

Markvision Enterprise 1.2

Database Access

July 2011
Lexmark International, Inc.
740 West New Circle Road
Lexington, KY 40550

Abstract

This white paper focuses on the database structure of Markvision Enterprise (MVE). The intent of this document is to enable users to understand the tables, fields, and relationships within the database so they can create custom queries and reports.

Edition: July 2011

The following paragraph does not apply to any country where such provisions are inconsistent with local law: LEXMARK INTERNATIONAL, INC., PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in later editions. Improvements or changes in the products or the programs described may be made at any time.

References in this publication to products, programs, or services do not imply that the manufacturer intends to make these available in all countries in which it operates. Any reference to a product, program, or service is not intended to state or imply that only that product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any existing intellectual property right may be used instead. Evaluation and verification of operation in conjunction with other products, programs, or services, except those expressly designated by the manufacturer, are the user's responsibility.

For Lexmark technical support, visit support.lexmark.com.

For information on supplies and downloads, visit www.lexmark.com.

If you don't have access to the Internet, you can contact Lexmark by mail:

Lexmark International, Inc.
Bldg 004-2/CSC
740 New Circle Road NW
Lexington, KY 40550
USA

© 2011 Lexmark International, Inc.

All rights reserved.

Trademarks

Lexmark, Lexmark with diamond design, and Markvision are trademarks of Lexmark International, Inc., registered in the United States and/or other countries.

All other trademarks are the property of their respective owners.

UNITED STATES GOVERNMENT RIGHTS

This software and any accompanying documentation provided under this agreement are commercial computer software and documentation developed exclusively at private expense.

MVE Database Access

Abstract	2
Edition: July 2011	2
1. Overview	5
2. Release Notes	5
2.1 MVE 1.1	5
3. Supported Databases	6
3.1 Database Differences	6
4. Advanced SQL Server Configuration	6
4.1 Named Instances	6
5. FRAMEWORK Tables and Field Descriptions	7
5.1 Printer	7
5.1.1 CONFIG_ITEM	7
5.1.2 NETWORK_ADAPTER	8
5.1.3 NETWORK_PRINTER	9
5.1.4 PRINTER_CURRENT_STATUS	11
5.1.5 PRINTER_ESF_APPS	11
5.1.6 PRINTER_INPUT_OPTIONS	12
5.1.7 PRINTER_INPUT_TRAYS	12
5.1.8 PRINTER_OPTIONS	12
5.1.9 PRINTER_OUTPUT_BINS	13
5.1.10 PRINTER_OUTPUT_OPTIONS	13
5.1.11 PRINTER_PORTS	14
5.1.12 PRINTER_STATISTICS	14
5.1.13 PRINTER_SUPPLIES	16
5.2 Printer History	17
5.2.1 REVINFO	18
5.3 Keywords	18
5.3.1 ASSIGNED_KEYWORDS	18
5.3.2 KEYWORD	18
5.3.3 KEYWORD_CATEGORY	19
5.4 Policies	19
5.4.1 ASSIGNED_POLICIES	19
5.4.2 FAILED_SETTINGS	19
5.4.3 FLASH_SUBSYSTEMS	19
5.4.4 FLASHFILE	20
5.4.5 POLICY	20
5.4.6 POLICY_SETTINGS	20
5.5 Discovery Profiles	21
5.5.1 DISCOVERY_PROFILE	21
5.5.2 DISCOVERY_PROFILE_CI	22
5.5.3 EXCLUDE_PROFILE_ITEM	22
5.5.4 INCLUDE_PROFILE_ITEM	22
5.6 Authentication and Authorization	22
5.6.1 MASTER_ROLE	22
5.6.2 USERS	23
5.6.3 USER_ROLE	23
5.7 Security Policy	23
5.8 Event Manager	24
5.8.1 ALERT	24
5.8.2 ASSIGNED_EVENTS	24
5.8.3 DESTINATION	24
5.8.4 EVENT	25
5.8.5 EVENT_ALERTS	25

5.8.6	EVENT_DESTINATIONS.....	25
5.8.7	PRINTER_EVENT_ACTIVE_CONDITIONS.....	25
5.9	Miscellaneous.....	26
5.9.1	APPLICATION_SETTINGS.....	26
5.9.2	BOOKMARK.....	26
5.9.3	CHANGED_SETTINGS.....	26
5.9.4	DATABASECHANGELOG.....	26
5.9.5	DATABASECHANGELOGLOCK.....	26
5.9.6	SMTP_CONFIGURATION.....	27
5.9.7	SYSTEM_LOG.....	27
6.	MONITOR Tables and Field Descriptions.....	27
6.1	TASK_INFO.....	27
7.	Historical Data.....	28
7.1	Historical Data Generation.....	28
7.2	Querying Historical Data.....	29
7.2.1	Time-based Queries.....	29

1. Overview

Markvision Enterprise (MVE) allows you to gather various types of data about your discovered devices and their settings. Some of this data includes:

- Identification information such as IP address, model name, and manufacturer
- Device capabilities such as color, copy, and speed
- Options installed on devices such as hard drives and paper trays
- Supply statistics and capacities
- Device discovery and audit statistics
- Policies and keywords assigned to devices

MVE stores this information in three databases.

- FRAMEWORK – The primary database where most of the device data is stored
- MONITOR – A smaller database where task statistics are stored
- QUARTZ – A database used by MVE's third-party scheduler (Quartz) to store scheduling information (not covered in this document)

This document lists and explains most of the tables in the FRAMEWORK and MONITOR databases and describes the fields each table contains. The document also describes how the tables and fields are related so users can construct custom queries and generate custom reports.

Note – The tables and columns in the database are subject to change from one release to the next. See the Release Notes section for details on these changes.

2. Release Notes

This section details the notable schema changes for each release. It is meant to be helpful for clients that perform SQL queries in the MVE database and/or integrate other applications at the database layer. For detailed descriptions, refer to sections 5 and 6 of this document.

2.1 MVE 1.1

- Table BUILDING_BLOCK_SETTINGS
 - Columns
 - SETTING_NAME added
 - SETTING_TYPE added
 - SETTING_VALUE added
 - SETTING_VALUE_BIN added
 - Keys
 - SETTING_ID added as primary key
 - FK_BB_SETTINGS_SETTING_ID dropped as foreign key
- Table INTERNAL_ACCOUNT_SETTINGS
 - Columns
 - SETTING_NAME added
 - SETTING_TYPE added
 - SETTING_VALUE added
 - SETTING_VALUE_BIN added
 - Keys
 - SETTING_ID added as primary key
 - FK_IA_SETTINGS_SETTING_ID dropped as foreign key

- Table POLICY_MISC_SETTINGS
 - Columns
 - SETTING_NAME added
 - SETTING_TYPE added
 - SETTING_VALUE added
 - SETTING_VALUE_BIN added
 - Keys
 - SETTING_ID added as primary key
 - FK_MISC_SETTINGS_SETTING_ID dropped as foreign key
- Table CHANGED_SETTINGS added as new table
- Table POLICY
 - Columns
 - CREDENTIAL_PASSWORD added
 - CREDENTIAL_PIN added
 - CREDENTIAL_REALM added
 - CREDENTIAL_USERNAME added
- Other changes
 - Table POLICY_SETTINGS
 - SETTING_VALUE contains an encrypted value
 - Table FAILED_SETTINGS
 - SETTINGVALUE contains an encrypted value

3. Supported Databases

MVE supports the following databases. Consult the MVE User's Guide or Release Notes for supported versions.

- Firebird (embedded into MVE)
- Microsoft SQL Server

3.1 Database Differences

The following table maps the Firebird data types used in MVE to their corresponding data types in the Microsoft SQL Server MVE database.

Firebird Data Types	SQL Server Data Types
BIGINT	bigint
VARCHAR(x)	varchar(x)
TIMESTAMP	datetime
INTEGER	int
SMALLINT	bit
BLOB SUB_TYPE 0	varbinary(1024)

4. Advanced SQL Server Configuration

4.1 Named Instances

The MVE install process does not support Microsoft SQL Server named instance configuration. However, MVE can support named instances through the manual edit of the database connection pool properties file. The following manual configuration must be performed as a post-installation task prior to using MVE.

1. Install MVE using the SQL Server database option and provide the credentials, host and port for the named instance.
2. After installation, MVE will not be functional, as the named instance was not collected and used to create the MVE database schema.
3. Stop MVE via the Windows Service panel.
4. Using Windows file explorer, navigate to the MVE installation directory (MVE_INSTALL) and edit the file %MVE_INSTALL%/apps/WEB-INF/classes/database.properties.
5. Append the text “;instance=<instance name>” to each of the database connection URL properties, substituting <instance name> with the name of the desired SQL Server named instance. For example, to configure an instance named “MVE” the three connection URL properties would resemble the following:

Sample database.properties with edits

```
database.connection.url=${database.jdbc.base.connection.url}/${database.dataSourcePath};instance=MVE
database.quartz.connection.url=${database.jdbc.base.connection.url}/${database.quartz.dataSourcePath};instance=MVE
database.monitor.connection.url=${database.jdbc.base.connection.url}/${database.monitor.dataSourcePath};instance=MVE
```

6. Start MVE via the Windows Service panel.

5. FRAMEWORK Tables and Field Descriptions

FRAMEWORK is the primary MVE database where most of the device data is stored. The tables and descriptions of each field are listed in the following table categories.

Note – The data types used in the “Data Type” columns of these tables relate to a Firebird database. Refer to the Database Differences section of this document to see how these correspond to Microsoft SQL Server data types.

5.1 Printer

The following tables deal with the logical representation of a physical printer.

5.1.1 CONFIG_ITEM

This table represents the ITIL Configuration Item portion of the printer. It has “state” and TIMESTAMPS of when the CI was created, initially managed, last discovered, etc. This does not represent any physical portion of a printer; it is simply a concept.

Field Name	Data Type	Description
CI_ID	BIGINT	Primary key
CI_STATE	VARCHAR(255)	The current state of the CI: NEW, MANAGED, MISSING, FOUND, CHANGED, UNMANAGED, RETIRED
CREATION_DATE	TIMESTAMP	When the CI first entered the system
INITIAL_MANAGEMENT_DATE	TIMESTAMP	When the CI first entered the MANAGED state (or sub-state)
LAST_AUDIT_DATE	TIMESTAMP	The last time an audit was attempted on the CI (may not have been successful)

PRINTER_ID	BIGINT	The foreign key to NETWORK_PRINTER.PRINTER_ID
LAST_DISCOVERY_DATE	TIMESTAMP	The last time a discovery was attempted on the CI (may not have been successful)
LAST_SUCCESSFUL_AUDIT_DATE	TIMESTAMP	The last time an audit was successful on the CI
LAST_SUCCESSFUL_DISCOVERY_DATE	TIMESTAMP	The last time a discovery was successful on the CI

5.1.2 NETWORK_ADAPTER

This table represents the physical printer s network adapter (a.k.a. print server).

Field Name	Data Type	Description
ADAPTER_TYPE	VARCHAR(31)	For now, always INA (Internal Network Adapter)
ADAPTER_ID	BIGINT	Primary key
FIRMWARE_REVISION	VARCHAR(255)	Current network firmware revision
MANUFACTURER	VARCHAR(255)	N/A
MODEL_NAME	VARCHAR(255)	N/A
SERIAL_NUMBER	VARCHAR(50)	N/A
SYSTEM_NAME	VARCHAR(255)	SNMP system.sysName (i.e. .1.3.6.1.2.1.1.5)
RETRIES	INTEGER	Number of times to retry a particular communication attempt
SNMP_READ_COMMUNITY_NAME	VARCHAR(255)	SNMP Community Name for reading
SNMP_WRITE_COMMUNITY_NAME	VARCHAR(255)	SNMP Community Name for writing
TIMEOUT	BIGINT	Number of milliseconds to wait for a response from this printer for a particular communication attempt
CONTACT_LOCATION	VARCHAR(255)	SNMP system.sysLocation (i.e. .1.3.6.1.2.1.1.6)
CONTACT_NAME	VARCHAR(255)	SNMP system.sysContact (i.e. .1.3.6.1.2.1.1.4)
DOMAIN_NAME_SUFFIX	VARCHAR(191)	The domain name suffix associated with this network adapter (i.e. foo.lexmark.com). Combine with HOSTNAME to get the Fully Qualified Domain Name (FQDN)
HOSTNAME	VARCHAR(63)	The hostname associated with this network adapter; MVE can be configured to retrieve this from DNS or from the network adapter itself. Combine with DOMAIN_NAME_SUFFIX to get the Fully Qualified Domain Name (FQDN)
IP_ADDRESS	VARCHAR(15)	The string representation of this network adapter s IP address; deprecated
IP_ADDRESS_INT	INTEGER	The integer representation of this network adapter s IP address
IP_ADDRESS_SUBNET	INTEGER	The integer representation of the subnet

		on which this network adapter resides
MAC_CANONICAL	VARCHAR(12)	The network adapter s MAC address, in canonical format
PORTS	INTEGER	The number of ports this network adapter supports; for now, always 1
RAND_MAC	SMALLINT	Flag indicating whether MAC_CANONICAL s current value was randomly generated. This will be true for a printer that has been imported or manually added and not yet discovered/audited.
CREDENTIAL_REQUIRED	SMALLINT	Flag indicating whether a credential is necessary in order to communicate with the associated printer
CREDENTIAL_PASSWORD	BLOB SUB_TYPE 0	Credential s encrypted password, if set
CREDENTIAL_PIN	BLOB SUB_TYPE 0	Credential s encrypted PIN, if set
CREDENTIAL_REALM	VARCHAR(64)	Credential s realm, if set
CREDENTIAL_USERNAME	VARCHAR(15)	Credential s username, if set
PORT_CONFIG_LST_TCP_OPEN	SMALLINT	Flag indicating whether this port on the associated printer is open
PORT_CONFIG_LST_UDP_OPEN	SMALLINT	Flag indicating whether this port on the associated printer is open
PORT_CONFIG_MDNS_OPEN	SMALLINT	Flag indicating whether this port on the associated printer is open
PORT_CONFIG_NPA_TCP_OPEN	SMALLINT	Flag indicating whether this port on the associated printer is open
PORT_CONFIG_NPA_UDP_OPEN	SMALLINT	Flag indicating whether this port on the associated printer is open
PORT_CONFIG_RAW_PRINT_OPEN	SMALLINT	Flag indicating whether this port on the associated printer is open
PORT_CONFIG_SNMP_OPEN	SMALLINT	Flag indicating whether this port on the associated printer is open
PORT_CONFIG_XML_TCP_OPEN	SMALLINT	Flag indicating whether this port on the associated printer is open
PORT_CONFIG_XML_TCP_OPEN	SMALLINT	Flag indicating whether this port on the associated printer is open
SECURE_COMMUNICATION_STATE	VARCHAR(255)	One of: UNSECURED, MISSING_CREDENTIALS, or SECURED

5.1.3 NETWORK_PRINTER

This table represents the actual printer portion of the physical printer.

Field Name	Data Type	Description
PRINTER_ID	BIGINT	Primary key
MANUFACTURER	VARCHAR(255)	Company that actually made the printer; may differ from DISPLAY_MANUFACTURER
MODEL_NAME	VARCHAR(255)	Model name of this printer

SERIAL_NUMBER	VARCHAR(50)	Serial number of this printer
AIO_CODE_LEVEL	VARCHAR(255)	Current All-In-One firmware revision
BASE_CODE_LEVEL	VARCHAR(255)	Current Base firmware revision
COLOR	VARCHAR(255)	The type of colour used by this model, for example, "CMYK Colour"
COPY	SMALLINT	Flag indicating whether this printer supports copying
DUPLEX	SMALLINT	Flag indicating whether this printer supports duplexing
ESF	SMALLINT	Flag indicating whether this printer supports eSF applications
MARKING_TECHNOLOGY	VARCHAR(255)	Type of marking technology used by this printer (i.e. Electrophotographic)
MEMORY	BIGINT	Amount of memory, in bytes
PROFILE	SMALLINT	Flag indicating whether this printer supports profiles
RECEIVE_FAX	SMALLINT	Flag indicating whether this printer supports receiving faxes
SCAN_TO_EMAIL	SMALLINT	Flag indicating whether this printer supports scan-to-email
SCAN_TO_FAX	SMALLINT	Flag indicating whether this printer supports scan-to-fax
SCAN_TO_NETWORK	SMALLINT	Flag indicating whether this printer supports scan-to-network
SPEED	VARCHAR(255)	Sheets per minute
DISPLAY_MANUFACTURER	VARCHAR(255)	The name that appears on the outside of the printer (i.e. MANUFACTURER could be "LEXMARK" but DISPLAY_MANUFACTURER could be "Dell")
ENGINE_CODE_LEVEL	VARCHAR(255)	Current Engine firmware revision
FAMILY_ID	INTEGER	NPA Family ID
FAX_CODE_LEVEL	VARCHAR(255)	Current Fax firmware revision
FONT_CODE_LEVEL	VARCHAR(255)	Current Font firmware revision
INITIAL_DISCOVERY_TIMESTAMP	TIMESTAMP	When this printer was first discovered
KERNEL_CODE_LEVEL	VARCHAR(255)	Current Kernel firmware revision
LIFETIME_PAGE_COUNT	BIGINT	Lifetime page count
LOADER_CODE_LEVEL	VARCHAR(255)	Current Loader firmware revision
MAINTENANCE_COUNTER	BIGINT	Maintenance counter
ADAPTER_PORT	INTEGER	The port on which this printer is connected to its associated network adapter; for now, will always be 1
NETWORK_CODE_LEVEL	VARCHAR(255)	Current Network firmware revision
NETWORK_DRIVER_CODE_LEVEL	VARCHAR(255)	Current Driver firmware revision
PANEL_CODE_LEVEL	VARCHAR(255)	Current Panel firmware revision
PRINTER_FIRMWARE_CODE_LEVEL	VARCHAR(255)	Current Printer firmware revision

PROPERTY_TAG	VARCHAR(255)	Asset/Brass/Property Tag
SCANNER_CODE_LEVEL	VARCHAR(255)	Current Scanner firmware revision
ADAPTER_ID	BIGINT	Foreign key to NETWORK_ADAPTER.ADAPTER_ID
RAND_SN	SMALLINT	Flag indicating whether SERIAL_NUMBER s current value was randomly generated. This will be true for a printer that has been imported or manually added and not yet discovered/audited.
DEV_STATUS_REG_COUNTER	INTEGER	A count of the number of device status registrations
SCANNER_SERIAL_NUMBER	VARCHAR(12)	For modular MFPs, the serial number of the scan unit

5.1.4 PRINTER_CURRENT_STATUS

This table represents status conditions that exist on printers at the point in time when data was collected. There will be a row in this table for each status condition on a given printer, all pointing to the same PRINTER_ID.

Field Name	Data Type	Description
STATUS_ID	BIGINT	Primary key
STATUS_MESSAGE	VARCHAR(255)	The text for this status. For example, "Tray 1 Low"
STATUS_SEVERITY	VARCHAR(255)	The severity of this status. For example, "Warning".
STATUS_TYPE	VARCHAR(255)	The type of this status. For example, "Printer" or "Supply".
PRINTER_ID	BIGINT	Foreign key to NETWORK_PRINTER.PRINTER_ID

5.1.5 PRINTER_ESF_APPS

This table represents eSF applications installed on printers. There will be a row in this table for each eSF application currently installed on a given printer, all pointing to the same PRINTER_ID.

Field Name	Data Type	Description
APPLICATION_ID	BIGINT	Primary key
NAME	VARCHAR(255)	Application name
STATE	VARCHAR(255)	Current state
VERSION	VARCHAR(255)	Current version
PRINTER_ID	BIGINT	Foreign key to NETWORK_PRINTER.PRINTER_ID

5.1.6 PRINTER_INPUT_OPTIONS

This table represents input options installed on printers. There will be a row in this table for each input option currently installed on a given printer, all pointing to the same PRINTER_ID.

Field Name	Data Type	Description
INPUT_OPTION_ID	BIGINT	Primary key
NAME	VARCHAR(255)	Name of the input option (i.e. Multi-purpose Tray)
PRINTER_ID	BIGINT	Foreign key to NETWORK_PRINTER.PRINTER_ID

5.1.7 PRINTER_INPUT_TRAYS

This table represents input trays associated with an input option. There will be a row in this table for each input tray associated with a given printer, all pointing to the same INPUT_OPTION_ID.

Field Name	Data Type	Description
INPUT_OPTION_ID	BIGINT	Foreign key to PRINTER_INPUT_OPTIONS.INPUT_OPTION_ID
CAPACITY	BIGINT	The maximum number of sheets this tray can hold
FEED_TYPE	VARCHAR(255)	Manual vs. Auto
FORM_SIZE	VARCHAR(255)	Current form size (i.e. Letter)
FORM_TYPE	VARCHAR(255)	Current form type (i.e. Plain Paper)
TYPE	VARCHAR(255)	For example, Multi-purpose Feeder

5.1.8 PRINTER_OPTIONS

This table represents options installed on printers. There will be a row in this table for each option currently installed on a given printer, all pointing to the same PRINTER_ID. Typically, this will be a storage device.

Field Name	Data Type	Description
OPTION_ID	BIGINT	Primary key
FREESPACE	BIGINT	Amount of space remaining
NAME	VARCHAR(255)	For example, DISK, FLASH, Random Access Memory, etc.
SIZE_	BIGINT	Total amount of space
PRINTER_ID	BIGINT	Foreign key to NETWORK_PRINTER.PRINTER_ID

5.1.9 PRINTER_OUTPUT_BINS

This table represents output bins associated with an output option. There will be a row in this table for each output bin associated with a given printer, all pointing to the same OUTPUT_OPTION_ID.

Field Name	Data Type	Description
OUTPUT_OPTION_ID	BIGINT	Foreign key to PRINTER_OUTPUT_OPTIONS.OUTPUT_OPTION_ID
BINDING	SMALLINT	Flag indicating whether this bin supports binding
BURSTING	SMALLINT	Flag indicating whether this bin supports bursting
CAPACITY	BIGINT	The maximum number of sheets this bin can hold
COLLATION	SMALLINT	Flag indicating whether this bin supports collation
FACE_DOWN	SMALLINT	Flag indicating whether this bin supports face down
FACE_UP	SMALLINT	Flag indicating whether this bin supports face up
LEVEL_SENSING	SMALLINT	Flag indicating whether this bin supports level sensing
PUNCHING	SMALLINT	Flag indicating whether this bin supports hole punching
SECURITY	SMALLINT	Flag indicating whether this bin supports security
SEPARATION	SMALLINT	Flag indicating whether this bin supports separation
STITICHING	SMALLINT	Flag indicating whether this bin supports stitching
TYPE	VARCHAR(255)	For example, Standard Bin, Bin 5, etc.

5.1.10 PRINTER_OUTPUT_OPTIONS

This table represents output options installed on printers. There will be a row in this table for each output option currently installed on a given printer, all pointing to the same PRINTER_ID.

Field Name	Data Type	Description
OUTPUT_OPTION_ID	BIGINT	Primary key
NAME	VARCHAR(255)	Name of the option (i.e. Integrated Hopper, Mailbox, Finisher, etc.)
PRINTER_ID	BIGINT	Foreign key to NETWORK_PRINTER.PRINTER_ID

5.1.11 PRINTER_PORTS

This table represents supported ports on printers. There will be a row in this table for each supported port on a given printer, all pointing to the same PRINTER_ID.

Field Name	Data Type	Description
PORT_ID	BIGINT	Primary key
NAME	VARCHAR(255)	For example, Standard Network, Standard USB, Internal 2, etc.
PRINTER_ID	BIGINT	Foreign key to NETWORK_PRINTER.PRINTER_ID

5.1.12 PRINTER_STATISTICS

This table contains information gathered from a printer's "Meters & Counters" data. Each row represents data for an individual printer. Depending on the model of printer with which the record is associated, not all columns will apply.

Field Name	Data Type	Description
STATISTICS_ID	BIGINT	Primary key
COVG_LAST_JOB_BLACK	BIGINT	Black toner coverage of the last job
COVG_LIFETIME_BLACK	BIGINT	Black toner coverage of lifetime jobs
CART_PAGES_PRINT_BLACK	BIGINT	Count of the pages printed that used this black toner cartridge
BLACK_TONER_LEVEL	VARCHAR(255)	Current supply level of the black toner cartridge
PHOTO_COND_LEVEL_K	VARCHAR(255)	Current supply level of the photo conductor (K)
BLANK_SAFE_SIDE_COPY	BIGINT	Count of the blank safe sides from a Copy
BLANK_SAFE_SIDE_FAX	BIGINT	Count of the blank safe sides from a Fax
BLANK_SAFE_SIDE_PRINT	BIGINT	Count of the blank safe sides from a Print
PAPER_CHANGE	BIGINT	Count of paper change events
COVER_OPEN	BIGINT	Count of cover open events
COVG_LAST_JOB_CYAN	BIGINT	Cyan toner coverage of the last job.
COVG_LIFETIME_CYAN	BIGINT	Cyan toner coverage of lifetime jobs
CART_PAGES_PRINT_CYAN	BIGINT	Count of the pages printed which used this cyan toner cartridge
CYAN_TONER_LEVEL	VARCHAR(255)	Current supply level of the cyan toner cartridge
CYAN_TONER_STATUS	VARCHAR(255)	Supply status for the cyan cartridge (i.e. Intermediate)
YELLOW_TONER_STATUS	VARCHAR(255)	Supply status for the yellow cartridge (i.e. Intermediate)
MAGENTA_TONER_STATUS	VARCHAR(255)	Supply status for the magenta cartridge (i.e. Intermediate)
BLACK_TONER_STATUS	VARCHAR(255)	Supply status for the black cartridge (i.e. Intermediate)

PHOTO_COND_LEVEL_C	VARCHAR(255)	Current supply level of the photo conductor (Cyan).
DEVICE_INSTALL_DATE	TIMESTAMP	TIMESTAMP of when the printer was first installed
FUSER_CURRENT_LEVEL	VARCHAR(255)	Current supply level of the fuser
IMG_SAFE_SIDE_COPY	BIGINT	Imaged Printed Sides – Copy
IMG_SAFE_SIDE_FAX	BIGINT	Imaged Printed Sides – Fax
IMG_SAFE_SIDE_PRINT	BIGINT	Imaged Printed Sides – Print
LAST_FAX_JOB_DATE	TIMESTAMP	TIMESTAMP of the last fax job
LAST_PRINTED_JOB_DATE	TIMESTAMP	TIMESTAMP of the last print job
LAST_SCAN_JOB_DATE	TIMESTAMP	TIMESTAMP of the last scan job
COVG_LAST_JOB_MAGENTA	BIGINT	Magenta toner coverage of the last job
COVG_LIFETIME_MAGENTA	BIGINT	Magenta toner coverage of lifetime jobs
CART_PAGES_PRINT_MAGENTA	BIGINT	Count of the pages printed which used this magenta toner cartridge
MAGENTA_TONER_LEVEL	VARCHAR(255)	Current supply level of the magenta toner cartridge
PHOTO_COND_LEVEL_M	VARCHAR(255)	Current supply level of the photo conductor (Magenta)
MAINT_KIT_LEVEL	VARCHAR(255)	Current supply level of the maintenance kit
MEDIA_SIZE_TYPE_MONO_SIDE_SAFE	BIGINT	Mono printed sides (safe)
MEDIA_SIZE_TYPE_COLOR_SIDE_SAFE	BIGINT	Colour printed sides (safe)
SUPPLY_EVENTS	BIGINT	Count of “other” supply events
PAPER_JAMS	BIGINT	Count of paper jam events
PAPER_LOAD	BIGINT	Count of paper load events
PRINT_SHEET_USE_PICKED	BIGINT	Printed sheets (picked)
PRINT_SIDE_USE_PICKED	BIGINT	Printed sides (picked)
POR	BIGINT	Count of power-ons
PRINT_AND_HOLD_JOB	BIGINT	Count of print-and-hold jobs
SAFE_SHT_COPY	BIGINT	Printed sheets (safe) – Copy
SAFE_SHT_FAX	BIGINT	Printed sheets (safe) – Fax
SAFE_SHT_PRINT	BIGINT	Printed sheets (safe) – Print
SCAN_PAPER_JAMS	BIGINT	Count of scanner jams
PRINTED_FROM_PRINT_AND_HOLD	BIGINT	Count of printed print-and-hold jobs
PRINTED_FROM_USB	BIGINT	Count of prints from USB
TRANS_BELT_LEVEL	VARCHAR(255)	Current supply level of the transfer belt
USB_DIRECT_JOB	BIGINT	Count of USB insertions
WASTE_TONER_LEVEL	VARCHAR(255)	Current supply level of the waste toner box
COVG_LAST_JOB_YELLOW	BIGINT	Yellow toner coverage of the last job

COVG_LIFETIME_YELLOW	BIGINT	Yellow toner coverage of lifetime jobs
CART_PAGES_PRINT_YELLOW	BIGINT	Count of the pages printed which used this yellow toner cartridge
YELLOW_TONER_LEVEL	VARCHAR(255)	Current supply level of the yellow toner cartridge
PHOTO_COND_LEVEL_Y	VARCHAR(255)	Current level of the photo conductor (Yellow)
IMG_SAFE_SIDE_PRINT_MONO	BIGINT	Imaged Mono Printed Sides (safe) – Print
IMG_SAFE_SIDE_PRINT_COLOR	BIGINT	Imaged Colour Printed Sides (safe) – Print
IMG_SAFE_SIDE_COPY_MONO	BIGINT	Imaged Mono Printed Sides (safe) – Copy
IMG_SAFE_SIDE_COPY_COLOR	BIGINT	Imaged Colour Printed Sides (safe) – Copy
IMG_SAFE_SIDE_FAX_MONO	BIGINT	Imaged Mono Printed Sides (safe) – Fax
IMG_SAFE_SIDE_FAX_COLOR	BIGINT	Imaged Colour Printed Sides (safe) – Fax
FAX_JOB_RECV	BIGINT	Received fax jobs
FAX_JOB_SENT	BIGINT	Sent fax jobs
FAX_PAGE_RECV	BIGINT	Received fax pages
FAX_PAGE_SENT	BIGINT	Sent fax pages
SCAN_COPY	BIGINT	Scans from Copy
SCAN_FAX	BIGINT	Scans from Fax
SCAN_LOCAL	BIGINT	Scan local
SCAN_NET	BIGINT	Scan to network
SCAN_FLAT	BIGINT	Scans from the flatbed
SCAN_ADF_SIMPLEX	BIGINT	Scans from the ADF (simplex)
SCAN_ADF_DUPLEX	BIGINT	Scans from the ADF (duplex)
SCAN_USB_DIRECT	BIGINT	Scans directly to USB
USB_DIRECT_INSERT	BIGINT	Count of USB insertions
CART_INST_DATE_CYAN	TIMESTAMP	TIMESTAMP of when this cyan cartridge was installed
CART_INST_DATE_YELLOW	TIMESTAMP	TIMESTAMP of when this yellow cartridge was installed
CART_INST_DATE_MAGENTA	TIMESTAMP	TIMESTAMP of when this magenta cartridge was installed
CART_INST_DATE_BLACK	TIMESTAMP	TIMESTAMP of when this black cartridge was installed
PRINTER_ID	BIGINT	Foreign key back to NETWORK_PRINTER.PRINTER_ID

5.1.13 PRINTER_SUPPLIES

This table represents supplies in printers. There will be a row in this table for each supply in a given printer, all pointing to the same PRINTER_ID. Depending on the TYPE, not all columns will apply.

Field Name	Data Type	Description
------------	-----------	-------------

SUPPLY_ID	BIGINT	Primary key
CAPACITY	BIGINT	Maximum sheet capacity of the supply
COLOR	VARCHAR(255)	For example, Black, Cyan, etc. Can be NULL
NAME	VARCHAR(255)	For example, Black Toner, Fuser, Waste Bottle
SMART_CARTRIDGE_PREBATE	SMALLINT	Flag indicating whether this supply is a smart cartridge prebate
SMART_CARTRIDGE_REFILLED	SMALLINT	Flag indicating whether this supply is a smart cartridge refill
SMART_CARTRIDGE_SERIAL_NUMBER	VARCHAR(255)	Smart Cartridge serial number
TYPE	VARCHAR(255)	For example, Toner, Transfer Belt, Fuser, Container, Image Unit, etc.
PRINTER_ID	BIGINT	Foreign key to NETWORK_PRINTER.PRINTER_ID
PERCENT_FULL	BIGINT	Calculated percent of the supply remaining.

5.2 Printer History

Each XXX_HISTORY table is a superset of the associated XXX table. The additions consist of two columns: REVTYPE and REV.

REVTYPE can currently have three values: 0, 1, and 2, which mean, respectively, ADD, MOD, and DEL. A row with a DEL revision type will contain only the ID of the entity and no data (all fields NULL) because it serves only as a marker that indicates "this entity was deleted at that revision."

REV is a foreign key to REVINFO.REV.

The J_XXX_HISTORY tables are special history tables for the PRINTER_XXX join tables. They contain REVTYPE and REV like the other XXX_HISTORY tables, but they also contain two pointers:

- PRINTER_ID, which points back to NETWORK_PRINTER.PRINTER_ID
- NNN_ID, which points back to PRINTER_XXX.NNN_ID. For example, J_PRN_ESFAPP_HISTORY.APPLICATION_ID points back to PRINTER_ESF_APPS.APPLICATION_ID.

The composite primary key for each of these tables will be REV and the two pointer columns.

The FRAMEWORK database consists of the following XXX_HISTORY tables.

- CONFIG_ITEM_HISTORY
- J_PRN_CURRENT_STATUS_HISTORY
- J_PRN_ESFAPP_HISTORY
- J_PRN_FONT_HISTORY
- J_PRN_INPUTOPTION_HISTORY
- J_PRN_OPTION_HISTORY
- J_PRN_OUTPUTOPTION_HISTORY
- J_PRN_PORT_HISTORY
- J_PRN_STATISTICS_HISTORY
- J_PRN_SUPPLY_HISTORY
- NETWORK_ADAPTER_HISTORY

- NETWORK_PRINTER_HISTORY
- PRINTER_CURRENT_STATUS_HISTORY
- PRINTER_ESF_APPS_HISTORY
- PRINTER_FONTS_HISTORY
- PRINTER_INPUT_OPTIONS_HISTORY
- PRINTER_OPTIONS_HISTORY
- PRINTER_OUTPUT_OPTIONS_HISTORY
- PRINTER_PORTS_HISTORY
- PRINTER_STATISTICS_HISTORY
- PRINTER_SUPPLIES_HISTORY

5.2.1 REVINFO

This table contains only two fields: the revision ID and the revision TIMESTAMP. A row is inserted into this table on each new revision; that is, on each commit of a transaction that changes data.

Field Name	Data Type	Description
REV	INTEGER	Primary key
REVTSTMP	BIGINT	The TIMESTAMP of this revision

5.3 Keywords

The following tables deal with MVE s keywords.

5.3.1 ASSIGNED_KEYWORDS

This table tells which keyword(s) are assigned to which CIs/printers.

Field Name	Data Type	Description
KEYWORD_ID	BIGINT	Composite primary key; foreign key to KEYWORD.KEYWORD_ID
CI_ID	BIGINT	Composite primary key; foreign key to CONFIGURATION_ITEM.CI_ID

5.3.2 KEYWORD

This table contains all the keywords defined in the system.

Field Name	Data Type	Description
KEYWORD_ID	BIGINT	Primary key
KEYWORD_VALUE	VARCHAR(255)	The keyword name
CATEGORY_ID	BIGINT	Foreign key to KEYWORD_CATEGORY.CATEGORY_ID

5.3.3 KEYWORD_CATEGORY

This table contains all the categories defined in the system. It is used for grouping keywords together.

Field Name	Data Type	Description
CATEGORY_ID	BIGINT	Primary key
CATEGORY_VALUE	VARCHAR(255)	The category name

5.4 Policies

The following tables deal with MVE s policies.

5.4.1 ASSIGNED_POLICIES

This table tells which policies are assigned to which CIs/printers.

Field Name	Data Type	Description
CI_ID	BIGINT	Composite primary key; foreign key back to CONFIGURATION_ITEM.CI_ID
POLICY_ID	BIGINT	Composite primary key; foreign key back to POLICY.POLICY_ID
CONFORMANCE_STATE	VARCHAR(255)	Current conformance state for this policy
LAST_CONFORMANCE_CHECK	TIMESTAMP	When the last conformance check was run

5.4.2 FAILED_SETTINGS

This table maintains any settings that are out of conformance so that this information can be communicated to clients.

Field Name	Data Type	Description
FAILED_SETTINGS_ID	BIGINT	Primary key
SETTINGNAME	VARCHAR(255)	The name of the out-of-conformance setting
SETTINGVALUE	VARCHAR(255)	The encrypted out-of-conformance value from the physical printer; not available outside of MVE
CI_ID	BIGINT	Foreign key back to ASSIGNED_POLICIES.CI_ID
POLICY_ID	BIGINT	Foreign key back to ASSIGNED_POLICIES.POLICY_ID

5.4.3 FLASH_SUBSYSTEMS

This table deals more granularly with the contents of imported flash files.

Field Name	Data Type	Description
FLASHFILEID	BIGINT	Composite Primary key; foreign key back to FLASHFILE.ID
REVISION	VARCHAR(255)	RIP or NET level, depending on the

		value of SUBSYSTEM
SUFFIX	VARCHAR(255)	Optional
SUBSYSTEM	VARCHAR(64)	Composite Primary key; one of FIRMWARE__NETWORK_CODE or FIRMWARE__PRINTER_CODE

5.4.4 FLASHFILE

This table deals with storing information retrieved from a flash file s header.

Field Name	Data Type	Description
ID	BIGINT	Primary key
FFID	VARCHAR(56)	Flash File ID. It uniquely identifies which printer family the flash file applies to.
RIP	VARCHAR(56)	RIP level of this flash file
NET	VARCHAR(56)	NET level of this flash file
MODIFYDATE	VARCHAR(32)	Date the flash file was created
FILENAME	VARCHAR(256)	The file name/location within MVE s repository

5.4.5 POLICY

This table defines the policy at the highest-level: a name and what model it applies to.

Field Name	Data Type	Description
POLICY_ID	BIGINT	Primary key
MODEL	VARCHAR(255)	The printer model to which this policy applies
NAME	VARCHAR(255)	A name for this policy
TYPE	VARCHAR(255)	The setting domain that this policy covers
DISCRIMINATOR	VARCHAR(255)	NON_SECURITY or SECURITY
CREDENTIAL_PASSWORD	BLOB SUB_TYPE 0	Credential s encrypted password, if set
CREDENTIAL_PIN	BLOB SUB_TYPE 0	Credential s encrypted PIN, if set
CREDENTIAL_REALM	VARCHAR(64)	Credential s realm, if set
CREDENTIAL_USERNAME	VARCHAR(255)	Credential s username, if set

5.4.6 POLICY_SETTINGS

This table represents settings contained within a given policy/type. There will be a row in this table for each setting associated with the policy/type, all pointing to the same POLICY.POLICY_ID. The values are encrypted and not available outside of MVE.

Field Name	Data Type	Description
------------	-----------	-------------

SETTING_ID	BIGINT	Primary key
SETTING_NAME	VARCHAR(255)	The name of a setting included in a policy
SETTING_VALUE	VARCHAR(255)	The value to associate with the above named setting
POLICY_ID	BIGINT	Foreign key back to POLICY.POLICY_ID

5.5 Discovery Profiles

The following tables are used to track MVE s discovery profiles.

5.5.1 DISCOVERY_PROFILE

This table represents the heart of MVE s discovery profile.

Field Name	Data Type	Description
ID	BIGINT	Primary key
NAME	VARCHAR(255)	User-supplied name for the profile
RETRIES	INTEGER	Number of times to retry a particular communication attempt with a printer
SNMP_READ_COMMUNITY_NAME	VARCHAR(255)	SNMP Community Name to use when reading
SNMP_WRITE_COMMUNITY_NAME	VARCHAR(255)	SNMP Community Name to use when writing
TIMEOUT	BIGINT	Number of milliseconds to wait for a particular communication attempt with a printer to succeed
INCLUDE_IN_SECURE_MODE	SMALLINT	Flag indicating whether restricted printers should be searched for during discovery

5.5.2 DISCOVERY_PROFILE_CI

This table contains the CI-specific pieces of the discovery profile.

Field Name	Data Type	Description
CI_DP_ID	BIGINT	Primary key; foreign key back to DISCOVERY_PROFILE.ID
AUTOMANAGE	SMALLINT	Flag indicating whether CIs discovered using this profile should be automatically managed

5.5.3 EXCLUDE_PROFILE_ITEM

This table represents the “exclude” list for a profile. Each excluded item will be a row in this table.

Field Name	Data Type	Description
DISCOVERY_PROFILE_ID	BIGINT	Composite primary key; foreign key back to DISCOVERY_PROFILE.ID
VALUE_	VARCHAR(255)	Composite primary key. This defines what to exclude.

5.5.4 INCLUDE_PROFILE_ITEM

This table represents the “include” list for a profile. Each included item will be a row in this table.

Field Name	Data Type	Description
DISCOVERY_PROFILE_ID	BIGINT	Composite primary key; foreign key back to DISCOVERY_PROFILE.ID
VALUE_	VARCHAR(255)	Composite primary key. This defines what to include.

5.6 Authentication and Authorization

The following tables are used by MVE's user authentication and authorization mechanism.

5.6.1 MASTER_ROLE

This table contains all the roles supported by MVE.

Field Name	Data Type	Description
ID	BIGINT	Primary key
ROLE_NAME	VARCHAR(255)	Name of the role

5.6.2 USERS

This table holds all of MVE s internal user accounts.

Field Name	Data Type	Description
ID	BIGINT	Primary key
USER_NAME	VARCHAR(15)	User-supplied username
USER_PASS	VARCHAR(32)	User-supplied password
ENABLED	SMALLINT	Flag indicating whether this account is currently enabled
NAME	VARCHAR(255)	User-supplied full name
LAST_LOGIN	TIMESTAMP	When the last login was attempted
LOGIN_ATTEMPT	BIGINT	Current number of attempts made at a successful login

5.6.3 USER_ROLE

This table holds the association of users to roles.

Field Name	Data Type	Description
ID	BIGINT	Primary key
USER_NAME	VARCHAR(15)	Foreign key back to USERS.USER_NAME
ROLE_NAME	VARCHAR(30)	Foreign key back to MASTER_ROLE.ROLE_NAME

5.7 Security Policy

The following tables deal with information related to MVE s security policies. However, as the Security Policy information is encrypted for data safety, unavailable outside of MVE and not useful in the scope of this document, the details of these tables will be omitted.

- ACCESS_CONTROL
- AUTH_GROUP
- BUILDING_BLOCK
- BUILDING_BLOCK_SETTINGS
- INTERNAL_ACCOUNT
- INTERNAL_ACCOUNT_GROUPS
- INTERNAL_ACCOUNT_SETTINGS
- POLICY_MISC_SETTINGS
- SECURITY_TEMPLATE
- SECURITY_TEMPLATE_BBS
- SECURITY_TEMPLATE_GROUPS

5.8 Event Manager

The following tables deal with information related to creating and managing events.

5.8.1 ALERT

This table contains all of the alerts MVE supports.

Field Name	Data Type	Description
ID	BIGINT	Primary key
NAME	VARCHAR(255)	The textual name of the alert. For example "Supply Alert"
SEVERITY	VARCHAR(255)	For example, "ERROR"
CATEGORY	VARCHAR(255)	For example, "SUPPLIES"

5.8.2 ASSIGNED_EVENTS

This table links Events with their assigned Configuration Items.

Field Name	Data Type	Description
CI_ID	BIGINT	Composite primary key; refers to CONFIG_ITEM.CI_ID
EVENT_ID	BIGINT	Composite primary key; refers to EVENT.EVENT_ID
EVENT_REGISTRATION_STATE	VARCHAR(255)	One of: REGISTERED or NOT_REGISTERED

5.8.3 DESTINATION

This table represents a Destination within the Event Manager module.

Field Name	Data Type	Description
DESTINATION_TYPE	VARCHAR(31)	The type, currently either email or shell/command. Depending on the type, not all columns will apply.
ID	BIGINT	Primary key
NAME	VARCHAR(255)	User-supplied name of the destination
EMAIL_BODY	VARCHAR(255)	Email body text
EMAIL_CC	VARCHAR(255)	Email CC list
EMAIL_FROM	VARCHAR(255)	Email From text
EMAIL_SUBJECT	VARCHAR(255)	Email Subject text
EMAIL_TO	VARCHAR(255)	Email To text
COMMAND_PATH	VARCHAR(255)	Full path to command to execute
COMMAND_PARAMS	VARCHAR(255)	Any parameters to send to the command

5.8.4 EVENT

This table contains user-created events, which consist of a name, a description and a collection of alerts to include.

Field Name	Data Type	Description
NAME	VARCHAR(255)	User-supplied name of the event
DESCRIPTION	VARCHAR(255)	User-supplied description of the event
EVENT_ID	BIGINT	Primary key
REPETITION_UNIT	VARCHAR(255)	Not used currently.
REPETITION_VALUE	INTEGER	Not used currently.
TRIGGER_DESTINATIONS	VARCHAR(255)	One of: on_active_onlyl or on_active_and_clear

5.8.5 EVENT_ALERTS

This table links an Event to the collection of alerts it includes.

Field Name	Data Type	Description
EVENT_ID	BIGINT	Composite primary key; refers to EVENT.EVENT_ID
ALERT_ID	BIGINT	Composite primary key; refers to ALERT.ALERT_ID

5.8.6 EVENT_DESTINATIONS

This table links an Event to an associated Destination.

Field Name	Data Type	Description
EVENT_ID	BIGINT	Composite primary key; refers to EVENT.EVENT_ID
DESTINATION_ID	BIGINT	Composite primary key; refers to DESTINATION.DESTINATION_ID

5.8.7 PRINTER_EVENT_ACTIVE_CONDITIONS

This table represents the active conditions/alerts for printers with events that trigger that condition/alert. Multiple conditions would be multiple rows, all pointing to the same PRINTER_ID.

Field Name	Data Type	Description
ACTIVE_CONDITION_ID	BIGINT	Primary key
LOCATION	VARCHAR(255)	For example, "Tray 1"
MESSAGE	VARCHAR(255)	For example, "Tray Missing"
TYPE	VARCHAR(255)	For example, "Intervention Required"
PRINTER_ID	BIGINT	Foreign key; refers to NETWORK_PRINTER.PRINTER_ID

5.9 Miscellaneous

The following tables provide useful storage but do not fit into any of the previous table categories. They are basically “one off” tables.

5.9.1 APPLICATION_SETTINGS

This table currently holds various “user preferences.”

Field Name	Data Type	Description
ID	BIGINT	Primary key
SETTING_KEY	VARCHAR(255)	Preference name
SETTING_VALUE	VARCHAR(255)	Preference value

5.9.2 BOOKMARK

This table contains all of MVE s bookmarks. They are currently stored as BLOBs; therefore, they cannot be edited outside of MVE.

Field Name	Data Type	Description
ID	BIGINT	Primary key
DEFAULT_SEARCH	SMALLINT	Flag indicating whether this bookmark is one of the defaults that ships with MVE
NAME	VARCHAR(255)	User-supplied name of the bookmark
SEARCH_CRITERIA	BLOB SUB_TYPE 0	The binary representation of the bookmark

5.9.3 CHANGED_SETTINGS

This table contains information about settings that changed between the last two Audits.

Field Name	Data Type	Description
ID	BIGINT	Primary key
CI_ID	BIGINT	Refers to CONFIG_ITEM.ID
SETTING_NAME	VARCHAR(255)	The name of the setting that changed.
CHANGE_TYPE	VARCHAR(255)	One of: ADD, UPDATE, REMOVE

5.9.4 DATABASECHANGELOG

This table is used by Liquibase, a schema migration tool. It contains a list of all the statements that have been run against the database to get the schema into the current state.

5.9.5 DATABASECHANGELOGLOCK

This table is used by Liquibase, as well. It is used to make sure two machines do not attempt to modify the database at the same time.

5.9.6 SMTP_CONFIGURATION

This table contains configuration for the Simple Mail Transfer Protocol, which allows MVE users, when necessary, to send emails. Currently, there will be only one row.

Field Name	Data Type	Description
ID	BIGINT	Primary key
FROM_ADDRESS	VARCHAR(255)	The email address that sent emails should be "from."
LOGIN_ID	VARCHAR(255)	User ID for the SMTP server
LOGIN_PASSWORD	VARCHAR(255)	Password associated with the user ID for the SMTP server
LOGIN_REQ	SMALLINT	Flag indicating whether or not the SMTP server requires a login
SMTP_PORT	BIGINT	The port of the SMTP server
SMTP_SERVER	VARCHAR(255)	The hostname/IP of the SMTP server

5.9.7 SYSTEM_LOG

This table can get very large. It contains all of the system log messages that are produced as MVE carries out its tasks.

Field Name	Data Type	Description
LOG_ID	BIGINT	Primary key
TIMESTAMP_	TIMESTAMP	Time the message was logged
TASKID	BIGINT	Task instance that generated the message
TASKNAME	VARCHAR(50)	Task that generated the message
LEVEL_	INTEGER	DEBUG, INFO, etc.
MESSAGE_	VARCHAR(8000)	The actual log message

6. MONITOR Tables and Field Descriptions

MONITOR is a smaller database where task statistics are stored. Currently only the TASK_INFO table is described in this document.

6.1 TASK_INFO

This table holds all task execution information. Each task submitted to the MVE server becomes an entry in this table.

Field Name	Data Type	Description
TASK_ID	BIGINT	Primary key
HOST_NAME	VARCHAR(150)	Hostname of the client that submitted the task
TASK_NAME	VARCHAR(80)	Name of the submitted task
SOLUTION_NAME	VARCHAR(80)	N/A
STATUS	VARCHAR(30)	Running, Completed, Failed
START_TIME	TIMESTAMP	Task execution begin time
END_TIME	TIMESTAMP	Task execution end time
PERCENT_COMPLETE	INTEGER	The task s current progress
USER_ID	VARCHAR(80)	N/A
PRINTER	VARCHAR(80)	N/A
ADDRESS	BIGINT	N/A
SCHEDULE_ID	VARCHAR(80)	This will be filled in if the task was run via scheduling
PARENT_ID	BIGINT	ID of this task s parent if it has one
THREAD_NAME	VARCHAR(80)	Name of the thread executing the task

7. Historical Data

MVE retains snapshots of selected data elements using a generated revision number and TIMESTAMP. This information is added to a duplicated table layout containing the data elements of interest. A snapshot of the modified data element can be obtained for any point in time, providing customers the ability to determine any changes that occurred.

7.1 Historical Data Generation

Whenever audited data elements are created, updated or deleted, MVE will generate a new revision number and TIMESTAMP and store this information in the REVINFO table of the FRAMEWORK database. If data from multiple tables are impacted within the same transaction, then all affected data elements will be stamped with the same (new) revision number and a new row will be added to the corresponding table with a “_HISTORY” name. For example, if a printer audit is executed and the lifetime page count for the printer has changed since the last audit, then there will be a new revision generated and an entry added to the CONFIG_ITEM_HISTORY table. The following diagram illustrates this.

Pre-Audit

NETWORK_PRINTER

PRINTER_ID	LIFETIME_PAGE_COUNT
25	1377

NETWORK_PRINTER_HISTORY

PRINTER_ID	LIFETIME_PAGE_COUNT	REV	REVTYPE
25	1377	255	0

REVINFO

REV	REVTSTMP
255	1302289417361

Post-Audit**NETWORK_PRINTER**

PRINTER_ID	LIFETIME_PAGE_COUNT
25	1401

NETWORK_PRINTER_HISTORY

PRINTER_ID	LIFETIME_PAGE_COUNT	REV	REVTYPE
25	1377	255	0
25	1401	256	1

REVINFO

REV	REVTSTMP
255	1302289417361
256	1302293432231

Note the following in the diagram:

- A new history row was inserted into the NETWORK_PRINTER_HISTORY table during the audit, since the value for lifetime page count changed.
- A new revision was created in the REVINFO table (256).
- The REVTYPE value for the new history row is a 1. Valid values are: 0 for an insertion of new data, 1 for an update of existing data, and 2 for a deletion of existing data.
- The REVTSTMP value represents the number of milliseconds since the standard base time of January 1, 1970, 00:00:00 GMT (see the java.util.Date class for more description).

7.2 Querying Historical Data**7.2.1 Time-based Queries**

In order to query historical data for revisions occurring over time, it is necessary to programmatically determine a `TIMESTAMP` value for the specific point(s) in time. This can be accomplished in several ways (stored procedure, script, Java program, etc.). For example, a Java class could be written that creates `java.util.Calendar` instances representing a date range. Using the millisecond values of the start and end dates, the REVINFO table could then be queried to determine any changed revisions during this date range. A second query could then be executed against the XXX_HISTORY table of interest (e.g., NETWORK_PRINTER_HISTORY) to fetch NETWORK_PRINTER snapshots within that date range.

Example:

This query will retrieve all NETWORK_PRINTER snapshots representing updates to data elements falling in a date range of „4/8/2011 15:00:41.33 and „4/11/2011 13:35:24.955 . The REVTSTMP values correspond to millisecond values for the date range and were computed using a simple Java program.

```
SELECT *
FROM network_printer_history nph
WHERE nph.revtype = '1' and
nph.rev BETWEEN
```

```
(SELECT MIN(rev.rev) FROM revinfo rev WHERE rev.REVTSTMP >= '1302289241033') and  
(SELECT MAX(rev.rev) FROM revinfo rev WHERE rev.REVTSTMP <= '1302543324955' )
```