ICE for Eclipse

Release 9.0.3

Disclaimer

This document is for informational purposes only and is subject to change without notice. This document and its contents, including the viewpoints, dates and functional content expressed herein are believed to be accurate as of its date of publication. However, Epicor Software Corporation makes no guarantee, representations or warranties with regard to the enclosed information and specifically disclaims any applicable implied warranties, such as fitness for a particular purpose, merchantability, satisfactory quality or reasonable skill and care. As each user of Epicor software is likely to be unique in their requirements in the use of such software and their business processes, users of this document are always advised to discuss the content of this document with their Epicor account manager. All information contained herein is subject to change without notice and changes to this document since printing and other important information about the software product are made or published in release notes, and you are urged to obtain the current release notes for the software product. We welcome user comments and reserve the right to revise this publication and/or make improvements or changes to the products or programs described in this publication at any time, without notice. The usage of any Epicor software shall be pursuant to an Epicor end user license agreement and the performance of any consulting services by Epicor personnel shall be pursuant to Epicor's standard services terms and conditions. Usage of the solution(s) described in this document with other Epicor software or third party products may require the purchase of licenses for such other products. Where any software is expressed to be compliant with local laws or requirements in this document, such compliance is not a warranty and is based solely on Epicor's current understanding of such laws and requirements. All laws and requirements are subject to varying interpretations as well as to change and accordingly Epicor cannot guarantee that the software will be compliant and up to date with such changes. All statements of platform and product compatibility in this document shall be considered individually in relation to the products referred to in the relevant statement, i.e., where any Epicor software is stated to be compatible with one product and also stated to be compatible with another product, it should not be interpreted that such Epicor software is compatible with both of the products running at the same time on the same platform or environment. Additionally platform or product compatibility may require the application of Epicor or third-party updates, patches and/or service packs and Epicor has no responsibility for compatibility issues which may be caused by updates, patches and/or service packs released by third parties after the date of publication of this document. Epicor® is a registered trademark and/or trademark of Epicor Software Corporation in the United States, certain other countries and/or the EU. All other trademarks mentioned are the property of their respective owners.

Publication Date: May 2, 2017

Table of Contents

System Maintenance Overview

Perform system maintenance functions to keep your Universe database, system applications, and system machine running smoothly.

Monitor the system functions through the system's activity log, which records activities such as file backups, and through system reporting. Use system maintenance reports to verify file backups, identify disk free space, record system activity and errors, read process status, and generate performance statistics.

Use the merge and purge utilities to free up disk space on your system. You can merge new product records with old ones, and then purge the old product records. You can also purge expired entries from logs, expired price sheets and system records, and report writer save-lists.

Use the Eclipse Dictionary and File Definition Maintenance to define and access all system data. Data files hold all of your system records. You can define your own files to store additional information. Dictionaries in the system apply instructions and rules to files in order to translate data into readable information. Set up maintenance logs to record file activity.

Only the system administrator and Eclipse personnel should perform system maintenance. The system administrator has complete access to the system and can lock other users out of the system in order to perform maintenance.

Data Replication & ICE Overview

The Epicor ICE (Internet Component Environment) is a software framework for integrating with ERP software. Epicor ICE uses connections to external datasources to create Business Activity Queries (BAQs). You can use BAQs to build your custom dashboards and deliver them to your users using various tools and integration points, such as Epicor Mobile Access™, Epicor Web Access™ or Microsoft® SharePoint®.

The data you retrieve from an external datasource can be displayed using the various views through the Dashboard program. Updatable dashboards are extensions of the standard dashboard capabilities. When you configure a dashboard to contain an updatable BAQ, the dashboard becomes similar to a data entry program, allowing you to review, enter, and update data within a dashboard you create. Data updatability is supplied by integration of updatable BAQs with Epicor Service Connect workflows.

You can also use BAQs to utilize Enterprise Search, which is an optional search application you can use to retrieve indexed content from within your application and then quickly launch specific programs to display the data returned from the search.

Because of the variety of functionality and possibility for use available within the Epicor ICE application, this application help system is intended for a wide audience. Managers and administrators can use it to explore and better understand the overall architecture and the functionality, and new users can use the help for quick answers to both program level and field level questions.

> **Note:** Eclipse SQL Replication is available as of Release 8.7.5 and the associated tasks are currently only available in the Eclipse Eterm application.

For Epicor Eclipse, you must first perform the Eclipse data replication to the Sequel Query Language (SQL) Data Warehouse database. Then, all the ICE tools become available to you. For more information about the Eclipse SQL database replication, see *Enabling Data Replication*.

> **Important:** Users are expected to have knowledge of Eclipse user-defined files and how the system uses Eclipse tables. In addition, a basic knowledge of SQL database terminology and use is assumed.

Data Warehouse Installation

The Data Warehouse Installation is used to set up the data replication used for both data warehouse tasks, SQL replication, and SSRS management within Eclipse.

Enabling Data Replication

The SQL Replication provides near real-time replication of the core Eclipse Universe files to SQL Server. This technology is built into the core Eclipse architecture and utilizes the Epicor Data Warehouse platform for fast and efficient replication of data.

Important: Before using the SQL Replication you must install the Epicor Data Warehouse platform by following the installation guide for that product

After installing the Data Warehouse platform, register the Eclipse system with the Data Warehouse and enable replication. The Data Warehouse can support multiple Eclipse instance registrations against it. Each Eclipse instance that is registered with the Data Warehouse receives a unique SQL database created and assigned to the replication of that Eclipse instance.

The Data Warehouse platform that was installed to support SQL Replication must be configured in Eclipse. Enter the IP Address and port where you installed the Data Warehouse. The Data Warehouse utilizes two ports: one for the Application Programming Interface (API) and one for the Data Replication.

To enable replication and configure the SQL server:

1. From the **System > User Custom Menu** menu, select **SQL Server Configuration** to register the Eclipse instance with the Epicor Data Warehouse.
2. Complete or modify the fields, as needed:

Field	Description
IP Address	Enter the IP Address and port where you installed the Data Warehouse.
API Port	Modify the port ID for application programming interfaces (API), if needed. The system defaults to the an assigned port, such as 8180.
Data Port	Modify the port ID used for and data replication, if needed. The system defaults to the an assigned port, such as 3000.
Domain	Enter the unique domain name that was assigned to your company from Epicor. This domain name is the same domain used when installing the Data Warehouse platform.
Database Name	Name the SQL database that you want created during the registration on the SQL Server. **Note:** We recommend naming the server so as to identify the instance clearly, such as PRODUCTION or TEST.
Login **Password**	Provide the user name and password of an Eclipse system account that you would like to use for communication to the Data Warehouse. **Note:** This user setup in Eclipse with full system privileges.
Shared Key	Enter the Shared Key that was created for the Eclipse system. You can access this key by going to the following link: http://[youreclipseserver]:2080/ds/home

3. Save your changes and exit the screen.
4. Continue to Enabling Eclipse Files for Syncing.

Enabling Eclipse Files for Syncing

Depending on your needs, you can enable specific files in your database to be continually synced for replication or you can leave some files out of the syncing process. For example, if you do not need to report on Activity Based Costing (ABC) codes, you can remove the ABC files from replication process.

Use the SQL Server Sync Maintenance screen to turn on or off Eclipse files that sync over to the Data Warehouse. Turning a file off disables it and no changes to that file are synchronized over to the Data Warehouse. Turning a file back on triggers a full replication of that file over to the Data Warehouse and turns file change tracking back on for near-real time synchronization.

If you use user-defined files, you cannot have a user-defined table and a standard Eclipse table sync off the same file. The sync process is mutually exclusive. If both tables are sync-enabled, such as product_class and product_class_ud, the system syncs the standard table only. In this example, only product_class is synced.

Important: Before beginning this procedure you should enable data replication on your system.

To maintain your Eclipse files:

1. From the **System > User Custom Menu** menu, select **SQL Server Configuration** to register the Eclipse instance with the Epicor Data Warehouse.

   The Eclipse files in your database display in the Eclipse File column. The associated table in which the files live displays in the Table column.

2. Use the **S** column (Sync) to enter an asterisk (*****) for each file you want regularly synced to the data warehouse.

3. Exit the screen.

4. From the **System > User Custom Menu** menu, select **SQL Server Database Maintenance** to view which Eclipse files are configured for replication along with the SWL tables into which the data is being stored.

5. Select a table to display in view-only mode which columns are replicating over to the table from the Eclipse file.

   The system displays the column name used in the SQL table and its attributes from the Eclipse file, including the data for the column and column data type.

6. Exit the screen.

About User-Defined Eclipse Columns and Tables

If you have user-defined Eclipse files, you can create new columns of data in the SQL table from an existing File Attribute in an Eclipse file. You can define the column name in SQL, the data type, and whether it is a primary key or foreign key on that SQL table.

To add a user-defined Eclipse column or table:

1. From the **System > User Custom Menu** menu, select **SQL Server Column Maintenance**.
2. Display the column or table you want to update.
3. Use the **Properties** hot key to display the properties screen.

4. Use the following fields to indicate where to get the information for the table:

Field	Description
Getter Subroutine	
Setter Subroutine	
Eclipse File Record	(For table properties only)

5. Save your changes and exit the screen.

About Indexing for ICE Eclipse Tables

When your Eclipse data has been replicated in the SQL server format, the system does *not* create indexes for you aside from those necessary for the real-time replication when files are enabled.

You have the ability in the Eclipse SQL screen to add indexes that will be meaningful to you. Each company has different needs. Therefore, there are aside from the necessary standard indexes Eclipse does pre-create additional indexes.

Your data is pushed into many different tables. Each table has a primary key to leverage for pulling your information, as needed. Also, an **eclipse_id** index is created by Eclipse to support the "deletes and inserts" that real-time replication performs as Eclipse records are changed. You are required to manage the Indexes that will enable your BAQs and Dashboards to perform quickly.

We highly recommend creating indexes *within* Eclipse. By doing this, you ensure that if a file version changed or the customer elected to re-sync a file regardless of the reason, those indexes are not deleted and you do not have to manually recreate the indexes in the SQL environment. This setup makes ongoing support and maintenance easier.

Important: We caution you not to create too many indexes. Adding indexes increases the SQL data size and "over-indexing" may slow the replication process and increase the size of the SQL database as a whole.

Troubleshooting

Eclipse query phrases are not pure SQL. Therefore,when you are looking at the error tool in the BAQ, you are trying to locate the missing indexes so you know what to create.

Identifying Required Indexes

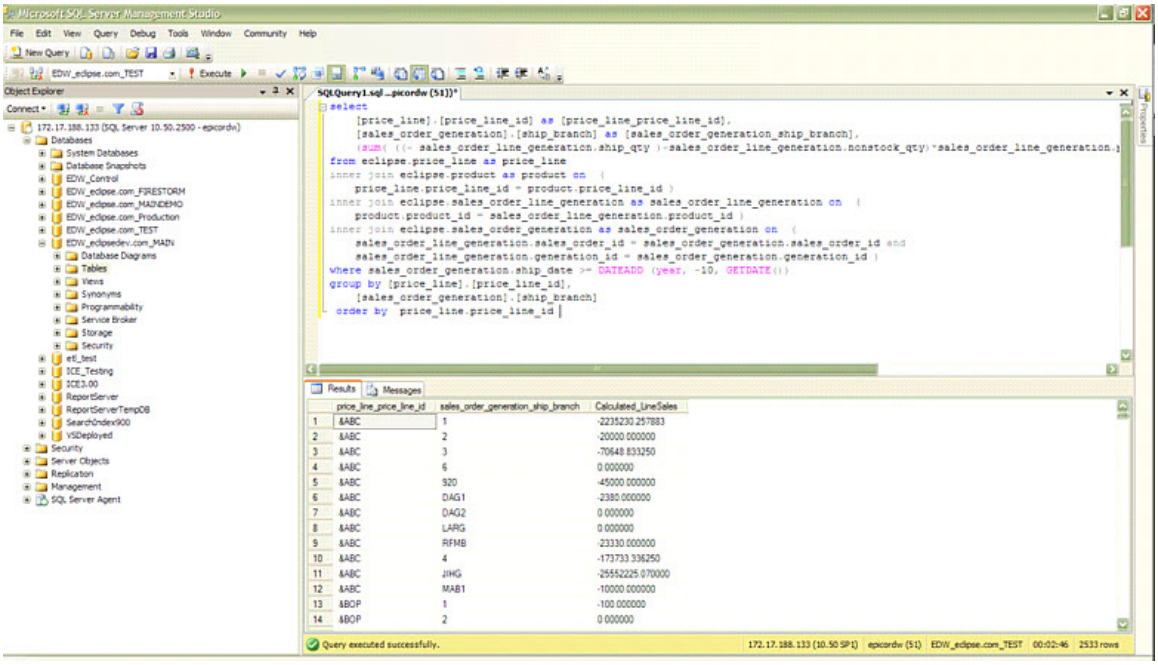
After you have constructed your business activity query (BAQ), and you either receive a timed out response when you use the Analyze tool, or the response is not as quick as you want, it is likely that some indexes could vastly improve performance.

> **Note:** The following instructions assume a basic working knowledge of SQL Server Management Studio which should already installed and connected to your replicated SQL Server instance.

1. Copy the **Query Phrase** text from the front page of the BAQ Editor
2. Paste the **Query Phrase** into SQL Server Management Studio in a **New Query** pane.
3. Click **Execute** to run your BAQ query directly in SQL
   - It is likely that the program will return SQL syntax errors for things such as Calculated fields or BAQ constants because the BAQ Query Phrase is not pure SQL. The SQL Server Management Studio error points to the problematic text which allows you to “clean up” the syntax by removing the offending variable (keep in mind, precision of results and business usefulness is not the objective at this point…the goal is to successfully execute the crux of the SQL command.
   - Once past any syntax errors, the query will run until it is complete in SQL Server Management Studio and will not timeout after 30 seconds as it does within ICE.

4. In SQL Server Management Studio, enable the **Include Actual Execution Plan** button.
   You can also enable this button from the **Query** menu.
5. Click **Execute**.
6. Click the **Execution Plan** tab in the results pane at the bottom.
   SQL identifies missing indexes with the actual execution plan and highlights them in green.
7. Place your cursor over the green text and note the specific recommended index.
8. Proceed to the next section to create that Index.
   For example:
   CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>]
   ON [eclipse].[sales_order_generation] ([ship_date])
   INCLUDE ([sales_order_id],[ship_branch],[generation_id])

SQL Server Management Studio (SQL Server 2008R2)



Adding Indexes for ICE Through Eclipse

We highly recommend creating indexes *within* Eclipse. By doing this, you ensure that if a file version changed or the customer elected to re-sync a file regardless of the reason, those indexes are not deleted and you do not have to manually recreate the indexes in the SQL environment. This setup makes ongoing support and maintenance easier.

To add an index in Eclipse:

1. From the **System > Custom > Add On Products > SQL** menu, select SQL Server Database Maintenance.
2. Place the cursor on the table you want to add an index to as instructed from SQL Server Management Studio
3. Use the **Expand** hot key to display SQL Server Table Maintenance screen.
4. Use the **Indexes** hot key and select **New** from the list.
5. Populate the following columns for the index you want to add.

Column	Description
Key Columns	Indicate which columns are Key Columns noted just to the right of the table name in SQL Server Management Studio.
Sort Order	Indicate which order you want to sort by: ascending or descending.
SQL Type	Defaults based on the SQL type of data for the table columns selected

6. Use the **Included** columns to add other table columns in the Index as directed by SQL Server Management Studio.
7. Use the **Create Index** hot key and type **Yes** at prompt to confirm that you want to create the index

   The Create Index command is added to the SQL Replication queue and is sent over to SQL when the next sync occurs. This could be a while if heavy replication is occurring at the same time, so the system holds the information until the next sync. The table does not need to be re-synced to activate the new index.

   You can confirm the index was created by viewing it from within SQL Server Management Studio by navigating in the left-hand, file tree, to the database, the table, and expanding the Indexes folder for that table. Re-executing the cleansed SQL command of your BAQ in SSMS with the indexes in place provides a revised completion time expectation.

Creating User-Defined Tables for SQL Replication

Eclipse provides the necessary tools for creating Eclipse user-defined schema mappings in your SQL Database Maintenance. With the data replication you have standard Eclipse schema mappings. However, these may provide to much or too little data for the types of queries you want to run. You cannot change the standard mappings for Eclipse schemas, but you can copy them and create your own.

> **Note:** You must have the SQL.SERVER Level 2 authorization key in order to create user-defined tables. If you are assign Level 1, the windows display as view-only.

Remember, you cannot have a user-defined tables (`eclipse_ud` schema) and a standard Eclipse tables (`eclipse` schema) sync-enabled the same file because the eclipse schema always takes priority over the `eclipse_ud` schema tables. The sync process for these tables is mutually exclusive. If both tables are sync-enabled, such as `eclipse_ud.product` and `eclipse.product`, the system syncs the standard eclipse table only. In this example, only `eclipse.product` is synced. You would have to disable the `eclipse_ud.product` sync.

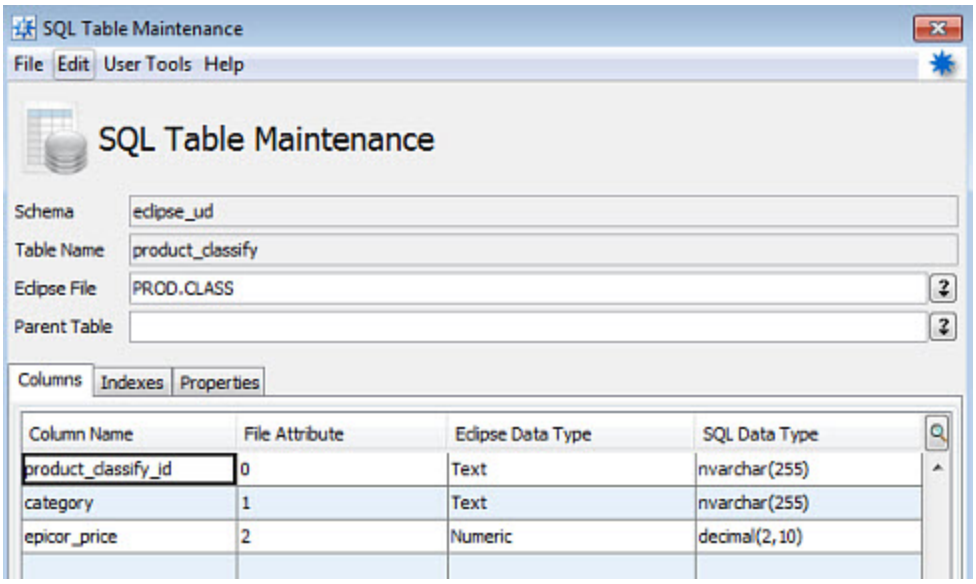
You may need to experiment with your user-defined file settings and attributes to make sure you are syncing all the data you require for your queries.

To create a user-defined table:

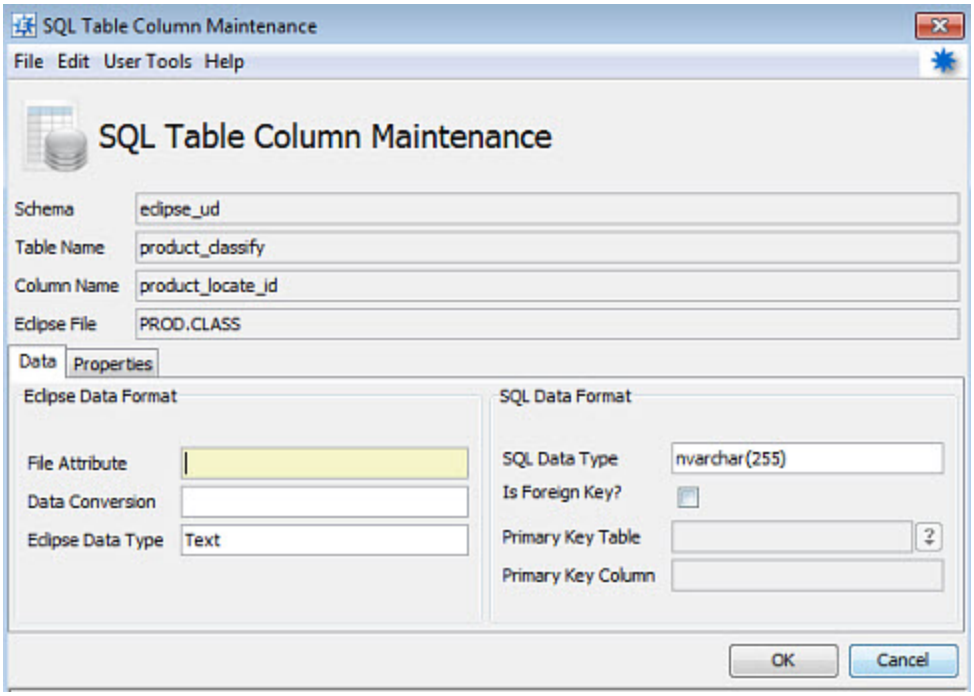
1. From the **System > Custom > Add On Products** menu, select **SQL Database Maintenance** to display the SQL Database Maintenance window.
2. In the **Schema** field, select **eclipse_ud**.

   The system updates the Table Names with the current eclipse_ud schemas in the system.
3. Click **New Table**.
4. At the **New Table Name** prompt, enter the name you want to use for the new table and click **OK**.
5. At the tracker prompt, enter the tracker number and click **OK**. The SQL Table Maintenance displays.

   The system prompts you for a tracker number as your mappings are versioned and become part of the source files for your site. We recommend creating a special tracker number to record your changes (or use AAA000) and enter a description that becomes part of the programming change log for your mapping intent.



6. In the **Eclipse File** field, select which file you want to get data from for your table, such as PROD.CLASS or CUST.CLASS.
7. If using a parent/child relationship, use the **Parent File** field to indicate which parent your file is a child of.
8. Use the **Columns**, **Indexes**, and **Properties** tabs to complete your table information details.



Note: The bottom status bar displays the table version number and the user to whom the schema is currently open. The number in brackets indicates the number of changes you have made.

Column	Description
Column Name	By default the system populates the first line item with the record ID. This entry ensures that each row has its own record ID. 1. In the next column, enter the second column name and press **Enter**. 2. The SQL Table Column Maintenance window displays. 3. Use the window to enter further details about the column. 4. Repeat the data entry for each piece of user-defined Eclipse file that you want to map to this SQL table. **Note:** Use the underscore character (_) to separate words. The system does not accept periods and spaces.
File Attribute	Enter the File Attribute number you want to sync. If you need to sync a sub-value, use a comma to separate the value and sub-value, such as 3, x. For example, if you are pulling all the e-mail addresses for a customer, all the e-mails are sub-values.
Eclipse Data Type	Indicate the data type for inside Eclipse. Typically Eclipse does not need them, however, when you are converting data, such as decimals or dates, setting the data type here tells the conversion program what kind of data it is and then can import/convert to the right SQL data type.
SQL Data Type	The SQL data type, such as `nvarchar`, plus the character limit on the column listed in parentheses.

- Indexes

Column	Description
Index Name	By default, the system creates two indexes. • `idx_eclipse_id` - An index on the Eclipse record ID. The system needs this index for quick deletes for the inserts that occur during real-time Eclipse file record updates. • `pk_[table_name]` - The primary key index on the table. If you need to create additional indexes for BAQ/Dashboard performance on your user-defined SQL table, see Adding Indexes for ICE Through Eclipse.
Index Type	SQL Server Management Stuido indicates the index type that is best for your BAQ performance. For more information about indexes see, About Indexing for ICE Eclipse Tables and Adding Indexes for ICE Through Eclipse.
Index Is Unique?	SQL Server Management Stuido indicates the index type that is best for your BAQ performance. For more information about indexes see, About Indexing for ICE Eclipse Tables and Adding Indexes for ICE Through Eclipse.

- Properties

Field	Description
Table Processing Routine	The routine that tells the system how to process the table. For example, you may want all entries that start with A to go to a specific table. Table and Column processing routines may require custom programming assistance to use. They are not required for standard file mapping of raw data.
Row Counter Column	The row counter for each column.

Your initial setup is complete. You can always revisit the mapping and make adjustments if the results are not what you expected.

9. Save your changes and exit the window.
10. From the **System > Custom > Add On Products** menu, select SQL Sync Maintenance window.
11. Select the file you are mapping and from the **File** menu, select **Sync Enable**.
12. Use a SQL tool, such as SQL Server Management Stuidio, to confirm the replicated data is what you expect and want.
13. Go back to the SQL Sync Maintenance window for your user-defined table.
14. From the **File** menu, select **Close** to create a full version of your mapping.

ICE Frequently Asked Questions

The following are frequently asked questions for the Eclipse ICE interface:

1. **Is there a way to reset the number of errors displayed in SQL Sync Maintenance?**

   No. The error count is the number of errors for that file since the last full sync. It can be reset by resyncing the file. There is no way to manually reset the value.

2. **Is the update trigger at the database level or the application level? For example, does a delete command at the TCL level trigger an update?**

   The update trigger is based on an index at the database level. Any change to the file triggers the update, both at the TCL and application level. The index just logs the record ID as needing to be synced. It is picked up by a background process to be sent to SQL. The background process picks up the change, creates the SQL commands, sends the commands to SQL and processes the results. This process only takes 1-5 seconds, but are performed in batches to be efficient. The only exception to a call to the index, is when the system performs a "clear-file." This *does not* fire index updates. If you clear a file, you must resync the file to apply the changes. Eclipse does not perform any clear-file process from the Eclipse application.

3. **Are all attributes of the file synced to the database? If not, is there a way to add them?**

   The attributes that are mapped to SQL fiels are sent to the database. Eclipse created mappings for every field in the database. The only values not being sent are some of the calculated fields, fields that are redundant, or fields that contain data that is not useful or incorrect. You can modify the mappings or create entirely custom mappings, if needed. All mappings are accessible from the SQL Database Maintenance page where you can select by table name or by Eclipse file. In order to allow Eclipse to change files during upgrades, the mappings are divided into schemas: standard Eclipse mappings and user-defined mappings. All changes you make should be done in the eclipse_ud schema (Eclipse should force this), so that an upgrade does not remove any changes you make. A custom us schema can be configured to point to any file, standard or not.

4. **Is there a way to add files to the replication? For example, UD.PRODUCT, UD.LEDGER, or any UD file we might create?**

   Yes. You can create mappings using the eclipse_ud schema for any file that is not in a shared account. Shared accounts do not work because the index calls a program that would have to live in both accounts; then, there would have to be processors in both accounts and they would step on each other.

5. **Is there a way to restrict data that is synced without deleting it?**

   Yes. You can remove a field from the mapping and it will not be replicated or removed from Eclipse. Alternatively, you can set up field-level permissions in SQL to restrict access to the replicated data. However, those permissions need to be recreated if the file is turned off and back on for syncing, since Eclipse must delete and recreate the table.

6. **Is there a way to see what the backlog is on the processors so that we know if we need to increase our number?**

   Yes. Eclipse shows the current queue and send queue values under the DATAREP.SERVICE.MONITOR on the Phantom Status screen. The "queue" contains IDs of records which have changed in Eclipse, but have not been turned into SQL commands and the "send queue" are the records which have been converted to their SQL commands, but have not been successfully sent to the SQL server.

If the queue is large, more mapping processors are needed. The *mapping processor speed* is based on how busy the Eclipse box/disk are; while the *send processor speed* is based on the network and SQL box/disk speed.

There will always be transient spikes in these values since background operations will sometimes write to thousands of records at a time, and that can skew the numbers. Look for averages before you make changes to the number of processors.

Note: Changing the number of processors is slow because it has to pause everything and restart it all so records are not lost.

7. **When reviewing the replication phantoms in the Phantom Status, there are numbers at the end of the phantoms. Do these numbers get reset? If so, when and how?**

   These numbers indicate the records processed by that particular phantom. Eclipse uses that number to see if something is stuck or not balancing well. The numbers change to a message, if the process is taking a long time with a single record change. This can occur for LEDGER records with thousands of items. All the processors are restarted at midnight, daily, and the counts resets at that time.

Additional Notes About ICE

Take the following into consideration:

- If you change a mapping, the mapping does not go into effect until the file has been stopped from syncing and restarted, because Eclipse must change the table on the SQL side. This keeps Eclipse from performing this kind of action on a live database. Therefore, the table must be deleted and recreated when turning on a file.
- As of this release, Eclipse does not have a way to copy a mapping from the standard schema to the ud schema. This may be a future enhancement.
- The schemas have separate versions so that changes can be tracked and reverted, if needed.

Index

