



Real-Time SQL Server Synchronization

Enterprise Data Loader[®]
Training
Manual & Workbook
Version 3.3.58

Notices

This document was produced by Prolink Corporation, 148 Eastern Blvd, Glastonbury, CT 06033.

Telephone: 860-659-5928
Fax: 860-633-7309
E-mail: sales@prolinksoftware.com
Internet: [http:// www.prolinksoftware.com](http://www.prolinksoftware.com)

Warranty

Prolink, Inc. warrants that the software will operate according to the specifications described in the software reference manual and be free from known defects in materials or workmanship for a period of one year from the date of shipment. During this warranty period, Prolink will, at its option, repair, replace or provide a work-around solution to any items that prove to be defective. In order to qualify for such warranty service, a complete description of the problem, with appropriate documentation (such as results, program listing, sample part and program) should be forwarded to Prolink for our inspection. Such items will be returned to the customer if requested. Prolink does not warrant that the operation of the software will be uninterrupted or error-free. The information in this manual is subject to change without notice.

THESE WARRANTIES AND PROLINK'S LIABILITY HEREUNDER, ARE THE SOLE AND EXCLUSIVE WARRANTIES AND ARE EXPRESSLY IN LIEU OF ALL OTHER WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL PROLINK BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR OTHER DAMAGES INCLUDING, AND NOT BY WAY OF LIMITATION, LOSS OF PROFITS, AND WITHOUT REGARD TO THE FORM OF THE ACTION OR THE NATURE OF THE CLAIM WHICH IS MADE.

The information contained herein is based on the experience and knowledge relating to the subject matter gained by Prolink, Inc. prior to publication.

No patent license is granted or implied by this information.

Prolink, Inc. reserves the right to change this information without notice and makes no warranty, express or implied, with respect to this information. Prolink shall not be liable for any loss or damage, including consequential or special damages, resulting from the use of this information, even if loss or damage is caused by negligence or other fault on the part of Prolink, Inc.

Due to the nature of this material, a number of hardware and software products may be mentioned by name. In most cases the companies that manufacture the products claim these product names as trademarks. It is not our intent to claim these names or trademarks as our own.

Table of Contents

1. INTRODUCTION	1
1.1 WHAT IS THE DATA SOURCE	1
1.2 HOW IS THE DATA ORGANIZED	1
1.3 WHAT IF I DON'T HAVE A NETWORK	1
1.4 NETWORK CONFIGURATION	1
2. SETUP	2
2.1 HOW DO I GET A COPY OF QC-CALC	2
2.2 SQL SERVER	2
2.3 DATABASE SETUP	3
2.3.1 <i>Setting up the Database Automatically</i>	3
2.3.2 <i>Setting Up the Database Manually</i>	5
3. THE BASICS	11
3.1 THE SCHEDULE GRID	11
3.2 EDL LOAD SERVICE	11
3.3 SCHEDULED EVENTS	12
3.4 THE EVENT SUMMARY GRID	13
4. THE FILE MENU	14
4.1 NEW LOAD EVENT	14
4.1.1 <i>Welcome Screen</i>	14
4.1.2 <i>Choose Files to Sync</i>	15
4.1.3 <i>Choose Database</i>	16
4.1.4 <i>Schedule Event</i>	17
4.1.5 <i>Overrides</i>	18
4.1.6 <i>Finish</i>	19
4.2 NEW EXPORT EVENT	20
4.2.1 <i>Welcome Screen</i>	20
4.2.2 <i>Choose Database Screen</i>	21
4.2.3 <i>Choose Files to Sync Screen</i>	22
4.2.4 <i>Schedule Event Screen</i>	23
4.2.5 <i>Finish Screen</i>	23
4.3 NEW ARCHIVE EVENT	24
4.3.1 <i>Welcome Screen</i>	24
4.3.2 <i>Choose Main Database</i>	25
4.3.3 <i>Choose Files to Sync</i>	25
4.3.4 <i>Choose Archive Database</i>	26
4.3.5 <i>Set Archive Rule</i>	26
4.3.6 <i>Schedule Event</i>	27
4.3.7 <i>Finish</i>	27
4.4 EXIT	28
5. THE TOOLS – OPTIONS MENU	29
5.1 GENERAL SETTINGS	29
5.1.1 <i>Debug Mode</i>	29
5.1.2 <i>EDL Unique Description</i>	29
5.1.3 <i>Application Language</i>	29
5.1.4 <i>Minimize To Tray</i>	29
5.2 CAUSES & ACTIONS	30
5.3 PERFORMANCE	31
5.3.1 <i>Check dimension sources for updates</i>	31
5.3.2 <i>Store text factors as dates if conversion is possible</i>	32
5.3.3 <i>Check for audit changes in QC-CALC</i>	32
5.3.4 <i>Do Express Loading to speed up larger lists of QCC files</i>	32

Table of Contents

5.3.5 Max Records.....	32
5.3.6 Continue Loading Until Files are Complete.....	32
5.4 QCC MONITORING.....	33
5.4.1 Check for compacted/purged QCC files during load.....	34
5.4.2 Check for renamed QCC files during load.....	34
5.5 NOTIFICATIONS.....	35
5.5.1 Email Administrators On Error.....	35
5.5.2 Create individual error files for each error.....	36
5.5.3 Email administrator on event completion.....	36
5.6 EMAIL.....	36
5.7 SERVICE SETTINGS.....	37
5.8 UPDATE.....	38
6. THE HELP MENU.....	39
6.1 CONTENTS.....	39
6.2 LICENSE CONFIGURATION.....	39
6.3 RUN UPDATE.....	39
6.4 GATHER EDL INFORMATION.....	39
7. ADVANCED SQL SERVER FORMAT.....	40
7.1 DIAGRAM.....	40
7.2 GENERAL DESCRIPTION.....	40
7.3 TABLE DEFINITIONS.....	41
7.4 EXAMPLE QUERIES.....	47
8. A WORD ABOUT WINDOWS SERVICES.....	51
8.1 EDL CLIENT VS. EDL LOAD SERVICE.....	51
8.2 LOGIN PERMISSIONS FOR SERVICES.....	52
8.3 MAPPED PATHS/DRIVE LETTERS ILLEGAL FOR SERVICES.....	53
8.4 CONFUSION BETWEEN THE CLIENT AND SERVICE.....	54
9. PROLINK XML FORMAT.....	55
9.1 DESCRIPTION.....	55
9.2 CONVENTIONS.....	55
9.3 FIELD DEFINITIONS.....	55
9.3.1 <i>prolink_xml</i>	55
9.3.2 <i>part</i>	55
9.3.3 <i>trace</i>	56

1. Introduction

The Enterprise Data Loader (referred to as the EDL throughout the remainder of this document) is a Windows Service that provides a fast and easy method of moving CMM and gage inspection results directly into an RDBMS database. Specifically, it was written to work with Microsoft SQL Server. EDL typically resides on a central server at each plant and is responsible for synchronizing data in QCC files created by QC-CALC with data in a SQL database. These synchronizations are done through recurring scheduled events that are setup by the user to sync dynamic or specific lists of QCC files with particular databases. What sets EDL apart from a standard SQL export from QC-CALC is its ability to both insert new records into the database and update existing records as changes occur to the QCC files. In addition, EDL centralizes the exportation of QCC files so there is no need to setup individual copies of QC-CALC to export their data.

1.1 What is the Data Source

Currently, the sources of inspection data are the QCC files generated by QC-CALC Real-Time.

1.2 How is the Data Organized

In the database, the data is normalized to provide flexibility and reduce data redundancy. EDL currently uses the ProLink Advanced SQL Output database schema. The full description is found in [Advanced SQL Server Format](#) on page [40](#).

1.3 What If I Don't Have a Network

A network is required to successfully run EDL in your plant. The only other option would be to put all applications on the same PC. This would include your CMM software, QC-CALC Real-Time, EDL and MS SQL Server.

1.4 Network Configuration

Since CMMs are usually located in an inspection room, the inspection results are not readily available unless you are saving your results to a shared network server.

Introduction

2. Setup

EDL is distributed via the Internet and must be installed in your Windows PC or server using the standard SETUP.EXE program. Perform the following steps:

1. Install EDL on the computer that will do the synchronization. Since EDL is a small application that doesn't consume a lot of resources, it is recommended that you install it directly on the server containing SQL Server. This will reduce network traffic and increase performance.
2. The following screen is displayed on first run:



3. If you purchased the software click the **Activate Software** button, otherwise click the **Begin Evaluation** button.
NOTE: You may activate your software at any time.
4. You are now ready to begin using the Enterprise Data Loader (EDL). If you already own QC-CALC and SQL Server, you can begin exporting your own data to SQL Server immediately.

2.1 How Do I Get a Copy of QC-CALC

If you do not own a copy of QC-CALC you can download a copy for free and try it for 30-days by visiting the **Downloads** page of our website at www.ProlinkSoftware.com. Make sure you download QC-CALC v.3.2 or higher as it contains full support for EDL.

If you have a Prolink account you must login to download. If you do not have an account, you can create one instantly by entering an email address and password. Don't forget to download the CMM manual so you can obtain the simple QC-CALC instructions to ensure successfully data collection with your particular CMM or Gage.

2.2 SQL Server

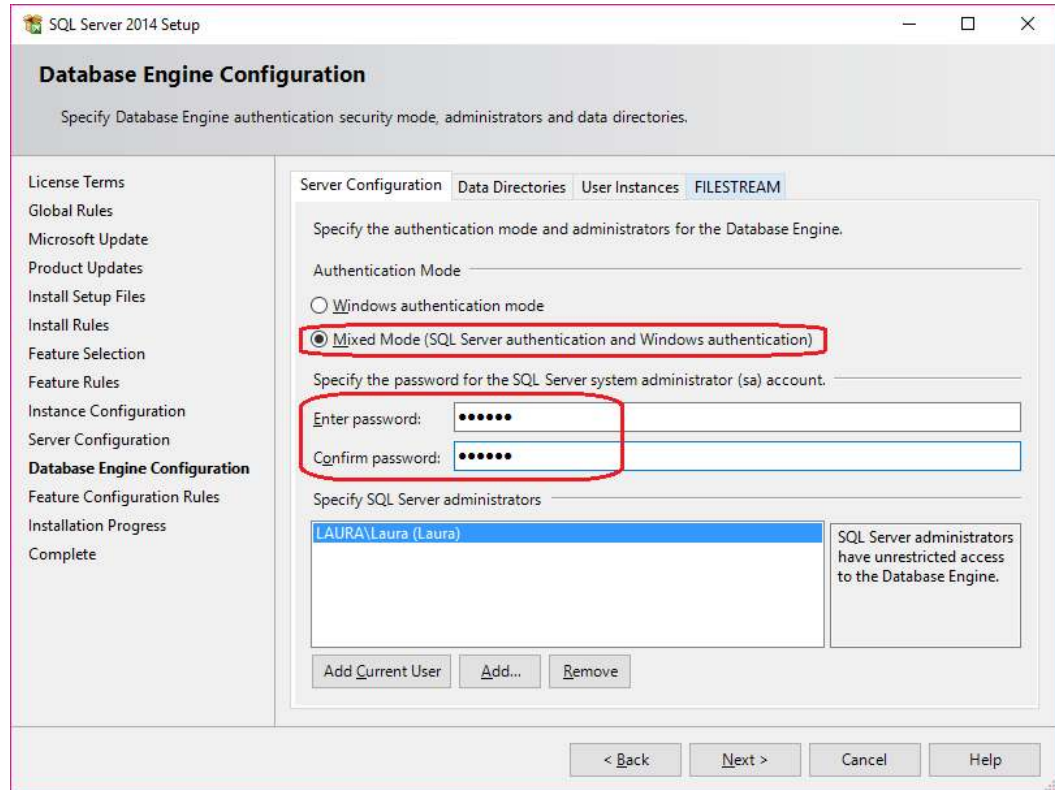
You must have a valid SQL Server database (version 2000 or higher) in order to send data via EDL. If you do not have SQL Server installed, you will need to either purchase a full version from Microsoft or you can download the latest SQL Server Express for free from Microsoft's website. This version has a couple of limitations, so it is not recommended for long-term use where a lot of data will be collected.

There is typically a link to download the latest version of SQL Server Express Edition for free from:

<http://www.microsoft.com/sqlserver>

Introduction

During the SQL Server install process, you will be asked which **Authentication Mode** to use. **Mixed Mode** (allows both SQL Server and Windows authentication) is the highly recommended method.



NOTE: The password for the SQL Server system administrator account entered on this screen will be needed later in the EDL setup process, so make sure you remember what was entered here.

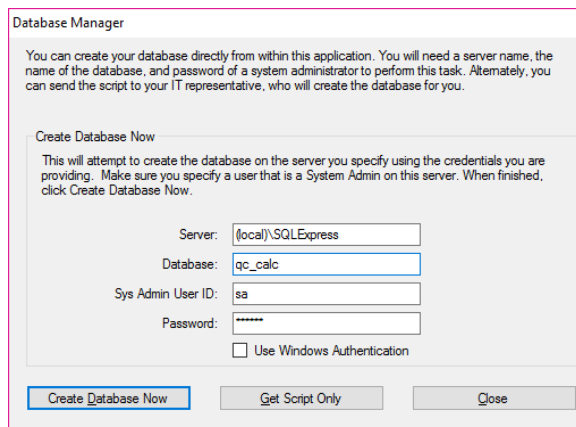
2.3 Database Setup

If your database does not exist, it is fairly simple to create. Either you can do it yourself manually or EDL can create the database for you.

2.3.1 Setting up the Database Automatically

You can choose to create the database automatically directly through either EDL or ERS.

1. In either application choose **Help – Create Database (Create Settings Database in ERS)**. The following screen will appear.



Introduction

2. Enter the server name and SQL instance (if necessary). If you are running SQL Express, this is “\SQLExpress” by default.
3. Enter the name you would like for your database. By default, we suggest “qc_calc”.
4. Enter “**sa**” as the System Administrator ID and the password you specified when installing SQL Server ([SQL Server](#) section on page 2).
5. If you are getting this information from your IT department, please make sure you get an ID that has System Administrator privileges, as the ID will be used to create a database, add logins to the server, and add a user to the database.

NOTE (for IT Professionals): The SA user account is only needed for the initial database creation and will not be used going forward. When the script is run to create the database, it creates a separate user account called “qccadmin”. This account is given db_datareader, db_datawriter, and db_owner access to this database only. This ID will then be presented via a message box at the end of the script so the user can use this for reporting. The default password for the ‘qccadmin’ account is ‘NimdaccQ12’.

NOTE (for IT Professionals): The db_owner role was added to allow the user to update the database when new versions of EDL are available. This can be removed if there are security concerns.

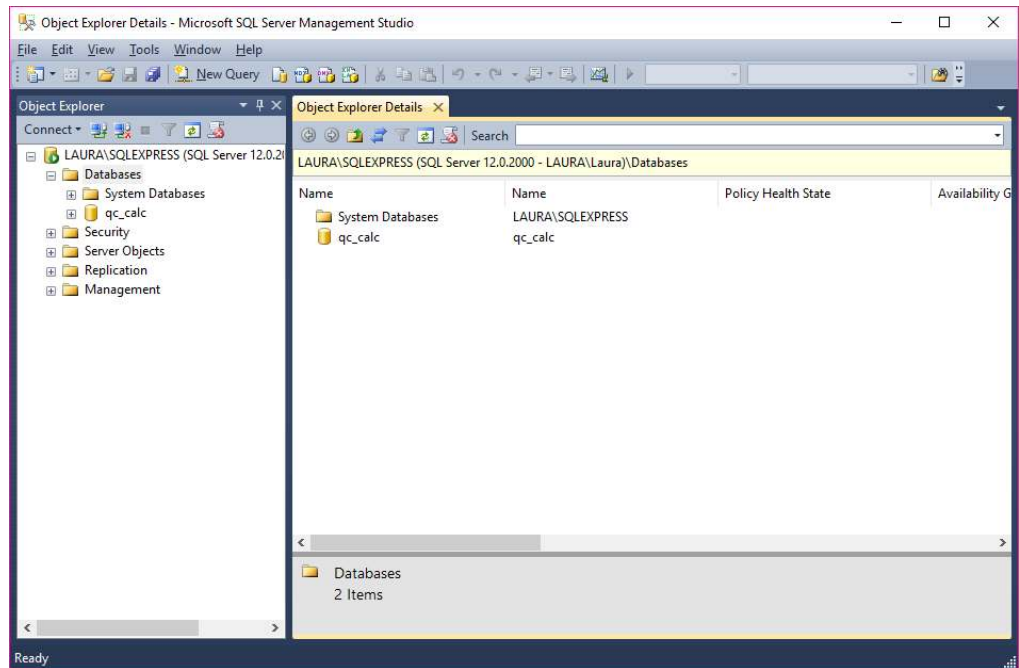
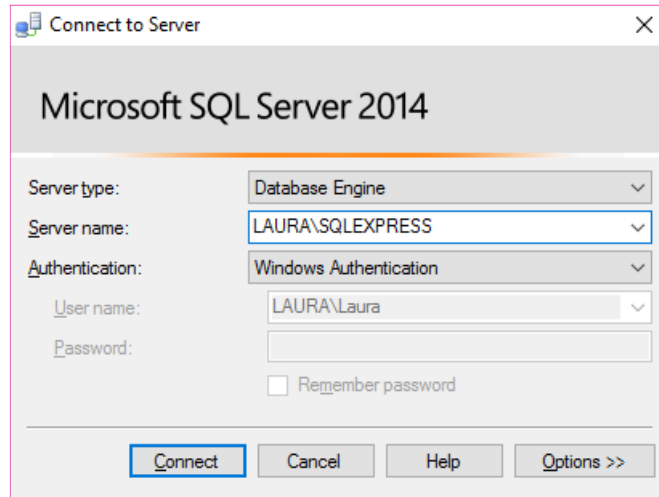
6. Create Database Now vs. Get Script -
 - a. When ready, click **Create Database Now**. Upon successful creation, you will see a message box containing the user and password to use for reporting. Please take care to write this down for future use. You are now ready to run EDL and ERS (they share the same database). You can skip section 2.4.
 - b. If there are errors creating the database, you may have to do it manually. If so, please read the next section.
 - c. You can alternately click **Get Script Only** to get the script that will be run. Please be aware that the script displayed is only for the tables of the database and does not include the actual CREATE DATABASE or other security related commands. The script in the window can be copied/pasted to SQL manually if needed.

Introduction

2.3.2 Setting Up the Database Manually

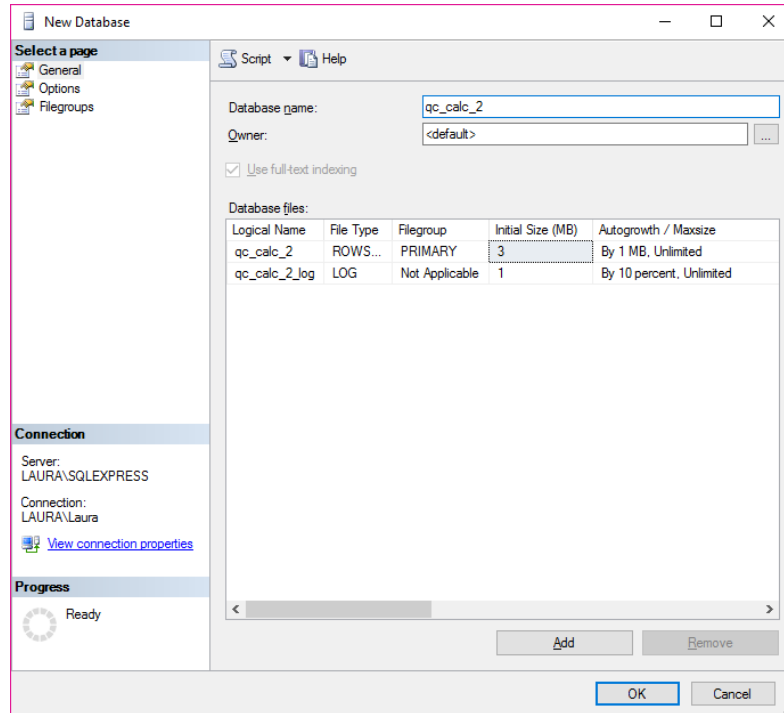
This section will help you create a database manually by adding the tables and user permissions. These steps will create the tables of your database and you will be ready to being using EDL and ERS once you have finished.

1. When **SQL Server Management Studio Express** opens, you will see the following screen. Click **Connect** to connect to the SQL Server using Windows Authentication for now.

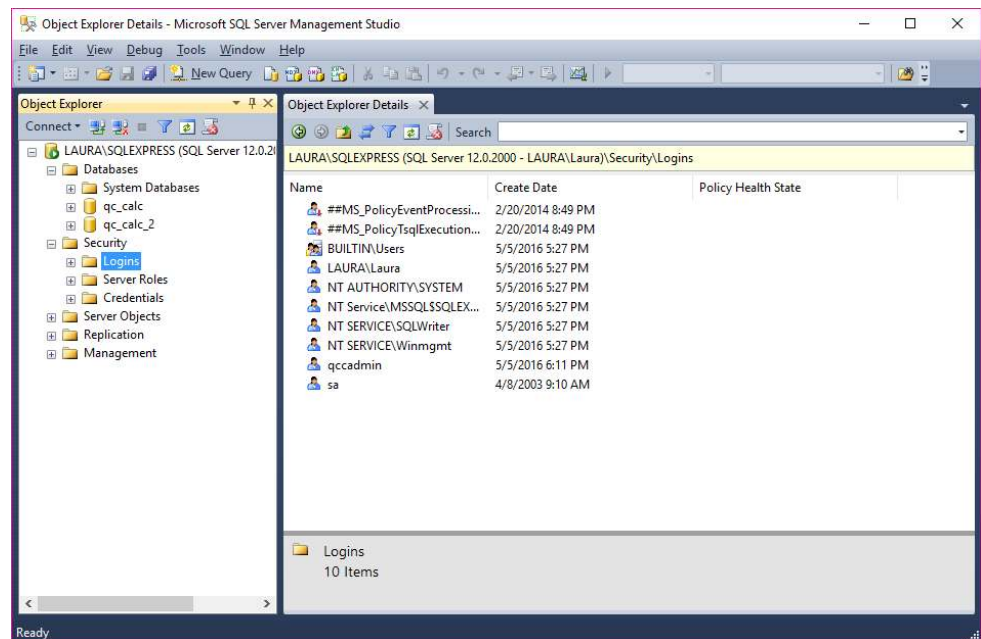


Introduction

2. Right click on **Databases** and select **New Database**.



3. Give your database a name. For this example, we will be calling our database **qc_calc_2** since the standard qc_calc name was used during the **Create Database** operation. Click **OK**.
4. Now, expand **Security** on the left side of the screen.



Introduction

5. Right click on **Logins** and select **New Login**.

Login - New

Select a page

- General
- Server Roles
- User Mapping
- Securables
- Status

Script Help

Login name: qccadmin

Windows authentication

SQL Server authentication

Password:

Confirm password:

Specify old password

Old password:

Enforce password policy

Enforce password expiration

User must change password at next login

Mapped to certificate

Mapped to asymmetric key

Map to Credential

Mapped Credentials

Credential	Provider
------------	----------

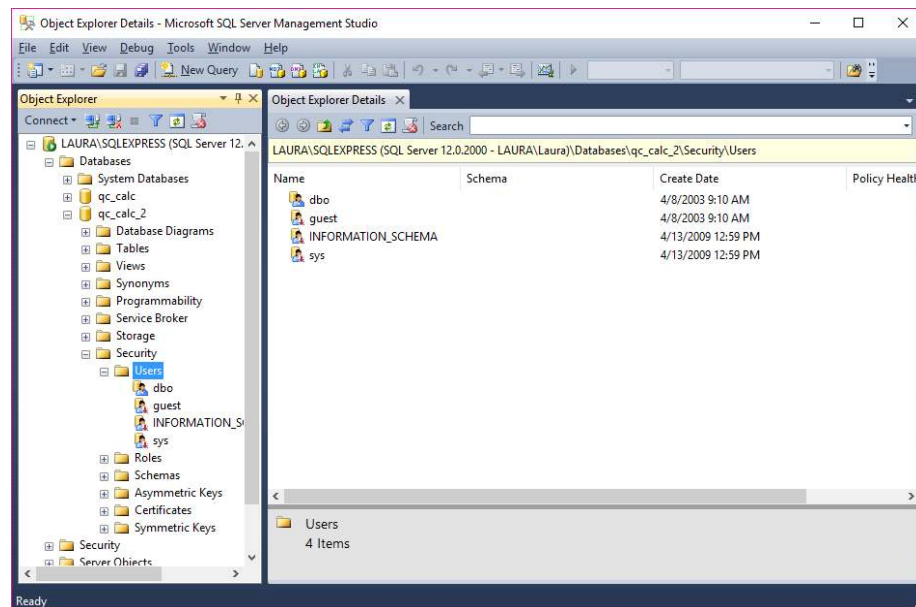
Remove

Default database: qc_calc_2

Default language: <default>

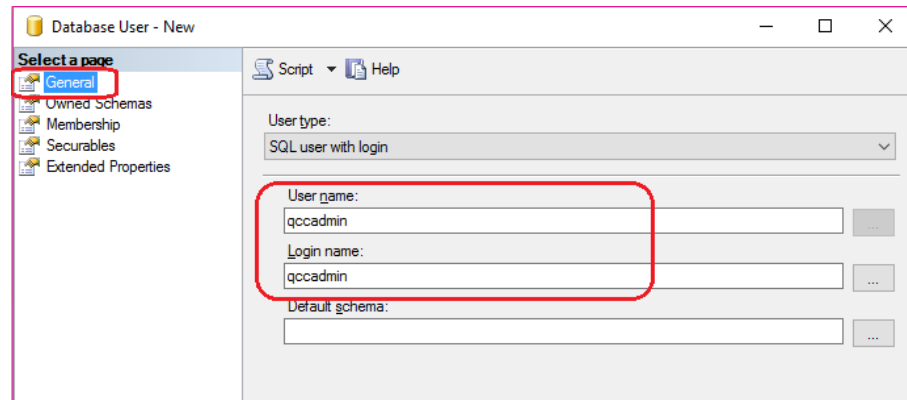
OK Cancel

6. Create a **Login name**. For this example we'll be using **qccadmin**
7. Make sure that **SQL Server Authentication** is selected and then create and confirm a **Password**. Make sure that **Enforce password policy** is unchecked.
8. Select the database you just created as your **Default database**, for this example it would be **qc_calc_2**. Click **OK**.
9. Expand **Databases** on the left hand side, and then expand your default database. Within that, expand **Security**.

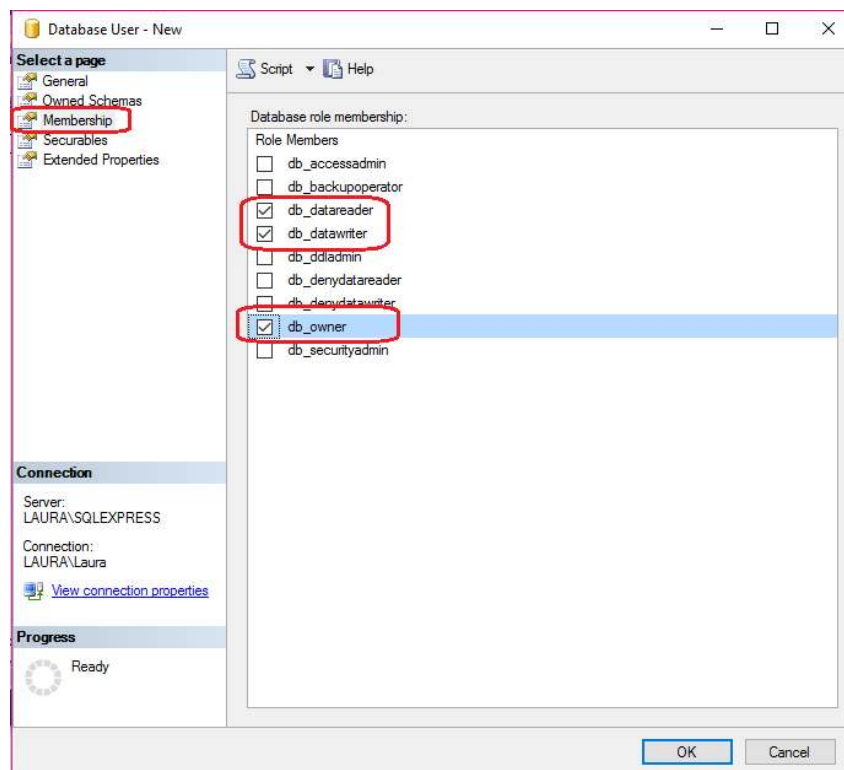


Introduction

10. Right click on **Users** and select **New User**.
11. On the **General** page, set your **User Name** and **Login Name** to match the one you just created.



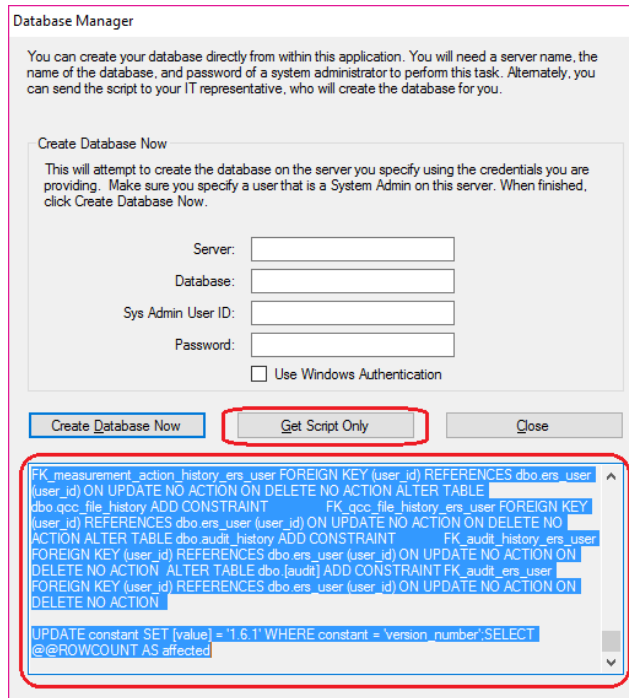
12. On the **Membership** page, under **Database Role Members** check **db_datareader**, **db_datawriter**, and **db_owner**. Click **OK**.



13. In the main **Microsoft SQL Server Management Studio Express** window, click **New Query**.

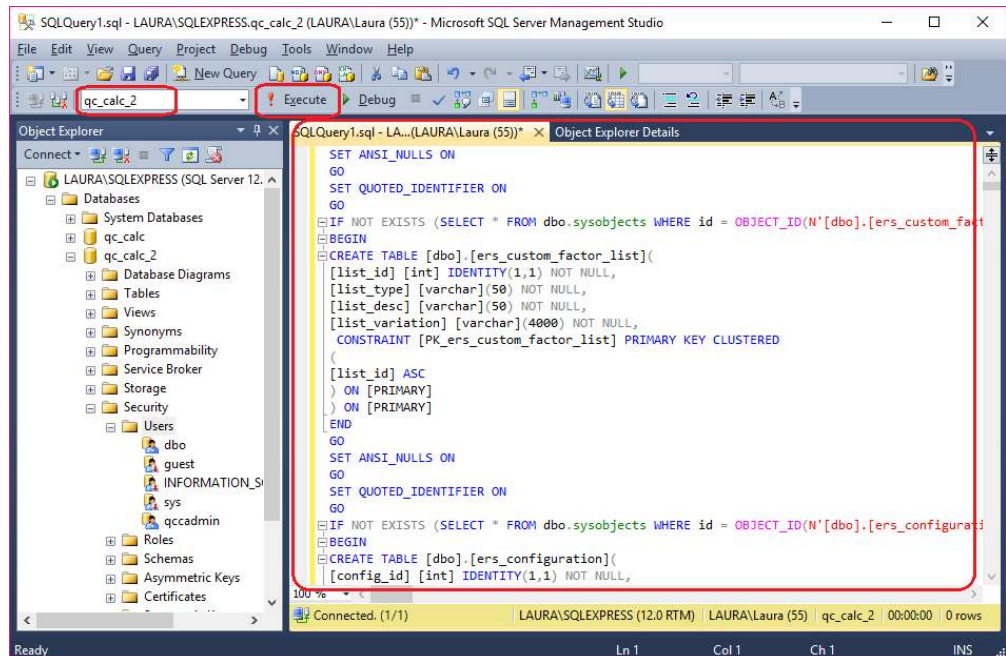
Introduction

14. In EDL, choose the **Help – Create Database** menu then click the **Get Script Only** button.



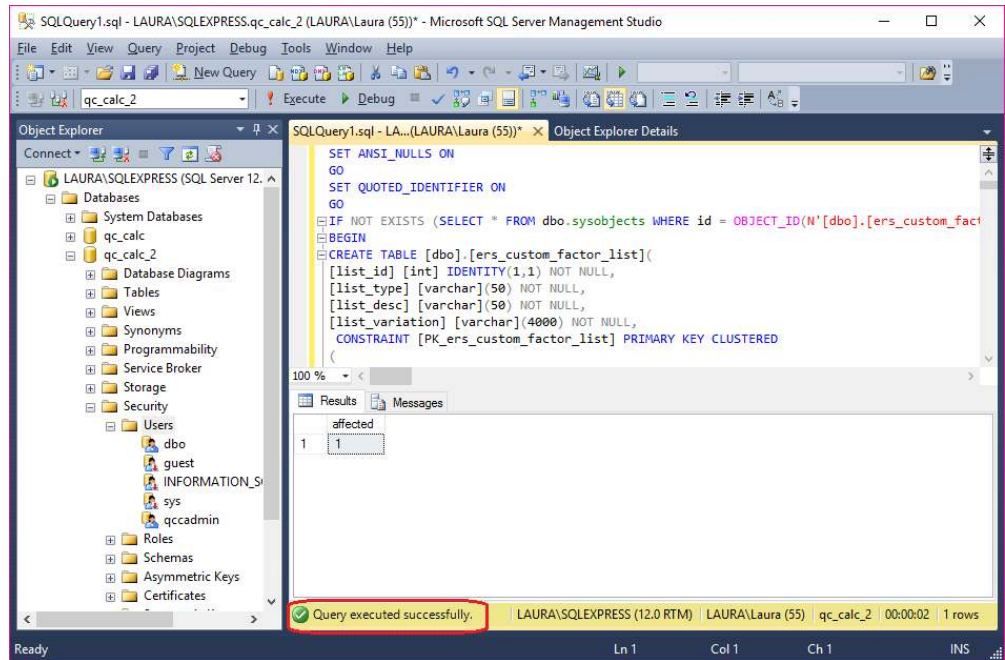
15. Click in the area that appears at the bottom of the screen, click the CTRL+Home keys to go to the very top of that area, then click the CTRL+Shift+End keys to highlight the entire script. Right-click on the highlighted text, and choose **Copy** to get the script into the clipboard.

16. Next, go back to the **Microsoft SQL Server Management Studio Express** window and paste your clipboard into your new query window.



Introduction

17. Make sure the database you created (**qc_calc_2** in our example) is shown in the upper left of the screen and click **Execute**.



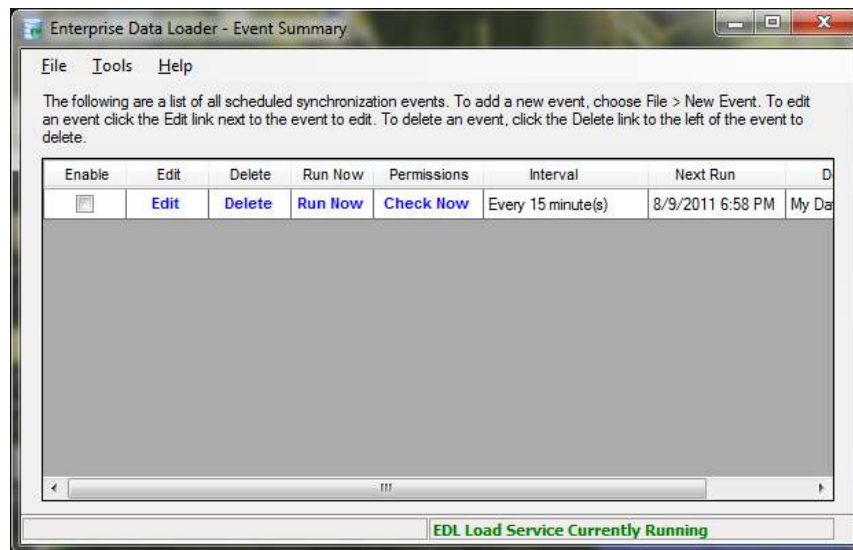
3. The Basics

Before getting into the details about this software product, let's quickly cover the basics. Double click the EDL icon on your desktop to start the EDL Client.

On first launch the **Event Summary** screen is opened for you so you can see currently scheduled events. When you first run EDL, you won't have any events scheduled so the main grid will be empty. A screen similar to the one shown below contains two areas of interest.

3.1 The Schedule Grid

As of version 3.3, EDL is actually split into two applications; the EDL Client and the EDL Load Service. The EDL Client uses a similar interface as older versions of EDL, but simply schedules events, maintains options, and controls whether or not the EDL Load Service is running.



The EDL Load Service runs in the background and has no interface (other than the EDL program itself (Client)). The EDL Load Service is responsible for actually doing the synchronization to SQL Server. Running the service is explained in the next section.

3.2 EDL Load Service

The first time you start EDL, you may notice a red message at the bottom of the screen indicating that the EDL Load Service is not currently running. As of version 3.3, EDL is now a Windows Service. This means that it will continue to run as long as the PC/Server is running even if nobody is logged in. This makes it convenient for IT deployments since IT typically doesn't leave servers logged in permanently.

To start the service, select **Tools – Start Load Service**. At this point, the service will start and the message at the bottom of the screen should turn green telling you the service is currently running.

To stop the service, select **Tools – Stop Load Service**. Generally, you should leave the service turned on at all times so it is synchronizing your QCC files with the server. When the service is stopped, it is not synchronizing your QCC files with SQL Server.

The Basics

You do not need to have the EDL Client (event summary screens, wizard, etc.) running in order for the service to run since they are two separate applications. The EDL Client can run either in the Windows system tray (lower right corner of screen near the clock) or be shut down once events are scheduled and options chosen.

3.3 Scheduled Events

EDL has been designed to run unattended and is based on scheduled load and export events. Once each minute EDL looks at the schedule and determines which events are due to be run.

Load events are events that synchronize existing individual QCC files on your network with the SQL Server database. These events continually look for changes in your local QCC files and move those changes into SQL. Each scheduled Load Event is broken into 3 major parts; which QCC files will be synchronized, which database will be used as the data repository, and how often the data will be synchronized. When you create a new event, you use the **Schedule Wizard** discussed in the [New Load Event](#) section on page 14.

Export events are events that attempt to synchronize the data in SQL Server with a downstream system (such as an MES system or other large corporate database). They are similar to load events, but instead generate an XML output file containing all records that changed inside the SQL database. Whenever a change occurs to a record in the SQL Server database (whether it is a new record or it has been updated in some way), the export event will get a full copy of the record and export it via the output xml file. All records that are exported contain a unique key that can be used by your downstream system to replace the existing record there. The XML is exported as the standard Prolink XML format and contains all information about the record. Rather than export just what changed, it was decided that it would be easier to simply export a copy of the record in its entirety. This is because we do not know the downstream system and whether or not it would have the capability to import just individual changes. The export event setup is similar to the load event, but in the opposite order. Again, there are three things that must be determined; the database to watch for changes, the parts that you wish to export (and export path), and the schedule on which the export will run.

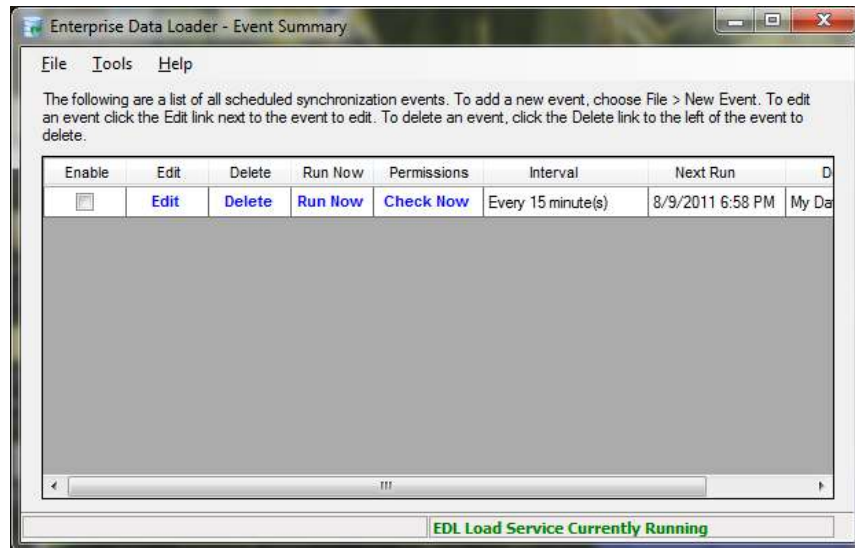
Once created, events appear in the main **Schedule Grid** on the **Event Summary** screen. You can create as many load and export events as desired.

The Basics

3.4 The Event Summary Grid

This grid contains the currently scheduled events. In this example, there is one event scheduled and it is shown in the grid. On each event, the first 4 columns are actions to be performed on the events:

- **Enable/Disable** - An event will only run if it is enabled. When you enable an event by checking the checkbox, the event is scheduled based on the next time it has been set to run in the **Recurrence Interval**.



- **Edit** - This will open the **Event Details Wizard** and allow you to make changes to existing events. See the [Welcome Screen](#) section on page [14](#) for details.
- **Delete** - You will be asked to confirm this before the deletion occurs.
- **Run Now** - Runs the event now rather than waiting until the scheduled time. This does not affect the **Next Run** time.
- **Check Now** - is used to check the availability of the QC-CALC files. This is needed to ensure EDL has the permissions to read the files. All files displayed are accessible to EDL. Any missing files must have their permissions checked. See [A Word about Windows Services](#) on page [51](#) for more details.

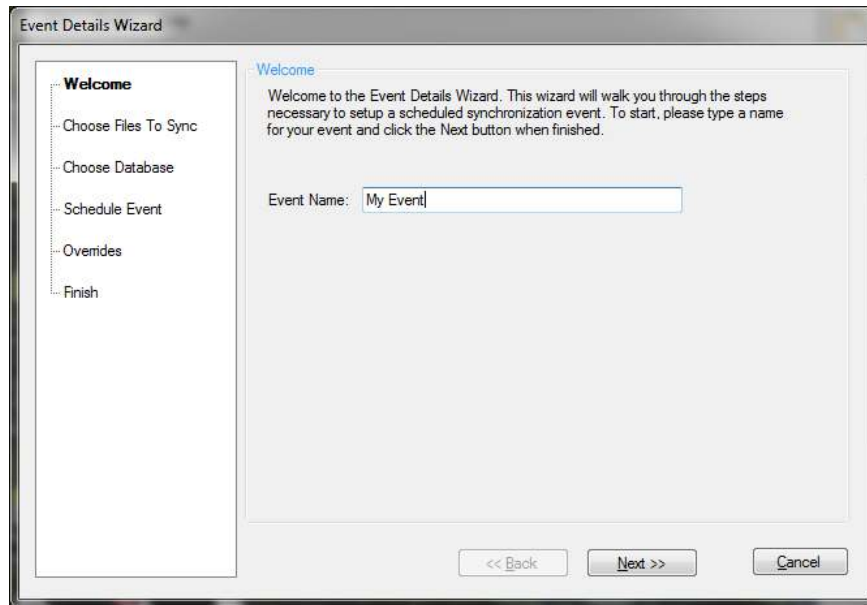
4. The File Menu

4.1 New Load Event

Use this menu to create a new event. This launches the **Event Details Wizard** with a new event and allows you to schedule it. The Event Details Wizard is used to both create new and edit existing scheduled events. The wizard (shown below) walks you through the 5 steps to create or update an event. These are outlined in the following 5 sections.

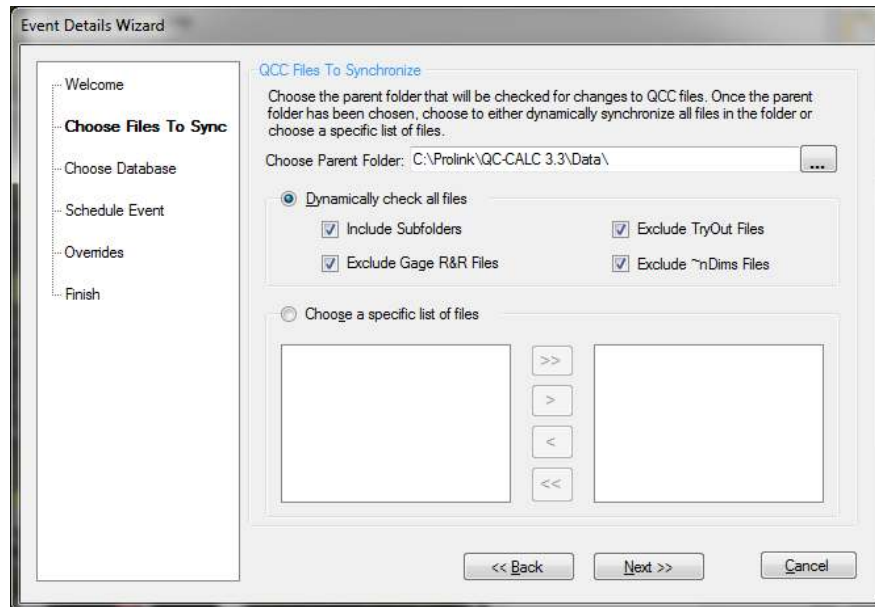
4.1.1 Welcome Screen

This is a simple screen that starts off the creation or editing of an event. On this screen you name your event.



4.1.2 Choose Files to Sync

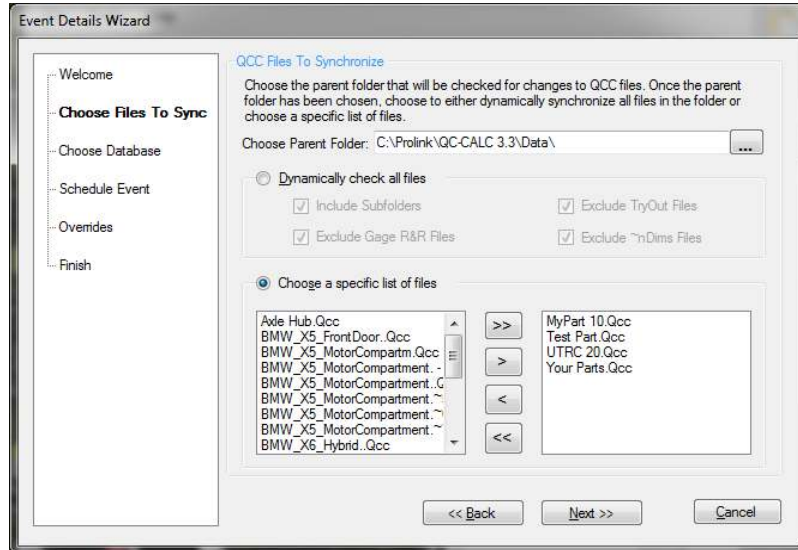
This screen allows you to specify which QCC files you want to synchronize in the database. You start by **Choosing a Parent Folder** in which your QCC files reside. This path must be a UNC style path (i.e. [\\myserver\share](#)) if you are pointing it to a network location. Once the path has been entered, choose a synchronization type and the options to go with it. There are two types of synchronizations; dynamic and static.



Dynamically check all files - This method automatically synchronizes all files in the specified parent folder. This is useful if you want to synchronize all QCC files and do not want to go back and specify new files whenever new QCC files are created. With this type, there are a few options:

- **Include Subfolders** - If checked, this option synchronizes not only the QCC files in the parent folder, but also all subfolders of the parent folder.
- **Exclude TryOut Files** - If checked, EDL will skip any files that have `_TryOut.QCC` in the filename. Files of this type are typically setup/test (non-production) data and you may not want to clutter up your SQL database with data from them.
- **Exclude Gage R&R Files** - If checked, EDL will skip any files that have `_GRR.QCC` in the filename. These are typically Gage R&R studies and you may not want them in the SQL database.
- **Exclude ~xDims Files** - If checked, EDL will skip any files that have `~xDims.QCC` in the filename where x is the number of dims (i.e. `widget~5Dims.qcc`, `widget~20Dims.qcc`, etc.). These files are automatically produced by QC-CALC if you have it set to rename the existing file when the number of dimensions does not match.

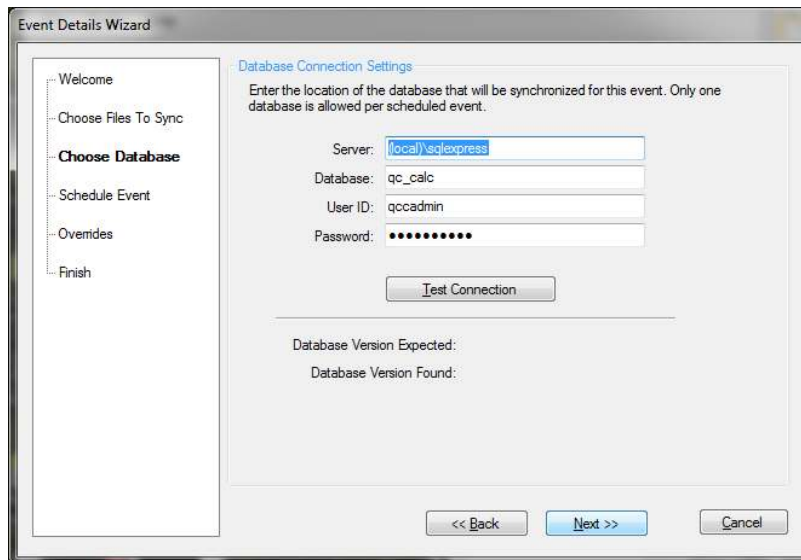
Choose a specific list of files - This static method allows you to choose specific files from a list. When you choose this method, a list of all QCC files in the parent folder appears on the left side.



Simply move the files you want to include to the list on the right. This method is useful if you have a very specific set of files you want to synchronize and it is unlikely to change very often.

4.1.3 Choose Database

This screen allows you to specify which database will receive the data from the QCC files. Simply fill in the name or IP address of the **Server**, the **Database** name, the **User ID** that will be used, and the **Password**.

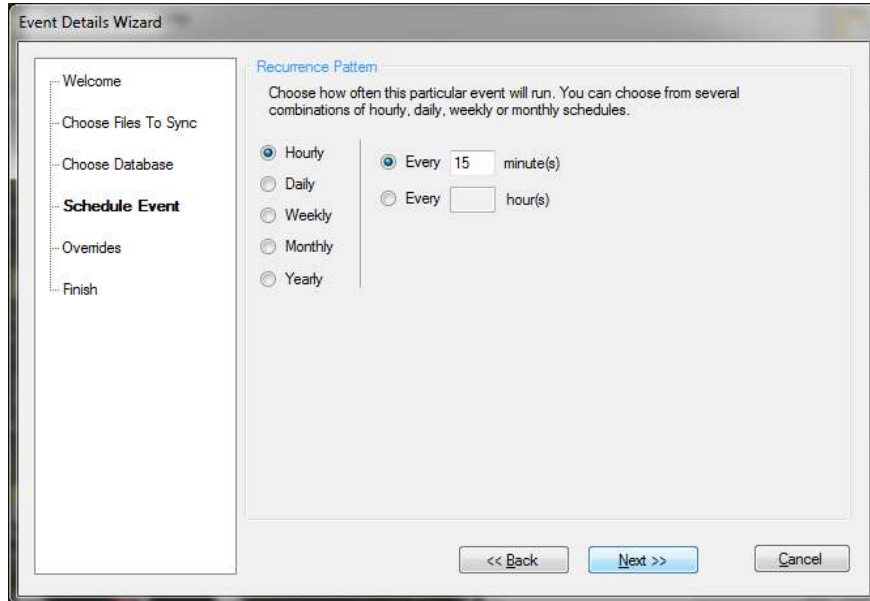


After entering the information, click the **Test Connection** button to make a quick connection and make sure you can get to the database. This button makes a connection only and does not update the database or check permissions on the tables.

NOTE: The **User ID** used must have sufficient permissions to read from and write to all tables in the database.

4.1.4 Schedule Event

This screen allows you to specify how often the synchronization is to take place. There are several recurrence options to choose from. Each of these is outlined below.



Recur	Description	Picture
Hourly	Allows you to scheduled events either every x minutes or x hours.	<input checked="" type="radio"/> Hourly <input type="radio"/> Daily <input type="radio"/> Weekly <input type="radio"/> Monthly <input type="radio"/> Yearly <input checked="" type="radio"/> Every 5 minute(s) <input type="radio"/> Every [] hour(s)
Daily	Allows you to schedule a specific time every x days.	<input type="radio"/> Hourly <input checked="" type="radio"/> Daily <input type="radio"/> Weekly <input type="radio"/> Monthly <input type="radio"/> Yearly Every 2 day(s) at 12:00:00
Weekly	Allows you to schedule a specific time to run on multiple days of each week.	<input type="radio"/> Hourly <input type="radio"/> Daily <input checked="" type="radio"/> Weekly <input type="radio"/> Monthly <input type="radio"/> Yearly Run every week at 12:00AM on <input type="checkbox"/> Sunday <input checked="" type="checkbox"/> Monday <input type="checkbox"/> Tuesday <input checked="" type="checkbox"/> Wednesday <input type="checkbox"/> Thursday <input checked="" type="checkbox"/> Friday <input type="checkbox"/> Saturday
Monthly	Allows you to schedule a specific day and time of each month. If the day of the month doesn't exist for the month you're running it will take the closest matching date. For example, if you schedule it for the 31 st of the month, it will run on February 28 th (assuming no leap year), April 30 th , etc.	<input type="radio"/> Hourly <input type="radio"/> Daily <input type="radio"/> Weekly <input checked="" type="radio"/> Monthly <input type="radio"/> Yearly Every month on the 15 day at 1:00PM

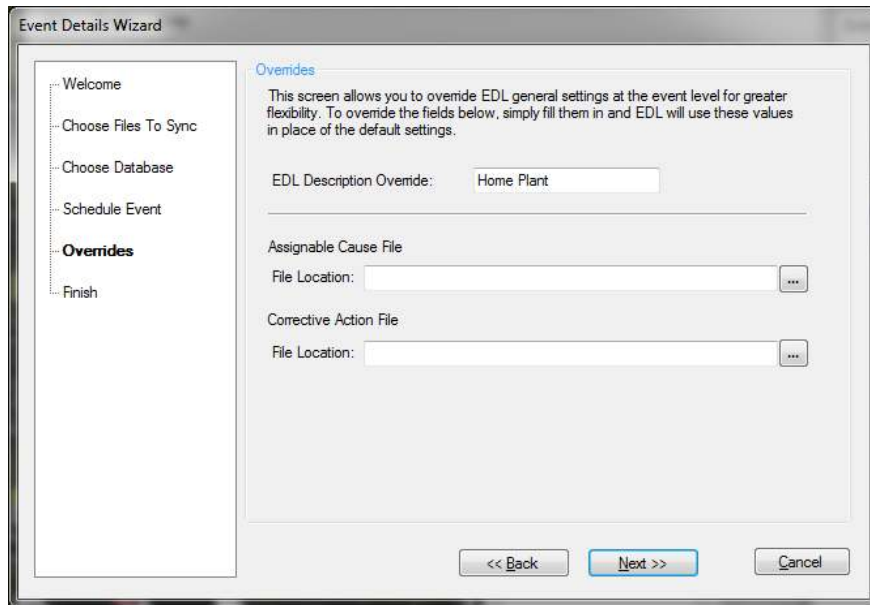
File Menu

Yearly	Allows you to schedule a specific date and time to run each year. Simply choose the month from the drop down and type the day. This must be a valid date.	<input type="radio"/> Hourly <input type="radio"/> Daily <input type="radio"/> Weekly <input type="radio"/> Monthly <input checked="" type="radio"/> Yearly Every <input type="text" value="January"/> at <input type="text" value="15"/> at <input type="text" value="11:00PM"/>
---------------	---	--

4.1.5 Overrides

EDL Description Override

The **EDL Unique Description** (discussed in the [General Settings](#) section on page 29) is text you can add to tag the origin of the data. The same description can be overridden here for each event you create. For example, say you have 3 plants saving data to one database on the Internet. The 3 locations are Singapore, Texas, and Ireland. Add something like Sing, TX, and IRE in the description and every part will be tagged with this label allowing you to compare the same parts made in three separate locations. If you leave these fields blank, EDL will attempt to use the General Settings (assuming they are set).



Assignable Cause / Corrective Action Files

All QC-CALC's should be using their assignable causes and corrective actions from a network location. However, in certain cases there are multiple network locations within the same plant. This event level override allows you to specify cause and action files that apply for the QCC files in this event only. i.e. [\\myserver\SPCData\Admin\](#).

4.1.6 Finish

This screen gives you an English description of your event and allows you to enable/disable the event before finishing up.

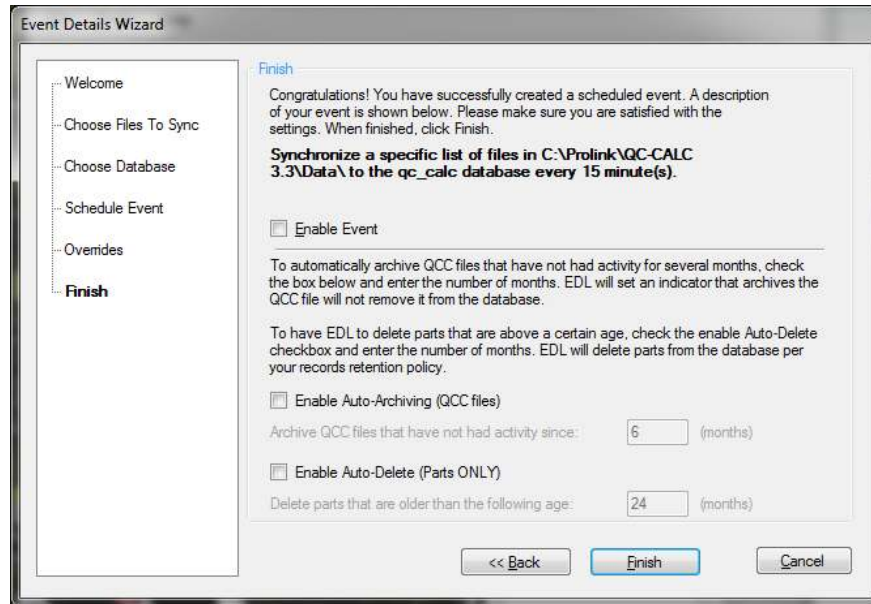
Auto-Archiving Files

You also have the ability to Auto Archive QC-CALC files in the database. Large lists of QC-CALC files can make locating & selection cumbersome. By using the Auto-Archiving option, QC-CALC essentially hides old file names. If the QC-CALC file has not had new parts added to it it is considered old and will disappear from the list after the set number of months. Please keep in mind that no archiving of actual data is being done here. It is simply a flag on the file to tell ERS not to load it when loading lists of QCC files. For archiving of data, please see the [New Archive Event](#) section on page 24.

Auto-Delete Parts

Sometimes you may wish to purge the SQL database of old parts (not part files). You can request ERS to delete very old records simply by checking this checkbox and entering a number of months until the parts are removed.

Click **Finish** to save the event.



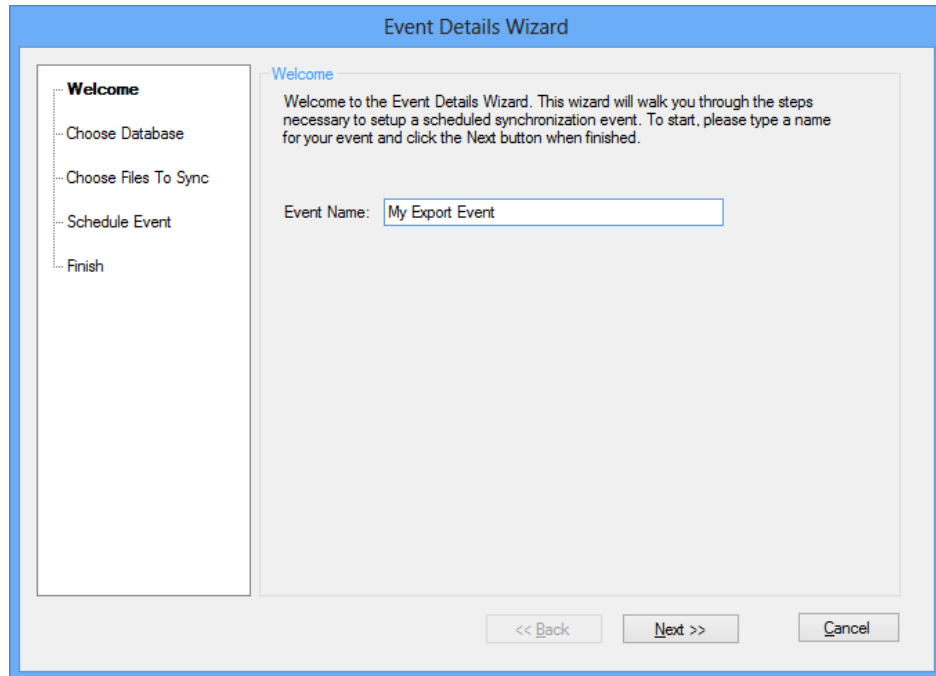
The screenshot shows the 'Event Details Wizard' window at the 'Finish' step. On the left is a vertical navigation pane with steps: Welcome, Choose Files To Sync, Choose Database, Schedule Event, Overrides, and **Finish**. The main area is titled 'Finish' and contains the following text: 'Congratulations! You have successfully created a scheduled event. A description of your event is shown below. Please make sure you are satisfied with the settings. When finished, click Finish.' Below this is a bolded event description: 'Synchronize a specific list of files in C:\Prolink\QC-CALC 3.3\Data\ to the qc_calc database every 15 minute(s)'. There are three checkboxes: 'Enable Event' (unchecked), 'Enable Auto-Archiving (QCC files)' (checked), and 'Enable Auto-Delete (Parts ONLY)' (checked). The 'Enable Auto-Archiving' section includes a text label 'Archive QCC files that have not had activity since:' followed by a text box containing '6' and '(months)'. The 'Enable Auto-Delete' section includes a text label 'Delete parts that are older than the following age:' followed by a text box containing '24' and '(months)'. At the bottom are three buttons: '<< Back', 'Finish', and 'Cancel'.

4.2 New Export Event

Use this menu to create a new export event. This launches the **Event Details Wizard** with a new event and allows you to schedule it. The Event Details Wizard is used to both create new and edit existing scheduled load events. The wizard (shown below) walks you through the steps to create or update an event.

4.2.1 Welcome Screen

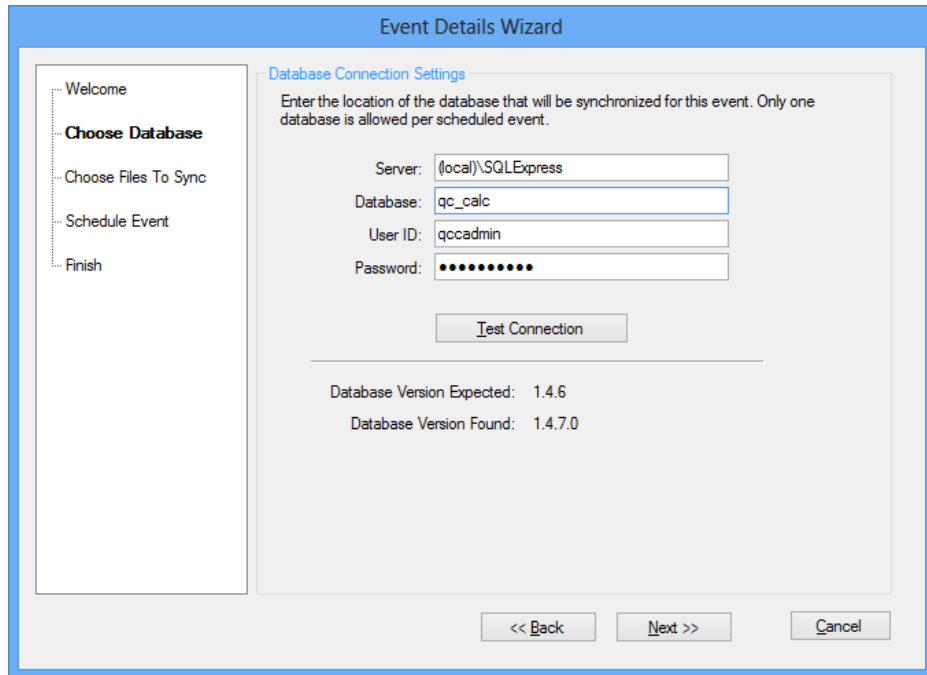
This is a simple screen that starts off the creation or editing of an event. On this screen you name your event.



The screenshot shows the 'Event Details Wizard' window. On the left is a vertical navigation pane with the following steps: 'Welcome' (selected), 'Choose Database', 'Choose Files To Sync', 'Schedule Event', and 'Finish'. The main content area is titled 'Welcome' and contains the following text: 'Welcome to the Event Details Wizard. This wizard will walk you through the steps necessary to setup a scheduled synchronization event. To start, please type a name for your event and click the Next button when finished.' Below this text is a text input field labeled 'Event Name:' containing the text 'My Export Event'. At the bottom right of the window are three buttons: '<< Back', 'Next >>', and 'Cancel'.

4.2.2 Choose Database Screen

This screen allows you to specify which database will be used as the source of the data. Simply fill in the name or IP address of the **Server**, the **Database** name, the **User ID** that will be used, and the **Password**.



The screenshot shows the 'Event Details Wizard' window with the 'Database Connection Settings' tab selected. On the left, a vertical navigation pane lists the steps: Welcome, Choose Database (highlighted), Choose Files To Sync, Schedule Event, and Finish. The main area contains the following fields and controls:

- Server:** (local)\SQLExpress
- Database:** qc_calc
- User ID:** qccadmin
- Password:** [masked with 10 dots]
- Test Connection** button
- Database Version Expected: 1.4.6
- Database Version Found: 1.4.7.0
- Navigation buttons: << Back, Next >>, and Cancel

After entering the information, click the **Test Connection** button to make sure you can get to the database. This button makes a connection only and does not update the database or check permissions on the tables.

NOTE: The **User ID** used must have sufficient permissions to read from and write to all tables in the database.

4.2.3 Choose Files to Sync Screen

This screen enables you to choose the export path for your output xml files. Much like the load events, you can choose to either dynamically export all part files found or statically choose specific part files from the list. If using the static method, move the desired part files from the list on the left to the selected list on the right.

The screenshot shows the 'Event Details Wizard' window, specifically the 'Export Files' step. On the left is a navigation pane with steps: Welcome, Choose Database, **Choose Files To Sync**, Schedule Event, and Finish. The main area contains instructions: 'The following is a list of part files that were found in the database you chose on the prior screen. Please type or browse to a valid path where the output data will be placed. Then choose which files you would like to export or choose dynamic to export them all.' Below this are fields for 'Export Path' (C:\MyExport), 'EDL Desc' (--All--), and two radio buttons: 'Dynamic - Export all part files' (selected) and 'Static - Choose specific part files from the list below'. There are two lists: 'Available' (containing part numbers and descriptions like '123988-Op20', '1473-00 mid shoulder rad & groove', etc.) and 'Selected' (empty). Navigation buttons '<< Back', 'Next >>', and 'Cancel' are at the bottom.

In addition, there is an option to filter the part files by EDL Description. If you choose to filter by EDL Description, both the dynamic and static methods will follow the filter. As an example, if you choose to filter on all parts made in NYC, it will dynamically find all part files with that EDL description. Part files without that EDL description will not be exported.

NOTE: As a review, the EDL Description is typically used to group QCC files by plant, department, or any other grouping of your choosing. It is a helpful way to tell files apart that may be duplicated (i.e. you have one large database for multiple plants and make the same part numbers at more than one plant).

4.2.4 Schedule Event Screen

Just like the load event, this screen allows you to specify how often the synchronization is to take place. There are several recurrence options to choose from. These are outlined in section 4.1.5 above.

The screenshot shows the 'Event Details Wizard' window. On the left is a vertical navigation pane with steps: Welcome, Choose Database, Choose Files To Sync, **Schedule Event**, and Finish. The main area is titled 'Recurrence Pattern' and contains the text: 'Choose how often this particular event will run. You can choose from several combinations of hourly, daily, weekly or monthly schedules.' Below this text are two columns of radio button options. The first column has: Hourly, Daily, Weekly, Monthly, and Yearly. The second column has: Every 5 minute(s) (with '5' in a text box), and Every hour(s) (with an empty text box). At the bottom right are three buttons: '<< Back', 'Next >>', and 'Cancel'.

4.2.5 Finish Screen

This screen allows you to enable the export event upon completion. Check the box if you wish to do this.

The screenshot shows the 'Event Details Wizard' window at the 'Finish' step. The left navigation pane has 'Finish' selected. The main area is titled 'Finish' and contains the text: 'Congratulations! You have successfully created a scheduled export event. When finished, click Finish to save your event.' Below this text is a checkbox labeled 'Enable Event', which is currently unchecked. At the bottom right are three buttons: '<< Back', 'Finish', and 'Cancel'.

4.3 New Archive Event

As any database gets larger, it inevitably slows down. To keep databases smaller, especially with higher volume databases, archive events were added to EDL. Archive events optionally move data from an active database to a separate history database based on the age of the data (i.e. last 12 months). This way, the active database stays smaller and manageable and the history database is allowed to grow. Since the Enterprise Report Scheduler can easily switch between multiple database connections, users can run reports in either database by simply switching connections.

During the creation of the event, there is an option to either copy the data to a secondary history database or simply delete it. Regardless of the choice, data that matches the rules will be permanently deleted from the active database. All archival (copying and deleting or just deleting) activity is done in transactions which only commit the changes if all parts are successful. If the commit of the changes is unsuccessful all data is rolled back to its original form. Therefore, there should never be any loss of data unless it is the intention of the user.

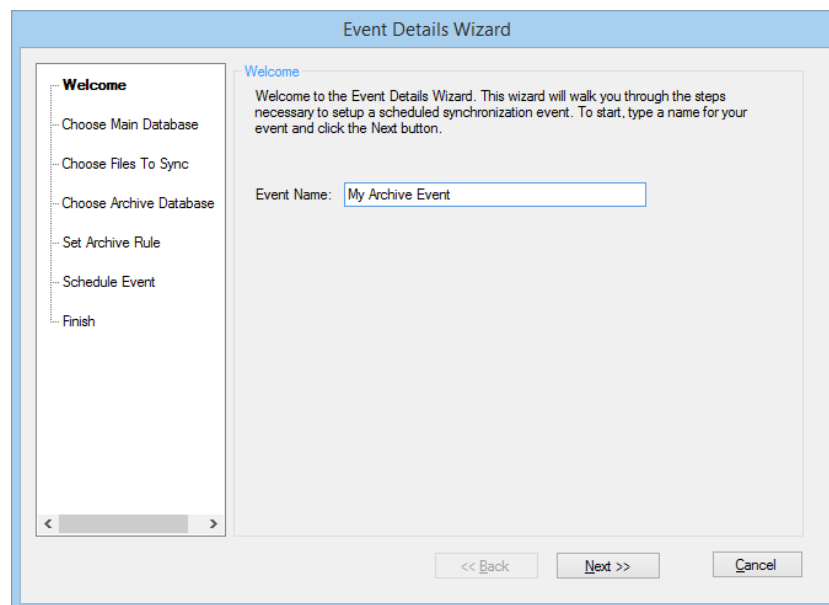
A point to note is that the latest record in any part file is never archived/deleted even if its age qualifies it for such. This is done by design since EDL load events look at the latest record to determine how many new records to add. If all records were archived, load and archive events could get locked in an endless loop of loading and archiving the same records.

Multiple archive events can be run on the same database with different parameters. For instance, higher volume part files can be added to an event that only keeps the last 2 months of data while the normal archive event keeps 12 months of data.

Archive events follow the same record governor (available via **Tools - Options - Performance** on page [31](#)) as load events to keep performance and memory under control.

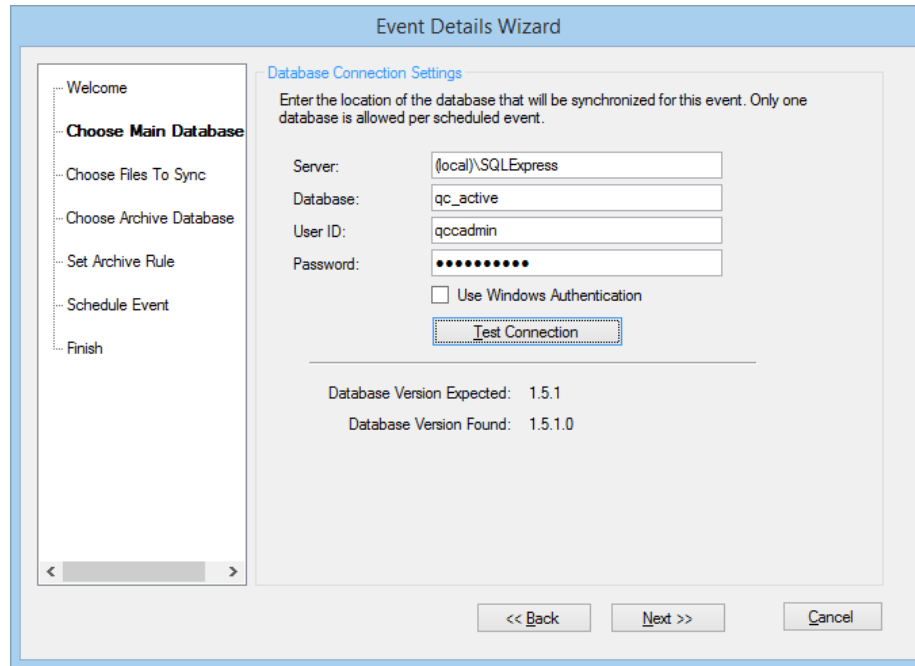
4.3.1 Welcome Screen

Just like the other event types, name the event here.



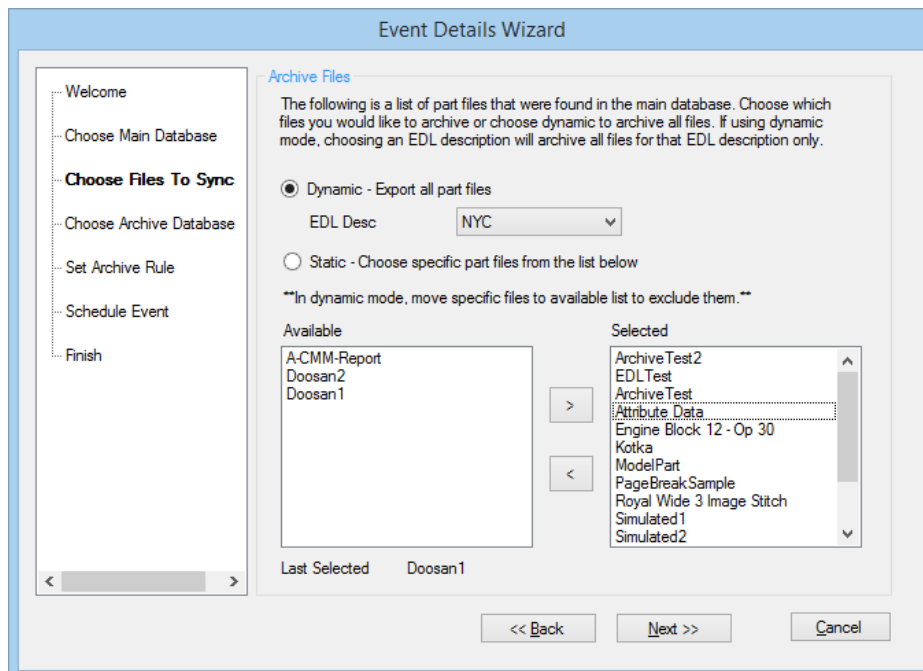
4.3.2 Choose Main Database

Choose the main or active database from which data will be archived or deleted. As with all database connection screens, type the name of the server, database, user, and password (or check to **Use Windows Authentication**).



4.3.3 Choose Files to Sync

Once the main database is known, choose which part files which will be included in this archive event. This can be done either dynamically or via a static list of files. Within the dynamic option, choosing an EDL description will archive all files with that description. In addition, moving specific files to the **Available** list while in dynamic mode will exclude them from the archive event.



4.3.4 Choose Archive Database

Choose to either copy the data to a secondary database or delete the data with the **Archive Option**. Assuming a history database is being used, add the connection information for the history (archive) database below.

The screenshot shows the 'Event Details Wizard' window. On the left is a navigation pane with steps: Welcome, Choose Main Database, Choose Files To Sync, **Choose Archive Database**, Set Archive Rule, Schedule Event, and Finish. The main area is titled 'Archive Database Connection Settings'. It features a dropdown menu for 'Archive Option' with three options: 'Copy data to archive database and delete from main data', 'Copy data to archive database and delete from main database', and 'Delete from main database ONLY'. Below this is a text prompt: 'Enter the location of the database that will be used as an archive for this event.' There are input fields for 'Server' (containing '(local)\sqlexpress'), 'Database' (containing 'qc_archive'), 'User ID' (containing 'qccadmin'), and 'Password' (masked with dots). A checkbox for 'Use Windows Authentication' is unchecked. A 'Test Connection' button is present. At the bottom, it shows 'Database Version Expected: 1.5.1' and 'Database Version Found: 1.5.1.0'. Navigation buttons '<< Back', 'Next >>', and 'Cancel' are at the bottom right.

4.3.5 Set Archive Rule

The **Archive Rule** determines which records will be kept. Any records with an age greater than the rule are archived/deleted. The Archive Rule can be determined 3 ways; age by month, age by records, or age by special date (i.e. this month). Choosing a special date will make the size of the database grown and shrink depending on the current date within the cycle. For instance, if “Last Month” is chosen and this is the first day of February, only January’s data will be kept. However, if the current date is the last day of February, both January’s and February’s data will be kept.

The screenshot shows the 'Event Details Wizard' window. The navigation pane on the left has 'Set Archive Rule' highlighted. The main area is titled 'Archive Rule' and contains the instruction 'Choose how you would like to archive the part files below.' There are three radio button options: 'Archive records older than the number of months below' (selected), 'Archive records older than the number of records below', and 'Archive records before the date chosen below'. The first option has a 'Months' input field with the value '12'. The second option has a 'Records' input field. The third option has a 'Before' dropdown menu with 'Today' selected. Navigation buttons '<< Back', 'Next >>', and 'Cancel' are at the bottom right.

4.3.6 Schedule Event

Like the other event types, the event scheduler sets the frequency the archive event will run. For archive events, it is recommended to run it once a day or less rather than every 30 seconds.

The screenshot shows the 'Event Details Wizard' window. On the left is a vertical navigation pane with steps: Welcome, Choose Main Database, Choose Files To Sync, Choose Archive Database, Set Archive Rule, **Schedule Event**, and Finish. The main area is titled 'Recurrence Pattern' and contains the text: 'Choose how often this particular event will run. You can choose from several combinations of hourly, daily, weekly or monthly schedules.' Below this text are five radio button options: Hourly, Daily (selected), Weekly, Monthly, and Yearly. To the right of these options are input fields: 'Every' followed by a text box containing '1', 'day(s) at' followed by a text box containing '11:00pm'. At the bottom of the window are three buttons: '<< Back', 'Next >>', and 'Cancel'.

4.3.7 Finish

Like other event types, Finish allows you to enable the event before saving it.

The screenshot shows the 'Event Details Wizard' window at the 'Finish' step. The left navigation pane has 'Finish' highlighted. The main area is titled 'Finish' and contains the text: 'Congratulations! You have successfully created a scheduled export event. Click Finish to save.' Below this text is a checkbox labeled 'Enable Event'. At the bottom of the window are three buttons: '<< Back', 'Finish', and 'Cancel'.

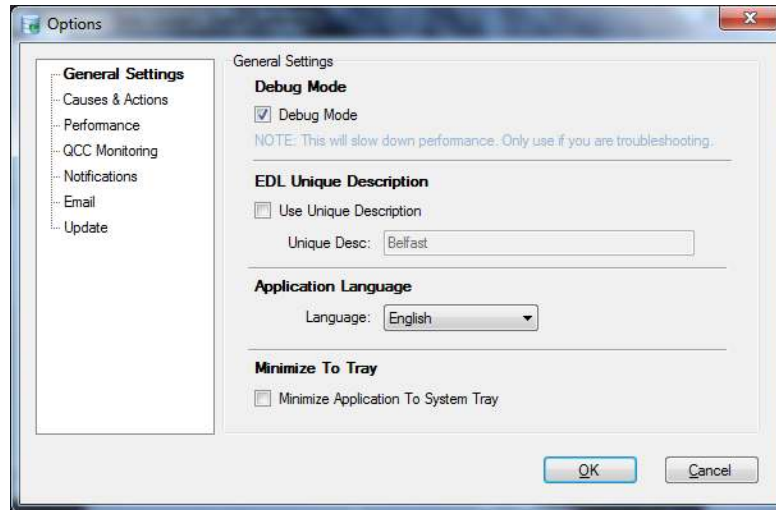
4.4 Exit

This operation will close the EDL Client. It will not affect the EDL Load Service which is controlled via the Tools menu of the EDL Client. The load service will continue to run without the EDL screen being visible.

5. The Tools – Options Menu

5.1 General Settings

The **General Settings** screen has all the general non-categorized settings listed here:



5.1.1 Debug Mode

If checked, a file is generated called debug.log in the directory where EDL is installed. This log contains detailed information on each activity done by EDL. Generally, you should keep this setting unchecked for performance reasons. However, it may help with troubleshooting if you are having troubles in the database and want to see the literal SQL being transmitted. **WARNING:** This file will get extremely large if this option is checked for long periods of time since EDL is literally logging everything it is doing.

5.1.2 EDL Unique Description

This option is a convenient description added at the QCC file level in the database. It helps to identify where the QCC file originates in cases where multiple plants are using the same inspection routines (and therefore have identical QCC file names). This should be a human readable description that positively identifies the QCC file source. This is an optional field and can be added at any time without harm. If added or changed, QCC files that are processed will automatically be updated in the system to include the new description.

5.1.3 Application Language

This identifies the current language being used by EDL for all menus, messages, labels, buttons, etc. in the application. **NOTE:** Debug and Error messages are always in English to help with troubleshooting by ProLink.

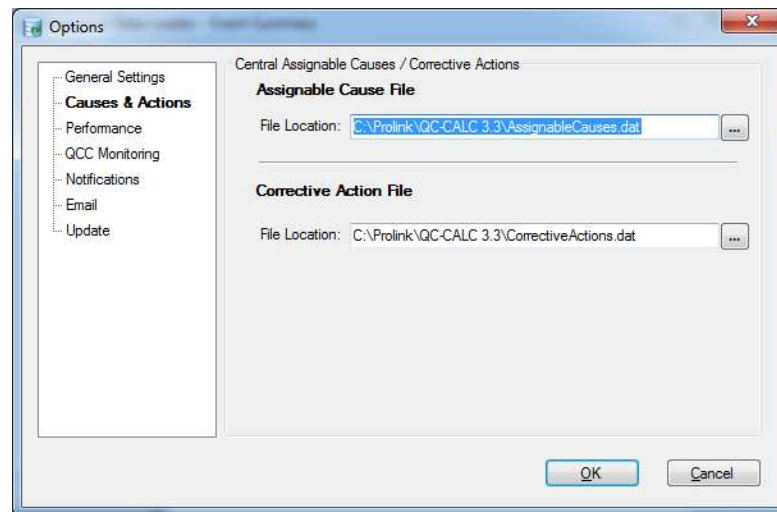
5.1.4 Minimize To Tray

This option changes the behavior of the Minimize button in the top right corner of the window. If checked, the EDL client will minimize itself to the tray of your desktop rather than just the task bar. When minimized to the tray, you can restore it by either right clicking the icon and choosing **Restore** or double clicking the icon.

5.2 Causes & Actions

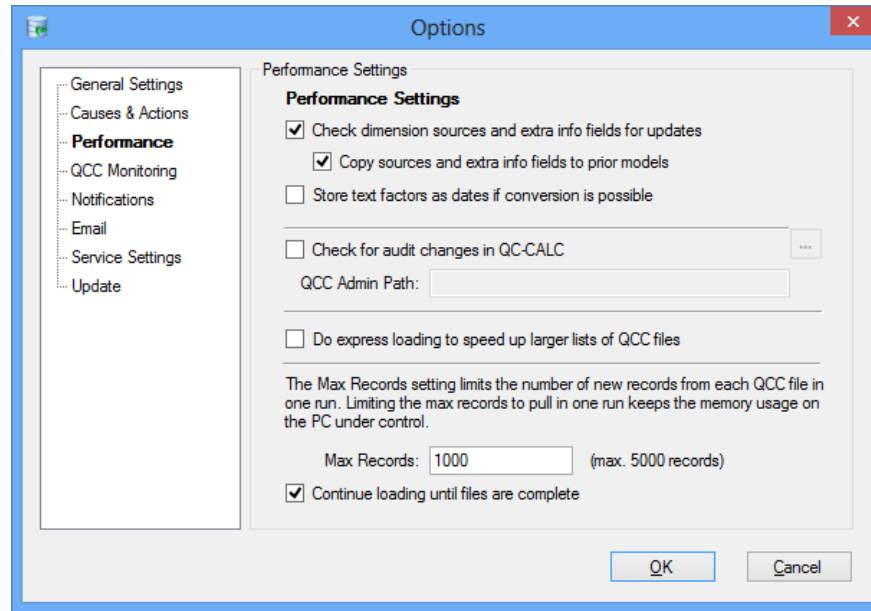
This screen allows you to set central **Assignable Cause** and **Corrective Action** files. These two settings are critical to the successful operation of EDL since QC-CALC does not store the actual descriptions of causes and actions in the QCC file. Instead there is a numeric reference to the entries stored in the files. This means when causes and actions are used in a QCC file, EDL needs to look them up in the corresponding central cause and action files to make sure the correct description is inserted in the database.

To set the file locations, simply browse to the files on your network. As a convenience features, when you select the assignable causes file, the corrective actions file will automatically fill itself in if the corrective actions file exists in the same folder. Otherwise, choose it manually.



5.3 Performance

There are certain options you can choose to toggle on/off that affect the amount of data stored in SQL Server and the performance of EDL. As expected, the more data is checked/stored, the slower the performance will be. As such certain non-critical types of data have been optioned out so you can decide to track them if you use them. If not, it will save time during the load to SQL Server. The available options are described below.



5.3.1 Check dimension sources for updates

Within each dimension in QC-CALC are two extra fields (Dimension Info and Dimension Source) that can be used for any purpose and serve to describe dimensions similar to the way a factor describes a part. The fields (accessed via **Edit – Nominals and Tolerances** in QC-CALC SPC) can be used to store anything desired since they are standard text fields. Some customers choose to add critical identifiers to critical dimensions to these fields. If you use these fields, check this box and they will be continuously updated in the SQL Server during the load. If you do not use these extra fields, leave this field unchecked. You can check/uncheck this at any time without causing harm.

NOTE: If you need to store critical dimension numbers or other descriptors (such as Key Product Characteristic numbers or other identifiers), use the Dimension Info field. This field will populate the extra_info field in the dimension table. If you need to store the machines that actually produced the dimension, use the Dimension Source field as this maps a one-to-many relationship to the dim_source table.

Copy sources and extra info field for prior models

This sub option causes EDL to copy the values found to prior models within the same part. As a review, models are snapshots of a QCC file in time and allow you to preserve historical changes to the QCC file. For instance, when a dimension is added or a tolerance is changed, a new model is created. Often times, QCC files will have multiple models in the SQL Server database. This checkbox tells EDL to copy any changes found to prior models in history whenever the values change.

5.3.2 Store text factors as dates if conversion is possible

This option allows you to save text factor values as dates if they are considered to be valid dates. For instance, if this option is enabled and the value of a text factor is "1/5/2013", then EDL will recognize it as a date and will store it both as a text factor and a date. Your text factor will still be preserved. This is an extra feature. Once it has been stored as a date, it will be available for filtering as a date in the ERS record filtering screen. This handy feature allows you to track more than just the measure date (like QC-CALC). You can actually send up to 30 different dates using the 30 text factors available to you in the QCC file. Please note that the name of the text factor has no bearing on whether or not the value is stored as a date.

5.3.3 Check for audit changes in QC-CALC

If your company is 21 CFR Part 11 compliant (subject to FDA regulations) then you may be required to track electronic document signatures. Normally QC-CALC saves all audit information along with the QCC file since the audit change is one that has happened to a part or measurement (such as changing a value, deleting a measurement, assigning a cause, etc). However, there are certain Part 11 related activities that occur outside the individual file (such as document signing of a particular report). These are stored in a separate audit file in the administrative tools area and can be loaded to the SQL Server database.

If you are subject to FDA regulations (including food producers, medical parts manufacturers, etc), then you should check this feature, and browse to the main administrative folder for the Administrative Tools in QC-CALC. This will be on your server and is different for every plant. EDL will find the audit file and synchronize it with the SQL Server database in the audit_history table.

5.3.4 Do Express Loading to speed up larger lists of QCC files

Normally, EDL goes through the list of QCC files one by one, opens each one, checks for new records, updates the EDL description, and updates changes to the dimension sources and dimension information fields. If you have a lot of QCC files (500+) this can take some time to perform especially if there are no changes. To help with performance for large numbers of QCC files, we added the Express Load option. When checked, EDL still goes through the list of QCC files, but only checks for new records. The rest of the checking and updates only occur if the QCC file has new records to be loaded. Otherwise, most of the above is skipped. This can bring the loading of large lists of files down substantially, but should only be used if the QCC files are not changing very often.

5.3.5 Max Records

The **Max Records** setting limits the number of records returned from the database. Each QCC file in the database may contain years' worth of data so set the Max Records to reduce the amount of data returned which improves performance and memory usage. This will also result in the event running faster since EDL doesn't upload every file in its entirety. The trade-off is that it may take several runs before EDL catches up to present data if the QCC files are very large.

5.3.6 Continue Loading Until Files are Complete

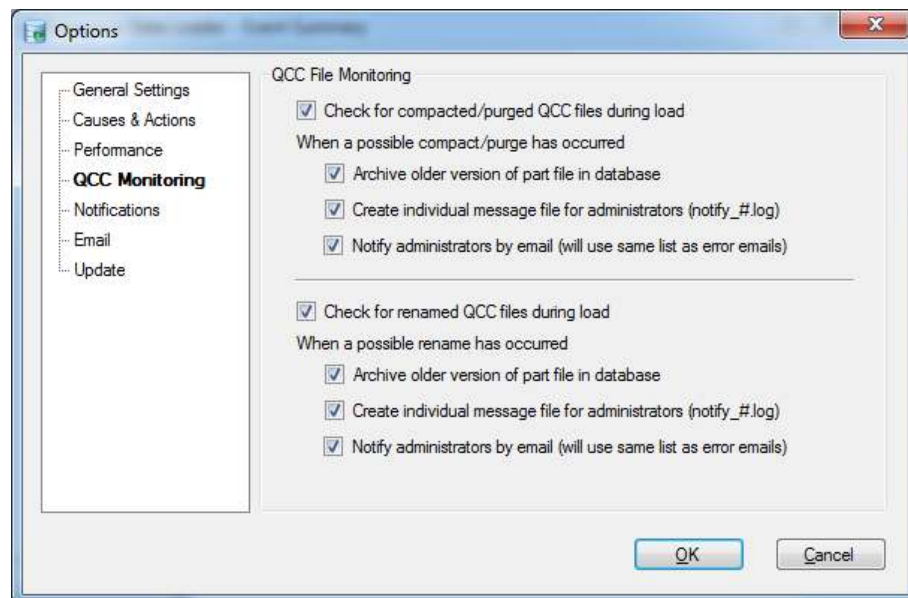
This checkbox is related to the Max Records setting. The Max Records setting is still observed to keep PC memory consumption low, but it continues grabbing chunks of data from the QCC file (in increments according to the Max Records) until the QCC files are all completely loaded. This has the added benefit of

continuing to load the QCC files until complete, but will cause the event to run longer.

5.4 QCC Monitoring

This screen allows you to monitor for changes to QCC files that could affect how EDL loads the files into the SQL database.

First a little information about how EDL identifies QCC files in the system. There are three identifiers to a QCC file; the file name itself, the creation date, and the EDL description you fill out in EDL (or at the event level). When going through the list of QCC files in an event, EDL must decide whether or not each QCC file already exists in the database. To do this, it queries the database using the file name and creation date of the file. If no combination of file name/creation date is found (to the second in time), it adds a new part file in the database and all the data. If found in the database, it looks at how many records have been added since the last load, loads them, and so forth. The assumption is that the likelihood of two QCC files being created at the same moment in time with the exact same name is extremely rare and constitutes a unique identifier.



There are a couple of user actions that can change the make-up of a QCC file to the point where you can fool EDL and duplicate data in the database. These are as follows:

1. You rename the QCC file. In this case, when EDL looks for the file name/date combo, it won't find it because you changed the name. As such, it will create new database entry again, from scratch, with duplicate data.
2. You compact or purge the QCC file. In these cases, QC-CALC resets the creation date inside the file. It does this since you are physically removing records from the QCC file that potentially still exist in the SQL database. To avoid causing problems with record counts and moving records, QC-CALC resets the date. Again, knowing about the uniqueness of the file name/date combo, EDL will create new files.

3. You delete and recreate the QCC file. This happens when you physically delete the QCC file either through QC-CALC (File > Delete) or through Windows Explorer.

NOTE: It is recommended that you NEVER compact or purge QCC files! There is no harm in leaving deleted records present in the file and therefore in the SQL database.

NOTE: The creation date is found inside the QCC file and is set by QC-CALC. The creation date is not the date assigned/visible in Windows.

QCC Monitoring allows you to specify what EDL should do in the above situations to keep your SQL database clear and concise. EDL can automatically monitor QCC files for both of the scenarios discussed above and then notify and/or fix the situation in the database.

5.4.1 Check for compacted/purged QCC files during load

By checking the first checkbox, you instruct EDL to look for QCC files that have been compacted and/or purged. It does this by running a query on the SQL database for all QCC files with the same EDL Description that have the same file name, but a different creation date. If the name is found with a different creation date, the assumption is that the file has been compacted. This is, of course, is not guaranteed which is why this is an option you can choose to turn on/off. It is also done at the EDL Description level which is typically at the plant level. Once found, you can choose to do any of the following three options:

Archive older version of part file in database

This sets the archive flag on the existing part file in the database and then adds the new one. Setting the archive flag effectively makes the older file disappear from ERS although its data is still there. It can be brought back to un-archived (normal) via the administration area in ERS.

Create individual message file for administrators

This puts the same error message that is emailed into a text file. This is for corporations that have event monitoring systems. The folder path of the file is taken from the **Create individual error files for each error** option in the **Notifications** area. However, the file name is notify_#.log where # is a number starting from 1. This number counts until the file name does not exist.

Notify administrators by email

This sends an email to admins telling them that a compacted/purged QCC file was found. Based on the **Archive older version of part file in database** option, the message sent states that it either archived the older file or that archiving is recommended to avoid duplication. The list of admins for this email is taken from the **Email Administrators On Error** email list in the **Notifications** area.

NOTE: When compacted QCC files are found, all edits that occurred since the prior load are discarded. This is because the older QCC file is typically archived and the new file is loaded to the SQL database from scratch. As such, all the edits are already a part of the new file so no further edits are necessary.

5.4.2 Check for renamed QCC files during load

By checking the checkbox, you instruct EDL to look for QCC files that have been renamed. It does this by running a query on the SQL database for all QCC files with the same EDL Description that have the creation date, but a different file name. If the date is found with a different name, the assumption is that the file

has been renamed. This is, of course, is not guaranteed. Once found, you can choose to do any of the following three options:

Archive older version of part file in database

This sets the archive flag on the existing part file in the database and then adds the new one. Setting the archive flag effectively makes the older file disappear from ERS although the data is still there. It can be brought back to un-archived (normal) via the administration area in ERS.

Create individual message file for administrators

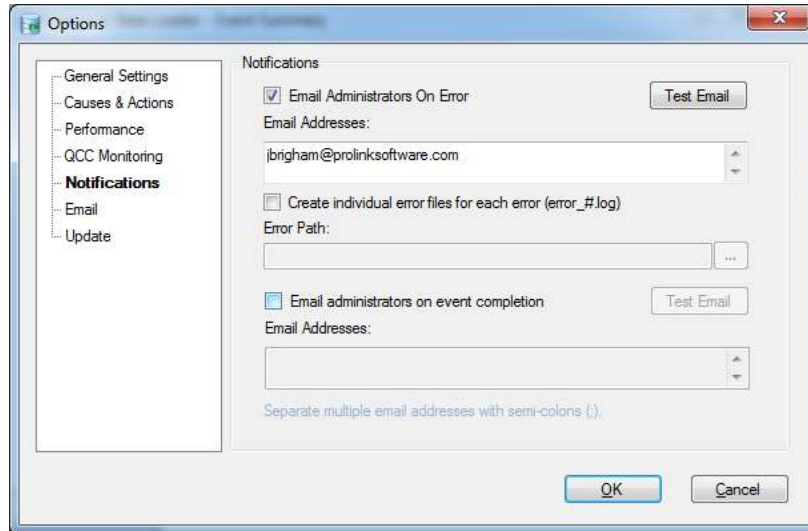
This puts the same error message that is emailed into a text file. This is for corporations that have event monitoring systems. The folder path of the file is taken from the **Create individual error files for each error** option in the **Notifications** area. However, the file name is notify_#.log where # is a number starting from 1. This number counts until the file name does not exist.

Notify administrators by email

This sends an email to admins telling them that a renamed QCC file was found. Based on the **Archive older version of part file in database** option, the message sent states that it either archived the older file or that archiving is recommended to avoid duplication. The list of admins for this email is taken from the **Email Administrators On Error** email list in the **Notifications** area.

5.5 Notifications

Since EDL has been designed to run unattended, there are no messages that pop up on the screen when it is running on the timer. Therefore, in order to know that an error or event has occurred, you may want to have EDL email so you can take some action. This screen allows you to do just that.



5.5.1 Email Administrators On Error

To enable the functionality simply check the **Email Administrators On Error** checkbox and enter as many email addresses (or distribution lists) as desired separated by semi-colons (;).

If you would like to generate a test email to make sure the email settings are correct, simply click the **Test Email** button. You should receive an email within a minute or two confirming the email settings are valid.

NOTE: If you have not set up the email settings (next section) you may not receive emails just yet. Make sure the SMTP settings are set and saved before testing the email. This means setting the SMTP settings, clicking OK to save the settings, and then going back into **Tools – Options** to test the email.

5.5.2 Create individual error files for each error

This option allows you to create an individual log file for each error encountered. This is useful for corporations who use scheduling software that monitors folders for file-based error reporting. The file name is error_#.log where # is a number starting from 1. This number counts until the file name does not exist and the new file is named that way.

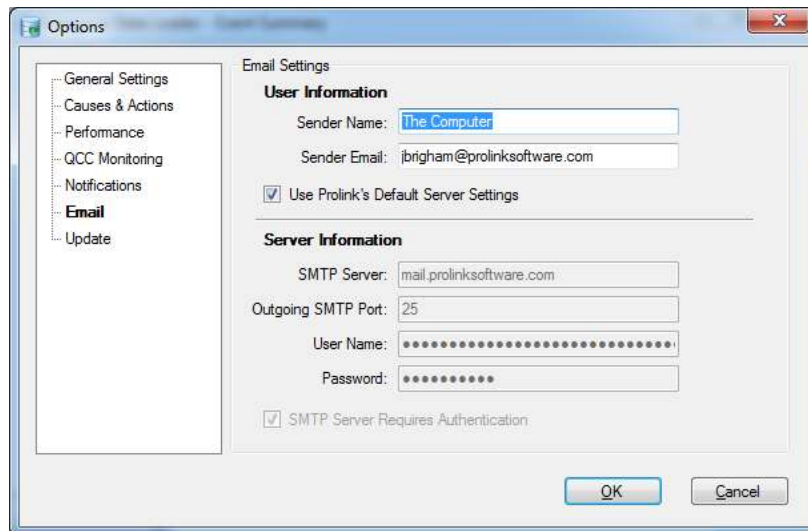
The file is a standard tab delimited ASCII output file with the following fields: date, error description, source, stack trace, and extra info. The stack trace is a trace path of events that lead to the error (i.e. button clicked launched the screen, which caused a mouse event, which caused the error, etc). The extra info field is usually blank, but is used for any extraneous information that can help with troubleshooting such as the id of the record where the failure occurred. This is helpful for internal Prolink personnel.

5.5.3 Email administrator on event completion

When a scheduled event is completed successfully an email will be sent to the list of email addresses in the area below the option. This is recommended if you plan to run EDL events daily and want to have someone checking for a successful completion each morning.

5.6 Email

This screen is used to configure your outbound SMTP server for sending email. SMTP (or Simple Mail Transfer Protocol) is required in order for any application to send mail via the internet. The application (in this case EDL) communicates with the SMTP server that resides on the internet and has the ability to route email. This is unrelated to and runs independent of any other email clients such as Lotus Notes or Microsoft Outlook. In fact, those email clients also must communicate with an SMTP server in order to send their mail as well.



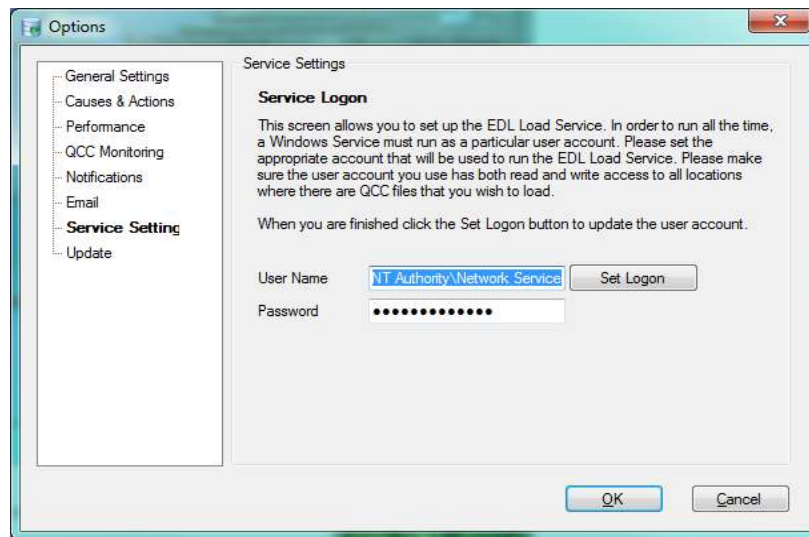
By default, EDL is set up to use Prolink's SMTP server located on the Internet. This way, the email option works out of the box. However, it is recommended that you enter your company's actual SMTP server information as it will keep the

traffic lower on Prolink's servers and increase performance (decrease the time it takes for you to receive an email from EDL). In addition, you may not be able to use Prolink's servers if your IT department blocks SMTP transmissions to servers outside your network (firewall).

Your IT department should be able to help you set up this screen.

5.7 Service Settings

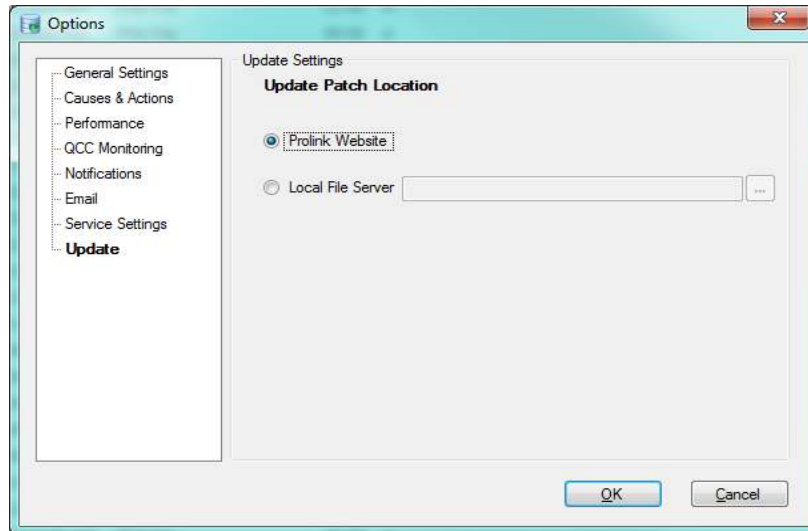
This screen allows you to set up the EDL Load Service. In order to run all the time, a Windows Service must run as a particular user account. In this screen you need to set the appropriate account that will be used to run the EDL Load Service. Make sure the user account you use has both read and write access to all locations where there are QCC files that you wish to load.



5.8 Update

This screen enables you to specify where to find new updates for EDL. If you are connected to the internet, it is recommended that you leave the default setting of the **Prolink Website**. This way, you can easily choose **Help – Run Update** without having to first download the update and place it on a local server.

However, if you wish to store the updates locally, you can choose **Local File Server** and specify a path to the update. The local server path is a folder path only. EDL will be looking for the EDLUpdate33.exe file that was downloaded manually from the Prolink website and saved to that folder when the **Help – Run Update** menu is chosen.



6. The Help Menu

6.1 Contents

This menu launches this help file.

6.2 License Configuration

This menu launches the **License Configuration** screen and is used in conjunction with the Prolink website to activate EDL. From the **My Account** area of the website, click the **Activate** link for the copy of EDL you want to unlock, then follow the directions for generating an **Activation Code**. You will need to enter the **Computer ID** shown in the **License Configuration**.

Enter the activation code generated on the Prolink website in the **Activation Code** area of the **License Configuration** screen and click the **Validate** button.

The **Kill License** button deletes the current license for EDL. If this is done by mistake another **Activation Code** for the same computer can be generated by using the **Details** link listed for this copy of EDL on the **My Account** page of the website.



If the **Computer ID** listed on the website does not match what is shown in the **Activate Software** screen you need to contact Prolink.

6.3 Run Update

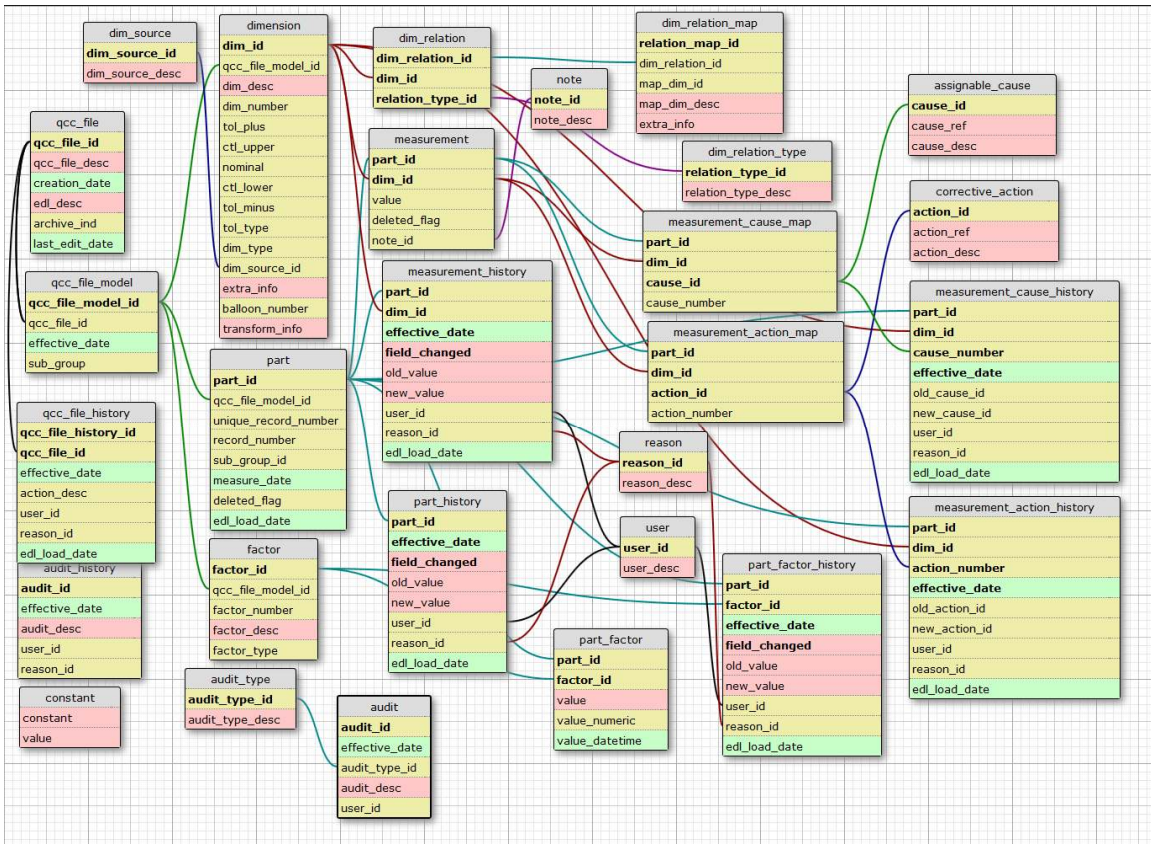
This menu option locates and runs the update patch required to update your current version of EDL, if a newer revision is available.

6.4 Gather EDL Information

If there is a problem with your software that cannot be solved by the help file or your manual, you can call Prolink for technical support. We may ask you to choose this menu option, which will automatically collect all of the information about your settings that we need to assess the problem. You will need to email the file EDL creates to us.

7. Advanced SQL Server Format

7.1 Diagram



7.2 General Description

The new format (v.1.4) involves several tables and splits both the actual and meta data into smaller chunks for easier storage and processing. Since QC-CALC is very flexible and allows the QCC database to seamlessly change over time, the database was designed to do the same.

Exact table definitions are below, but basically parts are stored in the qcc_file table at their highest level. The qcc_file table is synonymous with the QCC file itself and is meant to represent a particular type of part that is measured over time (i.e. a particular engine block). As things change with the part (i.e. tolerances change or another feature is added), the definition of the part is re-written to preserve the historical nature of the data. This new definition is defined as a qcc_file_model and is stored by effective date. For instance, the engine block we are measuring had 5 features as of 1/1/2007 and has 6 features as of 6/1/2007. These would be two different models both grouped under the same engine block in the qcc_file table. Each time anything about the part definition changes, a new model is added within the qcc_file_model table, and the definitions for dimensions and factors are redefined. All measurements are then linked to the new dims and factors of the new part type.

To explain further, a qcc_file_model is a snapshot of the definition of a particular part to be measured at a particular point in time. The model contains a unique set of dimensions and factors. When parts are actually measured, the part, measurement, and part_factor

tables are filled with actual data and are mapped back to their corresponding models, dimensions, and factors.

The assignable_cause and corrective_action tables define the entire list of assignable causes and corrective actions that can occur. The measurement_cause_map and measurement_action_map table allow specific causes and actions to literally be mapped to a measurement in a many-to-many fashion. Therefore, more than one part can use the same assignable cause and parts can have more than one assignable cause. In QC-CALC 3.x, assignable causes and corrective actions are mapped to the part and not the individual measurement. In QC-CALC 4.x, assignable causes and corrective actions will be mapped to the individual measurement. Therefore, the database is currently set up to store causes and actions at the measurement level, but actually stores them at the part level. This is temporary.

7.3 Table Definitions

qcc_file

This table represents the QCC file directly and groups all history of the QCC file under one umbrella. This allows for easier querying later across all parts of the same type. A combination of the QCC file name and the creation date on the file itself are used to positively identify QCC files from each other. This allows you to have similar QCC files from multiple plants sharing the same database. The EDL description (described above in the Options section) allows you to add a friendly name to describe the QCC file other than the file name itself. This way, if you have two QCC files with the same name from two different plants, you can tell the two apart in the database (i.e. widget (Shanghai) and widget (Los Angeles)).

- **qcc_file_id** - This is the auto-generated primary key that defines the qcc file.
- **qcc_file_desc** - The name of the QCC file without the .qcc extension.
- **creation_date** - This is the creation date from the control section of the QCC file. It helps to uniquely identify one QCC file from another if there are multiple PCs with the same QCC files.
- **edl_desc** - This is a field identifying which copy of EDL actually added the data. This can be set at the plant level if desired.
- **archive_ind** - This is a tristate that indicates whether or not the part file has been archived. The values are 0 (normal), 1 (archived) and 2 (override normal). Override normal can be set in ERS to force a file to be visible throughout the application even if it has not had activity for a long period.
- **last_edit_date** - This is a date indicating the last time an edit occurred to the part file. It is an internal indicator for EDL.

qcc_file_model

This table represents a particular historical snapshot of the part as of a particular date. Each time a part is changed in the QCC file, a new part type record is generated along with the latest definition for the part. This allows us to maintain a historical record of the data as it looked on the particular day of the export. For example, in the QCC file, if the tolerances become smaller over time, parts that were previously in spec can become out of spec if measured against the new tolerances. Keeping this table and part type definitions historical keeps the history intact.

- **qcc_file_model_id** - Auto-generated primary key
- **qcc_file_id_id** - Foreign key reference back to the qcc_file table.
- **effective_date** - The date as of which this new definition is effective.
- **sub_group** - Definition for the number of the subgroup for a particular model

dimension

This table is a definition of all dimensions (features) of a particular part type.

- **dim_id** - Auto-generated primary key
- **qcc_file_model_id** - Foreign key reference back to the model table
- **dim_desc** - The feature label
- **dim_number** - The number of the label in the Real-Time display
- **tol_plus** - The plus tolerance value (NULL if single sided lower type)
- **ctl_upper** - The upper control limit.
- **nominal** - The nominal
- **ctl_lower** - The lower control limit.
- **tol_minus** - The minus tolerance value (NULL if single sided upper type)
- **tol_type** - Defines the type of tolerance. Options are BI, SSU, SSL, NON for Bilateral, Single Sided Upper, Single Sided Lower, and Non-Toleranced respectively.
- **dim_type** - defines the source of the data (from the machine, manually entered, or calculated)
- **dim_source_id** - Foreign key reference to the dim_source table that declares the source of the dimension (i.e. which machine is literally responsible for the feature - currently only available via Zeiss Calypso)
- **extra_info** - The extra information that accompanies dimensions from QC-CALC. This is set in the Edit Nominals & Tolerances screen of QC-CALC SPC 3.0.
- **balloon_number** - This is the balloon number from the CAD drawing if supplied.
- **transform_info** - This field holds the settings for the Johnson Transform if one has been performed on the dimension. This way, we can use the settings to perform the transform consistently going forward without having to recalculate the algorithm.

dim_source

This table holds a reusable list of dimension sources. This allows you to track the source of a particular dimension. For instance, it may be a particular machine on your shop floor that produced the dimension. Since the same machine produces many dimensions, the same machine can be mapped to multiple dimensions in the dimension table.

- **dim_source_id** - An auto-generated primary key.
- **dim_source_desc** - The name of the source.

dim_relation_type

This table defines dimension relation types so ERS knows how to handle the relationship in charts, reports, etc.

- **relation_type_id** - Auto-generated primary key
- **relation_type_desc** - The name of the relation type.

dim_relation

This table sets up a particular dimension as the master dimension of a particular dimension relationship. The dimension's id is linked in this table and the child dimensions in the relationship are mapped through the dim_relation_map table.

- **dim_relation_id** - Auto-generated primary key
- **dim_id** - The id of the dimension that is the master in the relationship.
- **relation_type_id** - A foreign key mapped telling us what type of relationship this is.

dim_relation_map

This table adds child dimensions to the relationship and identifies their roles in the relationship. For instance, in the case of True Positions, the True Position dimension is the master dimension in the relationship. This will be in the dim_relation table. The other 3 dimensions in the relationship (X, Y, and Diameter) would be mapped in this table.

- **relation_map_id** - Auto-generated primary key

Advanced SQL Server Format

- **dim_relation_id** – A foreign key reference to the relationship definition including the master dimension and relationship type.
- **map_dim_id** – The id of the dimension that is one of the children in the relationship.
- **map_dim_desc** – The description of the role of the child dimension (i.e. X, Y, DiameterPin, DiameterHole)
- **extra_info** – An extra information field for future expansion.

factor

This table houses the definitions of the factors for a given part type.

- **factor_id** - Auto-generated primary key
- **qcc_file_model_id** - Foreign key reference back to the model table
- **factor_number** - The physical number of the factor to keep them in order.
- **factor_desc** - The description of the factor.
- **factor_type** - The type of factor (text vs. numeric)

part

This table represents an actual part measured in QC-CALC. It is the top level for the part and includes the measurement date, a sub group identifier, and the record number from the QCC database.

- **part_id** - Auto-generated primary key
- **qcc_file_model_id** - Foreign key reference back to the model table
- **unique_record_number** - A unique record number from the QCC file. This is unique even if the QCC database is circular.
- **record_number** - The current record number in the QCC database. This number is not guaranteed unique if the file is set up to be circular. For a guaranteed unique number, use unique_record_number.
- **sub_group_id** - The auto-incrementing sub group number that allows part grouping at the subgroup level. This is a convenience field to allow aggregate rollups since the measurement data is stored at the lowest level rather than at the “point” level in QC-CALC.
- **measure_date** - The date and time the part was measured.
- **deleted_flag** - Indicates whether or not the entire part was deleted.
- **edl_load_date** – Indicates when the record was added to the database. Used for internal tracking purposes and the EDL export events.

measurement

This table houses the actual values that were measured. The values are stored based on the part and dimension measured.

- **part_id** - Foreign key reference back to the part
- **dim_id** - Foreign key reference back to the dimension
- **value** - The actual value measured.
- **deleted_flag** - Indicates whether or not the point was deleted.
- **note_id** – Foreign key reference to a note for the measurement (notes are re-used).

note

This table houses all notes in a single place to avoid repeating the note in the measurement table. Therefore, if several points or entire subgroups of parts have the same note, the note will be added to this table once and then linked to all the appropriate measurements in the measurement table.

- **note_id** – Primary key identifier
- **note_desc** – The notes themselves.

measurement_history

This table houses the historical values for CFR21 Part 11 auditing reasons. Whenever a field is changed, the latest value is updated in the measurement table. The audit trail of the edit is stored in this table.

- **part_id** - Foreign key reference back to the part
- **dim_id** - Foreign key reference back to the dimension
- **effective_date** - the effective date of the edit
- **field_changed** - the name of the field that changed.
- **old_value** - The value before the change
- **new_value** - The value after the change
- **user_id** - The user who made the change. Foreign key reference to user table.
- **reason_id** - The reason code for the change. Foreign key reference to reason table.
- **edl_load_date** – Indicates when the record was added to the database. Used for internal tracking purposes and the EDL export events.

part_history

This table houses the historical values for CFR21 Part 11 auditing reasons. Whenever a field is changed, the latest value is updated in the part table. The audit trail of the edit is stored in this table.

- **part_id** - Foreign key reference back to the part
- **effective_date** - the effective date of the edit
- **field_changed** - the name of the field that changed.
- **old_value** - The value before the change
- **new_value** - The value after the change
- **user_id** - The user who made the change. Foreign key reference to user table.
- **reason_id** - The reason code for the change. Foreign key reference to reason table.
- **edl_load_date** – Indicates when the record was added to the database. Used for internal tracking purposes and the EDL export events.

user

This table houses the users of the system. Users are added to this table as they are used in QC-CALC - not when they are created.

- **user_id** - Auto-generated primary key
- **user_desc** - The description of the user as passed from QC-CALC

reason

This table houses the reason codes in the system. Reason codes are added as they are used in QC-CALC - not when they are created.

- **reason_id** - Auto-generated primary key
- **reason_desc** - The description of the reason as passed from QC-CALC.

part_factor

This table houses the actual value of the factors of a particular part. The values are stored according to the part and factor being measured.

- **part_id** - Foreign key reference back to the part
- **factor_id** - Foreign key reference back to the factor
- **value** - The actual value of the factor. This is stored as a 500 character field even though numeric values could also be stored here. For numeric values, convert the value to a numeric value. If the value is not set in QC-CALC, a NULL will be inserted.

assignable_cause

This table houses the complete list of assignable causes for all parts. This list is currently populated as the causes are used and not when they are literally created in the QCC file. This keeps the database more efficient.

- **cause_id** - Auto-generated primary key
- **cause_ref** - The short description or reference
- **cause_desc** - The description of the cause. 500 character field.

corrective_action

This table houses the complete list of corrective actions for all parts. This list is currently populated as the actions are used and not when they are literally created in the QCC file. This keeps the database more efficient.

- **action_id** - Auto-generated primary key
- **action_ref** - The short description or reference
- **action_desc** - The description of the action. 500 character field.

measurement_cause_map

This table allows the assignment of any number of assignable causes to a measurement. QC-CALC 3.0 saves assignable causes at the part level. This means that assignable causes will be mapped to each measurement in the part. In 4.0, assignable causes will be mapped at the measurement level.

- **part_id** - Foreign key reference back to the measurement
- **dim_id** - Foreign key reference back to the measurement
- **cause_id** - Foreign key reference back to the assignable_cause
- **cause_number** - The number of the cause since there can be multiple. Also matches to the action.

measurement_action_map

This table allows the assignment of any number of corrective actions to a measurement. QC-CALC 3.0 saves corrective actions at the part level. This means that corrective actions will be mapped to each measurement in the part. In 4.0, assignable causes will be mapped at the measurement level.

- **part_id** - Foreign key reference back to the measurement
- **dim_id** - Foreign key reference back to the measurement
- **action_id** - Foreign key reference back to the corrective_action
- **action_number** - The number of the action since there can be multiple. Also matches to the cause.

part_factor_history

This table holds historical factor values for each part. As changes occur to the factor values, a record of the change is automatically added to this table. If Part 11 mode is enabled in QC-CALC, the user_id and reason_id fields will be populated. Otherwise, these fields will be set to NULL.

- **part_id** - Foreign key reference back to the part
- **factor_id** - Foreign key reference back to the factor
- **field_changed** - The field that changed.
- **effective_date** - The effective date/time of the change.
- **old_value** - The value before the change.
- **new_value** - The value after the change.
- **user_id** - The user who made the change (Part 11 Mode only)
- **reason_id** - The reason the change was made (Part 11 Mode only)

- **edl_load_date** – Indicates when the record was added to the database. Used for internal tracking purposes and the EDL export events.

measurement_cause_history

This table holds the history changes to the measurement_cause_map table. As causes are added and removed, a record of the change is added to this table. This includes setting and removing assignable causes to/from the part and not just changes. In cases like this, the old or new value will be set to NULL.

- **part_id** - Foreign key reference back to the part
- **dim_id** - Foreign key reference back to the dimension
- **cause_number** - The number of the cause since there can be multiple.
- **effective_date** - The effective date/time of the change.
- **old_cause_id** - The cause_id before the change.
- **new_cause_id** - The cause_id after the change.
- **user_id** - The user who made the change (Part 11 Mode only)
- **reason_id** - The reason the change was made (Part 11 Mode only)
- **edl_load_date** – Indicates when the record was added to the database. Used for internal tracking purposes and the EDL export events.

measurement_action_history

This table holds the history changes to the measurement_action_map table. As actions are added and removed, a record of the change is added to this table. This includes setting and removing corrective actions to/from the part and not just changes. In cases like this, the old or new value will be set to NULL.

- **part_id** - Foreign key reference back to the part
- **dim_id** - Foreign key reference back to the dimension
- **action_number** - The number of the cause since there can be multiple.
- **effective_date** - The effective date/time of the change.
- **old_action_id** - The action_id before the change.
- **new_action_id** - The action_id after the change.
- **user_id** - The user who made the change (Part 11 Mode only)
- **reason_id** - The reason the change was made (Part 11 Mode only)
- **edl_load_date** – Indicates when the record was added to the database. Used for internal tracking purposes and the EDL export events.

constant

This table holds any constants that are needed in the system. Currently this includes the database version number.

- **constant** – A label describing the constant.
- **value** – The value of the constant.

audit_history

This table holds any 21 CFR Part 11 actions that have occurred. These include the creation of new records, the signing of reports. They are not changes to individual parts so they are placed in a more generic table.

- **audit_id** – An auto-generated primary key.
- **effective_date** – The date and time the action took place.
- **audit_desc** – The description of the action that took place.
- **user_id** – The user who performed the action.
- **reason_id** – The reason for the action.

audit_type

This table is related to the audit table (not audit_history) and tracks the type of activities that happen in the database. For instance, when EDL loads data an audit record is inserted to the audit table with an audit type of EDL Load.

- **audit_type_id** - An auto-generated primary key.
- **audit_type_desc** – A description of the type.

audit

This table tracks activities that happen in the database. For instance, when EDL loads data an audit record is inserted to this table.

- **audit_id** - An auto-generated primary key.
- **effective_date** – The date of the audit event
- **audit_type_id** – The type of event that occurred.
- **audit_desc** – The description of the event (records added, edited, etc)
- **user_id** – The user id performing the action.

7.4 Example Queries

These queries are meant as a guide for you to use to prove that the data is accurate. They are only samples.

Example 1: Retrieving The Latest Part Definition For A QCC File	
Description:	This query gets the latest definition for a given QCC file.
Parameters:	Pass the name of the QCC file in the quotes at the end without the .qcc extension. In this example, "sample" is passed.
Query:	<pre> SELECT qf.qcc_file_id, qf.qcc_file_desc, qfm.qcc_file_model_id, qfm.effective_date, qfm.sub_group FROM qcc_file qf INNER JOIN qcc_file_model qfm ON qf.qcc_file_id = qfm.qcc_file_id WHERE qfm.effective_date = (SELECT MAX(qfm.effective_date) FROM qcc_file_model qfm INNER JOIN qcc_file qf2 ON qfm.qcc_file_id=qf2.qcc_file_id WHERE qf2.qcc_file_desc=qf.qcc_file_desc) AND qf.qcc_file_desc = 'sample' </pre>
Notes:	Leaving out the last "and" clause will get the latest part type for all groups.

Example 2: Retrieving The Latest Part Dimensions For A QCC File	
Description:	This query gets the latest set of dimensions for a QCC file.
Parameters:	Pass the name of the QCC file in the quotes at the end without the .qcc extension. In this example, "sample" is passed.
Query:	<pre> SELECT d.dim_id, </pre>

	<pre> d.dim_desc, d.dim_number, d.tol_plus, d.ctl_upper, d.nominal, d.ctl_lower, d.tol_minus, d.tol_type, d.dim_type FROM qcc_file qf INNER JOIN qcc_file_model qfm ON qf.qcc_file_id = qfm.qcc_file_id INNER JOIN dimension d ON qfm.qcc_file_model_id = d.qcc_file_model_id WHERE qfm.effective_date = (SELECT MAX(qfm.effective_date) FROM qcc_file_model qfm INNER JOIN qcc_file qf2 ON qfm.qcc_file_id = qf2.qcc_file_id WHERE qf2.qcc_file_desc=qf.qcc_file_desc) AND qf.qcc_file_desc = 'sample' ORDER BY dim_number </pre>
--	--

Notes: Leaving out the last “and” clause will get the latest dimensions for all groups.

Example 3: Retrieving The Measured Values For A Part

Description: This query gets the measurements for part given the record number in QC-CALC

Parameters: Pass the record number in QC-CALC as the last number. (i.e. 26)

Query:	<pre> SELECT p.part_id, p.record_number, d.dim_id, d.dim_desc, d.dim_number, d.tol_plus, d.ctl_upper, d.nominal, d.ctl_lower, d.tol_minus, d.tol_type, d.dim_type, m.value FROM qcc_file_model qfm INNER JOIN part p ON qfm.qcc_file_model_id = p.qcc_file_model_id INNER JOIN dimension d ON qfm.qcc_file_model_id = d.qcc_file_model_id INNER JOIN measurement m ON p.part_id = m.part_id AND d.dim_id = m.dim_id WHERE p.record_number = 26 </pre>
---------------	---

Notes: If you add an INNER JOIN to the qcc_file table to the FROM clause:

```

INNER JOIN qcc_file qf
ON qf.qcc_file_id = qfm.qcc_file_id

```

Advanced SQL Server Format

	<p>And add the following section to the WHERE clause:</p> <pre> AND qfm.effective_date = (SELECT MAX(qfm.effective_date) FROM qcc_file_model qfm INNER JOIN qcc_file qf2 ON qfm.qcc_file_id = qf2.qcc_file_id WHERE qf2.qcc_file_desc = qf.qcc_file_desc) AND qf.qcc_file_desc = 'sample' </pre> <p>You will get the measurements for the sample.qcc file for record number 26.</p>
--	---

Example 4: Retrieving The Measured Values Using A Subgroup Average	
Description:	This query gets the measurements for a set of parts and averages them by sub group for a particular dimension.
Parameters:	Pass the qcc file name (widget) and the dimension description (x hole position).
Query:	<pre> SELECT ROUND(AVG(m.value), 4) FROM qcc_file_model qfm INNER JOIN qcc_file qf ON qf.qcc_file_id = qfm.qcc_file_id INNER JOIN part p ON qfm.qcc_file_model_id = p.qcc_file_model_id INNER JOIN dimension d ON qfm.qcc_file_model_id = d.qcc_file_model_id INNER JOIN measurement m ON p.part_id = m.part_id AND d.dim_id = m.dim_id WHERE qf.qcc_file_desc = 'widget' AND d.dim_desc = 'x hole position' AND qfm.effective_date = (SELECT MAX(qfm2.effective_date) FROM qcc_file_model qfm2 INNER JOIN qcc_file qf2 ON qfm2.qcc_file_id = qf2.qcc_file_id WHERE qf2.qcc_file_desc = qf.qcc_file_desc) GROUP BY p.sub_group_id </pre>

Example 5: Retrieving The Measured Values Across Models	
Description:	This query retrieves measurements over time of a given dimension and qcc file regardless of the qcc_file_model. In other words, if the structure of the part changes (i.e. tolerance changes or an additional feature is added) over time, you can still retrieve values across the qcc_file_models using LEFT JOINS.
Parameters:	Pass the name of the part and the dimension label.
Query:	<pre> SELECT qf.qcc_file_desc, p.measure_date, p.record_number, d.dim_desc, m.value FROM qcc_file qf LEFT JOIN qcc_file_model qfm ON qf.qcc_file_id = qfm.qcc_file_id </pre>

Advanced SQL Server Format

	<pre> LEFT JOIN dimension d ON qfm.qcc_file_model_id = d.qcc_file_model_id LEFT JOIN part p ON qfm.qcc_file_model_id = p.qcc_file_model_id LEFT JOIN measurement m ON p.part_id = m.part_id AND d.dim_id = m.dim_id WHERE qf.qcc_file_desc = 'widget1' AND d.dim_desc = 'Feature 6' </pre>
Notes:	Make sure to use LEFT JOIN rather than INNER JOIN or the non-matching records will be excluded.

Example 6: Retrieving Out Of Control Values	
Description:	This query retrieves measurements that are outside of control limits. This could easily be changed to look at spec limits as well.
Parameters:	No params for this query.
Query:	<pre> SELECT p.record_number, d.dim_desc, d.ctl_upper, d.ctl_lower, m.value FROM qcc_file qf INNER JOIN qcc_file_model qfm ON qf.qcc_file_id = qfm.qcc_file_id INNER JOIN part p ON qfm.qcc_file_model_id = p.qcc_file_model_id INNER JOIN dimension d ON qfm.qcc_file_model_id = d.qcc_file_model_id INNER JOIN measurement m ON p.part_id = m.part_id AND d.dim_id = m.dim_id WHERE m.value NOT BETWEEN d.ctl_lower AND d.ctl_upper </pre>
Notes:	This could be narrowed to include only those parts from a particular qcc_file_model or qcc_file by adding additional WHERE statements.

8. A Word about Windows Services

Windows services are a useful way to ensure your programs stay running even if the user doesn't stay logged into the workstation. Unlike normal Windows applications, services stay running on the computer even when the current user has logged off. This ensures that EDL continues to load the data on schedule even if nobody is currently logged into the PC. However, with this benefit come a few considerations. Below is a list of items to consider when deploying the Enterprise Data Loader:

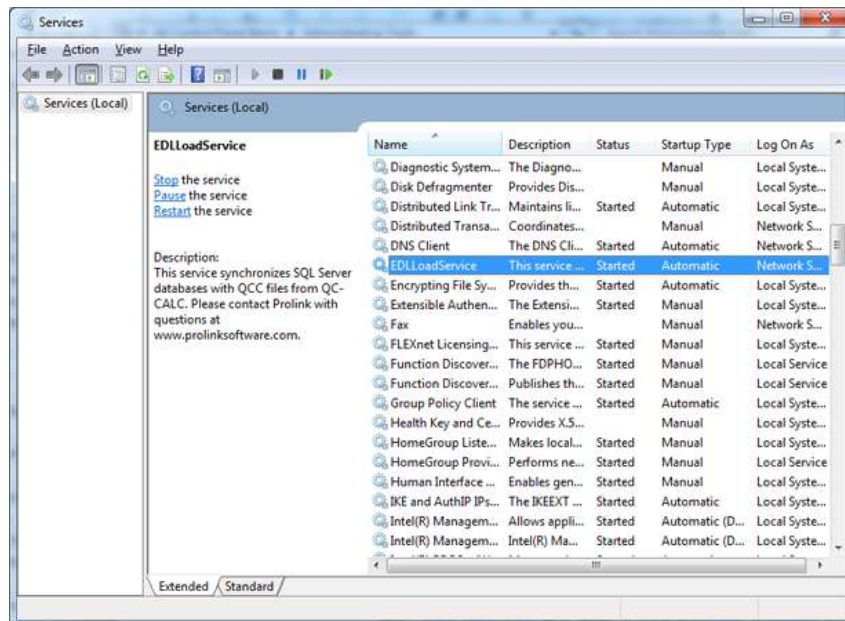
EDL Client vs. EDL Load Service	pg. 51
Login Permissions for Services	pg. 52
Mapped Paths/Drive Letters Illegal for Services	pg. 53
Confusion between the Client and Service	pg. 54

8.1 EDL Client vs. EDL Load Service

Since Windows services do not have screens, we split EDL into separate pieces; the EDL Client and the EDL Load Service. The client is the application that appears when you run EDL, change settings, create events, etc. The EDL Load Service is a silent application that runs in the background. It actually does the loads even if you click Run Now from the Summary screen within the client. Rather than running the load itself, the client actually sends a message to the service telling it to run the event when the Run Now link is clicked.

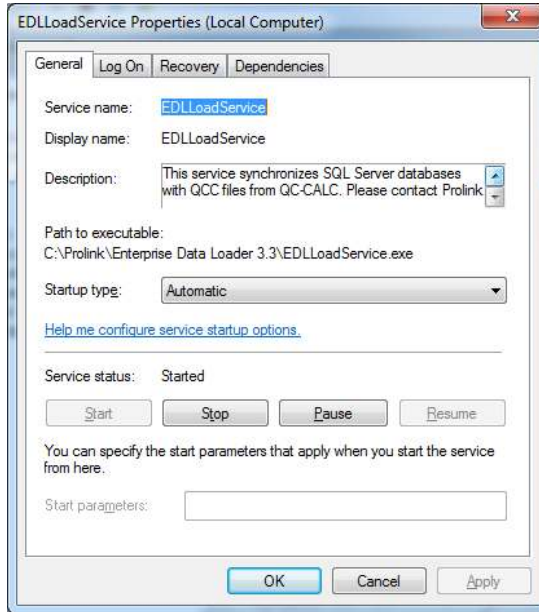
The client shows a red or green message at the bottom of the screen indicating whether or not the EDL Load Service is running. The service can also be toggled on/off using the Tools menu in the client.

You can also see the EDL Load Service running by going into the Services area in Windows. This is under **Start > Control Panel > Administrative Tools > Services**.



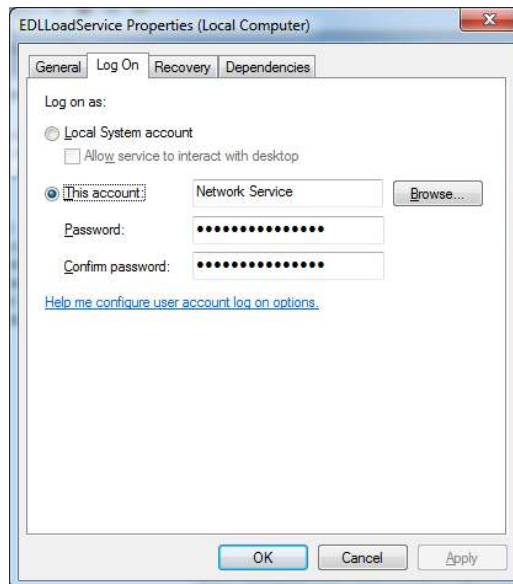
A Word about Windows Services

If you right click on the EDLLoadService service and choose Properties, the following dialog box appears. From this screen you can see the path to where the executable actually exists, and control the starting and stopping of the service.



8.2 Login Permissions for Services

Remember that the service is running regardless as to whether or not you are logged into the workstation. As such, it must have its own login settings in order to run and access files on the network. When you set up the security for the service, you're basically telling it to access the network as "xyz" user account. Whatever xyz has access to read, write, and delete, so does the service since it will log into the network as xyz. Typically you will want your IT department to create an "application" type user account on your domain. The difference between an application type account and a standard user account is that the application account doesn't usually follow all of the same rules.



For instance, it wouldn't make sense to have an expiring password since the service isn't going to be aware enough to change its password every 60 days.

A Word about Windows Services

Make sure the user account you have IT create has **read, write and delete** privileges to any folder where the QCC files will exist.

Once your IT department has given you a user id and password, switch to the **Log On** tab, choose **This account** and enter the user id and password.

NOTE: You will need to stop and start the service after applying this in order for the changes to take effect.

NOTE: You can type your own login user account and password for testing purposes if need be and it will work without problems. However, it is not recommended beyond testing because you will eventually change your password, forget to update it here, and the EDL Load Service will no longer have access to the network.

By default, you'll notice that the Log On is set to "Network Service". This is a special user account in Windows that should work out of the box. However, in many cases it will not work because your IT department's security policy removes access to the "Network Service" account. This is done for security reasons so viruses do not use your network for unintended purposes.

8.3 Mapped Paths/Drive Letters Illegal for Services

Unfortunately, using mapped paths does not work for services. This is because mapped drive letters (such as your "N" drive) are user specific. This means your N drive could be different from someone else's N drive. To simplify this, Microsoft has stated that services must use Unified Naming Convention (UNC) paths.

The UNC format will get you to the same place, but uses the server and shared directory directly rather than mapping it to a drive letter. The format is [\\servername\shared folder\folder\folder](#).

Using the example in the picture on the right, notice the H drive that is mapped in Windows Explorer. The actual path to which the H drive is mapped is [\\ntserver\d-drive](#). You can see this is defined just to the left of (H:). If you want to point to the Backup directory, then you would use [\\ntserver\d-drive\Backup](#). This is the equivalent of h:\Backup.



Anywhere you must specify paths when setting up EDL (including the parent path of the event and assignable cause/corrective action paths), make sure you use UNC style paths. Remember that the service will need to get to the path and it won't understand what your "M" drive is.

NOTE: You can use local drive letters such as C:\ or D:\ since they are local to the PC.

8.4 Confusion between the Client and Service

Often times, permissions can be confusing because you are dealing with the EDL Client and the EDL Load Service. The EDL Client is running under your permissions since you are logged into the local workstation. The EDL Load Service is running under its own user account (see the last section).



To help with this, we added the **Check Now** link on the main screen. The Check Now link launches a screen that sends a message to the EDL Load Service and asks it to report back all of the files it sees in the parent path within the event.

The EDL Load Service responds by listing every file it sees in the folder. This is displayed on the screen allowing you to confirm that the EDL Load Service actually has the ability to see the QCC files it is supposed to load. This is the most common place for confusion. When you set up the event, you are able to see the files in the directory, but that is because you are logged in as yourself. Remember that the service is logged in as a different user and may not have the same access to the network.

9. Prolink XML Format

9.1 Description

The Prolink XML format was developed as a standard XML format used to describe parts. This general format has not changed although fields have been added to it over the years as new features were added to QC-CALC.

One convenient feature to the Prolink XML format is its separation by part. Each part (record) is completely defined in the <part> tags. There are no outside references that need to be linked. Because of this, some of the data may appear redundant but it is necessary since meta data changes over time and QC-CALC must react to those changes. Another format option could have been to define the dimensions once as a separate section from the actual values and then have each part only contain the values. This method could have problems, however, since there could be multiple parts per file and the part type's tolerances could change over time. For instance, let's say there were 5 parts in the file all of the same part type. If we defined the nominals and tolerances in a separate section, then we must guarantee that no changes in nominal or tolerance can occur across the different part records contained. This would have reduced flexibility if user had chosen one part from the past and one from present each with different tolerances.

This document does not take into account subgroups and is only a raw data output. Subgroups may be handled in a future version of the spec.

9.2 Conventions

Lower Case - All tags are lower case with words separated by an underscore character (“_”). Since XML is a case sensitive language, this will reduce any confusion and bugs with capital letters.

Handling Labels - Any field that is a result of user input is declared as a separate element and will not appear as an attribute. This is because xml attributes cannot handle illegal characters. These elements will all have CDATA sections just in case the label has characters that would cause the xml to not be well formed.

9.3 Field Definitions

9.3.1 prolink_xml

This is the top most parent and contains two attributes; version and date. The version is the version of the xml spec and the date is the date/time the file was literally created.

9.3.2 part

This element represents each part that is inspected and a prolink_xml tag can have any number of part elements. All part details are contained within this element. The attributes of this element are date, id, deleted, and edited. Possible elements contained within the part element are:

1. Exactly one name element
2. One to many dim elements
3. Zero to many trace elements
4. Zero to many label elements.

Example:

```
part
  name
```

```
    dim
    dim
    dim...x
    trace
    trace...x
    label
    label...x
/part
```

Attributes:

- **date** - The date/time the part was inspected.
- **id** - The unique identifier of the part. This will be used to identify the part in case an update is needed after the initial import.
- **deleted** - Indicates if the deleted flag has been flipped for this part.

Elements:

- **name** - This element holds the part name as taken from the inspection machine's part routine name. It also matches the name of the .qcc file. There is a CDATA section in case the name has illegal characters.
- **dim** - This element holds everything about a particular dimension (feature) including the value. The attributes of this element include num, nominal, tol_plus, tol_minus, tol_type. The elements contained are exactly one name and exactly one value element.

Elements:

- **name** - This element holds the feature label. There is a CDATA section in case the name has illegal characters.
- **value** - This element holds the actual value of the record as reported by the inspection equipment. This is just the record value and does not take into account subgroups.

Attributes:

- **num** - The number of the *n*th feature.
- **nominal** - The nominal for this dimension.
- **tol_plus** - The amount of the plus tolerance. Add this to nominal to get Upper Spec Limit.
- **tol_minus** - The amount of the minus tolerance. Subtract this from nominal to get Lower Spec Limit. This is always a positive number.
- **tol_type** - The type of the tolerance. This value can be one of the following four constants: NONE, BI, SSU, SSL. These are Not Toleranced, Bilateral, Single Sided Upper, and Single Sided Lower respectively.
- **deleted** - A flag indicating if this particular point was deleted. Values: 0 or 1.
- **edited** - A flag indicating if this particular point was edited. Values: 0 or 1.

9.3.3 trace

This element holds any non-measurement trace information at the part level. It has two attributes; type and num. It also has two elements; name and value. Trace fields map to the factor fields in QC-CALC.

Elements:

- **name** - This is the name of the trace field. There is a CDATA section in case the name has illegal characters.
- **value** - This is the value of the trace field. There is a CDATA section in case the value has illegal characters.

Attributes:

- **type** - This is the type of the trace field. Valid types are "TEXT" and "NUMERIC".

- num - This is the number of the trace field in the order of the data. For instance, the first Text and Numeric factors would each have num="1". The second of each would have num="2" and so forth.
- label - This element holds any other information about the part such as Assignable Cause/Corrective action information. This element has two attributes; type and num. The value of this element is the value of the label. There is a CDATA section in case the value has illegal characters.

Attributes:

- type - This is the type of the label field. Current possible types include "CAUSE" and "ACTION", but more can be added as the standard evolves.
- num - This is the number of the label in the order of the data. For instance, the first assignable cause would have num="1". The second would have num="2" and so forth.