# Database Logging Module Documentation

## Purpose:

There is a need for application-level logging for database applications for informational and debugging purposes that is independent of any ongoing SQL transactions. The Database Logging Module (DLM) was developed to provide a method to log entries in an Oracle database for any modules that utilize an Oracle database. A view object is available for viewing log entries that can be filtered and searched for certain types of log entries or log entries with certain types of content based on SQL query logic. The DLM has been provided to log the different types of information from a given module independent of SQL transactions performed by that module.

## Resources:

* Database Logging Module Version Control Information:
  + URL: [git@gitlab.pifsc.gov:centralized-data-tools/centralized-tools.git](mailto:git@gitlab.pifsc.gov:centralized-data-tools/centralized-tools.git) in the DB\_log folder
  + Database: 0.1 (Git tag: db\_log\_db\_v0.1)
* [Database Diagram](data_model/DB_Log_diagram.png)
* [Database Naming Conventions](Database%20Logging%20Module%20DB%20Naming%20Conventions.docx)

## Database Setup:

* The DB Version Control Module ([git@gitlab.pifsc.gov:centralized-data-tools/centralized-tools.git](mailto:git@gitlab.pifsc.gov:centralized-data-tools/centralized-tools.git) in the DB\_version\_control folder) must be installed on the given schema before the DB Logging Module can be installed.
  + Version Control Information:
    - URL (Git): [git@gitlab.pifsc.gov:centralized-data-tools/centralized-tools.git](mailto:git@gitlab.pifsc.gov:centralized-data-tools/centralized-tools.git) in the DB\_version\_control folder
    - Application: 0.12 (Git tag: db\_vers\_ctrl\_v0.12)
    - Database: 0.2 (Git tag: db\_vers\_ctrl\_db\_v0.2)
* [Installing or Upgrading the Database Logging Module](file:///C:\Users\Jesse.Abdul\Documents\Version%20Control\Git\centralized-tools\DB_log\docs\Installing%20or%20Upgrading%20the%20Database%20Logging%20Module.docx)

## Database Design:

* Naming Convention: The database objects used in the module have the prefix "DB\_LOG\_"
* Tables:
  + DB\_LOG\_ENTRY\_TYPES: This table stores the different types of database log entries. Entry types include informational, errors, and success
  + DB\_LOG\_ENTRIES: This table stores log entries for a given module to enable debugging, logging errors, etc. This table is used in the Database Logging Module (DLM)
* Views:
  + DB\_LOG\_ENTRIES\_V: This query returns all log entries stored in the DB\_LOG\_ENTRIES table that includes the associated DB\_LOG\_ENTRY\_TYPES information for each log entry
* Packages:
  + DB\_LOG\_PKG: This package provides a single procedure ADD\_LOG\_ENTRY() that will insert a database log entry into the DB\_LOG\_ENTRIES table based on the parameters passed to the procedure. This procedure commits the DB\_LOG\_ENTRIES record in an autonomous transaction that is independent of any ongoing SQL transactions in the given module's execution so even if the transaction is rolled back the database log entry will be inserted barring any database errors encountered when the log entry is inserted.

## Implementation:

* When connected to a database with the Database Logging Module installed execute the commands listed in [execute DB\_LOG\_PKG.sql](../SQL/scripts/execute%20DB_LOG_PKG.sql)