COVIANT Software

ADMINISTRATOR GUIDE
DIPLOMAT MANAGED FILE TRANSFER
SQL AUDIT DATABASE
VERSION 8.1.1

V8.1.1
SQL AUDIT DATABASE

License Agreement
NOTICE TO ALL USERS: CAREFULLY READ THE APPROPRIATE LEGAL AGREEMENT CORRESPONDING TO THE LICENSE YOU PURCHASED, WHICH SETS FORTH THE GENERAL TERMS AND CONDITIONS FOR THE USE OF THE LICENSED SOFTWARE. IF YOU DO NOT AGREE TO ALL OF THE TERMS SET FORTH IN THE AGREEMENT, DO NOT INSTALL THE SOFTWARE.

Diplomat products may NOT be downloaded or otherwise exported or re-exported to any parties in Cuba, Iran, Libya, North Korea, Sudan, or Syria. You agree not to directly or indirectly export or re-export (including by transmission) these Diplomat products to any parties in the above countries without first obtaining any required export license or governmental approval.

By downloading or using Diplomat products, you are agreeing to the foregoing and you are representing and warranting that you are not located in and are not a national or resident of Cuba, Iran, Libya, North Korea, Sudan, or Syria.

DIPLOMAT PRODUCTS CONTAIN ENCRYPTION TECHNOLOGY THAT IS CONTROLLED FOR EXPORT BY THE U.S. BUREAU OF INDUSTRY AND SECURITY UNDER THE EXPORT ADMINISTRATION REGULATIONS. IN ADDITION TO OTHER RESTRICTIONS DESCRIBED IN THIS DOCUMENT AND THE DIPLOMAT LICENSE AGREEMENT, YOU MAY NOT USE DIPLOMAT PRODUCTS, OR EXPORT DIPLOMAT PRODUCTS TO ANY PARTY WHERE YOU KNOW, OR HAVE GOOD REASON TO BELIEVE, THAT DIPLOMAT PRODUCTS MAY BE USED IN CONNECTION WITH THE PROLIFERATION OF NUCLEAR, CHEMICAL OR BIOLOGICAL WEAPONS OR MISSILES.

Diplomat products are classified under ECCN 5D992B.1 with CCATS # G049200 as of June 14, 2006 which authorizes these products for export and re-export under Section 742.15 (B) (2) of the Export Administration Regulations (*Review Requirement for Mass Market Encryption Commodities and Software Exceeding 64 Bits*).

Contacting Coviant Software LLC
Installation and configuration support is provided under warranty for 45 days from initial purchase, as well as under annual maintenance agreements. Email and phone support is available from 9 a.m. ET to 5 p.m. ET weekdays. If you require assistance, contact Coviant Software support as follows:

Voice: 781.210.3310 x2
Fax: 781.210.3313
Web: www.coviantsoftware.com
E-mail: support@coviantsoftware.com

Table of Contents

1 SQL Audit Overview

Diplomat MFT Enterprise Edition allows the capture of audit data in a SQL database. SQL audit is recommended for enterprises with a high volume of jobs, stringent audit requirements, and/or the need for custom report generation.

Performance Optimization	Diplomat MFT Enterprise Edition is recommended for customers that anticipate the need to track the results of over 100 jobs per week – or 5,000 file transfer jobs per year. Implementations of Diplomat MFT Standard Edition that use the standard audit capabilities may experience slower performance when handling high transaction volumes.
Stringent Audit Requirements	Diplomat MFT Enterprise Edition allows the user to treat the audit trail creation as a critical part of each job. If T*reat as Critical* is selected, all jobs are suspended if an audit problem is encountered.
Custom Report Generation	The SQL audit database created by Diplomat MFT Enterprise Edition can be accessed directly to generate custom reports. Diplomat MFT Standard Edition allows only standard report generation from the Reports menu.

Diplomat MFT Enterprise Edition also provides a feature for archiving audit records on a daily basis to optimize runtime and reporting performance – plus an Archive Now feature to archive records immediately. This feature is not intended to replace regular database maintenance for performance optimization.

2 Set-up

2.1 Database Set-up

Diplomat MFT supports a SQL database as part of the audit function. The SQL audit database has three tables to capture job, file, and user activity data and three tables in which to archive job, file, and user activity data plus a table that stores the DB version.

- **Jobs and Files Tables**
  Each time a Diplomat MFT job runs new records are written to the Jobs and Files tables in the SQL audit database.

- **User Activity Tables**
  Each time a user creates, updates, or deletes records in the Diplomat MFT transaction database and associated configuration files a new record is written to the User Activity table in the SQL audit database.

  NOTE: The user activity tables are NOT supported with Diplomat's built-in audit database. They are only supported as SQL tables.

- **Archive Jobs, Files, and User Activity Tables**
  If desired, Diplomat MFT can move records automatically from the Jobs, Files, and User Activity tables in the SQL audit database to the archive tables. The archive tables are identical in format to the Jobs, Files, and User Activity tables.

- **Diplomat MFT Database Version Table**
  This table stores the Diplomat MFT audit database version number. The version number is used by Diplomat MFT to ensure compatibility between the database and the Diplomat Managed File Transfer Service and Client.

2.1.1 SQL Databases Supported

Diplomat MFT supports MySQL, SQL Server and other ANSI SQL-92 compliant databases.

MySQL
MySQL is an open source database. Version 5.1 of the MySQL database server running on Windows is supported by Diplomat MFT and is available at http://dev.mysql.com/downloads/mysql/5.1.html. Instructions on how to set up MySQL on Windows for use with Diplomat MFT are provided in Appendix B. Most Linux releases include a MySQL installation, which may need to be started as a service.

SQL Server
Select *SQL Server* or *SQL Server ODBC*, if you are using a Microsoft SQL Server database.

Custom JDBC
Select *Custom JDBC* to use an ANSI SQL-92 compliant database with a JDBC driver.

2.1.2 SQL Table Creation

Diplomat MFT can create the SQL audit database tables the first time the database is accessed or the tables in the SQL database may be created independently. If you choose to create the tables on your own, the DDL for creating the tables is shown in a later section of this guide. Also, a file named DiplomatAuditDB.ddl containing the DDL is provided as part of the Diplomat MFT Service installation. If Diplomat MFT Service is installed in the default directory, DiplomatAuditDB.ddl is located at C:\Program Files\Coviant Software\Diplomat-j\SQLaudit for Windows systems and /opt/coviant/diplomat-j/SQLaudit for Linux installations.

2.1.3 Database Access and Permissions

Diplomat MFT allows, but does not require, a username and password for access to the MySQL, MS SQLServer, and Other ODBC audit databases. If a username and password are used, the account associated with the username must, unless *Skip Table Creation* is selected, have permission to create tables and write records to existing tables.

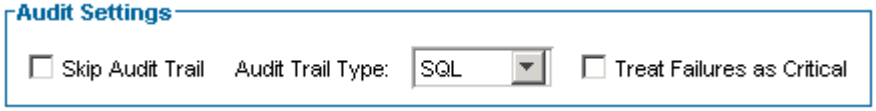
2.2 Settings

SQL audit and archival settings are entered on the Audit Trail Settings screen, which is accessed from the top menu bar of the Diplomat MFT Client by selecting Settings > Audit. An overview of the SQL audit and archival settings are provided below, as the Diplomat MFT user may need assistance from the SQL database administrator to properly enter these settings.

The Audit Trail Settings screen captures all information needed to set up and manage audit trail data, including the ability to automatically transfer SQL records to archive tables in the SQL audit database on a regular basis or immediately, if needed.

Audit trail data includes all data related to each file transfer job executed by Diplomat MFT that attempts to transfer files. Audit trail data is used to generate the Audit Detail Reports and the Audit Summary Reports available from the Reports menu item on the top menu bar. If a SQL database is used, user activity data is collected and a User Activity Report is also available.

2.2.1 Audit Settings



Audit settings determine whether or not Diplomat MFT captures audit data, what type of database is used, and what action to take if an error occurs during an attempt to write an audit record.

Skip Audit Trail

Check *Skip Audit Trail*, if you do not want audit records to be written. If you do not select *Skip Audit Trail*, an audit trail record is written for every job that is not automatically rescheduled due to File(s) Not Found (i.e., with a status of 'Success', 'Failure', or 'Warning', 'Error', or 'Critical Error').

NOTE: If *Fail if File Not Found* is checked on a transaction, then the job is a 'Failure' when the file is not found and an audit trail record is written. When *Fail if File Not Found* is NOT checked on a transaction, jobs that do not find files are simply rescheduled and no record is written to the audit trail.

Audit Trail Type

Diplomat MFT allows either a customizable SQL database or a built-in XML Diplomat MFT audit database. You can generate reports using the Reports menu item on the top menu bar for either type of audit database. If you want to create custom reports using a software product other than Diplomat, you must select 'SQL' and set up a SQL database to which Diplomat MFT can write audit records.

NOTE: If you select 'Built-in' as the *Audit Trail Type*, all fields on the remaining audit trail settings panels are disabled.

Treat Failures as Critical

Select *Treat Failure as Critical* to **SUSPEND ALL JOBS** when an audit trail problem occurs. Only select *Treat Failure as Critical* if an audit record is required for every file transfer job.

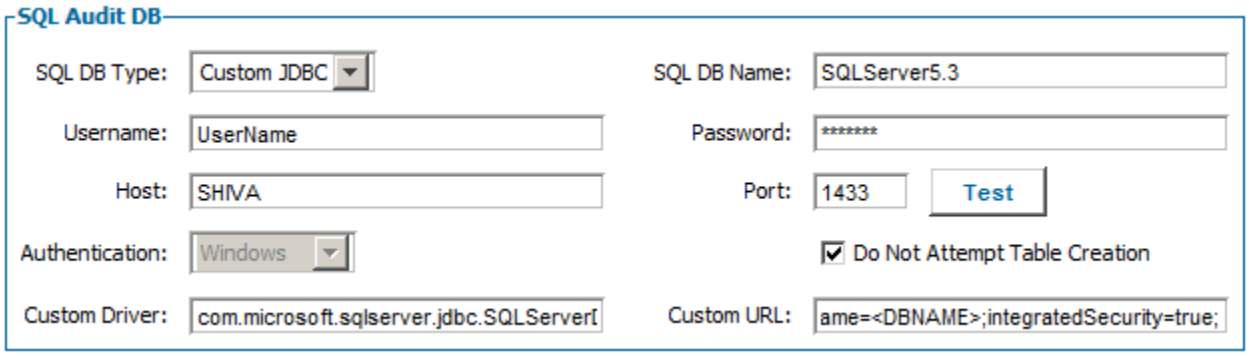
If *Treat Failure as Critical* is selected and an audit trail error occurs, job processing is suspended, which is indicated by pink status indicator '■' that is displayed next to the transactions folder in the navigation tree. In addition, an orange status indicator '■' is displayed next to all transaction objects in the tree. And, the audit trail error is treated as a critical error by email, paging, and logging.

Test jobs can be executed using '*Run Now*' to determine if an audit problem has been resolved. Once the problem has been resolved, release suspended transactions by selecting Jobs > Release > Release Critical Audit Suspend or right-click on the on the Transaction folder in the navigation tree and select **Release Critical Audit Suspend**.

If you have indicated that you want an audit trail, but it is not critical to your business (i.e., *Skip Audit Trail* is NOT checked and *Treat Failure as Critical* is NOT checked), and a job fails to write an audit record, then job processing continues. The audit trail error is treated as a critical error by email, paging, and logging.

NOTE: Email generated due to an audit trail failure is ONLY sent to IT Support. Business users do NOT receive any notification of an audit failure. If *Treat Failures as Critical* is selected, the failure email sent to IT Support includes the full contents of the records that would have been written to the audit database for the transaction in an XML format. If you have a stringent audit requirement, the data from this email can be entered manually into your SQL audit database or saved as an XML file.

2.2.2 SQL Audit DB



Contains all fields for setting up and using a SQL database for audit records. Each SQL audit database has three tables to capture job, file, and user activity data and three tables in which to archive job, file, and user activity data for improved performance, if desired.

NOTE: If you select 'Built-in' for *Audit Trail Type*, all fields on this panel are disabled.

NOTE: Changing SQL Audit DB settings while jobs are executing is potentially unsafe (e.g., audit records can be written without having their email and paging statuses set correctly). When prompted, you must select "Suspend" to suspend all jobs before updating the settings. If Diplomat MFT is unsuccessful in saving the new settings, all transactions will remain suspended. In addition, an orange status indicator '■' is displayed next to the transactions folder and all transaction objects in the navigation tree.

When setting up your SQL database, you must decide whether Diplomat MFT will be allowed to truncate data being written to character fields that are shorter than the string to be written. If Diplomat MFT does truncate data, a warning message and the complete string are written to the log file.

NOTE: This setting is NOT a Diplomat MFT setting, but must be made in the SQL database set-up.

SQL DB Type
Type of SQL database. Select Custom JDBC to use an ANSI SQL-92 compliant database with a JDBC driver.

NOTE: Linux systems do not support SQL Server or other ODBC databases. Only MySQL is supported for Linux implementations.

SQL DB Name
Name of SQL database used to capture audit records.

If you choose MySQL as your *SQL DB Type*, enter the name of the schema as it appears under Catalogs in the MySQL Administrator. If you choose SQL Server as your SQL DB Type, enter the name of the database as it appears under Databases in SQL Server Administrator

Username
If required, enter the username needed to access the SQL audit database.

NOTE: *Username* and *Password* fields are disabled when Windows Authentication is selected. The login account specified in the Diplomat MFT Service is used for Windows authentication. For detailed instructions on how to update the Diplomat MFT Service, see http://coviantsoftware.com/setting-windows-login.php.

Password
If required, enter the password needed to access the SQL audit database.

NOTE: *Username* and *Password* fields are disabled when Windows Authentication is selected. The login account specified in the Diplomat MFT Service is used for Windows authentication. For detailed instructions on how to update the Diplomat MFT Service, see http://coviantsoftware.com/setting-windows-login.php.

Host
Host name or IP address of the system where the SQL database is located.

NOTE: A login account on the Diplomat MFT Service must be specified when accessing SQL Server on a remote system. For detailed instructions on how to update the Diplomat MFT Service, see http://coviantsoftware.com/setting-windows-login.php.

Port
Specifies the port number used to access the SQL database. Default is 3306 for MySQL and 1433 for SQL Server.

NOTE: *Port* is not required for ODBC data sources.

Test Button
After entering the host and port information, press Test to test the connection to the SQL database.

NOTE: The test button only tests that the specified port is open on the host systems. It does NOT test the username and password for login to the database.

Authentication
When accessing a SQL Server database directly, select SQL Server or Windows authentication. If Windows authentication is selected, no *Username* or *Password* is required. Windows authentication uses the logon identity of the Diplomat MFT Service.

Do Not Attempt Table Creation
Each SQL audit database has three tables to capture job, file, and user activity data and three tables in which to archive job, file, and user activity data plus a table that stores the DB version. Select *Do Not Attempt Table Creation*, if you have already set up the seven tables required by Diplomat MFT in the SQL audit database. If you do NOT check *Do Not Attempt Table Creation* and if the tables do not already exist, Diplomat MFT attempts to create the seven required tables when the Audit Trail Settings are saved.

NOTE: If you do NOT check *Do Not Attempt Table Creation*, the account associated with the username and password specified above MUST have permission to create tables in the SQL database. If the account does not have the proper privileges, Diplomat MFT will NOT be able to create tables.

Custom Driver
Obtain a JDBC jar file from your SQL database vendor. Copy this jar file to C:\Program Files\Coviant Software\Diplomat-j\tomcatWebserver\webapps\diplomat\WEB-INF\lib\, opt/coviant/diplomat-j/Coviant Software/Diplomat-j/tomcatWebserver/webapps/diplomat/WEB-INF/lib or the corresponding directory for your installation.

Enter the JDBC driver class name in the JDBC jar (e.g., com.microsoft.sqlserver.jdbc.SQLServerDriver) in the *Custom Driver* field. Refer to the documentation from your SQL database vendor for more information.

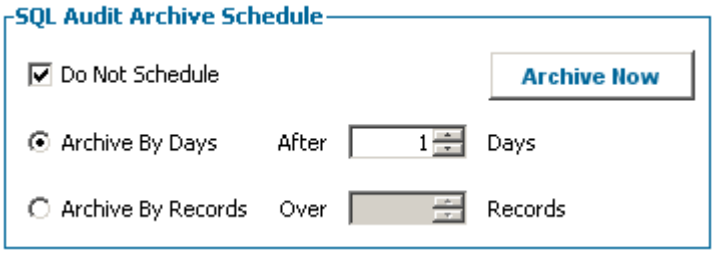
Custom URL

Connection URL associated with the specified *Custom Driver*. The **optional** parameters <HOST>, <PORT>, and <DBNAME> can be used in place of the host name, port number and SQL database name. At run-time, these parameters are replaced with the values in the *Host*, *Port*, and *SQL DB Name* fields.

At run-time, database authentication uses data from the *Username* and *Password* fields.

NOTE: Microsoft SQLServer also allows Windows authentication, which uses the logon identity associated with the Diplomat MFT Service. **If you are using a Microsoft SQLServer database, selection of *SQL Server* in the *SQL DB Type* field is recommended.**

2.2.3 SQL Audit Archive Schedule



Allows you to set-up automatic archival of audit records or to archive records immediately. Records are archived into the job, file, and user activity archive tables in the SQL audit database. Archiving of records is only available for SQL audit databases.

NOTE: Archiving records is only available for SQL audit databases. If you selected 'Built-in' for *Audit Trail Type*, all fields on this panel are disabled.

NOTE: Archiving SQL records may improve run-time job performance. However, performance may be adversely affected when generating reports that include archived records.

NOTE: When records are transferred to the archive tables in the SQL audit database, they are deleted from the active tables in the SQL audit database.

Do Not Schedule

Check *Do Not Schedule*, if do not want older audit records to be archived into separate SQL tables. If this field is not checked, then records are selected once a day based on the settings for *Archive by Date* or *Archive by Records* and written to the archive tables in the SQL audit database. Status of these daily jobs is shown in the *Archive Status* panel below.

Archive Now Button

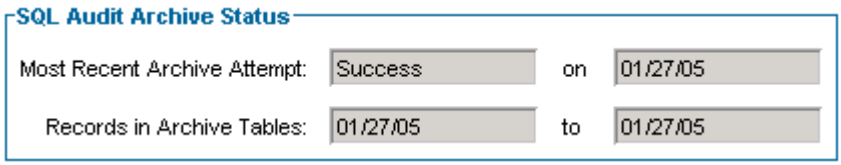
Archiving normally occurs when the Diplomat MFT Service is started and once a day thereafter. Press **Archive Now** to immediately execute a job to transfer SQL records to the archive tables in the SQL audit database, using the current settings on the *Audit Archive Schedule* panel. A pop-up dialog box displays the status of the archive process.

Archive by Days or Archive by Records

Audit records can be archived based on the number of days or records. If *Archive by Days* is selected, records older than the specified number of days are moved to the archive tables in the SQL audit database. If *Archive by Records* is selected, records in excess of the number of records specified are moved to the archive tables in the SQL audit database.

NOTE: All records for a day are moved as a block to the archive tables in the SQL audit database, even when *Archive by Records* is selected. Thus, the active and the archive audit databases never contain a partial day of records. And, since all records for a day are archived as a block, records for the current day are never archived using automatic archival or Archive Now.

2.2.4 SQL Audit Archive Status



Archiving of records is only available for SQL audit databases. If you selected 'Built-in' for *Audit Trail Type*, all fields on this panel are disabled.

Most Recent Archive Attempt
Date of the most recent attempt to transfer records to the archive tables in the SQL audit database and indicates whether the transfer was successful.

Records in Archive Tables
Range of creation dates for records in the archive tables in the SQL audit database. If not records are found in the archive tables, then 'No Records Archived' is displayed.

2.2.5 Audit Database Notices Button

Displays table with scheduled updates to SQL database fields. This table includes any fields scheduled for deletion. Fields scheduled for deletion in a subsequent release are typically no longer written in the current release.

3 Audit Database Tables

The Diplomat MFT SQL Audit Database captures data in six primary tables:

- JOB_AUDIT table contains a record for each job that either found one or more files to be transferred or was set to 'Fail if File Not Found'. The FILE_AUDIT table contains a record for each file found.
- JOB_AUDIT_ID column is the primary key and links the jobs in the JOB_AUDIT table with the files in the FILE_AUDIT table.
- JOB_AUDIT_ARCHIVE table contains all records that have been archived from the JOB_AUDIT table. The format of the JOB_AUDIT_ARCHIVE table is identical to the JOB_AUDIT table.
- FILE_AUDIT_ARCHIVE table contains all records that have been archived from the FILE_AUDIT table. The format of the FILE_AUDIT_ARCHIVE table is identical to the FILE_AUDIT table.
- USER_ACTIVITY table contains a record for each time a user creates, updates, or deletes a record in the Diplomat MFT transaction or configuration databases.
- USER_ACTIVITY_ARCHIVE table contains all records that have been archived from the USER_ACTIVITY table. The format of the USER_ACTIVITY _ARCHIVE table is identical to the USER_ACTIVITY table.

3.1 JOB_AUDIT and JOB_AUDIT_ARCHIVE Table (parent)

Data Field	Format	Description	Null Allowed	Restrictions (Field data valid ONLY IF)
JOB_AUDIT_ID	Text	Unique Identifier for each record.	No	
VERSION	Text	Version of Diplomat MFT which created record	No	
BUILD_NUMBER	Text	Build number of release being used	No	
OS_VERSION	Text	Reserved for future use	No	
DESCRIPTION	Text	Content of the *Description* field in the transaction	Yes	
TRANSACTION_ID	Text	Transaction Record ID	No	
TRANSACTION_TYPE	Text	Inbound or Outbound	No	

Data Field	Format	Description	Null Allowed	Restrictions (Field data valid ONLY IF)
STATUS	Text	Successful/Warning/Failure/Critical Failure Set to Failure if any portion of transaction fails. Should be the same as the status sent in email messages.	No	
STATUS_REASON	Text	Primary reason job was set to Warning, Failure, or Critical Failure. Should be the same reason as the status reason sent in email messages.	No	
OVERWRITE	Text	Overwrite settings for job. Values are: ▪ Overwrite "TRUE" ▪ Overwrite if source newer "DATE" ▪ Overwrite if different size "SIZE" ▪ Overwrite if source newer or different size "DT_SZ" ▪ Do not overwrite "FALSE"	No	
DELETE_SOURCE	Boolean		No	
LOG_LOCATION	Text	Path of log file directory	No	
LOG_FILENAME	Text	Name of log file	No	
ARCHIVE	Boolean	False if additional archive skipped on transaction screen.	No	
ARCHIVE_FILE_TYPE	Text	Source, Destination, or Both	Yes	ARCHIVE = True
ARCHIVE_LOCATION	Text	Path of additional archive directory from transaction screen	Yes	ARCHIVE =True
ARCHIVE_ZIP	Boolean	True, if additional archive files to be written as a single zip file	Yes	ARCHIVE =True

Data Field	Format	Description	Null Allowed	Restrictions (Field data valid ONLY IF)
ARCHIVE_ZIP_FILENAME	Text	Only valid if archived as single zip file.	Yes	ARCHIVE =True and ARCHIVE_ZIP =True
PRIMARY_ARCHIVE	Boolean	False if primary archive skipped	No	
PRIMARY_ARCHIVE_FILE_TYPE	Text	Source, Destination, or Both	Yes	PRIMARY_ARCHIVE = True
PRIMARY _ARCHIVE_LOCATION	Text	Path of primary archive directory	Yes	PRIMARY_ARCHIVE =True
PRIMARY _ARCHIVE_ZIP	Boolean	True, if primary archive files to be written as a single zip file	Yes	PRIMARY_ARCHIVE =True
PRIMARY_ARCHIVE_ZIP_FILENAME	Text	Only valid if archived as single zip file.	Yes	PRIMARY_ARCHIVE =True and PRIMARY_ARCHIVE_ZIP =True
START_TIME	DateTime	Date & time stamp based on system time of Diplomat MFT Service	No	
END_TIME	DateTime	Date & time stamp based on system time of Diplomat MFT Service	No	
SOURCE_PARTNER_ID	Text		No	
SOURCE_PARTNER_TYPE	Text	Public or Trusted	Yes	SOURCE_PARTNER_SAVED = True
SOURCE_PARTNER_SAVED	Boolean	True, if source partner profile was a 'saved' profile.	No	
SOURCE_TRANSPORT_METHOD	Text	Local Network, FTP, FTPS, SFTP, Email, HTTP, HTTPS, Cloud, Cloud PGP, Amazon S3, Azure Storage, Dropbox, Google Cloud, Oracle Cloud, ShareFile	No	
SOURCE_SERVER_ADDRESS	Text	IP, Domain or UNC	Yes	SOURCE_TRANSPORT_METHOD = Cloud, Cloud PGP, Email, FTP, FTPS, SFTP, HTTP, or HTTPS
SOURCE_SERVER_PORT	Integer		Yes	SOURCE_TRANSPORT_METHOD = Cloud, Cloud PGP, Email, FTP, FTPS, SFTP, HTTP, or HTTPS
SOURCE_SERVER_ACCOUNT	Text		Yes	SOURCE_TRANSPORT_METHOD = Email, FTP, FTPS, SFTP, HTTP, or HTTPS

Data Field	Format	Description	Null Allowed	Restrictions (Field data valid ONLY IF)
SOURCE_SERVER_DIRECTORY	Text		Yes	SOURCE_TRANSPORT_METHOD = Cloud, Cloud PGP, FTP, FTPS, SFTP, HTTP, or HTTPS
SOURCE_SERVER_TYPE	Text	Windows/Unix, AS400/IFS, AS400/Library	Yes	SOURCE_TRANSPORT_METHOD = FTP or FTPS
SOURCE_SERVER_PASSIVE	Boolean		Yes	SOURCE_TRANSPORT_METHOD = FTP or FTPS
SOURCE_FILE_LOCATION	Text		Yes	SOURCE_TRANSPORT_METHOD = Local Network
DEST_PARTNER_ID	Text		No	
DEST_PARTNER_TYPE	Text	Public or Trusted	Yes	DEST_PARTNER_SAVED = True
DEST_PARTNER_SAVED	Boolean	Yes, if destination partner profile was a 'saved' profile.	No	
DEST_TRANSPORT_METHOD	Text	Local Network, FTP, FTPS, SFTP, Email, HTTP, HTTPS, Cloud, Cloud PGP, Amazon S3, Azure Storage, Dropbox, Google Cloud, Oracle Cloud, ShareFile	No	
DEST_SERVER_ADDRESS	Text	IP, Domain or UNC	Yes	DEST_TRANSPORT_METHOD = Cloud, Cloud PGP, Email, FTP, FTPS, SFTP, HTTP, or HTTPS
DEST_SERVER_PORT	Integer		Yes	DEST_TRANSPORT_METHOD = Cloud, Cloud PGP, Email, FTP, FTPS, SFTP, HTTP, or HTTPS
DEST_SERVER_ACCOUNT	Text		Yes	DEST_TRANSPORT_METHOD = Email, FTP, FTPS, SFTP, HTTP, or HTTPS
DEST_SERVER_DIRECTORY	Text		Yes	DEST_TRANSPORT_METHOD = Cloud, Cloud PGP, FTP, FTPS, SFTP, HTTP, or HTTPS
DEST_RECIPIENT_ADDRESS	Text	Multiple email addresses delimited by commas	Yes	DEST _TRANSPORT_METHOD = Email
DEST_RECIPIENT_SUBJECT	Text		Yes	DEST _TRANSPORT_METHOD = Email
DEST_SERVER_TYPE	Text	Windows/Unix, AS400/IFS, AS400/Library	Yes	DEST_TRANSPORT_METHOD = FTP or FTPS

Data Field	Format	Description	Null Allowed	Restrictions (Field data valid ONLY IF)
DEST_SERVER_PASSIVE	Boolean		Yes	DEST_TRANSPORT_METHOD = FTP or FTPS
DEST_FILE_LOCATION	Text		Yes	DEST_TRANSPORT_METHOD = Local Network
ENCRYPT_DECRYPT	Boolean	True or False	No	
SIGN_AUTHENTICATE	Boolean	True or False	No	
ASCII_ARMORING	Boolean	True or False On Inbound transactions, remove armoring when True. On Outbound transactions, add armoring when True.	No	
CANONICAL_TEXT	Boolean		Yes	TRANSACTION_TYPE = Outbound
COMPRESSION	Boolean		Yes	TRANSACTION_TYPE = Outbound
SOURCE_FTP_MODE	Text	ASCII or Binary Used to set FTP mode of source file, if necessary; Set by Source File Format field in transaction	Yes	SOURCE_TRANSPORT_METHOD = FTP, FTPS or SFTP
DEST_FTP_MODE	Text	ASCII or Binary Used to set FTP mode of destination file, if necessary; Set by Destination File Format field in transaction	Yes	DEST_TRANSPORT_METHOD = FTP, FTPS or SFTP
RUN_NOW	Boolean	True, if job was executed using 'Run Now' in the Diplomat MFT Client or Job Monitor	No	
API	Boolean	True, if job was executed using the Diplomat MFT REST API	No	
FILE_MONITOR	Boolean	True, if job was executed using file monitoring	No	
THIRD_PARTY	Boolean	True, if job was executed using a Scripting Agent command	No	

Data Field	Format	Description	Null Allowed	Restrictions (Field data valid ONLY IF)
LINKED	Boolean	True, if job was executed as a post-process Linked Transaction	No	
SCRIPT_ARGUMENT1	Integer	Delay – # minutes between job submission and execution	Yes	THIRD_PARTY = True
SCRIPT_ARGUMENT2	Text	User ID – "Domain_name (if available) /Username" of process initiating scripting agent request	Yes	THIRD_PARTY = True
SCRIPT_ARGUMENT3	Boolean	User ID Authenticated - True if User ID of process initiating scripting agent request was authenticated	Yes	THIRD_PARTY = True
SCRIPT_ARGUMENT4	Text	IP Address – IP address of system from which scripting agent request originated	Yes	THIRD_PARTY = True
SCRIPT_ARGUMENT5	Boolean	Password Verified – True if password submitted with scripting agent request was verified	Yes	THIRD_PARTY = True
SCRIPT_ARGUMENT6	Text	Reserved for future use	Yes	THIRD_PARTY = True
POLLING_FREQUENCY	Text	Minutes, Hours, Days, Months	Yes	RUN NOW=False and THIRD_PARTY=False
POLLING_INTERVAL	Integer		Yes	RUN NOW=False and THIRD_PARTY=False
NUMBER_RETRIES	Integer	Maximum number of attempts allowed before rescheduling to next polling interval	Yes	RUN NOW=False and THIRD_PARTY=False and (POLLING_FREQUENCY = Months or POLLING_FREQUENCY = Days)
TOTAL_ATTEMPTS	Integer	N^{th} attempt to run this job	Yes	RUN NOW=False and THIRD_PARTY=False and (POLLING_FREQUENCY = Months or POLLING_FREQUENCY = Days)
BUSINESS_EMAIL	Boolean	Was there an attempt to send email to business users?	No	

Data Field	Format	Description	Null Allowed	Restrictions (Field data valid ONLY IF)
BUSINESS_EMAIL_ADDRESSES	Text	List of email addresses to which business email was sent successfully	Yes	BUSINESS_EMAIL = True
IT_EMAIL	Boolean	Was there an attempt to send debug email to IT?	No	
IT_EMAIL_ADDRESSES	Text	List of IT email addresses to which debug email was sent successfully	Yes	IT_EMAIL = True
PAGING_TYPE	Text	No Paging, Email, or File	No	
PAGING_LEVEL	Text	Minimum level to page: Warning, Error, or Critical Error	Yes	PAGING_TYPE ≠ No Paging
PAGING_PIN	Text		Yes	PAGING_TYPE ≠ No Paging
PAGING_ATTEMPTED	Boolean	Did we attempt to send the page?	Yes	PAGING_TYPE ≠ No Paging
PAGING_SUCCESSFUL	Boolean	Was page sent successfully?	Yes	PAGING_ATTEMPTED= True
PAGING_EMAIL_ADDRESS	Text		Yes	PAGING_TYPE = Email and PAGING_SUCCESSFUL = True
PRIMARY_PAGING_LOCATION	Text	Primary directory for paging files	Yes	PAGING_TYPE = File and PAGING_SUCCESSFUL = True
SECONDARY_PAGING_LOCATION	Text	Secondary directory for paging files	Yes	PAGING_TYPE = File and PAGING_SUCCESSFUL = True
PAGING_FILENAME	Text		Yes	PAGING_TYPE = File and PAGING_SUCCESSFUL = True
PRE_ZIP	Text	Description of zip process as first step of outbound jobs.	Yes	
POST_ZIP	Text	Description of zip process as last step of inbound jobs.	Yes	
PRE_COMMAND	Text	Command to be executed before the transaction is run	Yes	

Data Field	Format	Description	Null Allowed	Restrictions (Field data valid ONLY IF)
PRE_COMMAND_ATTEMPTED	Boolean	Was there an attempt to execute the pre-command?	Yes	PRE_COMMAND ≠ Null
PRE_COMMAND_RTRN_CODE	Integer	Return code from the PRE_COMMAND execution	Yes	PRE_COMMAND_ATTEMPTED = True
PRE_COMMAND_FAILURE_OVERRIDE	Boolean	Was the 'Continue on Before Command Failure' checkbox checked?	Yes	PRE_COMMAND ≠ Null
POST_COMMAND	Text	Command to be executed after the transaction is run	Yes	
POST_COMMAND_ ATTEMPTED	Boolean	Was there an attempt to execute the post-command?	Yes	POST_COMMAND ≠ Null
POST_COMMAND_RTRN_CODE	Integer	Return code from the POST_COMMAND execution	Yes	POST_COMMAND ATTEMPTED = True
POST_COMMAND_FAILURE_OVERRIDE	Boolean	Was the 'Execute After Command for Every Job' checkbox checked?	Yes	POST_COMMAND ≠ Null
LINKED_STATUS	Text	If executing job attempts to start a post-process linked transaction, scheduling status of that linked job	Yes	
CL_SERVER	Boolean	Was 3rd party command line server used for encryption/decryption?	No	CL_SERVER always set to "False" staring with v6.1.
CL_SERVER_TYPE	Text	Type of command line server: Authora, McAfee, PGP, Veridis	Yes	**Deprecated OpenPGP Command Line Console in v6.1. Data no longer written as of v7.2.** CL_SERVER = True
CL_SERVER_LOCATION	Text	Fully qualified path of directory containing command line server executable. Blank indicates default server command (e.g., "ebs" for the McAfee E-Business Server) used.	Yes	**Deprecated OpenPGP Command Line Console in v6.1. Data no longer written as of v7.2.** CL_SERVER = True

Data Field	Format	Description	Null Allowed	Restrictions (Field data valid ONLY IF)
CL_CONFIGURATION_FILE	Text	Fully qualified path of the configuration file or home directory. Blank indicates default configuration file or home directory used.	Yes	**Deprecated OpenPGP Command Line Console in v6.1. Data no longer written as of v7.2.** CL_SERVER = True
CL_ENCRYPT_DECRYPT_TYPE	Text	Type of encryption/decryption: OPENPGP, CONVENTIONAL, or SDA (encrypt only)	Yes	**Deprecated OpenPGP Command Line Console in v6.1. Data no longer written as of v7.2.** CL_SERVER = True
CL_ENCRYPTION_KEY_ID	Text	List of Key ID(s), User ID(s), or Groupname(s) used for encryption	Yes	**Deprecated OpenPGP Command Line Console in v6.1. Data no longer written as of v7.2.** CL_SERVER = True and TRANSACTION_TYPE = OUTBOUND and CL_ENCRYPT_DECRYPT_TYPE= OpenPGP and ENCRYPT_DECRYPT=True
CL_SIGN_VERIFY_KEY_ID	Text	User ID or Key ID of key used to sign (encrypt only) or verify (decrypt only)	Yes	**Deprecated OpenPGP Command Line Console in v6.1. Data no longer written as of v7.2.** CL_SERVER = TRUE and CL_ENCRYPT_DECRYPT_TYPE= OpenPGP and SIGN_AUTHENTICATE=True and (TRANSACTION_TYPE = INBOUND or CL_USE_DEFAULT_SIGNATURE_KEY = False)
CL_INC_DFLT_ENCRYPT_KEY	Boolean	Add CL server's default encryption key to CL_ENCRYPTION_KEY_ID list	Yes	**Deprecated OpenPGP Command Line Console in v6.1. Data no longer written as of v7.2.** CL_SERVER = True and TRANSACTION_TYPE = OUTBOUND and CL_ENCRYPT_DECRYPT_TYPE= OpenPGP and ENCRYPT_DECRYPT=True

Data Field	Format	Description	Null Allowed	Restrictions (Field data valid ONLY IF)
CL_USE_DFLT_SIG_KEY	Boolean	Use the CL server's default key to sign file	Yes	**Deprecated OpenPGP Command Line Console in v6.1. Data no longer written as of v7.2.** CL_SERVER = True and TRANSACTION_TYPE = OUTBOUND and CL_ENCRYPT_DECRYPT_TYPE= OpenPGP and SIGN_AUTHENTICATE=True and CL_SIGN_VERIFY_KEY_ID = Null
CL_DISCARD_PATHS	Boolean	Set switch in CL server command to discard path information when unzipping an archive	Yes	**Deprecated OpenPGP Command Line Console in v6.1. Data no longer written as of v7.2.** CL_SERVER = True and TRANSACTION_TYPE = OUTBOUND and (CL_ENCRYPT_DECRYPT_TYPE = SDA) and CL_SERVER_TYPE = supports "discard paths" feature

3.2 FILE_AUDIT and FILE_AUDIT_ARCHIVE Table (child)

Data Field	Format	Description	Null Allowed	Restrictions (Field data valid ONLY IF)
JOB_AUDIT_ID	Text	Unique Identifier for each record. References the JOB_AUDIT_ID field in the JOB_AUDIT table.	No	
SOURCE_FILENAME	Text	Name of file read with optional directories preceding it.	No	
SOURCE_SENDER_ADDRESS	Text		Yes	SOURCE_TRANSPORT_METHOD = Email
SOURCE_SENDER_SUBJECT	Text		Yes	SOURCE_TRANSPORT_METHOD = Email
SOURCE_XFERRED	Boolean	If source file is on the local network, was source file successfully accessed; if source file is on an FTP server, was source file successfully FTP'd,	No	
SOURCE_FILE_SIZE	Long	Number of bytes	No	
SOURCE_ASCII_ARMORED	Boolean	Was the inbound source file armored?	Yes	TRANSACTION_TYPE = Inbound
SOURCE_SIGNED	Boolean	Was inbound source file signed?	Yes	TRANSACTION_TYPE = Inbound and CL_SERVER = False
SOURCE_ENCRYPTED	Boolean	Was inbound source file encrypted?	Yes	TRANSACTION_TYPE = Inbound and CL_SERVER = False
SOURCE_DELETED	Boolean	Did we delete the source file?	No	
SOURCE_ATTEMPT_UNARMOR	Boolean	Did we attempt to remove ASCII-armoring from inbound source file?	Yes	TRANSACTION_TYPE = Inbound and SOURCE_ASCII_ARMORED = True and CL_SERVER = False
SOURCE_UNARMORED	Boolean	Did we remove ASCII-armoring from inbound source file successfully?	Yes	TRANSACTION_TYPE = Inbound and SOURCE_ATTEMPT_UNARMOR = True and CL_SERVER = False

Data Field	Format	Description	Null Allowed	Restrictions (Field data valid ONLY IF)
SOURCE_ATTEMPT_DECRYPT	Boolean	Did we attempt to decrypt inbound source file?	Yes	TRANSACTION_TYPE = Inbound and SOURCE_ENCRYPTED = True and CL_SERVER = False
SOURCE_DECRYPTED	Boolean	Did we decrypt inbound source file successfully?	Yes	TRANSACTION_TYPE = Inbound and SOURCE_ATTEMPT_DECRYPT = True and CL_SERVER = False
SOURCE_ATTEMPT_VERIFY	Boolean	Did we attempt to verify inbound source file?	Yes	TRANSACTION_TYPE = Inbound and SOURCE_SIGNED = True and CL_SERVER = False
SOURCE_VERIFIED	Boolean	Did we verify inbound source file successfully?	Yes	TRANSACTION_TYPE = Inbound and SOURCE_ATTEMPT_VERIFY = True and CL_SERVER = False
DECRYPT_KEYID	Hex	ID of key used to decrypt inbound source file	Yes	SOURCE_DECRYPTED = True
VERIFY_KEYID	Hex	ID of key used to verify inbound source file	Yes	TRANSACTION_TYPE = Inbound and SOURCE_ATTEMPT_VERIFY = True and CL_SERVER = False
DEST_WRITE_ATTEMPTED	Boolean	Did we attempt to write the destination file?	No	
DEST_WRITTEN	Boolean	Did we successfully write the destination file? If DEST_WRITTEN = True, transaction cannot be 'rolled back' even if a subsequent step of the job fails (e.g., archive or audit failures).	Yes	DEST_WRITE_ATTEMPTED = True
DEST_FILENAME	Text	File name with optional directories preceding it.	Yes	DEST_WRITTEN = True

Data Field	Format	Description	Null Allowed	Restrictions (Field data valid ONLY IF)
DEST_FILE_SIZE	Long	Number of bytes	Yes	DEST_WRITTEN = True If DEST_FILE_SIZE cannot be confirmed (e.g., using an FTP SIZE command), then DEST_FILE_SIZE is set to "-1".
DEST_COMPRESSED	Boolean	Did we compress the output file?	Yes	TRANSACTION_TYPE = Outbound and COMPRESSION = True and CL_SERVER = False
DEST_ATTEMPT_ARMOR	Boolean	Did we attempt to armor the output file?	Yes	TRANSACTION_TYPE = Outbound and CL_SERVER = False
DEST_ASCII_ARMORED	Boolean	Did we armor the output file?	Yes	TRANSACTION_TYPE = Outbound and DEST_ATTEMPT_ARMOR = True and CL_SERVER = False
DEST_ATTEMPT_SIGN	Boolean	Did we attempt to sign outbound file?	Yes	TRANSACTION_TYPE = Outbound and CL_SERVER = False
DEST_SIGNED	Boolean	Did we sign outbound file?	Yes	TRANSACTION_TYPE = Outbound and DEST_ATTEMPT_SIGN = True and CL_SERVER = False
DEST_ATTEMPT_ENCRYPT	Boolean	Did we attempt to encrypt outbound file?	Yes	TRANSACTION_TYPE = Outbound and CL_SERVER = False
DEST_ENCRYPTED	Boolean	Did we encrypt outbound file?	Yes	TRANSACTION_TYPE = Outbound and DEST_ATTEMPT_ENCRYPT = True and CL_SERVER = False
SIGNATURE_TIME	DateTime	Exact time of signature on outbound file	Yes	TRANSACTION_TYPE = Outbound and DEST_SIGNED = True and CL_SERVER = False
DEST_OVERWRITE	Boolean	Did we overwrite a file?	Yes	

Data Field	Format	Description	Null Allowed	Restrictions (Field data valid ONLY IF)
ENCRYPT_KEYID	Hex	ID of key used to encrypt outbound file	Yes	TRANSACTION_TYPE = Outbound and DEST_ENCRYPTED = True and CL_SERVER = False
AEK_ENCRYPT_KEYID	Hex	List of Key IDs for Additional Encryption Key(s) used to encrypt outbound file	Yes	TRANSACTION_TYPE = Outbound and DEST_ENCRYPTED = True and CL_SERVER = False
SIGNATURE_KEYID	Hex	ID of key used to sign outbound file	Yes	TRANSACTION_TYPE = Outbound and DEST_SIGNED = True and CL_SERVER = False
FILE_STATUS	Text	Success/Warning/Failure/Critical Failure (Individual files may succeed or fail within a 'failed' transaction.)	No	
FILE_STATUS_REASON	Text	Reason file failed or generated a warning.	No	
SOURCE_ARCHIVE_FILENAME	Text	Name of additional source file archived.	Yes	ARCHIVE_FILE_TYPE = 'Source' or 'Both'
PRIMARY_SRC_ARCHIVE_FILENAME	Text	Name of primary source file archived.	Yes	PRIMARY_ARCHIVE_FILE_TYPE = 'Source' or 'Both'
DEST_ARCHIVE_FILENAME	Text	Name of additional destination file archived.	Yes	ARCHIVE_FILE_TYPE = 'Destination or 'Both'
PRIMARY_DEST_ARCHIVE_FILENAME	Text	Name of primary destination file to be archived.	Yes	PRIMARY_ARCHIVE_FILE_TYPE = 'Destination or 'Both'
CL_COMMAND	Text	Command string executed	Yes	**Deprecated OpenPGP Command Line Console in v6.1. Data no longer written as of v7.2.** CL_SERVER = True

Data Field	Format	Description	Null Allowed	Restrictions (Field data valid ONLY IF)
CL_RETURN_CODE	Integer	Return code from command execution	Yes	**Deprecated OpenPGP Command Line Console in v6.1. Data no longer written as of v7.2.** CL_COMMAND ≠ blank
CL_RETURN_CODE_TEXT	Text	Descriptive text corresponding to return code	Yes	**Deprecated OpenPGP Command Line Console in v6.1. Data no longer written as of v7.2.** CL_RETURN_CODE ≠ blank

3.3 USER_ACTIVITY and USER_ACTIVITY _ARCHIVE Table

Data Field	Format	Description	Null Allowed	Restrictions (Field data valid ONLY IF)
SEQUENCE_NUMBER	Text	Unique Identifier for each USER_ACTIVITY record	No	
TIMESTAMP	DateTime	Date & time stamp based on system time from Diplomat MFT Service	No	
USER_ID	Text	Domain Name/User Name as returned by operating system	No	
USER_IP_ADDRESS	Text	IP address of user as returned by operating system	No	
OBJECT_TYPE	Text	Record type modified, includes OpenPGP Key Pair, OpenPGP Public Key, SSH Key, Partner, Transaction, User, Password, License, and various Settings	No	
OBJECT_ID	Text	ID of object modified	Yes	USER_TYPE = (Key or Partner or Transaction or User)
ACTION	Text	Action taken by user. Valid values include Add Subkey, Create, Delete, Import, Recover, Replace, Restore, and Update.	No	
COMMENT	Text	When available, further information about the action taken.	Yes	

4 Field Formats

All fields are stored in the database as string (VARCHAR) fields of various lengths (see the DDL Section). Diplomat MFT enters null values in all fields allowing null values when the field is not applicable.

The 'Format' column in the foregoing tables describes how each field should be interpreted. Those interpreted as 'Text' do not require any special interpretation. Other fields should be interpreted as follows:

Format	**Interpretation**
Boolean	True or false corresponding to the database text, which is 'true' or 'false'
DateTime	Database text is formatted as 'yyyyMMddhhmmssSSS' where: yyyy = the year; e.g. '2004' MM = the month as a decimal number in the range 1 (Jan) to 12 (Dec) dd = the day of the month, in the range 1 to 31 hh = hour, in the range 0 to 23 mm = minute, in the range 0 to 59 ss = second, in the range 0 to 59 SSS = millisecond, in the range 0 to 999
Hex	Database text is formatted as a 16-digit hexadecimal number
Integer	Database text is formatted as a signed decimal integer in the range -2^{31} to $2^{31}-1$
Long	Database text is formatted as a signed decimal integer in the range -2^{63} to $2^{63}-1$

5 DDL

The DDL for creating the required database tables is given below:

```
CREATE TABLE DB_VERSION (
      DBID INTEGER,
      VERSION VARCHAR(32) );

INSERT INTO DB_VERSION (DBID, VERSION) VALUES (1, "5.0");

CREATE TABLE JOB_AUDIT (
   JOB_AUDIT_ID                    VARCHAR(128) PRIMARY KEY NOT NULL,
   VERSION                         VARCHAR(32) NOT NULL,
   BUILD_NUMBER                    VARCHAR(8) NOT NULL,
   OS_VERSION                      VARCHAR(64) NOT NULL,
   DESCRIPTION                     VARCHAR(255),
   TRANSACTION_ID                  VARCHAR(128) NOT NULL,
   TRANSACTION_TYPE                VARCHAR(32) NOT NULL,
   STATUS                          VARCHAR(32) NOT NULL,
   STATUS_REASON                   VARCHAR(255) NOT NULL,
   OVERWRITE                       VARCHAR(5) NOT NULL,
   DELETE_SOURCE                   VARCHAR(5) NOT NULL,
   LOG_LOCATION                    VARCHAR(255) NOT NULL,
   LOG_FILENAME                    VARCHAR(64) NOT NULL,
   ARCHIVE                         VARCHAR(5) NOT NULL,
   ARCHIVE_FILE_TYPE               VARCHAR(32),
   ARCHIVE_LOCATION                VARCHAR(255),
   ARCHIVE_ZIP                     VARCHAR(5),
   ARCHIVE_ZIP_FILENAME            VARCHAR(255),
   PRIMARY_ARCHIVE                 VARCHAR(5) NOT NULL,
   PRIMARY_ARCHIVE_FILE_TYPE       VARCHAR(32),
   PRIMARY_ARCHIVE_LOCATION        VARCHAR(255),
   PRIMARY_ARCHIVE_ZIP             VARCHAR(5),
   PRIMARY_ARCHIVE_ZIP_FILENAME    VARCHAR(255),
   START_TIME                      VARCHAR(17) NOT NULL,
   END_TIME                        VARCHAR(17) NOT NULL,
   SOURCE_PARTNER_ID               VARCHAR(128) NOT NULL,
   SOURCE_PARTNER_TYPE             VARCHAR(32),
   SOURCE_PARTNER_SAVED            VARCHAR(5) NOT NULL,
   SOURCE_TRANSPORT_METHOD         VARCHAR(32) NOT NULL,
   SOURCE_SERVER_ADDRESS           VARCHAR(64),
   SOURCE_SERVER_PORT              VARCHAR(16),
   SOURCE_SERVER_ACCOUNT           VARCHAR(64),
```

SOURCE_SERVER_DIRECTORY	VARCHAR(255),
SOURCE_SERVER_TYPE	VARCHAR(32),
SOURCE_SERVER_PASSIVE	VARCHAR(5),
SOURCE_FILE_LOCATION	VARCHAR(255),
DEST_PARTNER_ID	VARCHAR(128) NOT NULL,
DEST_PARTNER_TYPE	VARCHAR(32),
DEST_PARTNER_SAVED	VARCHAR(5) NOT NULL,
DEST_TRANSPORT_METHOD	VARCHAR(32) NOT NULL,
DEST_SERVER_ADDRESS	VARCHAR(64),
DEST_SERVER_PORT	VARCHAR(16),
DEST_SERVER_ACCOUNT	VARCHAR(64),
DEST_SERVER_DIRECTORY	VARCHAR(255),
DEST_RECIPIENT_ADDRESS	VARCHAR(255),
DEST_RECIPIENT_SUBJECT	VARCHAR(64),
DEST_SERVER_TYPE	VARCHAR(32),
DEST_SERVER_PASSIVE	VARCHAR(5),
DEST_FILE_LOCATION	VARCHAR(255),
ENCRYPT_DECRYPT	VARCHAR(5) NOT NULL,
SIGN_AUTHENTICATE	VARCHAR(5) NOT NULL,
ASCII_ARMORING	VARCHAR(5) NOT NULL,
CANONICAL_TEXT	VARCHAR(5),
COMPRESSION	VARCHAR(5),
SOURCE_FTP_MODE	VARCHAR(32),
DEST_FTP_MODE	VARCHAR(32),
RUN_NOW	VARCHAR(5) NOT NULL,
API	VARCHAR(5) NOT NULL,
FILE_MONITOR	VARCHAR(5) NOT NULL,
THIRD_PARTY	VARCHAR(5) NOT NULL,
LINKED	VARCHAR(5) NOT NULL,
SCRIPT_ARGUMENT1	VARCHAR(64),
SCRIPT_ARGUMENT2	VARCHAR(64),
SCRIPT_ARGUMENT3	VARCHAR(64),
SCRIPT_ARGUMENT4	VARCHAR(64),
SCRIPT_ARGUMENT5	VARCHAR(64),
SCRIPT_ARGUMENT6	VARCHAR(64),
POLLING_FREQUENCY	VARCHAR(32),
POLLING_INTERVAL	VARCHAR(16),
NUMBER_RETRIES	VARCHAR(16),
TOTAL_ATTEMPTS	VARCHAR(16),
BUSINESS_EMAIL	VARCHAR(5) NOT NULL,
BUSINESS_EMAIL_ADDRESSES	VARCHAR(255),
IT_EMAIL	VARCHAR(5) NOT NULL,
IT_EMAIL_ADDRESSES	VARCHAR(255),

```
    PAGING_TYPE                     VARCHAR(32) NOT NULL,
    PAGING_LEVEL                    VARCHAR(32),
    PAGING_PIN                      VARCHAR(32),
    PAGING_ATTEMPTED                VARCHAR(5),
    PAGING_SUCCESSFUL               VARCHAR(5),
    PAGING_EMAIL_ADDRESS            VARCHAR(128),
    PRIMARY_PAGING_LOCATION         VARCHAR(255),
    SECONDARY_PAGING_LOCATION       VARCHAR(255),
    PAGING_FILENAME                 VARCHAR(64),
   PRE_ZIP                          VARCHAR(255),
    POST_ZIP                        VARCHAR(255),
    PRE_COMMAND                     VARCHAR(255),
    PRE_COMMAND_ATTEMPTED           VARCHAR(5),
    PRE_COMMAND_RTRN_CODE           VARCHAR(16),
    PRE_COMMAND_FAILURE_OVERRIDE    VARCHAR(5),
    POST_COMMAND                    VARCHAR(255),
    POST_COMMAND_ATTEMPTED           VARCHAR(5),
    POST_COMMAND_RTRN_CODE          VARCHAR(16),
    POST_COMMAND_FAILURE_OVERRIDE   VARCHAR(5),
    LINK_STATUS                     VARCHAR(128),
    CL_SERVER                       VARCHAR(5) NOT NULL,
    CL_SERVER_TYPE                  VARCHAR(32),
    CL_SERVER_LOCATION              VARCHAR(255),
    CL_CONFIGURATION_FILE           VARCHAR(255),
    CL_ENCRYPT_DECRYPT_TYPE         VARCHAR(32),
    CL_ENCRYPTION_KEY_ID            VARCHAR(64),
    CL_SIGN_VERIFY_KEY_ID           VARCHAR(64),
    CL_INC_DFLT_ENCRYPT_KEY         VARCHAR(5),
    CL_USE_DFLT_SIG_KEY             VARCHAR(5),
    CL_DISCARD_PATHS                VARCHAR(5) );

CREATE TABLE FILE_AUDIT (
    JOB_AUDIT_ID                    VARCHAR(128) NOT NULL,
    SOURCE_FILENAME                 VARCHAR(128) NOT NULL,
   SOURCE_SENDER_ADDRESS            VARCHAR(64),
    SOURCE_SENDER_SUBJECT           VARCHAR(64),
    SOURCE_XFERRED                  VARCHAR(5) NOT NULL,
    SOURCE_FILE_SIZE                VARCHAR(32) NOT NULL,
    SOURCE_ASCII_ARMORED            VARCHAR(5),
    SOURCE_SIGNED                   VARCHAR(5),
    SOURCE_ENCRYPTED                VARCHAR(5),
    SOURCE_DELETED                  VARCHAR(5) NOT NULL,
    SOURCE_ATTEMPT_UNARMOR          VARCHAR(5),
```

```
    SOURCE_UNARMORED                VARCHAR(5),
    SOURCE_ATTEMPT_DECRYPT          VARCHAR(5),
    SOURCE_DECRYPTED                VARCHAR(5),
    SOURCE_ATTEMPT_VERIFY           VARCHAR(5),
    SOURCE_VERIFIED                 VARCHAR(5),
    DECRYPT_KEYID                   VARCHAR(16),
    VERIFY_KEYID                    VARCHAR(16),
    DEST_WRITE_ATTEMPTED            VARCHAR(5) NOT NULL,
    DEST_WRITTEN                    VARCHAR(5),
    DEST_FILENAME                   VARCHAR(128),
    DEST_FILE_SIZE                  VARCHAR(32),
    DEST_COMPRESSED                 VARCHAR(5),
    DEST_ATTEMPT_ARMOR              VARCHAR(5),
    DEST_ASCII_ARMORED              VARCHAR(5),
    DEST_ATTEMPT_SIGN               VARCHAR(5),
    DEST_SIGNED                     VARCHAR(5),
    DEST_ATTEMPT_ENCRYPT            VARCHAR(5),
    DEST_ENCRYPTED                  VARCHAR(5),
    SIGNATURE_TIME                  VARCHAR(17),
    DEST_OVERWRITE                  VARCHAR(5),
    ENCRYPT_KEYID                   VARCHAR(16),
    AEK_ENCRYPT_KEYID               VARCHAR(255),
    SIGNATURE_KEYID                 VARCHAR(16),
    FILE_STATUS                     VARCHAR(32) NOT NULL,
    FILE_STATUS_REASON              VARCHAR(255) NOT NULL,
    SOURCE_ARCHIVE_FILENAME         VARCHAR(255),
    PRIMARY_SRC_ARCHIVE_FILENAME    VARCHAR(255),
    DEST_ARCHIVE_FILENAME           VARCHAR(255),
    PRIMARY_DEST_ARCHIVE_FILENAME   VARCHAR(255),
    CL_COMMAND                      VARCHAR(255),
    CL_RETURN_CODE                  VARCHAR(16),
    CL_RETURN_CODE_TEXT             VARCHAR(128),
    CONSTRAINT ID_FK FOREIGN KEY (JOB_AUDIT_ID) REFERENCES JOB_AUDIT (JOB_AUDIT_ID) ON DELETE CASCADE);

CREATE TABLE JOB_AUDIT_ARCHIVE (
    JOB_AUDIT_ID                    VARCHAR(128) PRIMARY KEY NOT NULL,
    VERSION                         VARCHAR(32) NOT NULL,
    BUILD_NUMBER                    VARCHAR(8) NOT NULL,
    OS_VERSION                      VARCHAR(64) NOT NULL,
    DESCRIPTION                     VARCHAR(255),
    TRANSACTION_ID                  VARCHAR(128) NOT NULL,
    TRANSACTION_TYPE                VARCHAR(32) NOT NULL,
    STATUS                          VARCHAR(32) NOT NULL,
```

STATUS_REASON	VARCHAR(255) NOT NULL,
OVERWRITE	VARCHAR(5) NOT NULL,
DELETE_SOURCE	VARCHAR(5) NOT NULL,
LOG_LOCATION	VARCHAR(255) NOT NULL,
LOG_FILENAME	VARCHAR(64) NOT NULL,
ARCHIVE	VARCHAR(5) NOT NULL,
ARCHIVE_FILE_TYPE	VARCHAR(32),
ARCHIVE_LOCATION	VARCHAR(255),
ARCHIVE_ZIP	VARCHAR(5),
ARCHIVE_ZIP_FILENAME	VARCHAR(255),
PRIMARY_ARCHIVE	VARCHAR(5) NOT NULL,
PRIMARY_ARCHIVE_FILE_TYPE	VARCHAR(32),
PRIMARY_ARCHIVE_LOCATION	VARCHAR(255),
PRIMARY_ARCHIVE_ZIP	VARCHAR(5),
PRIMARY_ARCHIVE_ZIP_FILENAME	VARCHAR(255),
START_TIME	VARCHAR(17) NOT NULL,
END_TIME	VARCHAR(17) NOT NULL,
SOURCE_PARTNER_ID	VARCHAR(128) NOT NULL,
SOURCE_PARTNER_TYPE	VARCHAR(32),
SOURCE_PARTNER_SAVED	VARCHAR(5) NOT NULL,
SOURCE_TRANSPORT_METHOD	VARCHAR(32) NOT NULL,
SOURCE_SERVER_ADDRESS	VARCHAR(64),
SOURCE_SERVER_PORT	VARCHAR(16),
SOURCE_SERVER_ACCOUNT	VARCHAR(64),
SOURCE_SERVER_DIRECTORY	VARCHAR(255),
SOURCE_SERVER_TYPE	VARCHAR(32),
SOURCE_SERVER_PASSIVE	VARCHAR(5),
SOURCE_FILE_LOCATION	VARCHAR(255),
DEST_PARTNER_ID	VARCHAR(128) NOT NULL,
DEST_PARTNER_TYPE	VARCHAR(32),
DEST_PARTNER_SAVED	VARCHAR(5) NOT NULL,
DEST_TRANSPORT_METHOD	VARCHAR(32) NOT NULL,
DEST_SERVER_ADDRESS	VARCHAR(64),
DEST_SERVER_PORT	VARCHAR(16),
DEST_SERVER_ACCOUNT	VARCHAR(64),
DEST_SERVER_DIRECTORY	VARCHAR(255),
DEST_RECIPIENT_ADDRESS	VARCHAR(255),
DEST_RECIPIENT_SUBJECT	VARCHAR(64),
DEST_SERVER_TYPE	VARCHAR(32),
DEST_SERVER_PASSIVE	VARCHAR(5),
DEST_FILE_LOCATION	VARCHAR(255),
ENCRYPT_DECRYPT	VARCHAR(5) NOT NULL,
SIGN_AUTHENTICATE	VARCHAR(5) NOT NULL,

```
ASCII_ARMORING                  VARCHAR(5) NOT NULL,
CANONICAL_TEXT                  VARCHAR(5),
COMPRESSION                     VARCHAR(5),
SOURCE_FTP_MODE                 VARCHAR(32),
DEST_FTP_MODE                   VARCHAR(32),
RUN_NOW                         VARCHAR(5) NOT NULL,
API                             VARCHAR(5) NOT NULL,
FILE_MONITOR                    VARCHAR(5) NOT NULL,
THIRD_PARTY                     VARCHAR(5) NOT NULL,
LINKED                          VARCHAR(5) NOT NULL,
SCRIPT_ARGUMENT1                VARCHAR(64),
SCRIPT_ARGUMENT2                VARCHAR(64),
SCRIPT_ARGUMENT3                VARCHAR(64),
SCRIPT_ARGUMENT4                VARCHAR(64),
SCRIPT_ARGUMENT5                VARCHAR(64),
SCRIPT_ARGUMENT6                VARCHAR(64),
POLLING_FREQUENCY               VARCHAR(32),
POLLING_INTERVAL                VARCHAR(16),
NUMBER_RETRIES                  VARCHAR(16),
TOTAL_ATTEMPTS                  VARCHAR(16),
BUSINESS_EMAIL                  VARCHAR(5) NOT NULL,
BUSINESS_EMAIL_ADDRESSES        VARCHAR(255),
IT_EMAIL                        VARCHAR(5) NOT NULL,
IT_EMAIL_ADDRESSES              VARCHAR(255),
PAGING_TYPE                     VARCHAR(32) NOT NULL,
PAGING_LEVEL                    VARCHAR(32),
PAGING_PIN                      VARCHAR(32),
PAGING_ATTEMPTED                VARCHAR(5),
PAGING_SUCCESSFUL               VARCHAR(5),
PAGING_EMAIL_ADDRESS            VARCHAR(128),
PRIMARY_PAGING_LOCATION         VARCHAR(255),
SECONDARY_PAGING_LOCATION       VARCHAR(255),
PAGING_FILENAME                 VARCHAR(64),;
PRE_ZIP                         VARCHAR(255),
POST_ZIP                        VARCHAR(255),
PRE_COMMAND                     VARCHAR(255),
PRE_COMMAND_ATTEMPTED           VARCHAR(5),
PRE_COMMAND_RTRN_CODE           VARCHAR(16),
PRE_COMMAND_FAILURE_OVERRIDE    VARCHAR(5),
POST_COMMAND                    VARCHAR(255),
POST_COMMAND_ATTEMPTED          VARCHAR(5),
POST_COMMAND_RTRN_CODE          VARCHAR(16),
POST_COMMAND_FAILURE_OVERRIDE   VARCHAR(5),
```

```
    LINK_STATUS                     VARCHAR(128),
    CL_SERVER                       VARCHAR(5) NOT NULL,
    CL_SERVER_TYPE                  VARCHAR(32),
    CL_SERVER_LOCATION              VARCHAR(255),
    CL_CONFIGURATION_FILE           VARCHAR(255),
    CL_ENCRYPT_DECRYPT_TYPE         VARCHAR(32),
    CL_ENCRYPTION_KEY_ID            VARCHAR(64),
    CL_SIGN_VERIFY_KEY_ID           VARCHAR(64),
    CL_INC_DFLT_ENCRYPT_KEY         VARCHAR(5),
    CL_USE_DFLT_SIG_KEY             VARCHAR(5),
    CL_DISCARD_PATHS                VARCHAR(5) );

CREATE TABLE FILE_AUDIT_ARCHIVE (
    JOB_AUDIT_ID                    VARCHAR(128) NOT NULL,
    SOURCE_FILENAME                 VARCHAR(128) NOT NULL,
    SOURCE_SENDER_ADDRESS           VARCHAR(64),
    SOURCE_SENDER_SUBJECT           VARCHAR(64),
    SOURCE_XFERRED                  VARCHAR(5) NOT NULL,
    SOURCE_FILE_SIZE                VARCHAR(32) NOT NULL,
    SOURCE_ASCII_ARMORED            VARCHAR(5),
    SOURCE_SIGNED                   VARCHAR(5),
    SOURCE_ENCRYPTED                VARCHAR(5),
    SOURCE_DELETED                  VARCHAR(5) NOT NULL,
    SOURCE_ATTEMPT_UNARMOR          VARCHAR(5),
    SOURCE_UNARMORED                VARCHAR(5),
    SOURCE_ATTEMPT_DECRYPT          VARCHAR(5),
    SOURCE_DECRYPTED                VARCHAR(5),
    SOURCE_ATTEMPT_VERIFY           VARCHAR(5),
    SOURCE_VERIFIED                 VARCHAR(5),
    DECRYPT_KEYID                   VARCHAR(16),
    VERIFY_KEYID                    VARCHAR(16),
    DEST_WRITE_ATTEMPTED            VARCHAR(5) NOT NULL,
    DEST_WRITTEN                    VARCHAR(5),
    DEST_FILENAME                   VARCHAR(128),
    DEST_FILE_SIZE                  VARCHAR(32),
    DEST_COMPRESSED                 VARCHAR(5),
    DEST_ATTEMPT_ARMOR              VARCHAR(5),
    DEST_ASCII_ARMORED              VARCHAR(5),
    DEST_ATTEMPT_SIGN               VARCHAR(5),
    DEST_SIGNED                     VARCHAR(5),
    DEST_ATTEMPT_ENCRYPT            VARCHAR(5),
    DEST_ENCRYPTED                  VARCHAR(5),
    SIGNATURE_TIME                  VARCHAR(17),
```

```
    DEST_OVERWRITE                  VARCHAR(5),
    ENCRYPT_KEYID                   VARCHAR(16),
    AEK_ENCRYPT_KEYID               VARCHAR(255),
    SIGNATURE_KEYID                 VARCHAR(16),
    FILE_STATUS                     VARCHAR(32) NOT NULL,
    FILE_STATUS_REASON              VARCHAR(255) NOT NULL,
    SOURCE_ARCHIVE_FILENAME         VARCHAR(255),
    PRIMARY_SRC_ARCHIVE_FILENAME    VARCHAR(255),
    DEST_ARCHIVE_FILENAME           VARCHAR(255),
    PRIMARY_DEST_ARCHIVE_FILENAME   VARCHAR(255),
    CL_COMMAND                      VARCHAR(255),
    CL_RETURN_CODE                  VARCHAR(16),
    CL_RETURN_CODE_TEXT             VARCHAR(128),
    CONSTRAINT ARCHIVE_ID_FK FOREIGN KEY (JOB_AUDIT_ID) REFERENCES JOB_AUDIT_ARCHIVE (JOB_AUDIT_ID) ON
DELETE CASCADE);

CREATE TABLE USER_ACTIVITY (
    SEQUENCE_NUMBER                 VARCHAR(9) PRIMARY KEY NOT NULL,
    TIMESTAMP                       VARCHAR(17) NOT NULL,
    USER_ID                         VARCHAR(64) NOT NULL,
    USER_IP_ADDRESS                 VARCHAR(64) NOT NULL,
    OBJECT_TYPE                     VARCHAR(32) NOT NULL,
    OBJECT_ID                       VARCHAR(128),
    ACTION                          VARCHAR(32) NOT NULL,
    COMMENT                         VARCHAR(255));


CREATE TABLE USER_ACTIVITY_ARCHIVE (
    SEQUENCE_NUMBER                 VARCHAR(9) PRIMARY KEY NOT NULL,
    TIMESTAMP                       VARCHAR(17) NOT NULL,
    USER_ID                         VARCHAR(64) NOT NULL,
    USER_IP_ADDRESS                 VARCHAR(64) NOT NULL,
    OBJECT_TYPE                     VARCHAR(32) NOT NULL,
    OBJECT_ID                       VARCHAR(128),
    ACTION                          VARCHAR(32) NOT NULL,
    COMMENT                         VARCHAR(255));
```

6 Support

Installation and configuration support is provided under warranty for 45 days from initial purchase, as well as under annual maintenance agreements. Email and phone support is available from 9 a.m. ET to 5 p.m. ET weekdays. If you require assistance, contact Coviant Software Support as follows:

Voice: 781.210.3310 x2
Fax: 781.210.3313
Web: www.coviantsoftware.com
E-mail: support@coviantsoftware.com

Web: www.coviantsoftware.com
E-mail: support@coviantsoftware.com

Diplomat Managed File Transfer products interoperate with other software applications, such as FTP, STMP, SMS, and OpenPGP software. File transfer and encryption failures can occur during a job created by Diplomat Managed File Transfer for many reasons, including:

- Inaccurate transaction or setting data
- Connection problems with FTP, email, or local systems
- Wrong encryption or signature keys on incoming files
- Missing files or keys
- Mismatch between file format and FTP transfer settings
- Compatibility issues with older OpenPGP versions
- Incorrect or incompatible FTP server settings

Typically, these problems are NOT due to a malfunction of your Diplomat MFT product. Data to diagnose these problems and others are provided in the log files, debug email messages, and audit trail data generated when the job or jobs were run. These types of conditions are usually the user's responsibility. Please review the diagnostic information provided before contacting Coviant Software for support.

If you require support assistance that appears to be due to a malfunction of your Diplomat MFT software, please have the following items available before contacting a support representative.

- Diplomat MFT Edition name, version installed, and serial number located in Help > About Diplomat
- Current log file containing entries for the failed job(s)
- IT Support emails containing debug information for the failed job(s), if available
- Audit detail report for the failed job(s), if available

You may be asked to send some of the above information to the Coviant Software Support representative in order to resolve your problem in a timely manner.

7 Appendix A: Configuration Requirements

Your environment may include not only the computer systems to run Diplomat, but other systems that provide functionality that co-exists with or is used by Diplomat, such as ftp servers, mail servers, paging servers, and OpenPGP software for key import/export. Specific software versions tested for Diplomat Managed File Transfer are shown below.

Supported Software	
Diplomat MFT Service	Windows 7[1], 8 and10 (64-bit) Windows Server 2008 R2[1], 2012 R2[1] and 2016[1] (64-bit) Red Hat Enterprise Linux v6.3 Intel x86 (64-bit)
Diplomat MFT Client[2] Diplomat Cloud Connector Diplomat MFT Job Monitor	Windows 7[1], 8 and 10 (64-bit) Windows Server 2008 R2[1], 2012 R2[1] and 2016[1] (64-bit)
Diplomat MFT Scripting Agent	Windows 7[1], 8 and 10 (64-bit) Windows Server 2008 R2[1], 2012 R2[1] and 2016[1] (64-bit) Red Hat Enterprise Linux v6.3 Intel x86 (64-bit) Other Unix systems running Java Runtime Environment (JRE) 1.8 or higher
Diplomat MFT Web Launch	Any system supporting Java Runtime Environment (JRE) 1.8
FTP Server	Any FTP server compliant with FTP (RFC 959), FTPS (RFC 2228 with Secure FTP Using TLS), or SFTP (SSH-2; Secure Shell Charter); FTP file transfers tested with Windows, UNIX, AS400/Library, and AS400/IFS systems; SFTP and FTPS tested with Windows systems
Mail Server	Any SMTP (RFC 2821), POP3 (RFC 1939) or IMAP (RFC 3501) compliant server
HTTP/S Server	Any HTTP or HTTPS (RFC 2616) compliant server
Paging Server (Optional)	Generic file and email-based communications to paging systems; no specific paging server products are explicitly supported.
SQL Database (Optional)	MySQL Server 5.1 or later Microsoft SQL Server 2005 v9.0 and 2008 v10.0 Most ANSI SQL-92 compliant databases with JDBC support

[1] When running Windows 7, Windows Server 2008 or follow-on products, the Diplomat MFT Service cannot run as a local system account. A logon account with administrator privileges must be specified. For detailed instructions on how to update the Diplomat MFT Service, see http://coviantsoftware.com/setting-windows-login.php.

[2] Unix operating systems are not currently supported. Contact Coviant Software Support for further information.

Supported Software (Cont'd)	
OpenPGP Software[3] (Optional)	OpenPGP compliant (RFC 2440, RFC 4880) products, including: ▪ Authora Edge v3.6 ▪ McAfee E-business Server v8.0 – v8.5.2 ▪ PGP Command Line v9.0 – v10.0 ▪ Veridis FileCrypt v3.6
SSH Software (Optional)	OpenSSH SSH Tectia Server from SSH Communications Security

The hardware configurations shown are based on approximately 100 simultaneous file transfer jobs with associated keys and partners. Your production environment may require less or more memory/disk space depending on the numbers of transactions, keys, and partners you use.

Minimum Hardware Configurations		
Diplomat MFT Service	Memory	1 GB
	Disk Space	250 MB
Diplomat MFT Client	Memory	512 MB
	Disk Space	100 MB
Diplomat MFT Job Monitor	Memory Usage	512 MB
	Disk Space	100 MB

[3] Diplomat has been tested with the listed software products, but should be compatible with any OpenPGP product compliant with RFC 2440 and RFC 4880.

8 Appendix B: MySQL Windows Set-up Instructions

1. Download the MySQL database server.

Diplomat MFT supports Version 5.1.x of the MySQL database server running on Windows (NT, 2000, XP). Go to http://dev.mysql.com/downloads/mysql/5.1.html. Download the Windows (x86) version (not the Windows Essentials or Windows without installer versions).

2. Install the MySQL database server.

a. Unzip the downloaded file into a temporary location.

b. Execute Setup.exe.

c. Choose installation type.

d. 'Typical' is recommended, which installs MySQL in 'C:\Program Files\MySQL\MySQL Server 5.1\'. Choose 'Custom', if you need to change the installation directory. On the 'Wizard Completed' screen, leave the 'Configure the MySQL Server now' box checked and click 'Finish'.

3. Configure the MySQL database server.

a. Click 'Next' on the Welcome screen.

b. Choose 'Detailed Configuration'. Click 'Next'.

c. Choose 'Server Machine'. Click 'Next'.

d. Choose 'Transactional Database Only'. Click 'Next'.

e. Choose a location for the database files. The default location is the directory where the MySQL database server is installed. Click 'Next'.

f. Choose 'Decision Support (DSS)/OLAP'. Click 'Next'.

g. Choose 'Enable TCP/IP Networking'. Leave the port # set to 3306, if possible. If port #3306 is already being used, you must supply an unused port #. Click 'Next'.

h. Choose 'Standard Character Set'. Click 'Next'.

i. Choose 'Install as Windows Service'. Leave service name set to 'MySQL', if possible. If you already have a service named 'MySQL', you must choose a unique name. Leave the 'Launch the MySQL Server automatically' box checked. Click 'Next'.

j. Choose 'Modify Security Settings' and supply the root password (twice). Check the 'Root may only …' box if you want to limit root connections to be only from the local machine. Diplomat MFT works whether this box is checked or not. Click 'Next'.

NOTE: Write down the root password you supplied, as you will need it later.

k. Click 'Execute'.

l. Click 'Finish'.

4. Download the MySQL Administrator.

Go to http://dev.mysql.com/downloads/administrator/1.0.html. Download the Windows (x86) version (not the Windows without installer version).

5. Install the MySQL Administrator.

a. Double-click the *.msi file you downloaded in the previous step.

b. Click 'Next' on the Welcome screen.

c. Accept the license agreement. Click 'Next'.

d. Choose an installation location. Click 'Next'.

e. Choose 'Complete'. Click 'Next'.

f. Click 'Install'.

g. Click 'Finish' when installation has completed.

6. Create the Diplomat MFT database

a. Run the Administrator program using the 'Start->MySQL->MySQL Administrator' menu selection.

b. To Connect to the Administrator:

i. Leave the Stored Connection field blank.

ii. Enter 'localhost' in the Server Host field

iii. If you did not change the port # above, leave the Port field set to 3306. Otherwise, enter the port number you entered above.

iv. Enter 'root' for the Username.

v. Enter the root password you specified above in the Password field.

vi. Click 'OK'.

c. To create the Diplomat MFT DB:

i. Click on 'Catalogs'.

ii. Click on the 'mysql' schema. (Schema is the term MySQL uses for databases. These terms may be used interchangeably in this document).

iii. Right-click in the panel where the schema are displayed. Choose 'Create New Schema'.

iv. In the 'Create New Schema' dialog, supply the name you wish to use for the Diplomat MFT database.

NOTE: Write down the database name you entered, as you will need to supply it when you enter the Diplomat MFT Audit Trail settings from the Diplomat MFT Client and if you choose to manually set up the Diplomat MFT tables.

7. Create the Diplomat MFT user.

a. Click 'User Administration'.

b. Right-click in the panel where the current users are displayed. Choose 'Add New User'.

c. On the 'User Information' tab, supply the username and password that Diplomat MFT will use when connecting to the database.

NOTE: Write down the username and password you entered, as you will need to supply it when you enter the Diplomat MFT Audit Trail settings from the Diplomat MFT Client.

d. On the 'Schema Privileges' tab, click the database you created above.

e. Move SELECT, INSERT, UPDATE and DELETE from the 'Available Privileges' to 'Assigned Privileges'. Click 'Apply Changes'.

f. If you want Diplomat MFT to create your database tables automatically:

   i. Move CREATE, REFERENCES and INDEX from the 'Available Privileges' to 'Assigned Privileges'. Click 'Apply Changes'.

   ii. Skip the final step below.

8. Create Diplomat MFT tables.

a. From the 'Tools' menu, select 'MySQL Command Line Client'.

b. At the 'mysql' prompt, type: connect *database* localhost

   [Replace *database* with the database name you supplied above.]

c. At the next 'mysql' prompt, type: source *DiplomatAuditDB.ddl*

   [Replace *DiplomatAuditDB.ddl* with the full path name of the DiplomatAuditDB.ddl file.]

9 Appendix C: Glossary

Additional Archive Directory – Directory on the network where backup files for a specific file transfer job are written.

Additional Encryption Key (AEK) – Public key used when the user wants to encrypt files to more than one key.

Active Window – Right-hand side of the main screen for Diplomat MFT Client that displays the active key, partner, or transaction that is being viewed or edited. Some data is displayed in panels that can be maximized for editing and then minimized to save screen space.

Business Users – Persons responsible for specific file transfers with trading partners or internal groups.

Debug – A setting that when activated inserts system messages into an email notification message. It is used primarily to troubleshoot problems in jobs.

Destination Directory – The directory on an FTP server or local network where a transaction file is to be written.

Diplomat MFT Administrator – Person administering the Diplomat MFT Service and Diplomat MFT Configuration Database.

Diplomat MFT Audit Database – Database containing detailed records of every job executed and user activity. The audit database is a set of XML files where each job has a single file or a SQL database with three tables to capture Job, File, and User Activity and three tables in which to archive Job, File, and User Activity records.

Diplomat MFT Client – Desktop application that enables creation and modification of key, partner, transaction information, and configuration settings, as well as license management, report generation, and job scheduling.

Diplomat MFT Configuration Database – Database containing all system-setting and transaction setting data. The configuration database is a single XML file.

Diplomat MFT Scripting Agent – Java application that submits for execution a specified transaction that has been created and saved in a Diplomat MFT transaction database that may require an optional password.

Diplomat MFT Service – Run-time engine that executes transactions stored in the Diplomat MFT transaction database and interfaces with FTP servers, mail servers, and other systems, as needed. The Diplomat MFT Service is a Windows service. After installation, the Windows operating system starts the Diplomat MFT Service, which then runs in the background creating jobs for each transaction. Plus, it creates a log file with system messages, an audit database, and archives transaction files, if desired.

Diplomat MFT Service Login – Windows login identity for the Diplomat MFT Service on the Diplomat MFT site. Defaults to Local Network.

Diplomat MFT Transaction Database – Contains all data needed to create and schedule jobs, including keys, partner profiles, and transaction data. The transaction database contains the Diplomat MFT keystore. The transaction database is comprised of a set of XML files that can be backed up and restored as a group.

Diplomat MFT Users – Persons setting up new keys, partners, and transactions that are allowed to automatically login to the Diplomat MFT Client, but do not have access to certain administrative functions.

Firewall – A software program that protects computers on a network from unauthorized Internet access.

FTP Server – A software program that allows the receipt and pick-up of files, which typically resides outside a corporate firewall.

Inbound Transaction – The process of receiving a file from another organization with optional decryption and verification.

IP Address – The numerical identification of a computer connected to a network. The IP Address appears with periods separating groups of numbers. (i.e. 192.168.0.1).

Job – A job is a particular execution of a transaction. For example, if a transaction is scheduled to run once a day, a new job will be created and executed once a day.

Job Monitor – A feature of Diplomat MFT that allows the real-time monitoring of job scheduling and execution.

License File –Diplomat MFT uses a license file named *diplomat.lic* to determine the number of keys you can have in your Diplomat MFT keystore and the expiration date of your license.

Log File – File containing chronological system messages generated as a result of Diplomat MFT operation.

Mail Server – A computer that acts as temporary recipient and storage for email messages sent to an individual.

Main Screen – Contains top menu bar, left-hand navigation tree, and active window for Diplomat MFT Client.

Menu Bar – Bar at the top of the main screen for Diplomat MFT Client that allows access to a variety of functions via sub-menus and pop-up dialog boxes.

Menu Item –Selection on the top menu bar of Diplomat MFT Client. When a menu item is selected either a sub-menu or a pop-up dialog box is displayed.

Navigation Tree – Left-hand side of the main screen for Diplomat MFT Client that displays folders, sub-folders, and objects with status indicators in a tree format for easy navigation

OpenPGP – Open PGP is one type of public key encryption technology. It is based on an asymmetric scheme that uses a pair of keys: a *public key*, which encrypts data, and a corresponding *private*, or *secret key* for decryption. The OpenPGP protocol, created by the Internet Engineering Task Force (IETF), defines standard formats for encrypted messages, signatures, private keys, and certificates for exchanging public keys.

OpenPGP Command Line Tool – OpenPGP products with a command line interface, such as PGP Command Line Server and McAfee e-Business Server.

Open PGP Key Pair – OpenPGP keys are always created as key pairs with a public key and a private key. The owner of a key pair keeps their key pair and gives their trading partner their public key.

OpenPGP Public Key – The OpenPGP key that is made available to an organization's trading partners to be used to encrypt data that is sent from the trading partner to the organization.

Outbound Transaction – The process of moving a file from within an organization to a receiving organization with optional encryption and signing of the file.

Paging Application – Software that converts email or files to a radio signal that is received by beepers.

Panel – Section of active window, usually surrounded by a blue border. Some larger panels can be maximized for editing and then minimized to save screen space.

Partner Profile – A set of information defining default parameters to be used when setting up a transaction with the trading partner.

Passphrase – Used by OpenPGP algorithms to encrypt your private key.

PGP – An acronym for Pretty Good Privacy, an encryption application developed by Phil Zimmerman that utilizes asymmetrical or public/key pairs to encrypt and decrypt files. Trademarked by PGP Corporation.

Pop-up Dialog Box – Window used to collect data for features accessed from the top menu bar in the Diplomat MFT Client.

Primary Archive Directory – Directory on the network where backup copies of files from all jobs are written.

Public Partners – Trading partners that provide you only their public keys for encryption and verification.

Signature Key– The OpenPGP key used to sign a file on encryption and authenticate/verify it on decryption.

Source Directory – The directory on an FTP server or local network where a transaction file is to be picked up.

SQL Audit Database – Contains two tables to capture Job and File records for each transaction and two tables in which to archive Job and File records, if desired.

Status Indicator – Colored icons that indicate scheduling status of transactions and suspend status of keys, partners, and transaction folders.

Trusted Partners – Trading partners that are considered part of your organization and can use key pairs for decryption or signing.

User Activity – Any action taken when using the Diplomat MFT Client, such as when a user creates, updates, or deletes records in the Diplomat MFT transaction database and associated configuration files.