key used to uniquely identify a group within the installation.
GroupID	INTEGER	4	Part-key used to uniquely identify a group within the installation.
epEventTimeStamp	TIMESTAMP	8	Same as EventTimeStamp but this is taken from Wgn3EventParticipants instead of Wgn3Event. Used to ensure predicates commute.

WGN_V_PARTPNTUSER_1

This view provides a link from the participant table to the users. RLS is not applied to this view but can be achieved by an appropriate join to Wgn_V_User_1 or Wgn_V_User_Group_1. It was not designed as a top level view but was used as a lower level view to link other tables or views to avoid multiple applications of RLS. However most of the views which use this view have now been re-written to use WGN_V_PNTUSER_1 instead.

View Stack

WGN_V_PARTPNTUSER_1

→ WGN_V_m_PARTPNTUSER_1

→ WGN_V_MD_PARTPNTUSER_1

→ WGN_V_CM_PARTPNTUSER_1

→ *Wgn3EventParticipant*

WGN_V_USER_ADDRESS_1

→ WGN_V_CM_USER_ADDRESS_1

→ *WGN3USERADDRESSEX*

Column Details

Column	Data Type	Length	Description
eventuid	IDENTITYDEF	13	Key used to uniquely identify a captured or imported event.
ParticipantIndex	INTEGER	4	Part-key used to identify an event participant.
eventtimestamp	TIMESTAMP	8	The time at which the event occurred.
AddressUID1	DECIMAL	13	Part-key used to uniquely identify an email address or the address of an IM participant.
AddressUID2	DECIMAL	13	Part-key used to uniquely identify an email address or the address of an IM participant.
ParticipantNodeUID	INTEGER	4	The classification of the participant.
InterventionNodeUID	INTEGER	4	The classification that CA DataMinder assigns to the participant.
LoginIDM	INTEGER	4	Part-key that indicates the participant who generated the event.
LoginID	INTEGER	4	Part-key that indicates the participant who generated the event.
NativeUser	VARCHAR	128	Name of the OS user that the CA DataMinder is logged on as.
LoginState	INTEGER	4	The state of the user login, that is, logged in or out.
LastLogonTime	TIMESTAMP	8	The time the user last logged in or out.
PolicyVersion	VARCHAR	255	The version of user policy that is being used.
SourceMachine	VARCHAR	255	The name of the machine on which the user logged in.
DefaultLoginFlag	BIT	1	This is the 'default' login record for the OS user. This is used to control automatic logins.
useridm	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
userid	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
IsRLSCandidate	INTEGER	4	This is always set to 1

WGN_V_INTPARTPNTUSER_1

This view provides a link from the participant table to the users. RLS is not applied to this view but can be achieved by an appropriate join to another RLS view. It was developed as a performance improvement to WGN_V_PARTPNTUSER_1 which returns all participants by use of an outer join. This view uses an inner join and only returns internal participants (users). As with WGN_V_PARTPNTUSER_1 it was not designed as a top level view but was used as a lower level view to link other tables or views to avoid multiple applications of RLS.

View Stack

WGN_V_INTPARTPNTUSER_1

→ WGN_V_CM_INTPARTPNTUSER_1

→ *wgn3eventparticipant*

WGN_V_USER_ADDRESS_1

→ WGN_V_CM_USER_ADDRESS_1

→ *WGN3USERADDRESSEX*

Column Details

Column	Data Type	Length	Description
eventuid	IDENTITYDEF	13	Key used to uniquely identify a captured or imported event.
ParticipantIndex	INTEGER	4	Part-key used to identify an event participant.
eventtimestamp	TIMESTAMP	8	The time at which the event occurred.
AddressUID1	DECIMAL	13	Part-key used to uniquely identify an email address or the address of an IM participant.
AddressUID2	DECIMAL	13	Part-key used to uniquely identify an email address or the address of an IM participant.
ParticipantNodeUID	INTEGER	4	The classification of the participant.
InterventionNodeUID	INTEGER	4	The classification that CA DataMinder assigns to the participant.
LoginIDM	INTEGER	4	Part-key that indicates the participant who generated the event.
LoginID	INTEGER	4	Part-key that indicates the participant who generated the event.
NativeUser	VARCHAR	128	Name of the OS user that the CA DataMinder is logged on as.

Column	Data Type	Length	Description
LoginState	INTEGER	4	The state of the user login, that is, logged in or out.
LastLogonTime	TIMESTAMP	8	The time the user last logged in or out.
PolicyVersion	VARCHAR	255	The version of user policy that is being used.
SourceMachine	VARCHAR	255	The name of the machine on which the user logged in.
DefaultLoginFlag	BIT	1	This is the 'default' login record for the OS user. This is used to control automatic logins.
useridm	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
userid	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.

WGN_V_PNTUSER_1

This view returns all columns from Wgn3EventParticipant along with those relating to login information, user IDs and group IDs. Additionally the EventMajorType is derived from the ParticipantNodeUID column on the Wgn3EventParticipant, thus eliminating the need to join to Wgn3Event within the view. This view should be used when you require participant, user and group information and does not require any other event attributes. Since there is no join to Wgn3Event in this view, this returns rows quicker if event information is not needed and could be used instead of either WGN_V_EVENTPARTPNTUSER_1 or WGN_V_EVENTPARTUSERGRP_1.

View Stack

WGN_V_PNTUSER_1

→ WGN_V_m_PNTUSER_1

→ WGN_V_MD_PNTUSER_1

→ *wgn3eventparticipant*

WGN_V_RLS_ADDR_1

→ *TMP_WGN3ADDRRLS*

Column Details

Column	Data Type	Length	Description
eventuid	IDENTITYDEF	13	Key used to uniquely identify a captured or imported event.

Column	Data Type	Length	Description
eventtimestamp	TIMESTAMP	8	The time at which the event occurred.
ParticipantIndex	INTEGER	4	Part-key used to identify an event participant.
AddressUID1	DECIMAL	13	Part-key used to uniquely identify an email address or the address of an IM participant.
AddressUID2	DECIMAL	13	Part-key used to uniquely identify an email address or the address of an IM participant.
ParticipantNodeUID	INTEGER	4	The classification of the participant.
InterventionNodeUID	INTEGER	4	The classification that CA DataMinder assigns to the participant.
LoginIDM	INTEGER	4	Part-key that indicates the participant who generated the event.
LoginID	INTEGER	4	Part-key that indicates the participant who generated the event.
NativeUser	VARCHAR	128	Name of the OS user that the CA DataMinder is logged on as.
LoginState	INTEGER	4	The state of the user login, that is, logged in or out.
LastLogonTime	TIMESTAMP	8	The time the user last logged in or out.
PolicyVersion	VARCHAR	255	The version of user policy that is being used.
SourceMachine	VARCHAR	255	The name of the machine on which the user logged in.
DefaultLoginFlag	BIT	1	This is the 'default' login record for the OS user. This is used to control automatic logins.
useridm	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
userid	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
groupidm	INTEGER	4	Part-key used to uniquely identify a group within the installation.
groupid	INTEGER	4	Part-key used to uniquely identify a group within the installation.
eventmajortype	INTEGER	4	Identifies the 'major' event type (such as, email or web page)

WGN_V_PNTUSER_2

This view returns all columns as WGN_V_PNTUSER_1 and from the point of view of Management group RLS it is identical to that view. However for the administrator view of this some of the joins have been changed from INNER to LEFT joins to be able to return rows that do not necessarily have groups associated to them

View Stack

WGN_V_PNTUSER_2

→ WGN_V_m_PNTUSER_2

→ WGN_V_MD_PNTUSER_2

→ WGN_V_MD_PNTUSER_1

→ wgn3eventparticipant

WGN_V_RLS_ADDR_1

→ TMP_WGN3ADDRRLS

WGN_V_ISSUE_1

This view was used to reflect an interface of the table Wgn3EventIssue prior to version 4.0 SP4, 4.5 SP2, and 4.7 of CA DataMinder.

It returns one row per issue. With these releases came a schema change where a new table was introduced (Wgn3IssueParticipant). It is in this table that the field ParticipantIndex now exists post these releases. This view has been maintained to allow backwards compatibility. Since the releases mentioned above this view has been superseded by two views: WGN_V_ISSUE_2 and WGN_V_CURR_ISSUE_PARTPNT_1. WGN_V_ISSUE_1 has been deprecated and should not be used post these releases as it arbitrarily chooses one participant for the issue which could result in unpredictable behaviour.

View Stack

Wgn_V_Issue_1

→ WGN_V_m_ISSUE_1

→ WGN_V_MD_ISSUE_1

→ *wgn3eventissue*

wgn3issueparticipan t

WGN_V_MD_ISSUE_PARTPNT_1

→ WGN_V_MD_PARTPNTUSER_1

→ WGN_V_CM_PARTPNTUSER_1

→ *Wgn3EventParticipant*

WGN_V_USER_ADDRESS_1

→ WGN_V_CM_USER_ADDRESS_1

→ *WGN3USERADDRESSEX*

wgn_v_md_user_1

→ *WGN3USER*

WGN3USERGROUP

WGN_V_RLS_1

→ *TMP_WGN3RLS*

WGN_V_CURR_PROP_VAL_1

→ WGN_V_CM_CURR_PROP_VAL_1

→ *WGN3USERPROPERTYVALUE*

wgn3address

Column Details

Column	Data Type	Length	Description
eventuid	IDENTITYDEF	13	Key used to uniquely identify a captured or imported event.
eventtimestamp	TIMESTAMP	8	The time at which the event occurred.
issueUID	IDENTITYDEF	13	Key used to uniquely identify an event issue.
issuename	VARCHAR	255	The name of the issue, as specified in the iConsole or Data Management console.
participantindex	INTEGER	4	Part-key used to identify an event participant.

WGN_V_ISSUE_2

This view reflects an interface of the table Wgn3EventIssue with RLS applied and returns 1 row per issue. For these releases the column ParticipantIndex was removed and put into a new table Wgn3IssueParticipant as part of a schema change to allow an issue to be associated with zero, one or many participants. It should be used wherever WGN_V_ISSUE_1 was previously used and the ParticipantIndex column is NOT required. If you want to see the field ParticipantIndex then this can be done using the view WGN_V_CURR_ISSUE_PARTICIPANT_1 instead. Do NOT revert back to using WGN_V_ISSUE_1 this has been deprecated post these release and is subject to performance problems.

View Stack

Wgn_V_Issue_2

→ WGN_V_m_ISSUE_2

→ WGN_V_MD_ISSUE_2

→ *WGN3EVENTISSUE*

WGN3ISSUEPARTICIPANT

WGN_V_MD_PNTUSER_1

→ *wgn3eventparticipant*

WGN_V_RLS_ADDR_1

→ *TMP_WGN3ADDRRLS*

Column Details

Column	Data Type	Length	Description
IssueUID	IDENTITYDEF	13	Key used to uniquely identify an event issue.
IssueName	VARCHAR	255	The name of the issue, as specified in the iConsole or Data Management console.
EventUID	IDENTITYDEF	13	Key used to uniquely identify a captured or imported event.
EventTimestamp	TIMESTAMP	8	The time at which the event occurred.

WGN_V_CURR_ISSUE_PARTPNT_1

This view was created as the results of a schema changes which introduced the table Wgn3IssuePaticipant. This is returns the same columns as WGN_V_ISSUE_1 but it returns 1 row per participant for all issues with RLS applied. It should be used wherever WGN_V_ISSUE_1 was previously used and the ParticipantIndex column is required.

Important: Do **not** revert back to using WGN_V_ISSUE_1. This view has been deprecated and has performance problems.

View Stack

WGN_V_CURR_ISSUE_PARTPNT_1

→ WGN_V_m_CURR_ISSUE_PARTPNT_1

→ WGN_V_MD_CURR_ISSUE_PARTPNT_1

→ *wgn3eventissue*

WGN3ISSUEPARTICIPANT

WGN_V_MD_PNTUSER_1

→ *wgn3eventparticipant*

WGN_V_RLS_ADDR_1

→ *TMP_WGN3ADDRRLS*

Column Details

Column	Data Type	Length	Description
issueuid	IDENTITYDEF	13	Key used to uniquely identify an event issue.
issuename	VARCHAR	255	The name of the issue, as specified in the iConsole or Data Management console.
eventuid	IDENTITYDEF	13	Key used to uniquely identify a captured or imported event.
eventtimestamp	TIMESTAMP	8	The time at which the event occurred.
participantindex	INTEGER	4	Part-key used to identify an event participant.

WGN_V_ISSUE_PARTPNT_1

This view has now been superseded by WGN_V_ISSUE_PARTPNT_2. It is retained for backwards compatibility.

View Stack

WGN_V_ISSUE_PARTPNT_1

→ WGN_V_m_ISSUE_PARTPNT_1

→ WGN_V_MD_ISSUE_PARTPNT_1

→ WGN_V_MD_PARTPNTUSER_1

→ WGN_V_CM_PARTPNTUSER_1

→ *Wgn3EventParticipant*

WGN_V_USER_ADDRESS_1

→ WGN_V_CM_USER_ADDRESS_1

→ *WGN3USERADDRESSEX*

wgn_v_md_user_1

→ *WGN3USER*

WGN3USERGROUP

WGN_V_RLS_1

→ *TMP_WGN3RLS*

WGN_V_CURR_PROP_VAL_1

→ WGN_V_CM_CURR_PROP_VAL_1

→ *WGN3USERPROPERTYVALUE*

wgn3address

More information:

WGN_V_ISSUE_PARTPNT_2 (see page 163)

WGN_V_ISSUE_PARTPNT_2

This view returns most of the columns from Wgn3EventParticipant along with AddressName, UserName and the user's Full Name, subject to RLS. This view was not specifically designed for searching, but for determining which participants can be seen by a user when auditing events. So this view can be used when you want to see which participants a user has access to. This view gives some extra information (such as FullName) but apart from that you can obtain the same information from WGN_V_PNTUSER_1.

View stack

WGN_V_ISSUE_PARTPNT_2

→ WGN_V_m_ISSUE_PARTPNT_2

→ WGN_V_MD_ISSUE_PARTPNT_2

→ WGN_V_MD_ISSUE_PARTPNT_3

→ *wgn3address*

WGN_V_CURR_PROP_VAL_1

→ WGN_V_CM_CURR_PROP_VAL_1

→ *WGN3USERPROPERTYVALUE*

WGN_V_MD_PNTUSER_1

→ *wgn3eventparticipant*

WGN_V_RLS_ADDR_1

→ *TMP_WGN3ADDRRLS*

Wgn3User

Column Details

Column	Data Type	Length	Description
eventuid	IDENTITYDEF	13	Key used to uniquely identify a captured or imported event.
eventtimestamp	TIMESTAMP	8	The time at which the event occurred.
participantindex	INTEGER	4	Part-key used to identify an event participant.
participanttype	INTEGER	4	The classification of the participant.
interventiontype	INTEGER	4	The classification that CA DataMinder assigns to the participant.

Column	Data Type	Length	Description
loginidm	INTEGER	4	Part-key that indicates the participant who generated the event.
loginid	INTEGER	4	Part-key that indicates the participant who generated the event.
address	VARCHAR	255	The actual email address or IM participant.
useridm	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
userid	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
username	VARCHAR	64	The name of the CA DataMinder user. The naming convention (if any) depends on the user creation policy settings.
userfullname	VARCHAR	255	User's Full Name

WGN_V_ISSUE_PARTPNT_3

This view is the same has WGN_V_ISSUE_PARTPNT_2 but included the additional column 'DeletedFlag'.

View Stack

WGN_V_ISSUE_PARTPNT_3

→ WGN_V_m_ISSUE_PARTPNT_3

→ WGN_V_MD_ISSUE_PARTPNT_3

→ *wgn3address*

WGN_V_CURR_PROP_VAL_1

→ WGN_V_CM_CURR_PROP_VAL_1

→ *WGN3USERPROPERTYVALUE*

WGN_V_MD_PNTUSER_1

→ *wgn3eventparticipant*

WGN_V_RLS_ADDR_1

→ *TMP_WGN3ADDRRLS*

Wgn3User

Column Details

Column	Data Type	Length	Description
eventuid	IDENTITYDEF	13	Key used to uniquely identify a captured or imported event.
eventtimestamp	TIMESTAMP	8	The time at which the event occurred.
participantindex	INTEGER	4	Part-key used to identify an event participant.
participanttype	INTEGER	4	The classification of the participant.
interventiontype	INTEGER	4	The classification that CA DataMinder assigns to the participant.
loginidm	INTEGER	4	Part-key that indicates the participant who generated the event.
loginid	INTEGER	4	Part-key that indicates the participant who generated the event.
address	VARCHAR	255	The actual email address or IM participant.
useridm	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
userid	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
username	VARCHAR	64	The name of the CA DataMinder user. The naming convention (if any) depends on the user creation policy settings.
userfullname	VARCHAR	255	User's Full Name
DeletedFlag	BIT	1	Indicates whether the user account is deleted or not.

WGN_V_TRIGGER_1

This view returns all columns from Wgn3Trigger subject to RLS being applied. This view has been created to support policy based RLS. The view stack below show the Policy (standard) model as the management group model simply point to the underlying table.

View Stack

WGN_V_TRIGGER_1

→ WGN_V_m_TRIGGER_1

→ WGN_V_PD_TRIGGER_1

→ Wgn3Trigger

WGN_V_rls_POLICY_1

→ TMP_WGN3POLICYRLS

Column Details

Column	Data Type	Length	Description
eventuid	IDENTITYDEF	13	Key used to uniquely identify a captured or imported event.
TriggerIndex	INTEGER	4	Part-key used to uniquely identify this trigger.
eventtimestamp	TIMESTAMP	8	The time at which the event occurred.
TriggerType	INTEGER	4	The type of trigger.
TriggerName	VARCHAR	255	The name of the trigger.
TriggerAttributes	VARCHAR	255	Attributes of the trigger.
TriggerText1	LONGVARCHAR	13	Trigger type specific text. Contains the most significant definition from policy that matched
TriggerText2	LONGVARCHAR	13	Trigger type specific text. Contains the extracted data from the event that matched the policy definition of the trigger
ActionName	VARCHAR	255	Name of the action that the trigger fired.
ActionAttributes	VARCHAR	255	Attributes of the action fired by the trigger.
ActionType	VARCHAR	255	Type of action fired by the trigger.
PolicyID	INTEGER	4	Identifies the policy classification associated with the trigger.
Severity	INTEGER	4	Indicates the severity value of the trigger.

WGN_V_CURR_ISSUE_TRIGGER_1

This view returns all columns from Wgn3IssueTrigger and Wgn3Trigger subject to RLS being applied. Specifically it shows which trigger(s) the issue is associated to. This view has been created to support policy based RLS. The view stack below show the Policy (standard) model as the management group model simply point to the underlying tables.

View Stack

WGN_V_CURR_ISSUE_TRIGGER_1

→ WGN_V_m_CURR_ISSUE_TRIGGER_1

→ WGN_V_PD_CURR_ISSUE_TRIGGER_1

→ Wgn3IssueTrigger

WGN_V_PD_TRIGGER_1

→ Wgn3Trigger

WGN_V_rls_POLICY_1

→ TMP_WGN3POLICYRLS

Column Details

Column	Data Type	Length	Description
Issueuid	IDENTITYDEF	13	Key used to uniquely identify an event issue.
eventuid	IDENTITYDEF	13	Key used to uniquely identify a captured or imported event.
eventtimestamp	TIMESTAMP	8	The time at which the event occurred.
TriggerIndex	INTEGER	4	Part-key used to uniquely identify this trigger.
TriggerType	INTEGER	4	The type of trigger.
TriggerName	VARCHAR	255	The name of the trigger.
TriggerAttributes	VARCHAR	255	Attributes of the trigger.
TriggerText1	LONGVARCHAR	13	Trigger type specific text. Contains the most significant definition from policy that matched

Column	Data Type	Length	Description
Issueuid	IDENTITYDEF	13	Key used to uniquely identify an event issue.
TriggerText2	LONGVARCHAR	13	Trigger type specific text. Contains the extracted data from the event that matched the policy definition of the trigger
ActionName	VARCHAR	255	Name of the action that the trigger fired.
ActionAttributes	VARCHAR	255	Attributes of the action fired by the trigger.
ActionType	VARCHAR	255	Type of action fired by the trigger.
PolicyID	INTEGER	4	Identifies the policy classification associated with the trigger.
Severity	INTEGER	4	Indicates the severity value of the trigger.

WGN_V_POLICY_PICKER_1

This view returns NodeUID/ParentNodeUID pairs relating to Policies and classes from the Classification tables along with the associated Classname's. It is primarily used for a policy picker where a policy based RLS user should not be able to choose policies they cannot review. While this view may not be extensively used in most report queries it could be used if, for example, a custom policy picker was required.

View Stack

WGN_V_POLICY_PICKER_1

→ WGN_V_m_POLICY_PICKER_1

→ WGN_V_PD_POLICY_PICKER_1

→ WGN_V_AD_POLICY_PICKER_1

→ WGN_V_AD_POLICY_CLASS

→ Wgn3ClassificationNode

Wgn3Classification

WGN_V_rls_POLICY_1

→ TMP_WGN3POLICYRLS

Column Details

Column	Data Type	Length	Description
lvl	INTEGER	13	Hierarchy level of the policy-class hierarchy tree
policynodeuid	INTEGER	13	NodeUID of the policy-class hierarchy
parentpolicynodeuid	INTEGER	13	Parent NodeUID of policynodeuid
policyclassname	VARCHAR	255	Policy-class name relating to policynodeuid
parentpolicyclassname	VARCHAR	255	Policy-class name relating to parentpolicynodeuid

WGN_V_QUARANTINE_EVENT_1

This view returns the EventUID and EventTimeStamp for events that have been quarantined but have not been actioned as yet. RLS is not applied to this view as it is unlikely to be used in isolation. RLS can be applied by joining it to an appropriate view in any query.

View Stack

WGN_V_QUARANTINE_EVENT_1

→ *Wgn3EA*

Wgn3EventAudit

Column Details

Column	Data Type	Length	Description
EventUID	IDENTITYDEF	13	Key used to uniquely identify a captured or imported event.
EventTimestamp	TIMESTAMP	8	The time at which the event occurred.

WGN_V_CURR_PROP_VAL_1

This view returns the property values and associated information from the table Wgn3UserPropertyValue. This table does not overwrite any old values; it just inserts new ones. This view just displays the current property values and does not display old values.

View Stack

WGN_V_CURR_PROP_VAL_1

→ WGN_V_CM_CURR_PROP_VAL_1

→ *WGN3USERPROPERTYVALUE*

Column Details

Column	Data Type	Length	Description
USERIDM	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
USERID	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
UserPropIndex	INTEGER	4	Part-key that uniquely identifies the user with an attribute.
UserPropValue	VARCHAR	255	The Actual Property Value
PropertyID	INTEGER	4	Identifies the Property Type
DateModified	TIMESTAMP	8	The date at which the property was last modified.

WGN3USERADDRESS

This view does not conform to the usual naming standard. It was created for compatibility reasons because of a small schema change in CA DataMinder 12.0. We recommend that reports use the Wgn_V_User_Address_1 view instead of this view. RLS is not applied to this view.

Previous to 12.0, Wgn3UserAddress was a table instead of a view. In order to removed the dual access path (the second path was via the WgnUserLogin table), each user login that generates an event (such as Web events) needed to be given an address. This required an extra column in the table. Creating this extra column had compatibility issues. Therefore, table Wgn3UserAddressEx was created for this and the old table became a view which returns the same data that the original would have returned.

View Stack

WGN3USERADDRESS

→ *WGN3USERADDRESSEX*

Column Details

Column	Data Type	Length	Description
UserIDM	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
UserID	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
AddressUID1	DECIMAL	13	Part-key used to uniquely identify an email address or the address of an IM participant.
AddressUID2	DECIMAL	13	Part-key used to uniquely identify an email address or the address of an IM participant.

WGN_V_USER_ADDRESS_1

This view provides an interface to the new table Wgn3UserAddressEx. Use this view in preference to Wgn3UserAddress. RLS is not applied to this view.

View Stack

WGN_V_USER_ADDRESS_1

→ WGN_V_CM_USER_ADDRESS_1

→ *WGN3USERADDRESSEX*

Column Details

Column	Data Type	Length	Description
UserIDM	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
UserID	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
AddressUID1	DECIMAL	13	Part-key used to uniquely identify an email address or the address of an IM participant.
AddressUID2	DECIMAL	13	Part-key used to uniquely identify an email address or the address of an IM participant.
AddressType	INTEGER	4	This is the address type which differentiates the address become the original normal addresses and the new auto generated addresses

WGN_V_ADDRESS_1

This view provides an interface to the new table Wgn3Address with the userid's also included. The view only returns addresses for event types that have a natural address (email, IM, file). It does not return artificial auto generated addresses or addresses with no associated user. RLS is not applied to this view.

View Stack

WGN_V_ADDRESS_1

→ WGN_V_CM_ADDRESS_1

→ *wgn3address*

wgn3useraddressex

Column Details

Column	Data Type	Length	Description
UserIDM	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
UserID	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.

AddressUID1	DECIMAL	13	Part-key used to uniquely identify an email address or the address of an IM participant.
AddressUID2	DECIMAL	13	Part-key used to uniquely identify an email address or the address of an IM participant.
AddressName	VARCHAR	255	The actual email address or IM participant.

WGN_V_RLS_1

This view references one of the underlying RLS tables. When a search user connects to the CA DataMinder database, this view populates the underlying table to this view with the current group hierarchy that the user is allowed to view. This view is typically used more in the Administration Console. For more verbose information, use WGN_V_RLS_ADDR_1.

Note: This table is not populated for admin users and can return zero rows in a report if used by an admin user.

View Stack

WGN_V_RLS_1

→ *TMP_WGN3RLS*

Column Details

Column	Data Type	Length	Description
GroupIDM	INTEGER	4	Part-key used to uniquely identify a group within the installation.
GroupID	INTEGER	4	Part-key used to uniquely identify a group within the installation.
ParentGroupIDM	INTEGER	4	Part-key identifying the child groups parent
ParentGroupID	INTEGER	4	Part-key identifying the child groups parent

WGN_V_RLS_ADDR_1

This view references to one of the underlying RLS tables. When a search user connects to the CA DataMinder database it populates this table with users; their addresses and groups that a particular user is allowed to view events for and review; and the start and end dates that each user was part of a particular group.

Note: This table is not populated for admin Users and can return zero rows in a report if used by an admin user.

View Stack

WGN_V_RLS_ADDR_1

→ *TMP_WGN3ADDRRLS*

Column Details

Column	Data Type	Length	Description
EffectiveStartDate	TIMESTAMP	8	End date when a particular reviewer can view a particular user's CA DataMinder events.
EndDate	TIMESTAMP	8	Start date when a particular reviewer can view a particular user's CA DataMinder events.
addressuid1	DECIMAL	13	Part-key used to uniquely identify an email address or the address of an IM participant.
addressuid2	DECIMAL	13	Part-key used to uniquely identify an email address or the address of an IM participant.
useridm	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
userid	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
groupidm	INTEGER	4	Part-key used to uniquely identify a group within the installation.
groupid	INTEGER	4	Part-key used to uniquely identify a group within the installation.

WGN_V_RLS_EX_ADDR_1

This view references to one of the underlying RLS tables. When a search user connects to the CA DataMinder database it populates this table with users and their addresses that the logged on user is excluded from reviewing. This is primarily used with the 'Self Excluded' RLS models which have been designed so that a reviewer cannot review events that they are a participant in.

Note: This table is not populated for admin users and can return zero rows in a report if used by an admin user.

View Stack

WGN_V_RLS_EX_ADDR_1

→ *TMP_WGN3EXADDRRLS*

Column Details

Column	Data Type	Length	Description
AddressUID1	DECIMAL	13	Part-key used to uniquely identify an email address or the address of an IM participant.
AddressUID2	DECIMAL	13	Part-key used to uniquely identify an email address or the address of an IM participant.
UserIDM	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
UserID	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.

WGN_V_RLS_POLICY_1

This view references to one of the underlying RLS tables. When a search user connects to the CA DataMinder database, it populates this table with policies and policy classes the reviewer can see. This is specifically used with the 'Policy' RLS models.

Note: This table is not populated for admin users and can return zero rows in a report if used by an admin user.

View Stack

WGN_V_RLS_POLICY_1

→ TMP_WGN3POLICYRLS

Column Details

Column	Data Type	Length	Description
PolicyID	INTEGER	4	Identifies the policy classification associated with the trigger.
PolicyName	VARCHAR	255	Policy name associated to the PolicyID
ParentPolicyID	INTEGER	4	Identifies the policy class classification associated with the trigger.
ParentPolicyName	VARCHAR	255	Class name associated to the ParentPolicyID

Non-Hierarchical Views

These are 4 views:

- WGN_V_NH_EVENTPNTUSER_1
- WGN_V_NH_EVENTPNTUSRGR_1
- WGN_V_NH_PNTUSER_1
- WGN_V_NH_PNTUSER_2

They were specifically designed for reporting purposes so that events could be distinguish between those that have no associated users (typically FSA type events) and events that have external participants. The RLS views for these simply point to the equivalent standard view. It is the admin views where these views become relevant as you want to be able to see all events as administrator but you also want to know if other participants on the event have internal users. The columns in these views make the equivalent standard views. However to distinguish between the 2 types of events, the UserID's and GroupID's are return with a value of -1 for non-hierarchical events.

Other Views

Below are listed a number of other views installed into the database but are for internal use only and should not be used for reporting purposes.

- WGN_V_READ_ONLY_ISSUE_1
- WGN_V_RLS_MODEL
- WGN_V_RLS_MODEL_TYPE

Review Queue Views Definitions

The Review Queue View's belonged to a different functional area from the main product and would only be applicable to customers that actively use the Review Queue functionality. With V12.0 these views are installed as part of the main Server installation. However they have been available since V4.7 as a separate installation.

More information:

Naming Convention

The naming convention is the same has the main product view and start WGN_V_RQ_. The views fall into three families of 3 views each, plus one other additional view. To summarise the three families: 'Event' containing events in the review queue; 'Entry' containing review queue entries; 'Event Entry' containing both events and the corresponding review queue entries. Within each family there are separate views for 'All', 'Closed', 'Open'; with an obvious naming convention indicating which; that return all, closed or open event records respectively. The additional view to these is WGN_V_RQ_CRITERIA_CL_1 and defines what a closed event is. Open events are then defined as those that are not closed.

The view names below are the same for both SQL Server and Oracle in r12.0. For SQL Server customers that have installed the review queue separately on an earlier version the names of these views differ. Any such customer upgrading to r12.0 will find that both sets of views exist, but the old named views now just point to the new views, and any future development of reports should use the new views referred to in this document. See the table below for a summary of the name changes.

View	Old SQL Server View
WGN_V_RQ_ENTRY_ALL_1	WGN_V_RQ_ENTRIES_ALL_1
WGN_V_RQ_ENTRY_CL_1	WGN_V_RQ_ENTRIES_CLOSED_1

WGN_V_RQ_ENTRY_OP_1	WGN_V_RQ_ENTRIES_OPEN_1
WGN_V_RQ_EVENT_ALL_1	WGN_V_RQ_EVENTS_ALL_1
WGN_V_RQ_EVENT_CL_1	WGN_V_RQ_EVENTS_CLOSED_1
WGN_V_RQ_EVENT_OP_1	WGN_V_RQ_EVENTS_OPEN_1
WGN_V_RQ_EVENTRY_ALL_1	WGN_V_RQ_EVENT_ENTRIES_ALL_1
WGN_V_RQ_EVENTRY_CL_1	WGN_V_RQ_EVENT_ENTRIES_CLOSED_1
WGN_V_RQ_EVENTRY_OP_1	WGN_V_RQ_EVENT_ENTRIES_OPEN_1
WGN_V_RQ_CRITERIA_CL_1	WGN_V_RQ_CRITERIA_CLOSED_1

WGN_V_RQ_ENTRY_ALL_1

This view returns all columns from Wgn3ReviewQueue. This view will return one row per review queue entry.

View Stack

WGN_V_RQ_ENTRY_ALL_1

→ WGN_V_m_RQ_ENTRY_ALL_1

→ WGN_V_MD_RQ_ENTRY_ALL_1

→ WGN_V_AD_RQ_ENTRY_ALL_1

→ *Wgn3ReviewQueue*

WGN_V_RLS_1

→ *TMP_Wgn3RLS*

Column Details

Column	Data Type	Length	Description
GroupIDM	INTEGER	4	Part-key used to uniquely identify a group within the installation.
GroupID	INTEGER	4	Part-key used to uniquely identify a group within the installation.
EventUID	IDENTITYDEF	13	Key used to uniquely identify a captured or imported event.
EventTimeStamp	TIMESTAMP	8	The time at which the event occurred.
ParticipantIndex	INTEGER	4	Part-key used to identify an event participant.

UserIDM	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
UserID	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
Category	INTEGER	4	Indicates which category of metric caused the entry to be written to the queue
Run	DECIMAL	13	This just shows a run number indicating which run wrote the entry.
Precedence	INTEGER	4	Not used, reserved for possible future use.

WGN_V_RQ_ENTRY_CL_1

This view returns all columns from WGN_V_RQ_ENTRY_ALL_1. But this view only returns events for which issues are closed (dependent on the definition of WGN_V_RQ_CRITERIA_CL_1). This view will return one row per review queue entry.

View Stack

WGN_V_RQ_ENTRY_CL_1
→ WGN_V_m_RQ_ENTRY_CL_1
→ WGN_V_MD_RQ_ENTRY_CL_1
→ WGN_V_MD_RQ_ENTRY_ALL_1
→ WGN_V_AD_RQ_ENTRY_ALL_1
→ *Wgn3ReviewQueue*
WGN_V_RLS_1
→ TMP_WGN3RLS
WGN_V_MD_RQ_ENTRY_OP_1
→ WGN_V_RLS_1
→ *TMP_Wgn3RLS*
WGN_V_AD_RQ_ENTRY_OP_1
→ *Wgn3ReviewQueue*
WGN_V_RQ_CRITERIA_CL_1
→ WGN_V_RQ_CRITERIA_CL_1_DEF
→ *Wgn3EventIssue*
Wgn3IssueParticipant

Wgn3EventAudit

WGN_V_RQ_ENTRY_OP_1

This view returns all columns from WGN_V_RQ_ENTRY_ALL_1. But this view only returns events for which issues are open (dependent on the definition of WGN_V_RQ_CRITERIA_CL_1). This view will return one row per review queue entry.

View Stack

WGN_V_RQ_ENTRY_OP_1
→ WGN_V_m_RQ_ENTRY_OP_1
→ WGN_V_MD_RQ_ENTRY_OP_1
→ WGN_V_RLS_1
→ *TMP_Wgn3RLS*
WGN_V_AD_RQ_ENTRY_OP_1
→ *Wgn3ReviewQueue*
WGN_V_RQ_CRITERIA_CL_1
→ WGN_V_RQ_CRITERIA_CL_1_DEF
→ *Wgn3EventIssue*
Wgn3IssueParticipant
Wgn3EventAudit

WGN_V_RQ_EVENTRY_ALL_1

This view returns all columns from Wgn3Event and most, not already returned by the event table, from Wgn3ReviewQueue. This view will return one row per review queue entry.

View Stack

WGN_V_RQ_EVENTRY_ALL_1
→ WGN_V_m_RQ_EVENTRY_ALL_1
→ WGN_V_MD_RQ_EVENTRY_ALL_1
→ *WGN3Event*
WGN_V_MD_RQ_ENTRY_ALL_1

→ WGN_V_AD_RQ_ENTRY_ALL_1

→ *Wgn3ReviewQueue*

WGN_V_RLS_1

→ *TMP_Wgn3RLS*

Column Details

Column	Data Type	Length	Description
EventUID	IDENTITYDEF	13	Key used to uniquely identify a captured or imported event.
EventTimestamp	TIMESTAMP	8	The time at which the event occurred.
SequenceIDM	INTEGER	4	Part-key used to identify the owning sequence and also uniquely identify the event.
SequenceID	INTEGER	4	Part-key used to identify the owning sequence and also uniquely identify the event.
EventIndex	INTEGER	4	Part-key used to uniquely identify the event.
EventMajorType	INTEGER	4	Identifies the 'major' event type (such as, email or web page)
EventMinorType	INTEGER	4	Identifies the 'minor' event type (such as, email sent)
EventSubType	INTEGER	4	Identifies the 'direction' of the event (such as, incoming or outgoing)
AesMajorType	INTEGER	4	Part-key that identifies the source object (software package) that captured the sequence.
AesMinorType	INTEGER	4	Part-key that identifies the source object (software package) that captured the sequence.
AesSubType	INTEGER	4	Part-key that identifies the source object (software package) that captured the sequence.
EventText1	LONGVARCHAR	255	Event specific string identifying the event.
EventText2	LONGVARCHAR	4000	Event specific string identifying the event.
EventAttributes	VARCHAR	255	Attributes of the event.

ExternalID	VARCHAR	255	This attribute contains an external (non-CA DataMinder) unique ID of an event to enable matching with duplicates of the same event captured elsewhere.
Duration	INTEGER	4	The length of time for which the event occurred.
ExpiryTimestamp	TIMESTAMP	8	Stores the time at which the event expires. Expired events are removed by the next scheduled database purge.
IsPermanent	BIT	1	If this column has a non-zero (true) value, then the event will never be deleted by a scheduled database purge.
UpdateTimestamp	TIMESTAMP	8	The time that the event was last modified.
PurgeState	INTEGER	4	Indicates whether the row can be purged if 'Purge On Replicate' is enabled. This column is ignored on a CMS.
GMTOffset	INTEGER	4	Specifies the difference (in minutes) between UTC time and the time zone in which an events source machine resides.
DSTOffset	INTEGER	4	Specifies the size of the daylight saving offset (in minutes) for the source machine's timezone.
BlobType	INTEGER	4	Identifies the type of the blob (such as a file based blob).
BlobLocation	VARCHAR	255	The location of the blob data. The format of this string depends on the BlobType.
BlobSize	DECIMAL	13	The size of the blob (in bytes) before encryption/compression.
BLOBPhysicalSize	DECIMAL	13	The size of the blob (in bytes) after encryption/compression.
epeventtimestamp	TIMESTAMP	8	Same as EventTimeStamp but this is taken from Wgn3EventParticipants instead of Wgn3Event. Used to ensure predicates commute.
GroupIDM	INTEGER	4	Part-key used to uniquely identify a group within the installation.
GroupID	INTEGER	4	Part-key used to uniquely identify a group within the installation.
ParticipantIndex	INTEGER	4	Part-key used to identify an event participant.

UserIDM	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
UserID	INTEGER	4	Part-key identifying the CA DataMinder user associated with the login.
Category	INTEGER	4	Indicates which category of metric caused the entry to be written to the queue
Precedence	INTEGER	4	Not used, reserved for possible future use.

WGN_V_RQ_EVENTRY_CL_1

This returns the same columns as WGN_V_RQ_EVENTRY_ALL_1. But this view only returns events for which issues are closed (dependent on the definition of WGN_V_RQ_CRITERIA_CL_1). This view will return one row per review queue entry.

View Stack

WGN_V_RQ_EVENTRY_CL_1

→ WGN_V_m_RQ_EVENTRY_CL_1

→ WGN_V_MD_RQ_EVENTRY_CL_1

→ WGN_V_MD_RQ_EVENTRY_ALL_1

→ *WGN3Event*

WGN_V_MD_RQ_ENTRY_ALL_1

→ WGN_V_AD_RQ_ENTRY_ALL_1

→ *Wgn3ReviewQueue*

WGN_V_RLS_1

→ *TMP_Wgn3RLS*

WGN_V_MD_RQ_EVENTRY_OP_1

→ *WGN3Event*

WGN_V_MD_RQ_ENTRY_OP_1

→ WGN_V_RLS_1

→ *TMP_Wgn3RLS*

WGN_V_AD_RQ_ENTRY_OP_1

→ *Wgn3ReviewQueue*

WGN_V_RQ_CRITERIA_CL_1

→ WGN_V_RQ_CRITERIA_CL_1_DEF

→ *Wgn3EventIssue*

Wgn3IssueParticipant

Wgn3EventAudit

WGN_V_RQ_EVENTRY_OP_1

This returns the same columns as WGN_V_RQ_EVENTRY_ALL_1. But this view only returns events for which issues are open (dependent on the definition of WGN_V_RQ_CRITERIA_CL_1). This view will return one row per review queue entry.

View Stack

WGN_V_RQ_EVENTRY_OP_1

→ WGN_V_m_RQ_EVENTRY_OP_1

→ WGN_V_MD_RQ_EVENTRY_OP_1

→ *WGN3Event*

WGN_V_MD_RQ_ENTRY_OP_1

→ WGN_V_RLS_1

→ *TMP_Wgn3RLS*

WGN_V_AD_RQ_ENTRY_OP_1

→ *Wgn3ReviewQueue*

WGN_V_RQ_CRITERIA_CL_1

→ WGN_V_RQ_CRITERIA_CL_1_DEF

→ *Wgn3EventIssue*

Wgn3IssueParticipant

Wgn3EventAudit

WGN_V_RQ_EVENT_ALL_1

This returns all the columns from Wgn3Event for events that are in the review queue. This view only returns one row per event.

View Stack

WGN_V_RQ_EVENT_ALL_1

→ WGN_V_m_RQ_EVENT_ALL_1

→ WGN_V_MD_RQ_EVENT_ALL_1

→ *Wgn3Event*

WGN_V_MD_RQ_ENTRY_ALL_1

→ WGN_V_AD_RQ_ENTRY_ALL_1

→ *Wgn3ReviewQueue*

WGN_V_RLS_1

→ *TMP_Wgn3RLS*

Column Details

Column	Data Type	Length	Description
EventUID	IDENTITYDEF	13	Key used to uniquely identify a captured or imported event.
EventTimestamp	TIMESTAMP	8	The time at which the event occurred.
SequenceIDM	INTEGER	4	Part-key used to identify the owning sequence and also uniquely identify the event.
SequenceID	INTEGER	4	Part-key used to identify the owning sequence and also uniquely identify the event.
EventIndex	INTEGER	4	Part-key used to uniquely identify the event.
EventMajorType	INTEGER	4	Identifies the 'major' event type (such as, email or web page)
EventMinorType	INTEGER	4	Identifies the 'minor' event type (such as, email sent)
EventSubType	INTEGER	4	Identifies the 'direction' of the event (such as, incoming or outgoing)
AesMajorType	INTEGER	4	Part-key that identifies the source object (software package) that captured the sequence.
AesMinorType	INTEGER	4	Part-key that identifies the source object (software package) that captured the sequence.
AesSubType	INTEGER	4	Part-key that identifies the source object (software package) that captured the sequence.
EventText1	LONGVARCHAR	255	Event specific string identifying the event.
EventText2	LONGVARCHAR	4000	Event specific string identifying the event.
EventAttributes	VARCHAR	255	Attributes of the event.
ExternalID	VARCHAR	255	This attribute contains an external (non-CA DataMinder) unique ID of an event to enable matching with duplicates of the same event captured elsewhere.

Column	Data Type	Length	Description
EventUID	IDENTITYDEF	13	Key used to uniquely identify a captured or imported event.
Duration	INTEGER	4	The length of time for which the event occurred.
ExpiryTimestamp	TIMESTAMP	8	Stores the time at which the event expires. Expired events are removed by the next scheduled database purge.
IsPermanent	BIT	1	If this column has a non-zero (true) value, then the event will never be deleted by a scheduled database purge.
UpdateTimestamp	TIMESTAMP	8	The time that the event was last modified.
PurgeState	INTEGER	4	Indicates whether the row can be purged if 'Purge On Replicate' is enabled. This column is ignored on a CMS.
GMTOffset	INTEGER	4	Specifies the difference (in minutes) between UTC time and the time zone in which an events source machine resides.
DSTOffset	INTEGER	4	Specifies the size of the daylight saving offset (in minutes) for the source machine's timezone.
BlobType	INTEGER	4	Identifies the type of the blob (such as a file based blob).
BlobLocation	VARCHAR	255	The location of the blob data. The format of this string depends on the BlobType.
BlobSize	DECIMAL	13	The size of the blob (in bytes) before encryption/compression.
BLOBPhysicalSize	DECIMAL	13	The size of the blob (in bytes) after encryption/compression.
epeventtimestamp	TIMESTAMP	8	Same as EventTimeStamp but this is taken from Wgn3EventParticipants instead of Wgn3Event. Used to ensure predicates commute.

WGN_V_RQ_EVENT_CL_1

This returns all the columns from WGN_V_RQ_EVENT_ALL_1. But this view only returns events for which issues are closed (dependent on the definition of WGN_V_RQ_CRITERIA_CL_1). This view only returns one row per event.

View Stack

WGN_V_RQ_EVENT_CL_1

→ WGN_V_m_RQ_EVENT_CL_1

→ WGN_V_MD_RQ_EVENT_CL_1

→ WGN_V_MD_RQ_EVENT_ALL_1

→ Wgn3Event

WGN_V_MD_RQ_ENTRY_ALL_1

→ WGN_V_AD_RQ_ENTRY_ALL_1

→ *Wgn3ReviewQueue*

WGN_V_RLS_1

→ *TMP_Wgn3RLS*

WGN_V_MD_RQ_EVENT_OP_1

→ *WGN3Event*

WGN_V_MD_RQ_ENTRY_OP_1

→ WGN_V_RLS_1

→ *TMP_Wgn3RLS*

WGN_V_AD_RQ_ENTRY_OP_1

→ *Wgn3ReviewQueue*

WGN_V_RQ_CRITERIA_CL_1

→ WGN_V_RQ_CRITERIA_CL_1_DEF

→ *Wgn3EventIssue*

Wgn3IssueParticipant

Wgn3EventAudit

WGN_V_RQ_EVENT_OP_1

This returns all the columns from WGN_V_RQ_EVENT_ALL_1. But this view only returns events for which issues are open (dependent on the definition of WGN_V_RQ_CRITERIA_CL_1). This view only returns one row per event.

View Stack

WGN_V_RQ_EVENT_OP_1

→ WGN_V_m_RQ_EVENT_OP_1

→ WGN_V_MD_RQ_EVENT_OP_1

→ *WGN3Event*

WGN_V_MD_RQ_ENTRY_OP_1

→ WGN_V_RLS_1

→ *TMP_Wgn3RLS*

WGN_V_AD_RQ_ENTRY_OP_1

→ *Wgn3ReviewQueue*

WGN_V_RQ_CRITERIA_CL_1

→ WGN_V_RQ_CRITERIA_CL_1_DEF

→ *Wgn3EventIssue*

Wgn3IssueParticipant

Wgn3EventAudit

WGN_V_RQ_CRITERIA_CL_1

This view returns event participant references for all participants that meet the criteria for removal from the review queue. It is used to determine which queue entries are to be purged and will no longer be presented for review, i.e. those that are closed. The default criteria for an issue being defined as closed is when the Audit Status is 'Approved' in the iConsole. The default value for this is defined as the event having a current audit record with an Audit Type of '1' and a corresponding value of '1'. However it is possible to customize this view so that customers can define what constitutes a closed issue. RLS is not applied to this view and it is used in the other review queue views to determine which events are open/closed.

View Stack

WGN_V_RQ_CRITERIA_CL_1

→ WGN_V_RQ_CRITERIA_CL_1_DEF

→ *Wgn3EventIssue*

Wgn3IssueParticipant

Wgn3EventAudit

Column Details

Column	Data Type	Length	Description
EventUID	IDENTITYDEF	13	Key used to uniquely identify a captured or imported event.
EventTimeStamp	TIMESTAMP	8	The time at which the event occurred.

ParticipantIndex	INTEGER	4	Part-key used to identify an event participant.

Administrative Searches

Due to performance problem with the admin views prior to r12.5 workarounds were available which involved building RLS table for all groups. The workaround involved the following:

The original workaround required writing the search or report not using the top level view but the _RLS view which sits directly below the top view. For example. Instead of using WGN_V_EVENTPARTPNTUSER_1 use WGN_V_EVENTPARTPNTUSER_RLS_1. As the search user also has SELECT privileges granted on this view, it is possible to still for non-admin user to run the search. However if the search is written like this, when an admin user tries to run the search it will return no rows as the underlying RLS tables will not have been populated. Therefore before the searches can be run the stored procedure (SQL Server) or function (Oracle) must manually apply RLS for the admin user. This is achieved by a call to the procedure WGN_POPULATE_GROUP_HIER_RLS. The procedure requires three parameters: IDM, ID and Group Filter flag. The values of the first two parameters are dependent on the third parameter. If the Group Filter flag is set to 0, then the other two parameters should be UserIDM and UserID for the admin user and the users management groups will be returned. If the Group filter flag is set to 1, then the other two parameters should be GroupIDM and GroupID for the management group to which RLS will be applied. SQL Server also requires an extra step to clean up the database once the search as completed as the tables here are permanent. The search stored procedure should call WGN_REMOVE_GROUP_HIER_RLS to delete all rows from the RLS tables associated with the admin user.

However the admin views have now been fixed to perform much better so this workaround is no longer need. Additionally if this workaround as been implemented it is strongly recommend that you now remove this and use the normal admin views. The work around will not work if the customers decide to implement additional RLS model, plus as already stated in Primary Views Definitions (see page 131) these views are being deprecated.

View Availability in CA DataMinder Releases

Below is a table listing each view, stating in which release the view was introduced. Additionally an indication is given whether RLS is applied based on the security model. Also given is an indication of how the views are written for a hybrid model. They can be categorized as one of six ways:

0 - None: RLS is not applied to this view

1 - Event based views: These require both type of RLS applied and are essentially provide in intersection of the data. Therefore the reviewer needs to be able to see both the users group and the policy that has been applied

2 - Policy: Only policy based RLS is required for these views.

3 - Management group: Only management based RLS is applied

4 - Issue based: For some issue views we need to apply both management group and policy based RLS. But for these views it is possible that issue is only create against the participant but not the policy so these events still needs to be visible to the reviewer.

5 - Union: This is a special case RLS is applied as a union rather than in intersection.

View	Released	Management	Policy	Hybrid
WGN_V_CURR_ISSUE_PARTPNT_1	4.7	Yes	Yes	4
WGN_V_CURR_ISSUE_TRIGGER_1	12.5	No	Yes	2
WGN_V_CURR_PROP_VAL_1	4.7	No	No	0
WGN_V_EVENT_1	4.0	Yes	Yes	1
WGN_V_EVENT_2	4.7	Yes	Yes	1
WGN_V_EVENTPARTPNTUSER_1	4.0	Yes	Yes	1
WGN_V_EVENTPARTUSERGRP_1	4.7	Yes	Yes	1
WGN_V_GROUP_1	4.0	Yes	No	3
WGN_V_GROUP_HIST_1	4.0	No	No	0
WGN_V_INTPARTPNTUSER_1	6.0	No	No	0
WGN_V_ISSUE_1	4.0	Yes	Yes	4
WGN_V_ISSUE_2	4.7	Yes	Yes	4
WGN_V_ISSUE_PARTPNT_1	4.0	Yes	No	3
WGN_V_ISSUE_PARTPNT_2	4.7	Yes	No	3
WGN_V_ISSUE_PARTPNT_3	6.0	Yes	No	3

WGN_V_NH_EVENTPNTUSER_1	12.5	Yes	Yes	1
WGN_V_NH_EVENTPNTUSRGR_1	12.5	Yes	Yes	1
WGN_V_NH_PNTUSER_1	12.5	Yes	Yes	1
WGN_V_NH_PNTUSER_2	12.5	Yes	Yes	1
WGN_V_PARTPNTUSER_1	4.0	No	No	0
WGN_V_POLICY_PICKER_1	12.5	No	Yes	2
WGN_V_PNTUSER_1	4.7	Yes	Yes	1
WGN_V_PNTUSER_2	12.5	Yes	Yes	1
WGN_V_QUARANTINE_EVENT_1	4.0	No	No	0
WGN_V_USER_1	4.0	Yes	No	3
WGN_V_USER_GROUP_1	4.5	Yes	No	3
WGN_V_USER_GROUP_HIST_1	6.0	Yes	No	3
WGN_V_USER_HIST_1	6.0	Yes	No	3
WGN_V_RLS_1	4.0	-	-	-
WGN_V_RLS_ADDR_1	4.0	-	-	-
WGN_V_RLS_EX_ADDR_1	12.0	-	-	-
WGN_V_RLS_POLICY_1	12.5	-	-	-
WGN_V_TRIGGER_1	12.5	No	Yes	2
WGN3USERADDRESS	12.0	No	No	0
WGN_V_USER_ADDRESS_1	12.0	No	No	0
WGN_V_ADDRESS_1	12.0	No	No	0
WGN_V_RQ_ENTRY_ALL_1	12.0	Yes	Yes	1
WGN_V_RQ_ENTRY_CL_1	12.0	Yes	Yes	1
WGN_V_RQ_ENTRY_OP_1	12.0	Yes	Yes	1
WGN_V_RQ_EVENT_ALL_1	12.0	Yes	Yes	1
WGN_V_RQ_EVENT_CL_1	12.0	Yes	Yes	1
WGN_V_RQ_EVENT_OP_1	12.0	Yes	Yes	1
WGN_V_RQ_EVENTRY_ALL_1	12.0	Yes	Yes	1
WGN_V_RQ_EVENTRY_CL_1	12.0	Yes	Yes	1
WGN_V_RQ_EVENTRY_OP_1	12.0	Yes	Yes	1
WGN_V_RQ_CRITERIA_CL_1	12.0	No	No	0