Labworks, LLC
230 North 1200 E #202
Lehi, UT 84043
Phone: 844 452-2967
www.labworks.com

LABWORKS LIMS v6.9

Admin Guide

Last Updated: May 2019

Notices

The information contained in this document is subject to change without notice.

Except as specifically set forth in its terms and conditions of sale, LABWORKS makes no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

LABWORKS shall not be liable for errors contained herein for incidental consequential damages in connection with furnishing, performance or use of this material.

Copyright Information

Produced in the U.S.A.

Trademarks

Registered names, trademarks, etc. used in this document, even when not specifically marked as such, are protected by law.

LABWORKS is a registered trademark of LABWORKS, LLC.

LABWORKS is a trademark of LABWORKS, LLC.

Microsoft, MS, and Microsoft Visual Basic are registered trademarks and Windows is a trademark of Microsoft Corporation.

Customer Support

This software is supported by LABWORKS Informatics Support.

LABWORKS LLC. | For the Better
https://labworks.com/contact/

Table of Contents

Contents

Notices 2
Copyright Information 2
Trademarks 2
Customer Support 2
Table of Contents 1
1. System Administrator Reference 1
Overview 1
Data Fields Setup 3
Error Codes and Troubleshooting 8
Translation Tool 9
Automatic Email Reporting 16
BarTender Label Setup 25
Utilities 31
Archive Setup and Configuration 39
LABWORKS INI File 49
Glossary 78
Additional Reference Material 81
SYSMGR/System Manager Settings 92
2. LABWORKS Sample Login - Sample Reporting ("Paperwork") 95
Introduction 95
LABWORKS Sample Login 95
Sample Login Configuration 96
Sample Configuration 98
List Screen Configurations 99
User Program Configuration 102
Group Configuration 104

Paperwork Configuration 104
Sample Login Reporting Overview – The Complete Process 108
New Paperwork Template Design or Modification 109
LABWORKS Reporting 113
Data Model Configuration 116
LABWORKS Sample Login – Logging in Samples 118
Sample Login Complete Report Rendering Process 121
Persisted Groups in LABWORKS 123
3. LABWORKS Crystal 11 Report Designer 132
About the Program and Getting Started 132
Create/Edit Reports Using Exported Data 134
Create/Edit Reports Using Direct Connect 144
Menu Items and Special Editing Features 151
Record Selection 163
Sorting and Grouping Data 171
Running Totals 183
Multiple Section Reports 188
Formatting 197
Charting 206
Cross-Tab Objects 224
Using Formulas 231
Creating Formulas with Basic Syntax 240
Creating Formulas with Crystal Syntax 272
Parameter Fields 306
Subreports 314
4. QA/QC Module Technical Reference 324
Technical Prologue 324
QA/QC Module Execution 324
QA/QC Application Specifics 326
Common QA/QC Problems 339

5. QA/AC Module Setup and Configuration Reference 340
Overview and Setup 340
QA/QC Template Builder 343
QA/QC Batching and Other LIMS Modules 364
Additional QA/QC Common Terminology 370
6. Discharge Monitor Technical Reference 382
Technical Prologue to DMR Usage 382
Existing Users: Legacy Migration 388
DMR Calculation Installations 393
DMR Configuration Settings 402
Common DMR Problems 403
7. Discharge Monitor User Reference 405
Labworks-Discharge Monitor 405
Getting Started 406
Workflow 413
8. Automated Results Posting 448
Introduction to Automated Results Posting 448
Automated Results Posting Tools: Configuration and Usage 457
The AUDIT Step: Automated Results Posting 473
The Result Conditioning Step: Automated Results Posting 475
The Specifications Check Step: Automated Results Posting 494
Appendix A: The Complete List of System Control Options in SYSMGR/System Manager 503
Appendix B: The Complete List of LABWORKS.INI Settings 521

1. System Administrator Reference

Overview

General Overview

Philosophy and Function

LABWORKS is a Laboratory Information Management System (LIMS). It is a database used for recording sample information and test results in a laboratory environment. LABWORKS essentially involves logging in a sample, entering test results for the sample, validating the sample, managing quality assurance for the sample, and providing invoices and/or reports for the sample. This chapter provides the System Administrator with the following information:

- Overall philosophy and function of LABWORKS
- General overview and terminology involved in the LIMS
- Various LABWORKS database options

Please note that additional instructions on how to set up and use most aspects of LABWORKS are contained in the Online Help for the product.

General Overview and Terminology

LABWORKS is a modular application. Each set of related functions in LABWORKS is contained in a subset or smaller streamlined programs. For example, a sample may be logged in by launching Multi-Sample Login, but sample results may be entered by launching Spreadsheet Results Entry or a related program. Each of these modular programs is specialized for its function to reduce redundant code and reduce the size of the applications while maintaining an efficient and easy-to-maintain code base. This allows the software programmers to update or modify the individual programs without affecting the functionality of other areas of the LIMS.

LABWORKS Database Options

LABWORKS is a client/server application. This means that client application us used to send, filter, and read data to and from the LABWORKS database. Additionally, the LABWORKS uses services that run on a server computer to authenticate users and provide and received data. The server resides on the server computer that is running LABWORKS Services. The file server is a computer that contains files common to all users. The LABWORKS base files are stored on the file server. The LABWORKS database is access from a database server. The database options are:

SQL® Server

SQL Server is from the Microsoft Corporation. This option requires a database administrator who can set up and maintain the database. It also accommodates a very large user base and can easily handle a large number of active samples and their related data.

Oracle®

Oracle® offers basically the same functionality and a much larger capacity than the SQL ®server. It also requires dedicated maintenance and setup in the form of a database administrator.

Default LABWORKS logon

After installing LABWORKS, the LABWORKS User Logon window is displayed for you to log on. LABWORKS provides a default User ID and password for initial log on. This User ID and password can and should be changed later either by the User or the System Administrator. The default log on User ID is USR, and the default Password is 1 (number one).

Default LABWORKS logon

If this is a brand new install, when you log onto LABWORKS for the first time, LABWORKS provides a default User ID and password for initial log on. This User ID and password can and should be changed later either by the User or the System Administrator. The default log on User ID is USR, and the default Password is 1 (number one).

Note: The first time you attempt to log into LABWORKS from a workstation, a dialog appears that prompts you to type in the name of the server you wish to connect to.

1. From the LABWORKS User Logon window, enter **USR** as the User ID.
2. Enter **1** (the number 1) for the **Password**.
3. Click **OK**.

Troubleshooting

Even though every attempt has been made to accommodate every conceivable situation, there are times when the installation process might not complete as expected. If you have trouble with the installation process and the documentation does not answer any of your questions, then please call LABWORKS Technical Support promptly for assistance.

Installing and Configuring BarTender

The BarTender software program is used to create custom labels within LABWORKS. If your laboratory plans to use BarTender, the BarTender installation manual provides a brief, but to the point, and detailed guide for its installation. However, it is important to note that the Enterprise version of BarTender requires installation of a license server component called the Seagull License Server. This component must be installed to a computer that allows access at all times by the workstations that will be printing labels from LABWORKS.

To install BarTender:

1. Install the Seagull License Server as detailed in the BarTender manual.
2. Install the BarTender component on each workstation that will be printing labels per the BarTender installation manual instructions.

NOTE: Since this is a third-party component, LABWORKS representatives can only provide limited support for the actual usage of BarTender itself.

The information for defining paperwork settings at login and creating a BarTender label format can be found in the LABWORKS LIMS Laboratory System Administrator Manual. Refer to that manual for details.

Data Fields Setup

Introduction

Data organization is a key factor when installing and setting up LABWORKS. The System Administrator must organize data in a manner that allows necessary access to sample data by users and customize the data in LABWORKS to meet company requirements. Data organization includes file location, user data accessibility, sample data ownership, defining sample-related data fields, establishing an analysis list and a location code/sample point list, and updating and maintaining information as needed. Data organization of user fields and database tables are described in this chapter.

LABWORKS allows the creation of extra sample-related data fields. These fields can be text, dates, or numeric. This feature allows the System Administrator to establish data fields that may not be available by default in the LIMS or to create fields necessary to meet their particular laboratory requirements. Field size is limited to 160 characters, and field names are limited to 16 characters and should not contain spaces or non-DOS filename characters.

In order to create user fields, two tables in the LABWORKS database must be edited. The two tables are titled SUSERFLDS and USERFLDS. The properties of the tables are listed below.

NOTE: The examples used below contain sample user fields for illustration only. Your tables will vary.The fields that are required by LABWORKS and that will remain static are SAMPNO inSUSERFLDS and LOCCODE in USERFLDS.

NOTE: If you do not know how to access and edit the SUSERFLDS and USERFLDS tables, obtain assistance from someone in your company or contact LABWORKS Technical Support.

User Field Setup

USUSERFLDS Table

The SUSERFLDS table contains actual data assigned to each LABWORKS sample.

Name	Type	Size (characters)
SAMPNO	Text	7
OUR_DASH_No	Text	14
INSPECTOR_REF	Text	11
SHIPDATE	Date/Time	8
DISPDATE	Date/Time	8
CATEGORY	Text	13
LATITUDE	Text	10
LONGITUDE	Text	10
SAMPTYPE	Text	3

SAMPFREQ	Text	3
SITECODE	Text	3
Permit_No	Text	15
Discharge_No	Text	10
Time	Date/Time	8
Matrix	Text	16
MSDS_REF	Text	16
MAP_REF	Text	16

NOTE: The first User-Defined Field you create should be the most queried field that does not already exist in LABWORKS – second to the sample ID and Location Code.

USERFLDS Table

The USERFLDS table contains default values for each user field for all location codes.

Name	Type	Size (characters)
LOCCODE	Text	8
OUR_DASH_No	Text	14
INSPECTOR_REF	Text	11
SHIPDATE	Date/Time	8
DISPDATE	Date/Time	8
CATEGORY	Text	13
LATITUDE	Text	10
LONGITUDE	Text	10
SAMPTYPE	Text	3
SAMPFREQ	Text	3
SITECODE	Text	3
Permit_No	Text	15
Discharge_No	Text	10
Time	Date/Time	8

Matrix	Text	16
MSDS_REF	Text	16
MAP_REF	Text	16

Actual values in these tables are populated by LABWORKS.

New User Field Setup

A maximum of 50 user-defined fields may be set up in LABWORKS. Fields closer to the top of the list have priority in the user interface for searches and other functions; so place the most commonly used fields at the top. User field 1 also has extended search and formatting capabilities. When defining the field name, be descriptive but be aware that LABWORKS limits the field name to 16 characters. Field size is limited to 160 characters, and field names are limited to 16 characters and should not contain spaces or non-DOS filename characters. Careful pre-planning of the user field setup is a necessity.

1. Type in the new field name, type, and size in both tables, making sure the field has the same properties in other tables. These fields can be text, date/time, or numeric.

NOTE: Limit the new field name to 16 characters, since this is maximum number of characters allowed for this field in LABWORKS.

2. Make sure the fields are in the same order in both tables.In the above example, for instance, notice that LONGITUDE follows LATITUDE and precedes SAMPTYPE in both tables.
3. Set up the data fields to accept blank entries, since samples may be logged in without data in one or more of the user fields.

NOTE: Fields must have the same properties and be in the same order in the SUSERFLDS and *USERFLDS tables.*

Other Setups

Trend plot program setup

Use of the trend plot program requires the presence of tables TPPLOT and TPHEADER in the LABWORKS database and the privileges TRENDRUN (Display trend plot groups) and TRENDSETUP (Define and configure trend plot groups) in the PRIVS table of the database.

Document reference maintenance program setup

The Document Reference Maintenance Program requires that two user privileges must be present in the PRIVS table of the database so that they can be assigned to the program users.

In order to reference MSDS documents, a text type user defined field named MSDS_REF must be added to the LABWORKS database. To reference map documents, a text type user defined field named MAP_REF must also be added to the database. User defined fields are created by adding columns to tables USERFLDS and SUSERFLDS. See the *New user field setup* section for instructions for adding user defined fields. No special user defined fields are required to support SOP document referencing.

System Maintenance

Maintenance Functions

Introduction

An integral part of the System Administrator's responsibilities is the on-going maintenance of LABWORKS and the associated databases. As company and users' requirements change, the system functionality must be updated and altered to meet demands. System security and assurance that data files are not lost in case of computer failure are extremely important. A method of maintaining, backing up, and securing data is crucial. Maintenance and backup are discussed in this section.

LABWORKS Backup Strategy

LABWORKS is a database application that is generally shared by several people in a work environment. As such, it may be difficult to control the environment in which LABWORKS resides, and frequent backups are always recommended. Although LABWORKS provides utilities for archiving old data, the archive function is not a replacement for backing up the programs and data paths.

What to Back Up

When backing up LABWORKS, remember that even if your data is stored in SQL® Server, or Oracle®, there are many other data related files and folders that should be backed up. As a general rule, it is best to backup your database, plus the entire LABWORKS datapath, and the LABWORKS program path. If this is done on a regular basis, restoring the system to its working configuration is as easy as restoring the backup. Backing up the programs and database also adds the convenience of removing the need for running the LABWORKS administrative install in the event of a problem that requires restoring the backup.

NOTE: It is best to backup your database, plus the entire LABWORKS datapath, and the LABWORKS program path on a regular basis.

When to Back Up

Although LABWORKS is considered to be very stable, many other variables are introduced when it is placed in a working environment. These variables, such as a network operating system, hardware integrity, user proficiency, and even the state of each computer running LABWORKS, all have an effect on the data system. If any one of these variables introduces a problem, the data may be corrupted, necessitating restoration of a backup. As a general rule, remember that if you backup each day, you should never lose more than a day's work, but if you backup only once a week, you may potentially have to re-enter a week's worth of data.

Most backup programs or packages have a suggestion for a good backup strategy. Following one of the suggested strategies should be sufficient. Additionally, if you have an IS department, the individuals in that department may be able to assist you in setting up a backup system or suggest a strategy.

A popular weekly cycle schedule includes a full backup on Monday and partial backups Tuesday through Friday.

NOTE: If you backup each day, you should never lose more than a day's work if your system fails to function properly.

Useful Database Maintenance Utilities

LABWORKS is primarily a database. It incorporates many complex and varied programs, but the heart of the system is the data that the LIMS maintains. As with any database, LABWORKS data must be periodically maintained and

managed. Since the underlying data structure varies with different versions of LABWORKS (SQL® Server, Oracle®), the LIMS manager is responsible for determining the best management and maintenance strategies for his/her specific installation. This section of the chapter attempts to propose a few options and provide a few helpful hints for maintaining LABWORKS.

Database Optimization

Most databases offer specific tools for optimizing the data. These tools should be used regularly to ensure optimum performance of the LIMS. A regularly scheduled database diagnostic is recommended (once a month at minimum, but more often depending on how heavily the LIMS is used – this is dependent on each individual installation of LABWORKS). Additionally, any errors should be repaired as soon as possible to maintain the speed and data integrity.

Network Speed

Networks vary greatly in their hardware and software implementation. They also vary greatly in the way they transfer data packets; this can affect the speed at which applications transfer data. The speed of LABWORKS is dependent in part upon the speed of the network.

Error Codes and Troubleshooting

Technical Support

Even though every attempt has been made to accommodate every conceivable situation, there are times when the installation process may not complete as expected. Please refer to this chapter for possible causes. If you go through the following sections and still need assistance, call LABWORKS Technical Support.

To contact LABWORKS, LLC. Technical Support:

- https://labworks.com/contact/

Error Logs

If errors are experienced during normal use of LABWORKS, check your user and program paths for errolog files. If the LABWORKS client can make a connection to the database these errorlog files are read and the contents added to the ErrorLog table. These files/database records detail errors received in LABWORKS under normal use. See the Frequently Asked Questions section below for questions and answers concerning errors and error logs.

If you try all the solutions in the above sections and rerun LWSysConfig.exe and are still having trouble with installation, please make detailed notes about the problems and email them to LABWORKS Support at web site at https://labworks.com/contact/

Translation Tool

Introduction to the Translation Tool

The Translation Tool is supplied with your LABWORKS application software and it enables you to customize the application to run in the language of your choice. You can use this tool to create, modify, review, import and/or export a language.

LABWORKS is shipped with a set of standard languages; however, if you find that you need to run LABWORKS in a different language, or that you need to modify an existing language to better reflect your organization's laboratory work flow and business practices you can use this tool. Once you have customized or created a language you can select to run the application in the language of your choice from the LABWORKS login screen.

The LABWORKS Translation Tool is a very flexible translation editor and the rest of this chapter will show you how to use this tool. The LABWORKS Translation Tool is installed in the "Server" folder as part of the server install.

The Main Translation Tool Window

The **Title Bar** at the top of the window displays the name of the product and the standard Windows buttons - minimize, maximize and close.

The **Menu Bar** contains the menu commands that enable you to work with the

Translation Tool, as described below:

Menu	Command	Description
File	**New Language > New …**	Opens the Add New Language dialog so that you can create
	Save language	Saves the language currently being edited.
	Export Language…	Opens a dialog that allows the user to export a language
	Import Language…	Opens a dialog that allows the user to import a language
	Exit	Closes the Translation Tool.
Edit	**Cut**	Cuts text from an edit cell and places it in the clipboard.
	Copy	Copies text from an edit cell and places it in the clipboard.
	Find	Finds the defined text/number in the selected field.
	Paste	Pastes the contents of the clipboard into a cell.

Menu	Command	Description
	Edit Language…	Opens dialog that allows the user to select a language for editing.
	Delete Language	Opens dialog that allows the user to delete a custom language.
View	**Tool Bar**	Switches the Tool Bar on and off. A checkmark is displayed when the Tool Bar is switched on.
	String ID	Switches the String ID column on and off.
Actions	**Sort by String ID**	Sorts the language by string ID.
	Sort by Reference language	Sorts the language by reference language text.
	Sort by New language	Sorts the language by new language text.
	Sort by Language ID	Sorts the language by ID number.
	Review Language Update	Opens Updating Language table dialog.
	Copy language to new database	Opens Copying language tables to new database dialog.
Help	**Contents**	Displays the Help window.
	About LABWORKS	Displays the About box.

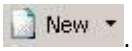
The **Toolbar**, located below the menu bar, contains icons for standard interactions that will be performed frequently. All of the commands on the tool bar are also available through the menu bar.

Create a New Language

There are two ways you can create a new language. You can create an entirely new language, while using a previously created language as a reference, or you can create a new language from an existing language.

To create a new language:

1. From the Translation Tool screen select **File > New Language** or click



2. The Add New Language dialog appears and a Language ID for the new language is automatically assigned.
3. Enter a **Name** for the new language.
4. Enter a **Description** for the new language.
5. Select a **Reference** Language.

Note: The Reference Language is the language that you will base your new language on. As you are translating the Reference Language's strings into your new language, you may find it necessary to see how or where a particular string is used in the application. If you click FindText from the Translation Tool's toolbar, the LABWORKS application will open in the selected reference language. The application will also display the string number associated with each string, so that you can browse through the application until you find the string number that corresponds to the string you wish to translate.

6. Click **OK**. You are returned to the main Translation Tool screen. The spreadsheet for the Translation Tool now shows the String IDs for each string, the Reference language column is now labeled according to and populated with the Reference Language you selected, and the New Language column is now labeled according to the name you gave the new language.
7. Enter your translations in the **Edit New Language** column.
8. Click **File > Save** to save the new language.

Now, when you log into LABWORKS, you can select this new language from the log in screen in order to have the LABWORKS application display in the new language.

To create a new language from an existing language:

1. From the Translation Tool screen select File > New Language > From Existing or click 

and select From Existing.
2. The Add New Language dialog appears and the Language ID for the new language is automatically assigned.
3. Enter a Name for the new language.
4. Enter a Description for the new language.
5. Select a Reference Language.
6. Click **OK**. You are returned to the main Translation Tool screen. The spreadsheet for the Translation Tool now shows the String IDs for each string, the Reference language column is now labeled according to and populated with the Reference Language you selected, and the New Language column is now labeled with the name you gave the new language; however this column is first populated with the strings contained in the Reference Language.
7. Make your edits in the **Edit New Language** column.
8. Click **File > Save** to save the new language.

Now, when you log into LABWORKS, you can select this new language from the log in screen in order to have the LABWORKS application display in the new language.

How Languages are Assigned an ID

Languages 0-99 are controlled by LABWORKS LLC. These are the languages that come standard with your system. The ID numbers 100-999 are assigned by the Translation Tool when you create a new language. LABWORKS automatically assigns the next available ID when you create a new language or make a copyof an existing language.

Export a Language

If you wish to create translations for a new language outside of LABWORKS, you can export a language directly into an Excel spreadsheet and use Microsoft Excel as your editing environment.

To export a language:

1. From the Translation Tool screen select **Actions>Export Language**. The Export Language dialog appears.
2. Select a language to export from the **Name** drop down menu.
3. Click **OK**.A file selector dialog appears.
4. Enter a **File** name and location for the language you are exporting.
5. Click **Export**.The selected language is exported.

Import a Language

If you have created translations for a language in a properly formatted Excel spreadsheet you can import the contents of that file directly into LABWORKS.

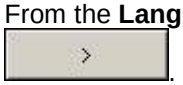
To import a language:

1. From the Translation Tool screen select **Actions > Import Language**. The Import Language dialog appears.
2. Enter a **Name** and **Description** for the language you wish to import.
3. Click **OK**.A file selector dialog appears.
4. Select the file you wish to import.
5. Click **Import**.The selected file is imported.

Edit a Language

1. From the Translation Tool screen click **Edit > Edit Language**. The Edit Language dialog appears.
2. Select a language from the **Name** drop down menu.
3. A description of the selected language appears in the **Description** field.
4. Select a Reference Language.
5. Click **OK**. You are returned to the main Translation Tool screen. The spreadsheet for the Translation Tool now shows the String IDs for each string, the Reference language column is now labeled according to and populated with the Reference Language you selected, and the New Language column is now labeled and populated with the name of the language you selected for editing.
6. Edit the strings contained in the **Edit Language** column.
7. Click **File > Save** to save your changes.

Copy Language Tables to a New Database

1. From the Translation tool screen click **Actions > Copy language to new database**.
2. The Copy language tables to new database dialog appears.
3. From the **From** drop down menu, select the database that contains the tables you wish to copy.
4. From the **To** drop down menu, select the database that you wish to copy the tables into.
5. From the **Languages to Copy** section, click on the languages you wish to copy and then click  . The selected languages now appear in the drop down menu on the right-hand portion of the screen. OR
6. Click  to copy all available languages. All of the languages now appear in the drop down menu on the right-hand portion of the screen.

Note: To deselect a language, select the language(s) listed in the right-hand portion of the screen and click  . The selected languages are returned to the drop down list on the left-hand portion of the screen; or click  to move all of the languages on the right-hand portion of the screen over the left-hand portion of the screen.

7. Click **OK**.

Note: You may receive a message that states: “Copying a language will overwrite the selected language in the target database. Do you want to continue?” If this message appears click **Yes** to overwrite the target database.

Update the Language Tables When There is a New Software Release

When you receive an update to the LABWORKS software, part of the installation process involves updating the language tables to accommodate for any changes that have been made to the user interface. When you install a LABWORKS update, the software will compare the current database tables to the previous database tables and any differences are placed into an exception table, which you can review in order to view the new translations and decide whether or not to keep the translations or customize them.

To review updated languages:

1. From the Translation Tool window select Actions > Review Language Update. The Review Language Update window appears.
2. From the Database drop down menu, select the name of the database you wish to review. The Languages to review list is populated with a list of available languages.
3. Select the languages you wish to review and then click  .The selected languages now appear in the drop down menu on the right- hand portion of the screen.

OR

4. Click  to select all available languages. All of the languages now appear in the drop down menu on the right-hand portion of the screen.

Note: To deselect a language, select the language(s) listed in the right-hand portion of

the screen and click  . The selected languages are returned to the drop down list on the left-hand portion of the screen; or click  to move all of the languages on the right-hand portion of the screen over the left- hand portion of the screen.

5. Click **OK**.You are returned to the Translation Tool window, and the editing portion of the window is now populated with four columns. The four columns are as follows:

String ID – This column contains a string ID for each string.

Language – This column contains the name of the language.

PKI Language – This column contains the new translations that are shipped with the software.

Local Language – This column contains any changes you have made to the standard language. Initially, this column is highlighted in Red. Items highlighted in red require action.

6. To select a string for use in the database, click in a cell in either the **PKI Language** column or the **Local Language** column.The cell color changes to green to indicate that you have selected that string to use in the database.

Note: If you change your mind you can click on the unselected sell and transfer the selection to it.

7. When you are done reviewing language updates click **File > Save**.

Note: If you do not complete the review session, but still click **File > Save**, then the next time you open this review, the previously reviewed and selected cells will not appear; only the cells that require action.

Automatic Email Reporting

Automatic Email Reporting Installation

Automatic Email Reporting is an optional LABWORKS feature that enables you to automatically send emails to clients for sample confirmation and results reports upon demand. If you have the Automatic Email Reporting tool set up and configured, then once samples have been logged into the LABWORKS database, emails containing a confirmation number and due date for receiving sample results, are sent to the sample submitter. Customers can then request reports by replying to the confirmation messages. The reports are then sent to the submitter's email address as PDF files.

This section shows you how to set up and configure LABWORKS for Automatic Email Reporting.

Installation Requirements

In order to configure LABWORKS for automatic email reporting you must have the following components:

- Microsoft Outlook must be properly configured and setup with a mailbox

Note: Microsoft Outlook must be Version lower than 2002 due to the security features that require one to answer whether or not the program can access Outlook.

Once you have the client installed you will need to perform the following steps, which are described in this document:

- Add new user defined fields to the USERFLDS and SUSERFLDS database tables
- Enable/disable the automatic email reporting privilege for the correct users
- Configure the System Manager Settings from the AER_Daemon6.exe program
- Configure a Multi-Sample Login template to include the new user-defined fields specified for email and the user program AER_LOGIN6.EXE
- You must have a sub-directory called AER_RPORTS in the LWDATA Directory
- You must have a sub-directory called EMAILTEMPLATES in the
- LWDATA\Crystal directory.
- Within the Crystal directory mentioned above, you must have at least one AER type confirmation .RPT file and one AER type confirmation .CEF file. i.e. AER_CONFIRM.RPT and AER_CONFIRM.CEF

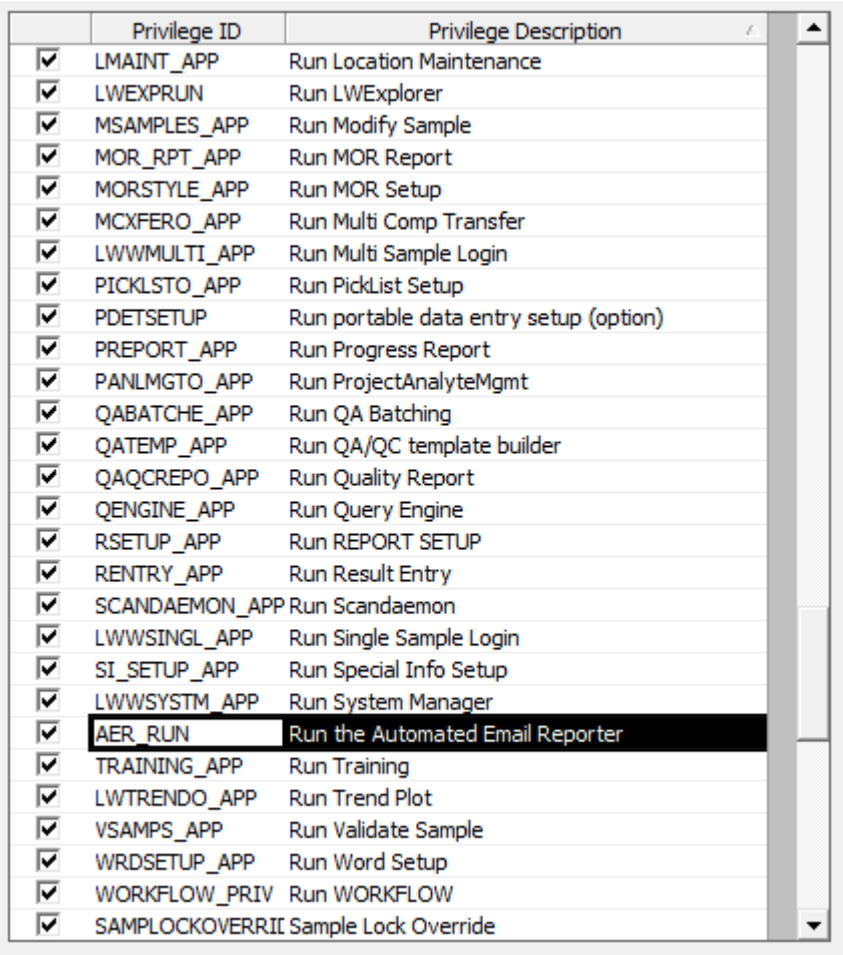
Note: The email reporting is designed to run UNATTENDED. The daemon checks only UNREAD messages in the inbox. If you open an email intended for LABWORKS processing, the daemon will not process this message because it will see the message as READ. If you accidentally open a message, mark the message as UNREAD (right mouse-click).

Enable Email Reporting

In order to run the Automatic Email Reporting tool users must be assigned the privilege to "Run automatic email reporting." The following section describes how to enable or disable this privilege.

1. From the LABWORKS Desktop click **Utilities > System Manger**. The System Manager Password screen appears.
2. Enter the System Manager password.The System Manager screen appears.

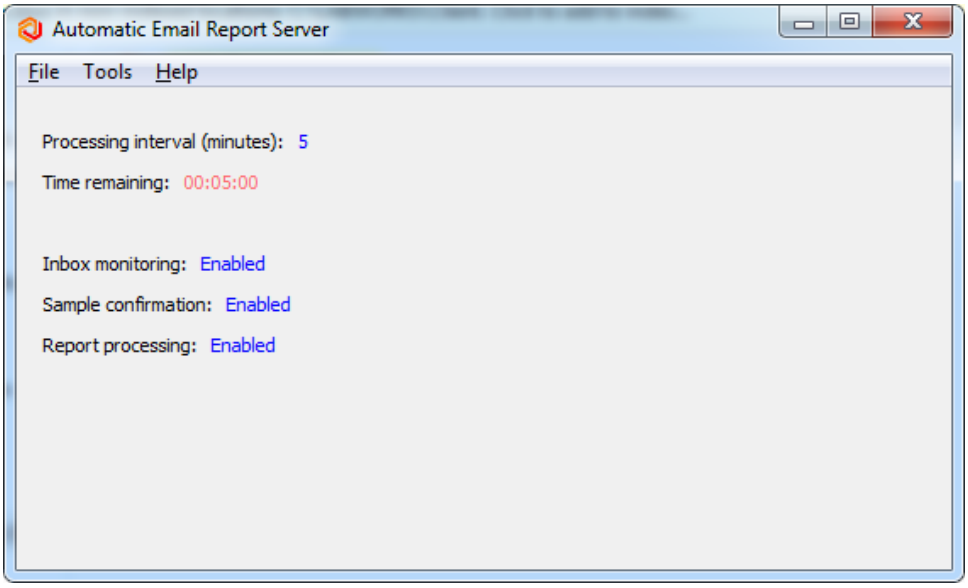
3. Click Privileges.
4. Enable or disable email reporting for each user by selecting the user and then checking or unchecking the "**Run automatic email reporting**" privilege.



Configure System Manager Settings for AER_DAEMON Table

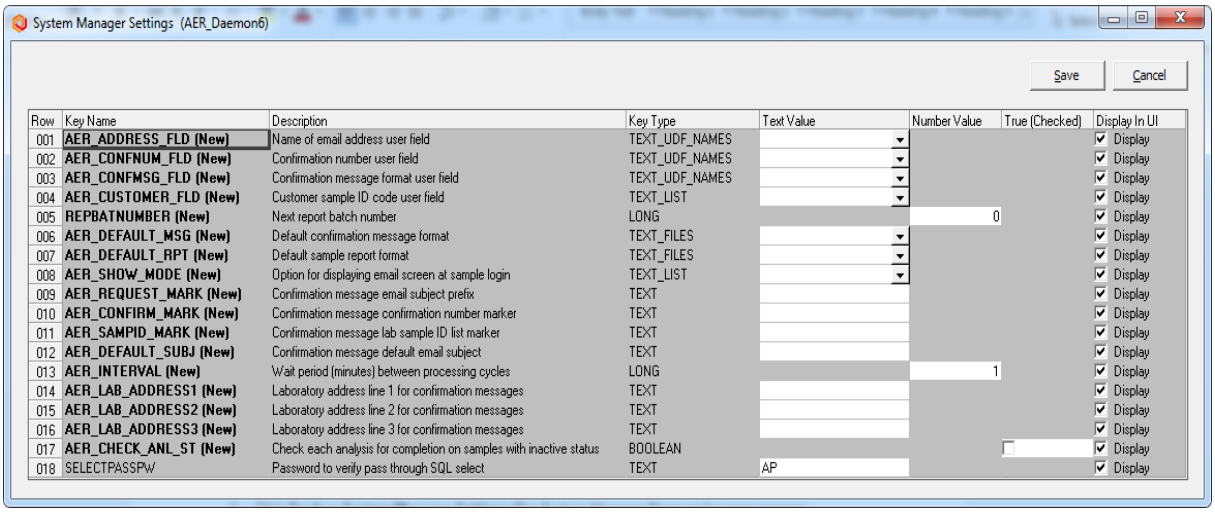
You are now ready to map the user defined fields you added to the USERFLDS and SUSERFLDS tables to the System Manager Settings for Automatic Email Reporting.

1. Open the LWEXE directory and click on **AER_DAEMON6.exe**. The automatic Email Report Server opens.



2. Click **Tools > System Manager Settings**.The System Manager Password screen appears.
3. Enter the System Manager password.The System Manager Settings (AER_Daemon) screen appears.

When you access this screen for the first time the **Key Names** will appear in bold and next to the name will be **(New)**. When you are defining these settings you need to associate a user defined field (listed in the **Text Value** drop down menus) with a Key Name so that the LABWORKS application displays the user defined field as a representation the actual Key Name.



Row	Key Name	Description	Key Type	Text Value	Number Value	True (Checked)	Display In UI
001	**AER_ADDRESS_FLD (New)**	Name of email address user field	TEXT_UDF_NAMES				Display
002	**AER_CONFNUM_FLD (New)**	Confirmation number user field	TEXT_UDF_NAMES				Display
003	**AER_CONFMSG_FLD (New)**	Confirmation message format user field	TEXT_UDF_NAMES				Display
004	**AER_CUSTOMER_FLD (New)**	Customer sample ID code user field	TEXT_LIST				Display
005	**REPBATNUMBER (New)**	Next report batch number	LONG		0		Display
006	**AER_DEFAULT_MSG (New)**	Default confirmation message format	TEXT_FILES				Display
007	**AER_DEFAULT_RPT (New)**	Default sample report format	TEXT_FILES				Display
008	**AER_SHOW_MODE (New)**	Option for displaying email screen at sample login	TEXT_LIST				Display
009	**AER_REQUEST_MARK (New)**	Confirmation message email subject prefix	TEXT				Display
010	**AER_CONFIRM_MARK (New)**	Confirmation message confirmation number marker	TEXT				Display
011	**AER_SAMPID_MARK (New)**	Confirmation message lab sample ID list marker	TEXT				Display
012	**AER_DEFAULT_SUBJ (New)**	Confirmation message default email subject	TEXT				Display
013	**AER_INTERVAL (New)**	Wait period (minutes) between processing cycles	LONG		1		Display
014	**AER_LAB_ADDRESS1 (New)**	Laboratory address line 1 for confirmation messages	TEXT				Display
015	**AER_LAB_ADDRESS2 (New)**	Laboratory address line 2 for confirmation messages	TEXT				Display
016	**AER_LAB_ADDRESS3 (New)**	Laboratory address line 3 for confirmation messages	TEXT				Display
017	**AER_CHECK_ANL_ST (New)**	Check each analysis for completion on samples with inactive status	BOOLEAN				Display
018	SELECTPASSPW	Password to verify pass through SQL select	TEXT	AP			Display

Below is an explanation of the System Manager Settings:

AER_ADDRESS_FLD: From the Text Value drop down menu select a user defined field for the name of the email address user field.

AER_CONFNUM_FLD: From the Text Value drop down menu select a user defined field for the name of the confirmation number user field.

AER_CONFMSG_FLD: From the Text Value drop down menu select a user defined field for the name of the confirmation message user field.

AER_CUSTOMER_FLD: From the Text Value drop down menu select the appropriate label for sample designation. Acceptable values are as follows:

SIDN: Lab Sample ID field LCOD: Location code DSCR: Ssample description PORD: Purchase order PROJ: Project code

SCOL: Sample collector

USnn: User defined field #nn

REPBATNUMBER: In the Number Value cell enter the batch number to assign to the next set of generated automatic email reports.

AER_DEFAULT_MSG: From the Text Value drop down menu select a default template for confirmation messages. These templates are located in the LABWORKS Data Directory > Crystal > Email Templates. If you wish to create your own template, then both the crystal .rpt and the .cef file for each template must exist and be stored in the Email Templates folder. If both the .rpt and .cef files are present, the report template name will appear in this drop down list.

AER_DEFAULT_RPT: From the Text Value drop down menu select a default Crystal Report template for completed samples. These templates are located in the LABWORKS Data Directory > Crystal. If you wish to create your own template, then both the crystal .rpt and the .cef file for each template must exist and be stored in the Email Templates folder. If both the .rpt and .cef files are present, the report template name will appear in this drop down list.

AER_SHOW_MODE: From the Text Value drop down menu select whether or not the Automatic Email Reproting post login user program shows itself. Your options are:

ALL Show for all login batches

AERTYPE Show for any samples with a nonnull entry in theAER_CONFMSG_FLD field

AER_REQUEST_MARK: In the Text Value cell enter in the text that you wish to appear in subject line for report request emails. Defaults to **<LAB CONFIRM>.**

AER_CONFIRM_MARK: In the Text Value cell enter the text that will preceed the confirmation number. Defaults to **CONFIRMATION NUMBER:**

AER_SAMPID_MARK: In the Text Value cell enter the text that will preceed the LABWORKS sample ID list. Defaults to **LAB SAMPLES:**

AER_DEFAULT_SUBJ: In the Text Value cell enter the default text for the subject line of a confirmation message email.

AER_INTERVAL: In the Number Value cell enter the number of wait minutes between daemon processing cycles. Valid range is 1 –1440

AER_LAB_ADDRESS1: In the Text Value cell enter your laboratory address line 1 for confirmation messages.

AER_LAB_ADDRESS2: In the Text Value cell enter your laboratory address line 2 for confirmation messages.

AER_LAB_ADDRESS3: In the Text Value cell enter your laboratory address line 3 for confirmation messages, Text

AER_CHECK_ANL_ST: In the True cell for AER_CHECK_ANL_ST, check the checkbox to check the status of each analysis assigned to samples whose sample status indicates that they are ready to report,. Uncheck the True cell for AER_CHECK_ANL_ST, to check only the sample status. This setting is used to prevent reporting of samples marked inactive by the Process Scheduler even though their analyses are all still pending.

Once you configure the settings listed on this screen and click **Save**, these keys will be added to the System Manager database table.

Configure Multi-Sample Login Template

In order to use the Automatic Email Reporting tool you must create a Multi- Sample login template that includes the following fields in the template:

- AER_EMAIL_ADDRESS
- AER_CONF_MSG

You can also add the following optional fields to the template; however, these fields should be treated as read-only.

- AER_CONF_NUMB
- AER_SEND_WHEN

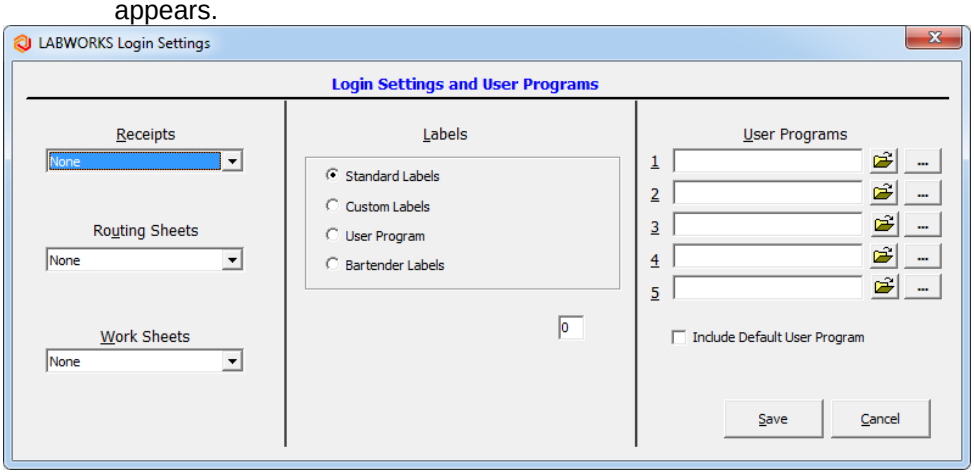
In addition you configure the Multi-Sample Login Template to run the AER_Login program after samples have been logged in. The following section describes how to configure the login settings associated with the Multi-Sample Login Template you just created.

Configure User Program

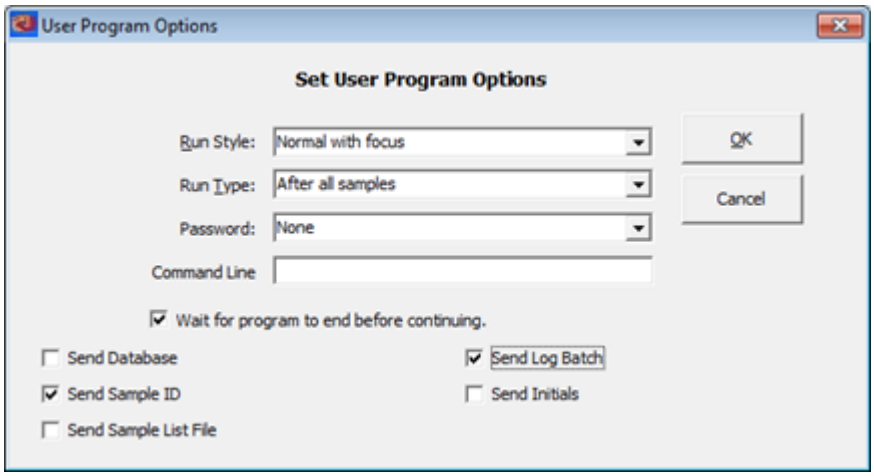
The Automatic Email login program (AER_Login6.exe), needs to be executed as a user defined program after samples are logged in. Therefore, the multi- sample login template that you are using for automatic email reporting must be configured to run AER_Login after all samples have been logged in.

To configure your Multi-Sample Login Template for Automatic Email Reporting:

1. From the Multi-Sample Login screen select Config > Configure Login Settings. The Login Settings screen appears.



2. Click  located at the end of the first User Programs field to select the executable program. The Find User Program window appears.
3. Click on AER_Login6.exe, and then click Open.
4. Click  located at the end of the **User Programs** field that now contains the AER_Login6.exe path.The User Program Options window appears.

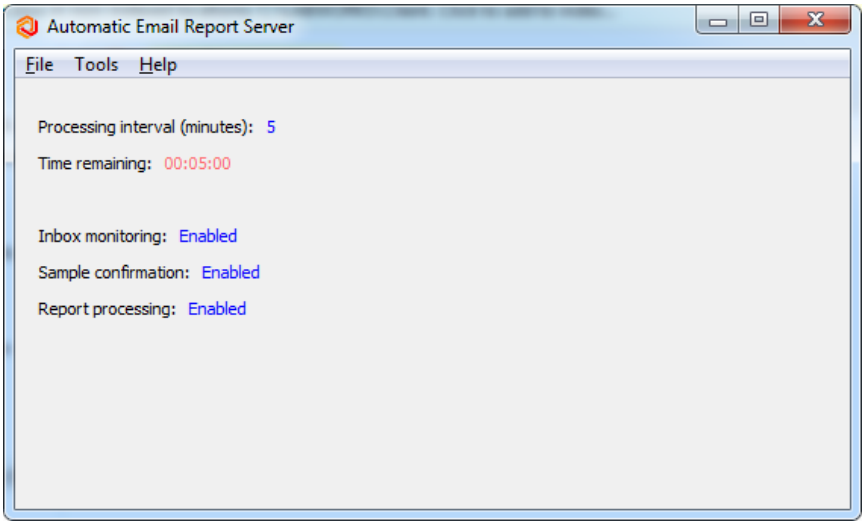


5. Select **Normal with focus** from the **Run Style** drop down window.
6. Select **After all samples** from the **Run Type** drop down menu.
7. Select **None** from the Password drop down menu.
8. Check the following options:
   - Wait for program to end before continuing.
   - Send Sample ID
   - Send Log Batch
9. Click **OK**.You are returned to the LABWORKS Login Settings window.
10. Click **Save**.

Using Automatic Email Reporting

Tell me about the Automatic Email Report Server window

The Automatic Email Report Server is an application that essentially runs in the background and it is responsible for picking up entries in the message queue and processing them and examining the inbox for replies to email confirmations. You can access this program from the LABWORKS executable directory by clicking on AER_DAEMON6.exe.

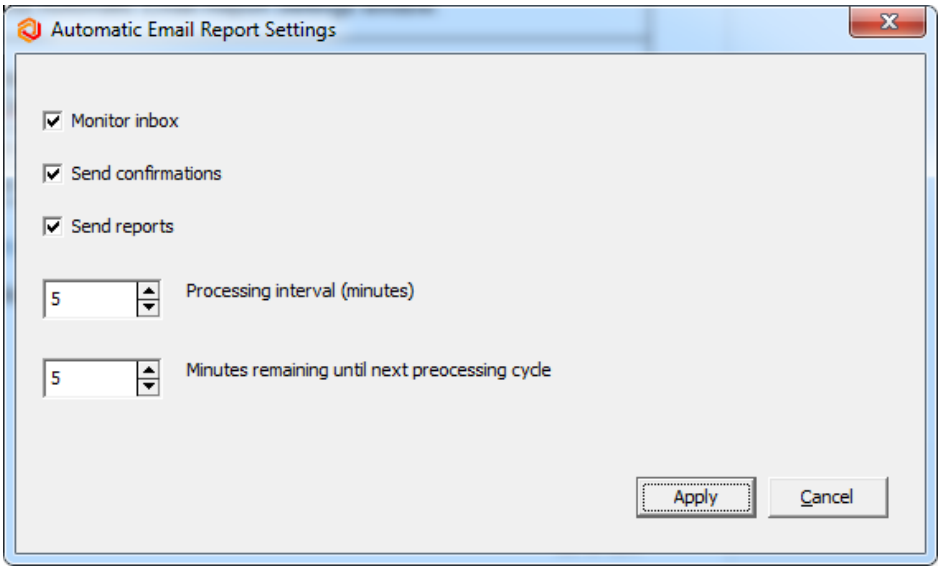


Tell me about the menu commands

Menu	Command	Description
File	**Process Tasks Now**	Select to have the Automatic Email Reporting Server program immediately process the inbox for replies to requests and send out confirmation messages or sample results.
	Pause/Resume	Select to pause the Automatic Email Reporting Server program. The program goes into “sleep mode” If the program is Paused, click Resume to resume processing.
	Exit	Closes the application.
Tools	**Temporary Settings**	Opens the Automatic Email Report Settings window.
	System Manager Settings	Opens the System Manager Settings (AER_Daemon) window.
Help	**Quick Help**	Displays context sensitive Help.
	About	Displays the LABWORKS About box.

How do I customize the Automatic Email Report settings?

1. From the Automatic Email Report Server click **Tools > Temporary Settings**.The Automatic Email Report Settings dialog appears:

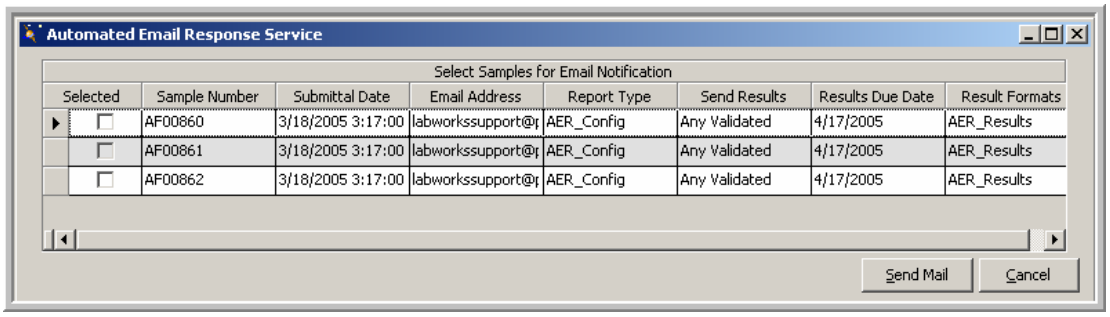


2. Check **Monitor inbox** to enable Inbox monitoring.
3. Check **Send confirmations** to enable Sample confirmation
4. Check **Send reports** to enable Report processing.
5. Select the amount of time the application spends going through the message queue and inbox from the **Processing interval (minutes)** spin box.
6. Select the amount of time that the application sleeps before resuming the tasks of message queue monitoring and inbox monitoring from the **Minutes remaining until next processing cycle** spin box. The default/minimum is 1 minute, maximum is 1440 minutes.

How do I use automatic email reporting?

1. From the LABWORKS Desktop select **Multi-Sample Login**. The Multi-Sample Login Template screen appears.
2. Select the Multi-Sample Login Template Configured for Automatic Email Reporting. The Multi-Sample Login screen appears.
3. Select Location Codes and fill out the Multi-Sample Login Template. Click **Login**.

Once you have logged in samples you are presented with the Automated Email Response Service dialog. This dialog lists the sample numbers, email address (if available), sample login date, types of confirmation messages available, and types of results reports to send.



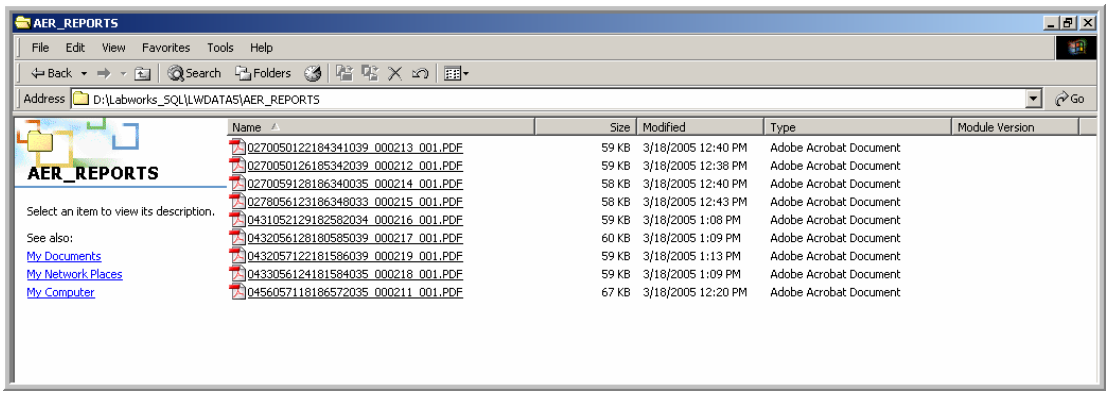
4. From the Automated Email Response service dialog select a **Results Report** template from the Result Formats drop down menu for each sample you wish to email and a Report Type from **the Report Type** drop down menu.
5. Select the samples you wish to email by checking the box next to the sample number.
6. Enter or edit the correct email, confirmation message, and due date for each sample you wish to send.
7. Click Send Mail.
8. A confirmation message is sent to the specified email addresses. This confirmation message includes a unique confirmation number.

How does the recipient get their report?

In order to receive the response, the recipient needs to reply to the confirmation email by simply pressing REPLY and not adding content to the confirmation e- mail. The LABWORKS mailer daemon will check the Outlook inbox for the reply and then parses the email for the confirmation number, sample ids, and email address. This information is then verified against the LABWORKS database to ensure that the message was not tampered with. If the message has not been tampered with, then the LABWORKS Mailer Daemon program creates a new message queue entry for this confirmation number. The daemon then searches the LABWORKS database to see if results are available for these samples. How often results are sent out to the customer depends on the value set in the AER_SEND_WHEN field.

Where are the reports stored after they are created for the lab?

The confirmation reports are stored as .PDF files in the LWUSER directory. The actual report is stored in the LWDATA\AER_REPORTS directory as a PDF document. The PDF document has the file name of the confirmation number, followed by a sequence number of what number the report was generated. This number is seen in the AER Daemon System Manager screen in the REPBATNUMBER field.



The information about when the Confirmation and Final Report was completed can be found in the MESSAGEQUEUE table and the e-mail address that the reports were sent to can be found in the SUSERFLDS table.

MessageQueue Table

MsgNumber	MsgType	Detail1	Detail2	Detail3	DispositionState	DispositionTime	MsgLocked	MsgLockTime	CreateDate	CreateL
49	SEND_CONFIRM	0456057118186572			COMPLETE	3/18/2005 12:00:4	0	3/18/2005 11:58:1	3/18/2005 11:57:4	DRS
50	REPORT_REQUEST	0456057118186572			COMPLETE	3/18/2005 12:21:0	0	3/18/2005 12:20:2	3/18/2005 12:20:2	DRS
51	SEND_CONFIRM	0270050126185342			COMPLETE	3/18/2005 12:35:5	0	3/18/2005 12:34:5	3/18/2005 12:34:2	DRS
52	SEND_CONFIRM	0270059128186342			COMPLETE	3/18/2005 12:36:2	0	3/18/2005 12:35:5	3/18/2005 12:34:2	DRS
53	SEND_CONFIRM	027005012218434			COMPLETE	3/18/2005 12:36:4	0	3/18/2005 12:36:2	3/18/2005 12:34:2	DRS
54	SEND_CONFIRM	0278056123186342			COMPLETE	3/18/2005 12:37:0	0	3/18/2005 12:36:4	3/18/2005 12:34:2	DRS
55	REPORT_REQUEST	0270050126185342			COMPLETE	3/18/2005 12:38:3	0	3/18/2005 12:38:0	3/18/2005 12:38:0	DRS
56	REPORT_REQUEST	027005012218434			COMPLETE	3/18/2005 12:40:2	0	3/18/2005 12:39:4	3/18/2005 12:39:4	DRS
57	REPORT_REQUEST	0270059128186342			COMPLETE	3/18/2005 12:41:0	0	3/18/2005 12:40:2	3/18/2005 12:39:4	DRS
58	REPORT_REQUEST	0278056123186342			COMPLETE	3/18/2005 12:43:4	0	3/18/2005 12:43:1	3/18/2005 12:43:0	DRS
59	SEND_CONFIRM	0432057122181582			COMPLETE	3/18/2005 1:11:29	0	3/18/2005 1:11:06	3/18/2005 12:58:4	DRS
60	SEND_CONFIRM	0433056124181582			COMPLETE	3/18/2005 1:00:05	0	3/18/2005 12:59:3	3/18/2005 12:58:4	DRS
61	SEND_CONFIRM	0432056128180582			COMPLETE	3/18/2005 1:00:34	0	3/18/2005 1:00:05	3/18/2005 12:58:4	DRS
62	SEND_CONFIRM	0431052129182582			COMPLETE	3/18/2005 1:00:51	0	3/18/2005 1:00:34	3/18/2005 12:58:4	DRS
63	REPORT_REQUEST	0431052129182582			COMPLETE	3/18/2005 1:08:55	0	3/18/2005 1:08:03	3/18/2005 1:08:00	DRS
64	REPORT_REQUEST	0432056128180582			COMPLETE	3/18/2005 1:09:38	0	3/18/2005 1:08:55	3/18/2005 1:08:02	DRS
65	REPORT_REQUEST	0433056124181582			COMPLETE	3/18/2005 1:10:06	0	3/18/2005 1:09:38	3/18/2005 1:08:03	DRS
66	REPORT_REQUEST	0432057122181582			COMPLETE	3/18/2005 1:13:20	0	3/18/2005 1:12:47	3/18/2005 1:12:47	DRS

BarTender Label Setup

Installing and Configuring BarTender

BarTender Setup for LABWORKS 6.X

Setting up BarTender labels in LABWORKS requires three main steps:

- Installing and configuring BarTender to operate properly on your system
- Setting up paperwork settings at login to create SAMPLE.DAT
- Creating a BarTender label format and linking it to SAMPLE.DAT The BarTender software program is used to create custom labels within LABWORKS. If your laboratory plans to use BarTender, the BarTender installation manual provides a brief, but to the point, and detailed guide for its installation. However, it is important to note that the Enterprise version of BarTender requires installation of a license server component called the Seagull License Server. This component must be installed to a computer that allows access at all times by the workstations that will be printing labels from LABWORKS.

To install BarTender

Install the Seagull License Server as detailed in the BarTender manual. Install the BarTender component on each workstation that will be printinglabels per the BarTender installation manual instructions.

NOTE: Since this is a third-party component, LABWORKS representatives can only provide limited support for the actual usage of BarTender itself.

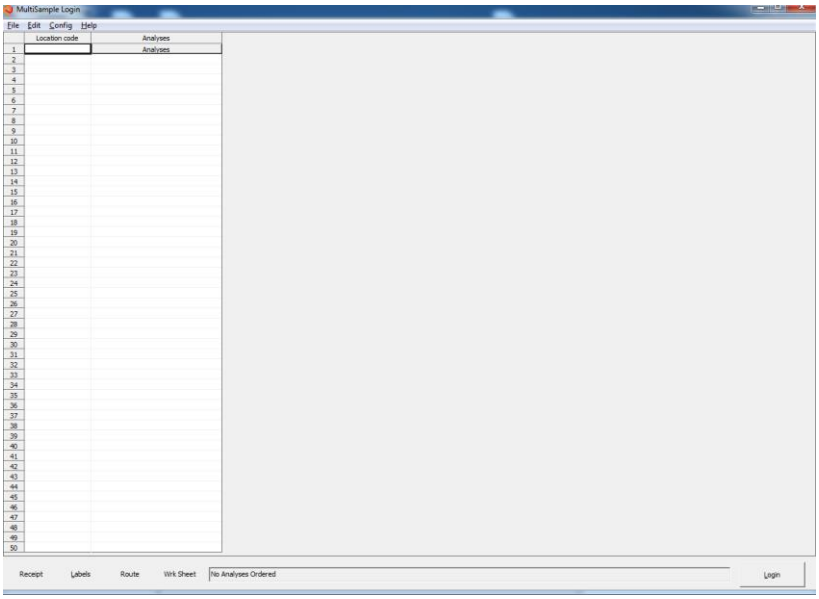
Set up Paperwork Settings at Login

BarTender can be used to create any static label format, but in order to utilize the BarTender software effectively in LABWORKS, BarTender needs to have the data present at each login and updated automatically. To facilitate this, the current version of LABWORKS creates a file called SAMPLE.DAT in the LABWORKS user path each time a group of samples are logged in and the BarTender label option is checked on in the paperwork settings in login.

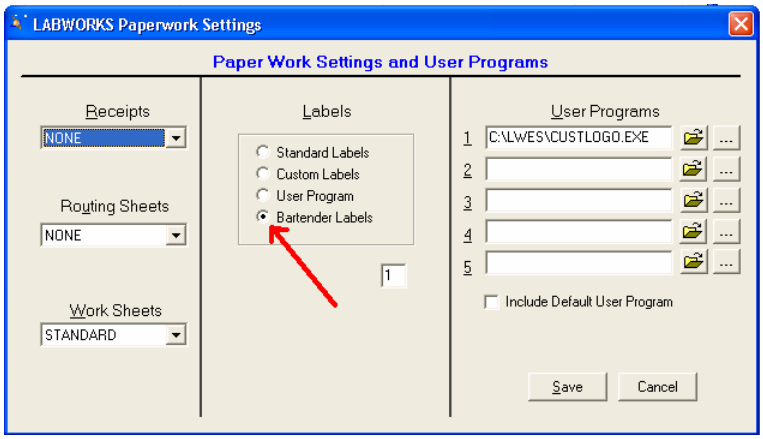
A few steps are needed to create the file SAMPLE.DAT in the LABWORKSuser path:

To create SAMPLE.DAT file

1. Run LABWORKS Multi Sample Login.
2. Click the CONFIG menu on the Multi-sample Login screen.

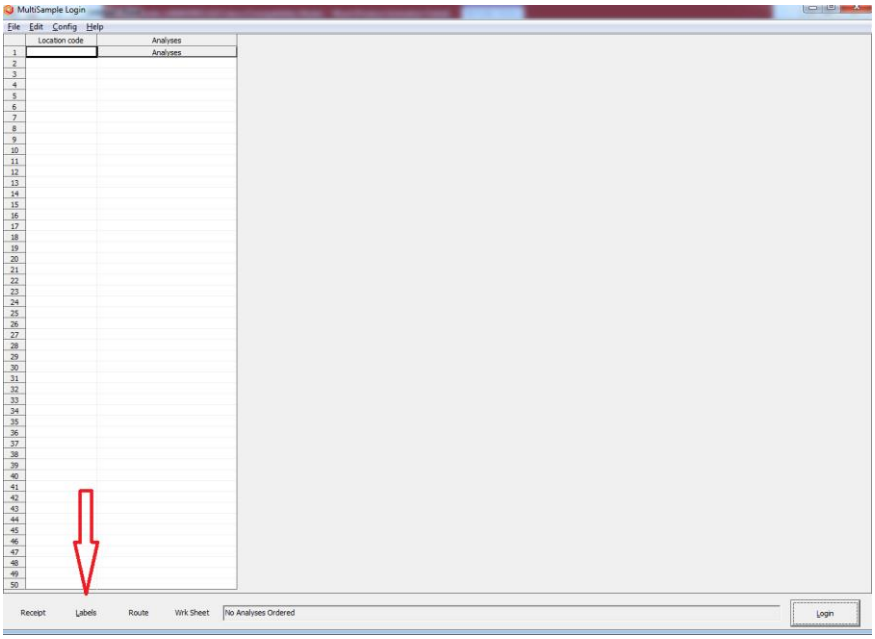


3. Click **Configure Paperwork** in the CONFIG pull-down menu and note the setting to create BarTender labels in the center of the screen.



Make sure the **Bartender Labels** button is clicked and that the number of labels is set to one or more.

4. Click **Save** to save the setting.
5. Locate the labels, receipt, routing sheet, and worksheet buttons at the lower left of the login spreadsheet.
6. Make sure the **Labels** option is selected and “on” as shown in the screen shot.



7. Log in one sample.
8. Retain the sample ID so that it may be deleted later.
9. Look in the LABWORKS user path using Windows Explorer for SAMPLE.DAT file. You should see a file called SAMPLE.DAT in the user path. If the Windows setting for “files of known type” is set to hide file extensions, the file will appear to be only named SAMPLE.

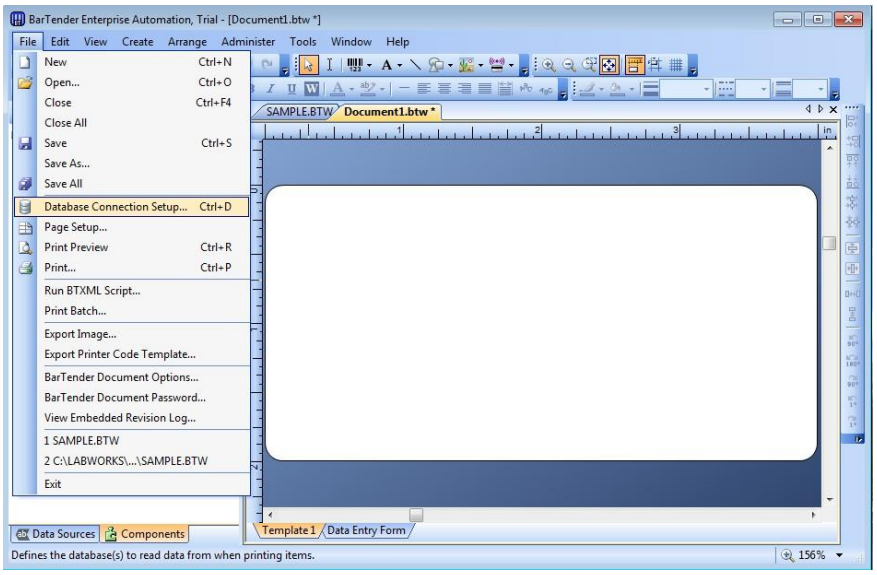
Create BarTender Label Format

Once you have a SAMPLE.DAT file, you then need to create a label format in BarTender that will extract data from SAMPLE.DAT, and print at login automatically. SAMPLE.DAT is a database file containing sample header fields that BarTender can read, and place on labels. The BarTender documentation explains how to link the label format to the database file; however, this section provides the basic steps.

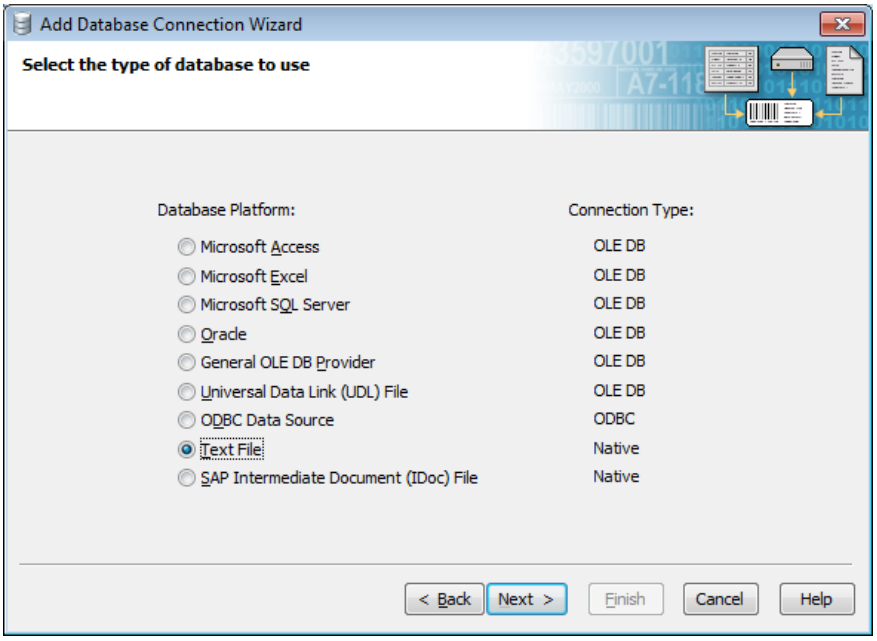
Create a link to SAMPLE.DAT file

To create a link to SAMPLE.DAT file

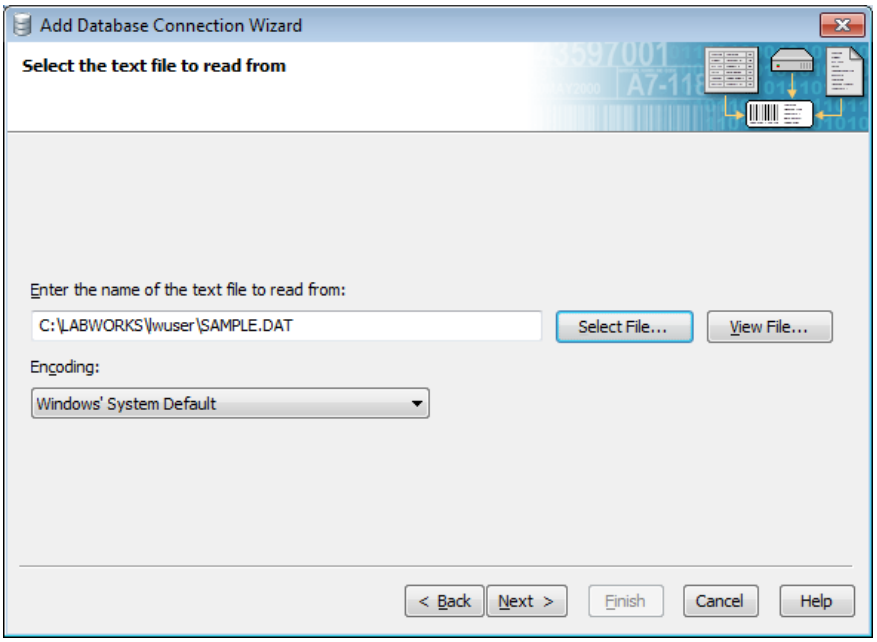
1. Launch BarTender.
2. Click File > Database Setup.



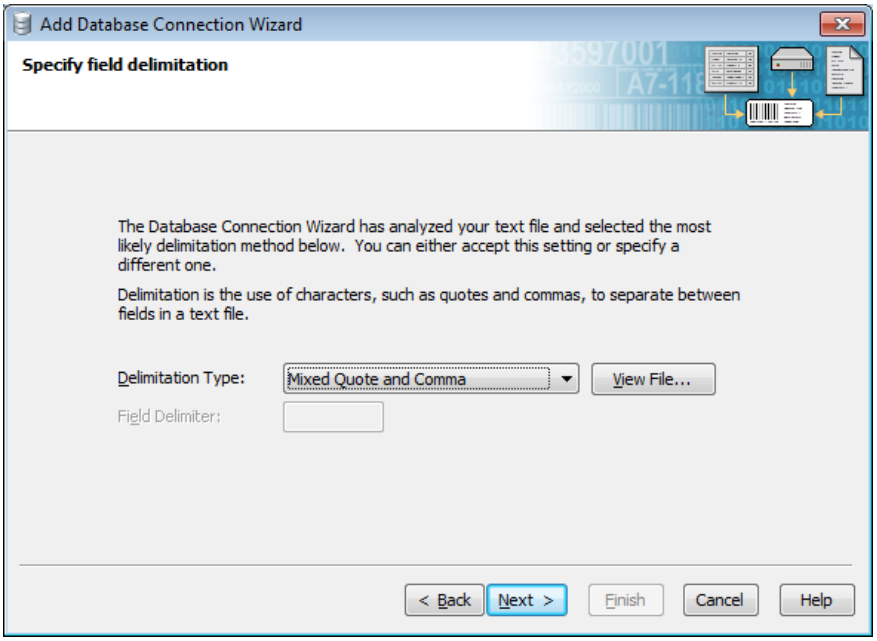
3. Use the database wizard in BarTender to link to SAMPLE.DAT.
4. Make sure you select Text File as the database type on the Add Data base Wizard – Type screen.



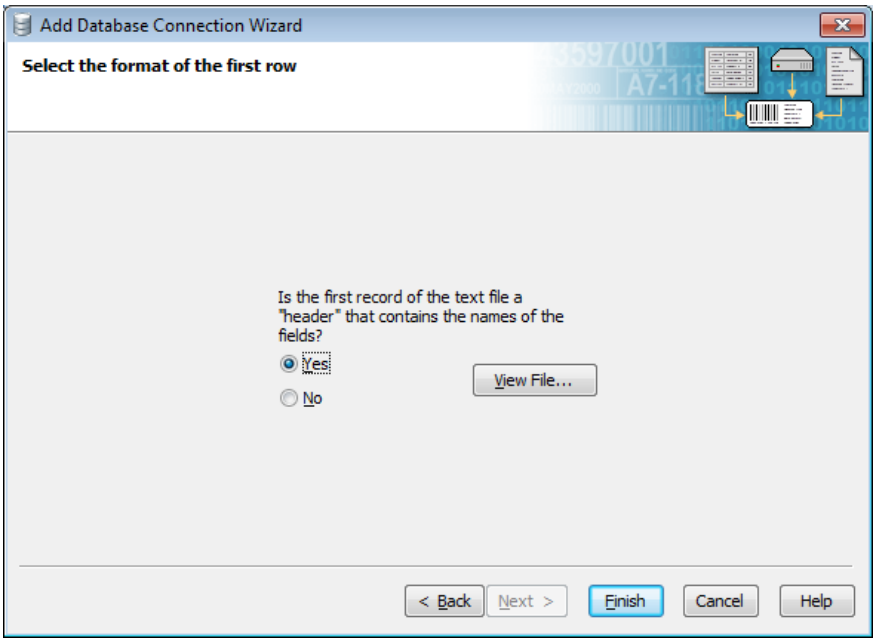
5. Use the browser/file dialog in BarTender to locate your SAMPLE.DAT file.



6. Select **mixed quote and comma** as the delimitation type.



7. Since there are headers in the DAT file, select **Yes** on the Add Database Wizard – Fields screen to indicate that the first record of the text file is a header.

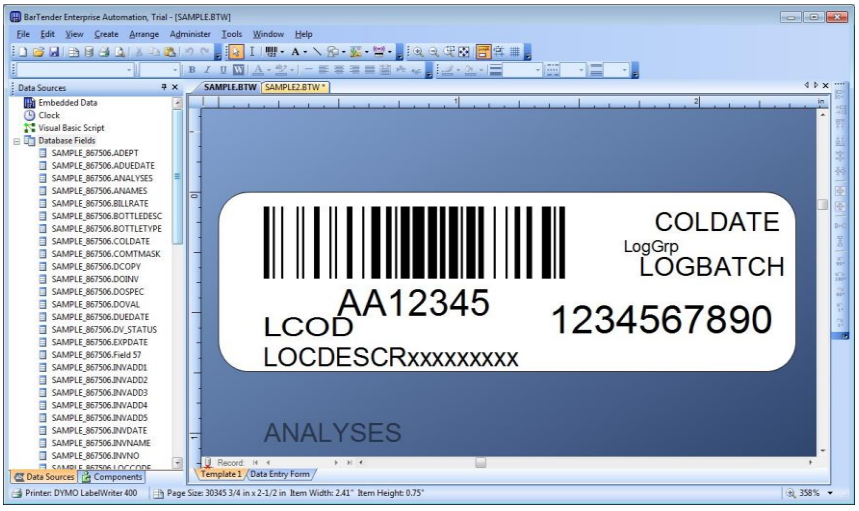


8. Click **Finish**.
9. Click **OK** on the next screen.

You have now added the data in SAMPLE.DAT for use with the BarTender label format. You will see a list of fields available for the label on the left. For a description of these fields, see the LABWORKS LIMS System Administrator Reference – Additional Reference Material appendix at the end of this chapter.

To create label format:

1. Drag and drop the desired fields onto the BarTender formatting screen. Refer to the BarTender documentation for setting up label formats.

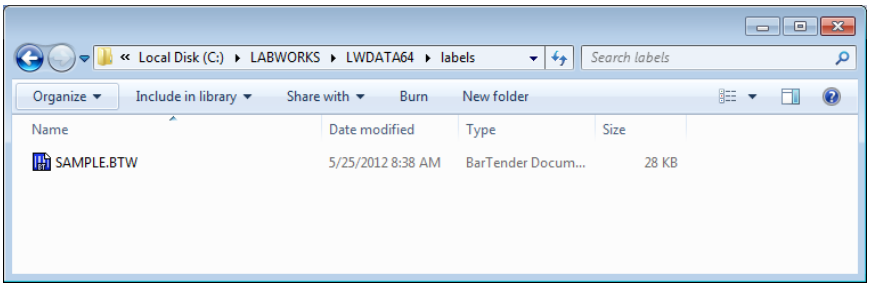


Preview, print, and save format

1. After creating the format, click the **Print Preview** icon on the menu bar to preview the label format; edit as needed.

2. Click the **Print** icon to print the label while editing to make sure it prints correctly on your printer(s).
3. Click the **Save** icon to save the format and save the file as SAMPLE.BTW in your LABWORKS data path in a folder named **Labels**.

If the **Labels** folder is not already present, create it. The example below shows proper naming and placement of the file.



If the preceding instructions are followed correctly, your BarTender label will print the next time samples are logged in to LABWORKS.

Utilities

Setup and Customization

The Laboratory System Administrator (SA) is the individual responsible for setting up and customizing LABWORKS for its intended use. The Laboratory System Administrator uses the Utilities functions to customize LABWORKS as needed to establish system controls; to customize data fields for generating results and reports; and to set up authorized User access information and privileges.

The steps for accessing and using the following Utilities functions are outlined within this section:

System Manager Mode

Configuration

Before customizing LABWORKS information within the Utilities function, it is to the Laboratory System Administrator's advantage to obtain and have on hand certain information and data required for filling in the screen fields. Information to obtain before setup includes:

Next invoice number (if using the Invoicing functionality).

Next sample identification number.

Next sequential Quality Assurance (QA)/Quality Control (QC) batch number for the samples.

Data field name preferences for reports and output.

Names of Users.

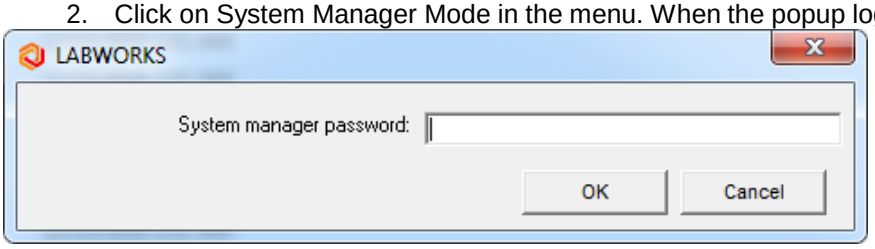
List of or facts about activities performed by Users for setting up User privileges.

System Manager Mode

Access the system manager mode

To access the system manager mode:

1. Click on Utilities on the LABWORKS menu bar.
2. Click on System Manager Mode in the menu. When the popup login window appears, enter a Password.

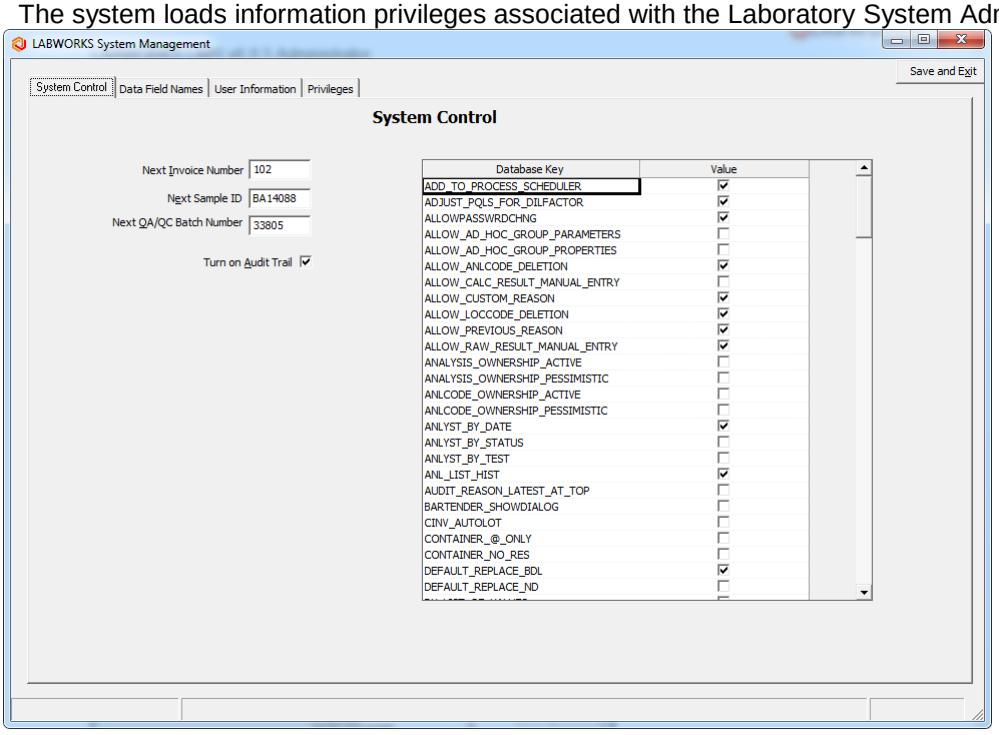


LABWORKS has installed the default Password “MASTER” for demonstration purposes and for use by the Laboratory System Administrator during installation. The Laboratory System Administrator may use the “MASTER” Password for initial set up, but it is recommended that the password be changed at some point if not now.

Note: If you change the Password, write down the new password so it will not be forgotten. If the Password is forgotten or lost, it is impossible to recover it and regain access to the LABWORKS information beyond this point.

3. Press Enter.

The system loads information privileges associated with the Laboratory System Adminstrator status.

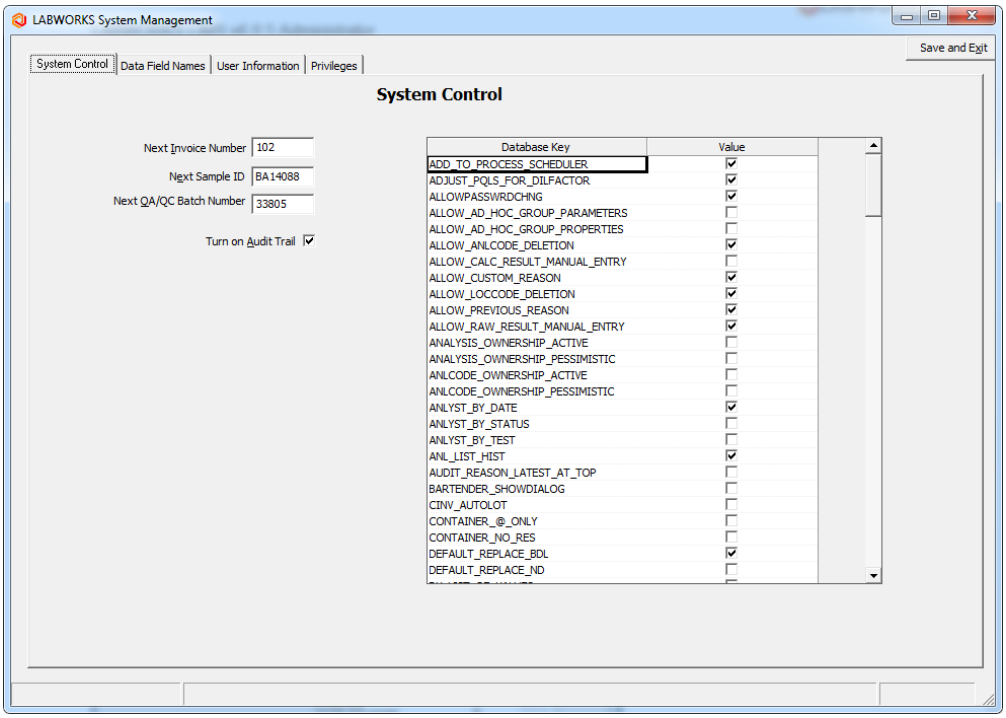


Set up system controls

The System Control function is used by the Laboratory System Administrator to establish the desired controls for the system. It is used to activate the audit trail function; to set up the invoicing number sequence, sample identification number sequence, and Quality Assurance (QA) and Quality Control (QC) batch number sequence; and for changing the master password using the locking/unlocking master password function. System control function is used to set up the initial LABWORKS system settings and to perform maintenance after using LABWORKS for some time.

To set up system controls

1. Click on System Control.



2. If you will be using the Invoice functionality of LABWORKS to send invoices to customers, enter the next invoice number in the Next Invoice Number field.

The Next Invoice Number can be changed to any integer, but cannot be set to a number lower than the current one.This information must be obtained before set up from the person responsible for invoicing. LABWORKS enters a default entry of 101. This number may be changed later but must be changed before invoicing.

3. If you will not be using the Invoice functionality of LABWORKS, ignore the default entry in the Next Invoice Number field.

4. To go to the next field, press the Tab key or click in the field.

5. In the Next Sample ID field, enter the next sequential identification number for the next sample.The LABWORKS Sample ID is preceded with two alpha characters. Initially, the first Sample ID is AA00001. The next Sample ID is AA00002 and so on as the samples are logged in. When the Samples reach AA99999, the next Sample ID becomes AB00001. The alpha character changes after the digit character of the ID reaches 99999, and the digit character increases when samples are logged in. The Next Sample ID can be increased but not decreased. This information must be obtained before set up.

6. In the **Next QA/QC Batch Number** field, enter the next sequential Quality Assurance (QA)/Quality Control (QC) batch number for the samples. The **Next QA/QC Batch Number** can be changed to any integer but cannot be decreased. This information must be obtained before set up.

7. Click (check) the box next to **Turn on Audit Trail** if LABWORKS is to be used to track changes in the system for audit purposes.

Basically, the Audit Trail function is about accountability, e.g., who changed or deleted previously entered data and when these data transactions occurred. This function is particularly valuable if more than one person has write access to make data changes in the system. By turning on this function, the Laboratory System Administrator can determine if data has been deleted, when it was deleted, and by whom. Deletion or revision of data or system information may be necessary for corrective action, such as deleting duplicated sample data entries, or for updating the database to reflect new sample parameters. This function allows the Laboratory System Administrator to track data updates to ensure entered sample data generates the intended results and reports. The intent is to enhance the system by recording sample data revisions and providing the Laboratory System Administrator with a revision history.

Ownership

Samples are logged in based on Location Codes. Location Codes are assigned OWNER values in order to define default OWNERS for Samples. Once items are turned on through System Control, saved, exited, and then returned back to the System Manager screen, a new button appears in the System Control

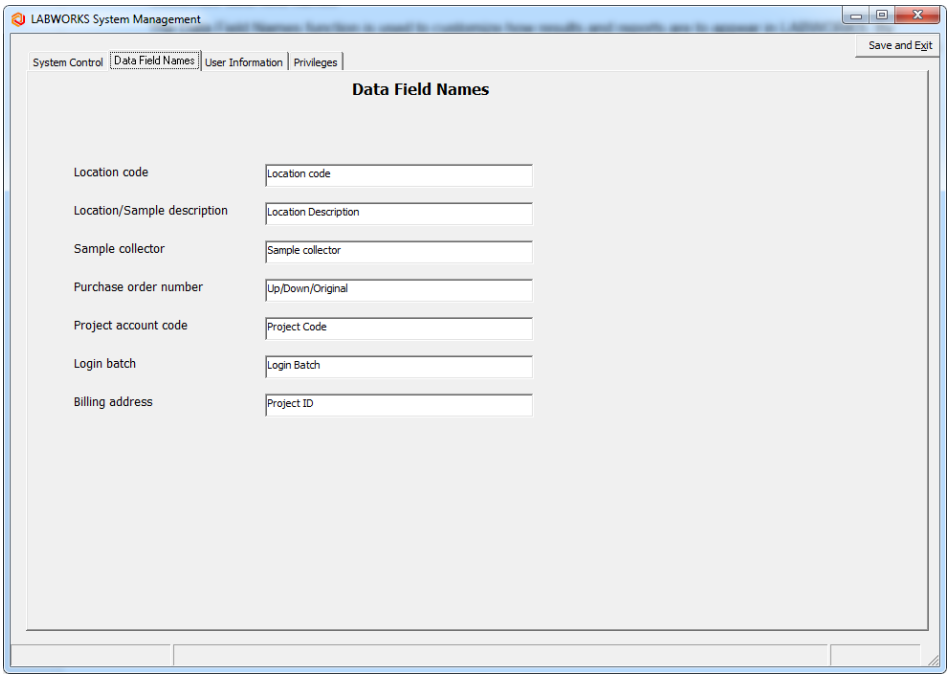
section that allows the Laboratory System Administrator to set up Ownership. See section entitled *Sample Ownership* for a description.

Customize data field names

The Data Field Names function is used to customize how results and reports are to appear in LABWORKS. By changing the information in the LABWORKS standard fields here, the Laboratory System Administrator changes data fields throughout LABWORKS.

To customize field names

1. Click on Field Names.



Notice that the system enters the default **Location code** in the top field and corresponding default entries in the fields below.

Note: As you become more familiar with LABWORKS and the various reports it generates, you may be tempted to further customize these fields later. However, it is recommended that these custom names not be changed often, because doing so may result in confusion, necessary retraining of users, and having to modify reports containing the old field names.

EXAMPLE

Your company uses invoicing rather than billing. Select Billing address from the pull down menu in the top field. Change the default entries to reflect invoicing versus billing, such as:

- For menu field – change Billing address to Invoicing address.
- For text reports field – change Billing to Invoicing.
- For column headings field – change BillAddr to InvAddr.

- Export and report formatting field – change Bill_addr to Inv_addr.

Set up user information

The User Information function is used by the Laboratory System Administrator to identify the system Users, assign User access passwords, and remove Users as needed. The Laboratory System Administrator may define individual Users for establishing individual system privileges or define a group of individuals using the Clone User function. Individuals who basically perform the same tasks and will, therefore, be assigned the same access and User privileges can be “cloned”. The Clone User function allows the Laboratory System Administrator to copy a user’s privileges and other parameters to a new user account.

For instance, if one user has already been defined and a new user is to be created that is nearly identical to the first user, the first user’s account can be “cloned”. This results in a new user account that is setup exactly the same as the first user.

Creating templates for different job functions is suggested to simplify assigning privileges and cloning users. For example, you may create templates for the following positions:

Technician

Chemist

Sr. Chemist

Sample Receiving

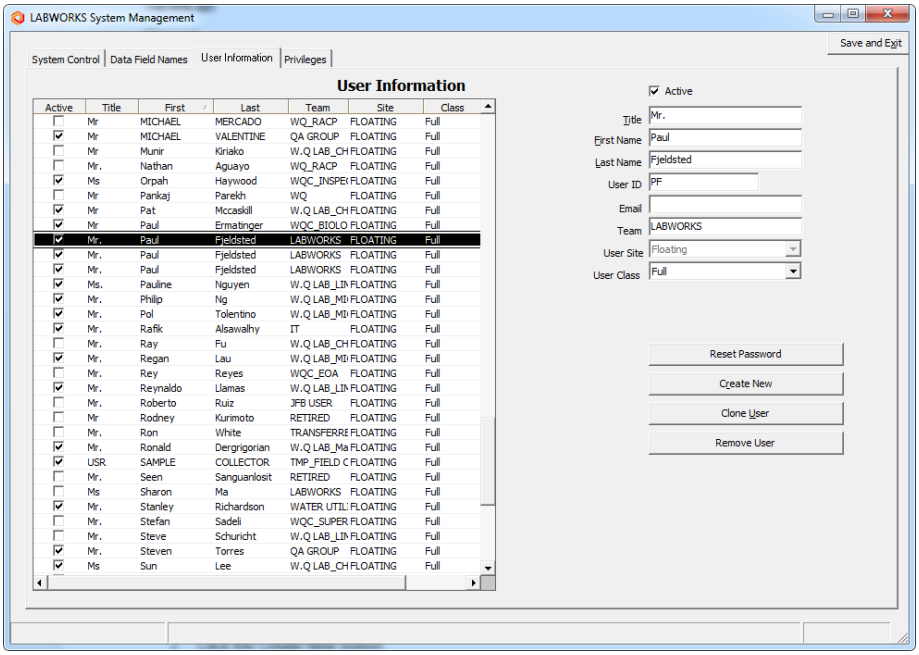
Report Writer

Supervisor

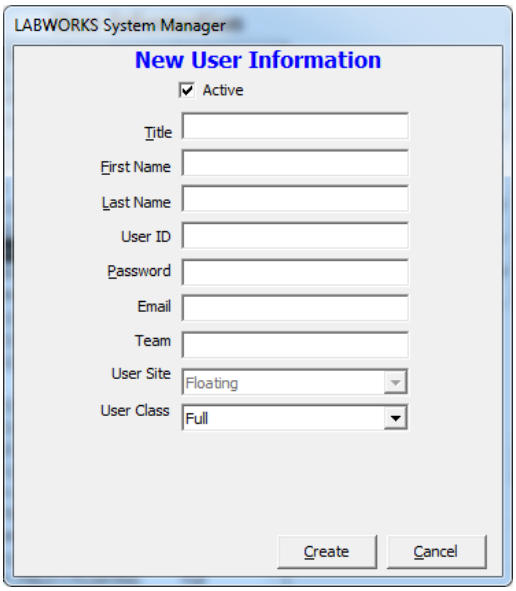
By creating these templates with the associated privileges, you can clone this template for any new employee that fits that job title without having to remember which privileges that job needed.

To create new users

1. Click on the User Information tab.



2. Click the Create New button.



3. Enter the User information in the fields on the New User Information popup window as follows:

Passwords assigned at this point may be changed later by the user.

4. Click Create button to save data.

5. If you do not want to save the data, click **Cancel** to exit without saving.

6. Repeat steps 3 and 4 until User Information has been entered for all Users.

To clone users

The Clone User function is used to define a new account for a User who has the same requirements as a previously defined User. This function is advantageous if you have many Users who all require the same access privileges in the system. You may want to look ahead to the Privileges section of this chapter to view the privileges and determine which Users would be assigned the same privileges. Having that information is beneficial in using this function. See the next section entitled Privileges to help determine the Users who will be assigned the same privileges and can be cloned.

1. In the Users list box, click on the name of the User whose account privileges are to be copied (cloned), and then click the Clone User button.

2. In the New User Information popup window, enter the new User's name and password.

3. Click on the checkboxes for copying the User's privileges to the new "cloned" account.

4. Click Create button to save data.

To remove users

1. Click on the User's name to be removed.

2. Click the Remove button.

3. When the system asks if you want to remove (delete) this User, select YES to permanently delete the User from any access to LABWORKS data.

Privileges

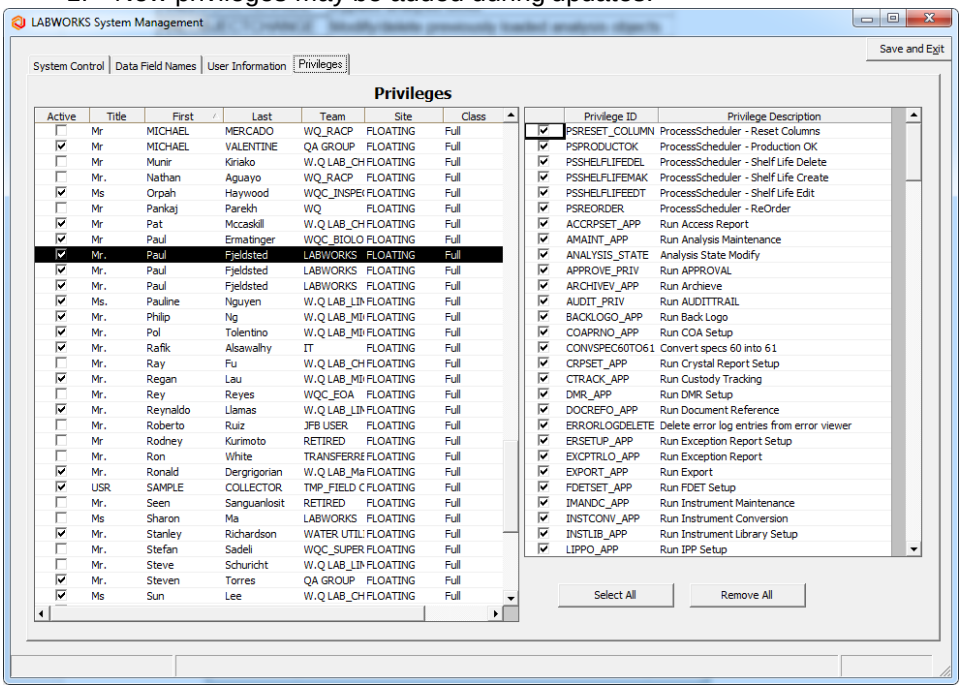
LABWORKS is designed so that each individual user account may be allowed or denied certain privileges within the various LABWORKS program modules.

The Laboratory System Administrator can select among these privileges andtailor individual accounts so that the users are presented with only those privileges that are directly related to their job functions. The Laboratory System Administrator may define individual User privileges or define established privileges using the Clone User function. The Clone User function can save time in set up (see Clone User in the User Information section above). The privileges are located in the PRIVS table of the LABWORKS database, and a listing of those privileges and a brief description of each privilege follows:

PRIVS	
URVALUE	**PRIVDESCR**
AER_RUN	Auto Email Reporting Option
ANALVALIDATE	Validate analysis results
ANLLOCKOVERRIDE	Bypass analysis locks
ANLOBJECTCHANGE	Modify/delete previously loaded analysis objects
ANLOBJECTLOAD	Load analysis objects
ANLOBJECTVIEW	View analysis objects
AUDITRUN	Display or print audit trail
BACKLOGRUN	Generate backlog reports
BARCODERUN	Print barcodes lists (option)
BATCHSETUP	Perform QA/QC batching tasks
CALCSETUP	Setup analysis result calculations
…	…

To assign privileges:

1. New privileges may be added during updates.



2. Click on the Privileges tab.Notice that the Users list contains the names of the Users created by the User Information function.
3. Click on the User's name to select it.
4. If the User, such as the Laboratory System Administrator, requires the ability to perform all the listed tasks in the Privileges list, click the Select All button located below the Privileges list to select all privileges.
5. For individual Users who require limited privileges, click on the box to insert a checkmark in the Privileges field next to the appropriate activities that the User needs to perform his/her work tasks.
6. You may also click on the Select All button to select all the privileges and then click to deselect the activities that do not apply to the User.
7. If several Users require the same privileges, these may be assigned by using the Clone User function (see *Clone User* in the *Set Up User Information* section) or by right clicking and Copy/Paste Prvileges.
8. Assign the privileges associated with these Users by selecting the appropriate activities from the list in the **Privileges** field.

Exiting system manager mode

When you have finished customizing the System Manager Mode functions, click the **Exit** button.

You are returned to the LABWORKS Desktop screen.

If additional changes are needed, click **Utilities > System Manager Mode**; make changes and save to overwrite current settings.

Archive Setup and Configuration

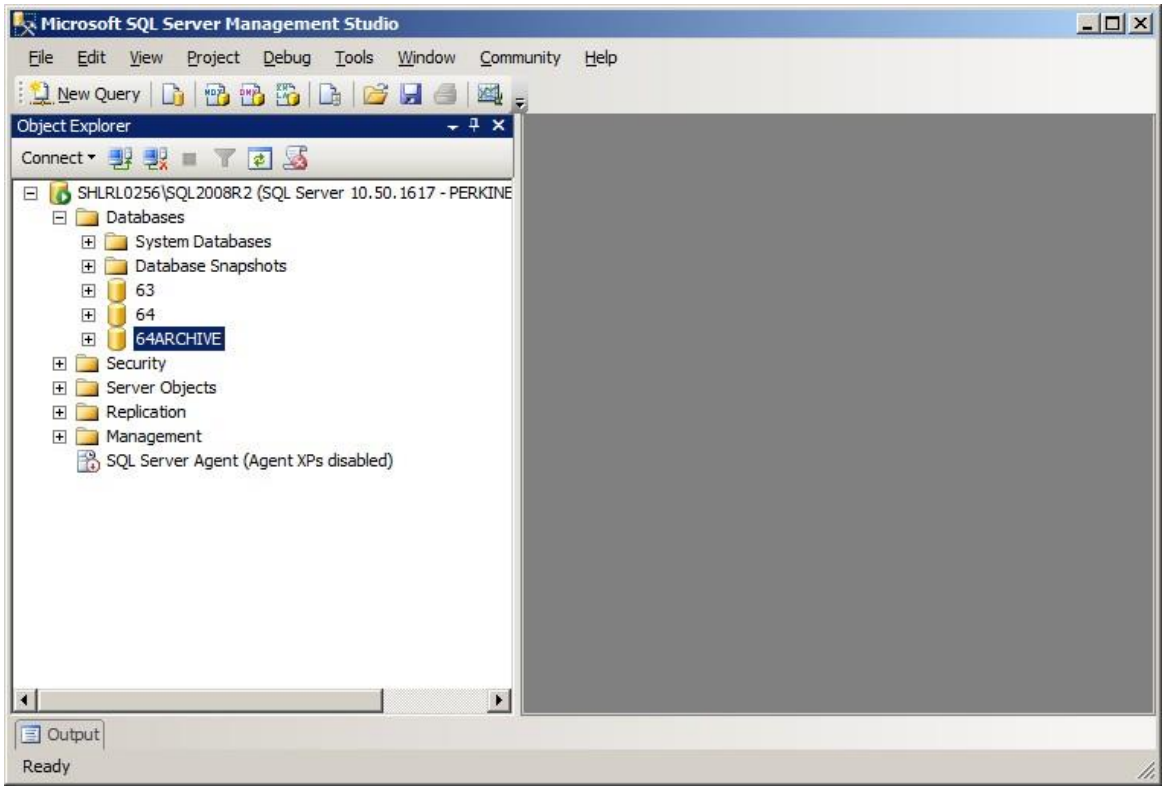
Prerequisites

1. Microsoft .Net framework 4.0 on server and clients
2. LABWORKSServer6.4

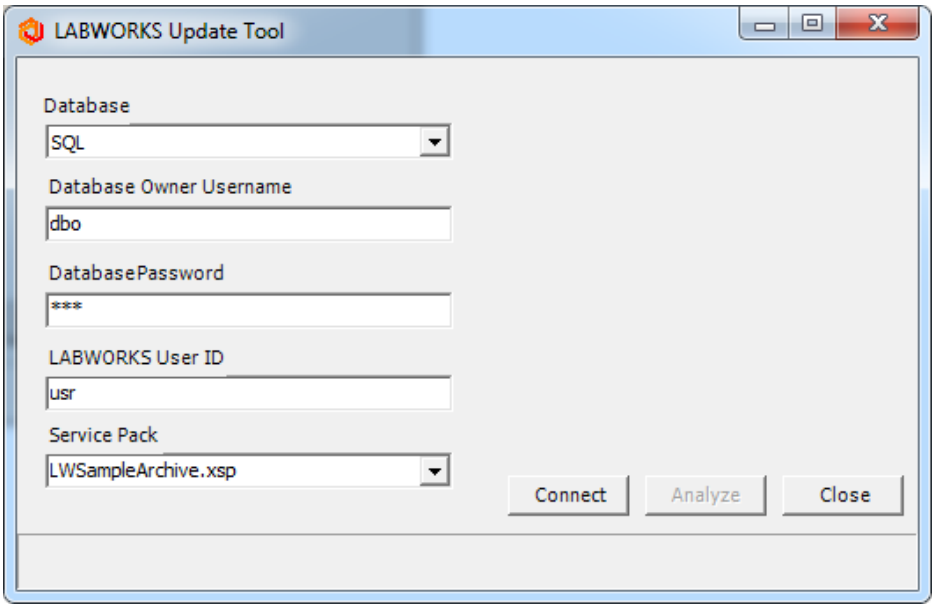
Configuring the Sample Archive Database

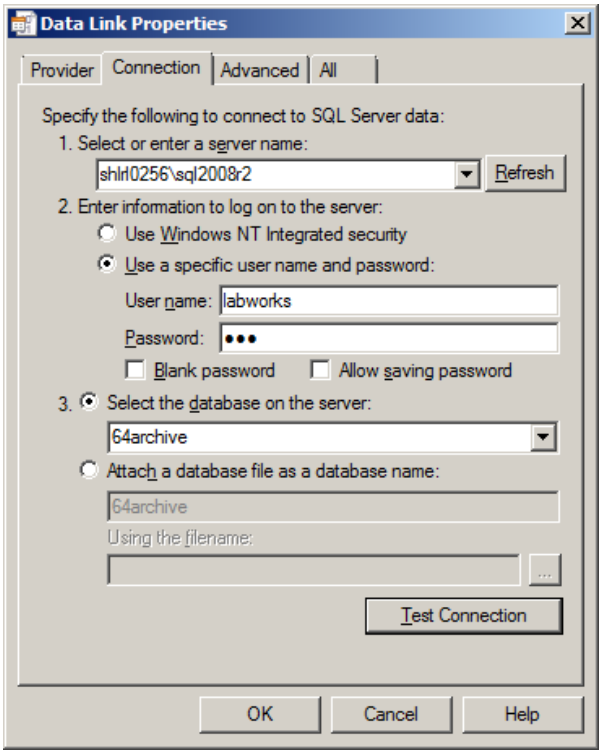
This database will keep a catalog of all sample movement/copy transactions. This database is also where Archive configuration data is stored.

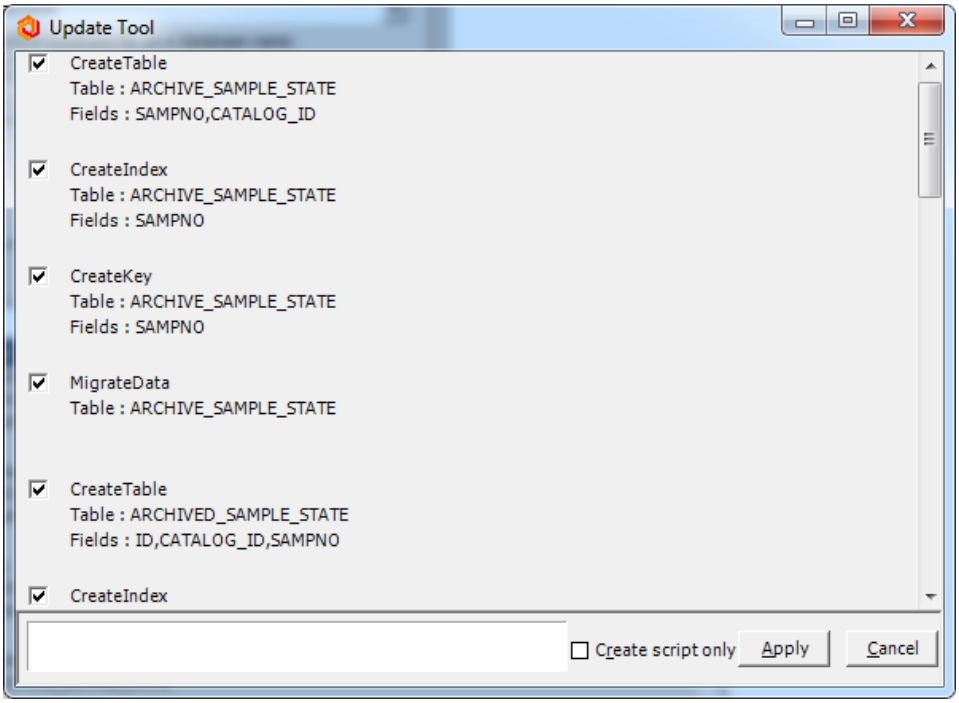
1. Create a blank 6.4 database

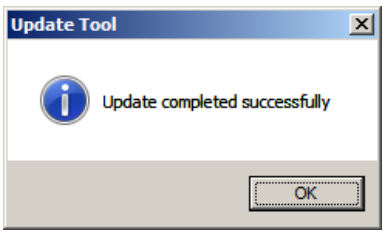


2. Run the UpdateTool and apply the sampleArchive.xsp against the new database.

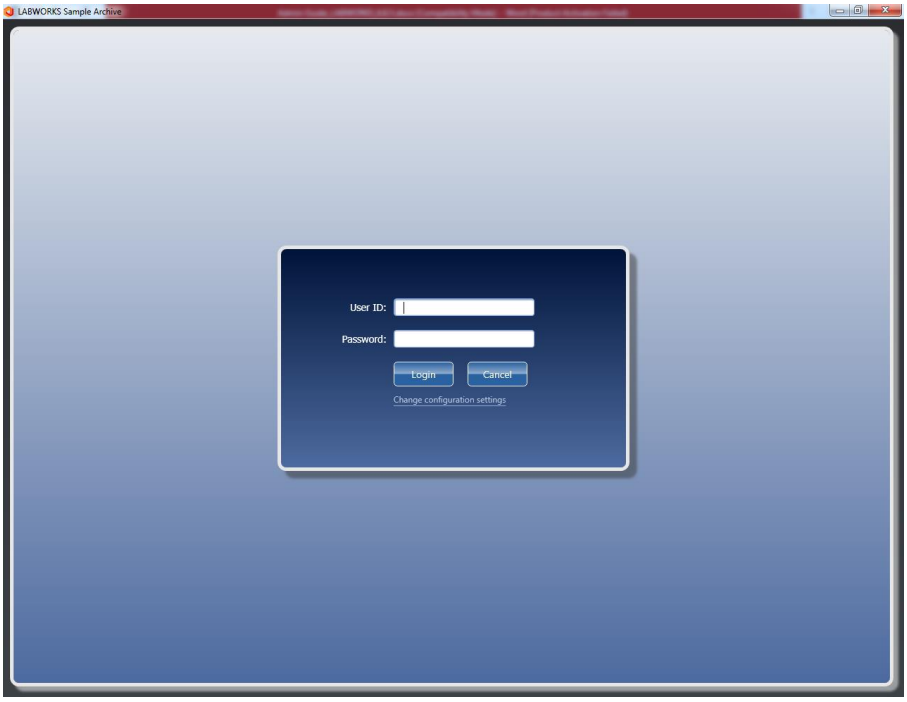




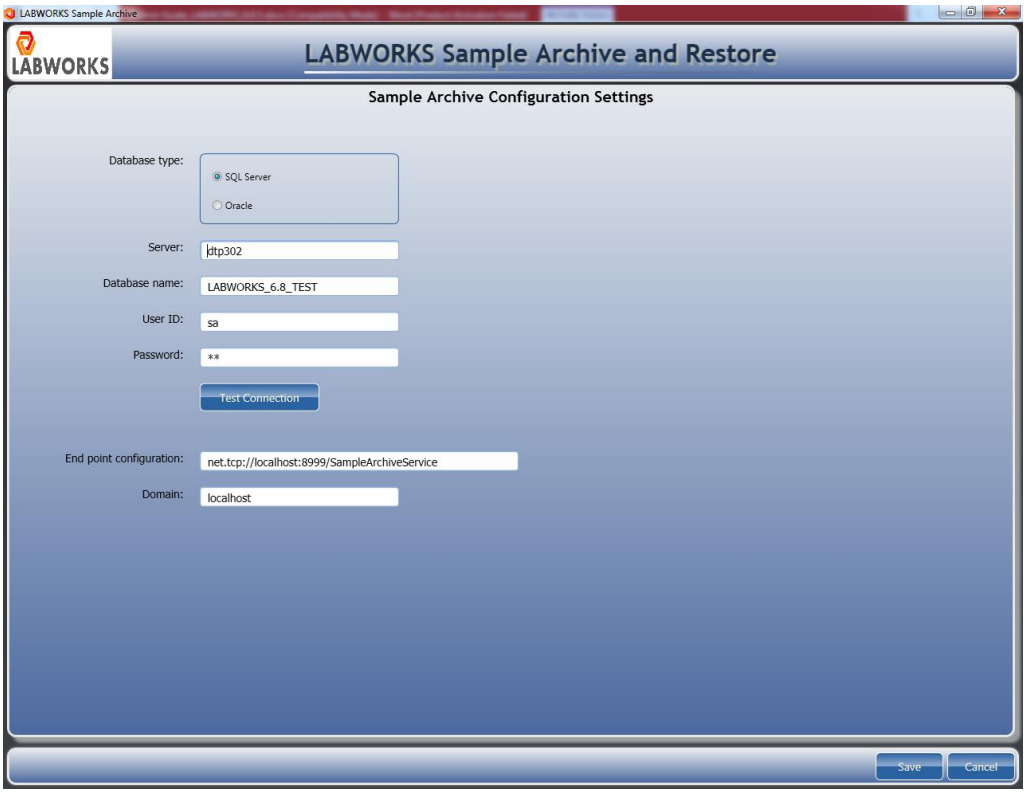




3. Run the LWSampleArchive.exe application found at \\<server install>\Server\LWSampleArchive.exe



Select “Change configuration settings” and enter the details about connection to the archive catalog database.



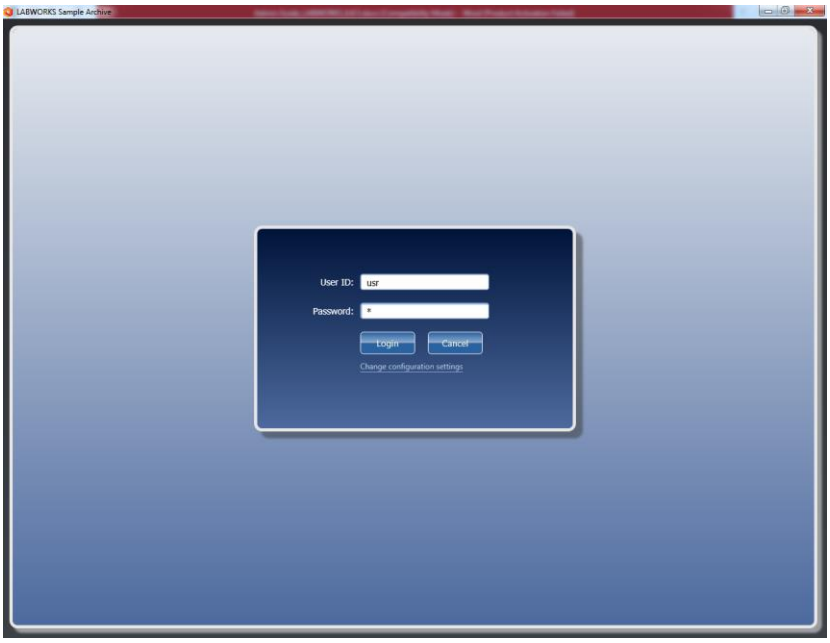
The endpoint address is the connection to the LWDataServices64. In the example, the server name is SHLRL0256.

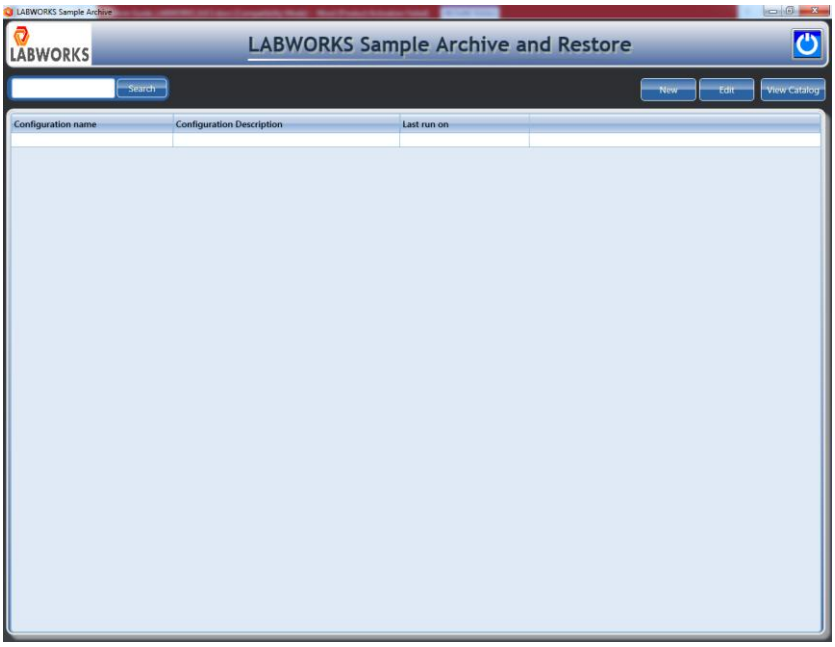
Once the configurations are done, save and close the configuration.

4. Execute the sample archive application once again and the application is ready to use.

Using Sample Archive and Restore

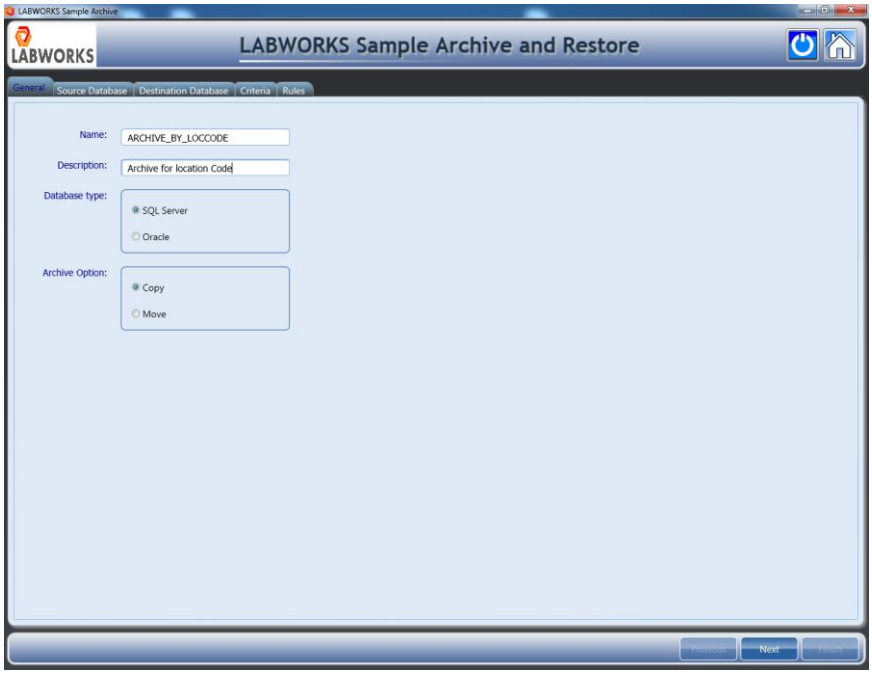
Login to the LWSampleArchive.exe application.





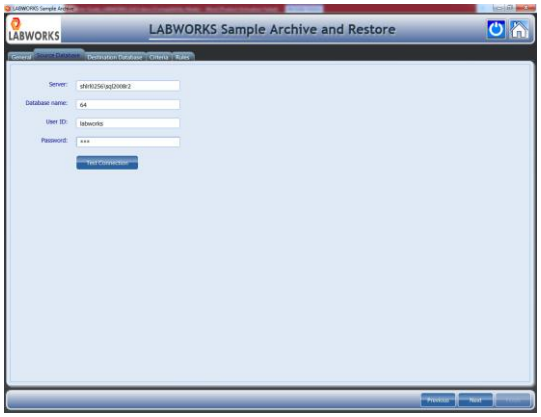
Creating New Archive configurations

1. Enter the archive configuration name and description.
2. Then select the database type and the Archive option.
   a) Copy makes a copy of the records into the destination database.
   b) Move removes the data from the source database and moves to the destination database.
3. Then click ‘Next’.



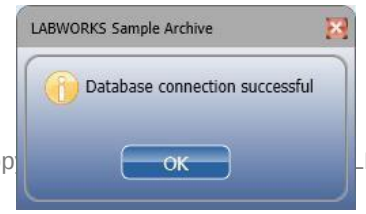
Enter the specifics for the source database.

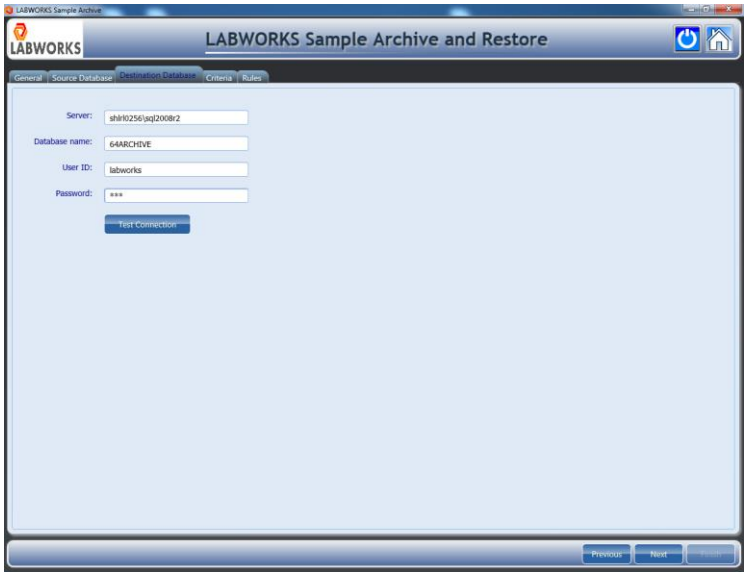
a) Enter the Server name. This is the SQL/Oracle Server name.
b) Enter the database name (actual database name on the database server).
c) Enter the database access credentials, user ID and Password.



4. Then Click ‘Next’.

5. Next, enter the destination database parameters.

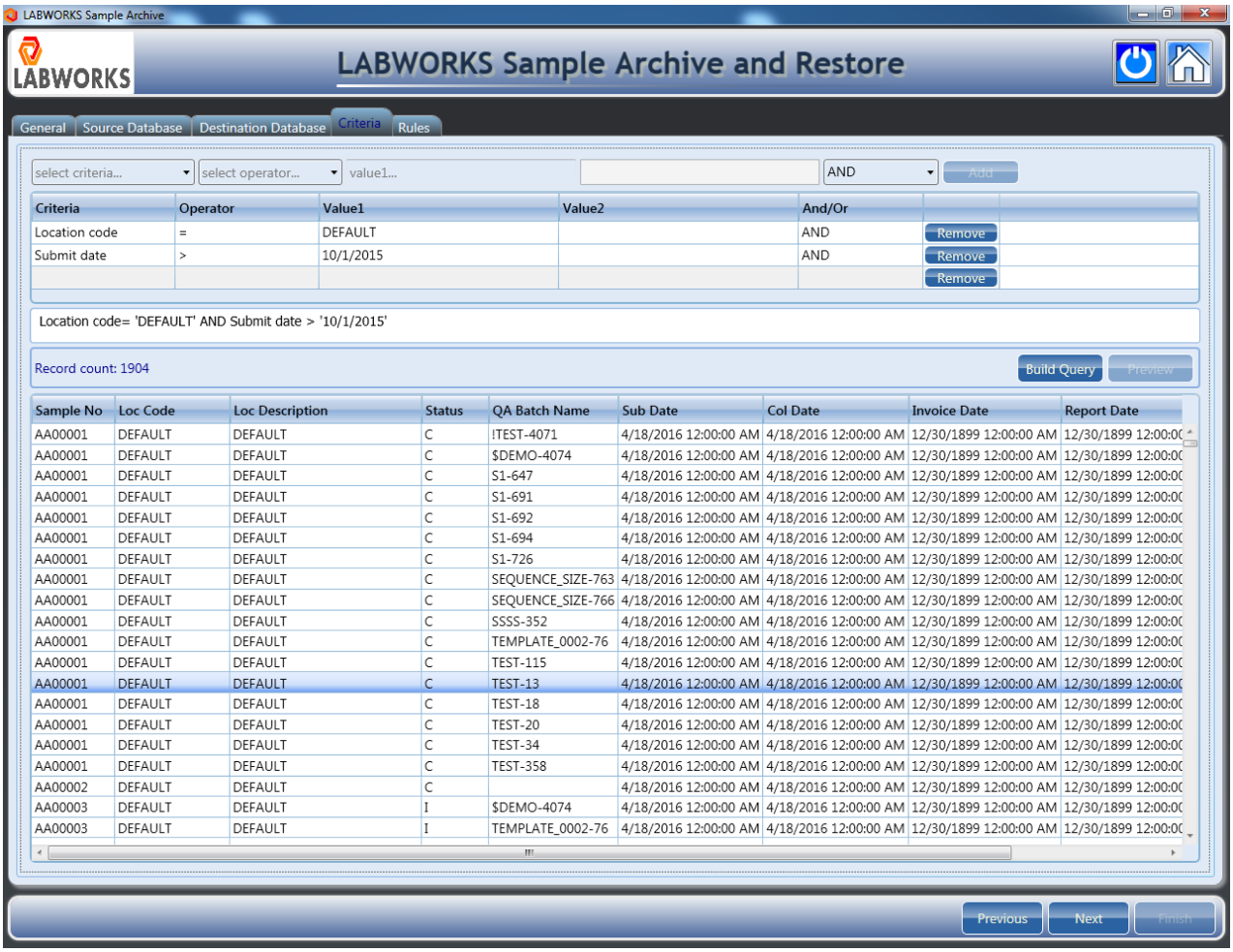


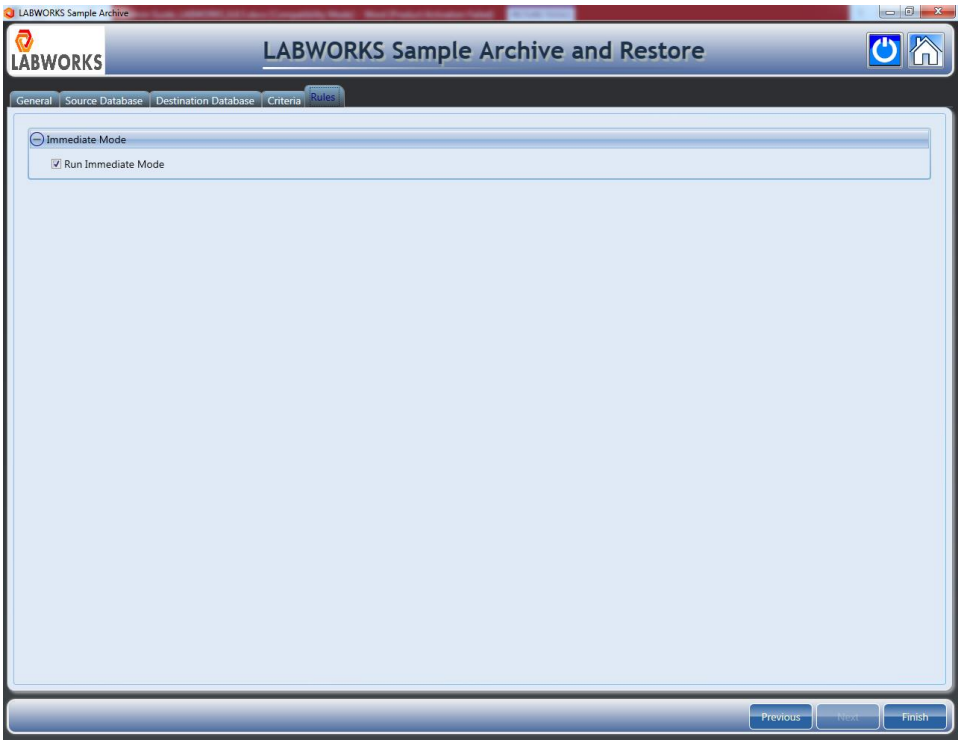


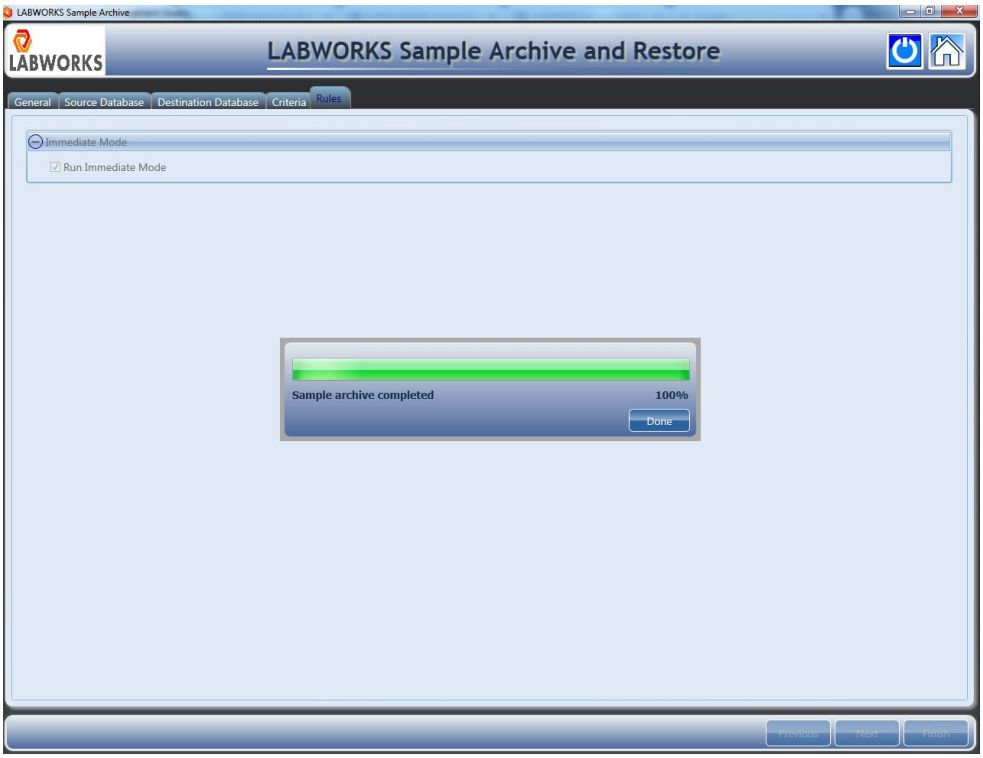
6. Click ‘Next’.

Here you can specify what samples will be included in the archive.

1. Select a criterion.
2. Enter the operator type.
3. Enter the appropriate values.
4. Select AND/OR. (Even if you are only selecting one criteria, you need to select AND/OR). The QueryBuilder will build the correct query.
5. Click ‘Add’.
6. When you have all of the criteria entered, click “Build Query” to see the database query.
7. Then click ‘Preview’ to see the list of samples to be included in the archive set.



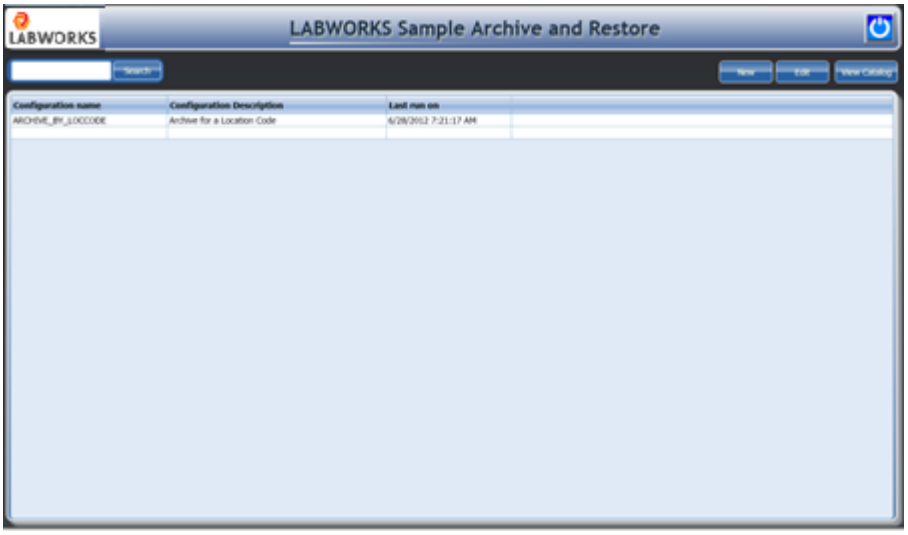




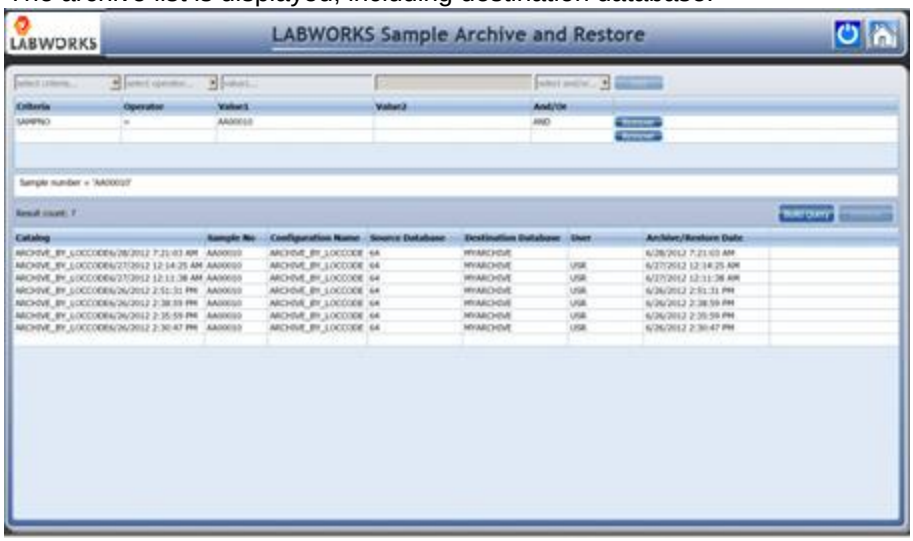
Searching Archive

The Archive catalog can be searched to find archive sets containing particular data.

1. Click ‘View Catalog’.



The archive list is displayed, including destination database.



Rules of Archive Data Movement

Rules for Copy

a) Copies to new destination
The archive utility copies the data from source to destination when there are no matching records exists.

b) If data already exists, what happens?
This is handled through validation in utility as follows:

If data already exists in destination table, the application compares mod date column values in source and destination table for each row. If the value in source table row is found latest then it overwrites data in destination table.

LABWORKS INI File

LABWORKS INI File

LABWORKS employs a text file that contains customizable settings for most LABWORKS programs. This file is the LABWORKS.INI file, and it is located in the LABWORKS program path by default. However, you can also have copies of the whole file, or sections of the file, in either the user paths or the data path. The settings are searched for in the user path first, then the data path, and, if not found in either of those locations, settings are read from the global INI file located in the LABWORKS program path.

Most settings are documented in the INI file itself, but this chapter lists the INI file by section and summarizes each section's function.

Note that many lines begin with the single quote character ('). This indicates to LABWORKS that this line should be ignored; it is a comment line. Lines beginning with a single quote are ignored by LABWORKS and are usually there as an explanation or example line.

The Process Scheduler program should also be configured in the LABWORKS system. For details for configuring and using the Process Scheduler program, refer to the Process Scheduler Manual.

Special Modifications to the INI File

Modifications for Project Analyte Management

There are two modifications that can be made to the LABWORKS INI file to control Project Analyte Management usage. One is required, and one is optional.

To set up required modification

Make the required modification in the LABWORKS.INI file section

[ALOGINO].

Set the key, PROJECT_ANALYTE_GROUP, to one of the following values:

LCOD (location code)

SCOL (sample collector)

PORD (purchase order number) PROJ (project code)

USxx (user defined field where xx is replaced by the two digit field number 01, 02, etc.)

This key specifies which field (whose value is taken from the location list or filled in during login) will be used to hold the name of the PAM group to use for the sample.

During sample login, the specified field will be read.

If it contains the name of a valid PAM group, that group will be applied to the sample.

To set up optional modification

Make the optional modification in the LABWORKS.INI file section

[Spread_Result_Entry] to key ADJUST_PROJECT_REPORT_LIMITS.

Set this key to **Yes** or **No**.

A Yes value indicates that the reporting limits should be multiplied by the dilution factor of the analysis at results entry time.

Modifications for Exception Reports

The Laboratory System Administrator must perform the steps in this section to ensure that the email program works properly if your laboratory plans to send exception reports via email. To accomplish this, you must modify the

LABWORKS.INI file as described in this section.

In order to be able to automatically send exception reports by email, the LABWORKS program needs to know the name or IP address of the mail server and what email address to show as the sender of those reports.

To modify the LABWORKS.INI file

NOTE: These steps are to be completed by the Laboratory System Administrator.

Insert a new section “[Email]” in the LABWORKS.INI file to provide this

information. An example appears below: [EMail]

'specifies the name of the e-mail server in use at the workstation

EMAIL_SERVER = 111.222.333.444

Exception_Report_Sender = lims@laboratory

[EndEMail]

NOTE: In the example above, EMAIL_SERVER is assigned as its value the IP address of the mail server. The name of the mail server could also be used as in this example:

EMAIL_SERVER = MAILMANAGER_01

The exception report sender entry is required by the email services but does not need to be an actual email address. It can be any string containing an internal @ character. In the example above, automatically generated exception reports received via email are labeled as having been sent by lims@laboratory even though such an address does not exist.

If someone is to process the replies to emailed exception reports, set the Exception_Report_Sender to the address where those replies are to be sent.

Modifications for LABWORKS Explorer

Location code description

To display location description in place of location code Set up the following setting in the LABWORKS.INI file: SHOW_LOCDESCR = YES

To set up configuration settings in LABWORKS.INI file

Configure/edit the INI file as follows. [LABWORKS_EXPLORER] SHOW_SQC = YES SHOW_EXCEPTION = YES

‘ Show Location Descriptions instead of Location Codes. This field is not normally indexed in the LABWORKS database. If using this setting, LABWORKS recommends creating an index on the field SAMPLE.LOCDESCR, non unique.

SHOW_LOCDESCR = YES

‘ Show the result validation status, PLV = Pending Laboratory Validation

SHOW_VALID_STATUS = YES [ENDLABWORKS_EXPLORER]

Save the INI file to save your settings.

LABWORKS INI File Settings

MC result MDL

This section is used to determine where the MDL is obtained for multicomponent results that are imported via LABWORKS instrument interfaces.

[MC_Result_MDL]

' specify whether to use instrument or analysis list MDL for result' loading

' options are: ANLIST, INSTR (default), HIGHER, LOWER

' specfy as MDL_ + InstType = Option

' example: MDL_GRF = HIGHER this sets .grf file imports to use the

' higher mdl

MDL_GRF = ANLIST

[EndMC_Result_MDL]

Certificates of Analysis (CERTOFANALYSIS)

This section contains the settings for the optional Certificate of Analysis option for LABWORKS. Default options and custom settings for COA formats and optional COA parameters are set here.

[CertOfAnalysis]

' default settings for certificate of analysis option (DOS version)

' coa_version specify as : NEW or OLD (default is NEW)

COA_VERSION =

'specifies a .cbf for default (LCOD, SCOL, PORD, PROJ, USXX)

'CUST_BASE_DEFAULT=PROJ

'how to build a lot number (NONE / BASF or SCOL / BASF1 / US01 / US02 /default)

' NONE (no lot number)

' SCOL or BASF (use sample collector)

' SIDN (use sample number as lot number)

' US01 (user field 1)

' US02 (user field 2)

' BAS1 (ponumber-project-coldate)

' default (numeric_part_of_sampno-project-coldate)

```
 LOTNUMBER =
' limit coa to 1 lot sample (default NO) or allow multiple lot
' samples (YES)
'print notice of violation on standard format coa (yes / no (default)
PRINT_VIOL_WARNING=NO
'disable printing of coa if violation (yes / no default)
DISABLE_ON_VIOL=NO
MULTISAMPLE =
' review / modify results on screen before printing coa
RESULT_REVIEW =
' specify sample for coa header info if multisample – LOWEST (default) / HIGHEST / SELECT
' HEADER_SAMP = HIGHEST
'is a blank (empty or pending) result for a numeric or non-numeric spec a violation? (Yes / No Default is Yes)
VIOL_IF_BLANK =
'is a non-numeric result for a numeric spec a violation? (Yes / No Default is Yes)
VIOL_IF_NOT_NUMBER =
'preview custom format coa before printing (default = NO)
CUSTOM_PREVIEW =
' pause for envelope printing (hp laser only) (default = NO YES or #)
ENVELOPE =
'printer bin for feeding envelopes (if ENVELOPE = YES)

' 5 if automatic envelope loading is available on printer
' 6 if envelopes are manually fed into printer (default)
ENVELOPE_BIN = 6
'envelope size option
' 20 Envelope #10 (4 1/8 x 9 1/2) default
' 19 Envelope #9 (3 7/8 x 8 7/8)
' ENVELOPE_SIZE = 20
' method reference prefix substitutions for COA
' specify as: METREFx = PREFIX
' example: METREFA = ASTM- (changes "A" to "ASTM-")
METREFA =
```

'parameters control as shipped history display (SQL compliant only)

'number of most recent results points to display default is 10 min is 3 max is 100

HIST_POINTS = 10

'sort order for as shipped history display (ASCENDING / DESCENDING) default is ASCENDING

HIST_SORT_ORDER = ASCENDING

'path for saving excel workbook files after printing

'EXCEL_COA_COPY_PATH =

[EndCertOfAnalysis]

Single sample login

This section contains default settings for the LABWORKS Single Sample Login program. History list settings, login record group settings, custom program settings, and login display settings are all located in this section.

[Windows_Single_Sample_Login]

' To turn saving entries into History Lists, set the

' following variable to No

UPDATE_HISTORY_FILE=No

' text description to display on special button.

' example: Button_Text = "Select Sample". Button_Text =

' Special button assignment

' can be one of the following:

' "Sample Description", "Project Account Code", "User Field 1",

' "User Field 2".

' example: Special_Button = "Sample Description". Special_Button =

' Set "Use_Login_Record_Group" equal to "Yes" if you want to add/create

' a sample login group. Use_Login_Record_Group = No

' to create a custom login record group set "Custom_Login_Record_Group" to the exe program.

' be sure to include the entire path to the program. CUSTOM_LOGIN_RECORD_GROUP =

' Information to display when sample login completed. Up to 5 fields

' can be displayed.

FIELD1 = Sample ID: = SIDN

FIELD2 = Location Code: = LCOD

FIELD3 = Submit Date: = SDAT

FIELD4 = Due Date: = DDAT

' FIELD5 = User 1: = USR1

'FIELD's should be set to the 4 character LABWORKS data item

' from the list below.

'TDAT = Current Date

'SIDN = Sample ID

'LCOD = Location Code

'DSCR = Sample Description

'SCOL = Sample Collector

'CDAT = Collection Date

'CTIM = Collection Time

'SDAT = Submit Date

'STIM = Submit Time

'DDAT = Due Date

'PORD = P.O. Number

'PROJ = Project Code

'RFOR = Report Format

'LUSR = Login User

'LBAT = Login Batch

'RAD1 = Report Address 1

'RAD2 = Report Address 2

'RAD3 = Report Address 3

'RAD4 = Report Address 4

'RAD5 = Report Address 5

'BAD1 = Billing Address 1

'BAD2 = Billing Address 2

'BAD3 = Billing Address 3

'BAD4 = Billing Address 4

'BAD5 = Billing Address 5

'USR1 = User Field 1

'USR2 = User Field 2

'USR3 = User Field 3

'USR4 = User Field 4

'USR5 = User Field 5

'COM1 = Comment Line 1

'COM2 = Comment Line 2

'COM3 = Comment Line 3

'COM4 = Comment Line 4

'COM5 = Comment Line 5

'COM6 = Comment Line 6

'COM7 = Comment Line 7

'COM8 = Comment Line 8

' Turn on adding samples to process scheduler

ADD_TO_PROCESS_SCHEDULER=NO

[EndWindows_Single_Sample_Login]

Multi sample login

This section contains default settings for the LABWORKS Multi Sample Login program. History list settings, login record group settings, custom program settings, sample selection methods, behavior of the enter key, and login display settings are all located in this section.

[Windows_Multi_Sample_Login]

' To turn off saving entries into History Lists, set the

' following variable to No

UPDATE_HISTORY_FILE=No

' text description to display on special button.

' example: Button_Text = "Select Sample". Button_Text =

' To change the way due dates are calculated for tests, set this entry to

' One of the following: LOCCODE (default) COLDATE SUBDATE

Test_DueDate_BaseDate = COLDATE

' Special button assignment can be one of the following:

' "Sample Description", "Project Account Code", "User Field 1",

' "User Field 2".

' example: Special_Button = "Sample Description".

Special_Button =

' Set "Use_Login_Record_Group" equal to "Yes" if you want to add/create

' a sample login group. Use_Login_Record_Group = Yes

' to create a custom login record group set "Custom_Login_Record_Group"

' to the exe program name.

' be sure to include to entire path to the program.

CUSTOM_LOGIN_RECORD_GROUP =

' this entry is used to transport the login record group information

' from a custom program back to login. Setting it to a complete filespec

' will change the way the data is passed back to login. The filespec will

' be passed to the program as a command line parameter with " Result:" preceding

' the file name. If it is empty or missing, the transfer will take place as

' normal, through the keyboard.

LOGIN_RECORD_FILE_TRANSFER =

' Setting this entry to YES will cause login to create a file

' containing the location codes that are about to be logged in.

' The filename will be passed as a command line parameter with "Export:"

' preceding the filename.

EXPORT_LOCATIONS =

' Information to display when sample login completed.Up to 5 fields

' can be displayed.

FIELD1 = Sample ID: = SIDN

FIELD2 = Location Code: = LCOD

FIELD3 = Submit Date = SDAT

FIELD4 = Due Date = DDAT

' FIELD5 = User 1 = USR1

'FIELD's should be set to the 4 character LABWORKS data item

' from the list below.

'TDAT = Current Date

'SIDN = Sample ID

'LCOD = Location Code

'DSCR = Sample Description

'SCOL = Sample Collector

'CDAT = Collection Date

'CTIM = Collection Time

'SDAT = Submit Date

'STIM = Submit Time

'DDAT = Due Date

'PORD = P.O. Number

'PROJ = Project Code

'RFOR = Report Format

'LUSR = Login User

'LBAT = Login Batch

'RAD1 = Report Address 1

'RAD2 = Report Address 2

'RAD3 = Report Address 3

'RAD4 = Report Address 4

'RAD5 = Report Address 5

'BAD1 = Billing Address 1

'BAD2 = Billing Address 2

'BAD3 = Billing Address 3

'BAD4 = Billing Address 4

'BAD5 = Billing Address 5

'USR1 = User Field 1

'USR2 = User Field 2

'USR3 = User Field 3

'USR4 = User Field 4

'USR5 = User Field 5

'COM1 = Comment Line 1

'COM2 = Comment Line 2

'COM3 = Comment Line 3

'COM4 = Comment Line 4

'COM5 = Comment Line 5

'COM6 = Comment Line 6

'COM7 = Comment Line 7

'COM8 = Comment Line 8

' Indicates to MultiLog the maximum number of samples that can be logged in.

' Default is 50

MAX_Samples = 50

' ENTER_KEY_ACTION determines the movement of the cell pointer when the ENTER key is pressed.

' current options are "DOWN" and "NEXT".

ENTER_KEY_ACTION = NEXT

' FIELD_LOCKED allows data to only be selected using the history lists.

' Use the four character codes to specify which field is to be

' locked. In order for the lock to take place the field must

' be placed on the spreadsheet.

' USE USXX, (US01) for userfields

FIELD_LOCKED =

' FIELDS_FROZEN allows data to remain on screen when scrolling horizontally.

' Set the value equal to the number of rows to be frozen

' starting from the first editable column (not the row number

' column).

FIELDS_FROZEN = 0

' These are the fields for selecting location codes using the "Custom Selection

Screen"

' up to five fields can be used. the first column is the field number, the second column

' is the description to display on screen, and the third column is the name of the field

' contained in the database using the syntax "TableName.FieldName" If the field name

' contains spaces then it must be surrounded with brackets "[]" such as

USERFLDS.[Analysis number]

' Field Counter - Description - TableName.FieldName

' Select_Field = Example 1 = USERFLDS.USER1

' Select_Field = Example 2 = USERFLDS.USER2

' Select_Field = Example 3 = LOCLIST.DESCR

' Turn on adding samples to process scheduler ADD_TO_PROCESS_SCHEDULER=NO

[EndWindows_Multi_Sample_Login]

Spreadsheet results entry

This section contains parameters for the LABWORKS Results Entry program. Calculation settings, MDL settings, automatic validation, sample selection, and other related settings are located here.

[Spread_Result_Entry]

enables custom program to run after results are saved

' path defaults to programs path, extension defaults to .EXE

'POST_RESULT_PROGRAM = ProgramName ProgramParams (optional)

'POST_RESULT_PROGRAM = SHELTEST

'specifies preferences for windows spreadsheet results entry routine.
'specifies result labels to be used for 0 and quantities below mdl in single component instrument imports

'replaces environmental variables SCND and SCMDL used in DOS for same purpose

SCND = Not detected

SCBDL = Present < MDL

'specifies result labels to be used for 0 and quantities below mdl in multicomponent instrument imports

'and defaulted manual entries in results entry (RENTERO.EXE rev 4 only)

'replaces environmental variables used in earlier versions for same purpose

MCND = Not detected

MCBDL = Present < MDL

' specifies option for auto validation of sample upon completion of results entry

' Set to 'NO_VIOLATION' to automatically validate samples with no exceptions

AUTO_VALIDATE_RULE =

' This setting defaults to 20 and controls the number of the most recent samples

' considered current when selecting them by sample designation and entering

' either a location code or user field 1 value.

' CURRENT_SAMPS = N

CURRENT_SAMPS = 20

' This setting controls whether results entry will automatically bring up the

' current samples from a location code or user field 1 value, bypassing the sample

' selection screen.

' SELECT_CURRENT_SAMPS = Yes/No

SELECT_CURRENT_SAMPS = No

This setting specifies program response to enter key in sample designation text box

' OK (default) - pushes OK button NONE - no action SPACE - adds space to end of currently entered text

' used to allow multiple barcode scans into text box before loading samples

' SAMP_DESG_ENTER_ACTION = SPACE/NONE/OK

SAMP_DESG_ENTER_ACTION = OK

'specifies adjusting project analyte reporting (PAM) limits for analysis be dilution factor

'sql compliant only. Values: Yes / No (default)

ADJUST_PROJECT_REPORT_LIMITS = Yes

'specifies if Excel spreadsheet appears during calculations - YES (default) or NO

'EXCEL_CALC_VISIBLE = YES

' Set SpecSound = ON to enable the playback of WAV files for spec violations

' (requires sound card and windows multimedia drivers)

SpecSound =

'specify .WAV files for each type of violation that may occur

UpperSpec_Sound =

UpperWarning_Sound =

Target_Sound =

LowerWarning_Sound =

LowerSpec_Sound =

[EndSpread_Result_Entry]

Single component results import

This section configures single component results import through LABWORKS single component instrument interfaces. This section determines how LABWORKS handles negative values or non-numeric results. Each interface can be configured separately as shown below.

[SC_Result_Import]

'what to do if instrument reports negative values

' INSTRTYPE_NEG = ZERO change negative result to 0

' INSTRTYPE_NEG = OMIT don't report negative result

' INSTRTYPE_NEG = MDL report negative result as default MDL value for analysis

' INSTRTYPE_NEG = <MDL report negative result as 'less than ' default MDL value

' INSTRTYPE_NEG = RESULT report negative result as is

' INSTRTYPE_NEG = other report negative result as 'other' ASCII_NEG = 0

' PE_5100 = 0

' PE_ELAN_NEG = ZERO

' PE_OPTIMA_NEG = ZERO

' how wide to make result review columns in spreadsheet

' (default = 8 chars).

' INSTRTYPE_COL_WIDTH = 14:

' make columns 14 characters wide for INSTRTYPE results review

ASCII_COL_WIDTH = 12

' PE_5100_COL_WIDTH = 11

' PE_ELAN_COL_WIDTH = 11

[EndSC_Result_Import]

Interim result transfer

This section configures the custom interim results transfer program.

[Interim_Result_Transfer]

' specifies if results being saved get sent to results directory for

' immediate transfer

' dos rev 3.40 and vb (dos 3.3 rev used l:lwks.ini for this feature) INTERIM_RESULT=

' settings for windows interim result export program

RESULT_EXPORT_FREQ=3

RESULT_EXPORT_OUTFILE=Z:\DATA\LW300\ACCESS\SITEWIDE.DAT

RESULT_EXPORT_PROG=X:\APPS\LAB340\SITEWIDE.EXE

[EndInterim_Result_Transfer]

Multi component

This section configures the multi component import through multi component instrument interfaces. Settings for installed data systems and results types, dilution factors, and other multi component related defaults are set here.

[MC_Result_Import]

' specifies installed data systems and defaults for windows

' mc result transfer.

' Examples:

'GRF - LABWORKS GRF format

'TURBO41 - PE Nelson TurboChrom 4.1 (.RST)

'TURBO40 - PE Nelson TurboChrom 4.0 (.RST)

'TURBO33 - PE Nelson TurboChrom 3.x (.RST)

'PEN2600 - PE Nelson 2600 (.ATB)

'QMASS - PE Q-Mass (.QPT)

'HPCHEM - HP PC Chemstation (.TXT)

'HPMSD - HP MSD (.REP)

'HPMS_CSV - HP MSD CSV (.CSV)

'HPMS_RR - HP MSD Reduced Result File (.RR)

'AC_SRP - AC PIONA/Reformulyzer (.SRP)

'ACSIMDIS - AC HT-750 SIMDIS (.REP)

'ACPIONA - AC PIONA Analyzer (.REP)

'ADA - Precision ADA (.???)

'DIONEXAI450 - Dionex AI-450 (.R??)

'EZCHROM - EZChrom (.SQC)

'LABQUEST - Tremetrics LabQuest (.SQC)

'VSTAR - Varian GC Star (.RPB)

'CHROMPERFECT - ChromPerfect (.Axx)

'MILLENNIUM - Waters Millennium

'VG_MINICHROM - VG MiniChrom

DataSystem=GRF

DataSystem=

DataSystem=

DataSystem=

DataSystem=

DataSystem=

DataSystem=

ResultPath=c:\

DSType=GRF

NDResult = Not Detected

BDLResult = Below Det Lim

ResultType = A

DilFactor = 1

RenameOption = N

[EndMC_Result_Import]

Serial Instruments

[Serial_Instruments]

' ANL & Analysis Name = Instrument_Name (in InstrumentLibrary) for results entry

' ANL_-TSSTARE = BALANCE_01

' ANL_-TSSFWT = BALANCE_01

' ANL_TAREWT = BALANCE_01

' ANL_FINALWT = BALANCE_01

' ANL_PH = PHMETER

' SEQ_WEIGHT_ & INSTR CODE = Instrument_Name (in InstrumentLibrary) for sequence loading

' SEQ_WEIGHT_PE_5100 = BALANCE_05 [EndSerial_Instruments]

Invoicing

This section sets up default fields that can appear on invoices in addition to the invoice number.

[Invoicing]

' specifies additional fields to appear in invoices.prn line

' fields appear in order specified immediately after invoice number, before

' the other standard fields

' options currently supported: PROJ, SCOL, SBAT, US01-USxx

' PRN_FIELD = PROJ

' PRN_FIELD = SCOL

' PRN_FIELD = US01

[EndInvoicing]

Windows standard labels

This INI file section configures the standard labels printed at login and sample modification. The fields to be displayed, text size, and captions are set up here. Additionally, the printer to be used to print labels is setup here also.

[Windows_Standard_Labels]

' KEY WORDS or the line syntax must not be changed.

' all key words must be followed by an equal (=) sign.

' List of current KEYWORDS supported:

' FIELD1

' FIELD2

```
' FIELD3
' FIELD4
' FIELD5
' PRINTER_NAME
'FIELD's should be set to the 4 character LABWORKS data item
' from the list below.
'TDAT = Current Date
'SIDN = Sample ID
'LCOD = Location Code
'DSCR = Sample Description
'SCOL = Sample Collector
'CDAT = Collection Date
'CTIM = Collection Time
'SDAT = Submit Date
'STIM = Submit Time
'DDAT = Due Date
'PORD = P.O. Number
'PROJ = Project Code
'RFOR = Report Format
'LUSR = Login User
'LBAT = Login Batch
'RAD1 = Report Address 1
'RAD2 = Report Address 2
'RAD3 = Report Address 3
'RAD4 = Report Address 4
'RAD5 = Report Address 5
'BAD1 = Billing Address 1
'BAD2 = Billing Address 2
'BAD3 = Billing Address 3
'BAD4 = Billing Address 4
'BAD5 = Billing Address 5
'USR1 = User Field 1
'USR2 = User Field 2
```

'USR3 = User Field 3

'USR4 = User Field 4

'USR5 = User Field 5

'COM1 = Comment Line 1

'COM2 = Comment Line 2

'COM3 = Comment Line 3

'COM4 = Comment Line 4

'COM5 = Comment Line 5

'COM6 = Comment Line 6

'COM7 = Comment Line 7

'COM8 = Comment Line 8

'Barcodes and barcode text can be toggled on or off depending on user preference.

'The parameter BARCODES turns the barcode on and off on the label, and the parameter

'BARCODE_TEXT turns the caption under a barcode on or off. Note that the barcode

'option for LABWORKS must be installed for these parameters to work. Default settings

'are ON for both these parameters.

'BARCODES = OFF

'BARCODE_TEXT = OFF FormFeed = N

FIELD1 = Location: = LCOD

FIELD2 = Description: = DSCR

FIELD3 = Submit Date: = SDAT

FIELD4 = Due Date: = DDAT

FIELD5 = My Field: = USR1

Printer_name = "SELECT"

'HP LaserJet 4/4M

'Epson FX-850

[EndWindows_Standard_Labels]

User field increments

This section is used to configure user field incrementing. User defined fields can be set to increment within a g roup of sample logged in at Multi Sample Login. This allows use of incremental numbers or codes to keep track of the order of samples logged in.

[User_Field_Increments]

' Only USER FIELDS support Incrementing.

' Each field must start with a seed.

' The seed will be used as the first number and will be incremented each

' time it is used.

' Seeds may include numbers and letters.

' Make sure you include enough digit positions in the seed to cover your

' needs.ex: if your seed is 1 then it will increment up to 9 and roll

' over back to 0.If you want to increment up to 10000 then make your

' seed equal to 00001.

' You can not use spaces at the beginning or end of the seed.

' Fields must be setup using the following syntax with the word Seed being

' replaced by your actual seed data:

'USER1 = AA-AAA-00091

'USER2 = Seed

[EndUser_Field_Increments]

Exception report

[Exception_Report]

'specifies date/time to use for sorting and displaying exception trail

'COLDATE (collection) SUBDATE (submittal) ANSDATE (anl start) ANEDATE (anl end) VIOLDATE (violation posting - default)

'EXCEPTION_DATE = COLDATE

[EndException_Report]

Validation

This section specifies default settings for the validation routines in LABWORKS. You can turn the search for QA samples on or off here, which may affect the speed of loading samples in validation. Event logging for validation is also controlled here.

[Validation]

'parameters for windows sample validation program

'should a search be done for qa/qc results for validation samples (ON / OFF default is ON)

'QAQC_SEARCH=OFF

'specifies event logging to lwuserpath\validato.log

LOGACTIVE=YES

This setting specifies program response to enter key in sample designation text box

' OK (default) - pushes OK button NONE - no action SPACE - adds space to end of currently entered text

' used to allow multiple barcode scans into text box before loading samples

' SAMP_DESG_ENTER_ACTION = SPACE/NONE/OK

SAMP_DESG_ENTER_ACTION = OK

[EndValidation]

QA batching

This section controls the behavior and defaults for the QA batching programs inLABWORKS.

[QA_Batching]

' settings control default behavior of Windows QA/QC batching routines

' sets default batch size for all analyses (if not set here default is 10 samples)

' DEFAULT_BATCH_SIZE = 15

' sets default batch size for a particular analysis

' BATCH_SIZE_analcode = nn

' BATCH_SIZE_CD = 20

[EndQA_Batching]

Result scan

This section sets the defaults for automatic results scanning from certain instrument interfaces. This is an optional program.

[ResultScan]

'default settings for automatic instrument result scanning and loading

'post log entries during operation (default is 'NO')

'LOGACTIVE = YES

'how often to load data in minutes

LOAD_FREQ=3

'loading modes:

' "MULTI" parse all result files at once then post to database (default)

' "SINGLE" parse and post each result file one by one

'LOAD_MODE = SINGLE

'SCAN_INSTR = inst_codedata_pathrlt_file_template addtest (optional)

nochange (optional)norename (optional)

'"NORENAME" option applies only to single load mode causes file to be renamed AFTER posting rather than before parsing (default)

' Philips PW-1480 XRF (imports PW1480 results from c:\results\xray*.lst files with result changes but no work order additions)

'SCAN_INSTR = IAI_PHILIPS_X40D:\DATA1*.SAV

'SCAN_INSTR = IAI_CANNON_VISCD:\DATA2\ CAVDAT.DAT SCAN_INSTR = IAI_ASCII C:\data\lw300\user *.rlt

SCAN_INSTR = IAI_ASCII C:\data\lw300\user *.asc

[EndResultScan]

ALOGINO

Sets defaults for the Power Programmer package option ALOGINO.

[ALOGINO]

'settings for ole power programmer login routine

'specifies event and resource logging to lwuserpath\alogino.log

LOGACTIVE=YES

'specifies a project analyte reporting group for default(LCOD, SCOL, PORD, PROJ, USXX)

'enterprise with sql compliant databases only

PROJECT_ANALYTE_GROUP = PROJ

[EndALOGINO]

History display

This section sets the defaults for the history display found in Results Entry and other areas of LABWORKS. Default entries for sorting and grouping of results are set here.

[History_Display]

'parameters control history display (rev4 only)

'number of most recent results points to displaydefault is 10 min is 3 max is 100

POINTS = 10

'date to sort history by (supercedes HISTSORT environmental variable)

'values (COL / SUB / ANL) coldate / subdate / analysis_date - default is COL

' SORTDATE = ANL

'display earliest or latest point at top (EARLY / LATE)default is EARLY

'TOP = LATE

TOP = EARLY

'display secondary result also (YES / NO) Enterprise and ES - default is NO

RESULT2 = No

'display qualifier also (YES / NO) Enterprise and ES - default is NO

QUALIFIER = No

[EndHistory_Display]

Sample modify

The sample modify section contains settings used to determine how the sample modify/delete module behaves in LABWORKS. Here, settings for auto- validation are stored. Also, the default setting for the number of samples to be displayed in modify/delete is set in this section.

[Sample_Modify]

'parameters for windows sample modification programs

'should sample be auto validated when no test is pending

'Auto_Validate_Rule=No_Violations

' specifies number of samples (default=20) considered current for modification by sample designation of loccode or user field 1

CURRENT_SAMPS = 14

This setting specifies program response to enter key in sample designation text box

' OK (default) - pushes OK button NONE - no action SPACE - adds space to end of currently entered text

' used to allow multiple barcode scans into text box before loading samples

' SAMP_DESG_ENTER_ACTION = SPACE/NONE/OK

SAMP_DESG_ENTER_ACTION = OK

'specifies the base date option to use for calculating due dates of analyses added during sample modification

'ADD_TEST_BASEDATE = LOCCODE (default) COLDATE SUBDATE CURDATE

ADD_TEST_BASEDATE =

'specifies program to run after each modified sample in single sample modify

'include full path and extension or program path and .exe are assumed

'SINGLE_SAMPLE_USER_PROG = SHELTEST

'specifies program to run after multisample modifications are saved

'include full path and extension or program path and .exe are assumed

'MULTI_SAMPLE_USER_PROG = SHELTEST

[EndSample_Modify]

Progress report

Defaults for the Windows Progress report program are stored here. Settings for the default number of samples, the

[Progress_Report]

'parameters for windows progress report

' specifies number of samples (default=20) considered current for progress report by sample designation of loccode or user field 1

CURRENT_SAMPS = 14

' specifies if automatically selects current samples (default=No)

'SELECT_CURRENT_SAMPS = YES

This setting specifies program response to enter key in sample designation text box

' OK (default) - pushes OK button NONE - no action SPACE - adds space to end of currently entered text

' used to allow multiple barcode scans into text box before loading samples

' SAMP_DESG_ENTER_ACTION = SPACE/NONE/OK

SAMP_DESG_ENTER_ACTION = OK

[EndProgress_Report]

Custody tracking

This section sets defaults for the custody tracking programs in LABWORKS. Report labels for null values, default initial location for containers, etc., are set here.

[Custody_Tracking]

'parameters for sample container custody tracking

' specifies report label for null (not entered) contained location(default = Unspecified)

'NULL_LOCATION = Not in custody

'specifies default initial location for all container types

'INITIAL_LOCATION = Receiving

'specifies report description for location of sample containers before login

PRELOGIN_LOCATION = Not in house

'specifies initial locations for specific container codes which supercede default

'IL_@TCOL = Bact receiving

'IL_@WET = Chem receiving

'IL_@VOA = Orgn receiving

'IL_@VOAD = Orgn receiving

'IL_@VOAS = Orgn receiving

'IL_@VOAT = Orgn receiving

'specifies reason to be associated with initial placement of all container types during login

INITIAL_REASON = New sample submitted to laboratory

' specifies number of samples (default=20) considered current for tracking by sample designation of loccode or user field 1

CURRENT_SAMPS = 12

' specifies if automatically selects current samples (default=No)

'SELECT_CURRENT_SAMPS = YES

'specifies if reason must be entered for moving sample container (default=No)

FORCE_MOVE_REASON = YES

'specifies if entered reason for moving sample container is saved to list (default=Yes)

SAVE_MOVE_REASON = YES

'specifies if entered destination for sample container move is saved to list (default=No)

SAVE_MOVE_DESTINATION = No

[EndCustody_Tracking]

Analysis selection

This section specifies optional selection fields to be displayed in the analysis selection dialogs throughout LABWORKS.

[Analysis_Selection]

'optional fields to display in analysis selection dialog (analysis code and name are standard)

'options are: DEPT (department) REF (reference) MDL UNIT PRICE

'FIELD = REF

'FIELD = UNIT

[EndAnalysis_Selection]

Process Scheduler

[Process_Scheduler]

' Settings for Security Mode

' GLOBAL or DISCRETE SECURITY_MODE = GLOBAL

' Settings for optional process scheduler

' Preview time = # of minutes before sample requested time sample displays in list

PREVIEW_TIME = 59

' Preview_Hours_Daily = Number of hours prior to schedule hourly samples will be displayed, max = 23

' PREVIEW_HOURS_DAILY = 12

' Preview_Days_Weekly = Number of days prior to schedule weekly samples will be displayed, max = 6

' PREVIEW_DAYS_WEEKLY = 4

' Preview_Days_Monthly = Number of days prior to schedule monthly samples will be displayed, max = 28

' PREVIEW_DAYS_MONTHLY = 20

' Restrict Preview for pre/logged in samples, Show all samples or Only Samples with request time in preview window.

RESTRICT_PREVIEW = NO

' Refresh Interval is frequency data is updated

REFRESH_INTERVAL = 5

' Retain Window the number of hours a sample is displayed after Production OK

RETAIN_WINDOW = 1

' Validation Type, NONE = results finished only, SAMPLE = sample validation, LAB = manual lab validation

VALIDATION_TYPE = SAMPLE

' ADD_TIMED_SAMPLES, adds samples in time scheduled database, only one machine need to run this

ADD_TIMED_SAMPLES = YES

' AUTOLOG_TIMED_SAMPLES , automatically logs in Timed Samples

AUTOLOG_TIMED_SAMPLES = YES

' LOGIN PASSWORD , User password for Autologin of Samples

AUTOLOG_PASSWORD = 1

' PRIMARY_SYSTEMis the LABWORKS Machine ID for the system doing autologin and adding of timed samples

PRIMARY_SYSTEM = PF-DELL650

' Progress Report Format ' format for displaying progress report

PROGRESS_FORMAT = DEFAULT

' Result Entry Format ' format for displaying results entry

RESULT_FORMAT = RESULTS

' View results format , if native then one sample at a time , No is native, YES = Progress Report, default

VIEW_PROGRESS = YES

' User field descriptions / Leave blank to omit field

'USER1_DESC = Collect Type

'USER2_DESC = Location

'USER3_DESC = QC Type USER4_DESC = NUSER1_DESC = NUSER2_DESC =

' User field to LABWORKS Field Mappings

'USER1_ASSIGN = USR15

'USER2_ASSIGN = USR1

'USER3_ASSIGN = USR20

USER4_ASSIGN =

'USER1_COMBO = YES

USER2_COMBO = NO

USER3_COMBO = NO

USER4_COMBO = NO

USER1_REQUIRED = NO

USER2_REQUIRED = NO

USER3_REQUIRED = NO

USER4_REQUIRED = NO

' PROJ

' DSCR

' PORD

' LBAT

' RAD1

' USR1 - USR20

LOGIN_PRIORITY = 1

LOGIN_DEPT = LOGIN_NOPROMPT = YES

'LOGIN_DEPT_FIELD=PROJ

LOGIN_INITIAL_STATUS = 2

'0 = NOT READY

'1 = READY TO COLLECT

'1.5 = COLLECTED

'1.8 = IN TRANSIT

'2 = WAITING ANALYSIS LOGIN_USELRF = YES

' Show Description displays the location code description instead of the location code in the main display.

'SHOW_DESC = YES

SHOW_REQ = Yes

SHOW_COLOR = NO

SHOW_COL = NO

SHOW_WAITINGANALYSIS = NO

SHOW_ANALYSISCOMPLETE = NO

SHOW_LABOK = NO

SHOW_PRODOK = NO

SHOW_VIOLATION = YES

SHOW_PSID = NO

'display earliest or latest point at top (EARLY / LATE) default is EARLY

'TOP = LATE

TOP = LATE

' uses status collected between ready to collect and login

USECOLLECTED = NO

' Logs sample in at Ready to collect for label printing

LOGIN_AT_READYTOCOLLECT = NO

' Show Collected samples in Backlog, Default is NO, Sample is inactive until lab receipt

COLLECTED_IN_BACKLOG = NO

SHOW_COLLECTED = NO

' REQUIRE_PRODUCTIONOK default is YES, Must be reviewed by requester, NO and Last status is LABOK

REQUIRE_PRODUCTIONOK = YES

' Default Number of Labels

DEFAULT_NO_LABELS=1

' Eliminate Login OK Prompts / make barcoding easier

NO_PS_PROMPTS=NO

' Show Barcode Enhancements

SHOW_BARCODE = YES

' Include Status Transit to Lab

USE_IN_TRANSIT=NO

' Use Bottle/Analysis Definition in PSCONTAINER table

USE_PSCONTAINER = NO

' Use Bartender Labeling Software

USE_BARTENDER = NO

' Days of Audit Trail to keep - Default is 30

PSAUDIT_DAYS_TO_KEEP = 30

' For pen computers used left or right click

LEFT_CLICK = NO

' Sample Collector is required by default when assigning LABWORKS SampleID

' Default COLLECTOR_REQUIRED = YES

COLLECTOR_REQUIRED = YES

' Show Coc

SHOW_COC = YES

' Use Multi-Sample login for Sample Login

USE_MULTILOG = YES

USE_SINGLELOG = YES

SHOW_PRELOGBATCH = NO

SHOW_LBPROMPT = YES

' ' to create a custom login record group set "Custom_Login_Record_Group"

' ' to the exe program name.

' ' be sure to include to entire path to the program.

'

' CUSTOM_LOGIN_RECORD_GROUP =SAMNUMB.EXE

[ENDProcess_Scheduler]

LABWORKS explorer

[LABWORKS_EXPLORER]

SHOW_SQC = YES

SHOW_XREF = YES

SHOW_EXCEPTION = YES

' Show Location Code Description

SHOW_LOCDESCR = NO

' Show Pending Validation Status

SHOW_VALID_STATUS = NO

[ENDLABWORKS_EXPLORER]

DMR

[DMR]

' entry to tell DMR to change COMP24 to 24HC

SMP_TYP = COMP24 = 24HC

Smp_Frq = Continuous = CONT

Smp_Frq = INSTAN = RCORD

[ENDDMR]

EPA control chart

[EPA_CONTROL_CHART]

' The user field associated with the matrix

MATRIX_FIELD = SAMPTYPE

' The default number of points in a chart group

NPOINTS = 25

' The number of standard deviations for calculating control limits

STDEV = 3

' The default starting data for looking for new datapoints.

STARTDATE = 1/1/95

[ENDEPA_CONTROL_CHART]

Email

[EMail]

'specifies the name of the e-mail server in use at the workstation

'EMAIL_SERVER =

'Exeption_Report_Sender =

[EndEMail]

GPS

[GPS]

' The GPS data is stored in two user defined fields, these fields are defined below

' as GPS_LAT_USERFIELD and GPS_LON_USERFIELD.

' Set each keyword equal to the field number to use.

' Ex: GPS_LAT_USERFIELD = 3 will place the Latitude data in user field 3

GPS_LAT_USERFIELD = 11

GPS_LON_USERFIELD = 12

[EndGPS]

BarTender labels

[BARTENDER_LABELS]

' this section is for labels using bartender and COC's using Access

' Label types

' 1 = all analyses on one record

' 2 = one record for each department

' 3 = one record for each bottletype.

' 4 = one record for each pscontainer

' 5 = one field for each analysis department

LABEL_TYPE= 1

SHOW_PRINT_DIALOG = NO

[ENDBARTENDER_LABELS]

[BARTENDER_LABELS]

PRINTER_SEARCH = DMX

Will loop through connected printers looking for a match. This is useful in a Citrix configuration when the path will be different but the printer name will be the same.

CRYSTAL_COC = YES

Use crystal report file COC.RPT as format for COC

[ENDBARTENDER_LABELS]

Glossary

Analysis Code

One to 24-character code that uniquely identifies each analysis in LABWORKS. If QA/QC analyses will exist in the database, only use 22 characters for each analysis that will have a QA/QC analysis associated with it. The analysis code cannot contain characters such as periods, commas, question marks, asterisks, or internal spaces.

Analysis Hold Days

Specifies the time, in days, used for calculating the analysis due date. The analysis due date base is specified in the Location Code list database.

Analysis Name

One to 40-character field that describes the analysis. This description appears on most reports.

Billing Address

The Billing Address contains the address to which invoices should be sent if invoicing is to be used. It contains as many as five lines with each line having as many as 40 upper- and lowercase characters.

Billing Rate

The billing rate sets the percentage of list price used for calculating charges for work performed on samples from the location code if invoicing is to be used.

Check Specifications

Check specifications function allows the user to choose whether specification checking should be performed on the sample from the sample point.

Component Abbreviation

The component abbreviation is used for instrument interfacing and must match the component designation in the instrument result file for correct matching of LABWORKS analytes with instrument results.

Component Name

The LABWORKS component name is the full name of an analyte and can contain forty characters. This is the analyte name that is used on reports.

Default MDL

The Minimum Detection Limit (MDL) is a numerical expression of the lowest concentration that the analysis can quantify (or the smallest measurement increment that the analysis can distinguish), expressed in the analysis reporting unit. See **Round Off For Calculations** for details about using the MDL as decimal places for rounding when transferring data from instruments and performing analysis result calculations.

Department

24- character field that defines which laboratory department receives orders to perform a particular analysis. Backlog and final reporting can use this field for sorting or grouping. All characters are converted to uppercase and spaces are not allowed.

Disk Copy Final Report

Disk copy final report specifies whether to print a DOS standard final report to disk. This field is only used by native database systems operating with both DOS and Windows® workstations.

Due Date Based On

The Due Date Based On option allows analysis due dates to be calculated as hold days from the collection date or submittal date.

Hold Days

The sample holding days value specifies the time in days between sample collection or sample submittal date and the sample due date. This value may be any integer from 0 to 999. A value of zero specifies that the sample analysis is due on the same date at the collection or submittal.

Invoiceable

Indicates if the analysis is billable on invoices.

Invoice Name

The invoice name is used to specify custom DOS invoice formats if a format other than the standard one is to be used.

Invoice Samples

Invoice Samples allows the user to choose whether to invoice samples for the location code or bypass invoicing.

LIMS

Laboratory Information Management System

Location Code

The Location Code is the alphanumeric code assigned to all samples entered into the LIMS. From this point on, Location Code will be known as Sample Point.

Method Reference

Sixteen-character field used to specify the sources of procedures being used for analyses.

Minimum Detection Limit (MDL)

The Minimum Detection Limit is a numerical expression of the lowest concentration that the analysis can quantify (or the smallest measurement increment that the analysis can distinguish), expressed in the analysis reporting unit.

Multicomponent Analysis

Code must start with a dollar sign ($).

Narrative Text Analysis

Code must start with an exclamation point (!).

Price

List price expressing the basic price per analysis.

Project Account Code

The Project Account Code is a re- nameable field that allows samples to be treated as a group for purposes of processing or billing. Groups may comprise all of the samples for a particular research project or all samples being processed under one contract. Multiple locations may share a single project account code. The Project Account Code consists of as many as 24 uppercase alphanumeric characters, including all printing characters except the asterisk.

Reportable

Indicates if the analysis is reportable in final reports. Analyses used in calculations are not report-able, but the final calculated analysis will be reported.

Report Hard Copies

The Report Hard Copies field is used by LABWORKS to specify from 0 to 9 final report hard copies. Entering 0 causes LABWORKS not to print a final report, and causes the sample to bypass the final reporting queue.

Reporting Address

The reporting address field contains the address where reports of results on samples from the location code are to be sent. The address may contain as many as five lines with each line having as many as 40 upper- and lowercase characters.

Reporting Units

Fourteen-character field defines the measurement unit used to store results of a particular analysis.

Report Name

The Report Name field is used by LABWORKS to specify the final report format for a sample.

Round Off For Calculations

Two different methods for rounding off results are available when importing data from instruments.

Significant figures indicates that results will be rounded to the number of significant figures indicated

Use MDL places indicates that results are to be rounded to the number of decimal places used in the MDL.

Use Ranges indicates that specific rounding rules are defined for specific value ranges.

Sample Description

A Sample Description characterizes the location from which a sample is collected. It may consist of as many as 60 upper- and lowercase characters.

Sample Point

One to 24-character code that uniquely identifies each location/sample point/client in LABWORKS. The location code may contain any printing characters except an asterisk, a question mark, a comma, a colon, an internal space, a slash, a backslash, or a period, or the characters <, >, or |. The system manager can rename the location code field. Some examples are sample point, client code, etc.

Single Component Analysis

Code does not require a special prefix.

Special Info Form

Special info form allows the user to specify the name of a special information screen to be associated with this sample point.

Special Price List

If a price list other than the standard one is to be used for specifying final prices for analyses performed on samples from the sample point, the name of the alternate price list can be entered here.

Study Group Analysis

Code must start with a pound sign (#).

Laboratory System Administrator (LSA)

The individual responsible for maintaining the LIMS and is the contact person for all technical support issues.

User Program

User program field can be used to specify an executable program or batch file which is to be run before a sample is logged in, after a sample is logged in, before all samples are logged in or after all samples are logged in.

Validate Samples

Validate Samples allows the user to specify whether to validate samples from the location code or bypass validation.

Work Units

Time in any units desired used to perform an analysis.

Additional Reference Material

Ownership Groups for Controlled Data Access

When implementing a multi-site Laboratory Information Management System (LIMS) or a single site LIMS with multiple business units, one necessity is the ability to segregate the data so that designated users can view only the data specific to their operation. This can be done by either separating the data into multiple databases or by restricting the available data on a per user basis. The merits of each of these two choices are based on business, legal, and infrastructure issues. This section presents the theory and application of Ownership Groups in LABWORKS Version 6.

LABWORKS applies ownership to Location Codes and their related Samples, Analysis Codes, and Reports.

LABWORKS Users

LABWORKS Users have a User ID and Password as well as specific privileges.

Groups

LABWORKS maintains a list of Owner Groups. An Owner Group has Group ID, Group Description, and Group Active properties. Groups are created with members. The members may be individuals or other groups.

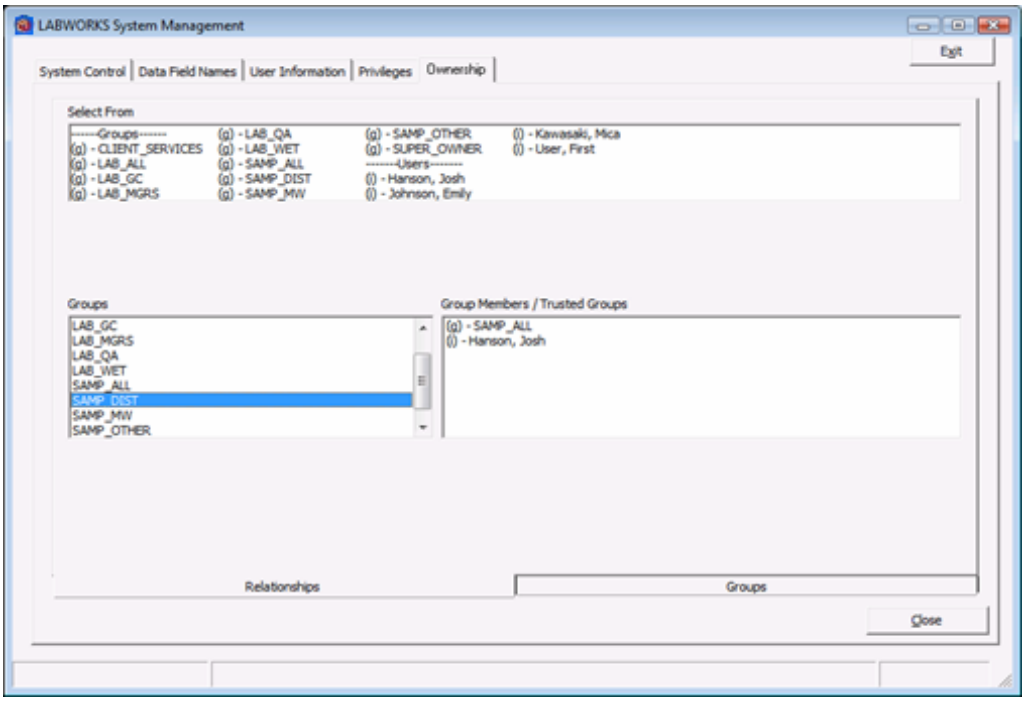
Group membership

LABWORKS maintains a list of Groups of which a user is a member. A User can be a member of one or more groups.

Trusted groups

A Group Supervisor can permit other groups to access samples he/she owns, thereby, setting up a Trusted relationship with the other group.

The following example shows details for the group SAMP_DIST (Distribution Samples). This group has a member, Josh Hanson, and also trusts the members of the group SAMP_ALL.



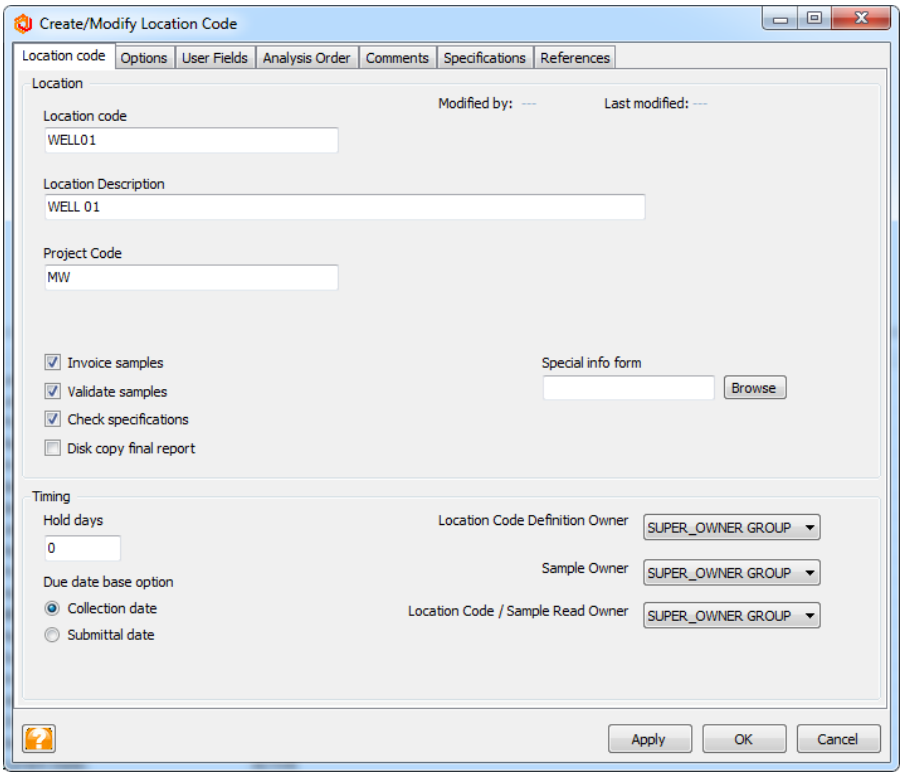
Sample ownership

The base level of ownership in LABWORKS is the Sample. The sample inherits its ownership setting based on the Location Code that was used when that sample was originally logged in.

In LABWORKS 6, another ownership setting can be associated with the Location Code itself, which can be used to limit the users who are permitted to access the definition of the Location Code.

Thus, the settings related to both Sample Ownership and Location Code ownership are maintained in the Location Code Maintenance module.

Ownership for Location Codes and Samples has multiple parameters to configure for operation.



There are 3 owner fields that are set in the Location Code Record:

- Location Code Definition Owner
- Sample Owner
- Location Code/Sample Read Owner

Location Code Definition Owner

This owner group is uses who can modify the location code. This ownership remains at the Location Code definition level and is not inherited by the sample.

Sample Owner:

This owner group is users who can load the sample in results entry in edit mode or for sample modify. However, the privileges to Run Results Entry or Modify Samples are still required. Ownership only provides visibility to the sample, whereas privileges are required to perform an action.

Location Code/Sample Read Owner:

The Location Code/Sample Read Owner is available for users who need access to the location code or sample but with opening them for modify. For example, Sample Login personnel can be members of this group, where they have access to the location code, but not to modify them, only to create samples from those location codes. Another example is the use by a Client Services group. This group requires only read access to view results or generate reports.

.

Sysmgr Settings for Location Code/Sample Ownership

There are System Manager settings that control Location Code/Sample ownership.

LOCCODE_OWNERSHIP_ACTIVE	Yes/No
LOCCODE_OWNERSHIP_PESSIMISTIC	Yes/No
SAMPLE_OWNERSHIP_ACTIVE	Yes/No
SAMPLE_OWNERSHIP_PESSIMISTIC	Yes/No

Details on these settings are

provided below:

LOCCODE_OWNERSHIHP_ACTI

VE

This parameter turns on and off Location Code Ownership. In many configurations location code ownership can be turned off, and still use sample

ownership. A configuration like this would not restrict location codes used at login, but still enforce sample ownership where results are concerned.

LOCCODE_OWNERSHIP_PESSIMISTIC

The parameter controls the visibility of location codes that don't have ownership defined. When location ownership is turned on, by default location codes without owners defined are visible to all users. When LOCCODE_OWNERSHIP_PESSIMISTIC is turned on, location codes without owners defined are not visible.

SAMPLE_OWNERSHIP_ACTIVE

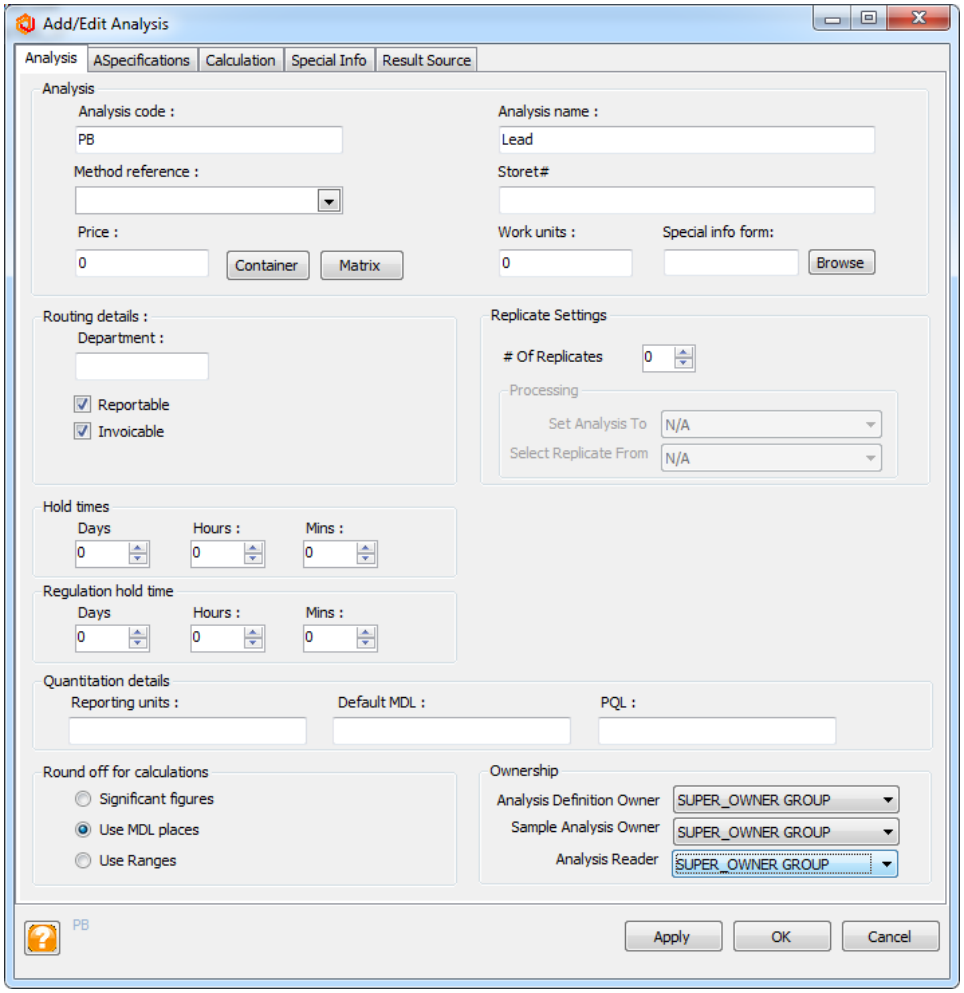
This parameter turns on and off sample ownership. Sample ownership is used to limit visibility of samples and associated results to specific users. (Replaces 5.8 OWNER_ACTIVE)

SAMPLE_OWNERSHIP_PESSIMISTIC

The parameter controls the visibility of samples that don't have ownership defined. When sample ownership is turned on, by default samples without owners defined are visible to all users. When SAMPLE_OWNERSHIP_PESSIMISTIC is enabled, samples without owners defined are not visible. (Replaces 5.8 OWNER_PESSIMISTIC)

Analysis Ownership

Analysis ownership configuration is similar to Location Code/Sample ownership. There is Analysis Code ownership as well as Analysis ownership. Analysis code ownership works with analysis code definitions. Analysis ownership works with Analyses that have been assigned to samples and can contain results.



In the preceding example the Group LAB_WET has full visibility to PH analyses. The LAB_MGRS group is configured as the group that can maintain the definition of the PH analysis code, and CLIENT_SERVICES has read access to the PH analysis

Sysmgr Settings for Analysis Ownership

There are System Manager settings that control Location Code/Sample ownership.

ANLCODE_OWNERSHIP_ACTIVE	Yes/No
ANLCODE_OWNERSHIP_PESSIMISTIC	Yes/No
ANALYSIS_OWNERSHIP_ACTIVE	Yes/No
ANALYSIS_OWNERSHIP_PESSIMISTIC	Yes/No

ANLCODE_OWNERSHIHP_ACTIVE

This parameter turns on and off Analysis Code Ownership. In many configurations analysis code ownership can be turned off, and still use analysis ownership. A configuration like this would not restrict analysis codes used at login, but still enforce analysis ownership where results are concerned.

ANLCODE_OWNERSHIP_PESSIMISTIC

The parameter controls the visibility of analysis codes that don't have ownership defined. When ANLCODE ownership is turned on, by default analysis codes without owners defined are visible to all users. When ANLCODE_OWNERSHIP_PESSIMISTIC is turned on, analysis codes

without owners defined are

not visible.

ANALYSIS_OWNERSHIP_

ACTIVE

This parameter turns on and off analysis ownership. Analysis ownership is used to limit visibility of analyses linked to samples and associated results to specific users.

ANALYSIS_OWNERSHIP_PESSIMISTIC

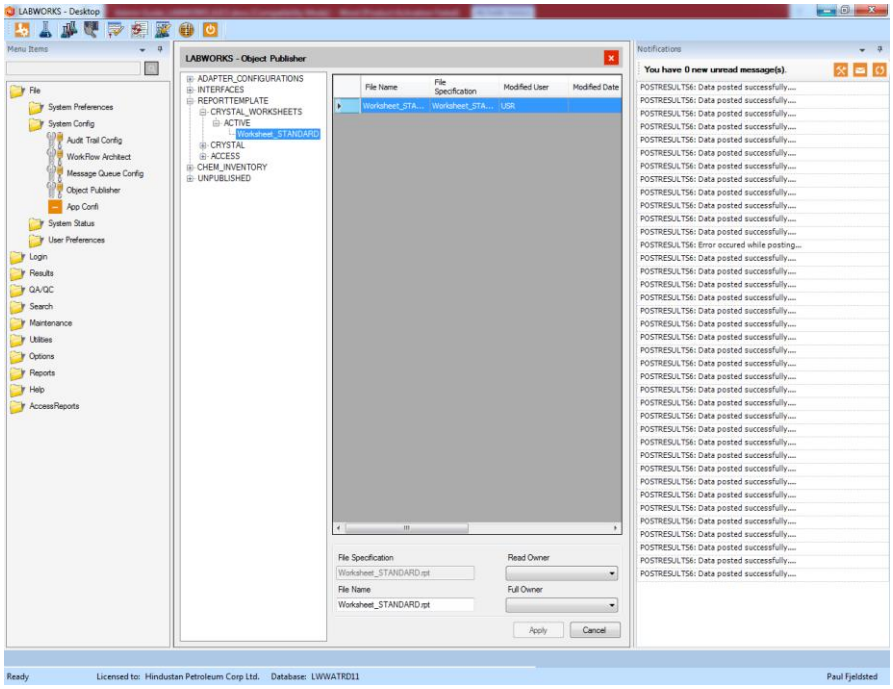
The parameter controls the visibility of analyses that don't have ownership defined. When analysis ownership is turned on, by default analyses without owners defined are visible to all users. When

ANALYSIS_OWNERSHIP_PESSIMISTIC is turned on, analyses without owners defined are not visible.

Report Ownership

Report Ownership is configured using Object Publisher. Object Publisher is a new feature for LABWORKS 6.0. It is used to manage report ownership and report state. With LABWORKS 6.0 the types of items managed by Object Publisher are:

- Access Reports
- Crystal Reports
- Word Reports
- Export Formats



Before a report is visible in LWExplorer (.NET) version or in Process scheduler it must be published by Object Publisher. Then using rules similar to Samples and Analyses ownership can be configured.

Sysmgr Settings for Report Ownership

There are System Manager settings that control report ownership.

REPORT_OWNERSHIP_ACTIVE	Yes/No
REPORT_OWNERSHIP_PESSIMISTIC	Yes/No

REPORT_OWNERSHIP_ACITVE

This parameter turns on and off report template ownership. Report ownership is used to limit visibility of reports templates available for report generation

REPORT_OWNERSHIP_PESSIMISTIC

The parameter controls the visibility of report templates that don't have ownership defined. When report ownership is turned on, by default report templates without owners defined are visible to all users. When REPORT_OWNERSHIP_PESSIMISTIC is turned on, report templates without owners defined are not visible.

Data Model for LABWORKS

The LABWORKS database contains several main tables of information as well as tables which are only used by add-on applications. This document is an overview of the main tables and their relationships.

Location Code Related

LOCLIST

The LOCLIST table is used to store Location Code information. Location Codes can be used as Sample Points, Customers, Products, Formulations, or uses to describe information about a sample. The default Location Code information is stored in the LOCLIST table. At the time of sample login this information is copied to the SAMPLE table. Changes to the default information made at the time of login are stored with the sample. The LOCLIST table has ancillary tables used to store defaults specific to an analysis or that location code.

Required Fields:

LOCCODE

LABOBJECTCOM

MENTS

The LABOBJECTCOMMENTS table stored the default comments for a location code. At the time of sample login the default comments for the location code are loaded so they can be edited and stored as sample comments after login is completed.

LOCTESTS

The LOCTESTS table stores the list of tests for a location code. These tests are the default tests assigned for the location code for sample login.

Required Fields:

- LOCCODE
- ACODE

Foreign Key:

- LOCLIST.LOCCODE
- ANL_LIST.ACODE

LABOBJSPECINFO

The Special Info Default (LABOBJSPECINFO) table stores information used with special information forms. A special information form permits the creating of a data structure, which is specific to one or more location codes. The defaults for the special information form are stored in this table. After login the special information associated with a sample is stored in the SPECINFO table.

USERFLDS

The User Fields (USERFLDS) table stores user defined information common to all location codes and samples. The USERFLDS table stores the default user field data for a location code and the SUSERFLDS stores the User Field data associated with a sample

Required Fields: LOCCODE

Foreign Key: LOCLIST.LOCCODE

Analysis Code Related

ANL_LIST

The Analysis List table contains a list of all of the available analyses and their properties. There are entries in the ANALYTES table for each analyte in every analysis.

Required Fields: ACODE

ANALYTES

The analytes table contains one record for each analyte in a multi-component in the

ANL_LIST table. Required Fields:

- ACODE
- ANLNAME (Analyte Name)

  Foreign Key:

  ANL_LIST.ACODE ***SGTESTS***

The SGTESTS table stores lists of analyses in sample analyses groups (#XXXXX) Required Fields:

- ACODE
- SGACODE

Foreign Key: ANL_LIST.ACODE

LABOBJSPECINFO

The Special Analysis Info Default (LABOBJSPECINFO) table stores information used with special information forms. An analysis special information form permits the creating of a data structure which is specific to one or more analysis codes. The defaults for the special information form are store in this table.

Results Entry

The process of entering results into the LABWORKS database requires look-ups from several tables to do specification checking, calculations, and analyte substitution.

Standard Results

When entering results the user can enter an abbreviation and LABWORKS looks up the complete result from the standard Results picklist “RLTCODES” Example: If the RLTCODES picklist contains the line: "POS", "Positive", and the user enters “POS” as the result, then “Positive” is stored as the result.

Specification Checking

Each analysis code has storage for specifications. These are Lower Control, Lower Warning, Target, Upper Warning, and Upper Control. These values are stored in the RESULTSPECS table. Specifications generic to the analysis code are stored with the value “&GENERAL” for the location code. Specifications specific to a location code are stored with that location code, analysis code, and analyte name as the key.

When entering the sample results for an analysis, the program checks the RESULTSPECS table for specifications matching the location code, the analysis code, and analyte. If no entries are found, the program checks for specifications that are generic to the analysis code.

Calculations

LABWORKS has two calculation engines, native and Excel.

- LABWORKS Native Calculations

As LABWORKS loads analysis codes for results entry, it inspects each analysis to determine if it has calculation inputs and if all of the defined inputs are present (Analysis Codes INVAR1-10). After results are entered for an analysis, the software rechecks the sample to determine if all inputs are completed and then performs the calculation.

- LABWORKS Excel Calculations

As LABWORKS loads analysis codes for results entry, it inspects each analysis to determine if it has calculation inputs from the XLPARMS table and if all of the defined inputs are present. After results are entered for an analysis, the software rechecks the sample to determine if all inputs are completed and then performs the calculation.

Results Data

- Data Tables (see next section)
- SAMPLE
- ANALYSIS
- LABOBJCOMMENTS
- RESULTS
- LABOBJSPECINFO
- SUSERFLDS
- AUDITTRAIL

SAMPLE

The SAMPLE table stores data about logged in samples. As a sample is logged in, a unique SampleID is generated and stored in the SAMPNO field. The LOCCODE information is copied into the SAMPLE table with the ability to modify the LOCLIST table during the login process. De-normalizing data makes sure that sample specific information is retained even if the LOCLIST table is modified.

Required Fields: SAMPNO

Foreign Key: LOCLIST.LOCCODE ***ANALYSIS***

Tests assigned to logged in samples are stored in the

ANALYSIS table. Required Fields:

- SAMPNO
- ACODE

Foreign

Keys:

- SAMPLE.SAMPNO
- ANL_LIS

T.ACODE

COMMENTS

Sample and analysis comments are stored in the COMMENTS table. The value of the field ACODE is "_SAMPLE_" for sample comments and the actual analysis code for analysis comments.

Required Fields:

- SAMPNO
- ACODE
- Foreign Keys:
- SAMPLE.SAMPNO
- ANL_LIST.ACODE (Analysis

Comments Only) LABOBJCOMMENTS

The Narrative Text Results (LABOBJCOMMENTS) contains the results of narrative tests (!XXXXX). These are free form text results for tests requiring a narrative, such as an observation.

RESULTS

This is the main results table. It stores results and violation information. For single component tests, the Analysis Name (not code) and ANALYTE field are the same. For multi-component tests, the ACODE and ANALYTE fields contain their described values.

Required Fields:

- SAMPNO
- ACODE
- ACODE

Foreign

Keys:

- SAMPLE.SAMPNO
- ANL_LIST.ACODE
- ANALYTES.ANLNAME (Multi-

component tests) LABOBJSPECINFO

This table stores information from the special information fields.

Required Fields: SUSERFLDS

The sample user fields, SUSERFLDS, table stores user defined information common to all location codes and samples. The USERFLDS table stores the default user field data for a location code, and the SUSERFLDS stores the User Field data associated with a sample.

Required Fields:

SAMPNO Foreign Key:

SAMPLE.SAMPNO

SYSMGR/System Manager Settings

System Manager Options added/updated in LABWORKS 6.4

SYSMGROPT	Description	Type	Values	Notes
Sample Login				
LOGBATCHIDFORMAT	Possible Login ID formats	IT	MMDDYYHHNNSS MMDDYY_HHNN MMDDYY MMDDYYYY MMDDHHNN YYMMDDXXX YYMMXXXXX YMMXXXXX YYMMXXXX XXXXXXXX YYYY-MM-DD-XXX MMDDYYXXXX XXXXXX USR-YYYY-MM-DD YYJJJXXX Default is: yyMMddhhmmss	Supported login batch id formats.
LOGINBATCHID_MODIFY	Permit the user to modify the login batch format	IB	0: FALSE -1: TRUE	
MAX_SAMPLES		IL	0-1000	Login Screen number of rows.

PG_ALLOW_AD_HOC_GROUP_PARAMETERS	Controlling visibility of “Add Unspecified Parameters” checkbox in Parameter Configuration screen in **Sample Login** application	Boolean		
PG_ALLOW_AD_HOC_GROUP_PROPERTIES	Controlling visibility following controls in **Sample Login** application 1) “Add Unspecified Documents” checkbox in Document Object Configuration screen 2) “Add Unspecified Documents” checkbox in Special Information Configuration screen	Boolean		
New Sample Login				
SAMPLELOGIN_MAX_SAMPLES	Number of rows to be displayed in Sample Login Screen of **Sample Login** application.	Numeric		
Sample Login and Process Scheduler				
ADD_TO_PROCESS_SCHEDULER	PS Add Sample during login	IB	-1: True 0: False	Adds sample to process scheduler during login
LOGIN_DEPT	ProcessScheduler department for samples from Login	IT	String	Department for samples from login to use for process scheduler
LOGIN_INITIAL_STATUS	ProcessScheduler initial status for samples from Login	IT	0	Initial Process Scheduler status for samples from Login
LOGIN_PRIORITY	ProcessScheduler Priority for samples from login	IL	Value matching entry in PSPRIORITY TABLE	
Results Entry				
ALLOW_CALC_RESULT_MANUAL_ENTRY		IB	-1: True 0: False	Permit users to manually enter results into a calculated field. Default is False
WebTop				

BARTENDER_DOMAIN BARTENDER_PASSWORD BARTENDER_USERNAME				WebTop: Account for running Bartender Application
LABELPRINTERSEARCH		IT		WebTop: Name of label printer to search from in users printer list.
SHOWCRYSTALEXPORT	Controlling visibility of Export button of Report viewer page in **WebTop** application	Boolean		

Refer to "*Appendix A: The Complete List of System Control Options in SYSMGR/System Manager"* for the complete list of all system control options available in System Manager in LABWORKS 6.5.

System Manager Options added in LABWORKS 6.5

SYSMGROPT	Description	Type	Values	Notes
SL_ACODE_LIMITING	Controlling Analysis codes not to be displayed in **Sample Login** application	String		
SL_LOCCODE_LIMITING	Controlling Location codes not to be displayed in **Sample Login** application	String		
SL_VALID_DAYS_BACK	Controlling Submit, Collection and Due dates to be go number of days back in **Sample Login** application	Numeric		
SL_VALID_DAYS_FORWARD	Controlling Submit, Collection and Due dates to be go number of days forward in **Sample Login** application	Numeric		

Refer to "*Appendix A: The Complete List of System Control Options in SYSMGR/System Manager"* for the complete list of all system control options available in System Manager in LABWORKS 6.5.

2. LABWORKS Sample Login - Sample Reporting ("Paperwork")

Introduction

This document is a supplement to the user manual of LABWORKS Sample Login that focuses on the reporting aspects of the system. Upon configuration of a login template, the user can begin logging in samples and generating needed reports including sample receipts, analysis worksheets, and department route sheets. The system is installed with several example reports (in Crystal Reports format) that can be easily modified for the specific needs of the lab.

The sample reports generated at sample login is generally referred to as "paperwork" although no usage of paper is required. The reports can be automatically printed to a default or specific printer, but they can also be created as a file (in PDF, Word, or Excel format) and/or emailed instead.

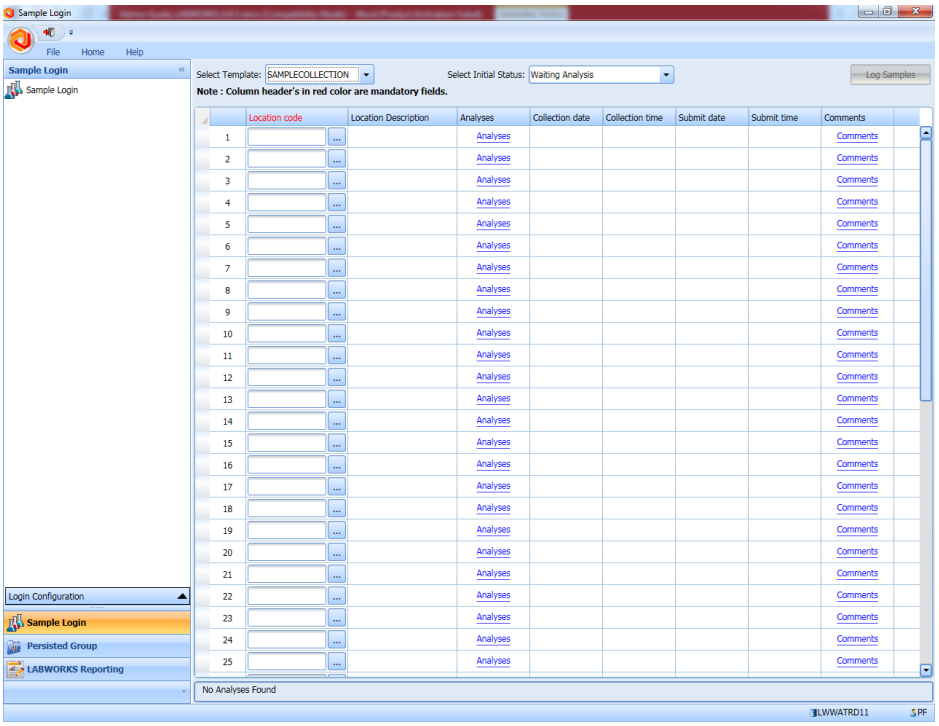
The report templates used in the LABWORKS Sample Login as of version 6.4 and later is all based on SAP Crystal Reports. These reports utilize what is called a *data model* as their data source. These data models are a user-defined subset of information in an XML structured format generated at runtime. The user never has to adjust the default data model if they are happy with the amount of data being reported. However, advanced users or IT/Administration people have the capability of quickly adding or removing detail in/out of any data model if desired. The disconnected data model from the core database allows users to keep reports "canned" without the risk of reporting problems each time a change is made to the database. Report generation performance is high due to the standard XML data source being generated and passed to the report via memory only (no temp files are ever created).

LABWORKS Sample Login

The Sample Login module in LABWORKS is used to log in samples for later analysis / results entry, validation, and ultimately final reporting. The initial launch of the module opens a generic sample login template with only the few required fields. Initially you will need to configure a template to add your own data fields needed by the lab to capture the required information on samples at the point of login.

The main Sample Login screen has a user interface that conforms to the new Microsoft user interface standards introduced in MS Office 2007. There are three basic areas in the main screen used for navigation around the module. The top portion of the screen is the ***main menu*** of the application allowing the user to scroll through a number of different options related to the selected area below. The lower left portion of the screen is called the ***navigation pane*** which allows you to switch among any different areas (separate applications) inside the module related to sample login. These other areas of the modules include configuration tools for sample login templates, paperwork report settings, data model configuration (for reports), or defining persisted sample groups. Finally, the center-right portion of the screen is the ***contents*** area of the selected application. The top left portion of the screen under the menu always shows you the area/application you currently have selected.

When you first open Sample Login, the last selected login template is opened and ready for use. If you want to configure any settings, you would click on the appropriate configuration module to the lower left. Otherwise you can begin the login process by assigning samples.

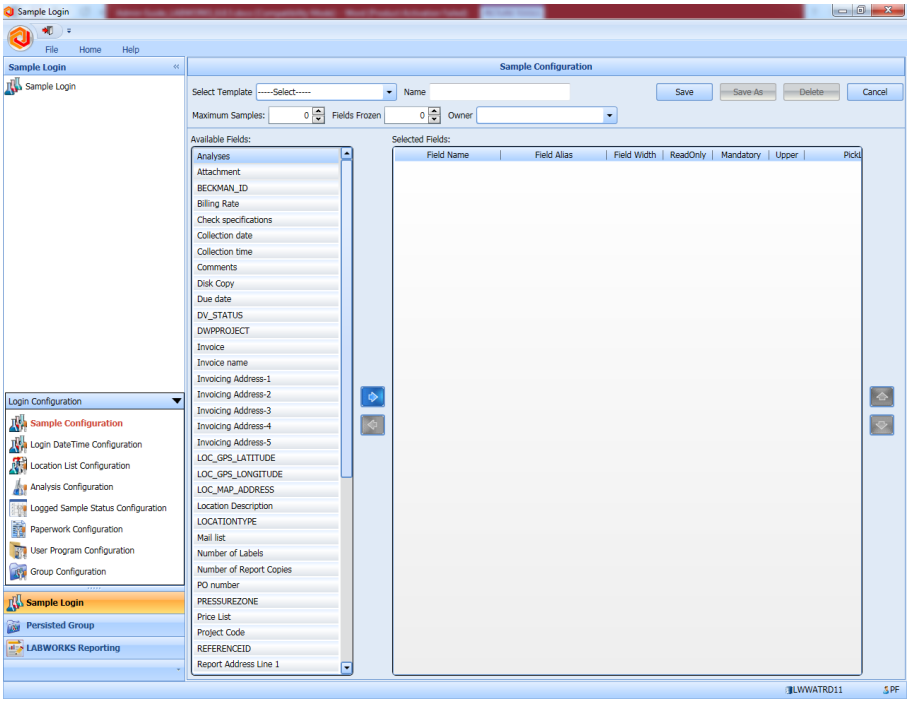


LABWORKS Sample Login – Main Screen

Sample Login Configuration

As mentioned previously, you are allowed to configure as many login templates as needed by your organization in order to capture and track the necessary data throughout the system. You are able to add any field from your location codes listing into your custom sample login template. Keep in mind, the default values you've assigned to your location codes will be stored in your samples at login. It is only necessary to add fields to your login template that your users plan on possibly modifying, entering, or simply checking at login. You can make certain fields (columns) required as well forcing your sample login users to enter (or double-check) specific fields prior to initiating the login process.

Sample Login Configuration is a separate area inside the *navigation pane* on the lower left portion of the screen. Click the **Login Configuration** option to configure an existing, delete, or add a new Sample Login template. You must also access this area if you want to configure any other area of the system such as the desired fields to be displayed inside the locations list, analyses list, paperwork (reports) settings, user program configuration, etc.

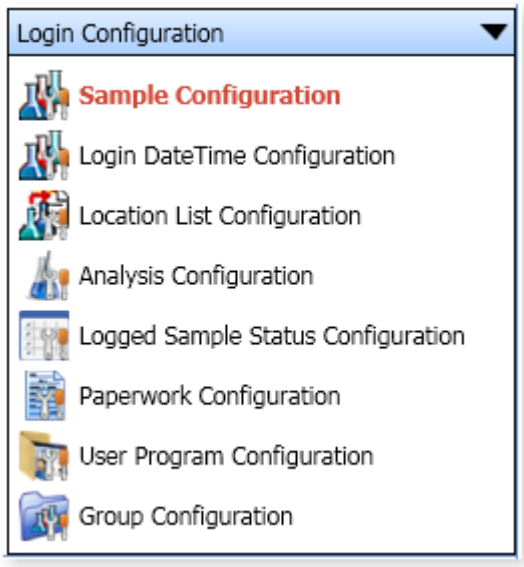


LABWORKS Sample Login Template Configuration

The Login Configuration tool gives you the following options.

- **Sample Configuration** (Selected by Default) – the sample login template modification area allowing the user to add/remove fields to existing login templates, remove, or add new templates.
- **Login Date Time Configuration** – the configuration area for how all date/time fields are processed in login templates.
- **Location List Configuration** – the configuration of the location code selection list. This selection screen is displayed when adding new samples to your login template. You can use this area to select the fields for display.
- **Analysis Configuration** – the configuration of the analyses list for assignment of tests to location codes. This selection screen is displayed when clicking the **Analyses** link to show the assigned test list within a specific sample. You can use this area to select the fields for display.
- **Logged Sample Status Configuration** – this area will configure the screen that shows the list of samples (the logged sample batch) *after login* has completed and before paperwork (reports) is generated. By default only the assigned sample IDs are displayed on this screen. However, you can make any adjustments to show detailed information for the logged batch as well as print the list as well.
- **Paperwork Configuration** – this area will allow you to assign specific login paperwork report templates (Crystal Reports) to the system for generation after login. You can assign any one of a number of sample receipts, analyses worksheets, or department route sheets. Note that this area only assigns the templates. If you want to configure the templates or settings themselves, you need to access a different area in the *navigation pane* called **LABWORKS Reporting**.

- **User Program Configuration** – this area will allow you to assign external programs to run after samples have finished the log in process. The sample batch information can be passed to these programs using command parameters for integration of 3rd party tools.
- **Group Configuration** – this area will allow you to assign a persisted group (previous defined) to a particular Sample Login Template. If you want to configure the persisted group itself or add/remove groups, you need to access a different area in the *navigation pane* called **Persisted Group**.



Login Configuration Options

Sample Configuration

Clicking the first option under Login Configuration will bring you to the sample login template configuration, or Sample Configuration for short. Use this option to select from all available fields on the left to assign to your template. Fields assigned to your template will be displayed on the right.

The top of the application contents area shows the current template selected in the dropdown box. If you select the **Edit Configuration** option, you will be able to pick from a list of available login templates. Initially, you won't have one, so you'll need to select **New Configuration** first. You can then start adding fields to your template on the left by *double-clicking* or using the arrow keys.

The arrow keys to the far right allow the user to change the column order of the fields at login. Fields at the top are closer to the left-side of the login sample listing. For example, the first field at the top of this list (on the right) will be the first column after the required columns (Location Code, Analyses, and Comments).

After selecting all the fields desired for your template, you can also set a few other selections that define how the field (column) operates at login.

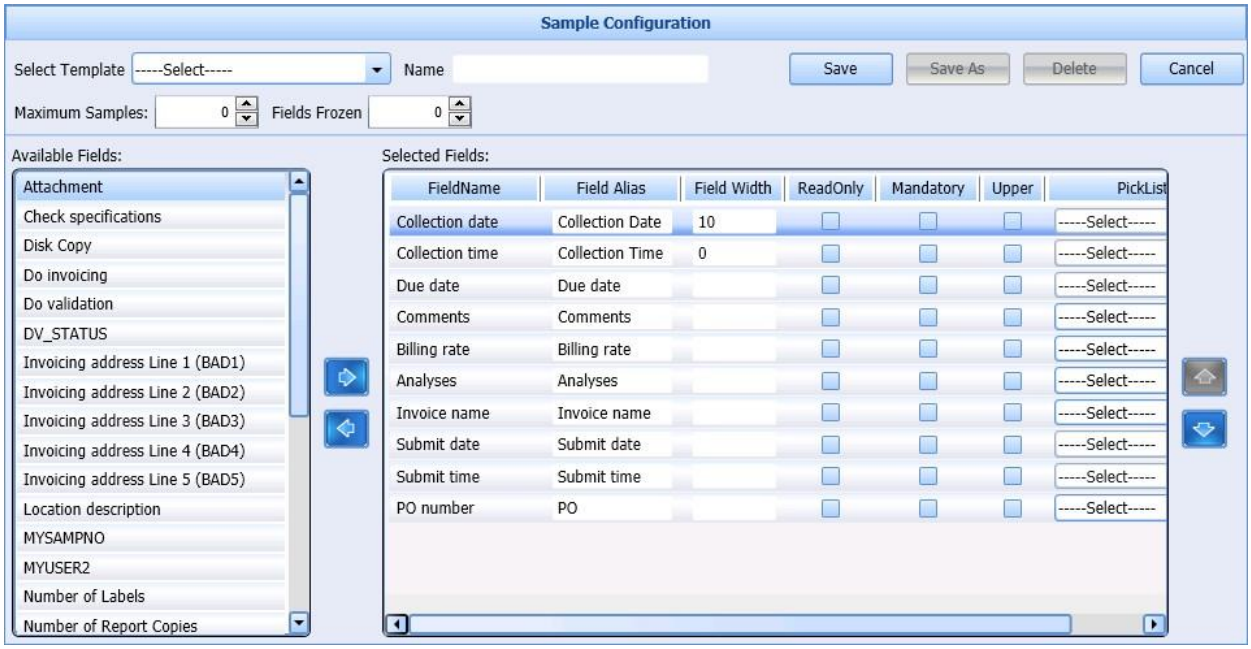
1. **Field Alias** – the displayed name of the field (column header). By default, the column header on your login template will be the same name as the actual field, but you can change it here.
2. **Field Width** - the minimum width of each field. If you set this value as 0 (zero) for a field, it will be available on the template, but the field will be hidden by default in the sample login window.
3. **Read Only** – this checkbox, if selected, will show the default values as samples are added to the

login template but not allow the user to make any modifications of the values in the cells.

4. **Mandatory** – this checkbox, if selected, will force a value in the field cell. The user will not be able to leave it blank.

5. **Upper** – this checkbox, if selected, will force the values to be in upper case.

6. **PickList** – this allows the user to select from available picklists in LABWORKS and assign them to this column. If a picklist is assigned, the user can dropdown a list of values for selection.



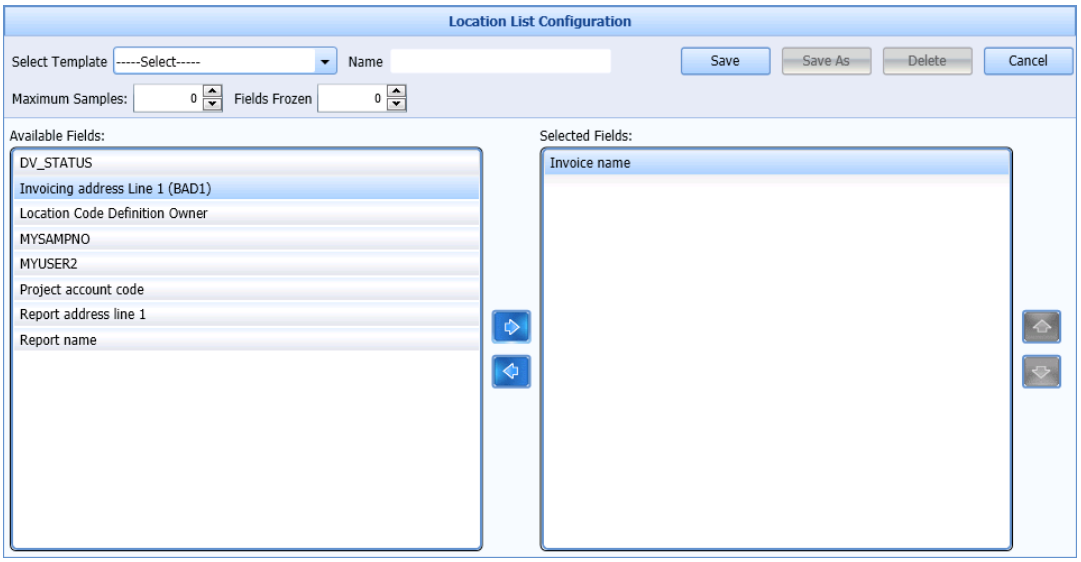
Sample Configuration

When you have your order of fields and selections exactly as needed, click the **Save** button at the top to store your new or modified template to the database.

List Screen Configurations

The screens in the Sample Login module that lists data for selection (samples displayed after login, location code lists, and analyses list) all can be configured much the same way as the Login Template itself in the previous section.

The **Location List Configuration** option on the left will open the current settings for the location code selection screen. If you need to change the fields displayed when selecting samples for your template (called the Location Codes list), this is the option that will accomplish that.



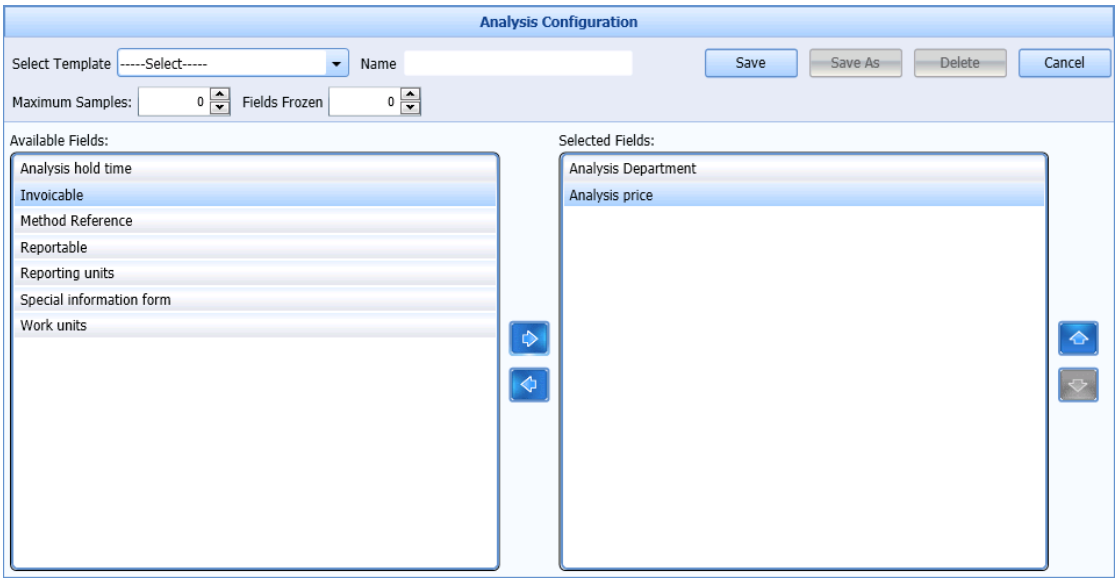
Location List Configuration

In this configuration screen, the selection of fields works the same as previously explained. Double-click or use the arrows in the center to assign fields (or columns at run time). Using the arrows on the right changes the order of the columns in the list allowing you to move fields farther up (or more to the left at runtime) or farther down (more to the right at runtime).

The top of the configuration screen shows the Login Template this List Configuration will be assigned to. When you are satisfied with the order of the fields (columns) click **Save**. The next time you select a sample to log in from the list of location codes in this *Login Template*, your new configuration will be used.

The **Analysis Configuration** option on the left will open the current settings for the analyses selection screen used to modify or assign tests to samples at login. If you need to change the fields displayed when selecting tests for samples, this is the option that will accomplish that.

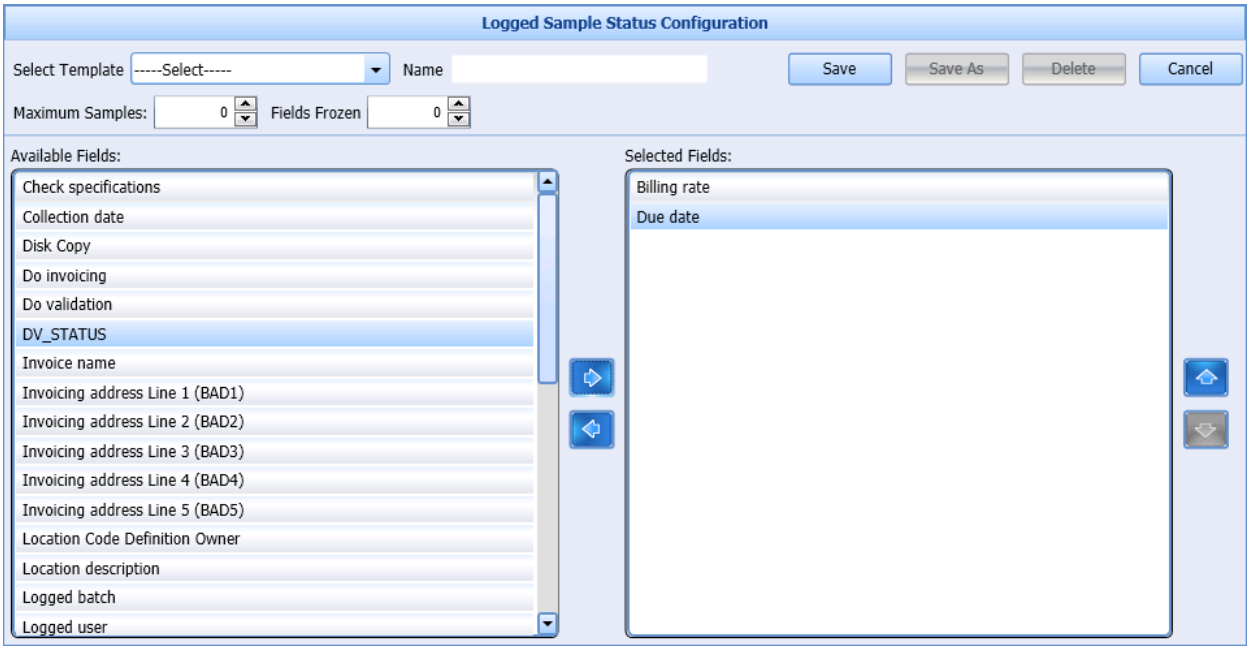
This configuration screen, once again, works the exact same as the Location List configuration. The available fields are on the left and assigned fields on the right. Double-click or use the arrows in the center to assign fields while using the arrows on the right to change the order.



Analysis Configuration

Once again, the **Save** button at the top will store your modifications to the database once completed.

Finally, the **Logged Sample Status Configuration** will adjust the fields on the screen that shows the list of samples in the newly created batch immediately following a successful login. As with all other lists, the default only shows the required fields. In this case, it will display only a list of sample IDs. However, the user may want to show more detail about the newly logged in samples.



Logged Sample Status Configuration

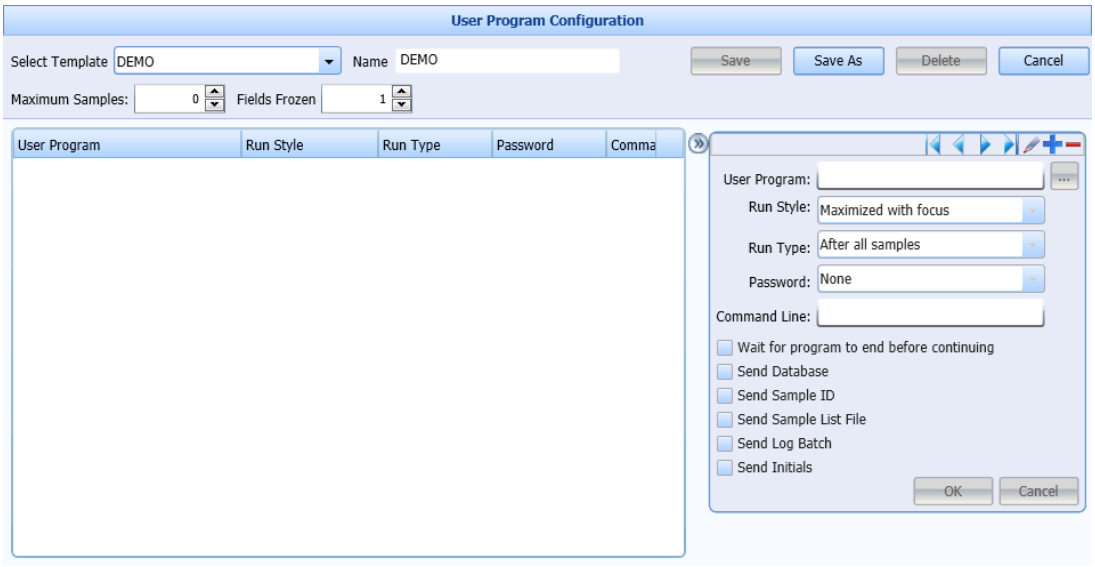
This configuration screen, once again, works the exact same as the Location List and Analyses list configuration. The available fields are on the left and assigned fields on the right. Double-click or use the

arrows in the center to assign fields while using the arrows on the right to change the order.

Click the **Save** button at the top to store your modifications to the database once completed.

User Program Configuration

Clicking the option under Login Configuration called **User Program Configuration** will bring you to a screen for setting up user programs (3rd party applications) to run after all samples are logged in.



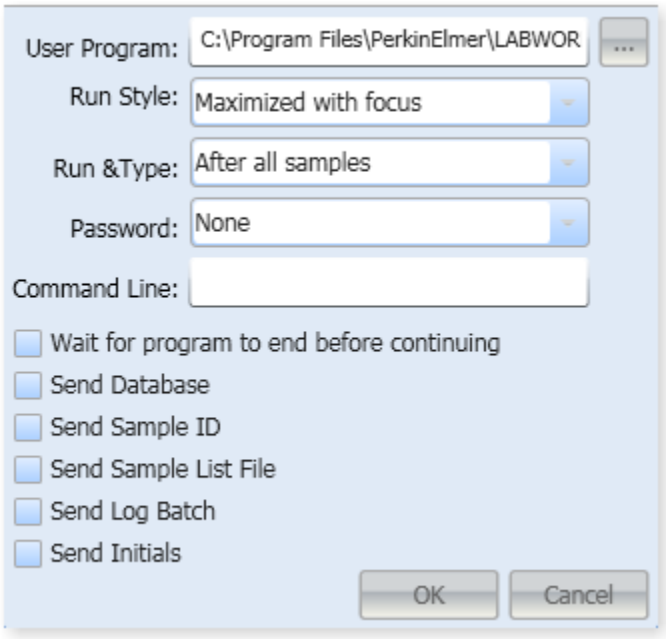
User Program Configuration

As with other configuration areas, the top portion of the center contents section shows the selected template this user programs list is assigned to. The user program list below shows the programs (in order) that will be launched immediately following sample login. The right-side of this section shows the configuration options available for the selected user program.

To remove a selected user program, use the red **X** displayed to the right of the selected program row. Each row represents a program that will launch after login. For all other options (order change, add, modify, etc.) use the toolbar to the top of the options on the right.

Options Toolbar

The options toolbar can change the order (move the program up or down in the execution order list) using the arrow keys, the pencil icon will modify a user program configuration line, the plus button will create a new row for a program, and the minus sign will also remove a program row.



User Program Configuration Options

The first box in the configuration area shows the program path and filename. Use the ellipse button **(…)** to browse the PC or network to find a program to run.

The next box is the **Run Style** which defines how the program is launched. You can have a program launched in the background or "silent" or force it to run in the foreground with focus.

The next box is the **Run Type** which defines *when* it runs during the login process. You have the option of running this program after each sample or all samples are logged, or even before login takes place when samples are posted to the database. Which option you select here depends on what the program does. Maybe this is an interface program that requires some preliminary steps to be done prior to the samples hitting the database, or something that runs after all samples are logged in to further process a batch in another system. Whatever the process, this option defines at which point the program will be launched.

The next box allows the user to define a **Password**. If this is a module that requires authentication into LABWORKS first, the program will pass the currently logged in user's password to the program being launched using the "PW:" command parameter. Any additional command parameters are passed to the program using the **Command Line** option following the Password.

Other options below can be selected based on the needs of the program. You can send a variety of information such as the name of the **Database**, the first **Sample ID**, a **Sample List** file (a text file with a list of all sample IDs just logged in), the **Log Batch** number, or the **user Initials** (user ID). All are passed to the program with specific command line abbreviations in front.

DB: Database
ID: Sample ID
FILE: Sample list file
LB: Log Batch number
USER: User ID

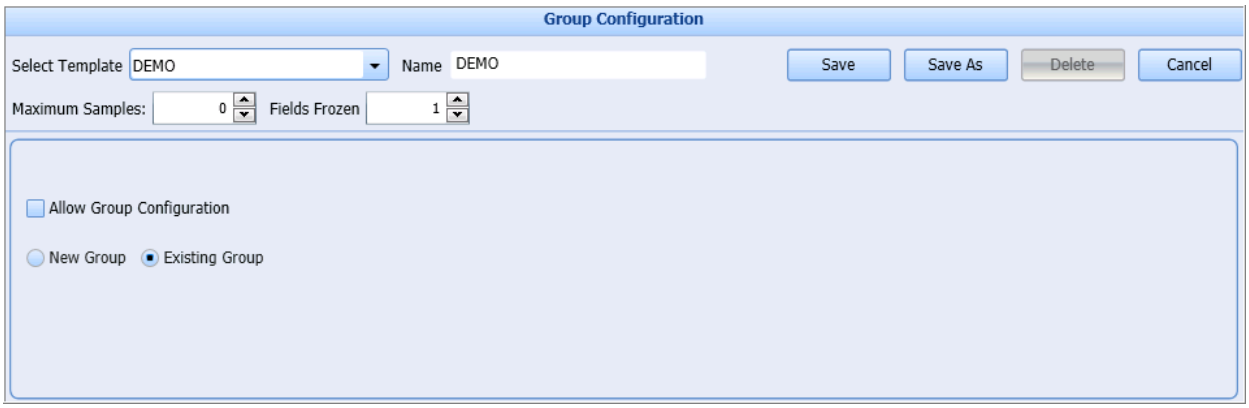
If you want your program to finish its job and go to completion before Sample Login continues to the next

program in the execution list, select the "**Wait for program to end before continuing**" option at the top.

Click **OK** to save the User Program configuration. If it is new, it will be added to the list.

Group Configuration

Clicking the **Group Configuration** option on the left allows the assignment of a persisted sample group to the assigned Login Template.



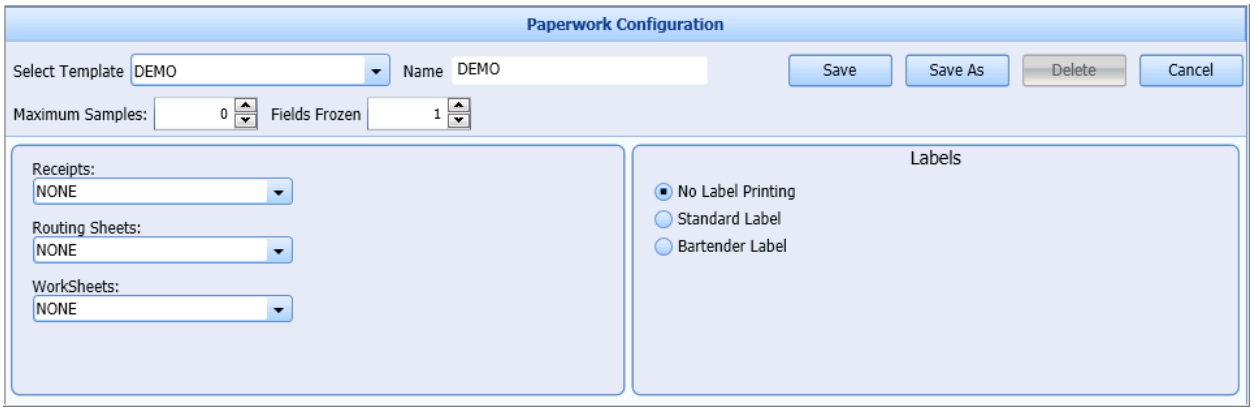
To setup or configure groups, reference the Persisted Groups section.

Paperwork Configuration

Clicking the Paperwork Configuration option will bring the user to a screen for assignment of paperwork templates (Crystal Reports) to the receipt, worksheet, or route sheet areas. It also can configure any labels required to be generated for bottles.

The "paperwork", as it is referred to, are generally documents that are required to be rendered (printed or otherwise) for all samples at login. Sample Receipts showing price information is commonly used for contract labs, for example. Department Routing sheets are often used for larger organizations with different labs testing different analyses, so the paperwork showing their department samples are sent to them. Analysis worksheets, for example, can be used by operators at chemical plants to gather readings or results on paper for specific tests in a sample. Often operators will write down these values in the production area where computers are unavailable or not permitted. The filled-out worksheets are later used for results entry. All of these are examples of "paperwork" generated after sample login.

By default, there is no paperwork set up to be generated at login. The user is required to go into this configuration area and assign specific templates.



Paperwork Configuration

As with other configuration areas, the top shows the selected template this paperwork corresponds to. There is also an option to designate the **maximum samples** allowed for this paperwork. The user may not want to generate paperwork on large logins for performance purposes or to conserve paper, so they can set a limit here.

The middle right area allows you to define your labels. By default there are no labels generated at login. You can assign the standard (built in) labels or use Bartender (if installed) to generate your labels. Bartender is a 3rd party tool by Seagull Scientific that is integrated into LABWORKS. You will have to predefine your labels in Bartender before using them here.

The middle left area shows up to three paperwork (Crystal Report) templates you can select, one for each paperwork type. There are a number of available templates installed with LABWORKS that can be selected here. You can use Crystal Reports to edit existing templates, remove templates, or add your own from scratch. You do this under the **LABWORKS Reporting** area in the navigation pane on the left. Refer to that section later in this document.

There is only a single report template that is allowed to be selected for each *Login Paperwork Type*. These three types will render differently in the system after the LOGIN process is initiated.

- **Sample Receipts** – these reports are rendered ***once per sample***. The batch of logged in samples are queried by Sample Number (or Sample ID), then a call to the Document Data Provider component (which provides the *data model* to the report mentioned previously) is made to return all data by sample. For each sample the report rendering component is called with the template and returned data model XML. These reports are state managed in Object Publisher under the **CRYSTAL_RECEIPTS** type.

- **Routing Sheets** – these reports are rendered ***once per analysis department***. The batch of logged in samples are queried by Analysis Department (ADEPT), then a call to the Document Data Provider component (creates the *data model*) is made to return all data grouped by department. For each department the report rendering component is called with the template and returned data model XML. These reports are state managed in Object Publisher under the **CRYSTAL_ROUTESHEETS** type.

- **Work Sheets** – these reports are rendered ***once per analysis***. The batch of logged in samples are queried by Analysis Code (ACODE), then a call to the Document Data Provider component (creates the *data model*) is made to return all data grouped by analysis code. For each analysis

the report rendering component is called with the template and returned data model XML. These reports are state managed in Object Publisher under the **CRYSTAL_WORKSHEETS** type.

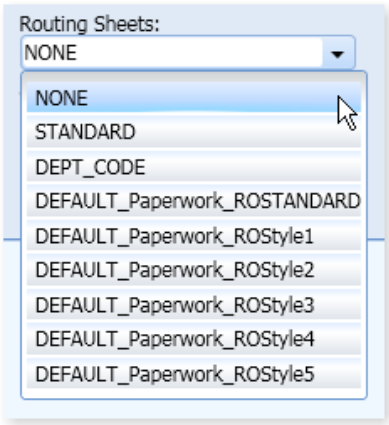
If you do have the State Management option turned on under System Manager settings, any change made to a Crystal paperwork template will cause it to not be available for selection here until it is approved into an ACTIVE state in Object Publisher. Initially, all default templates are active and ready for use.

None or *Automated* Template Selection

Besides the templates themselves in the 3 boxes for selection in **Paperwork Configuration** of Sample Login, there are also two other selections. The user is able to select NONE as an option where no paperwork is generated for that type, or an option to automatically select the template based on the data returned.

For route sheets (where analyses are grouped by department, one department per sheet) the following selections are available:

- **NONE:** will not print any route sheets
- **DEPT_CODE:** will group analyses by department code then look for ACTIVE report template formats with the name = each department code. Analyses go on the route sheet with the name that matches their department code. Analyses where no template is found matching their department code do not (by design) have route sheets printed. For example, for a department of “WetLab”, the system will look for a template of “WetLab.rpt” and place all matching analyses for that department on that report.
- **The template list** – after the first 2 selections, a list of templates will be displayed for selection which are approved and ACTIVE in the **CRYSTAL_ROUTESHEETS** object type. There will always be a “Standard” report in that list. Selecting a specific template here will print only this template for each department.

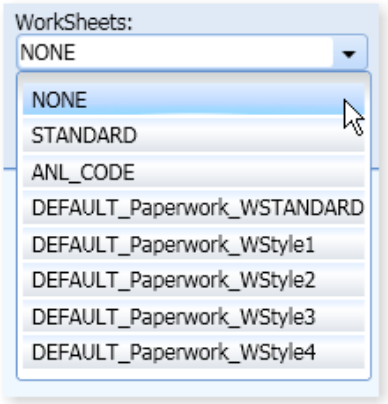


For worksheets (where samples for each analyses are grouped one analysis code per sheet) the following selections are available:

- **NONE:** will not print any worksheets
- **ANL_CODE:** will group assigned analyses by analysis code then look for ACTIVE report template formats with the name = each analysis code. Analyses go on the worksheet with the

name that matches their analysis code. Analyses that do not have a template matching their analysis code do not (by design) have worksheets printed. For example, for an analysis code of “PH”, the system will look for a template of “PH.rpt” and place all matching analyses on that report.

- **The template list** – after the first 2 selections, a list of templates will be displayed for selection which are approved and ACTIVE in the **CRYSTAL_WORKSHEETS** object type. There will always be a “Standard” report in that list. Selecting a specific template here will print only this template for each analysis code.

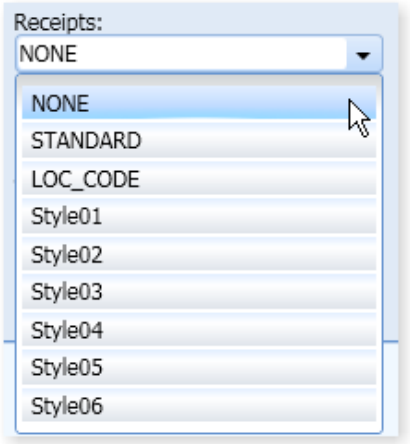


For sample receipts (where all analyses for samples are printed on the same sheet, one sheet per sample) the following selections are available:

- **NONE:** will not print any receipts

- **LOC_CODE:** will group assigned samples and associated analyses by their location code (sample point) then look for ACTIVE report template formats with the name = each location code. Samples go on the receipts with the name that matches their location code. Sample locations that do not have a template matching their location code do not have receipts printed. For example, for all samples with a location code of “Jones Creek”, the system will look for a template of “JonesCreek.rpt” and place all matching samples on that report (1 printout for each sample using that template).

- **The template list** – after the first 2 selections, a list of templates will be displayed for selection which are approved and ACTIVE in the **CRYSTAL_RECEIPTS** object type. There will always be a “Standard” report in that list. Selecting a specific template here will print only this template for each sample.



Note that during the automation, certain symbols and spaces are removed to preserve good file system practices. Symbols not allowed in the file system will also not be allowed here. Although spaces are allowed in the file system, they are filtered. However, the system looks for both versions of the report just in case the space was also used in the report template filename.

Also note that all three paperwork types have a **STANDARD** option as well. This will select the template marked “Standard” in the associated *\CRYSTAL\LOGIN* path. We recommend never changing this template since this is the common basic format. However, you can always model after any template in the system to create your own paperwork templates.

The following sections will detail each aspect of both the *template design* and *assignment* process on through to the application usage and report rendering process.

Sample Login Reporting Overview – The Complete Process

Report template objects (Crystal) used in Sample Login are separate from the standard Crystal templates (located in the \CRYSTAL\ path of the LWDATA folder) used in other areas of LABWORKS. The templates for paperwork are located in the \CRYSTAL\LOGIN\ path of the LWDATA folder and require approve in a separate area of the Object Publisher before they can be used (if new templates are added or existing templates are modified). As explained previously, initially all default templates are fully active and ready for use.

The object types that define paperwork reports in *Object Publisher* are **CRYSTAL_RECEIPTS**, **CRYSTAL_ROUTESHEETS**, and **CRYSTAL_WORKSHEETS**. The Sample Login module will use the specific templates assigned under **Paperwork Configuration** within the application.

A “*Data Model*”, as mentioned previously, is created based on which data areas are needed for the reports to function. The reports utilize these data models at print time. For example, a login paperwork *data model* would not need *results data*, so that information is left out of the *data model configuration*. Both *data model* and *report* configuration is done under the **LABWORKS Reporting** area of the navigation pane.

New paperwork reports must go through a *template creation process* from the design in Crystal Reports, to the placement in the correct location (*\LWDATA\CRYSTAL\LOGIN* area) to *Object Publisher* approval and *data model* and/or *report settings configuration* (optional if needed)…

After a report is created (using Crystal Reports 2008 or later recommended) along with its supported *data model*, it can be approved from *DRAFT* to an *ACTIVE* state into one of the three types used by Sample Login **Sample Receipts**, **Routing Sheets**, or **Work Sheets** as defined previously.

New Paperwork Template Design or Modification

Paperwork templates are designed using Crystal Reports. At least version XI (11.5) is required for designing XML-based reports. However, we recommend Crystal Reports 2008 (12) or later because of the additional features included in data source configurations. The template design process utilizes the installed (off-the-shelf) version of Crystal Reports.

People who are familiar with legacy reporting tools will remember an internal Crystal-based designer was used (called the LABWORKS Report Designer). However, this is no longer used in Crystal 2008.

Unless a SysMgr key is turned on (the **CRYSTAL_VERSION** SysMgr key) by setting it to a value of "XI", the internal Report Designer module will never launch. The "Setup Crystal Report" option on the LABWORKS Desktop, will instead launch the installed (off-the-shelf) copy of Crystal Reports. If the SysMgr key is set, users are capable of using the legacy report designer, however we caution against that if using LABWORKS 6.4 or later versions.

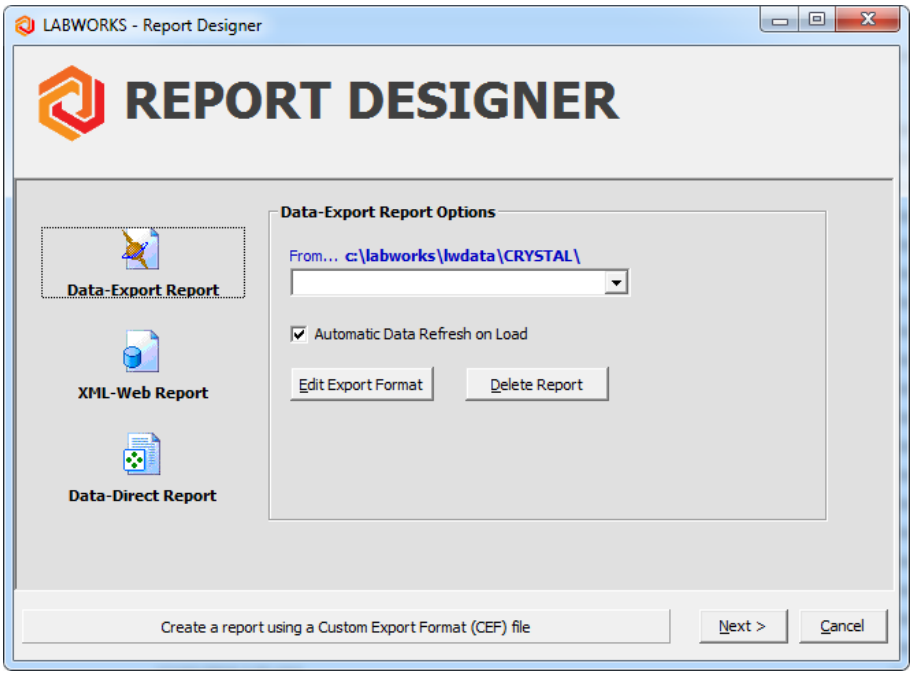
If reports are edited (a right-click option) from the LABWORKS *Object Publisher* of any new CRYSTAL type, then the ***Crystal Reports 2008 package*** will open that template directly for editing (if installed). The template path and filename is passed to the Crystal Reports executable as a command parameter to open the report in design mode at any time. Modification and saving the report, just as before, will drop the RPT file back into a *DRAFT* state.

NOTE: all reports for Sample Login (for all 3 object types) will be physically located under the ***CRYSTAL**LOGIN*** path of the LWData folder.

Note that the legacy "Setup Crystal Report" option is not for Sample Login Paperwork although you can use it to launch a template for modification. Generally the best place to launch a template for modification is from the Object Publisher inside the specific object type list of choice: **CRYSTAL_RECEIPTS**, **CRYSTAL_ROUTESHEETS**, and **CRYSTAL_WORKSHEETS** which are *only* specific to the Sample Login tool.

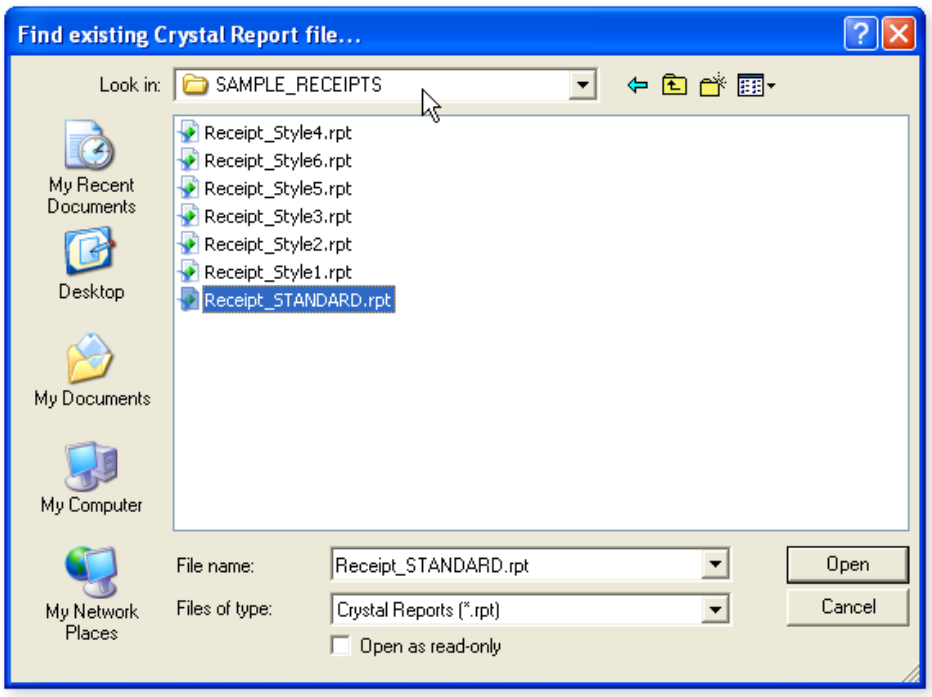
If a user wanted to use the "Setup Crystal Report" option to edit paperwork reports, they could. However, they would have to use the following steps.

1. Click **Setup Crystal Report** from the LW Desktop
2. In the From box, click the ellipse **(…)** button to browse for the report



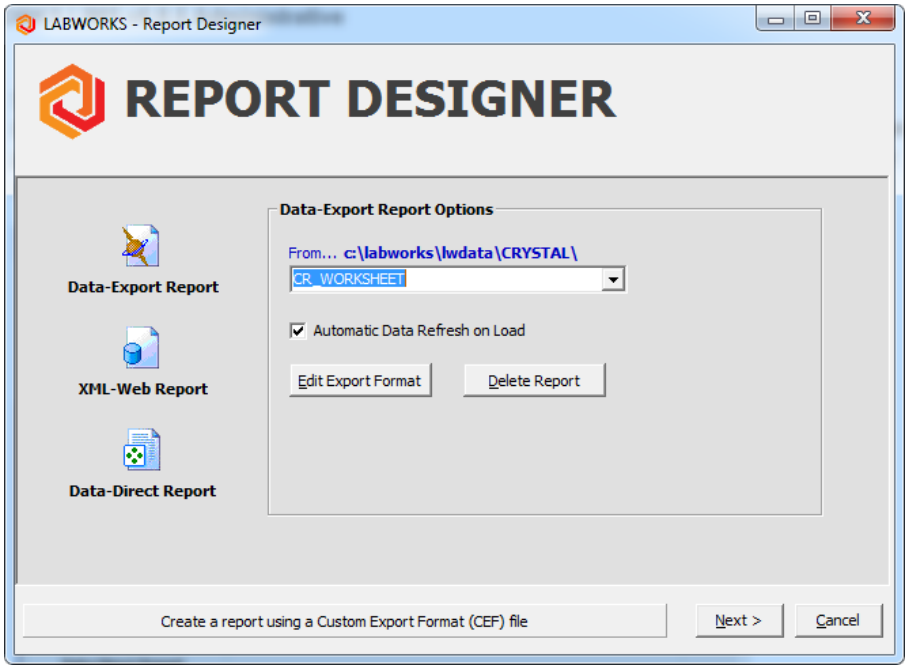
Legacy LABWORKS Report Designer 2008 UI

3. Go into the path where login paperwork reports are stored which is \LWDATA\CRYSTAL\LOGIN\...

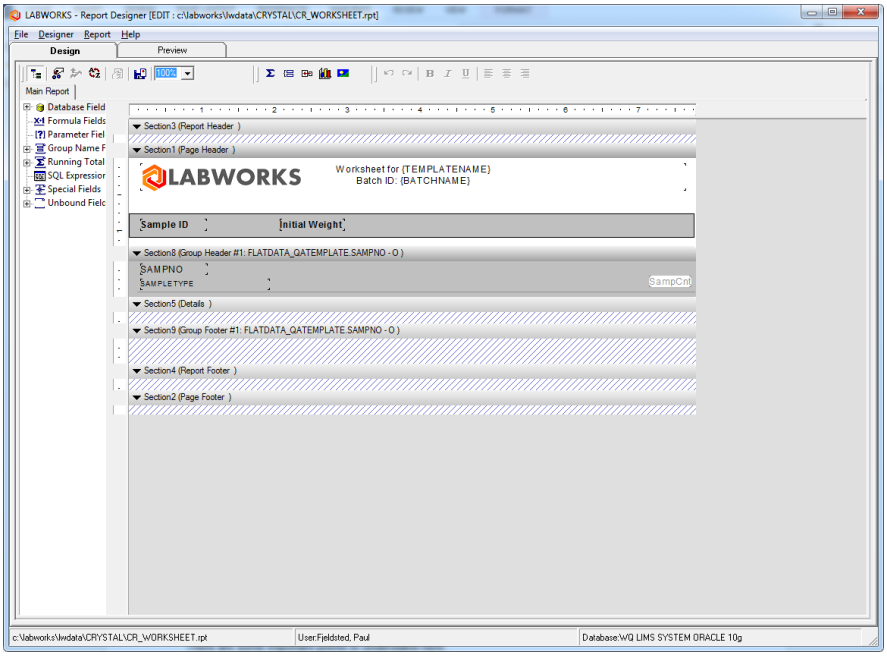


(Note: Inside the LOGIN folder you have a folder for the 3 types of paperwork templates).

4. Select the template you want and click **Open**
   (Note: if this is a new template you wish to create, you can right-click a template, copy it, then paste it right from here, then select it. The new template will later appear in Object Publisher under the specific type where it can be approved into an ACTIVE state for use)



5. Click **Next** to open the template now and make your modifications



Crystal Reports with Paperwork Template Opened

Now you can make any changes needed to the report and **save** your settings before exiting the template.

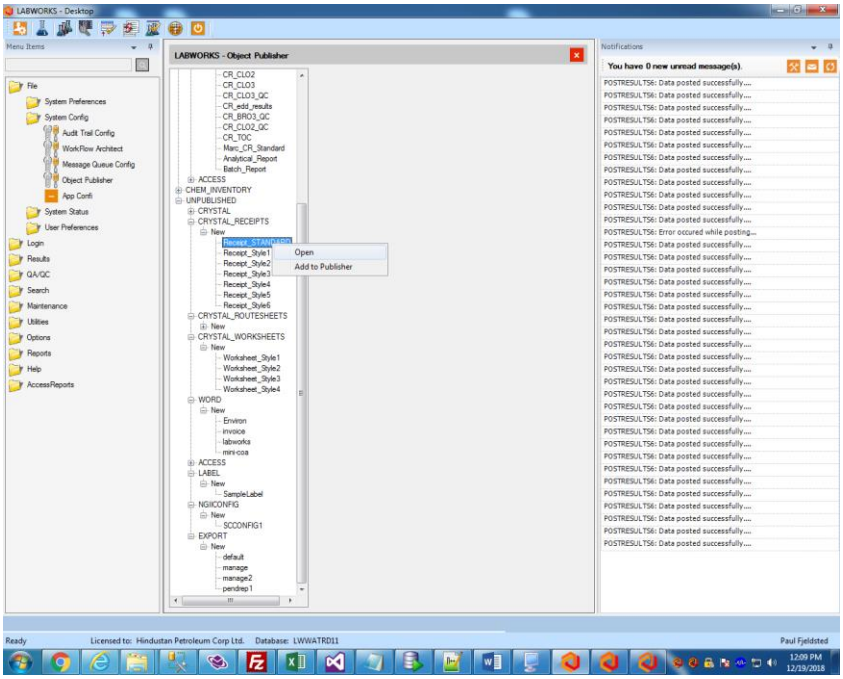
There are some important points to understand here.

- No *data model* is generated from here – Crystal Reports will use the last data saved in the report template (from the last design) allowing you to make quick changes here and there as needed. However, if you attempt to refresh the data source an error will occur. You can access the LABWORKS Reporting area to create a data model XML for report design purposes and store it somewhere for safe keeping (if needed). ***No files*** are ever created during report rendering of paperwork, so you will only need a sample XML *data model* file created for report design purposes.

- Default reports are in *Crystal 2008* – if you do have the **CRYSTAL_VERSION** System Manager key turned on for usage of the internal Report Designer in XI, keep in mind when you SAVE the report it will revert back to the old format. This could cause problems since these reports are based on new technology. However, other forms of reports in LABWORKS are ok to use with Crystal XI, but we recommend only using Crystal 2008 or later for the newer paperwork templates.

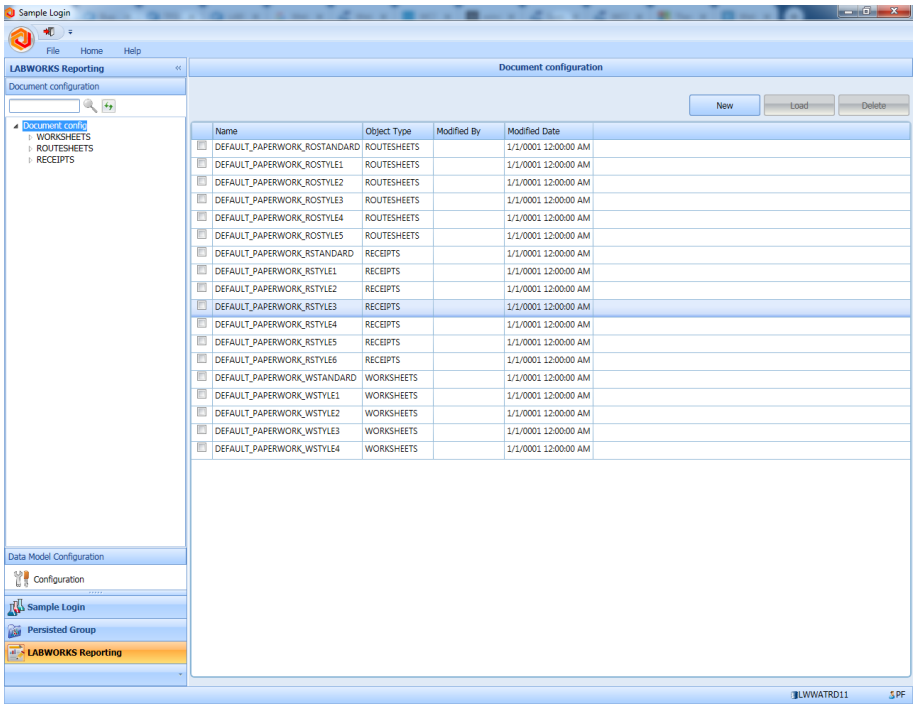
- Editing a template can also happen from *Object Publisher* – as mentioned previously you can also edit any paperwork report by right-clicking and editing it from *Object Publisher* using the **Open** option. If State Management is turned on, then you'll be forced to enter *Object Publisher* to approve the newly changed or created template into an ACTIVE state anyway. So making your template modifications from *Object Publisher* will save you a step. Immediately following editing the template, you can approve it to ACTIVE.

  LABWORKS Object Publisher Report Template Open



LABWORKS Reporting

For any configuration needed to Sample Login paperwork *data models* (data selected for reports) or *report configuration options* to define *how* paperwork is rendered or delivered is all done in the **LABWORKS Reporting** area selectable from the navigation panel.



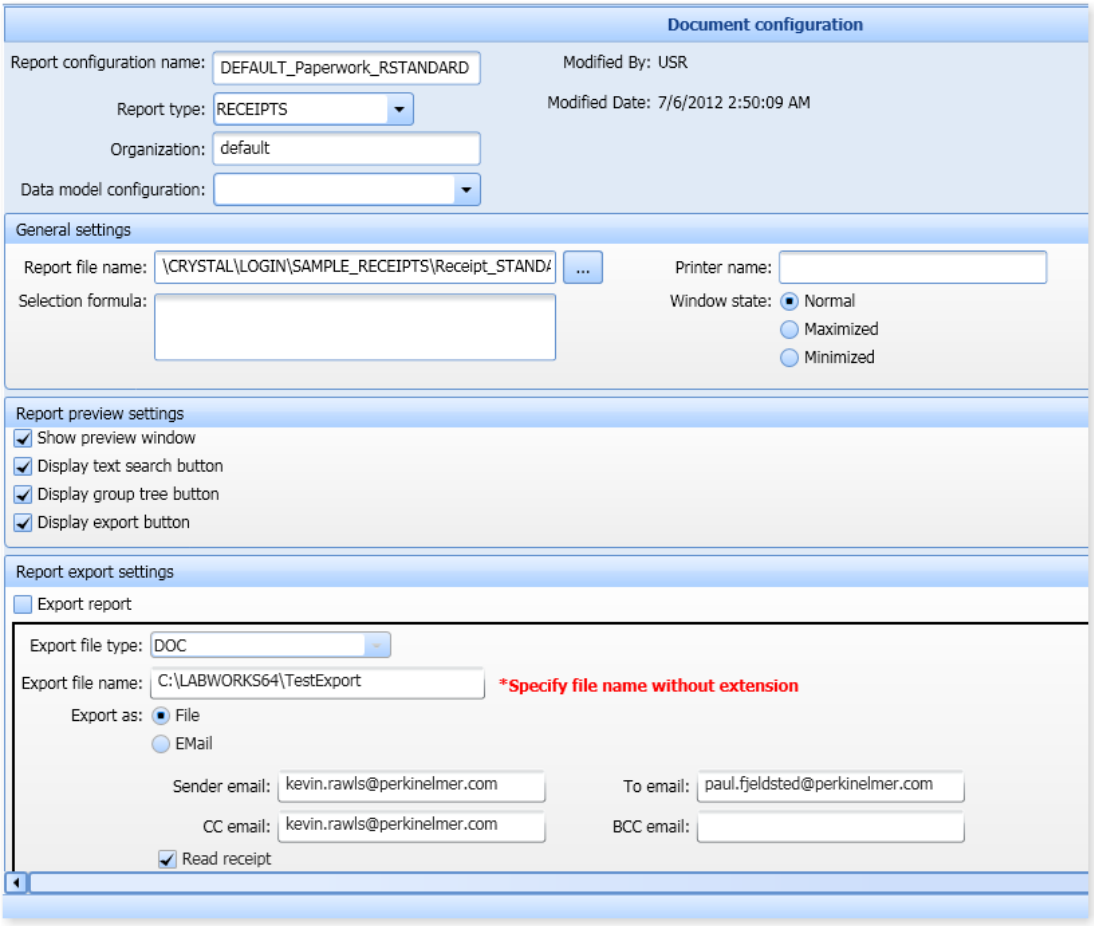
LABWORKS Reporting

The LABWORKS Reporting area inside the Sample Login module shows a list of available paperwork report templates assigned in the system. It is a module strictly designed for use by System Administrators, so it does not show active versus inactive states. All available templates are shown.

There is an area in the navigation pane on the left for Data Model Configuration to set up the report data source. To configure a specific report template, you simple check which template you want in the list and click the Load button. You can also click **New** to add a new report configuration (generally done after creating a new template from Crystal in order to define specific settings needed) or **Delete** to remove a report configuration from the list.

Report Settings Configuration

Once a report settings record is selected from the list (checked) and the **Load** button is clicked at the top, the full configuration of this specific report is displayed.



A **name** for the report configuration is provided at the top. This simply defines the configuration in the database. Keep in mind this configuration record can only be tied to a specific Crystal Report template, but the same Crystal Report template can be added to multiple report configuration records. For example, you may have two report configuration records called “SampleReceipt_Printed” and “SampleReceipt_Emailed” in the system. Both have the same report template file assigned to them. However, you have one configured to generate paperwork to a specific lab printer while the other fires off an email to the supervisor or department manager. Both reports are the same, but are generated differently. Either or both of them can be assigned as paperwork at login.

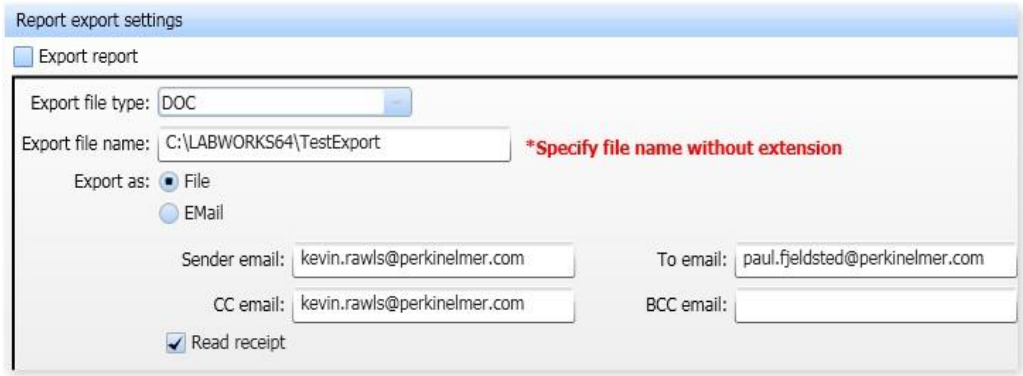
The **Report Type** defines how the report is used and corresponds to the paperwork type, and the **Organization** box is nothing more than a field used to group reports of similar function only.

The **Data Model Configuration** is a required field. You must assign a data model to each report configuration in order to have it function.

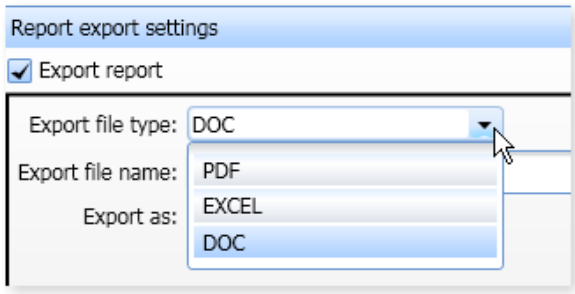
Under **General Settings** is where the physical report template file is assigned. This is the Crystal RPT file located in the *\LWDATA\CRYSTAL\LOGIN* path. You can also add a record **Selection Formula** that is sent to the report at runtime for further data filtering. Defining a specific **printer** (default will be used if left blank) and/or the **window state** (minimized, maximize, etc.) if not printed is also done from here.

Under **Report preview settings**, you define how the window used to display the report will look. Keep in mind the **Show preview window** must be selected before any of the other options will work. The following options simply turn areas of the toolbar on the preview window on/off.

Under **Report export settings**, you can define how a report *FILE* is created, *which format*, and where it is stored or emailed only if the **Export Report** option at the top of the section is checked.



You are allowed to export this report in one of three formats.

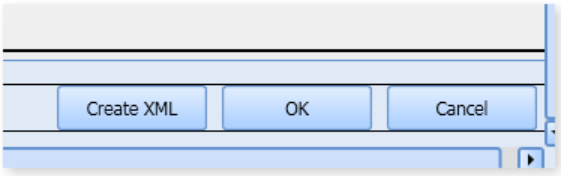


- **PDF** – exports in Adobe Acrobat (*.pdf file) format
- **EXCEL** – exports in Microsoft Excel (*.xls file) format
- **DOC** – exports in Microsoft Word (*.doc file) format

The **Export File Name** box defines not only the *filename* for the exported file (without the extension), but it also expects the full path of the file location as well.

The following settings define how it is emailed (if selected) after exporting. You can assign email addresses where you want this report to go and select to have a **Read Receipt** email returned as well when they open the email.

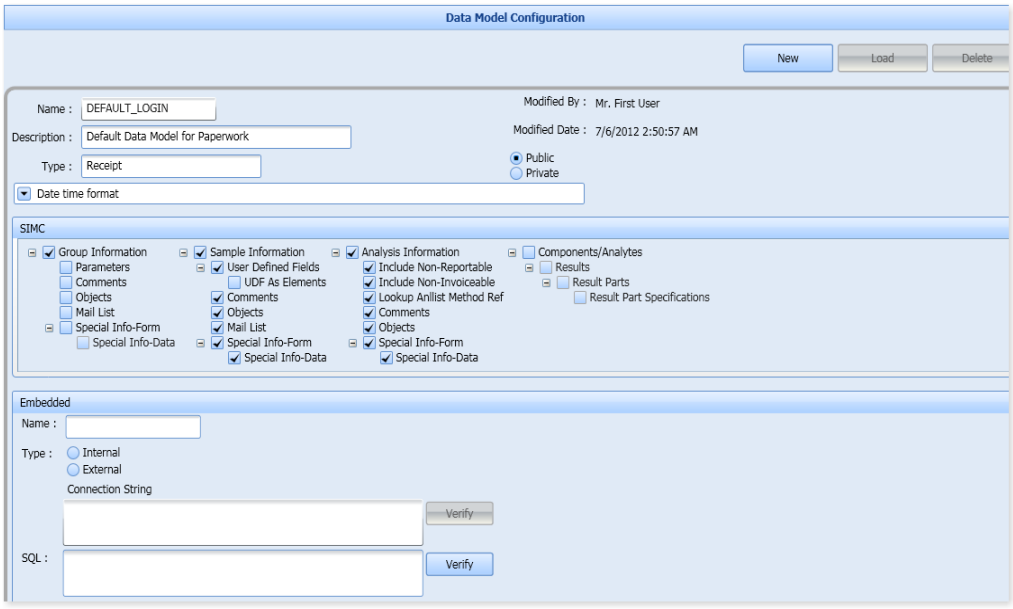
At the bottom of the report configuration record is three buttons. Click **OK** to save the configuration you just completed. **Cancel** will exit without saving.



Note there is a **Create XML** button used to generate the XML *data model* source used for report design purposes as mentioned in the previous section. Prior to report design, we recommend creating the XML that will be used when the report is generated and storing it somewhere. A sample selection box will open once this button is clicked, and the assigned samples are generated into an XML data source and stored in the user path. The System Administrator can build additional paperwork reports from this data model. If the *data model structure* changes at any time, this file should be regenerated from here and the reports that use it refreshed.

Data Model Configuration

The **Data Model Configuration** area in the navigation pane on the left will bring you into the screen that adjusts the *Data Models* for reports. Note that this area is also only for a System Administrator and not recommended for general LABWORKS users. If an area is removed that is used by reports, it could cause errors at report generation time. Make sure only users that know how to use this area accesses it.



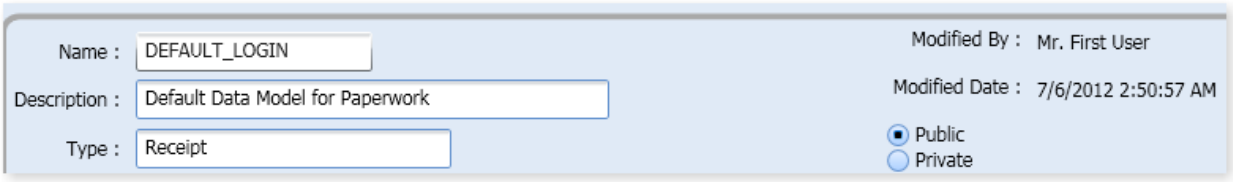
Data Model Configuration

Initially a list of data models is displayed. You can select (check) the specific one to modify and click the **Load** button. Clicking **Delete** will remove a *data model* from the system and **New** will create new data

models from scratch.

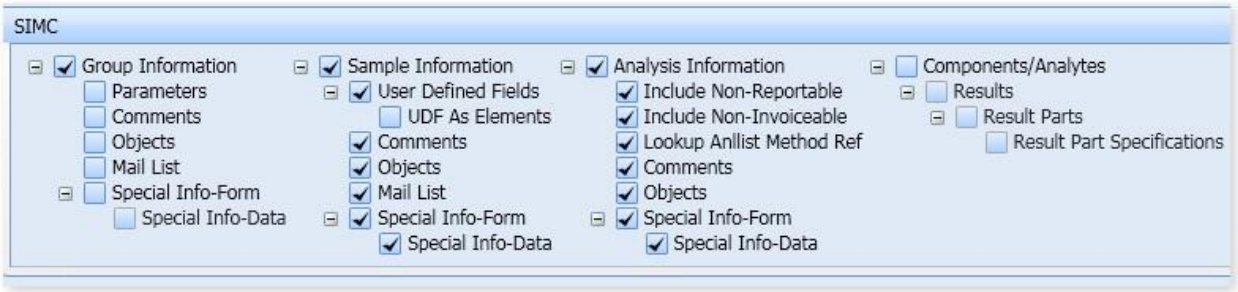
Note: *use caution when deleting a data model*! If there are reports that use the data model you are removing from the system, those reports will no longer be functional.

The top portion of the data model configuration allows the user to define a **Name** (required), a **Description** (optional), and **Type** (also optional, just used for grouping similar data models). It also tells you when the data model was created and who did it.



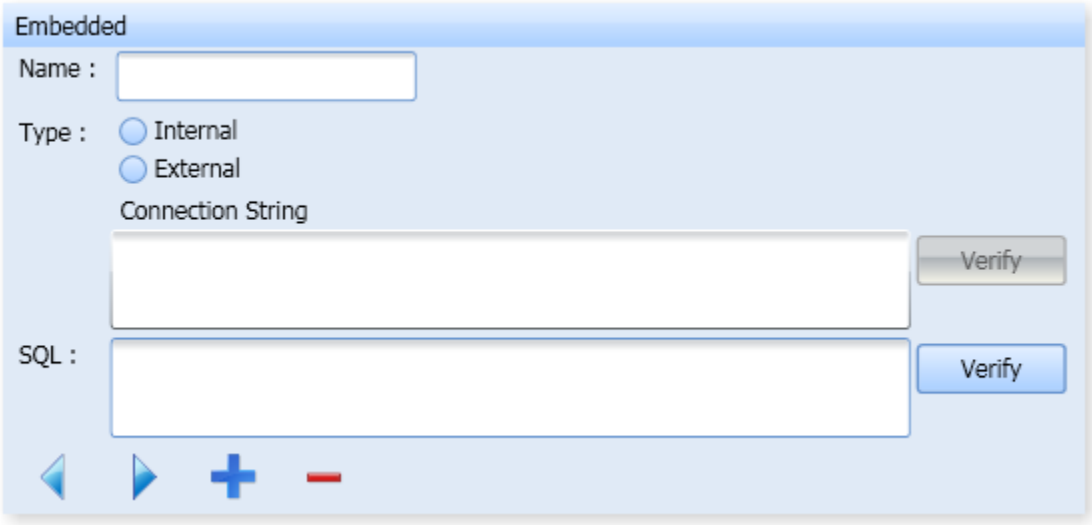
At the bottom right-side of the top section is an option to set the data model to **Public** or **Private**. This will give only specific users access to their data models (if set to **private**) or allow it for use by everyone (**Public** – default option).

The following section defines which data areas this data model will include. For example, as mentioned previously paperwork reports never use results since sample login always happens ***before*** results entry. Therefore, for login paperwork reports, the *data models* always have the **Components/Analytes** results tree node unselected. Other reports in the system, however, may use results (outside of the Sample Login module). In those cases, a different data model will be created and assigned to those reports which include that node.



Note: This is considered the SIMC section since the XML data is created directly from the SIMC (Sample Information Management Component) that holds all sample information (all database tables and fields related to sample information) in the system. The SIMC controls the flow of sample information between all modules in LABWORKS.

Finally, the last section (for more advanced configurations) allows a user to add a SQL string that can point to either the internal LABWORKS database or an external database. This can be used to query data that may not have anything to do with LABWORKS specifically, and include that as XML in the same *data model* data source for the reports. This is a powerful feature allowing a streamlined interface into external system tables to be included in the same data source for reports.

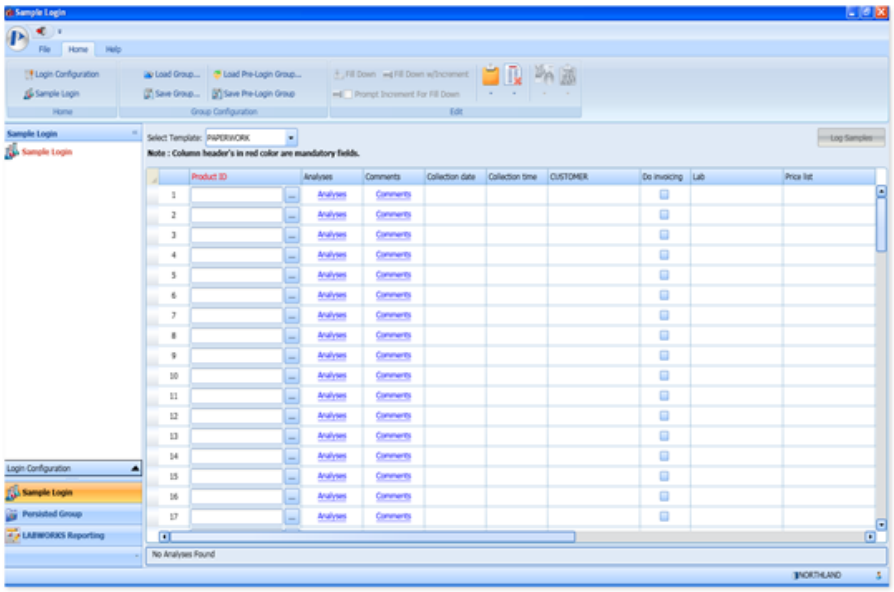


When you are satisfied with your data model changes, click the **OK** button at the bottom to store the changes to the database. Or, you can click **Cancel** to quit without saving.

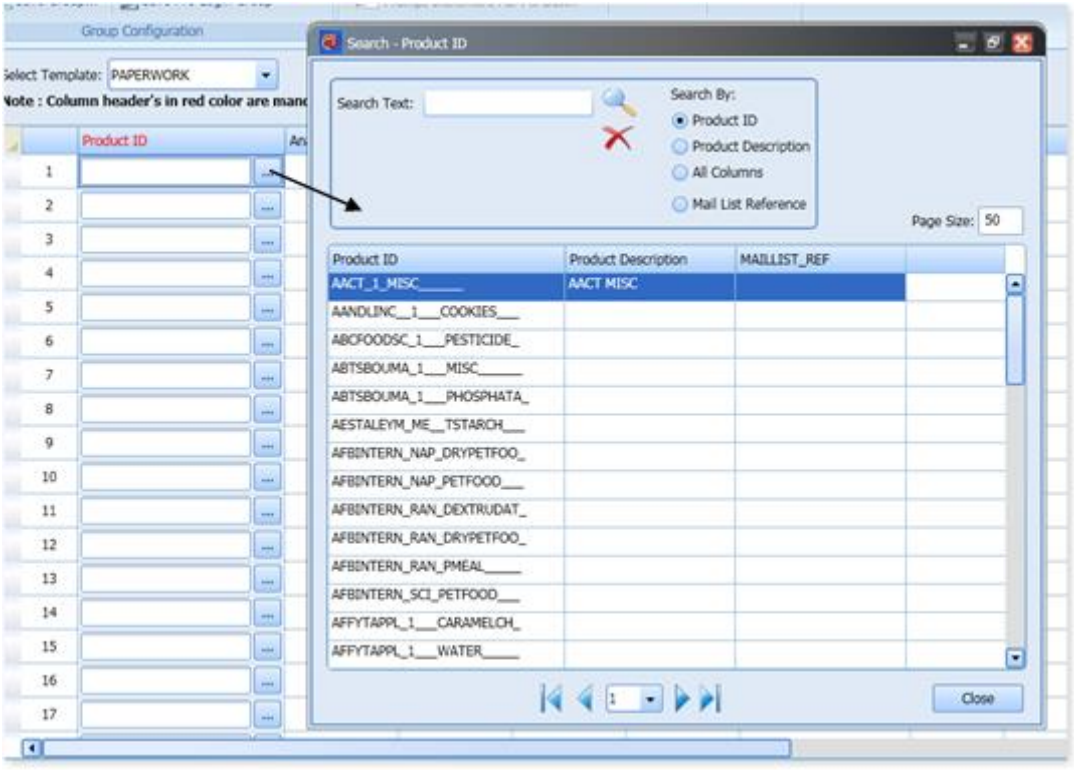
LABWORKS Sample Login – Logging in Samples

After any and all template and paperwork configuration is completed, you can log in your samples. The following steps will define a simple process of logging samples into the system and generating paperwork.

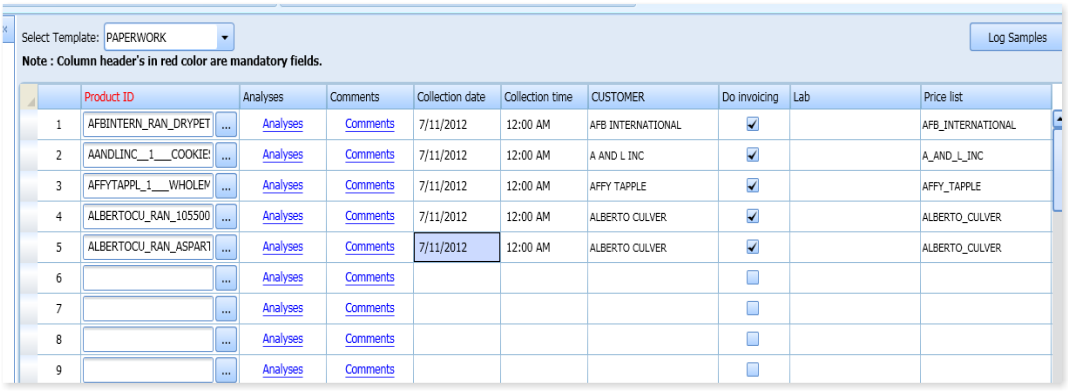
1. Launch Sample Login – the last used template is automatically assigned (you can change the template at the top if needed)



2. Add samples by clicking the ellipse buttons of the empty rows. This will launch the Location List screen and add samples.



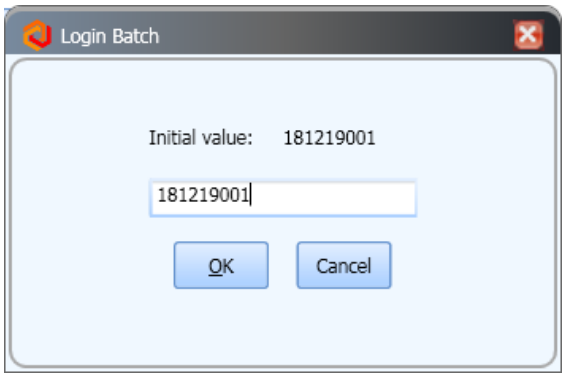
3. From the *Location List* screen, begin searching and double-clicking the samples you want to assign to the template in the background, then click **Close** when finished. The list closes and returns back to the samples list.



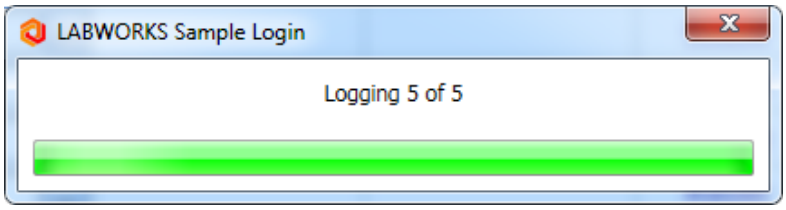
4. (Optional) Click to modify any specific field (defaults are all loaded in initially from the Location Codes).

5. (Optional) Click **Analyses** to modify any specific sample's test assignments – note you have many right-click features to copy a sample's tests and paste them into one or many other tests, for example.

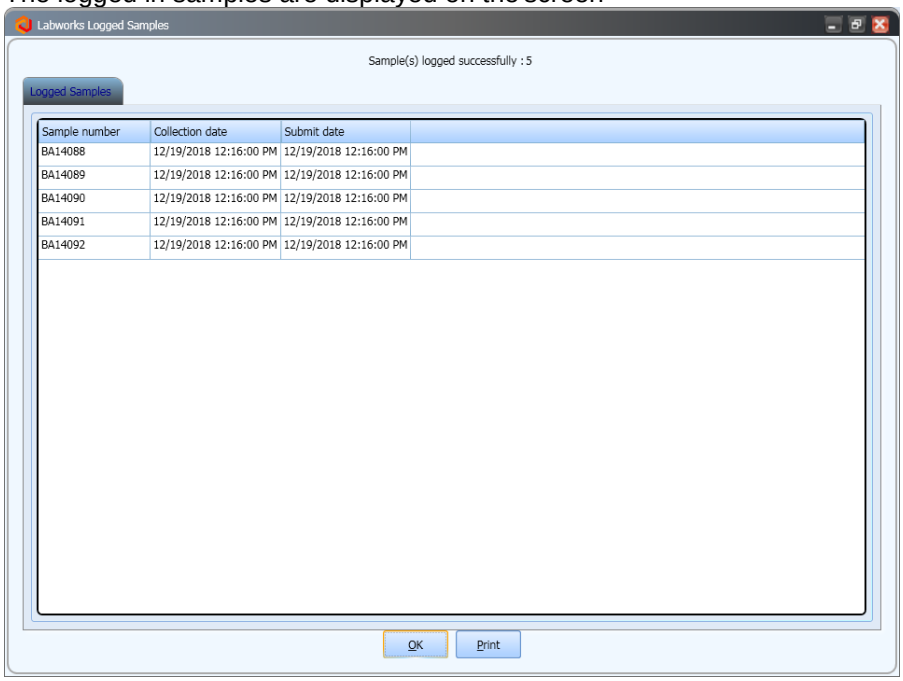
6. (Optional) Click **Comments** to adjust sample comments or any other related field
7. If satisfied with the sample list to be logged in, click **Log Samples** to fire off the process.
8. A Login Batch number is automatically assigned identifying these samples about to be logged into the system. You can elect to modify the value or not.



9. Click **OK** to continue. The sample login process begins.



10. The user programs are then launched one-by-one
11. The logged in samples are displayed on the screen



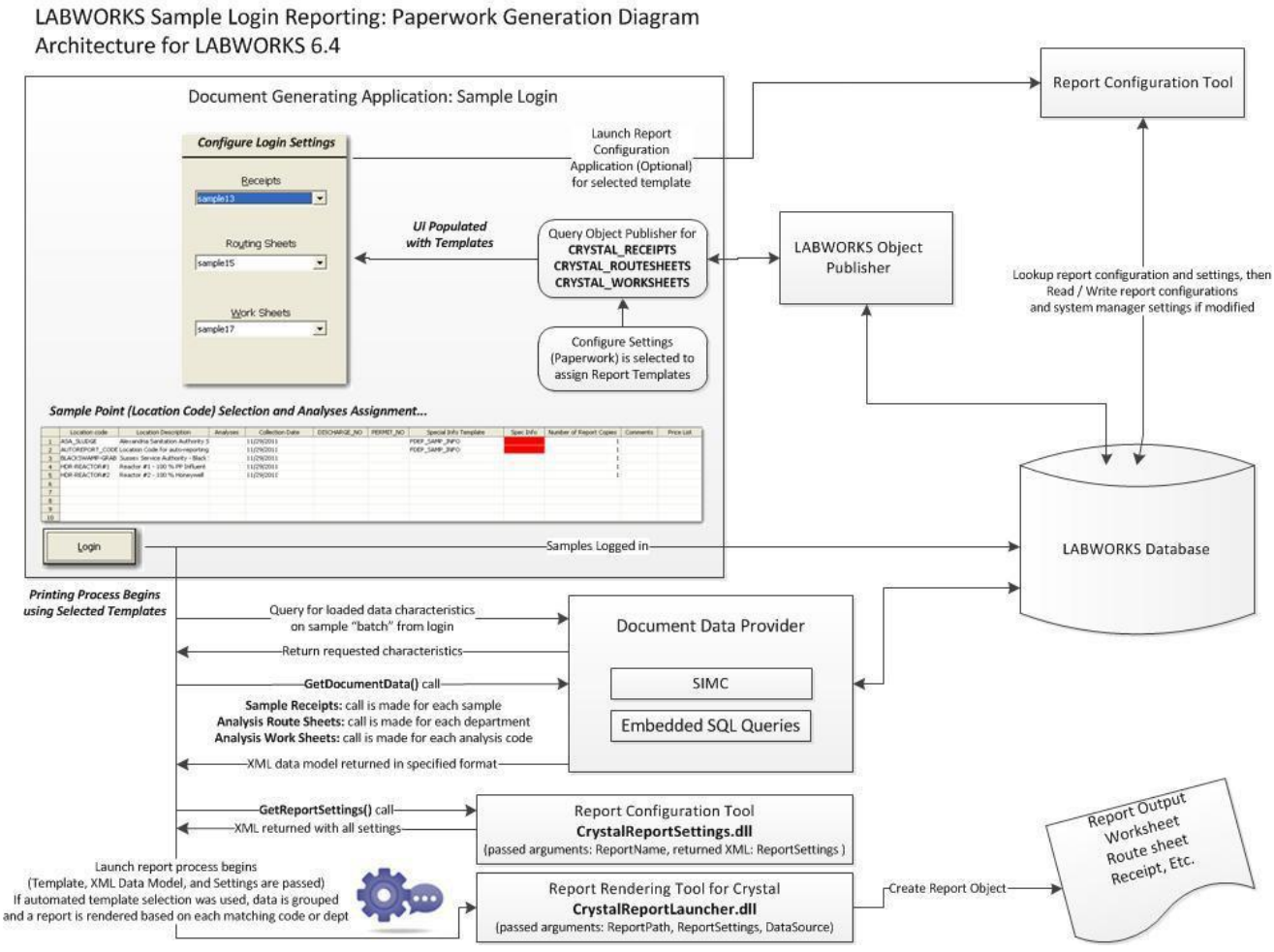
This screen doesn't show much by default. However, as mentioned previous, you can configure this screen to show as many fields as you need under the **Login Configuration** area.

12. Click the Print button to get a copy of the logged samples.
13. Click the OK button to complete the login process.
14. All paperwork will now be generated.

This completes the login process. Samples are now ready for results entry or any other additional processing required in the workflow.

Sample Login Complete Report Rendering Process

The following diagram places all of these discussed pieces together in an overview for System Administrators.



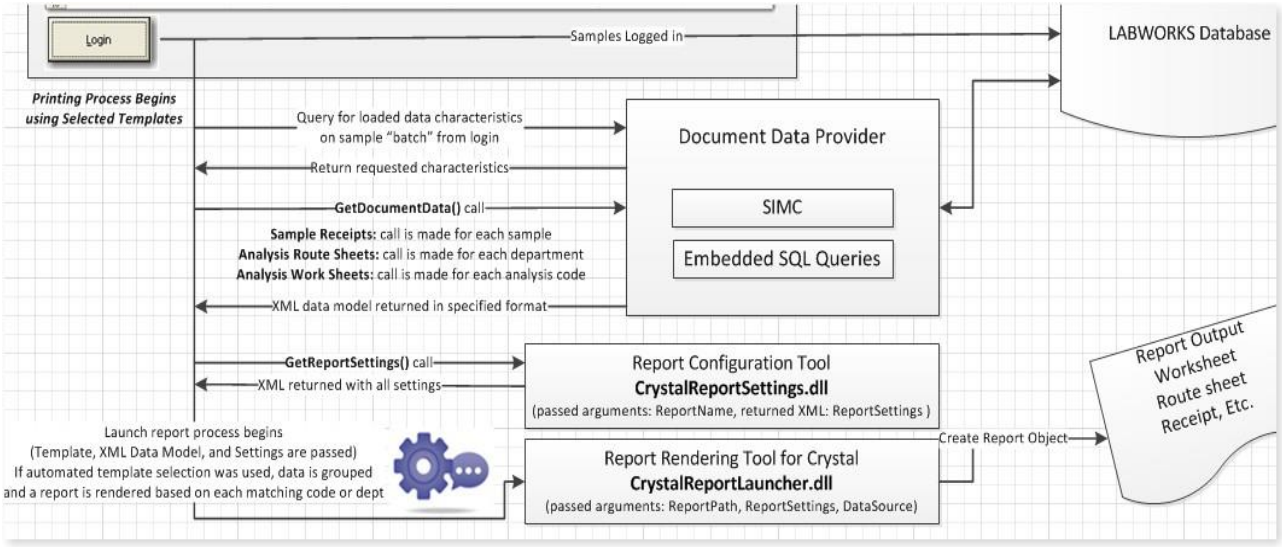
The bottom portion of the diagram details the rendering process which will be different depending on which type of report is used/assigned in the paperwork configuration screen or whether the automated template selection option is used.

Sample receipts, route sheets, and work sheets all operate differently in the application. However, the

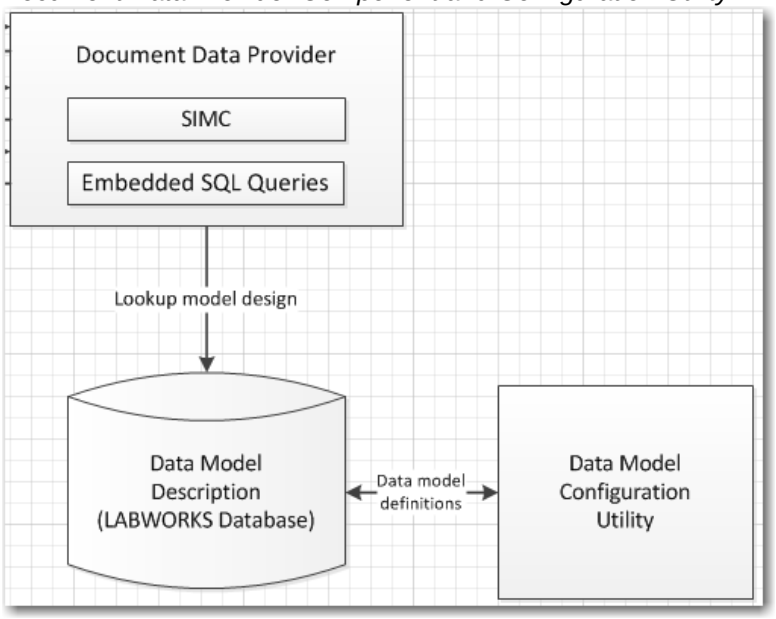
Data Models could be the same (if desired by the user and no non-standard sample data is needed for the reports). Each step of the rendering process will be discussed separately.

First of all, as mentioned previously the **LOGIN** process calls the following components after posting the samples to the database specifically for report rendering.

- **Document Data Provider Component** – called to receive data model XML
- **Report Configuration Tool** – called to receive the report settings XML (but also includes UI which is called under Paperwork Settings also, however no UI is displayed here)
- **Report Rendering Tool** – called to render report document (by default will display to the screen if no report settings are passed otherwise it does whatever the settings require)



Document Data Provider Component and Configuration Utility



Persisted Groups in LABWORKS

The sample is a key unit of information manipulated in LABWORKS. Samples have many properties and collections of objects (assigned analyses and results for example). Samples are usually processed in groups for activities such as login, analysis, reporting to customers, invoicing, and quality assurance. **Persisted groups** can be used to permanently memorialize the groupings of samples for various purposes during their lifetimes in LABWORKS.

The structures of persisted groups are very flexible, making them ideal for managing samples and related information for client projects, environmental regulatory reporting, research and development projects, and many other purposes.

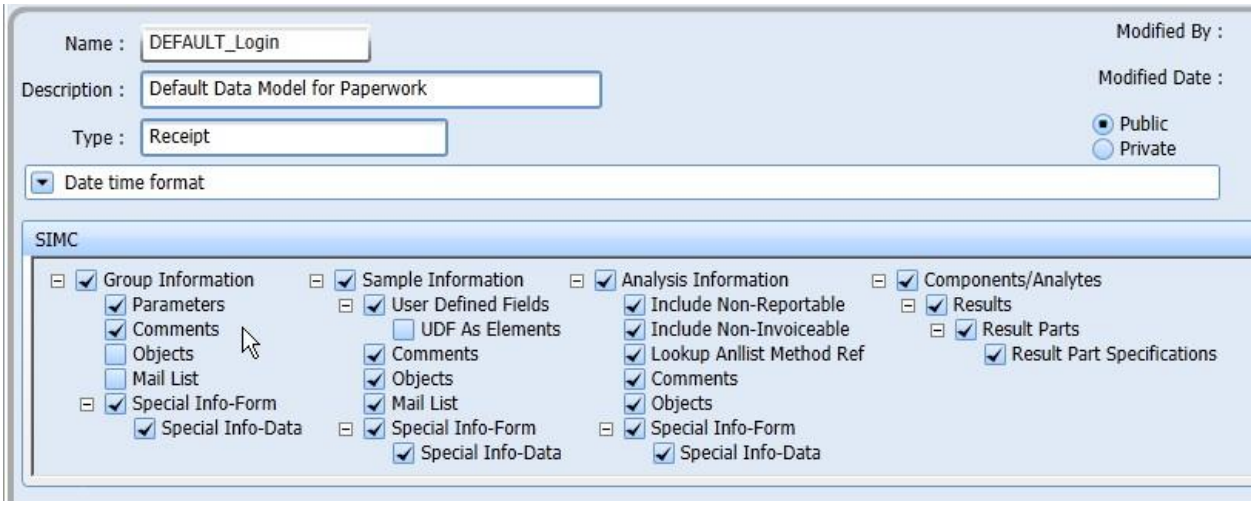
Persisted Groups Overview

Persisted groups are used to memorialize the grouping of samples for any purpose at any time desired. After grouping, the samples may be tracked and operated upon as a group at later times.

Persisted groups have properties and collections of objects that can be used to store information (for example comments, special information, and document objects) that that pertains more to the group (or the purpose of the grouping) than to the individual member samples of the group. Storing information at the group level reduces duplication, thereby improving database performance and storage efficiency.

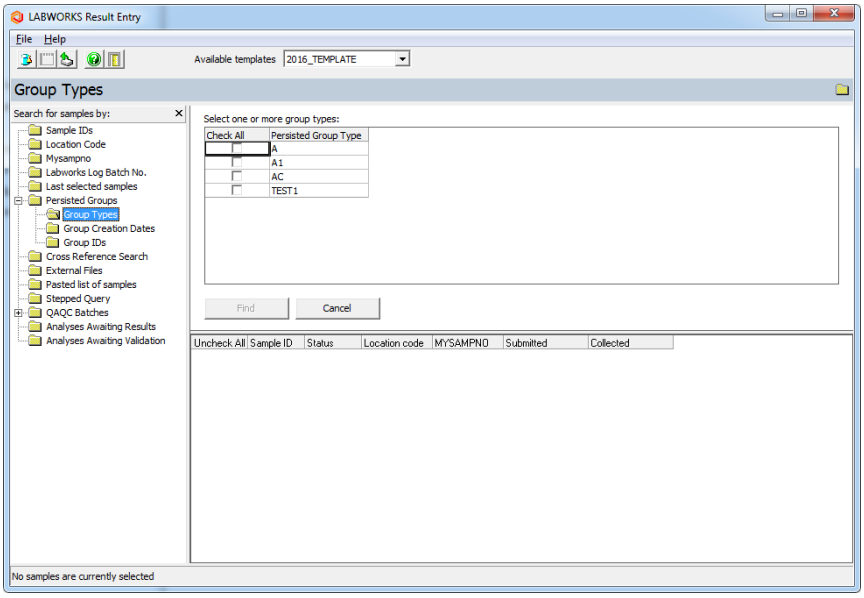
Persisted groups can be nested. The membership of a persisted group can be an unlimited number of samples or other persisted groups. The allowable level nesting within groups is unlimited. Sample members of any groups nested within a parent group are considered to also be members of the parent group.

LABWORKS custom reports that use XML data models can access the properties, information, and data objects of a persisted groups as easily as properties, information, and data objects of any sample (see example below).



Selection of persisted group items for inclusion in reporting:

Selection of samples by persisted group is available in all LABWORKS applications that select samples using the tree view sample selection component (see example below):



Selecting samples by persisted group in result entry application:

Persisted Group Properties and Data Objects:

In addition to their member collections persisted groups have several important properties which are:

Group ID: This is the unique identifier for the group. LABWORKS applications that create persisted groups (such as sample login) provide their own mechanisms for generating unique Group IDs. When manually creating groups, a default unique Group ID is generated by concatenating the following:

Group type + "_" + value from counter PGMANUMBER (padded to 6 digits)

Group Type: This property specifies the kind of group for possible searching later. Named persisted group types can be pre-configured for routine later use. Group type configuration is discussed later in this document.

Group Title: This is a human friendly descriptive title for the group. The of a new group defaults to the Group ID value if no title is specified.

Create Date: Date-time of group creation is automatically set.

Create User: Logged in user at the time of group creation is automatically set.

Modify Date: Date-time of most recent group update is automatically set.

Modify User: Logged in user at the time of most recent group update is automatically set.

Persisted groups may own any or all of the following types of data objects and collections:

Custom Group Parameters (collection): An unlimited number of custom parameters can be specified for any persisted group. Custom parameters consist of three parts: A name for the parameter, a type (text / date-time / number / Boolean / named pick list), and a value. Group parameters allow any group to have an infinite number of properties that may be the same or different from those of other groups.

Comments (collection): An unlimited number of comments can be attached to any persisted group.

Special Information Forms (collection): An unlimited number of special information forms can be attached to any persisted group.

Document Objects (collection): An unlimited number of document objects can be attached to any persisted groups.

Mail List Addresses (collection): An unlimited number of mail list addresses can be attached to any persisted group. For many purposes it makes more sense to associate mail list addresses with a group rather than the individual samples within the group.

Named Persisted Group Types

Named persisted group types can be configured to be used later as templates for creating persisted groups. Named group type configurations are stored as XML documents in the Datastore. The stored configuration of a named group can specify the following for groups that will be created from the type:

Default set of special information forms to associate at group creation time

Default set of prompts for document objects to attach at group creation time or later time.

Default set of group parameters for which to assign values at group creation or later time.

Additionally, the group configuration specifies the following:

Whether or not to allow non-default special information forms to be added to groups at group creation or later time.

Whether or not to allow the attachment of non-default (unprompted) document objects to persisted groups at group creation or later time.

Whether or not to allow ad hoc (non-configured) parameters to be added to persisted groups at group creation or later time.

Specifying a named group type as a property of a sample login template, causes a group of the type to be automatically created whenever the template is used to login new samples. The login application user will be prompted to enter values for group parameter values and group special information, and to attach the document objects configured for the group type. The creation of the persisted group becomes an integral part of the sample login process.

If a sample login template does not specify a persisted group type, groups will not be created when the template is used to log in new samples.

An unlimited number of named group types, for just about any sample grouping purpose, can be configured. For example, different named group types can be configured for logging in production control samples and environmental regulatory samples. Group types can be configured for specific clients of the laboratory to track information of interest to each client.

Preparations for Configuring a New Group Type

Custom parameters and special information forms are the main building blocks of persisted group types. Both custom group parameters and any fields on special information forms can have named pick lists assigned to provide selections of values.

Before creating a named group type, a list should be prepared of the items of information that will be

tracked by groups created of the type. Then decisions should be made regarding which items (if any) should be grouped onto special information forms and which should be tracked as individual group parameters.

LABWORKS lists should be created (if they do not previously exist) for all group parameters and special information fields whose type will be specified as named pick lists.

All special information forms that will be assigned to the group type must be designed if they do not already exist.

Using Special Information Forms vs. Custom Group Parameters

The choice of whether to use special information form or custom group parameters for tracking group information is based upon several consideration. Either will work in most cases and a special information forms can be thought of a set of custom parameters. When choosing which to use, the following guidelines should be followed:

If only a few items will be tracked (five or less), custom parameters may be more convenient.

If the information item being tracked is relevant to only one group type, custom parameters are more appropriate.

If the items to be tracked are related and fall into logical groupings, putting them on special information forms is appropriate.

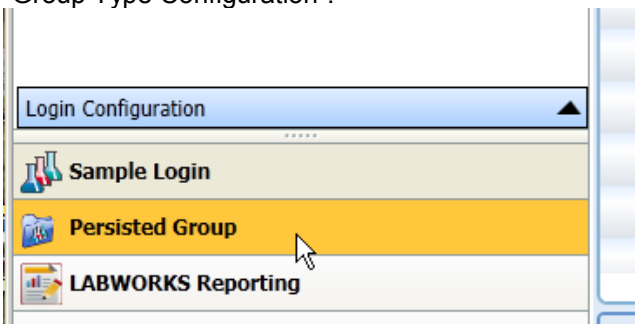
If sets of items to be tracked are relevant to multiple named group types, they should be placed on special information forms that can be reused in any number of group types.

Unlike samples and assigned analyses, persisted groups can have multiple special information forms assigned to them. Therefore, data items that will be entered at different time points of a persisted group's life time should be placed on separate special information forms.

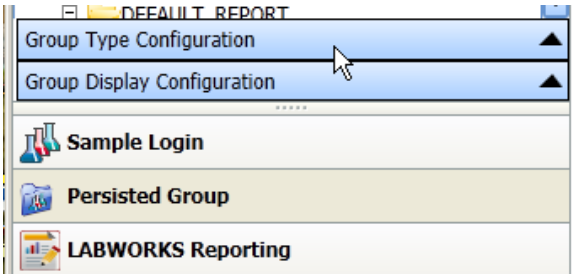
For example, items to be entered when a group of samples is checked into the laboratory can be on one special information form while items to be entered later, when completed results for the group are assessed prior to publication, can be on a different form. Both forms can be assigned as defaults for persisted groups to be created from the named group type.

Configuring Persisted Group Types

The tool for configuring persisted group types can be accessed from within the sample login application. Open group type configuration by first clicking "Persisted Groups" in the left pane and then clicking on "Group Type Configuration".

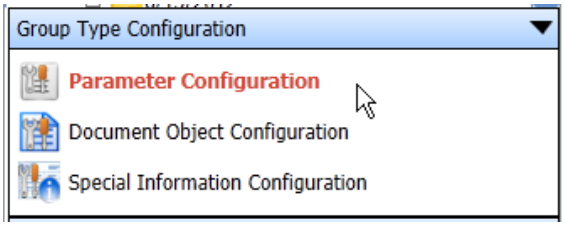


Clicking on Persisted Group



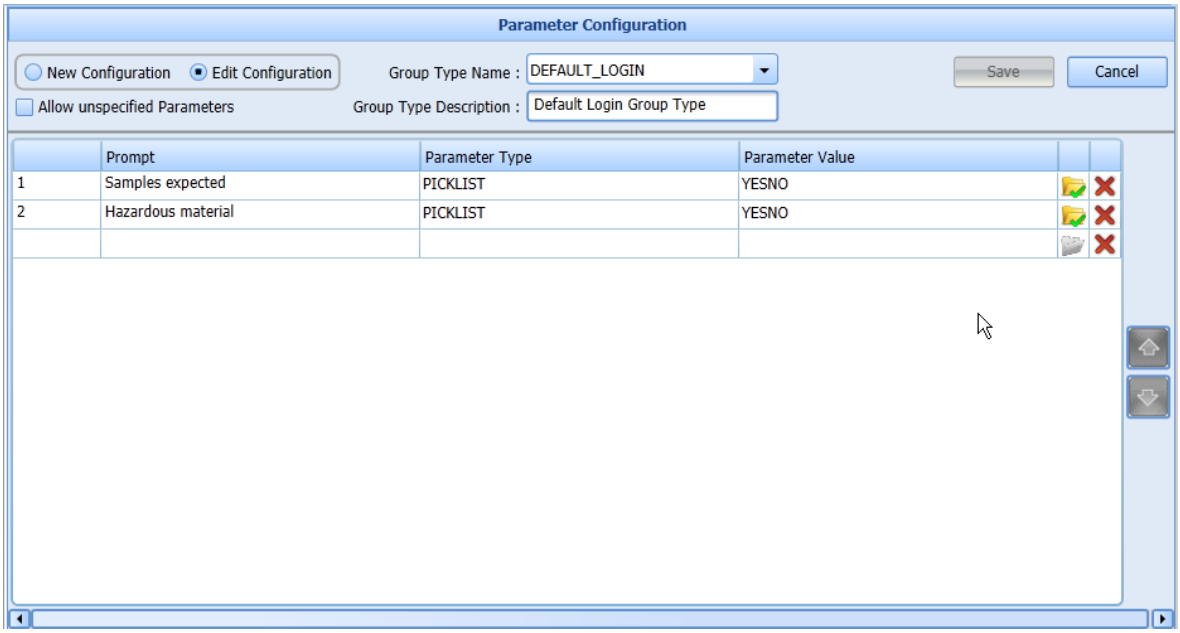
Expanding Group Type Configuration menu

The expanded Group Type Configuration menu contains choices for configuring group parameters, document object prompts, and special information forms for the named group type being configured.

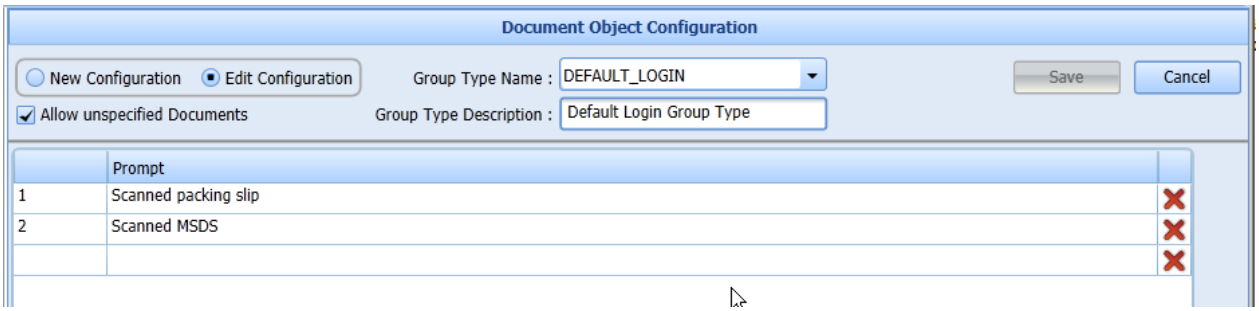


Expanded group type menu choices

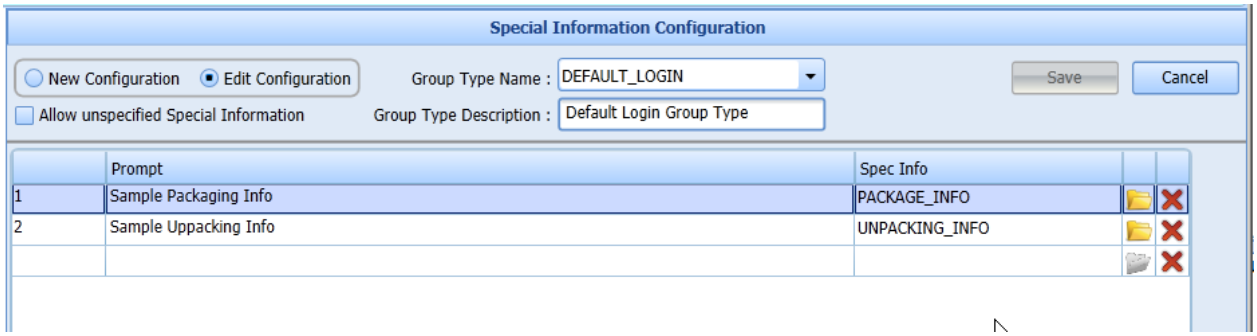
Selecting any of these choices opens a configuration window for the in the right pane for the group item being configured.



Configuration window for editing the list of parameters assigned to the “DEFAULT_LOGIN” group type

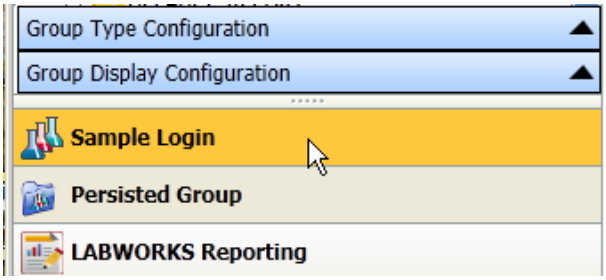


Configuration window for specifying the document object run time prompts for the “DEFAULT_LOGIN” group type

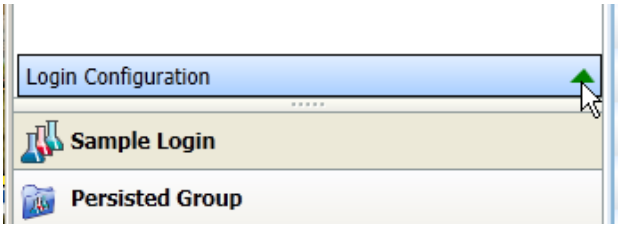


Configuration window for assigning special information forms to the “DEFAULT_LOGIN” group type

After configuration, a named group type can be associated with any number of sample login templates using the Sample Login configuration tool. That tool is opened by clicking the “Sample Login” menu choice in the right pane of the sample login application then expanding the sample login menu (see below):

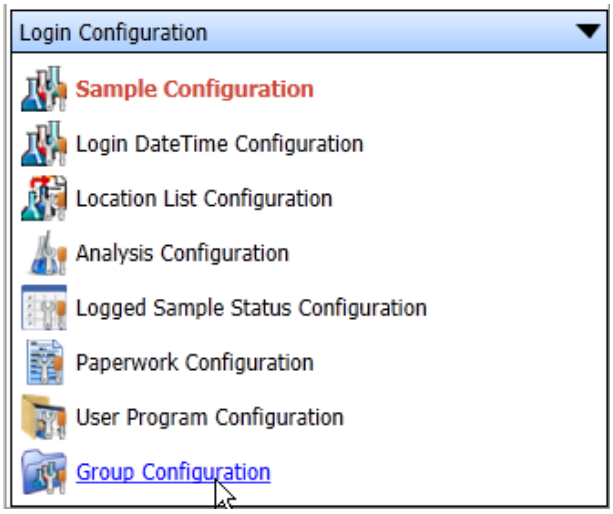


Clicking on the Sample Login menu choice in the left pane

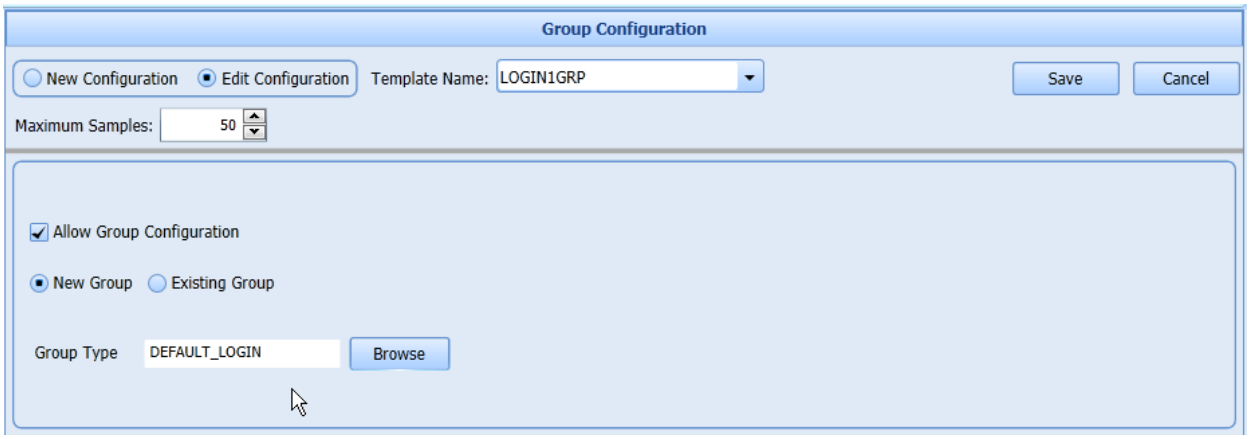


Expanding the Sample Login menu

Selecting “Group Configuration” from the expanded Login Configuration menu will open the persisted group assignment window for the login template being configured (see below):



Selecting Group Configuration from the expanded Login Configuration menu



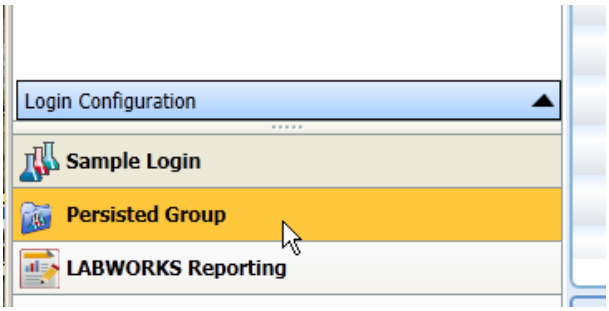
Login template configuration persisted group type assignment window

As previously stated, if a persisted group type is assigned to a login template, creation of the group becomes an integral part of every sample login process using the template. However, if a sample login template does not specify a persisted group type, groups will not be created when that template is used to log in new samples.

Managing Persisted Groups

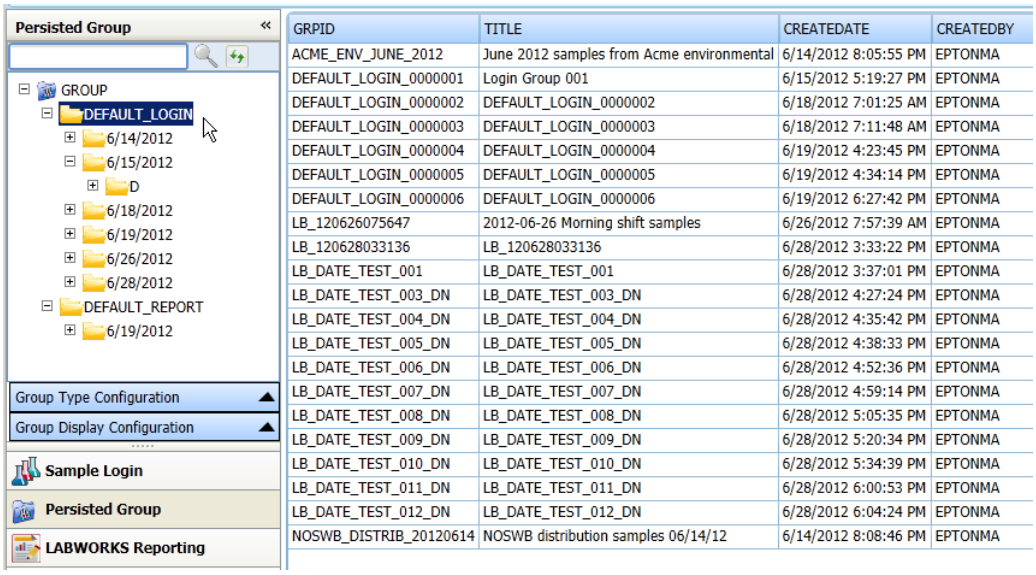
After they are created, persisted group it can be located and managed using the Persisted Group Management Application (PGMA). **A user must have the “Persisted Group Management” user privilege assigned to use the PGMA.**

An authorized user can access the PGMA by clicking “Persisted Groups” in the left pane of the sample login application.



Clicking on Persisted Group in the left pane of sample login

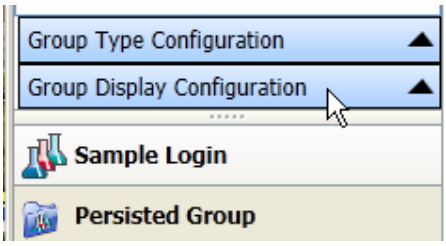
The persisted group selector will open. The group selector consists of a tree view control in the left pane and a grid view in the right pane listing the groups belonging to the currently selected node of the tree view (see example below):



GRPID	TITLE	CREATEDATE	CREATEDBY
ACME_ENV_JUNE_2012	June 2012 samples from Acme environmental	6/14/2012 8:05:55 PM	EPTONMA
DEFAULT_LOGIN_0000001	Login Group 001	6/15/2012 5:19:27 PM	EPTONMA
DEFAULT_LOGIN_0000002	DEFAULT_LOGIN_0000002	6/18/2012 7:01:25 AM	EPTONMA
DEFAULT_LOGIN_0000003	DEFAULT_LOGIN_0000003	6/18/2012 7:11:48 AM	EPTONMA
DEFAULT_LOGIN_0000004	DEFAULT_LOGIN_0000004	6/19/2012 4:23:45 PM	EPTONMA
DEFAULT_LOGIN_0000005	DEFAULT_LOGIN_0000005	6/19/2012 4:34:14 PM	EPTONMA
DEFAULT_LOGIN_0000006	DEFAULT_LOGIN_0000006	6/19/2012 6:27:42 PM	EPTONMA
LB_120626075647	2012-06-26 Morning shift samples	6/26/2012 7:57:39 AM	EPTONMA
LB_120628033136	LB_120628033136	6/28/2012 3:33:22 PM	EPTONMA
LB_DATE_TEST_001	LB_DATE_TEST_001	6/28/2012 3:37:01 PM	EPTONMA
LB_DATE_TEST_003_DN	LB_DATE_TEST_003_DN	6/28/2012 4:27:24 PM	EPTONMA
LB_DATE_TEST_004_DN	LB_DATE_TEST_004_DN	6/28/2012 4:35:42 PM	EPTONMA
LB_DATE_TEST_005_DN	LB_DATE_TEST_005_DN	6/28/2012 4:38:33 PM	EPTONMA
LB_DATE_TEST_006_DN	LB_DATE_TEST_006_DN	6/28/2012 4:52:36 PM	EPTONMA
LB_DATE_TEST_007_DN	LB_DATE_TEST_007_DN	6/28/2012 4:59:14 PM	EPTONMA
LB_DATE_TEST_008_DN	LB_DATE_TEST_008_DN	6/28/2012 5:05:35 PM	EPTONMA
LB_DATE_TEST_009_DN	LB_DATE_TEST_009_DN	6/28/2012 5:20:34 PM	EPTONMA
LB_DATE_TEST_010_DN	LB_DATE_TEST_010_DN	6/28/2012 5:34:39 PM	EPTONMA
LB_DATE_TEST_011_DN	LB_DATE_TEST_011_DN	6/28/2012 6:00:53 PM	EPTONMA
LB_DATE_TEST_012_DN	LB_DATE_TEST_012_DN	6/28/2012 6:04:24 PM	EPTONMA
NOSWB_DISTRIB_20120614	NOSWB distribution samples 06/14/12	6/14/2012 8:08:46 PM	EPTONMA

View of the persisted group selector

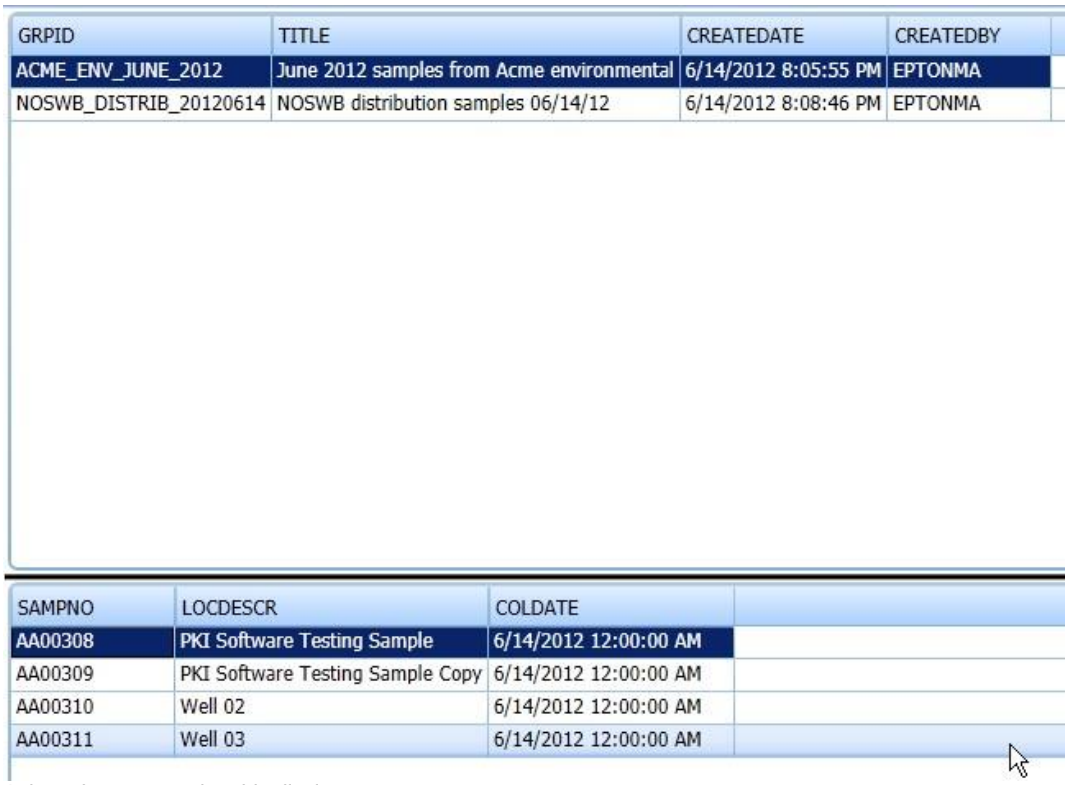
Both the tree view and grid views can be configured using the “Group Display Configuration” menu (see below):



Selecting the Group Display Configuration menu

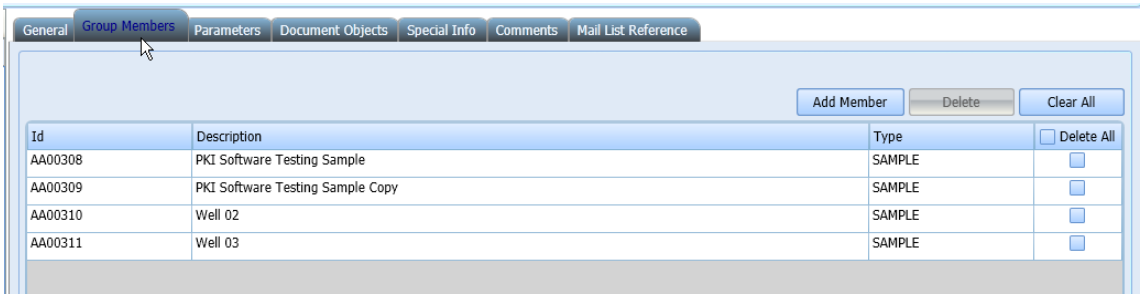
Clicking any row in the grid selects a group and displays its members in the lower right pane (see

example below).



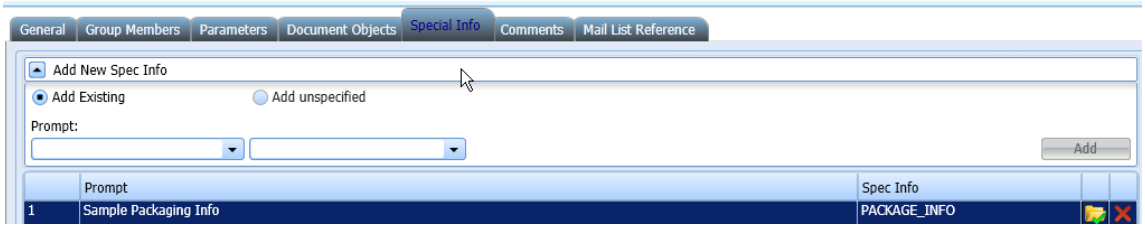
Selected group membership display

Double clicking a row in the grid opens the selected group for modification in the right pane (see example below):



The persisted group modification window showing membership tab

A set of tabs provides access to dialogs for modifying any properties or objects belonging to the group, including its membership list.



Group modification special information edit tab selected:

3. LABWORKS Crystal 11 Report Designer

About the Program and Getting Started

About the LABWORKS Report Designer

The LABWORKS Report Designer program is designed to include all the advanced report writing functionality of the award winning Crystal Reports™ Standard Edition software with a seamless integration of your LABWORKS database and an array of additional features for special exporting of reports, automatic emailing, website publishing, etc. This LABWORKS module is packed with features utilizing the latest reporting technology to provide users with a dynamic report writer with unparalleled laboratory report design potential. LABWORKS is sure Users will find this report designer program to be their most valuable asset for creating custom reports that meet their specifications.

As an advanced report designer with these capabilities plus integrated LABWORKS functionality, the LABWORKS Report Designer gives the User the ability to create unlimited custom reports in Crystal format. A report file built in Crystal format uses the Crystal Reports™ engine when printed that gives the User several great options such as converting their printed report to Acrobat PDF, Word, Excel, Lotus, etc. or emailing the report right from the report's print preview screen. The purpose of this chapter is to assist you with unlocking the potentials of this extraordinary report design program.

The LABWORKS installation automatically installs an example database file in the USER folder for use when creating reports. This allows the immediate design and creation of reports using the LABWORKS Report Designer without having to enter laboratory data in LABWORKS.

LABWORKS provides three methods of Crystal report creation:

- Using exported data
- Using direct database connect

Report Creation Using Exported Data

The most common and stable method of Crystal Report ™ creation is by the use of LABWORKS exported data. This method involves a few added steps in the report creation process, but it ensures that reports function properly regardless of the database type.

An export file (with a filename extension of CEF) is used by LABWORKS to determine which database fields to extract for each report query. When a report is printed, LABWORKS searches for a CEF file matching the report name within the Crystal folder in the LABWORKS data folder. The listing of fields in the CEF file is used to build a new Microsoft Access database (MDB) file in the LABWORKS user directory specifically for the report. Each time the report is printed, LABWORKS rebuilds the MDB file of exported data. For the program to function, the report file (RPT) must be kept in the same Crystal folder with the matching CEF file.

The building of the export file is the extra step in the report creation process for using exported data. See

Create Reports Using Exported Data section for details.

Report Creating Using Direct Database Connection

The direct connection to a LABWORKS database file or server is another method of creating Crystal reports. This method allows you to skip the CEF file building process but requires the selection of tables and fields from within your report. A major benefit of this method is the ability to include non-exported data in the report, e.g., user/security or audit trail information. The report creation process is also faster, since the export database is not created each time you print.

This method allows printing large monthly or annual reports in nearly a fraction of the time and without the constraints of Microsoft Access. However, if you change your database or connection type, you must change each of your reports as well. In addition, these reports do not take into account version changes of LABWORKS. If your database fields change in updated builds of LABWORKS, your reports may be negatively effected.

Creating a Model before you Start

Perhaps one of the most important steps in creating a Crystal report is the creation of a model for building it. If you have a report that you want to duplicate, use it to design and create the report with modifications as needed. You may want to start with a blank sheet of paper and lay out the report with header, body, date, charts, data sections, footer, page numbers, font size for the text, e.g., large font for the title, placement of text to be centered or left justified, paper size, etc. The more creation you do up front so that you have a vision of what the end report is to look like, the easier it is to create the report using the Crystal designer features.

Two basic elements of the report are content and layout. Decide what you want to include and how you want it to appear on the finished report. Consider drawing boxes for text areas, adding desired shading, and making a list of what information is required to be on the report. Lay out that information on a blank sheet of paper.

Once you create the initial report, you can easily open and modify it until you are satisfied with the final layout. You may want to save several variations of the same report until you decide which one works best. You may also open and modify the sample reports installed with LABWORKS and save them with a different name. It is usually much easier to modify an existing report to create a new one than starting completely from scratch.

As you become more adept at using the LABWORKS Report Designer program, the easier it is to create and modify new reports. The integration of the Crystal Report Writer function with LABWORKS data facilitates the report generation process and its possibilities. By becoming skilled at using the designer program, you can tap into its potential report capabilities.

Data Management

It is important to know the type of database you are reporting from and the location of the data used by the reports. Some things to consider when setting up the report are:

- Type of database you are using for the reports.
- Location of the data on your local or network computer system.
- Available database tables.
- Data type for the data fields, e.g., numeric data, date or time fields.
- Data fields to be included and their location.

- Data field values that require calculations.
- Text objects to be included.
- Data that needs to be flagged and how you want it flagged.
- Data that needs to be organized in groups, e.g., by date, customer.
- How to sort the data.
- Data to be summarized, e.g., totals, averages, counts, maximum, minimum.

Printing Characteristics

Every report area may have different printing characteristics that affect when and how often the different report objects are printed. Areas on the report print in the order they appear on the design editor screen. If you set up more than one section in an area, the sections print in the order they appear. For example, if you have three Report Header sections, all three of those sections print, in order, before the section(s) in the Page Header area begin to print.

The way objects print determines how you design your report. This may help you decide where to place charts, Cross-Tabs, and formulas to get specific results.

Available Assistance

This chapter provides the basics of using LABWORKS Report Designer functions for creating reports. If you need additional assistance, it is recommended that you contact PerkinElmer Instruments. PerkinElmer Instruments also provides LABWORKS Report Designer Training on how to use the program.

Create/Edit Reports Using Exported Data

Create or Edit Reports Using Exported Data

Using LABWORKS exported data is the most common and stable method of creating Crystal reports. Creating reports using this method ensures that reports function properly regardless of the database type. This creation process uses an export file (CEF) to determine which database fields to extract for each report query.

When a report is printed, LABWORKS searches for a CEF file matching the report name within the Crystal folder in the LABWORKS data folder. LABWORKS uses the listing of fields in the CEF file to build a new Microsoft Access database (MDB) file in the LABWORKS user directory for the report. Each time the report is printed, LABWORKS rebuilds the MDB file of exported data. For the program to function, the report file (RPT) must be kept in the same Crystal folder with the matching CEF file.

Topics covered in this section include:

- Prepare a Report Model
- Access the LABWORKS Report Designer Program
- Select Samples
- Enter Report Name
- Create CEF File
- Use LABWORKS Report Designer to Set Up the Report
- Edit an Existing Report

- Delete a Report

Prepare a Report Model

Obtain or prepare a model for designing your report. Having a model facilitates the design process in that you know how the final report is to look and function. You also may create a new report by modifying an existing one. If you are modifying an existing report, consider printing it and then mark it up with the changes you want to make.

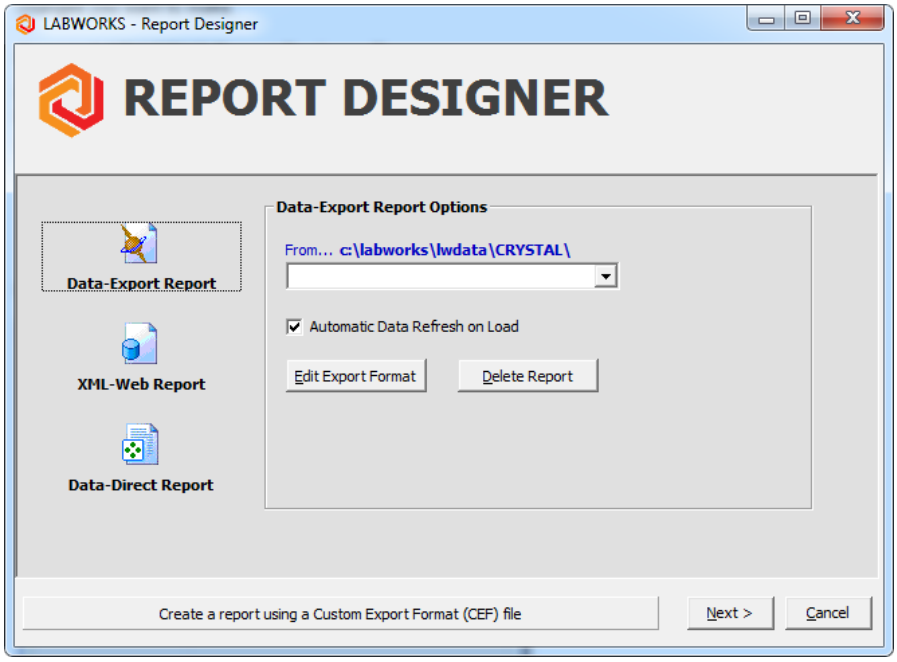
Access LABWORKS Report Designer Program

Users may access the LABWORKS Report Designer program through the Reports menu items on the LABWORKS Enterprise desktop or from a toolbar icon.

To access the LABWORKS Report Designer:

1. At the LABWORKS desktop double click on the **Database icon.**
2. Open the **Applications** folder, and then open the **Reports** folder.
3. Double click on **Crystal Report Setup** program.

Notice that the Crystal Report Format Setup screen appears for selecting or entering a format name.



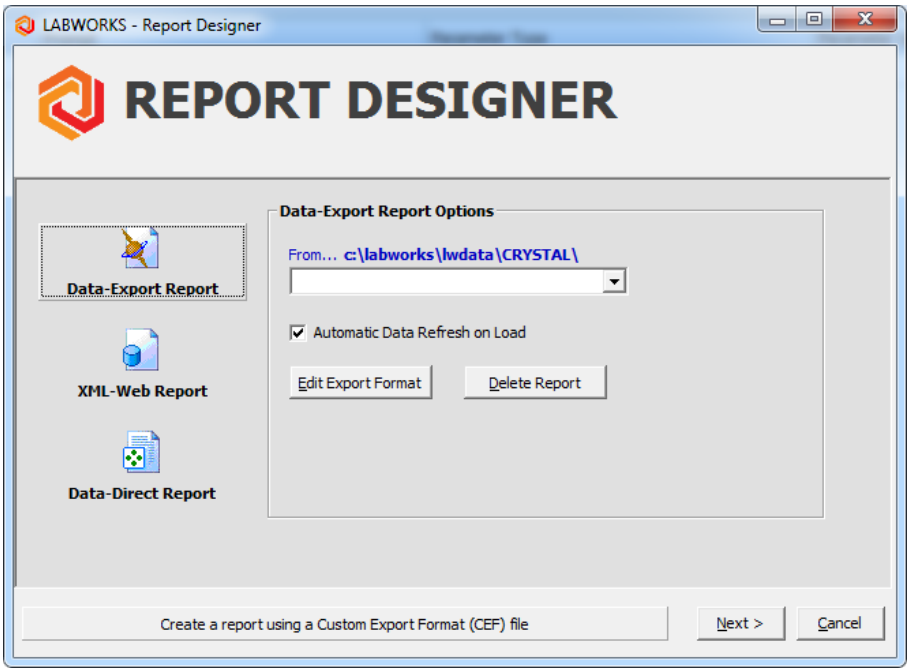
Enter Report Name

The report name is limited to 24 characters with no spaces or file name extensions. You may use the underline key to connect words, e.g., Summary_Report. If you want all your Exported Data Crystal Reports to be easily recognized, consider beginning each report title with CR_.

To enter a new report name:

1. Click on the Data Export Report icon that is displayed on the Report Designer screen.

Notice that if you hold your cursor over an item on the screen, instruction text associated with that item appears.



A brief description of the options on the Crystal Report Format Setup window is as follows.

From – allows entering a report name for a new report or selecting a report filename from the pull-down list.

Edit Export Format – allows you to edit the selected export format file.

Delete Report – allows you to select and delete report formats..

2. To create a new report, enter a new report name in the **From** field.

OR

To create a new report from an existing report, click on the down arrow at the end of **From** field and select the report you want to modify to create a new one (see Edit and Existing Report section).

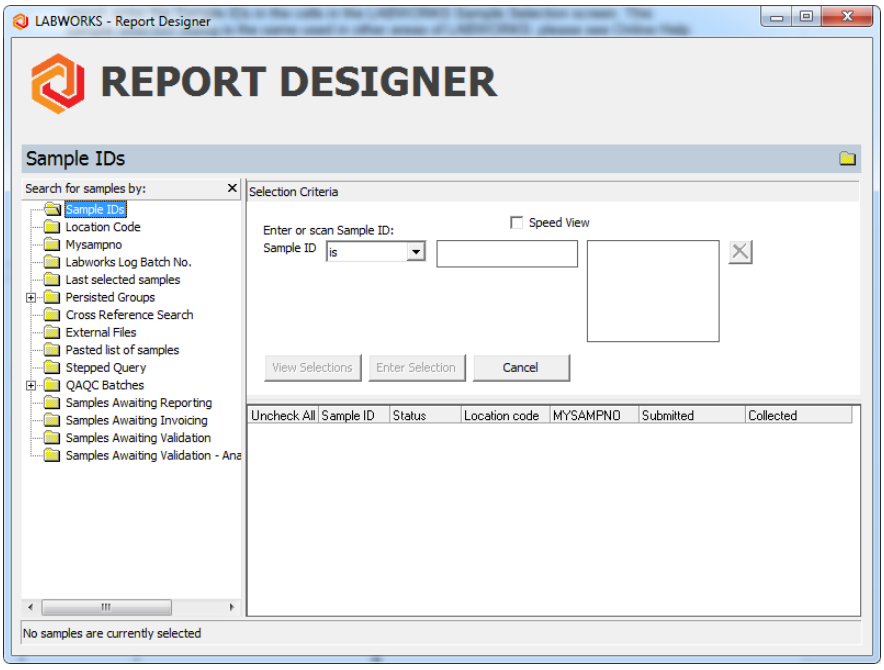
3. Click **OK** to go to the next step.

Select Samples

If you have not entered data in LABWORKS, LABWORKS installs a database with test data that allows you to set up your reports before entering data.

To select sample data for the report:

1. If you have established data and know the ID for the specific samples to be included in the report, enter the Sample IDs in the cells in the LABWORKS Sample Selection screen. This sample selection dialog is the same used in other areas of LABWORKS; please see Online Help if you need information about how the sample selection screen operates.



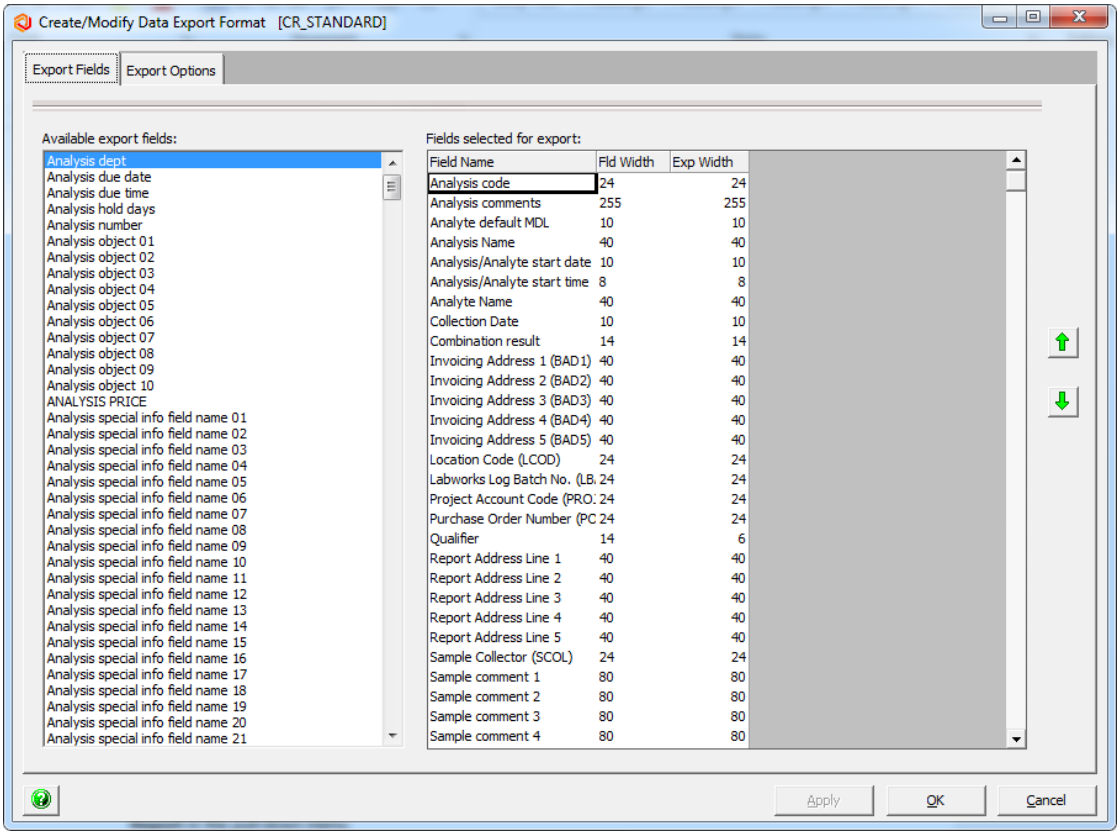
Create CEF File

Data Export Reports require creation of an export file (CEF) that determines which database fields to extract for each report query. If you do not select to create a Data Direct Report, the CEF export file editor appears when you click **OK** on the Crystal Report Format Setup window. This section describes how to create the CEF file.

To create the CEF File:

1. After you select to create a Data Export Report and click **OK,** the Create/Modify CEF File screen appears with a list of fields from which to select the fields to include in the report.

The Create/Modify CEF File screen has two tabs:
CEF Fields tab – provides a selection list of data fields to export to a report.
CEF Options tab – provides several options for determining how data from

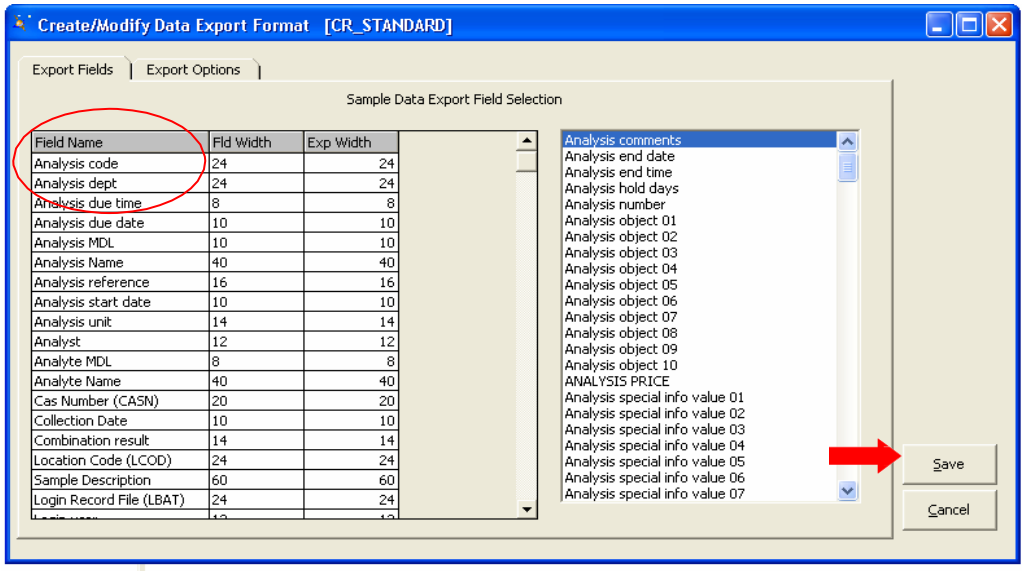


LABWORKS is exported to Access.

NOTE: It is not necessary to set the options on the CEF Options tab.

2. Double click the fields you want exported in the list to the right.

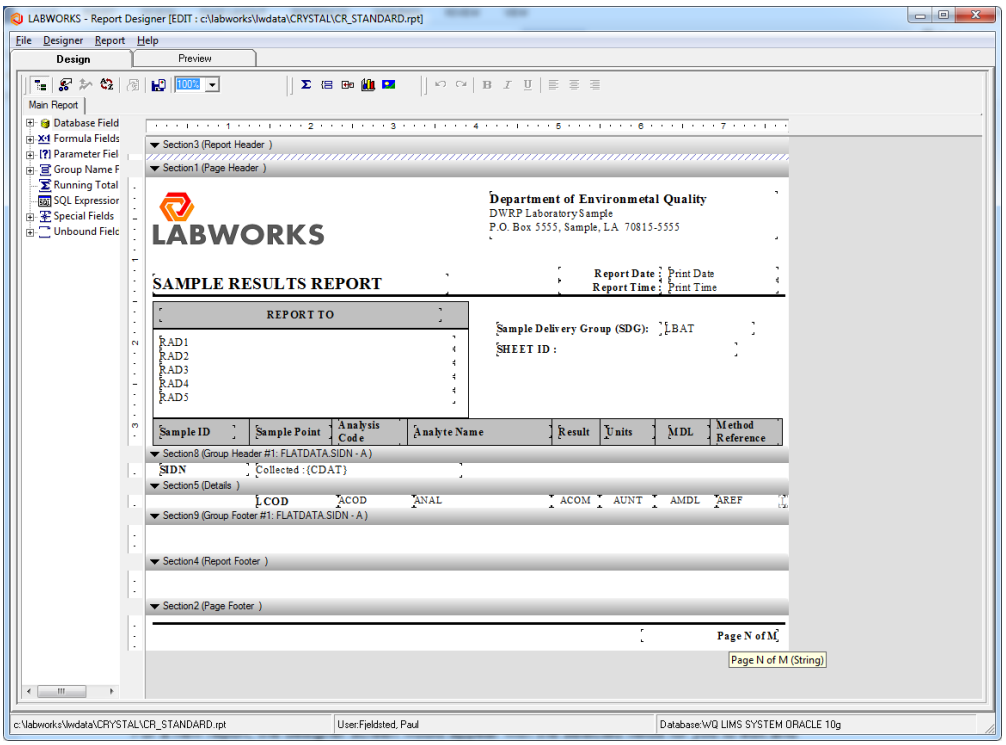
3. Notice that the selected fields appear in the **Field Name** column with their respective field width.
4. To remove a selected field from the **Field Name** column, double click on the field name; notice that it reappears in the list of available field names for selection on the right.



NOTE: Select Combination Result if you want all types of results to be exported. If you select Numeric Results, only numbers are exported. If you select Text Result, only text data (no numbers) are exported. For details on using the CEF editor, see Create/Modify Export Format section of the Maintenance Functions chapter of the LABWORKS User Manual.

5. After selecting fields for your report, click **Save**.

6. Notice that the LABWORKS Report Designer appears with either the report selected or a blank screen if you are designing a new report from scratch.

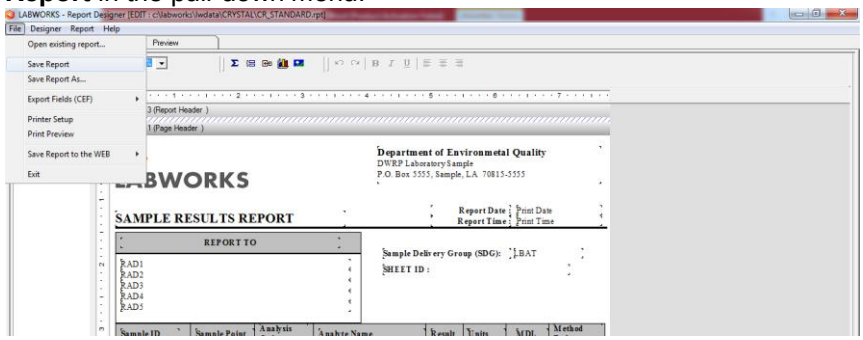


For a new report, the designer screen would appear with the selected fields for you to edit and

customize to your requirements.

Save and Add Database Fields to Report

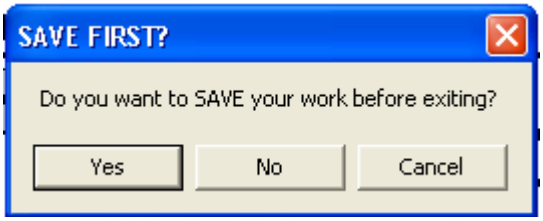
1. Before you begin setting up the report, save the report by clicking **File** and selecting **Save Report** in the pull-down menu.



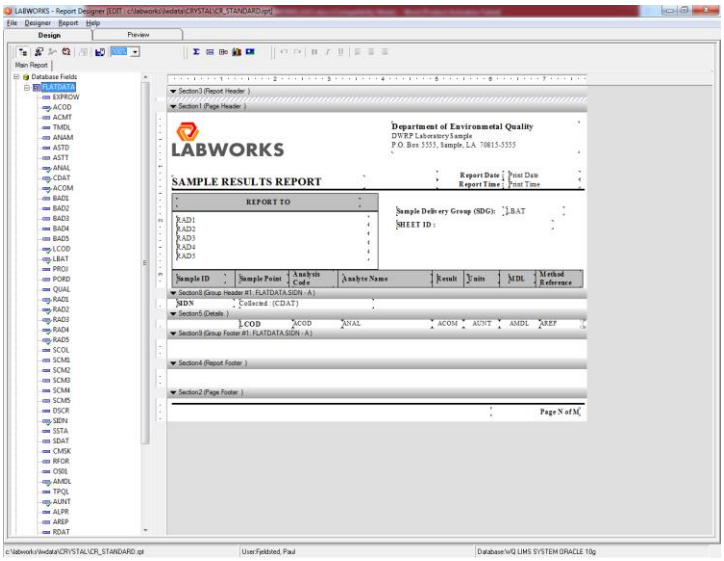
2. Save the report file with the **same name** you entered on the Crystal Report Format Setup popup window.

3. Save it in the Crystal folder if you want to be able to select it from the Crystal Report Format Setup pull-down list.

4. If you attempt to close the report before saving it, the program provides a prompt that asks if you want to save before exiting.



5. Click **Yes** on the Save First pop up window to save your report; click **No** to exit without saving; or click **Cancel** to return to the design screen and continue creating or editing the format. Once saved, the report will appear in the Crystal Report Format Setup pull- down list for selection.
6. Double click **Database Fields** in the upper left portion of the screen to display the list of all your exported fields from the previous step.



The codes for the exported fields appear in the list below the database name FLATDATA on the screen. When you double click Database Fields using direct connect, a list of all the LABWORKS database tables is displayed.

7. Click and drag fields to your report as needed and begin editing.

8. See Menu Items and Special Editing Features, for how to create the report using the designer.

CEF File Menu

The CEF File menu is available **only for export file reports** (NOT direct connect) and has the following options:

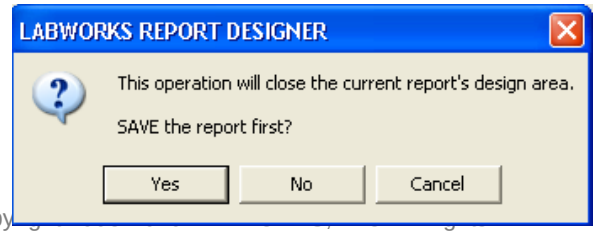
Add/Remove Database Fields – opens the CEF editor screen (see Create CEF File section) and allows you to add or remove database fields to the currently opened report. After adding or removing fields, the CEF file is saved and the export MDB database file is recreated. The report then refreshes itself to reflect the new fields in the top left Database Fields dropdown menu.

Verify/Refresh Labworks Data – refreshes the database if changes were made prior to reopening this report. Once this is done, the list of database fields under the Database Fields menu item reflects the new changes. You can refresh the database at any time to reflect new changes whether the report uses direct connect or exported data.

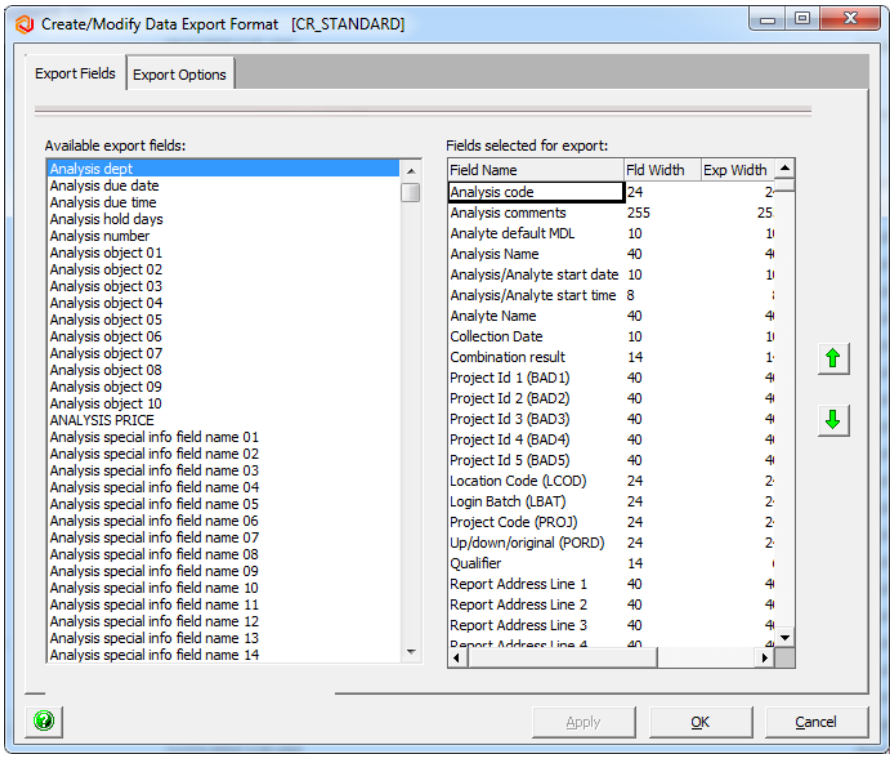
To display names for database field codes

1. Click on **File** on the LABWORKS Report Designer menu bar.

2. Click **Add/Remove Database Fields**.



3. Notice that you get a prompt to save your report first, click YES.
4. The Create/Modify CEF File screen appears with the list of field names you selected. The database field codes on the designer screen correspond to and are in the same order as the field names you selected on this screen.

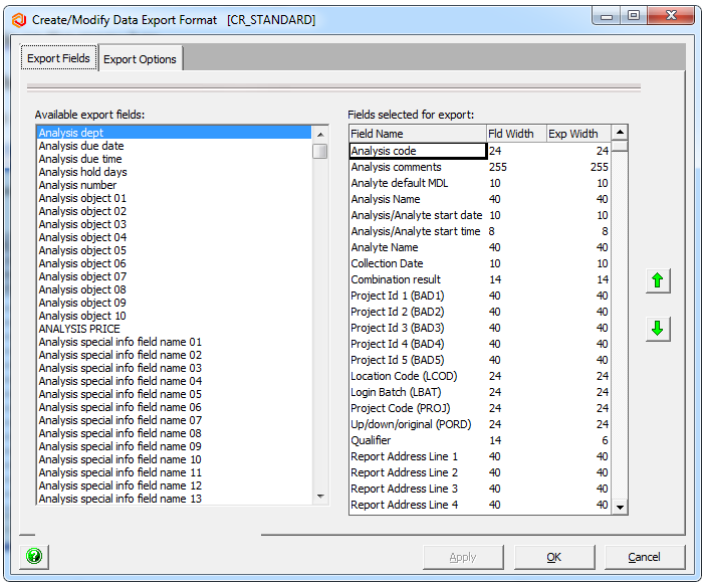


5. Keep this screen open in the background and refer to it as needed when selecting database fields for the report.

To add or remove database fields from the CEF file:

1. Click on **File** on the menu bar.
2. Click **Add/Remove Database Fields**.

The field names on this screen correspond to the database field codes listed on the designer screen.



3. Double click on field names you want to add to or double click on the **Fld Width** column to remove a field from the report.

4. Click **Save** to save the CEF file and recreate the export MDB database file.

5. Notice that the report refreshes itself and displays the new fields in the **Database Fields** dropdown menu on the designer screen.

Edit an Existing Exported Data Report

To edit an existing report:

1. To edit or create a new report from an existing report, click on the down arrow at the end of the **From** field.

2. Scroll through the dropdown list and select the report to be edited.

3. Click **OK** to go to the next step.

4. Before you begin editing, double-click **Database Fields** on the left to display the list.

5. Click and drag fields to your report as needed and begin editing.

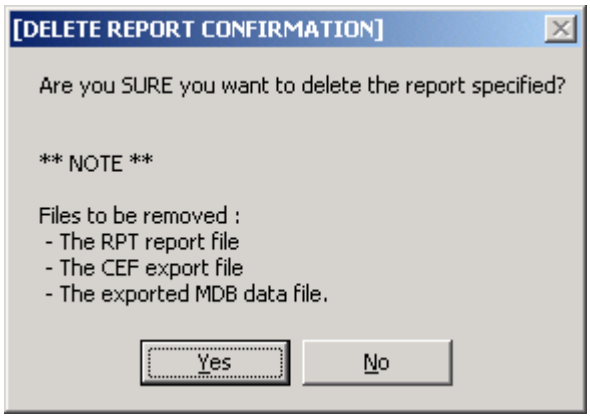
Delete an Exported Data Report

To delete a report:

1. To delete an existing report, enter the report name or click on the down arrow at the end of the **From** field.

2. Scroll through the dropdown list and select the report to delete.

3. Click **Delete**.



3. Click **Yes** on the Delete Report Confirmation screen to delete the selected report.

Create/Edit Reports Using Direct Connect

Create or Edit Reports Using Direct Connect

The direct connection to a LABWORKS database file or server is another method of creating Crystal reports. This method allows you to skip the CEF file building process but requires the selection of tables and fields from within your report. A major benefit of this method is the ability to include non-exported data in the report, e.g., user/security or audit trail information. The report creation process is also faster, since the export database is not created each time you print.

This method allows printing large monthly or annual reports in nearly a fraction of the time and without the constraints of Microsoft Access. However, if you change your database or connection type, you must change each of your reports as well. In addition, these reports do not take into account version changes of LABWORKS. If your database fields change in updated builds of LABWORKS, your reports may be negatively affected. Topics covered in this chapter include:

Prepare a Report Model

Access the LABWORKS Crystal Repot Designer Program

Select Samples

Enter Report Name

Use LABWORKS Crystal Designer to Set Up the Report

Edit an Existing Report

Delete a Report

Prepare a Report Model

As described in Create a Model Before You Start section, obtain or prepare a model for designing your report. Having a prepared model facilitates the design process in that you know how the final report is to look and how it is to function to accomplish its purpose. You may also modify an existing report to create a new one. Consider printing and marking up an existing report as a model to create a new report.

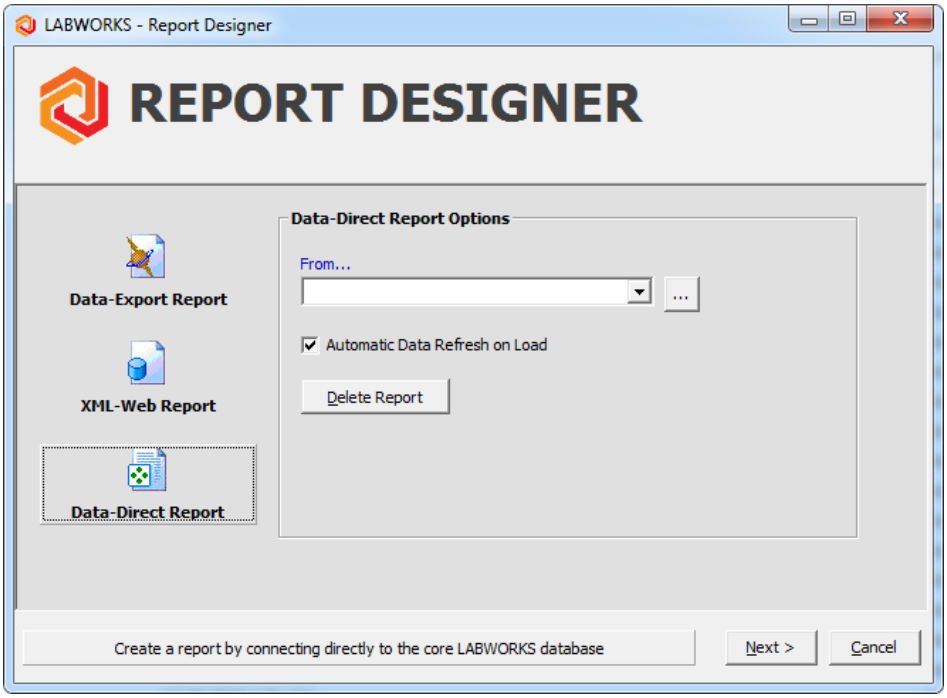
Access LABWORKS Report Designer Program

Users access the LABWORKS Report Designer program through the Options menu items on the LABWORKS Enterprise desktop or from a toolbar icon.

To access the LABWORKS Report Designer:

1. At the LABWORKS Desktop, click on the **Database** icon.
2. Open the **Applications** folder.
3. Open the **Reports** folder.
4. Double click on **Crystal Report Setup** program.

Notice that the Crystal Report Designer screen appears.



Enter Report Name and Save

The report name cannot have spaces and is limited to 24 characters. You may use the underline key to connect words, e.g., Summary_Report. If you want all your Direct Connect Crystal Reports to be easily recognized, consider beginning each report title with CRD_.

To enter a new report name:

1. Click on the **Data Direct Report** icon. Notice that an ellipsis button (…) now appears at the end of the **From** field and allows you to select from a list of previously set up formats.
2. To create a NEW report, enter a new report name in the **From** drop down menu.

   OR

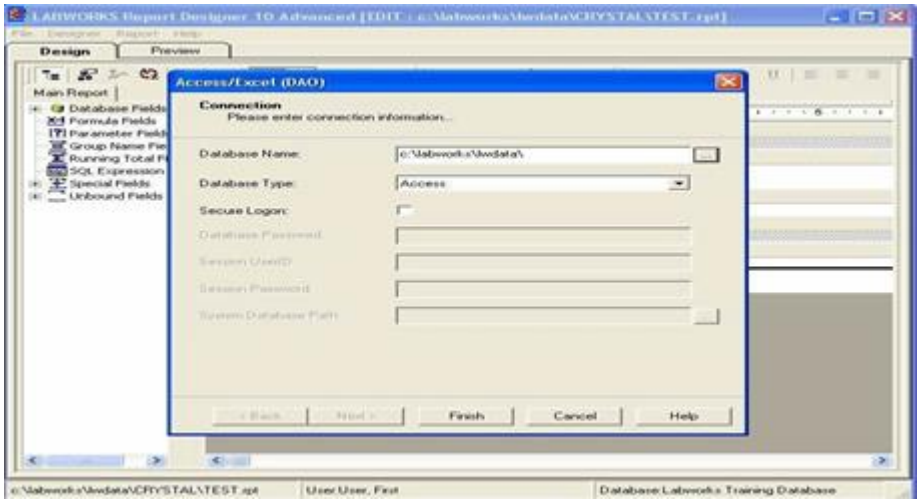
   To create a new report from an existing report, click on the ellipsis button at the end of **From** drop down menu and select the report you want to modify to create a new one (see Edit an Existing Report section).

3. Click **OK** to go to the next step.
4. Notice that when you click **OK** the LABWORKS Report Designer screen opens for setting up the report.

The program reads your registry to determine how you connect to your default LABWORKS database and opens up the Crystal Report editor.

To save the report:

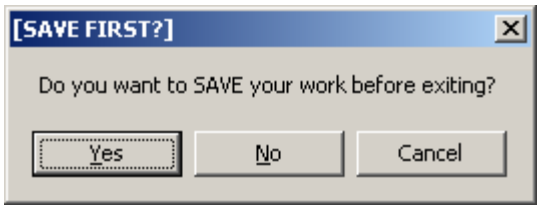
1. Before you begin setting up the report, save the report by clicking **File > Save Report**.



2. Save the report file with the **same name** that you entered on the Crystal Report Format Setup popup window.

3. Save it in the Crystal folder if you want to be able to select it from the Crystal Report Format Setup pull-down list or the list accessed when you click the ellipsis button at the end of the **Select Format Name** field.

4. If you attempt to close the report **before** saving it, the program provides a prompt that asks if you want to save before exiting.



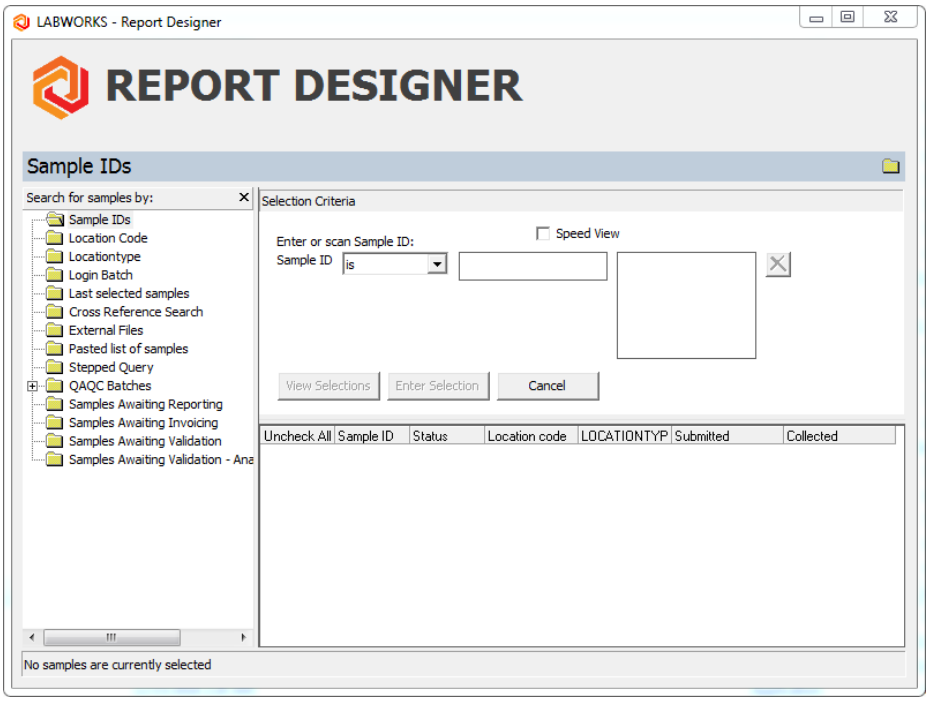
5. Click **Yes** on the Save First popup window to save your report; click No to exit without saving; or click Cancel to return to the design screen and continue creating or editing the format. Once saved, the report will appear in the Crystal Report Format Setup pull-down list for selection.

Select Samples

If you have not entered data in your LABWORKS LIMS, LABWORKS installs a database with test data that allows you to set up your reports before entering data.

To select sample data for the report:

1. If you have established data and know the ID for the specific samples to be included in the report, enter the Sample IDs in the cells in the LABWORKS Sample Selection screen. This is the standard LABWORKS sample selection screen used in other LABWORKS modules; if you need additional information on how to use this dialog, please refer to the Online Help.
2.

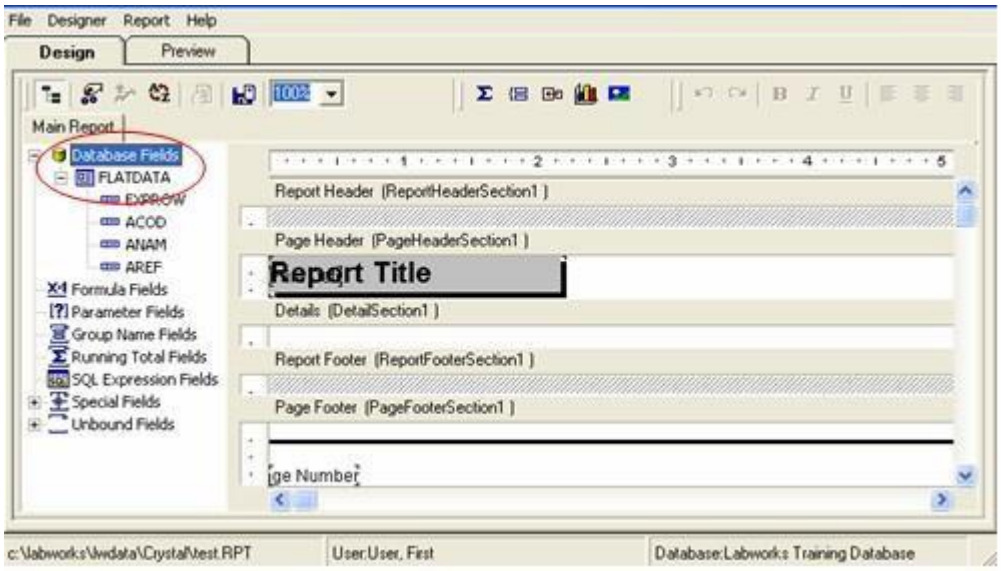


Add Database Fields to the Report

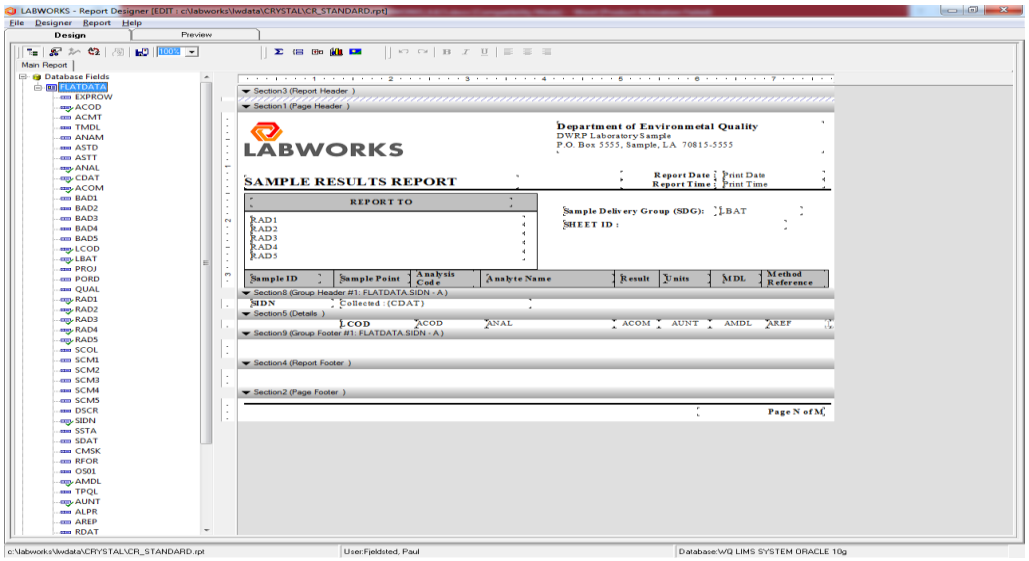
Add database fields to the report by following the steps in this section. It is important that you be familiar with database fields and have a model for creating the report before you start. Creating a report from an existing one greatly reduces time and effort.

To display and add database fields to the report:

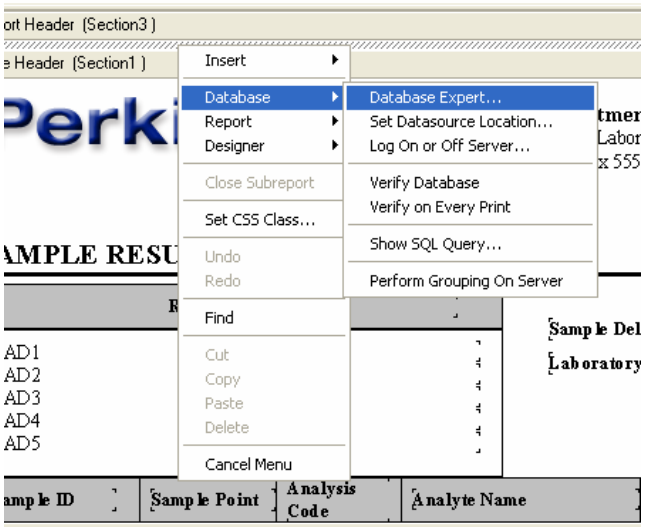
1. Double click **Database Fields** to display a list of LABWORKS database tables. When you double click **Database Fields** here using direct connect, a list of all the LABWORKS database tables is displayed.



2. Double click on the specific tables you want in your report to display a list of fields in that table.
3. Click and drag these database fields onto the **Details** area of the report or other area of the report as appropriate.



4. Right click anywhere under the **Database Fields** to access a list of pull-down menu items for enhancing the database, such as table linking or adding a new database or table (only useful in Direct Connect mode).

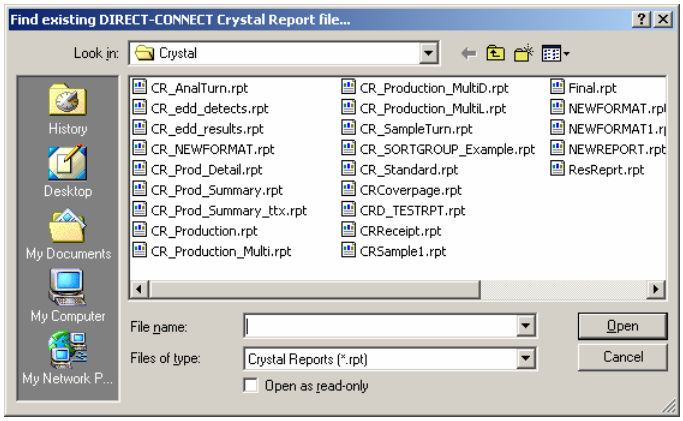


5. Notice that right clicking on any object or field accesses a popup menu to perform additional tasks and change attributes.
6. Refer to other section in this chapter for details on using these menu items.

Edit Existing Direct Connect Report

To edit an existing report:

1. From the Report Designer window click on the **Data Direct Report** icon.
2. To edit or create a new report from an existing report, click on the ellipsis button (…) at the end of **From** field.
3. Notice that the program opens the Find existing Direct Connect Crystal Report file selection screen and the folder where you have saved Crystal reports.

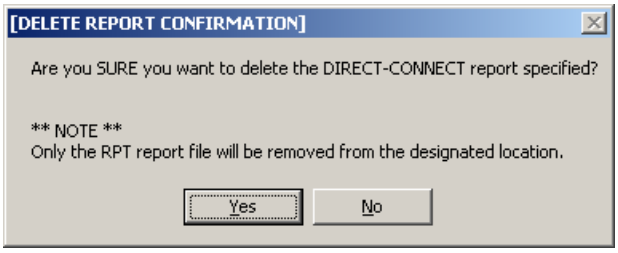


4. Click on the report you want to modify.
5. Click **Open**.
6. Click **OK** on the Crystal Report Format Setup popup window to go to the next step.
7. Before you begin editing, double-click **Database Fields** on the left to display the list.
8. Click and drag fields to your report as needed and begin editing.

Delete Direct Connect Report

To delete a report

1. To delete an existing Direct Connect report, enter the report name or click on the ellipsis button at the end of the **From** field.
2. Select the report to delete.
3. Click **Open.**
4. Make sure the name of the report that appears in the **From** field is the one you want to delete.
5. Click the **Delete** button.



6. Click **Yes** on the Delete Report Confirmation screen to delete the selected Direct-Connect report.

Menu Items and Special Editing Features

Menu Items and Special Editing Features

The LABWORKS Report Designer program provides almost all the functionality of the complete Crystal Reports™ software (standard edition) with some additional LABWORKS features for better integration and ease of report creation. The LABWORKS Report Designer screen contains five main sections for inserting data and several menu items to assist you in setting up the report. This chapter describes the report sections and menu items. In the LABWORKS Report Designer editor, nearly everything has a right click menu. Right click in any section or on any object on the screen to change properties, attributes, or settings.

If you need additional help creating complex reports using formulas, graphs, or other LABWORKS Report Designer functions, contact PerkinElmer Instruments for assistance. You may also request LABWORKS Report Designer Training.

Topics covered in this section include:

Designer Screen Sections

File Menu

Designer Menu

Report Menu

CEF File Menu

Help Menu

Use the Zoom Feature

Move and Resize Fields or Sections

Create Footers or Headers After the First Page

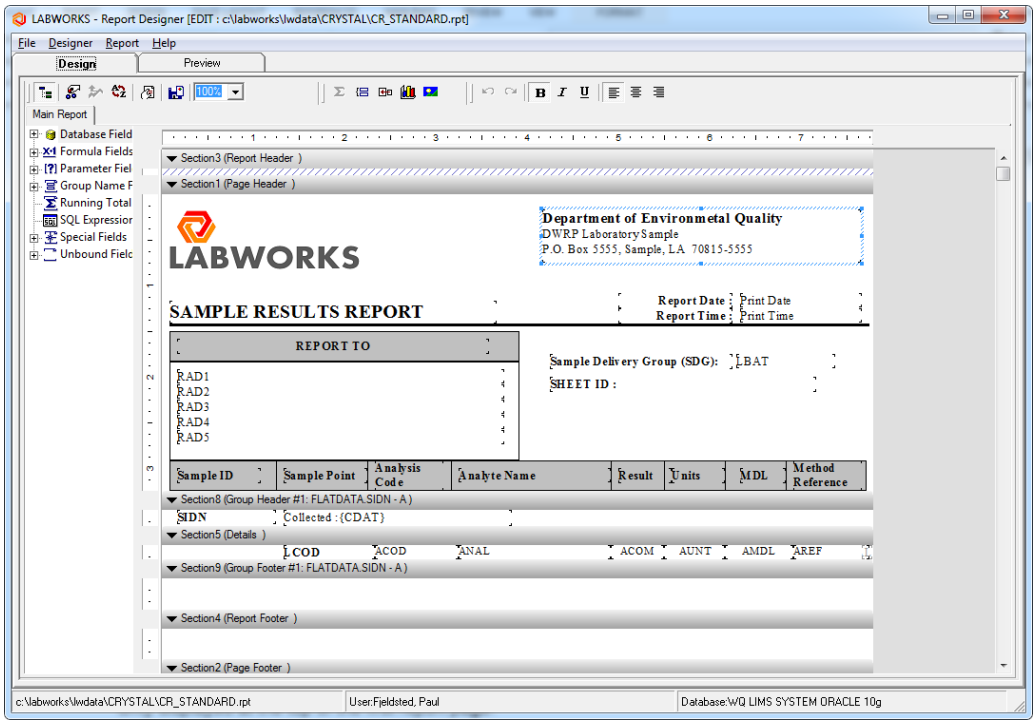
Add Summary Information to the Report

Inserting Fields

Save Report before Exiting

Designer Screen Sections

The LABWORKS Report Designer screen contains five main sections. These sections and a brief description of each are as follows.



Report Header

Generally used for the report title and other information that is to appear at the beginning of the report.

Only displayed at the top of the first report page.

Information here prints once at the beginning of the report.

Charts and Cross Tabs inserted in this area include data for the entire report.

Formulas inserted into this area are evaluated once at the beginning of the report.

Page Header

Generally used for information that that is to appear at the top of each page; for example, chapter names, document name, etc.

Can be used to display field titles above the fields on a report.

Objects inserted in this area print at the beginning of each new page.

Charts or Cross-Tabs cannot be inserted in this section.

Formulas inserted in this area are evaluated once per page at the beginning of each new page.

Details

Used for the body of the report.

Report data generally appears in this section.

Area where you drag fields to return multiple records – 1 detail line per record.

Objects inserted in this area print with each new record.

Charts or Cross-Tabs cannot be inserted in this area.

Formulas inserted in this area are evaluated once for each record.

Report Footer

Used for information that is to appear only once at the end of the report (such as summaries or totals) and for charts and cross-tabs.

Objects inserted in this area print once at the end of the report.

Charts and Cross-Tabs inserted in this area include data for the entire report.

Formulas inserted in this area are evaluated once at the end of the report.

Page Footer

Usually contains the page number and other information located that is to appear at the bottom of the page.

Objects inserted in this area print at the bottom of each page.

Charts and Cross-Tabs cannot be inserted in this area.

Formulas inserted in this area are evaluated once per page at the end of each new page.

Group Header

If a group, summary, or subtotal is added to the report, the program creates a Group Header area directly above the Details area, and the Group Footer area appears directly below the Details area. The Group Header area has two additional sections.

Contains the group name field.

Objects inserted in this area print at the beginning of each new group.

Charts and Cross-Tabs inserted in this area include data just for the group.

Formulas inserted in this area are evaluated once for each group at the beginning of the group.

Group Footer

Generally holds the summary value, if any.

Objects inserted in this area print at the end of each group.

Charts and Cross-Tabs inserted in this area include data just for the group.

Formulas inserted in this area are evaluated once for each group at the end of the group.

File Menu

The **File** menu gives you the following options.

- Open existing report – disregards the current report and opens another.

- **Save Report -** saves this report in one of the previous 2 versions (to support old systems) or currently installed version (if applicable) of Crystal Reports.
- **Save Report As…** - saves the report with a new name.
- Export Fields (CEF) – opens the CEF editor.
- **Printer Setup –** Windows setup dialog box for your installed printers.
- **Print Preview –** used to test your report (will run your report based on your exported test data or currently selected database).
- **Publish Report to WEB Server –** launches Crystal Enterprise Report Publication Wizard and publishes reports to a web site (if available); if not available, it prompts you to search for the program. If your laboratory has purchased and configured Crystal Enterprise for web reports, this function publishes reports to a web site. See your System Administrator for assistance.
- **Exit –** closes the LABWORKS Report Designer.

Designer Menu

The **Designer** menu gives you the following options.

- **Display –** a sub-menu of various setting preferences (enables or disables specific screen features of the designer).
- **Snap To Grid –** forces all report fields to fit within a screen grid to facilitate object alignment (with this option checked it is easier to line up objects in the report).
- **Grid Size –** allows you to set the length of the report grid cells in the report (smaller lengths make it MORE difficult to line up objects).

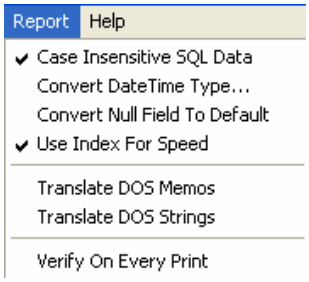
Report Menu

The **Report** menu has the following options.

- **Case Insensitive SQL Data –** data coming from a server-based source (**Direct Connect only)** will not look at case when this is checked (this is useful when using a record selection formula where case DOES matter).
- **Convert DateTime Type –** this function gives you the option to process incoming dates as string values or format them specific to a Date or Time standard.
- **Convert Null Field to Default –** when checked, the system treats nulls as blank spaces (otherwise their value is actually NULL).
- **Use Index For Speed – this** function improves report rendering by creating an additional index in memory.
- **Translate DOS Memos / Strings –** automatically translates old DOS text abbreviations for you at report creation.
- **Save Report As …** – this function allows the user to specify the version of LABWORKS Report Designer the opened report is saved as (versions 7 & 8 are the latest versions that support new data reporting technology shipped in LABWORKS, the “latest version” option saves the version

that is the latest version currently installed on your machine).

- **Verify On Every Print –** verifies the database is current (and does a refresh if it is not) each time the report is printed (slows down performance at run time).



CEF File Menu

The **CEF File** menu is available **only** for **export file reports** (NOT direct connect) and has the following options:

- **Add/Remove Database Fields –** opens the CEF editor screen (see Create CEF File section) and allows you to add or remove database fields to the currently opened report. After adding or removing fields, the CEF file is saved and the export MDB database file is recreated. The report then refreshes itself to reflect the new fields in the top left **Database Fields** dropdown menu.

- **Verify/Refresh Labworks Data –** refreshes the database if changes were made prior to reopening this report. Once this is done, the list of database fields under the **Database Fields** menu item reflects the new changes.

  You can refresh the database at any time to reflect new changes whether the report uses direct connect or exported data.

Help Menu

The Help menu gives you information about the version of Labworks you are running as well as any error log viewing if problems occur.

Use the Zoom Feature

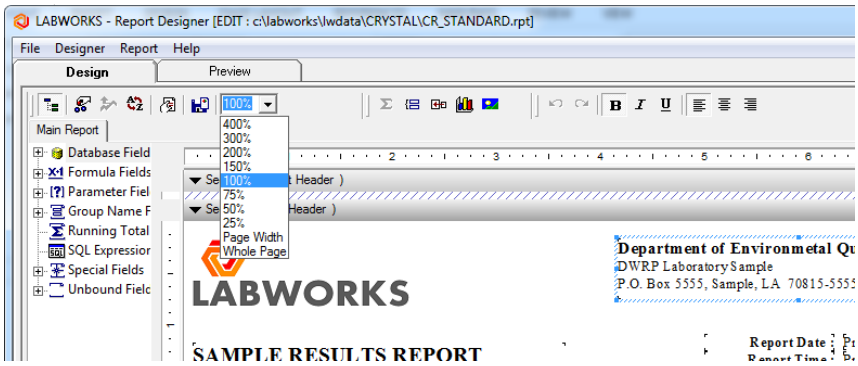
Use the Zoom Control box found on the Standard toolbar to set the zoom level for viewing the report while designing or previewing using the Print Preview function. The magnification options are from 25% to 400%, page width, or whole page.

To use the zoom feature:

1. Click on the down arrow at the end of the **Zoom Control box**.

2. Select the desired magnification from the options.

It is helpful to view reports at low magnifications so that you get an overall picture of the report layout. Use the higher magnifications to focus on report details.



Move and Resize Fields or Sections

Fields, objects, and sections may be resized by selecting and then dragging the edges to the desired size. Objects appear with a broken line frame with resizing handles when in the move/resize mode. In this mode, you can resize the field/object by dragging any of the resizing handles, or you can move it by placing the cursor inside the object and dragging it to a new location. You can also insert fields in this mode, but you cannot insert text.

To resize a field:

1. Click on the field to select it, e.g., report title field.
2. Move the cursor over the resizing handle on the edge you want to change until the cursor turns into a resizing cursor and drag the edge to the desired field width or length.
3. To resize two fields/objects at once, select one field, press the Ctrl key, and click on the other field to select both objects.

To resize or add space between rows or sections:

1. Move the cursor over the lower section boundary line until it changes to a resizing cursor.
2. Drag the line until the row is the desired width.
3. Alternatively, right click in a section and click Insert > Line from the pull-down menus. Using the Insert function, the program resizes the section automatically and adds the amount of space necessary to hold a line of typical database fields.

To move fields:

1. Click in the field to select it.
2. Drag the field to the desired location.

Add a Title

There are two ways to add a title to a report. One way is to use a text object; the other way is to set up the title information in the Title text box in the Document Properties dialog box.

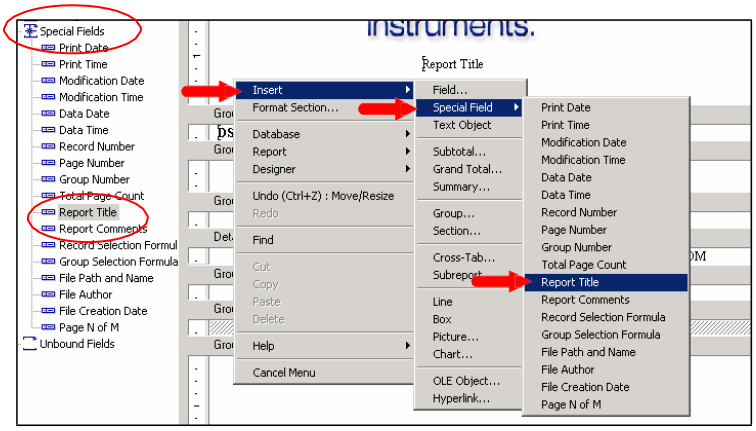
To insert a report title field:

1. Double click on Special Fields in the items on the left portion of the screen below **Database Fields** to display a list of special field choices.

2. Click on Report Title in the special field list and drag it to the **Page Header** area where you want it to appear on the report.

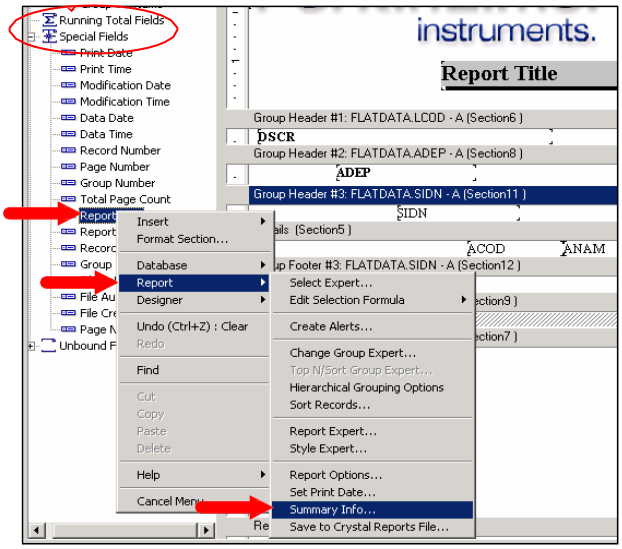
3. Alternatively, right click in the **Page Header** (or any blank) area; click **Insert**

    a. **Special Field > Report Title** in the series of pull-down menus.



To add or edit the report title text

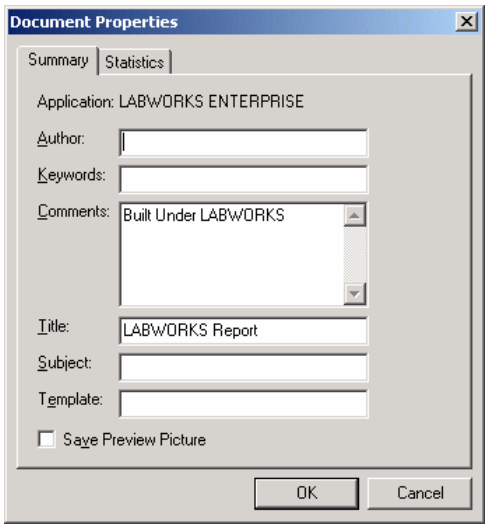
4. Double click on **Special Fields**.



5. Right click on **Report Title** special field.

6. Click on **Report** in the pull-down menu.

7. Click on **Summary info** in the next pull-down menu to open the Document Properties dialog box.

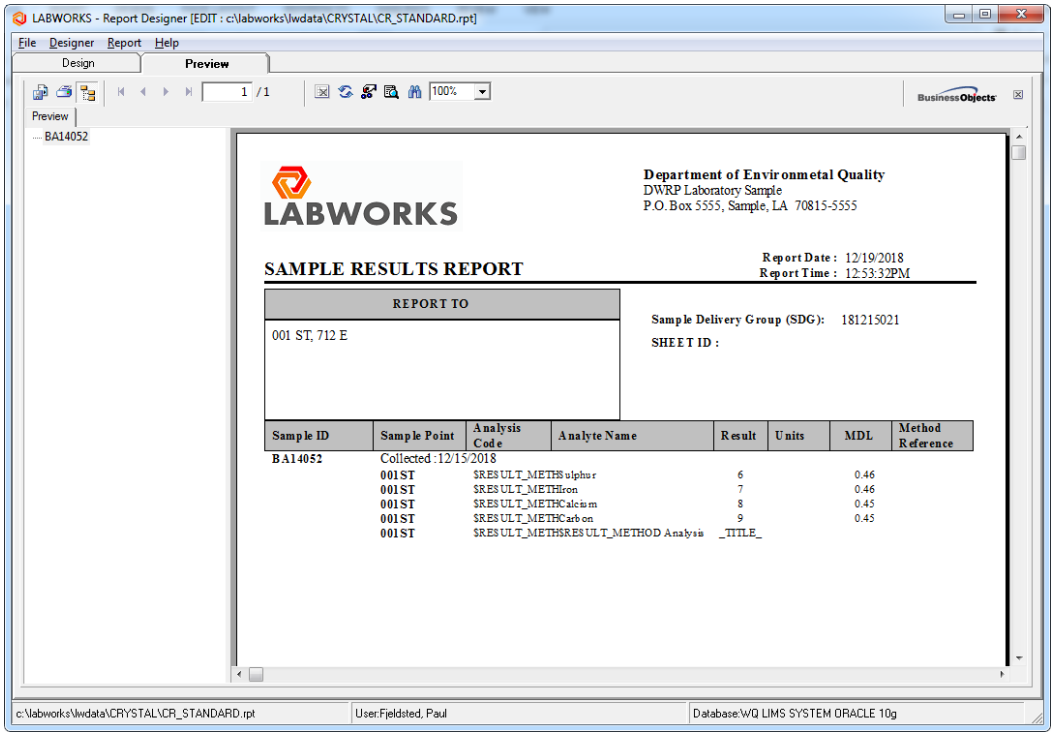


8. Enter the title of the report in the **Title text** field.

9. Add comments, author, keywords, subject, or template if desired.

10. Click **OK**.

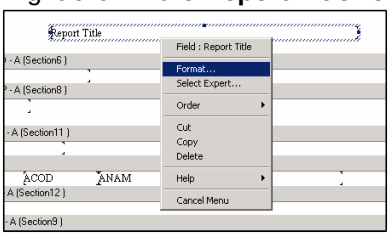
11. Click on **File** and click **Print Preview** to view the results.



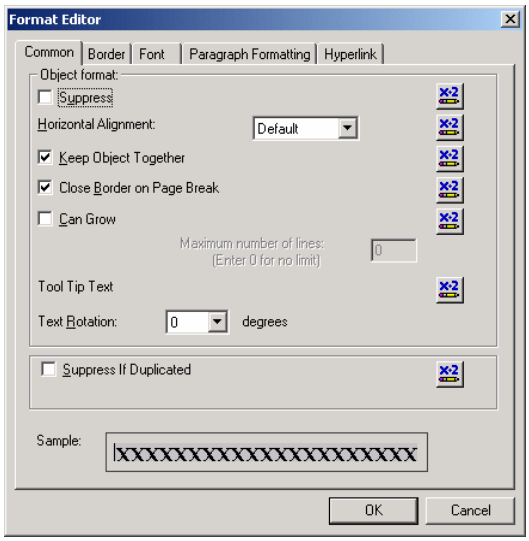
The report title object now displays the title that you entered in the Title text box of the Document Properties dialog box.

To format the title:

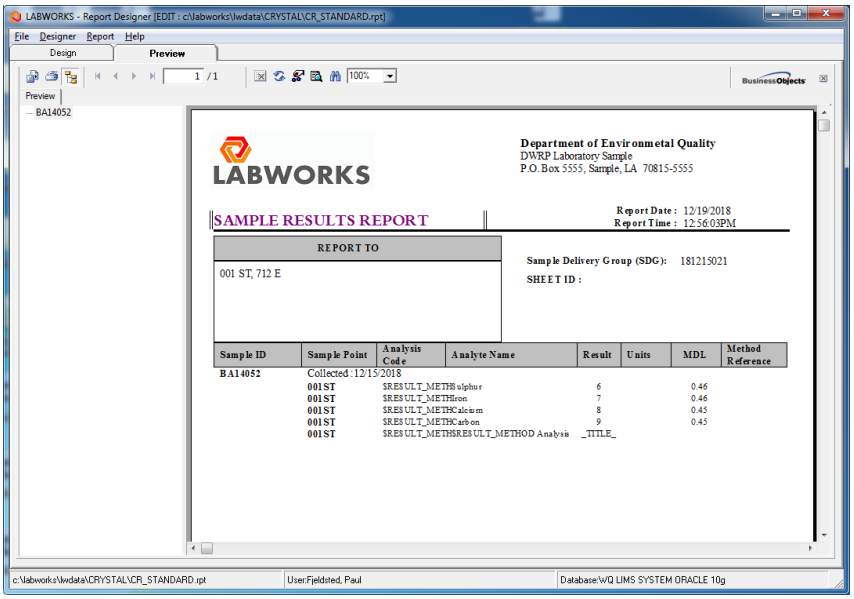
1. Close the Report Preview screen to return to the designer screen.
2. Right click in the **Report Title** field.



3. Click **Format** in the pull-down menu.



4. Click the **Border** tab to specify border color and background.
5. Click the **Font** tab to select the size and type of font.
6. Click the **Paragraph Formatting** tab to set up how you want the title to appear, e.g., indentation, spacing.
7. Click the **Hyperlink** tab if you want to add a hyperlink to the title.
8. Click **OK** when finished.
9. Click **File** and click **Print Preview** to preview the results. The report now shows the new formatted title as it will appear on the report.



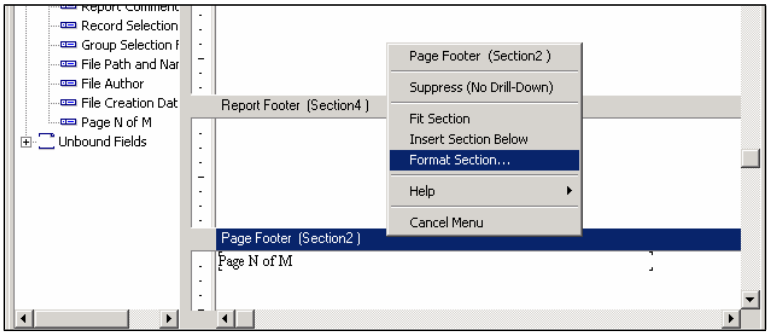
10. Close the Report Preview screen to return to the designer screen to continue editing.

Create Footers or Headers after the First Page

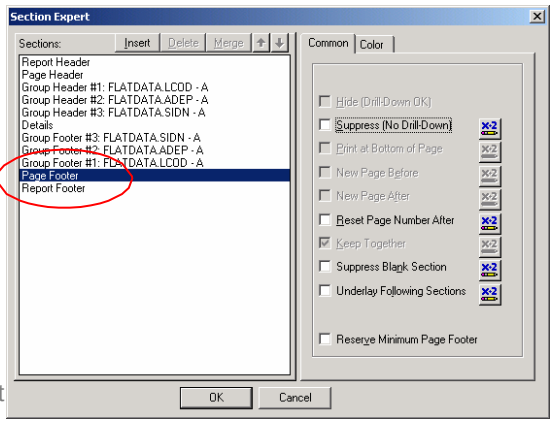
If you want to print a page footer or header on all pages except the first page, you can do this by conditionally formatting the Page Footer or Page Header section using an on or off property.

To create footers after the first page:

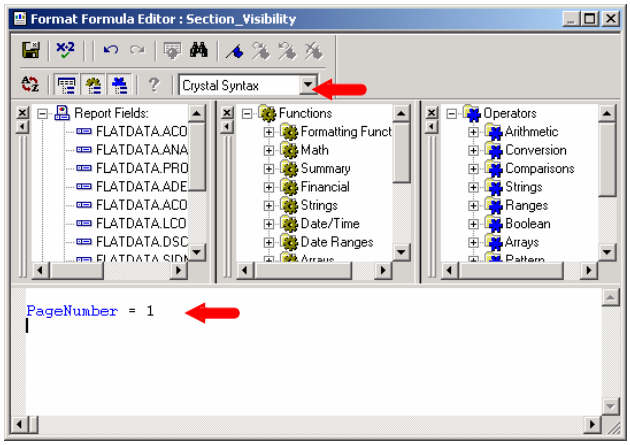
1. Place the field you want displayed as a page footer in the **Page Footer** section of the report.
2. Right click on **Page Footer** section.



3. Click **Format Section** on the menu to access the Section Expert dialog box.



4. Click **Page Footer** in the Sections area.
5. Click the **Formula** button to the right of the Suppress (No Drill-Down) checkbox to open the Format Formula Editor.



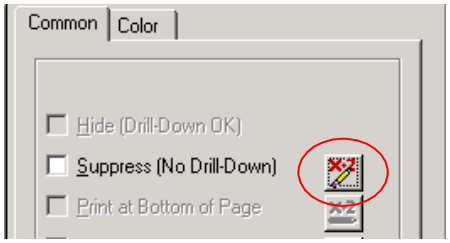
6. Enter the following formula in the Format Formula Editor: Crystal syntax

   example: PageNumber = 1

   Basic syntax example: Formula = PageNumber = 1

   Either of these formulas suppresses the page footer on the first page but not on any of the other pages. If there is an error in the formula, a message box appears, asking whether to exit without examining the error. If you click No, a second message box is displayed, detailing the error. If there is no error in the formula, you are returned to the Section Expert.
7. Click the Save button on the Formal Formula Editor.

Notice that the formula button on the Format Formula Editor screen has changed.



8. Click **OK** to return to the designer screen.
9. Click **File** and click **Print Preview** to preview the report and ensure that the page footer appears on all pages but the first. If you have a multi-line page footer and have inserted the lines into separate Page Footer sections, you will need to suppress each section conditionally, using the formula above.

To create a page header on all pages but the first:

1. Enter the header information in the **Page Header** section.

2. Suppress that section conditionally using the same formula that was used for suppressing the **Page Footer** section as described above.

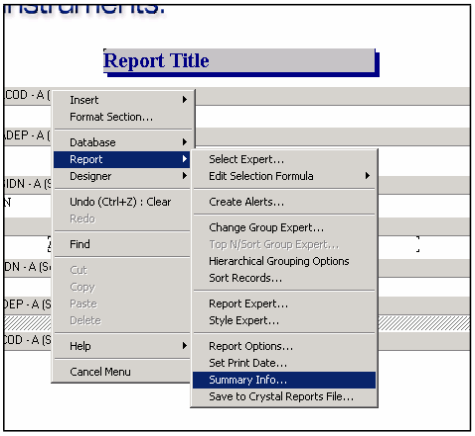
Add Summary Information to the Report

You may want to include non-printing comments with a report such as a note to explain the data included in the report, a report title, a comment about some particular data on the report, etc. Use the Summary

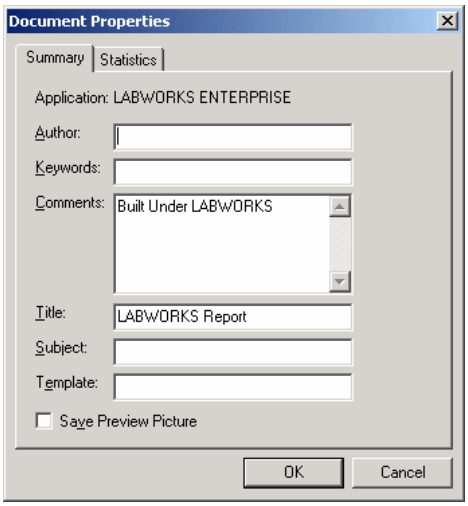
Info feature for including short or long notes of text with your report. The comments do not print with the report and remain in the Summary tab of the Document Properties dialog box where they can be reviewed.

To enter summary information:

1. Right click on a blank area of the report.
2. Click on **Report** in the pull-down menu.



3. Click on **Summary info** in the next pull-down menu to open the Document Properties dialog box.



4. Enter Summary information for the report in the **Comments** field of the Document Properties dialog box.
5. Click **OK** to save.

Inserting Fields

The LABWORKS Report Designer allows you to insert a variety of fields in the report such as general text fields (titles or column headings), special fields such as page numbers, formula fields, or parameter fields by right clicking on a blank area of the report to access a list of menu items. The general menu contains a wide range of options.

These options launch various Crystal Report functions such as the formula editor, the parameter field editor, and the chart export. How to insert formula fields, parameter fields, charts, and other fields are discussed in individual sections in this chapter. Refer to those for details.

Save Reporting before Exiting

To save the report before exiting:

Note: Remember, if you want to work on the report later, save it before you exit.

1. Click **File.**
2. Select **Save Report** from the pull-down menu to save the report.

Record Selection

Record Selection

Record selection allows you to select and include only the data required for your report. It is an important step in report creation. Usually, you do not need a listing

of all the information in a database, but you need certain segments of that database information. For example, you may want to compile data regarding a particular product for a certain time period.

This sectionr includes the following topics regarding record selection.

- Selecting Records
- Using the Select Expert
- Using Formulas
- Troubleshooting Record Selection Formulas

Selecting Records

Field values for the data fields you selected for the report are printed by default

from every record in the active tables. You may not want to include all the values or only a subset of those values. For example, you may want to include:

- Records for only a specific group of samples.
- Records for a specific range of sample IDs.
- Values that fall within a particular date range.

Options for selecting records

LABWORKS Report Designer program includes a formula language that you can use to specify virtually any type of record selection. In those instances in which you may not need the flexibility in record selection that the formula language provides, you can use the Select Expert. You can select records in one of two ways:

- Use the Select Expert.
- Use formulas.

After you become familiar with the Select Expert and formulas, you can use your record selection techniques to improve the performance of your reports.

Determining which field(s) to use

When you select records, you are setting up the report to include only those records that meet your specified conditions. You base those conditions on the kind of information you want in the finished report. For example, you want a report that shows only data from the Environmental Laboratory Department. The challenge is to find the best way to identify those records. If the table used in a report has a department field, you can specify in your request that the program use only those records in which the value in the department field is equal to Environmental

Laboratory. Generally, if you can base your record selection on a number of fields, you should select an indexed field instead of a field that is not indexed.

Using the Select Expert

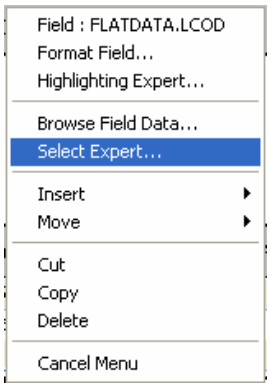
The Select Expert makes it easy to specify the records you want included in your report. Using the Select Expert allows you to select the field to which you want to apply selection conditions and then specify those conditions. The Select Expert can be used to set up simple or complicated record selection requests.

For range limit requests, one or more constants define the range. The program compares the field value in each record to the constant(s) and rejects records with values outside the range. The report is limited to values within the range. You can set up all of these types of record selection requests without any previous knowledge of the formula language.

The Select Expert can be used to set up both record selection and group selection requests. When a group name or summary field is selected, the program knows that the selection criteria set up is intended for group selection. In all other cases, the program knows that you are setting up record selection.

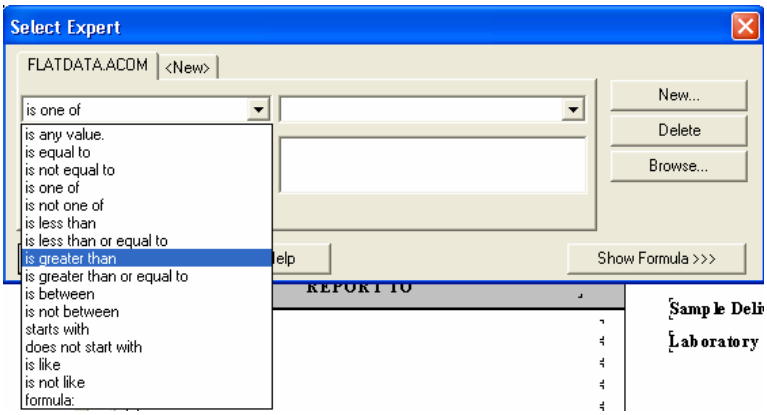
To set up record selection using the Select Expert:

1. Right click on the field on which you want to base record selection and click **Select Expert.**



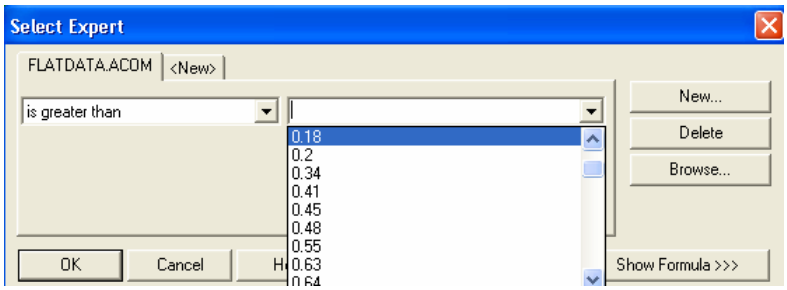
If you click the **Select Expert** button without first highlighting a field in your report, the Choose Field dialog box appears.

2. Notice that the Select Expert popup window appears.



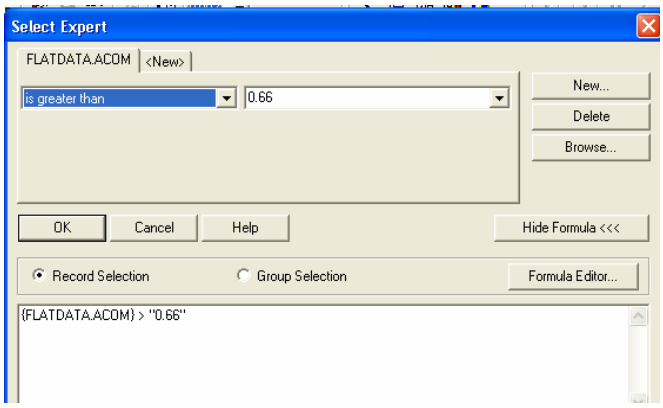
3. Click on the down arrow at the end of the text field to select the value for limiting the record selection, e.g., is greater than.

4. Notice that another text field appears on the Select Expert window for entering the limiting value.



5. Enter a limiting value in the field or click the down arrow at the end of the field to select from the listed values.

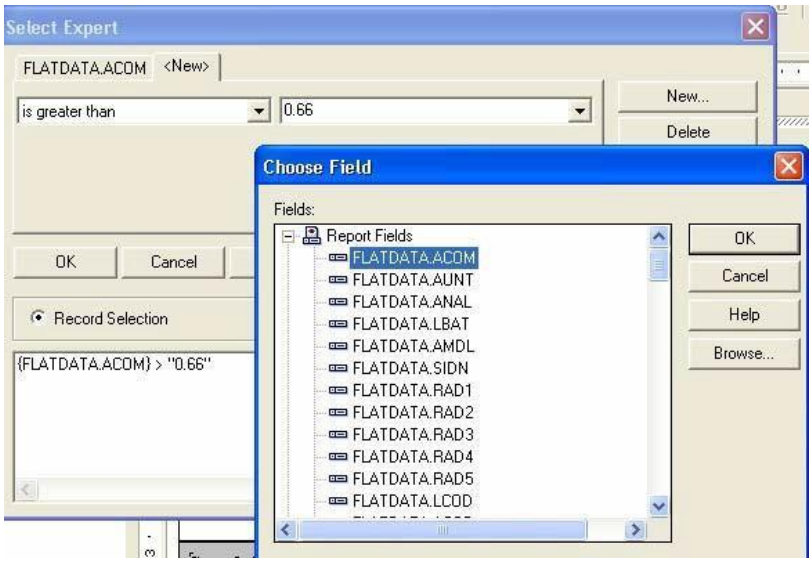
6. Click on the **Show Formula** button to view the formula.



7. Click **OK** when finished.

To base record selection on more than one field:

1. Click the **New** button on the Select Expert window.

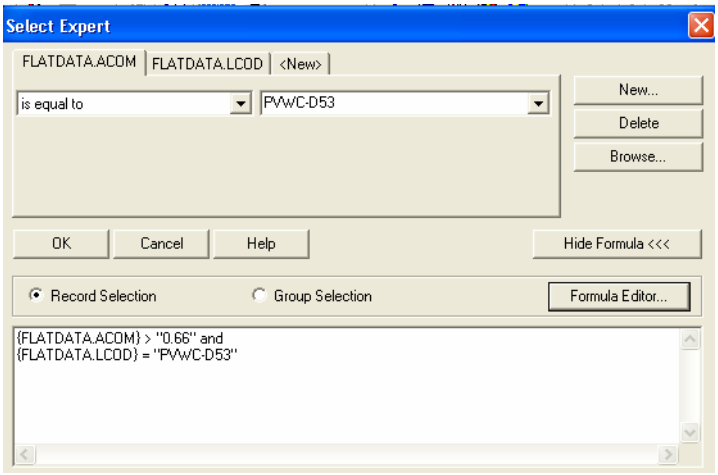


2. Select the next field from the **Choose Field** dialog box.

3. Click **OK**.

4. Click on the **Formula Editor** button to view the formula.

A selection formula is generated based on your specifications that limit the report to the records you indicated.



To modify the formula:

1. Click the **Show Formula** button to view the selection formula.

2. Click the **Formula Editor** button to modify the formula.

To delete a selection or formula:

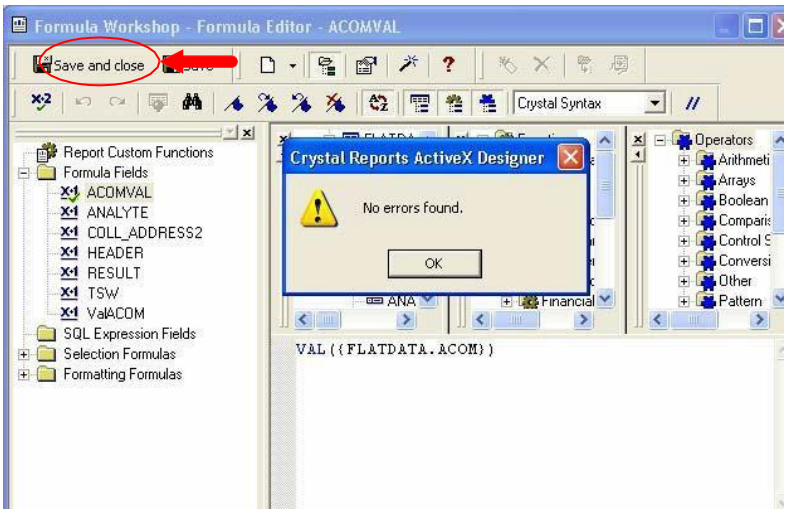
1. Click on the tab of the selection you want to delete.

2. Click the **Delete** button.

Using Formulas

To set up record selection using a formula:

1. Click on the selection field and click **Select Expert** as described in the Select
   a. Expert section above.
2. Click **Show Formula**.
3. Click **Formula Editor**.
4. Enter the formula by typing in the components or selecting them from the component trees.



5. Click Check **to identify any errors** in the formula.
6. Fix any syntax errors the Formula Checker identifies and click **OK**.
7. Click the **Save and Close** button.
8. Edit the formulas as needed and save.
9. Click the **New** tab to add new formulas, if desired.
10. Select the formula and click Delete to delete a formula.

Example Formulas

This section provides some example formulas that you may use or edit and save as templates for creating your formulas for record selection.

To select records using character strings:

Formula: {file.FIELD} startswith "P"

Selects records in which the value in the {file.FIELD} field begins with the character "P"; includes values like PerkinElmer, Parkway, and Pennsylvania; excludes values that do not begin with P.

Formula: not ({file.FIELD} startswith "P")

Selects records in which the value in the {file.FIELD} field does not begin with the character "P"; includes all values that do not begin with “P”; excludes values that begin with “P”.

Formula: "999" in {file.FIELD}[2 to 4]

Selects records in which the 2nd through 4th digits of the {file.FIELD} field is equal to "999" (includes values like 1999, 7999, and 0999; excludes values like
99901 and 19990).

Formula: "Perkin" in {file.FIELD}

Selects records in which the value in the {file.FIELD} field contains the string "Perkin"; includes values such as PerkinElmer, PerkinElmer Instruments; PerkinElmer Labworks; excludes values that do not begin with “Perkin”

To select records using numbers:

Single values

Formula: {file.FIELD} > 99999

Selects records in which the value in the {file.FIELD} field is greater than
99999.

Formula: {file.FIELD} < 99999

Selects records in which the value in the {file.FIELD} field is less than 99999. Range of values

Formula: {file.FIELD} > 11111 and {file.FIELD} < 99999

Selects records in which the value in the {file.FIELD} field is greater than
11111 but less than 99999; neither 11111 nor 99999 is included in the range of values.

Formulas: {file.FIELD} >= 11111 and

{file.FIELD} <= 99999

Selects records in which the value in the {file.FIELD} field is greater than
11111 but less than 99999; both 11111 and 99999 are included in the range of values.

To select records using dates:

The Month, Day, and Year functions can all be used in examples like the following:

Formula: Year ({file.DATE}) < 2000

Selects records in which the year found in the {file.DATE} field is earlier than
2000.

Formula: Year ({file.DATE}) > 2000 and

Year ({file.DATE}) < 2002

Selects records in which the year found in the {file.DATE} field falls between
2000 and 2002; 2000 and 2002 not included.

Formula: Year({file.DATE}) >= 2000 and

Year({file.DATE}) <= 2002

Selects records in which the year found in the {file.DATE} field falls between
2000 and 2002; 2000 and 2002 are included.

Formula: Month({file.DATE}) in 1 to 3

Selects records in which the month found in the {file.DATE} field is one of the first three months of the year; includes January, February, and March..

Formula: Month({file.DATE}) in [1,3]

Selects records in which the month found in the {file.DATE} field is the first or third month of the year; includes January and March, excludes February.

To select records using preset data ranges:

The preset date ranges can be used to create selection formulas similar to these:

Formula: {file.DATE} in LastFullMonth

Selects records in which the date found in the {file.DATE} field falls within the last full month; if month is June, this selects all records with a May date.

Formula: not({file.DATE} in LastFullMonth)

Selects all records except those in which the date found in the {file.DATE} field falls within the last full month; if the month is June, this selects all records except those with a May date).

Formula: {file.DATE} < CurrentDate

Selects all records in which the date found in the {file.DATE} field falls before today's date.

To select records using date/number/character combinations:

These formulas simply mix and match formulas from the categories above.

Formula: "P" in {file.FIELD}[1] and Month({file.DATE}) in [1,3]

Selects records in which the value in the {file.FIELD} field begins with "P" and the month is either January or March. For example, if this kind of formula was used with an order database, you could be asking for a report showing all customers whose names begin with "P" and who placed orders in January or in March.

Formula: "AOK" in {file.HISTORY}[2 to 4] and {file.OPENACCT} >= 5000

Selects records in which the {file.HISTORY} field shows the characters "AOK" as the 2, 3, and 4 characters and the {file.OPENACCT} field (the amount of their account) is at least 5000.

These templates can be used as they are with your own data, or they can be combined to create complex formulas.

Troubleshooting Record Selection Formulas

To troubleshoot your selection formula, begin by ensuring that all the fields referenced in the selection formula are inserted on your report. Then delete the selection formula, and test it as you rebuild it.

To troubleshoot record selection formulas:

1. Write down the record selection formula on paper. This written copy will help you reconstruct the selection formula one step at a time.

2. Delete the record selection formula from your report by deleting the formula from the formula text box in the Record Selection Formula Editor.

3. Click **Save and Close** when finished.

4. Make sure that all fields referenced in the record selection formula are on the report and are not hidden.

   For example, if one of the selectors is:

   {customer.POSTAL CODE} > "80000"

   but the {customer.POSTAL CODE} field is not used on your report, then insert the {customer.POSTAL CODE} field into the report.

5. If one of the fields referenced in the selection formula is on the report but is hidden, unhide it

by deselecting the Suppress option in the Format Editor for that field.

6. Print the report and verify that the data in those fields that are referenced in the selection formula print satisfactorily

7. Make certain that all the data prints. For example, if there are eight total records in the database you should have eight records printing for each of the referenced fields. This establishes a baseline against which you can compare the results of printing with the selection formula.
8. When you are obtaining satisfactory results without using the selection formula, enter the selection formula using only one of the selectors.

   For example, to use this as the final selection formula:

   {customer.POSTAL CODE} > "80000" and {customer.CONTACT LAST NAME}[1] = "P" and {customer.LAST YEAR'S SALES} >= 5000

   this formula will select all of those records that show a Postal code greater than 80000, a value in the {customer.CONTACT LAST NAME} field beginning with "P", and a value in the {customer.LAST YEAR'S SALES} field greater than or equal to 5000.

   You might start with this as the first test selection formula: {customer.POSTAL CODE} > "80000"

9. Print the report and evaluate the data that prints when you have only one selector activated. If it does, then you know that this part of the selection formula is working. If it does not, then troubleshoot this part of the selection formula.

10. When the selection formula with one selector activated is working properly, add a second selector.

11. Print the report and evaluate the data that prints when you have two selectors activated. If it does, then this part of the selection formula is working. If it does not, then troubleshoot this part of the selection formula.

12. When the selection formula with two selectors activated is working properly, add a third selector, then a fourth, etc., until you have tested each selector in the selection formula.

Correcting selections that do not generate data

You may encounter a situation in which you create a record selection formula and, while header and footer information prints on your report, no detail information appears. The problem is the selection formula is rejecting all records. This usually occurs because of an error in the creation of the selection formula.

There are a couple of potential causes of your problem in the selection formula:

To correct uppercase/lowercase inconsistency:

1. Record selection formulas are case sensitive. This means that “PerkinElmer” only matches with “PerkinElmer”. It does not match with "perkinelmer", “PERKINELMER”, “Perkin Elmer”, “Perkinelmer”, or “perkinElmer”. If your selection formula is set to include only those records with "PERKINELMER" in the {customer.CONTACT NAME} field, but all the entries in the {customer.CONTACT NAME} field are mixed case, as "PerkinElmer", the selection formula will

not find any matches and will not print any details for the report.

2. To solve this problem, use the UpperCase (str) or LowerCase (str) functions in your selection formula to convert field data to a consistent case before the program begins its selection.

   For example, if you were using this formula: {customer.CONTACT NAME} = "PERKINELMER", change the formula to this: UpperCase({customer.CONTACT NAME}) = "BOB"

   This second formula first converts the value of the {customer.CONTACT NAME} field to upper case characters and then checks to see if the value in that field is equal to "PERKINELMER". Using this formula, any instance of the letters “PerkinElmer” will be a match, regardless of case, because the case will be converted to uppercase.

3. Consider using the LowerCase function in a similar manner to match with “PerkinElmer”.

4. Check your selection formula closely and ensure you have the correct case for any text you are trying to match. If in doubt, use the UpperCase or LowerCase function to ensure consistency and proper matching.

To correct unwanted spaces in selection formula:

Spaces are characters, and when you include spaces in the search key of a record selection formula, the formula looks for records with the exact match in the selected field including spaces.

For example, the following formula: "Ms . " in {customer.TITLE} will not find any matches with the form of address "Ms." because there is an extra space in the search key between the letter "s" and the period. Likewise, "Ph. D" will not match "Ph.D".

Check your selection formula closely and make sure that the spaces in the selection formula match the spaces in the fields you are trying to match.

Sorting and Grouping Data

Sorting Data

Sorting is used to place data in an order that helps you find and evaluate it. When you first insert a database field into your report, the data within the fields appears in the order in which it was originally entered into the database. Finding information in this kind of report is difficult. Sorting helps make it easier to review or find information in a logical format.

Sort Options

When you sort, the program asks you to define two things:

- Field on which to base the sorting
- Sort direction

Sort field

A sort field is the field that determines the order in which data appears on your report. Almost any field can be used as a sort field. A field's data type determines the method in which the data from that field is sorted.

Field Type	Sort Order
Single-character string fields	Blanks punctuation numbers uppercase letter lowercase letters
Multiple character string field	two letters three letters four letters, etc.
Currency fields	numeric order

Field Type	Sort Order
Number fields	numeric order
Date fields	chronological order
DateTime fields	chronological order same-date values sorted by time
Time fields	chronological order
Boolean comparison fields	False values (0) True values (1)
Null values	null values non-null values

Sort direction

The sort direction is the order in which values are displayed once sorted.
Ascending – sorts from smallest to largest based on the values in the sort field
Descending – sorts largest to smallest based on the values in the sort field

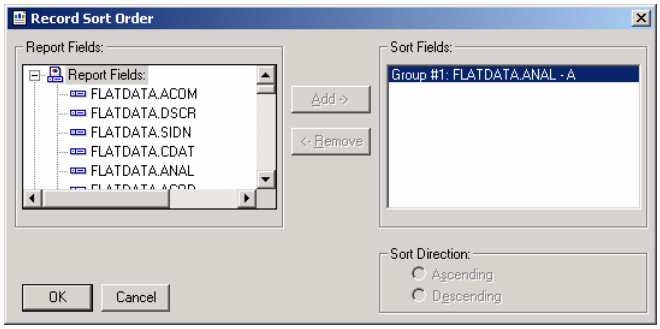
Sorting single and multiple fields

For single field sorting, all the records used in the report are sorted based on the values in a single field. For multiple field sorting, the LABWORKS Report Designer program first sorts the records based on the values in the first selected field and places them in ascending or descending order as specified. When two or more records have the same field value in the first sort field, the program then sorts those records based on the value in the second sort field.
To sort data:

1. Right click on a blank area of the report.

2. Click on **Report** in the pull-down menu.

3. Click **Sort Records** in the next pull-down menu.



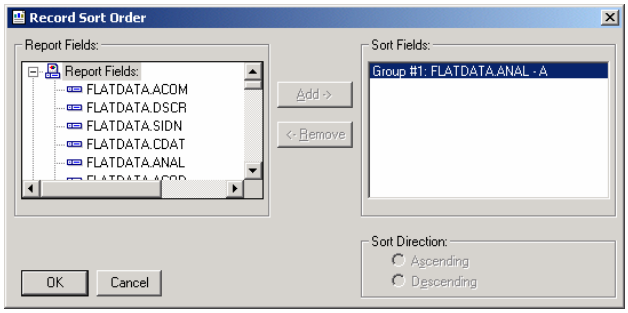
4. Click on the field to be sorted from in the **Report Fields** list.
5. Click the **Add button**.
6. Notice that the selected field is added to the **Sort Fields** list.
7. Click **Ascending** or Descending to specify the sort direction.
8. If sorting by more than one field, select the second field you want the data to be sorted by and add it to the **Sort Fields** list. The order of the fields listed in the Sort fields box is the order by which data will be sorted.
9. Select the sort direction.
10. Click **OK** when finished. Records are sorted based on the values in the Sort Fields list.

To sort records within groups:

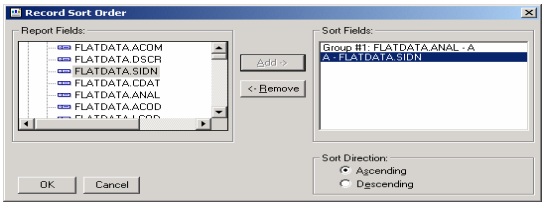
If you have grouped your data, you can sort the records within the groups.

1. Right click on the report and click **Report** in the pull-down menu.

2. Then click **Sort Records** in the next pull-down menu. Sort fields that begin with Group indicate that the sort was done automatically when the data was grouped.



3. Select the field by which you want the records sorted within the groups; and then click the **Add** button to add it to the **Sort Fields** list. The order of the fields in the Sort Fields box is the order by which data will be sorted.



4. Specify the sort direction.

5. Click **OK** when finished.

Grouping Data

Grouped data is sorted data that is divided into respective planned groups for organizing in a logical manner that makes data handling and processing more useful.

Group and sort direction

Four sort and group direction options are available for grouped data.

Ascending – order is smallest to largest; records are sorted in ascending order and then begins a new group whenever the value changes.

Descending – order is means largest to smallest; records are sorted in descending order and then begins a new group whenever the value changes.

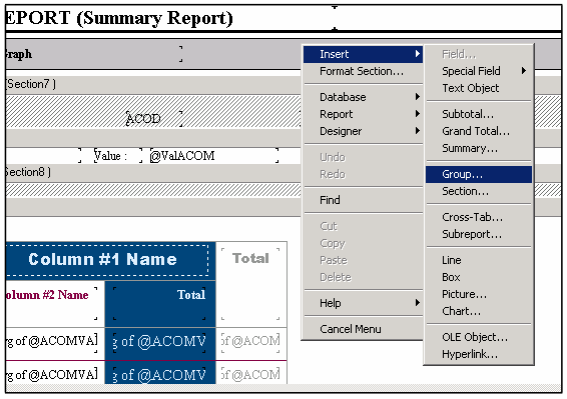
Original – order that the data was originally saved in the database; program leaves the records in the order in which they appear in their originating database table and begins a new group whenever the value changes in the group field you select.

Specified order – order is a user-defined order; program places each record into the custom group you specify, leaving the records in each group in original order or it sorts them in ascending or descending order, depending on your instructions.

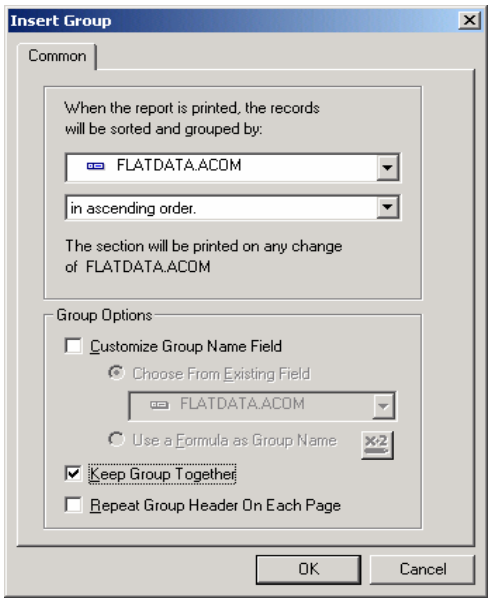
To group data:

1. Click in a blank area of the report.

2. Click **Insert** on the pull-down menu.



3. Click **Group** in the next pull-down menu. The Insert Group dialog box appears.



4. Select the field you want the data grouped by from the top drop-down list.

5. Select the sort direction from the second drop-down list.

6. Select the Customize Group Name Field checkbox if you want to show a different value in the group header. By default, the group header of the report displays the value of the field you are grouping by.

7. If you want to hide the group header name, right-click the group header, select Format Field, and click Suppress on the Common tab of the Format Editor.

8. Click **OK**.

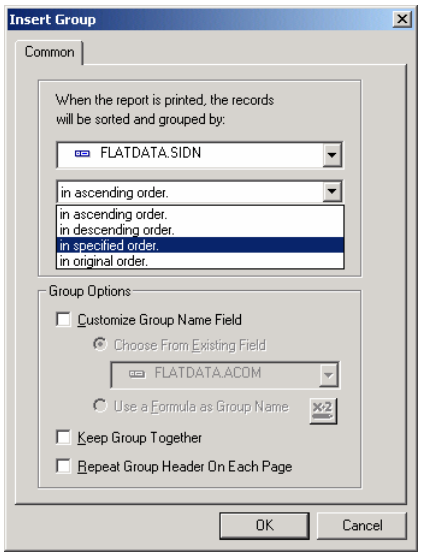
9. If your records within each group are unsorted, you need to sort the records within each group using the procedure in the previous section To sort records within groups.

Creating custom groups

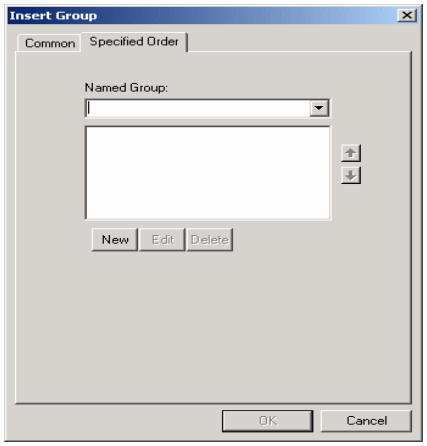
Data is usually sorted based on the values from a field in the report. If you want to group data based on the values found in a field that is not on your report, you may create a custom group.

To create a custom group:

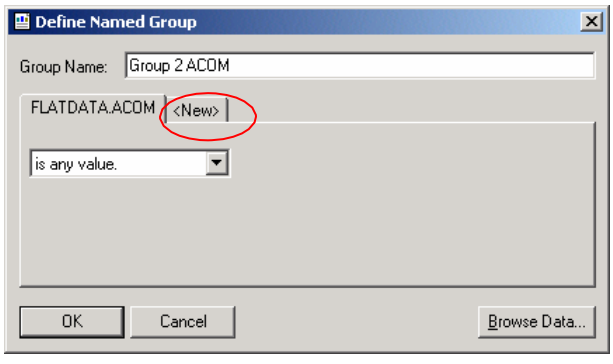
1. Click in a blank area of the report.
2. Click **Insert** on the pull-down menu.
3. Click **Group** in the next pull-down menu. The Insert Group dialog box appears.



4. Select the field the data is to be grouped by from the top drop-down list.
5. Select **in specified order** as sort option from the second drop-down list.



6. Enter the name of the group in the **Named Group** field on the Specified Order tab.
7. Click **New**.



8. In the Defined Named Group dialog box, use the drop-down lists to select the data to be part of the group.
9. Click the **<New>** tab to add more selection criteria to your specified group, if necessary.
10. Click **OK**.
11. Click **New** to create more custom groups as necessary.
12. Click the **Others** tab to specify how you want to organize the data that is not part of the group(s) you defined.
13. Click **OK**.

Selecting group using select expert

When you group or summarize data, all the groups in the report are included by default. If you do not want to include all the groups, you can select the groups that appear in the report in by using the Select Expert or selection formulas.

The Select Expert can be used to select groups of records in the same way that you select individual records. When setting up group selection criteria, instead of basing the selection criteria on standard fields, as you do for record selection, you base the criteria on group name fields or summary fields.

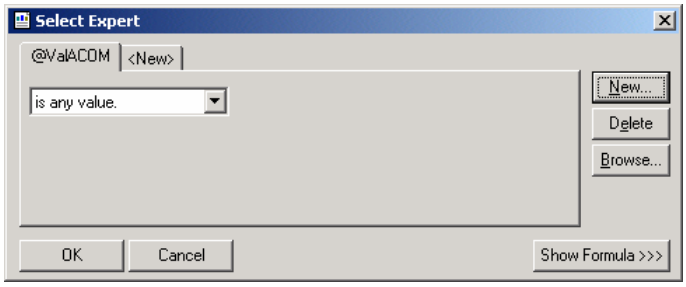
If you have grouped your data but have not summarized it, you can only set up group selection based on the group name field. If you have summarized your data, you can set up group selection based on either the group name field or the summary field.

The Select Expert can be used to set up record selection and group selection requests. When a group name or summary field is selected, the program knows that the selection criteria you set up is intended for group selection. In all other cases, the program knows that you are setting up record selection.

To set up group selection using the Select Expert:

1. Right-click the summary field on which you want to base group selection

2. Click **Select Expert** from the menu. The Select Expert dialog box appears. If you click the Select Expert button without first selecting a summary field in your report, the Choose Field dialog box appears.



3. Click on the down arrow of the drop-down list to enter your selection criteria for the indicated field.

4. To base the group selection on more than one field, click the **New** tab and choose the next field from the Choose Field dialog box.

NOTE: If you have not already previewed the report or refreshed the data, there will not be any data saved with the report. Without the data, the program cannot calculate group values, which means that no values appear when you click the arrow in the right drop-down list. In this case, you must type in the values you want. If you want real values to work with, you need to preview your report first. This calculates the actual summary values available for you to work with.

5. Click **OK** when finished in the Select Expert to return to the report.

Selecting group using selection formulas

With the Formula Editor, you can build your group selection request using group fields, group name fields, and other formulas. As with record selection formulas, your only restriction is that the formula you create must be Boolean; that is, it must return either a True or False value.

To create a record or group selection formula:

1. Right click on a blank area of the report and click **Report** in the pull-down menu.
2. Click **Edit Selection Formula** in the next pull-down menu.
3. Click **Record** to create a record selection formula.

   OR
   Click **Group** to create a group selection formula. The Formula Workshop appears.

4. Enter your selection formula in the **Group Selection Formula Editor**. The resulting formula must be Boolean; that is, it must return either a True or a False value.
5. Click **Check** to identify any errors in the formula.
6. Fix any syntax errors the Formula Checker identifies.
7. When the formula has the correct syntax, click Save and Close. When the program runs the report, it will include only those records or groups of records that you specified.

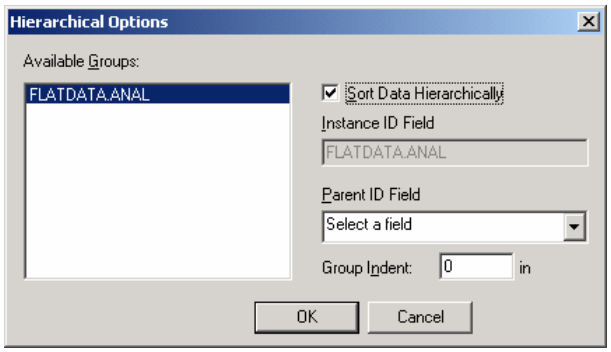
Grouping data hierarchically

When you group data hierarchically, you sort information based on the relationship between two fields.

To group data hierarchically

1. If you already have a group set up that you want to be the basis of your hierarchy, you can skip the steps for inserting the group and go to step 6.
2. Right click on the report and click **Insert** in the menu.

3. Click **Group.**
4. In the Insert Group dialog box, select the field that you want to use as the basis of your hierarchy (the child field). For example, if you want to see the hierarchical structure of a company's employees, select the employee field.
5. Select **in ascending order**. By default, the group header of the report displays the value of the field you are grouping on.
6. If you want to show a different value in the group header, click the **Options** tab and select the **Customize Group Name Field** check box. For example, if you grouped by the employee field, at each change of a group you will see the corresponding employee name. If you want to display a different value (employee ID instead of employee name), customize the group name field by choosing an alternate data field, or by creating a formula.
7. Click **OK**. The group you created is added to the report.
8. Right click on the report, and click **Report** in the menu.
9. Click **Hierarchical Grouping Options** on the next pull-down menu to access the Hierarchical Options dialog box.



10. Select the group you want to organize hierarchically in the **Available Groups** list. If you have created only one group on your report, then it will be selected automatically in the Available Groups list.
11. Select the **Sort Data Hierarchically** checkbox.
12. In the **Parent ID Field** list, select the field by which you want the Instance ID Field organized. For example, for an employee hierarchical report, you might select the data field listing the supervisor to whom the employee reports.

Note: The Instance ID Field and Parent ID Field must be of the same data type. For example, if the Instance ID Field holds string data, then the Parent ID Field must also hold string data.

13. In the **Group Indent** field, enter the amount you want to indent for each subgroup
14. Click **OK.**

The report data is now grouped hierarchically. If necessary, you can now calculate summary fields across your new hierarchical grouping. The top level of the hierarchy is determined by group instances that match the Instance ID and Parent ID. If a group instance is not connected to any Parent ID, it appears at the top of the hierarchy.

Editing groups

To edit a group:

1. Right click on the report and click **Report** in the pull-down menu.
2. Click **Change Group Expert** in the next pull-down menu.
3. Select the group you want to edit in the Change Group dialog box.
4. Click **Options**.
5. Edit the group as necessary in the Change Group Options dialog box.
6. Click **OK**. The report reflects the changes you have made to the group.

Summarizing Grouped Data

One of the primary purposes for breaking data into groups is to run calculations on each group of records instead of on all the records in the report. When the program summarizes data, it automatically sorts the data, breaks it into groups, and then summarizes the values in each group.

Depending on the data type of the field you plan to summarize, the program includes a number of summarizing options.

- Sum the values in each group
- Count all the values or only those values that are distinct from one another
- Determine the maximum, minimum, average, or nth largest value

Calculate up to two kinds of standard deviations and variances. You can also calculate summary fields across hierarchical groupings. To do so, select Sum across hierarchy in the Insert Subtotal, Insert Grand Total, or Insert Summary dialog box.

To summarize grouped data:

1. Right click on the report and click Insert in the pull-down menu.
2. Click **Summary** in the next pull-down menu to access the Insert Summary dialog box.
3. Select the desired field to summarize from the **Choose the field to summarize** list.
4. Select a summary operation from the **Calculate this summary** list.

5. Select a location in which to place the summaryfrom the **Summary location** list.

Tip: You can create a new group for you report by clicking the Insert Group button.

6. If you want to display your summaryvalue as a percentage of a total, select **Show as a percentage of** from the Options area, and then select a total field from the list.
7. If you want to summarize across a hierarchy, select **Summarize across hierarchy**.
8. Click **OK** when finished.

Percentages

Calculating a Percentage

You can calculate the percentage of one group within a broader grouping.

To calculate a percentage:

1. Right click on the report and click Insert in the pull-down menu.
2. Click **Summary** in the next pull-down menu. The Insert Summary dialog box appears.
3. Enter **sum** in the first drop-down list.
4. Select the field for which you want to calculate the sum.
5. Select how you want the data grouped.
6. Click the **Show as a percentage** of checkbox.
7. Select the group you want the percentage based on. You can choose to show a percentage of a group within another group, or show a percentage of the grand total.
8. Click **OK**.

Group Headers

Creating Group Headers

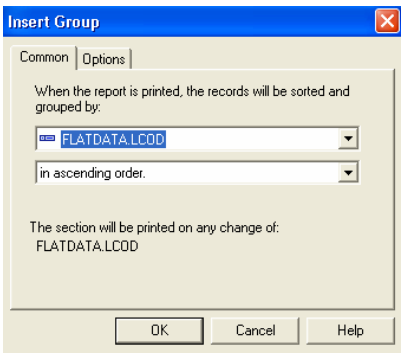
When you create a group, a subtotal, or a summary, the LABWORKS Report Designer program creates a Group Footer section that contains any subtotal or summary value and a Group Header section that contains the group name/header. Group Headers are useful, if you want your report data to be clear and easily understood. Even though the program creates a group header automatically, you may want to modify the header to meet your specifications.

To create a standard header:

1. Right click on the report and click **Insert**.
2. Click **Text Object** in the pull-down menu.
3. Place the object frame in the **Group Header** section.
4. Enter the text you want to use for the header.
5. Click outside the frame when finished. The same header will appear at the beginning of each group.

To create a header for custom groups:

1. Double click on the **Group Name Fields** folder.

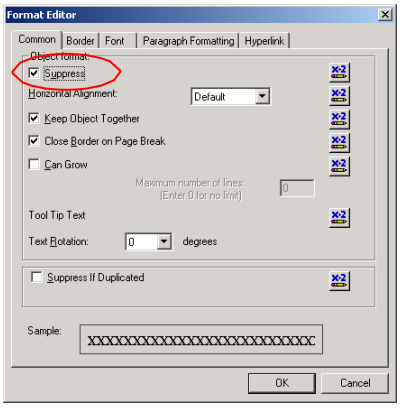


2. Click the **Group Name** field for the custom group and drag it into the **Group Header** section for that group. The program automatically applies each of the group names you assigned to the appropriate groups.
3. Make sure that when you assign the names to the groups using the Define Named Group dialog box, the names you assign are the names you want to appear as group headers.

Suppressing group headers

To suppress group headers:

1. Right click the group header and select **Format Field**.



2. Click the **Suppress** checkbox.

3. Click **OK**.

4. To show the group header again, clear the Suppress checkbox.

Running Totals

Running Totals

Running totals provide a way to create specialized summaries and continual incrementing totals. This chapter describes how to add a basic running total and a running total within a group to your report.

Running total fields are similar to summary fields but allow more control over how the total is calculated and when it is reset. Running total fields may be specifically used to perform the following totaling functions:

- Show values of a total accumulate as it is calculated record by record.
- Total a value independent of the report's grouping.
- Total a value conditionally.
- Total a value after a group selection formula has been applied.
- Total a value from the driving table in a one-to-many linking relationship.

How Running Totals Work

A running total field is created using the Running Total Expert. The Running Total Expert creates a running total field based on your specifications for how to select a field to summarize, the summary operation to use, a condition upon which to base the evaluation, and a condition upon which to reset the evaluation.

A running total field can be used on database fields and first-pass formulas but cannot be used on second-pass formulas or formulas that reference other second- pass formulas.

Placement of Running Total Fields

The Running Total field calculation is determined by the settings selected in the Running Total Expert. Where you place the running total affects the value that appears on the report. For example, if a Running Total field that evaluates every record and never resets, such as a grand total in the Report Header, only the value of the first record appears. Placing the same Running Total field in the Report Footer returns the desired value. The Running Total field is properly calculated in both cases, but it is displayed too early in the header.

Following is a list of report sections and records that the running total uses.

Section	Records
Report Header	Returns first record of the report
Page Header	Creates running total up to and including the first record of current page.
Group Header	Creates running total up to and including first record of current group.
Details	Creates running total for every record.
Group Footer	Creates a grand total for each group.
Page Footer	Creates running total up to and including the first record of the next page.
Report Footer	Creates a grand total that includes all records.

Creating Running Totals

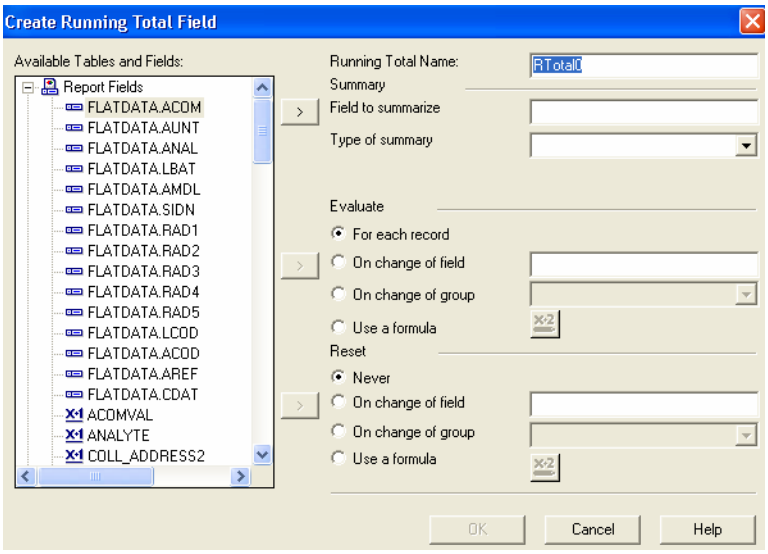
Creating running totals in a list

Running totals are totals that are generally displayed on a record-by-record basis. They total all records up to and including the current record. The most basic form of a running total is a single running total maintained throughout a list. Running total fields are prefixed by the # sign.

To create a running total in a list:

1. Open the report in the LABWORKS Report Designer to which you want to create a running total.

2. Right click **Running Total Fields** and click **New** in the pull-down menu. The Create Running Total Field dialog box appears.



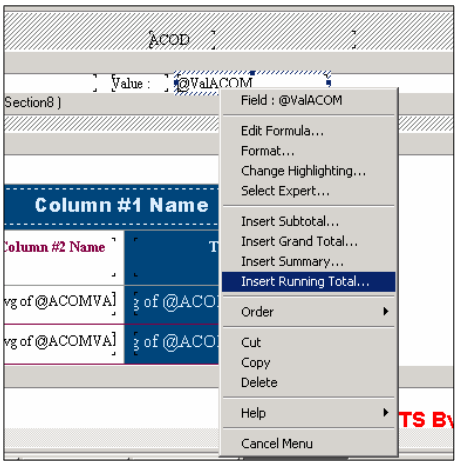
3. Enter a name for the field in the **Running Total Name** text field.
4. Click on the field you want to create the running total in the **Available Tables and Fields** list.
5. Click on the arrow button (>) next to **Summary** to select it as the **Field to summarize**.
6. Select **sum** from the **Type of summary** list.
7. Click **On change of field** in the **Evaluate** section of the dialog box.
8. Select the field you want as the **On change of field**. The running total will execute each time this field changes.
9. Click **Never** in the **Reset** section of the dialog box. This gives you a running total that never resets and continues running throughout the report.
10. Click **OK** to save the running total field.
11. Insert the running total field in the **Details** section of the report.

Creating running totals for a group

The running totals can also be used for tallying items in a group. The running total starts with the first item in the group and ends with the last. Then it starts all over again for the next group, then the next, and so on.

To create a running total for a group:

1. Right click on the **Group** on the report for which you want to create a running total.
   OR
   Right click **Running Total Fields** and click **New** in the pull-down menu.



2. Enter the name in the **Running Total Name** field.
3. Click on the field you want to create the running total in the **Available Tables and Fields** list.
4. Click on the arrow button next to **Summary** to select it as the **Field to summarize**.
5. Select **sum** from the **Type of summary** list.
6. Click **For each record** in the **Evaluate** section of the dialog box.
7. Click **On change of group** in the **Reset** section of the dialog box and accept the default group name.
8. Click **OK** to save the running total field.
9. Insert the running total field in the **Details** section of the report. If you want to view a grand total of each group, place the running total field in the Group Footer section of your report.

Creating conditional running totals

To create a conditional running total:

1. Right click on the report and click **Report** in the pull-down menu.
2. Click Sort Records in the next pull-down menu.
3. Sort the records based on the desired field you want to use to sort records.
4. Right click on **Running Total Fields** and click **New**.
    1. The Create Running Total Field dialog box appears.
5. Enter a name in the **Running Total Name** field.
6. Click on the field you want to create the running total in the **Available Tables and Fields** list.
7. Click on the arrow button next to **Summary** to select it as the **Field to summarize**.
8. Select **sum** from the **Type of summary** list.

9. Click **Use formula** in the **Evaluate** section of the dialog box and then click the **Formula** button. The Running Total Condition Formula Editor appears.
10. Enter the formula in the formula box for the data you want the program to generate the running total.
11. When the formula has the correct syntax, click **Save and Close**.
12. Click **Never** in the **Reset** section of the dialog box.
13. Click OK to save the running total field.
14. Place the running total field in the **Details** section of your report.
15. Create other running total fields, if necessary.

If you only want to see a grand total two running totals, create two running total fields and place them in the Report Footer section of your report.

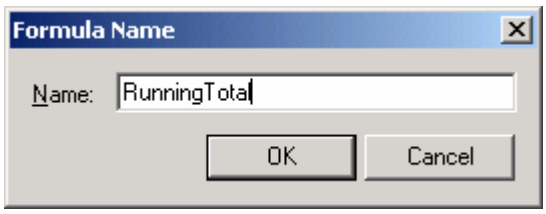
Creating running totals using a formula

When you create a running total manually, you need to create three formulas:

- Summary formula
- Reset formula to set a variable to zero
- Display formula to display the variable

To create running totals using a formula:

1. Right click on Formula Fields and click **New**.
2. Enter a name for the formula in the Formula Name popup window.



3. Click **OK**.
4. Enter the formula for the total in the Formula Editor.
   Example: WhilePrintingRecords;
   CurrencyVar Amount;
   Amount := Amount + {orders.ORDER AMOUNT}
   This formula prints the running total of the values in the Order Amount field.
5. Click **Save** and **Close**.

6. Place the formula in the **Details** section of your report just to the right of the field used in the formula, e.g., {orders.ORDER AMOUNT} field.

7. Right click on the report and click **Insert** in the pull-down menu; click **Group**.

8. Group the report on the field for obtaining running totals, e.g., {customer.CUSTOMER NAME}.

9. In the Formula Editor, create another formula.
   Example "AmountReset":
   WhilePrintingRecords; CurrencyVar
   Amount := 0;
   This formula says: Set the value in the Amount variable to 0.

10. Place this formula in the Group Header #1 section of your report. Because the Group Header #1 section appears once for every group, the formula @AmountReset will execute each time the group changes, and the Amount variable is reset to 0 each time a new group begins.

11. Select the formula (@AmountReset) on the report and use the Format Editor to suppress it so that it will not appear in the final printout.

12. In the Formula Editor, create "AmountDisplay": WhilePrintingRecords; CurrencyVar Amount formula. This formula simply displays the current value of the Amount variable at any time.

13. Place this formula in the Group Footer #1 section of your report. Because the Group Footer #1 section appears once for every group,@AmountDisplay will execute each time a group ends, and the value stored in the Amount variable will be printed each time the group changes. This formula prints the same value that @RunningTotal prints as the running total for the last record in each group. However, since it is printing it in the Group Footer section, it acts as a group subtotal, not as a running total.

Multiple Section Reports

About Sections

LABWORKS Report Designer provides five design areas to use when building your report:

- Report Header
- Page Header
- Details
- Report Footer
- Page Footer

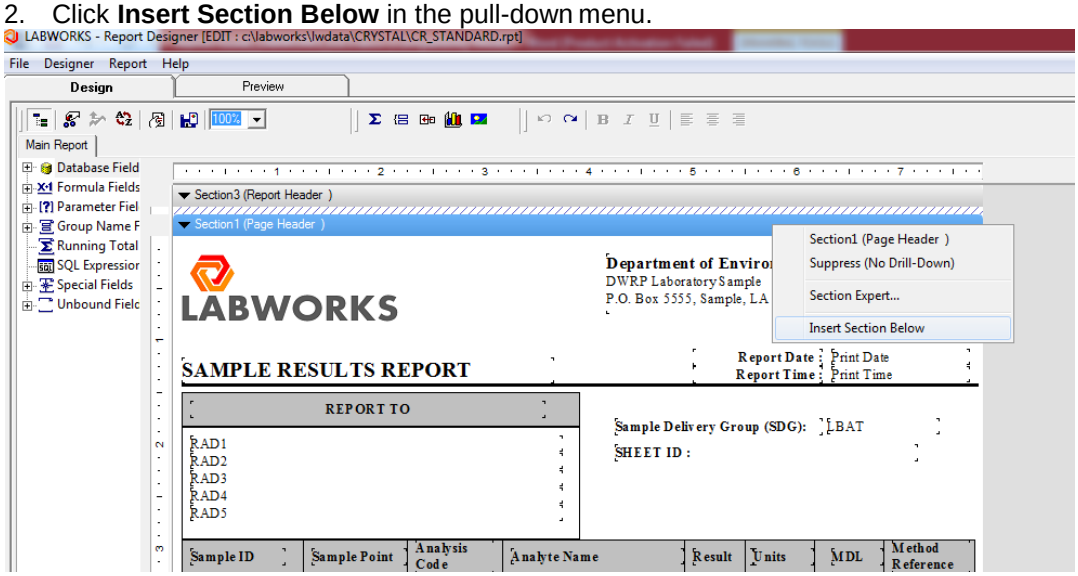
Each area contains a single section when you first create a new report. You cannot delete any of these original sections, but you can add to them. After adding sections, you can delete them, move them in relation to other similar sections, or merge related sections together.

Working with Sections

Inserting a section

To insert a section directly from the report:

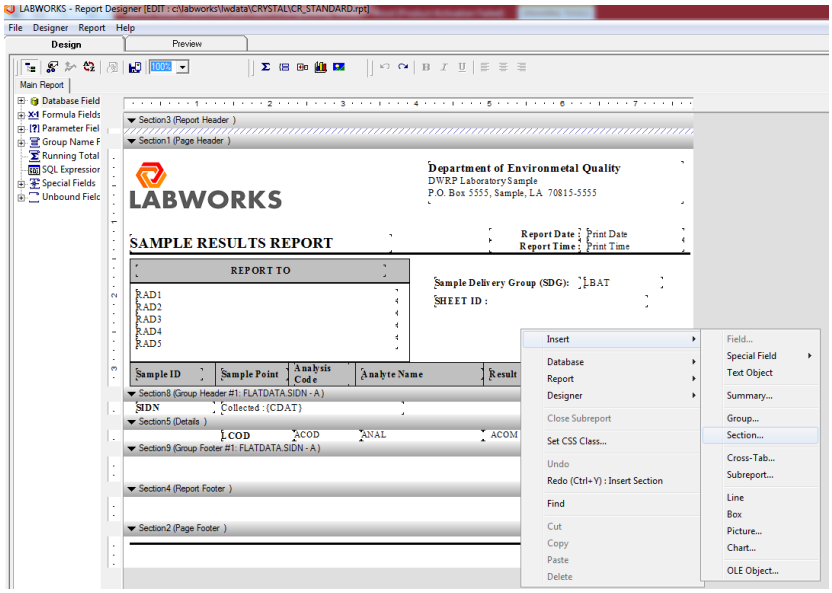
1. Right click on the report section where you want to insert a section.
2. Click **Insert Section Below** in the pull-down menu.



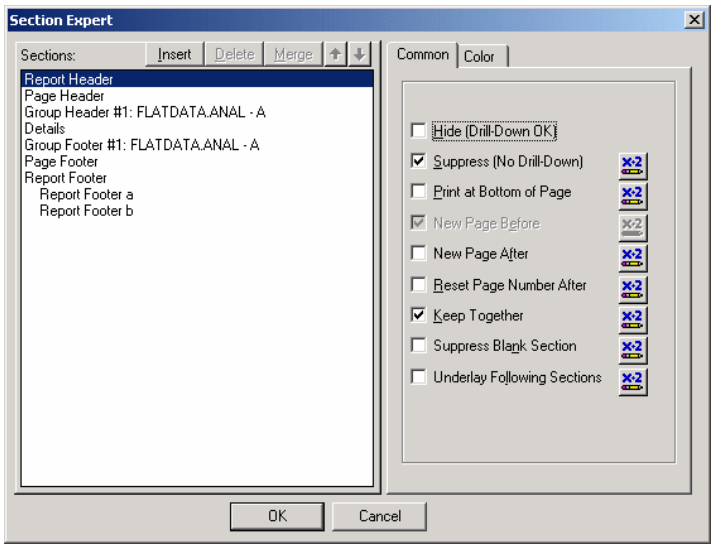
3. Notice that a new section is inserted immediately below the section.

To insert a section using the Section Expert:

1. Right click on the report and click **Insert** in the pull-down menu.



2. Click **Section** in the next pull-down menu.



3. Select the section you want to insert a section after.

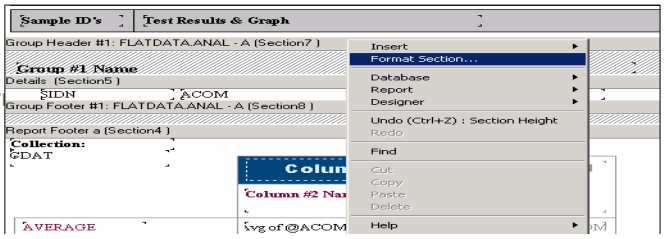
4. Click **Insert**.

5. Notice that the new section is inserted immediately below.

Deleting a Section

To delete a section:

1. Right click on the report section and click **Format Section** in the pull-down menu.



2. At the **Section Expert**, select the section you want to delete.
3. Click **Delete**. You cannot delete the originating sections.

Moving a section

To move a section:

1. Right click on the report section and click **Format Section** in the pull-down menu.
2. At the **Section Expert**, select the section you want to move.
3. Click the **Up** or **Down** arrow to move the section.
4. Click the arrow more than once to move it higher or lower. You can only move a section up or down within an area.

Merging two related sections

There may be times when you have placed objects in two sections where they print sequentially, but you want to put them all in a single section where they print simultaneously. To do this, you may merge the two sections and then rearrange the objects as needed in the new section.

To merge two related sections:

1. Right click on the report section and click **Format Section** in the pull-down menu.
2. At the **Section Expert**, select the section you want to move. The program enables only those options (free form, new page before, etc.) that apply to the selected section.
3. Move the sections so the two sections you want to merge follow each other in the list.
4. Click on the top section to select it.
5. Click **Merge**.
6. Notice that the lower section is merged with the top (selected) section to form one section.
7. Rearrange the objects as needed.

Resizing a section

To resize a section:

1. Move the pointer over the bottom boundary of the section you want to resize.

2. When the pointer becomes a Resizing cursor, drag the boundary to make the section larger or smaller. A section automatically expands vertically in two instances:

When you place an object and the object is vertically larger than the section you place it in.
When you expand an object vertically so it becomes larger than the section it is in.

NOTE: You cannot resize a section smaller than an object in the section.

To resize or add space between rows or sections:

1. Move the cursor over the lower section boundary line until it changes to a resizing cursor.

2. Drag the line until the row is the desired width.

3. Alternatively, right click in a section and click **Insert > Line** from the pull- down menus. Using the Insert function, the program resizes the section automatically and adds the amount of space necessary to hold a line of typical database fields.

Formatting Objects Conditionally

You may want to create a report that uses different formats. You can do that using multiple sections.

To format objects conditionally:

1. Click on the report section and click **Format Section** in the pull-down menu.

2. At the **Section Expert**, select the section you want to move.

3. Select the section in the Sections box.

4. Select the Suppress (No Drill-Down) checkbox.

5. Click the **Conditional Formula** button to the right of the **Suppress (No Drill-Down)** checkbox.

6. Use the **Format Formula Editor** to create a formula that specifies the conditions under which the section should be suppressed.

7. Repeat the steps for each additional section.

Eliminating Blank Lines When Fields Are Empty

If you create a report in which some field may be left empty, such as address fields, the field prints as a blank line. You can eliminate this blank line by using multiple sections or suppressing blank lines.

To eliminate blank lines by using multiple sections:

1. Use the **Section Expert** to create two new **Details** sections so that you have a total of three.

2. Place the **Address 2** field in the middle section and the other data in the sections above and below it as you want it to appear in the report.

3. Select the middle section in the **Section expert**.

4. Select the Suppress Blank Section checkbox. Now, when the report prints, if the Address 2 section is blank, the program will not print it, and you will not get unwanted blank lines in the report.

To suppress blank lines in embedded fields:

1. Open your report in the LABWORKS Report Designer.

2. Click the text object that causes blank lines for some records. To ensure that you have clicked a text object, look for the word Text in the status bar at the bottom left corner of the screen.

3. Right-click the text object and click **Format Text**.

4. Select the **Suppress Embedded Field Blank Lines** option in the Format Editor.

5. Click **OK**.

6. Click **File** and click **Print Preview** to display your report and confirm your changes.

Adding Blank Lines Conditionally

You may want to use multiple sections to print a blank line on your report under specific conditions. For example, you may want to insert a blank line after every fourth record in the report.

To add blank lines conditionally:

1. Right click on the report and click **Format Section** in the pull-down menu.

2. At the **Section Expert**, create two Details sections.

3. Place the report detail data in the top section.

4. Leave the second section empty.

5. Click on the second section to highlight it.

6. On the **Common** tab, select the **Suppress (No Drill-Down)** checkbox then click the conditional formatting button to its right.

7. Enter the following formula in the Format Formula Editor: Remainder (RecordNumber,4) <> 0.

   This formula divides each record number by 4. If the division produces a remainder, it suppresses the blank section. If the division produces no remainder (zero for every fourth record printed), the program prints the second section and inserts a blank line.

   To insert a blank line under different conditions, you can modify your formula appropriately.

Formatting

Formatting

Formatting is used to change the layout and design of a report including the appearance of text, objects, or entire report sections. This chapter describes methods you can use to draw attention to data; change the presentation of dates, numbers, and other values; hide unwanted sections; and perform a variety of other formatting tasks to give a report a professional appearance.

- Formatting includes:
- Dividing sections of a report.
- Calling attention to certain data.
- Changing the presentation of dates, numbers, Boolean values, currency values, and text strings.
- Hiding unwanted sections.
- Giving the report a professional appearance.

If you are designing reports that are distributed to different environments, consider how the report design format affects it purpose and function. This section describes some useful formatting concepts and tips.

Section Characteristics

A report consists of several sections, including the Report Header, Page Header, Group Header, Details, Group Footer, Page Footer, and Report Footer. Each report section is made up of a series of lines. When a text-based object is placed in a section, it is placed on a line so that the text is aligned to the baseline. The line's height is then adjusted by the printer driver so that it is high enough to accommodate the object. If you place another text-based object on the same line with a font size larger than that of the first object, the line's height extends to accommodate the second object. A line's height is determined by the text-based object with the largest font size on the line.

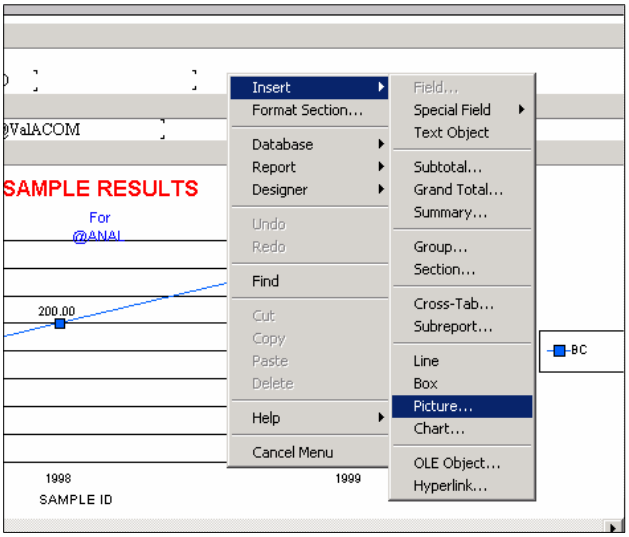
As you add text-based objects to a report, in either the same section or other sections, the line height adjusts to accommodate the various fonts. Since this vertical spacing is set up by the printer driver, it is difficult to create reports designed for pre-printed forms when they are printed in various environments.

When designing reports, you should do the following:

- Always print a test page.
- Keep all font sizes the same.
- Be sure to print pre-printed forms on the same machine.

Inserting a Picture

1. Right click on the report and click Insert in the pull-down menu.



2. Click **Picture** in the next pull-down menu.

3. Select the picture file (jpg, bmp, etc.), then place it in the Page Header or other section in the desired location.

4. Place the picture to avoid underlying the text.

5. Click **File** and select **Print Preview** to preview how the picture appears in the report.

To make the picture underlay the following section:

1. Right click on the report and click **Format Section**. The Section Expert dialog box appears.

2. In the **Sections** list, click **Page Header** (or the section where you placed the picture), then select the **Underlay Following Sections** checkbox.

3. Click **OK** to preview the report again. The picture should now print in both the Group Header and Details sections, next to (instead of above) the text in the body of the report. Using the technique of placing a picture to the right of the body of the report, you can set up a chart or picture to print beside the details pertaining to that chart or picture.

4. Resize the object vertically to make it two or three times larger.

5. Preview the report again. The bitmap file now underlays more sections. The area in which the picture underlays depends on: size of the picture; section in which the picture was originally placed; and position of the picture in the section. You can create a variety of visual effects using the underlay feature by modifying size and placement of an object.

Placing Text-Based Objects

When a text-based object is placed on a report, the object is represented by an object frame. The height of the object frame is based on the height of the font. The width is determined differently, depending on the object.

For database fields that are not memo fields, the width is initially determined by the width of the field as defined in the database, and by the average character width as provided by the selected font and font size. For example, you have a database field that is defined as a text field with a length of 35 characters. When you place this field on your report, the width of the boundary is 35 times the average character width of the font and font size that the database field is formatted to. Remember that this is the initial default boundary width. The width can always be resized to increase or decrease as you see fit.

For text-based objects, the default width is approximately 17 average character widths wide. Objects are different from database fields in that their width automatically expands as you enter text and/or database fields into the object. As with all other text-based objects, the user can resize the width. For different number fields, the default widths are all different. As with all other text-based objects, the user can resize the width.

Preventing truncation of text inside an object

A problem may arise if the text inside the object prints right to the edge of the object frame. Even though the report may look fine on the machine it was designed on, when the report is printed using another printer driver that measures the font wider, the length of the text grows, but the object frame remains fixed. The resulting text is cut-off or truncated.

To prevent truncation of text inside an object:

1. Right-click the text object you want to format.
2. Click **Format Text** to open the Format Editor dialog box.
3. Select the **Can Grow** checkbox on the **Common** tab.
4. Click **OK** to save your changes.

The object is then formatted to print on multiple lines. If the text prints wider than the object, the text wraps onto additional lines.

Preventing breaks in non-spacing text inside an object

For text strings that do not contain spaces, such as single words, the text string is broken at the edge of the object frame before the line starts to wrap.

To prevent breaks in non-spacing text inside an object:

1. Select the object you want to format.
2. Expand the object frame to make it wider than the widest block of text inside the frame. There are many times when the actual text in a database field is far less than the maximum amount the field can contain.
3. Reduce the width of the field but include enough space to account for growth.

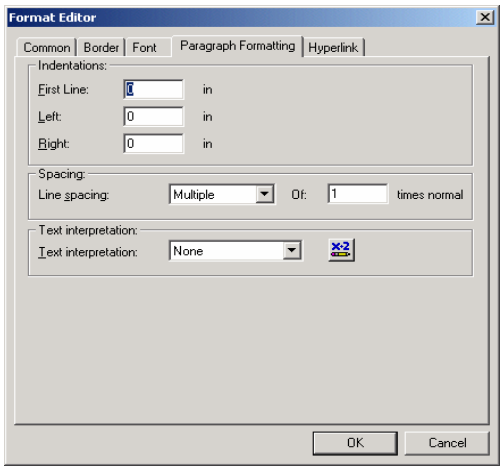
4. When sizing an object, consider its placement with regard to other objects in the section.
5. Avoid designing reports where the space between each object is very tight.
6. Leave room for growth by expanding the width of the object by approximately 5 per cent.
7. If this is not possible, consider reducing the size of the font.

Indenting Lines

You can control line indentation for memo fields, string fields, and text-based objects. For objects, you have the option of indenting lines for a particular paragraph by positioning the cursor at the start of that paragraph. Or, if you select an object in its entirety, you can apply the same indenting specifications to all the paragraphs within that object.

To indent lines:

1. Right-click the field or object you want to format.
2. Click **Format** in the pull-down menu.



3. Click **Paragraph Formatting**.
4. In the **Indentations** area, indent the first line of the paragraph; indent every paragraph line from the left margin; or indent every paragraph line from the right margin. Only indentation values within the range of the field or object width are accepted.
5. Click **OK** to save your changes.

Selecting Multiple Objects

You can select multiple objects, including text, field, chart, map, bitmap, Cross-Tab and OLE objects, to format them together. Once selected, you can move, align, size, cut, and copy and paste them as a group. You can also change their font, color, and paragraph style. Objects are moved, aligned, and sized based on a main object, which is usually the last object you select. You can change the main object to another by right clicking the desired object.

When moving multiple objects, if the new location does not accommodate all of the selected objects

but can accommodate the main object, then the main object and those objects that can be accommodated will be moved. The other objects will remain in their original position.

To select multiple objects:

1. Click one object, press the **Ctrl key**, and click the other objects you want to select.
2. Right click the main object.
3. Select the appropriate formatting option from the pull-down menu.

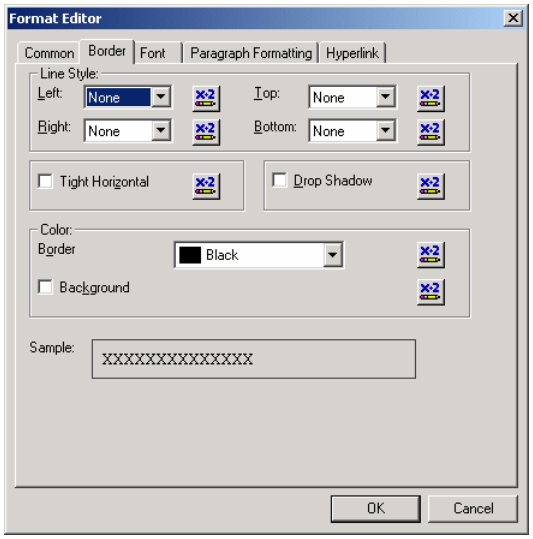
Absolute Formatting

Absolute formatting is formatting that applies under any condition.

Adding borders, color, and shading to a field

To add borders, color, and shading to a field:

1. Right-click the field you want to format and click **Format** in the pull-down menu.



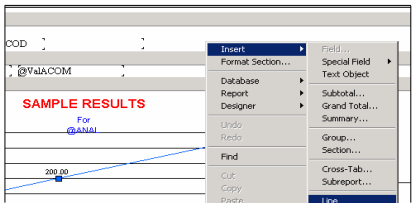
2. Click the **Border** tab of the Format Editor.
3. Select the line style, color, and background color of the field.
4. Click **OK** to save your changes.

Adding and editing lines

You may add lines to a report to emphasize important data and create professional- looking reports.

To add lines to a report:

1. Right click on the report and click **Insert**.



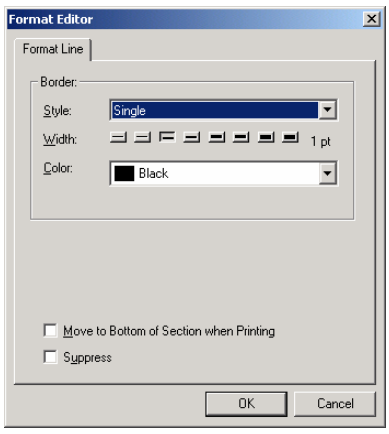
2. Click **Line** in the pull-down menu.

3. Use the pencil cursor to draw the line where desired.

To edit lines on a report:

1. Right click the line you want to format.

2. Click **Format Line**. The Format Editor dialog box appears.



3. Make the desired changes to the line on the **Format Line** tab.

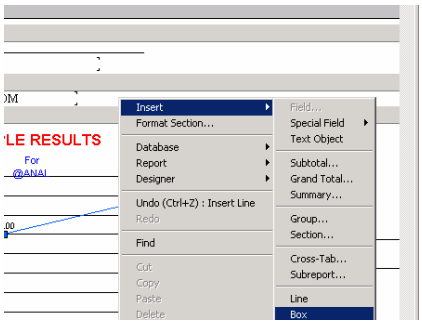
4. Click **OK** to save your changes.

Adding and editing boxes

You may add boxes to a report to emphasize important data and create professional-looking reports.

To add boxes to a report:

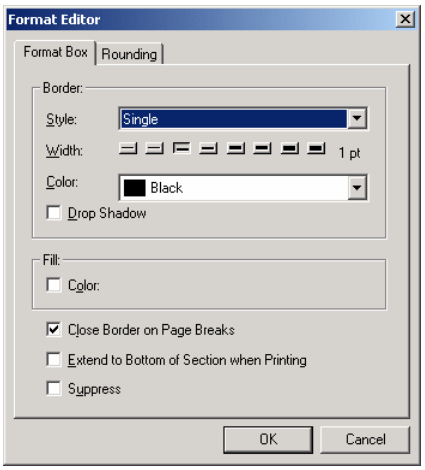
1. Right click on the report and click **Insert** in the pull-down menu.



2. Click **Box** in the next pull-down menu.
3. Use the pencil cursor to draw the box where desired.

To edit boxes on a report:

1. Right click the box you want to format.
2. Click Format Box. The Format Editor dialog box appears



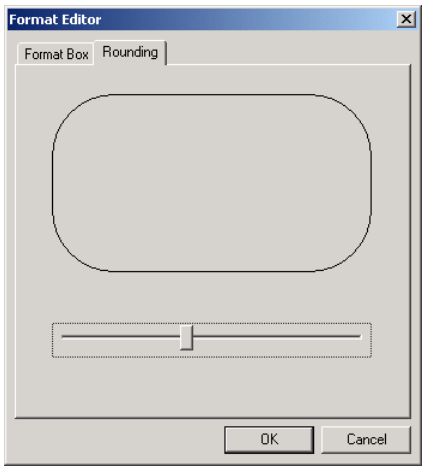
3. Make the desired changes to the box on the **Format Box** tab.
4. Click **OK** to save your changes.

Adding shapes to a report

When designing report format, you can insert a variety of shapes such as circles, eclipses, and boxes with rounded corners, as part of your report. This is especially useful for formatting reports in languages that require these shapes.

To add shapes to a report:

1. Add a box to your report as described in previous section.
2. Right click on the box.
3. Click **Format Box**.
4. In the Format Editor that appears, click the **Rounding** tab.



5. Move the slider to the right to increase the curvature of the box corners. The box that you started with gradually changes to an ellipse or circle, depending on how far you move the slider to the right.

6. Once the appropriate shape is created, click **OK** to save your changes.

Formatting numerical and currency values

LABWORKS Report Designer allows you to decide on how to display the currency symbol, negative values, and zero values in reports.

To format numerical and currency values:

1. Right click the currency field or number field you want to format.

2. Click **Format**. The Format Editor dialog box appears with the Number tab open.

3. In the **Style** area, select how you want the system number format to appear for either positive or negative values.

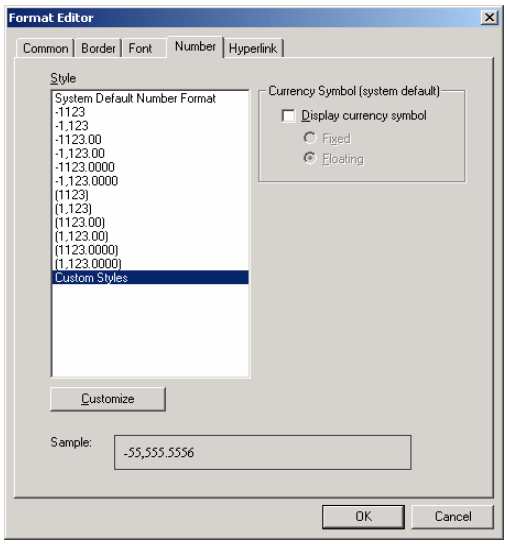
4. In the **Currency Symbol** area, specify how you want the currency symbol to appear with the values on your report.

5. Click **OK** to save your changes.

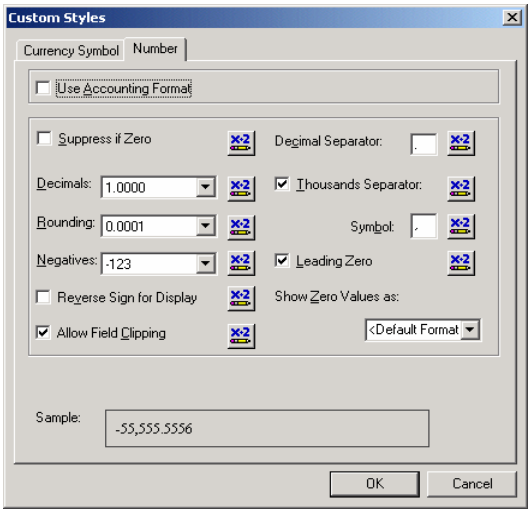
To customize numerical or currency formatting:

1. Right click the currency field or number field you want to format.

2. Click **Format Field**. The Format Editor dialog box appears with the Number tab open.



3. Click **Custom Style** in the **Style** list and then click the **Customize** button. The Custom Style dialog box appears with the Number tab open.



4. Click on the **Use Accounting Format** checkbox.

When you make this selection, the following settings appear:

Negatives – shows how the negative values appear on your report as determined by the Windows locale settings; negative values are represented by either a minus sign or brackets.

Show Zero Values – displays the dash symbol to represent zero values on your report.

Currency symbol on the Currency Symbol tab – displays the currency symbol positioned on the left side of the currency and numeric values.

NOTE: You must exit and restart the designer program to implement any changes you make to the Windows locale settings.

5. Make changes and selections to meet your specifications.

6. Click **OK** to save your changes.

7. Click **OK** again to return to your report.

Charting

Charting

LABWORKS Report Designer provides the ability to insert a variety of chart layouts and types in your report. This section demonstrates how to create charts and how to use them in reports to make report data more meaningful and easier to understand.

Chart Layouts

The Chart Expert provides four layouts that correspond to certain sets of data.

Advanced – used for multiple chart values or when there are no group or summary fields in the report; functions include:

- Supports one or two condition fields.
- Ability to create a 2-D or 3-D chart.
- Values can be grouped in ascending, descending, or specified order, as well as by Top N or Sort totals.
- Values can be plotted for each record.
- Values can be plotted as a grand total for all records.
- Charts can be based on formula and Running Total fields.

Group – used for simplified layout that shows a summary on change of field for topics; to create a chart using the Group layout, you must have at least one group and at least one summary field for that group.

Cross-Tab – used to chart on a Cross-Tab object; uses the fields in the cross-tab for its condition and summary fields.

OLAP – used to chart on an OLAP grid; uses the fields in the OLAP grid for its condition and summary fields.

Chart Placement in the Report

Where you place a chart in the report determines which data is displayed and where it is printed. For example, if you place it in a Group Header or Group Footer section, it displays group specific data. If you place a chart in the Report Header section, the chart includes data for the entire report.

Creating an Advanced Layout Chart

The Advanced layout allows you to create a chart based on specific values and are often based on a summary field in your report. Using this layout, you can create a chart without the need for a summary field by using values that appear in the Details section of your report.

To create a chart based on the Advanced layout, you must specify two things:

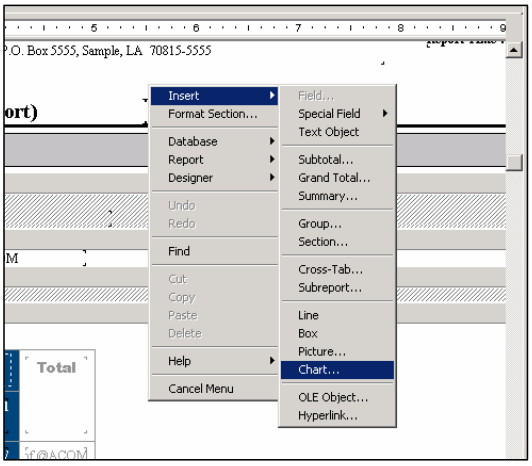
Conditions – can be two; condition is used to indicate when to plot the point; option of plotting a point for each record or plotting one point for all records.

Values – can be multiple values; indicates what information is plotted as the points on your chart.

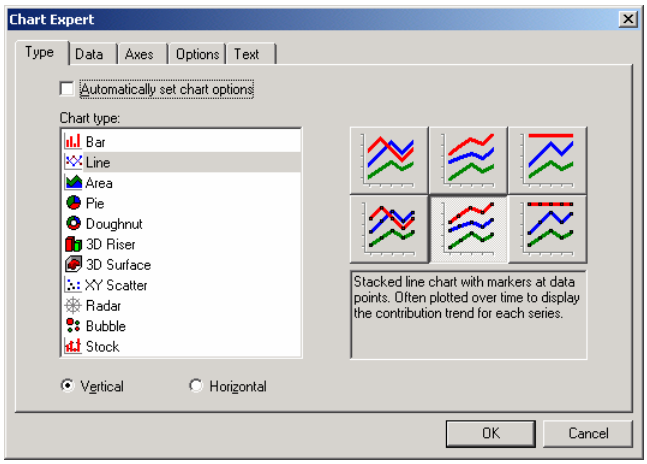
Creating charts based on details or formula field

To set up chart type specifications:

1. Right click on the report and click Insert in the pull-down menu.



2. Click **Chart**. The Chart Expert dialog box appears.



3. Select a chart type from the **Chart type** list on the **Type** tab.

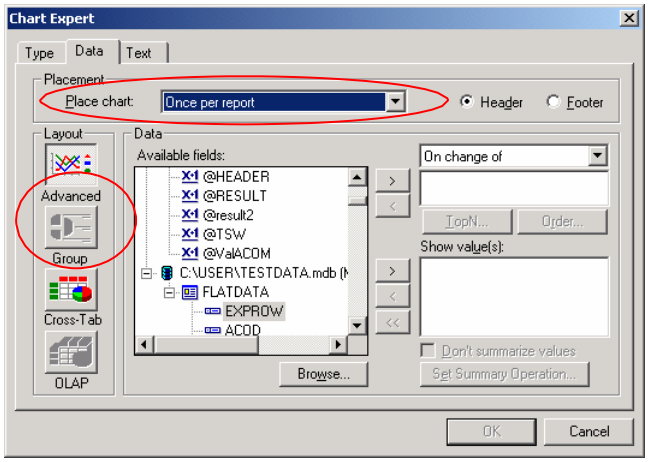
4. Click the chart subtype image on the right that best depicts how you want your data displayed. When you click on an illustration, text below provides useful information about the chart type.

5. Select **Automatically set chart options** if you want the Chart Expert to use the default options on the Axes and Options tabs.

6. Leave **Automatically set chart options** unchecked if you want to manually select options on the Axes and Options tabs. If you leave this option unchecked, the Axes and Options tab appear for setting up those options.

To set up data specifications:

1. Click the **Data** tab.

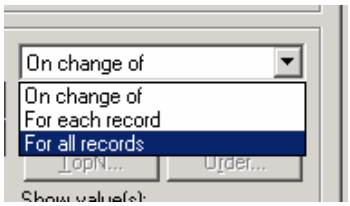


2. Click **Advanced** as the **Layout** if it is not already selected.

3. Click on the down arrow at the end of the **Place chart** field to specify how often you want the chart to appear on the report.

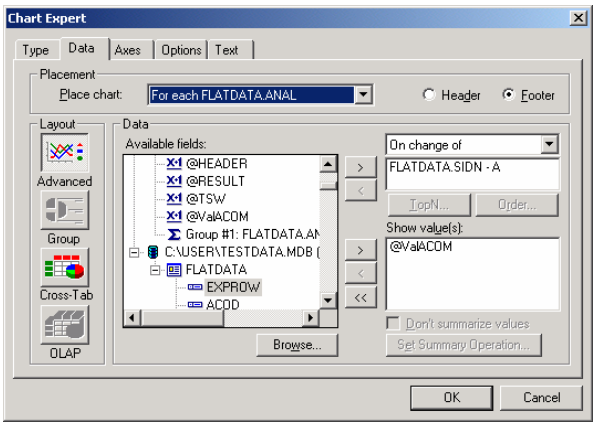
4. Click **Header** or **Footer** to specify where to place the chart.

5. Select the data condition by clicking on the down arrow and select your preference from **On change of, For each record,** or **For all records**.



6. In the **Data** area, click on the data field then click the arrow button to place it in the box to the right to specify the database fields you want to use as conditions. The arrow buttons on the Chart Expert dialog box allow you to move fields from one list to the other. Single arrows move only the selected field; double arrows move all fields at the same time.

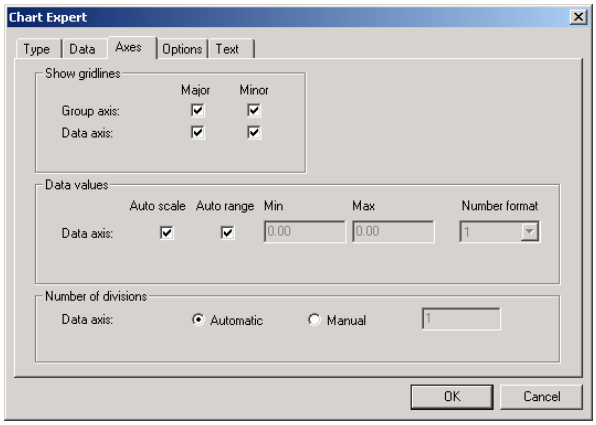
7. Add the database fields you want to use as values to the **Show value(s)** list.



8. If you do not want LABWORKS Report Designer to automatically summarize the chart values, select the Don't summarize values checkbox.

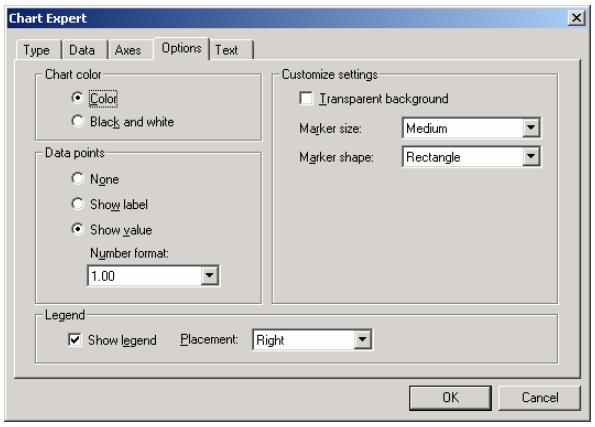
To set up axes and options specifications:

1. Click the **Axes** tab.



2. Click on the desired options for gridlines, data values, and number of divisions for data axis. After previewing your completed chart, edit your selections as needed to obtain the desired results.

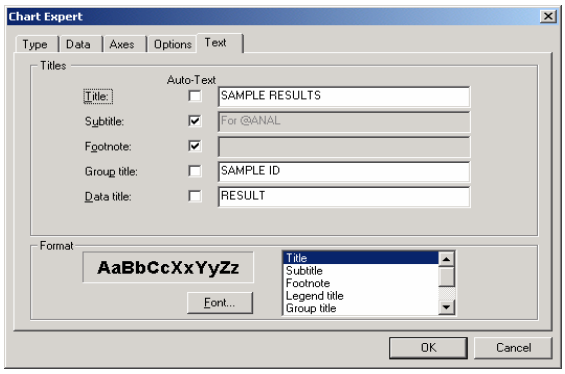
3. Click the **Options** tab.



4. Select whether you want the chart to print in color or black and white in the **Chart color** area.
5. Select how you want data points to appear in the **Data points** area.
6. Specify if you want a legend and its placement in the **Legend** area.
7. Specify marker size and shape and if you want a transparent background in the **Customize settings** area.

To set up text specifications:

1. Click the **Text** tab.

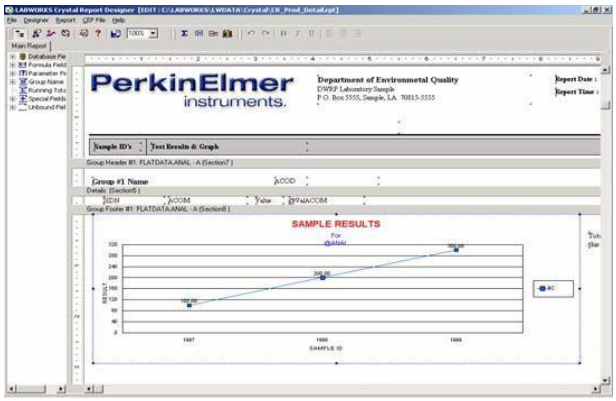


2. Accept the default title information or add new titles to your chart.
3. Click the **Font** button to change the font for the selected text to the right, if desired.
4. Click **OK**.

To refresh and preview completed chart:

1. Right click on the report and click **Database** then **Verify Database** to refresh the data.
2. Click **File** and click **Print Preview** to preview the finished chart.
3. Move and resize the chart so that it fits properly within the report, if needed.

EXAMPLE: LABWORKS Report Designer – Advanced Layout chart



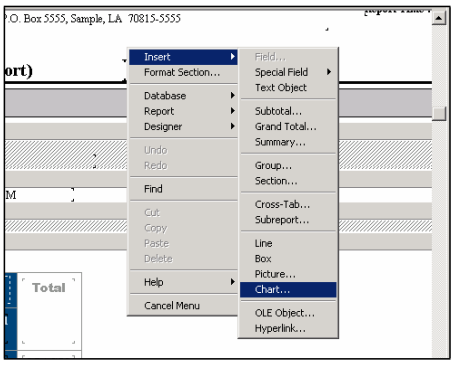
Creating a Group Layout Chart

Many of the charts you create are based on summary or subtotals within your report. You must have a summary or subtotal inserted into your report to create a group layout chart.

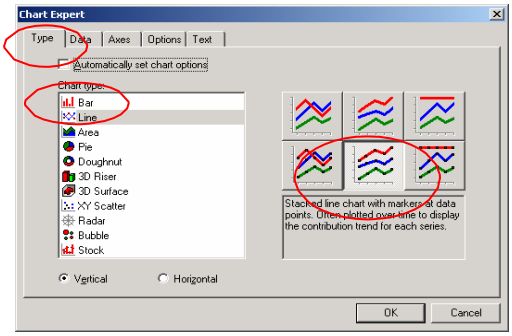
Creating a chart based on a summary or subtotal field

To set up chart type specifications:

1. Right click on the report and click Insert in the pull-down menu.



2. Click **Chart**. The Chart Expert dialog box appears.



3. Select a chart type from the **Chart type** list on the **Type** tab.

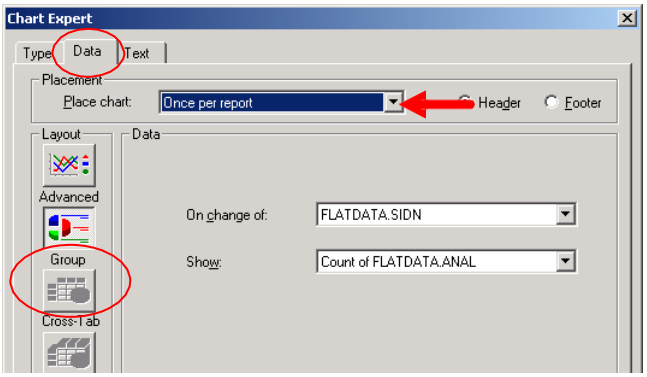
4. Click the chart subtype image on the right that best depicts how you want your data displayed. When you click on an illustration, text below provides useful information about the chart type.

5. Select **Automatically set chart options** if you want the Chart Expert to use the default options on the Axes and Options tabs.

6. Leave **Automatically set chart options** unchecked if you want to manually select options on the Axes and Options tabs. If you leave this option unchecked, the Axes and Options tabs appear for setting up those options.

To set up data specifications:

1. Click the **Data** tab.



2. Click **Group** in the **Layout** area if it is not already selected. When summarizing or subtotaling a field, the data is automatically grouped.

3. Click on the down arrow at the end of the **Place chart** field to specify how often you want the chart to appear on the report.

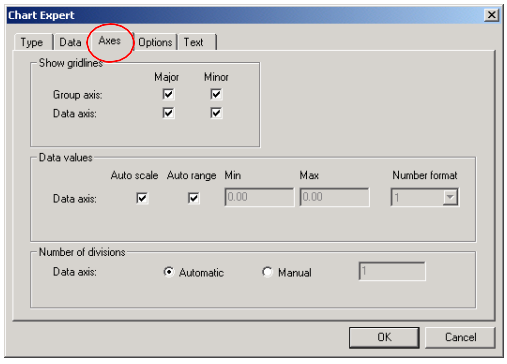
4. Click Header or Footer to specify in which section to place the chart. The Place chart list includes all groups in your report that have summary fields except for the innermost group.

5. Click the group field you want to base your chart on in the **On change of** list in the **Data** area.

6. Click the summary field you want to display on your chart in the **Show** list.

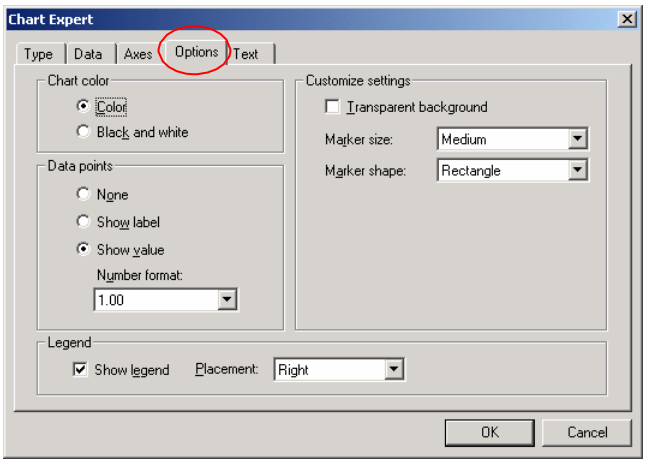
To set up axes and options specifications:

1. Click the **Axes** tab.



2. Click on the desired options for gridlines, data values, and number of divisions for data axis. After previewing your completed chart, edit your selections as needed to obtain the desired results.

3. Click the **Options** tab.



4. Select whether you want the chart to print in color or black and white in the **Chart color** area.

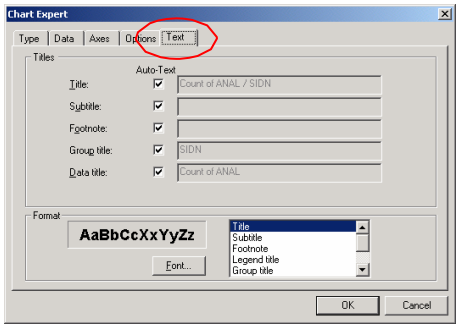
5. Select how you want data points to appear in the **Data points** area.

6. Specify if you want a legend and its placement in the **Legend** area.

7. Specify marker size and shape and if you want a transparent background in the **Customize settings** area.

To set up text specifications:

1. Click the **Text** tab.



2. Accept the default title information or add new titles to your chart.

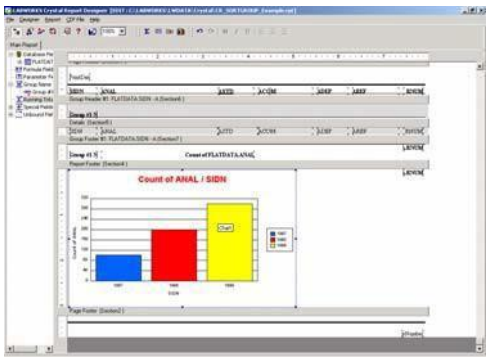
3. Click the **Font** button to change the font for the selected text to the right, if desired.

4. Click **OK**.

To refresh and preview completed chart:

1. Right click on the report and click **Database** then **Verify Database** to refresh the data.
2. Click **File** and click **Print Preview** to preview the finished chart.
3. Move and resize the chart so that it fits properly within the report, if needed.

EXAMPLE: LABWORKS Report Designer – Group chart

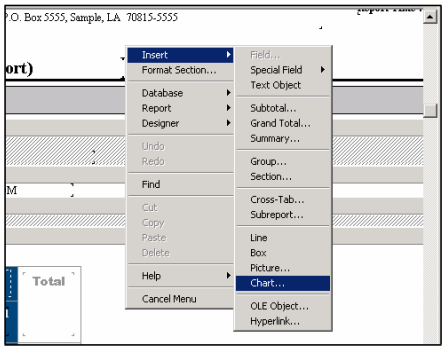


Creating a Cross-Tab Layout Chart

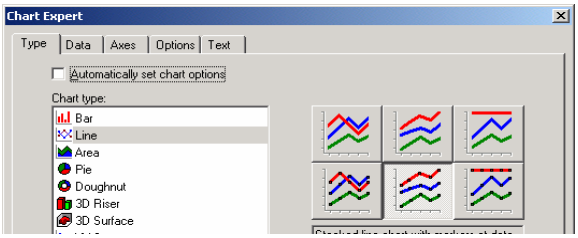
LABWORKS Report Designer allows you to include a chart based on summary values in your Cross-Tab report. You must first have a Cross-Tab in your report to create a Cross-Tab chart.

To set up chart type specifications:

1. Select the Cross-Tab on which you want to chart.
2. Right click and click **Insert** in the pull-down menu.



3. Click **Chart**. The Chart Expert dialog box appears.



4. Select a chart type from the **Chart type** list on the **Type** tab.

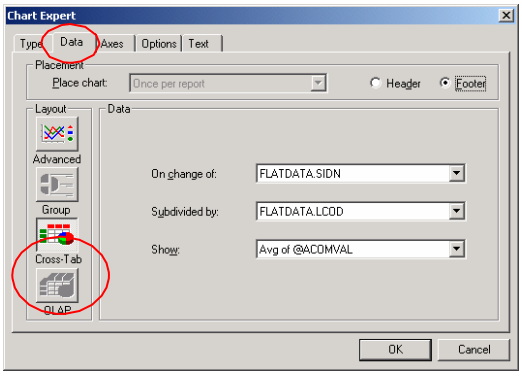
5. Click the chart subtype image on the right that best depicts how you want your data displayed. When you click on an illustration, text below provides useful information about the chart type.

6. Select **Automatically set chart options** if you want the Chart Expert to use the default options on the Axes and Options tabs.

7. Leave **Automatically set chart options** unchecked if you want to manually select options on the Axes and Options tabs. If you leave this option unchecked, the Axes and Options tabs appear for setting up those options.

To set up data specifications:

1. Click the **Data** tab.



2. Click **Cross-Tab** in the **Layout** area if it is not already selected. When summarizing or subtotaling a field, the data is automatically grouped.

3. Click Header or Footer to specify in which section to place the chart. How often your chart appears on the report depends on where the Cross-Tab summary field has been placed.

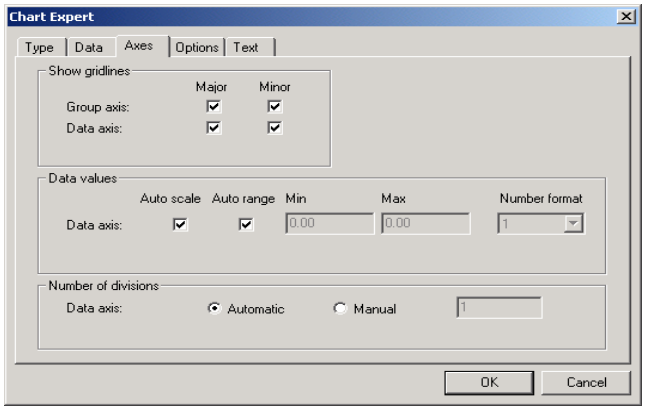
4. Click the group field you want to base your chart on in the **On change of** list in the **Data** area.

5. Click a secondary row or column you want to base your chart on in the **Subdivided by** list, if necessary.

6. Click the summary field you want to display on your chart in the **Show** list.

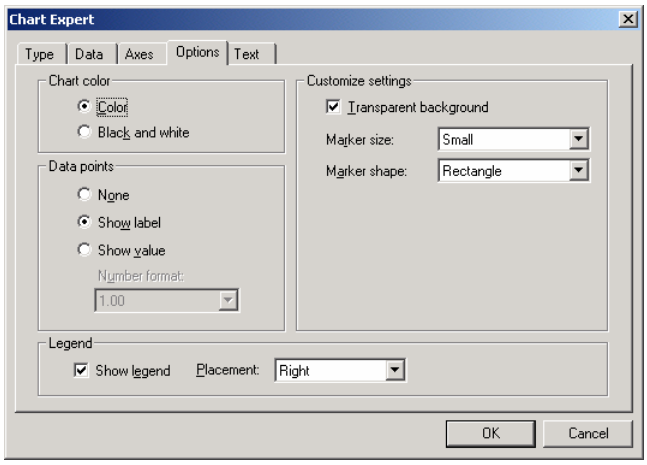
To set up axes and options specifications:

1. Click the **Axes** tab.



2. Click on the desired options for gridlines, data values, and number of divisions for data axis. After previewing your completed chart, edit your selections as needed to obtain the desired results.

3. Click the **Options** tab.



4. Select whether you want the chart to print in color or black and white in the **Chart color** area.

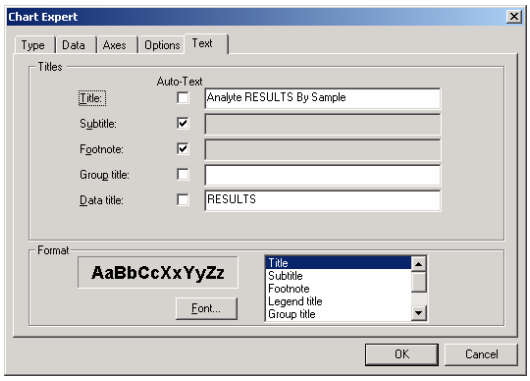
5. Select how you want data points to appear in the **Data points** area.

6. Specify if you want a legend and its placement in the **Legend** area.

7. Specify marker size and shape and if you want a transparent background in the **Customize settings** area.

To set up text specifications:

1. Click the **Text** tab.

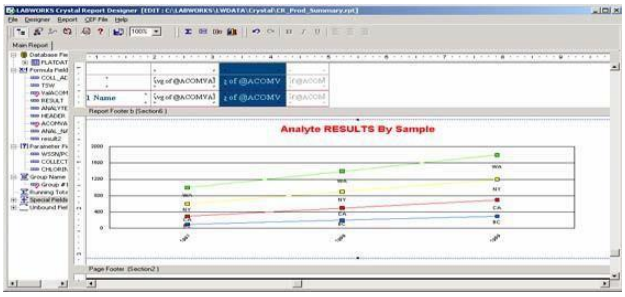


2. Accept the default title information or add new titles to your chart.
3. Click the **Font** button to change the font for the selected text to the right, if desired.
4. Click **OK**.

To refresh and preview completed chart:

1. Right click on the report and click **Database** then **Verify Database** to refresh the data.
2. Click **File** and click **Print Preview** to preview the finished chart.
3. Move and resize the chart so that it fits properly within the report, if needed.

EXAMPLE: LABWORKS Report Designer – Cross-Tab chart

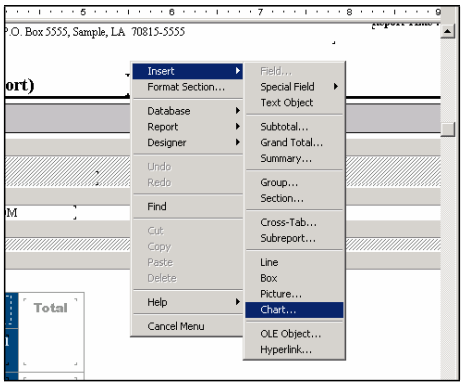


Creating an OLAP Layout Chart

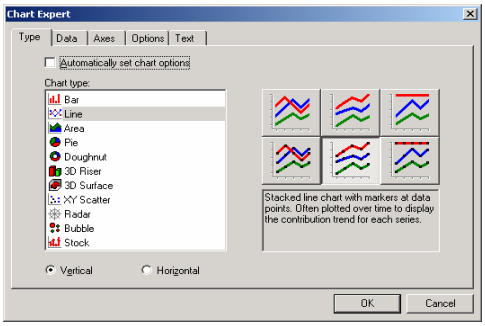
The OLAP layout lets you chart on an OLAP grid. You must have an OLAP grid in your report in order to create an OLAP chart.

To set up chart type specifications:

1. Select the OLAP grid on which you want to chart.
2. Right click and click **Insert** in the pull-down menu.



3. Click **Chart** to access the Chart Expert dialog box.



4. Select a chart type from the **Chart type** list on the **Type** tab.

5. Click the chart subtype image on the right that best depicts how you want your data displayed. When you click on an illustration, text below provides useful information about the chart type.

6. Select **Automatically set chart options** if you want the Chart Expert to use the default options on the Axes and Options tabs.

7. Leave **Automatically set chart options** unchecked if you want to manually select options on the Axes and Options tabs. If you leave this option unchecked, the Axes and Options tab appears for setting up those options.

To set up data specifications:

1. Click the **Data** tab.

2. Click OLAP in the **Layout** area if it is not already selected.

3. Click Header or Footer to specify in which section to place the chart. How often your chart appears in the report depends on the report section that the OLAP grid has been placed.

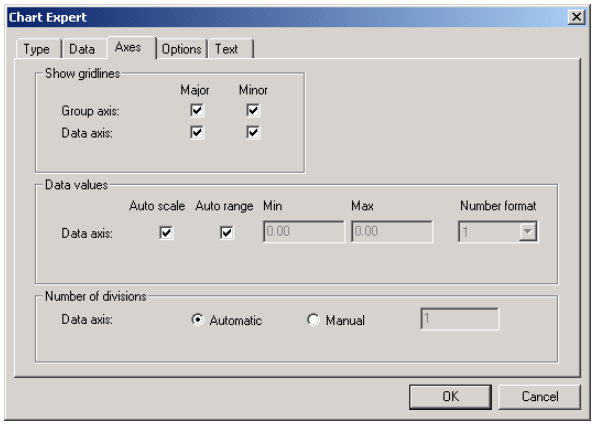
4. Click the field you want to base your chart on in the **On change of** list in the **Data** area.

5. Click a secondary row or column you want to base your chart on in the **Subdivided by** list, if necessary. Make sure the selected chart type supports a secondary charting field.

6. Click the summary field you want to display on your chart in the Show list.

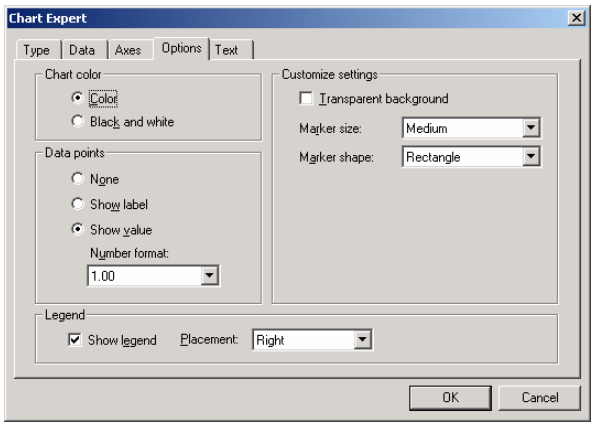
To set up axes and options specifications:

1. Click the **Axes** tab.



2. Click on the desired options for gridlines, data values, and number of divisions for data axis. After previewing your completed chart, edit your selections as needed to obtain the desired results.

3. Click the **Options** tab.



4. Select whether you want the chart to print in color or black and white in the **Chart color** area.

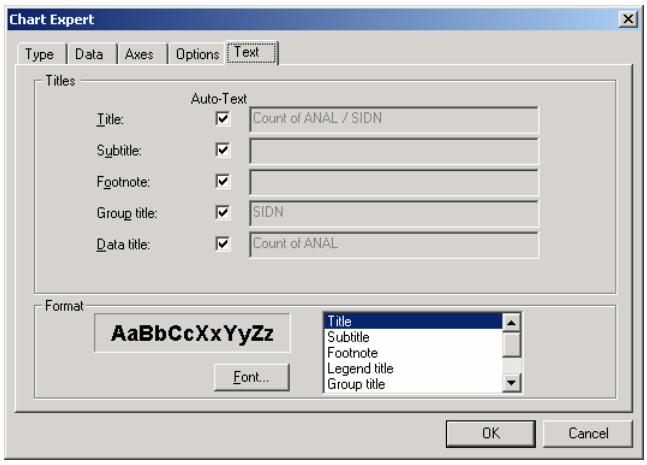
5. Select how you want data points to appear in the **Data points** area.

6. Specify if you want a legend and its placement in the **Legend** area.

7. Specify marker size and shape and if you want a transparent background in the **Customize settings** area.

To set up text specifications:

1. Click the **Text** tab.



2. Accept the default title information or add new titles to your chart.
3. Click the **Font** button to change the font for the selected text to the right, if desired.
4. Click **OK**.

To refresh and preview completed chart:

1. Right click on the report and click **Database** then **Verify Database** to refresh the data.
2. Click **File** and click **Print Preview** to preview the finished chart. Move and resize the chart so that it fits properly within the report, if needed.

Working with Charts

Editing charts using the chart expert

To edit a chart using the Chart Expert:

1. Right click the chart and click **Chart Expert**.
2. Make the desired changes in the Chart Expert dialog box.
3. Click **OK**.

Changing chart size and position

To change chart size and position:

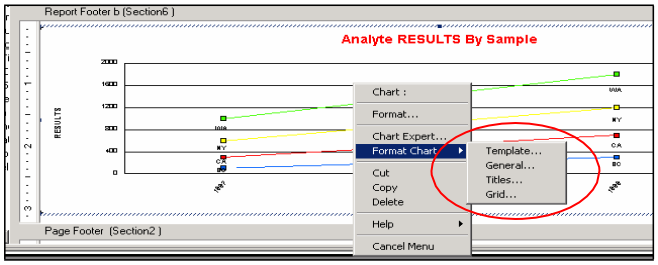
1. Click on the chart to drag and move it.
2. Click on the resizing handles to resize the chart.

Formatting a chart

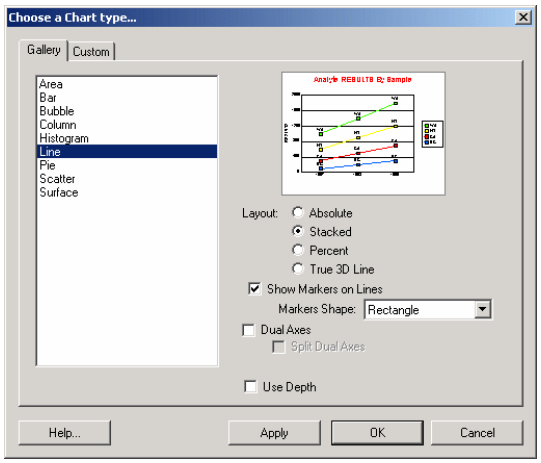
To change the chart format :

1. Right click on the chart.
2. Click **Format Chart** in the pull-down menu. The Format Chart has four options for customizing and enhancing your chart:

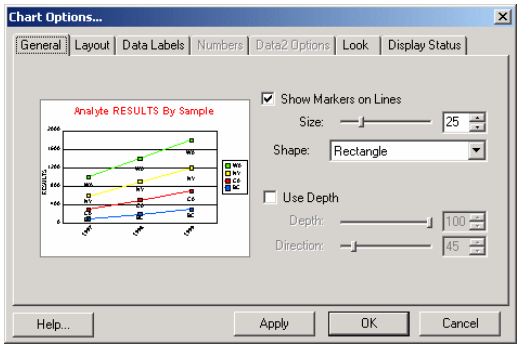
   Template, General, Titles, and **Grid.**



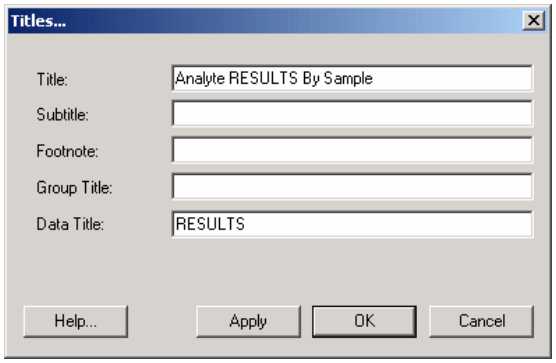
3. Click on **Template** to change and customize the chart type.



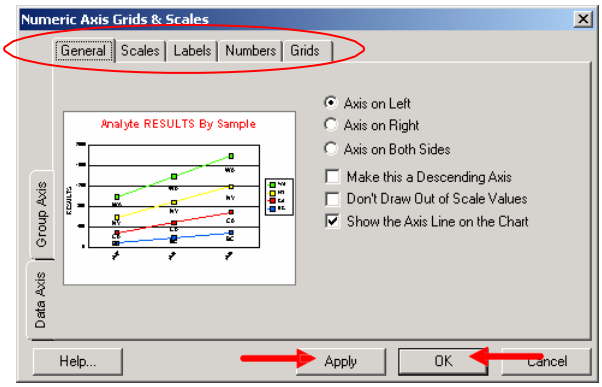
4. Make changes to the chart type and click **Apply** to see the immediate results on the designer screen.
5. If you want to keep the changes, click **OK**.
6. Right click on the chart, click **Format Chart**, and click **General** in the pull- down menu to select layout and other options for changing the chart.



7. Make changes to items, such as **Show Markers on Lines** on the **General** tab.
8. Click the **Layout** tab to change the chart layout.
9. Click the other available tabs and make changes as needed.
10. Click **Apply** to see the changes; click **OK** to save the changes.
11. Right click on the chart, click **Format Chart**, and click **Titles** to change the chart tittles.



12. Edit titles as needed and click **OK**.
13. Right click on the chart, click **Format Chart**, and click **Grids** to changes the axis grids and scales.



14. Click on each of the tabs and make changes as desired.
15. Click **Apply** to see the changes.
16. Click **OK** to save the changes.

To change chart border:

1. Right click on the chart and click Format.
2. Click the **Border** tab of the Format Editor dialog box.
3. Change the line style, color, background color, and add or remove a drop shadow from the chart border

4. Click **OK** to save your changes.

Cross-Tab Objects

Cross-Tab Objects

A Cross-Tab object is a grid that displays values that are grouped and summarized in two directions. It returns values based on the criteria you specify. Data is presented in compact rows and columns that make it easy to compare data and identify trends. It is made up of three elements:

Rows – run horizontally from side to side.

Columns – run vertically up and down.

Summary fields – found at the intersection of rows and columns. Cross-Tab also

includes several totals:

At the end of each row – total for that row.

At the bottom of each column – total for that column.

At the intersection of the Totals column and the Totals row – grand total.

Creating a Cross-Tab Report

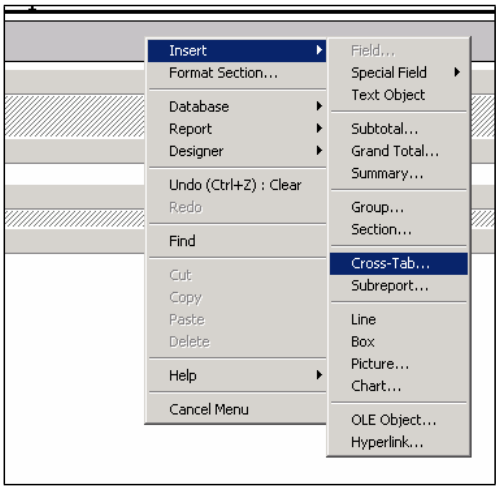
This section describes the steps to create a Cross-Tab object in a new report and to add a Cross-Tab object to an existing report.

Items to keep in mind when using Cross-Tab objects:

- You can have multiple rows, columns, and summarized fields.
- You can insert as many Cross-Tab objects in a report as you need.
- You can insert the Cross-Tab into either the Report Header or Footer, or the Group Header or Footer.
- You can place Cross-Tab objects in subreports; this is useful when you want to refer to the results from another report.

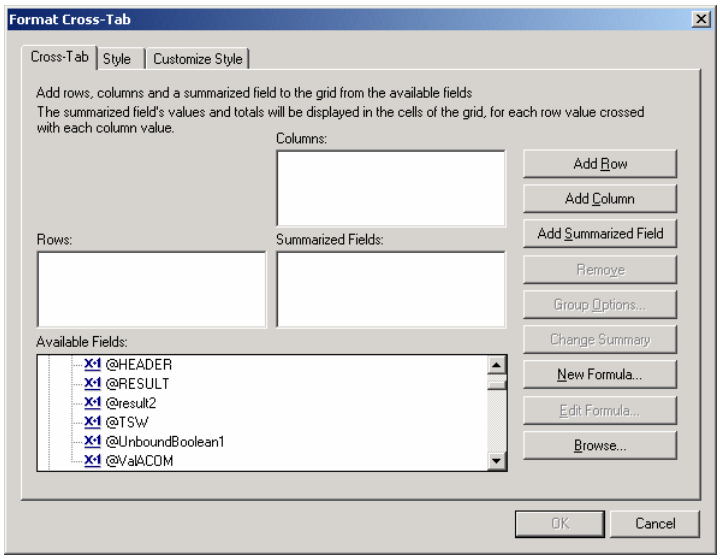
To create a cross-tab report:

1. Create a report using exported data or direct connect in LABWORKS Report Designer and insert cross-tab object.
2. Alternatively, open an existing report in LABWORKS Report Designer and insert cross-tab object.
   OR
   Click **File** on the menu bar of an open report and click **Create New Report**.
3. Right click on the report.
4. Click **Insert** from the pull-down menu.



5. Click on **Cross-Tab** in the next pull-down menu.

6. Drag the cross-tab field to the desired location and click to release. When you click to place the field, the Format Cross-Tab dialog box appears for entering specifications for the cross-tab object.



To specify cross-tab structure:

1. Add fields from the **Available Fields** area to the Rows, Columns, and Summarized Field areas.

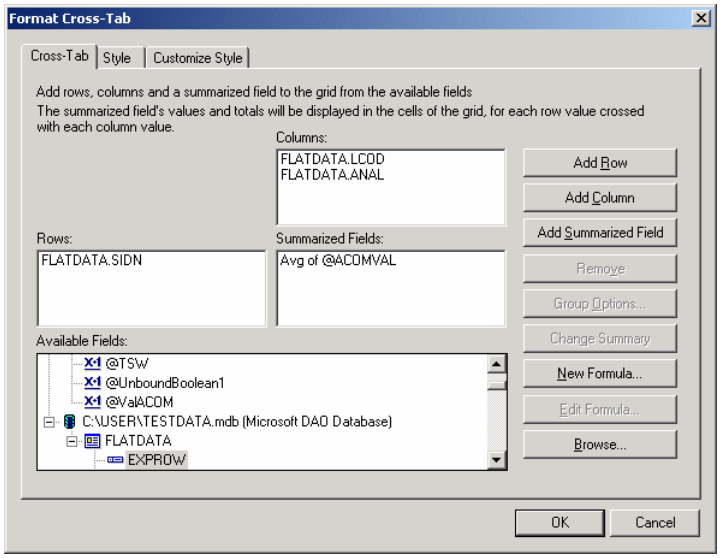
2. Click on a field and click the **Add Row** button to add it to the **Rows** list; add as many rows and fields as needed. You may also drag and drop fields to appropriate Rows, Columns, or Summarized Fields lists.

3. Click on a field and click the **Add Column** button (or drag and drop) to add it to the **Columns** list; add as many as needed.

4. Click on a field and click the **Add Summarized Field** button (or drag and drop) to add it to the

Summarized Fields list; add more summarized fields as needed.

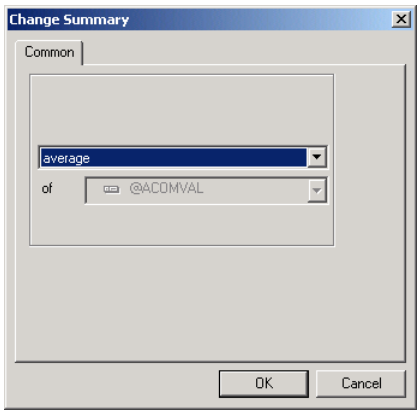
5. Click on a field and click **Remove** to delete it from the list.



6. Click on a field then click **New Formula** to add a new formula to the field using the Formula Editor, if desired. See Using Formulas section of this chapter.

7. Click on a field then click **Group Options** to specify the sort method, i.e., ascending, descending, or specified order, if available.

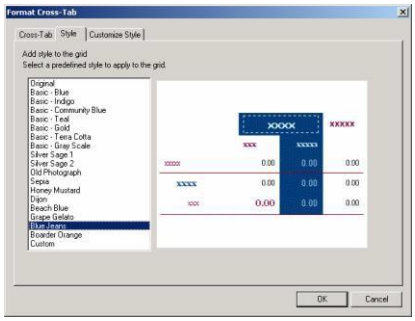
8. Click on a field then click **Change Summary** to change the summary operation.



9. Change summary and click **OK** or click **Cancel** to close.

To define style for the cross-tab:

1. Click on the **Style** tab.

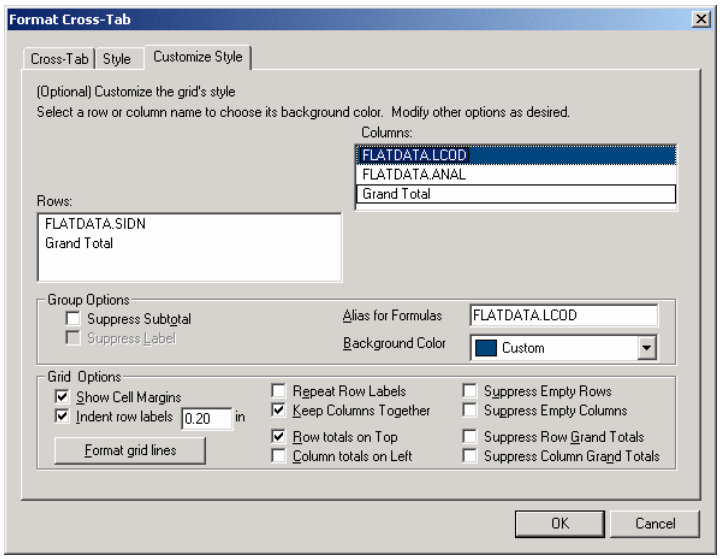


2. Click on any of the predefined styles to have them appear in the right view pane.

3. Select the style you want to use.

To customize the style:

1. Click on the **Customize Style** tab.

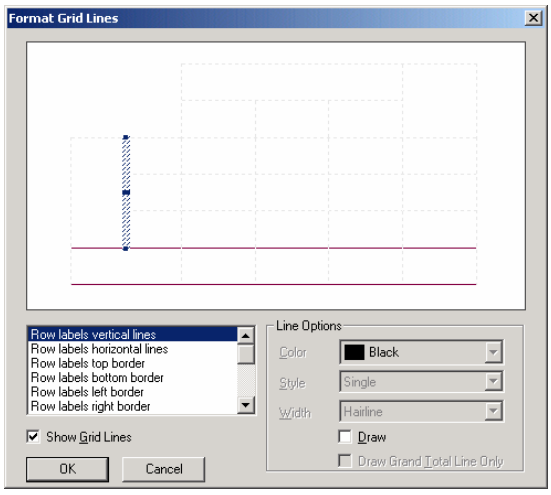


2. To change the background color of a row or column, select the row or column and select the color from the **Background color** drop-down list.

3. To create an alias for a row or column, select the row or column and enter the alias name in the **Alias for Formulas** field. You can create aliases to shorten long row or column names. This is useful if referring to a row or column in conditional formatting formulas.

4. Click on the **Grid Options** to change the formatting of your grid.

5. Click on the **Format Grid Lines** button to specify formatting for the grid lines



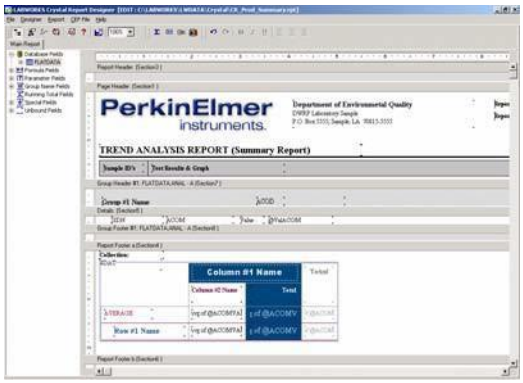
6. Scroll through the list and select the description of the area where you would like the lines to appear; or click the appropriate area in the Format Grid Line diagram.

7. Click **Show Grid Lines** to show the grid lines; otherwise, leave unchecked.

8. Select the color, style, and width.

9. Click **OK**.

EXAMPLE of Cross-Tab object



Formatting Cross-Tabs

LABWORKS Report Designer provides several options for formatting cross-tabs in your report to create a professional report.

Changing width and height of cross-tab fields

To change width or height of fields:

1. Open the report in the LABWORKS Report Designer.

2. Click a Row or Column heading within the Cross-Tab to activate the sizing handles.

3. Click and drag one of the resizing handles of a Cross-Tab cell to the desired width or height.

NOTE: Changing one row or column heading affects all rows or columns.

Formatting background color of entire rows/columns

To format background color:

1. Right click the blank top left area of the Cross-Tab and select **Cross-Tab Expert.** The Format Cross-Tab dialog box appears.
2. Click the **Customize Style** tab.
3. Click the row or column and select a color from the **Background Color** drop- down list.
4. Click **OK** to save changes and return to the cross-tab.

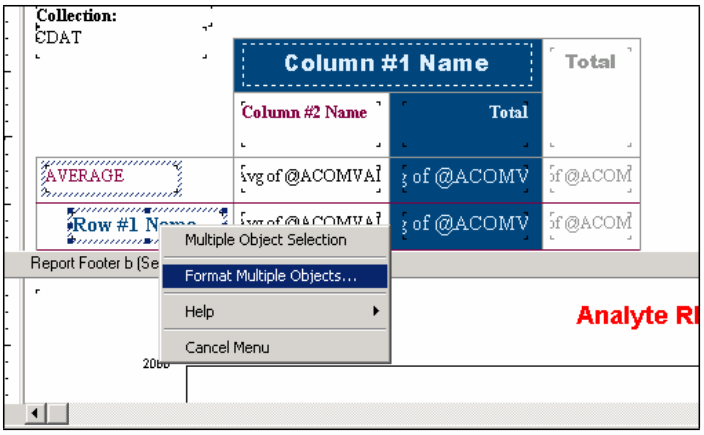
Formatting fields

To format individual fields:

1. Right click the field you want to format and click **Format** in the pull-down menu.
   The Format Editor appears.
2. In the Format Editor, select font, background, borders, numbering, currency symbols, and printing characteristics options.
3. Click **OK** to return to the Cross-Tab.

To format several fields at one time:

1. Use the Shift-click method to select the desired fields.



2. Right click on a field and click **Format Multiple Objects**. The Format Editor appears.
3. In the Format Editor, select font, background, borders, numbering, currency symbols, and printing characteristics options.
4. Click **OK** to return to the Cross-Tab. The selected fields are formatted as specified.

Suppressing cross-tab data

You can suppress:
Empty rows and columns
Row and column grand totals
Subtotals and their labels

To suppress empty rows and columns:

1. Right click the blank top left area of the Cross-Tab and select **Cross-Tab Expert.** The Format Cross-Tab dialog box appears.
2. Click the Customize Style tab.
3. Select either the Suppress Empty Rows or Suppress Empty Columns checkbox.
4. Click **OK**. When you print the report, empty rows and/or columns will not appear.

To suppress row and column grand totals:

1. Right click the blank top left area of the Cross-Tab and select **Cross-Tab Expert.** The Format Cross-Tab dialog box appears
2. Click the **Customize Style** tab.
3. Select either the Suppress Row Grand Totals or Suppress Column Grand Totals checkboxes.
4. Click **OK**.

To suppress subtotals and labels:

If you have more than two groups in your Cross-Tab you can suppress the subtotal and label for one of them.

1. Right click the blank top left area of the Cross-Tab and select **Cross-Tab Expert.** The Format Cross-Tab dialog box appears
2. Click the **Customize Style** tab.
3. Click the field whose subtotal you want to suppress. The Suppress Subtotal and the Suppress Label checkboxes become active.
4. Click the **Suppress Subtotal** checkbox in the **Group Options** area.
5. Click the **Suppress Label** checkbox to suppress the label associated with subtotal.
6. Click **OK**.

Printing Cross-Tabs on Multiple Pages

If a Cross-Tab is wider or longer than the specified page size, the program automatically spans the printing across enough pages to accommodate its size. For ease in reading, column headings are repeated on subsequent pages. Row headings can also be repeated using the Keep Columns Together option.

To repeat row labels:

1. Right click the blank top left area of the Cross-Tab and select **Cross-Tab Expert.** The Format Cross-Tab dialog box appears
2. Click the **Customize Style** tab.
3. Select the **Repeat Row Labels** checkbox, if desired.

Using Formulas

Using Formulas

This section explains the basics of formulas and using the Formula Editor to create formulas. Data needed for a report usually exists in database table fields, but there are times when you may need to create a formula to put data on the report that does not exist in any of the data fields.

Formula Components and Syntax

Formulas contain two critical parts: the components and the syntax. The components are the pieces that you add to create a formula, and the syntax is the rules that you follow to organize the components.

Formula components

Creating a formula in LABWORKS Report Designer is similar to creating a formula in any spreadsheet application. You can use any of the following components in your formula:

Fields Numbers Text

Operators – actions you can use in your formulas.

Functions – perform calculations such as average, sum, and count; all available functions are listed with their arguments and are arranged by their use.

Control Structures

Group field values – these values summarize a group. For example, you could use group field values to find the percentage of the grand total contributed by each group.

Other formulas

Formula syntax

Syntax rules are used to create correct formulas. Some basic rules
are:

- Enclose text strings in quotation marks.
- Enclose arguments in parentheses, where applicable. Referenced formulas are identified with a leading sign.

Crystal and Basic syntax

You have the option of using either Crystal or Basic syntax. Almost any formula written with Crystal syntax can be written with Basic. Reports can contain formulas that use Basic syntax as well as formulas that use Crystal syntax.

If you are familiar with Microsoft Visual Basic or other versions of Basic, then Basic syntax may be more familiar to you. In general, Basic syntax is modeled on Visual Basic except that it has specific extensions to handle reporting. If you are already comfortable with Crystal syntax, you can continue to use it.

Report processing is not slowed down by using Basic syntax. Reports using Basic syntax formulas can run on any computer running LABWORKS Report Designer. Using Basic syntax formulas does not require distributing any additional files with your reports.

Specifying Formulas

LABWORKS Report Designer contains four groups of formulas. The majority of formulas in a report are report formulas and conditional formatting formulas.

Report formulas – use these as stand-alone formulas in a report; e.g., calculating days between submittal date and completion date.

Conditional formatting formulas – use these formulas to change the report layout and design or appearance of text, database fields, objects, and report sections; format text through the Format Editor; create formatting formula using the Formula Editor in the Format Editor.

Selection formulas – use to specify and limit the records and groups that appear in a report; usually done by specifying the selection using the Select Expert. LABWORKS Report Designer then generates the record selection and group selection formula. You have the option to edit these formulas manually, but you must use Crystal syntax.

Search formulas – use these formulas to help locate data in your report; like selection formulas, you usually do not enter these formulas directly but specify search criteria using the Search Expert. LABWORKS Report Designer generates the formula. You have the option to edit these formulas manually, but you must use Crystal syntax.

If you already know Basic syntax, only a minimal knowledge of Crystal syntax is needed to modify most selection and search formulas.

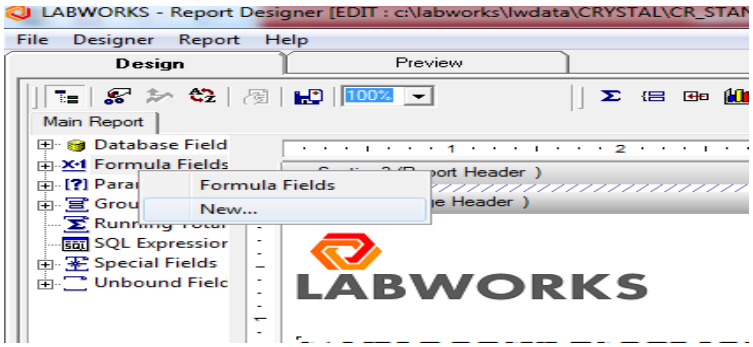
Using the Formula Editor

Other than selection and search formulas, you create all formulas in the Formula Editor.

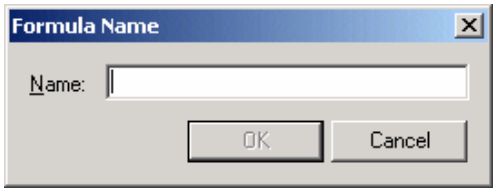
Accessing the formula editor

To access the Formula Editor:

1. Right click on **Formula Fields**.

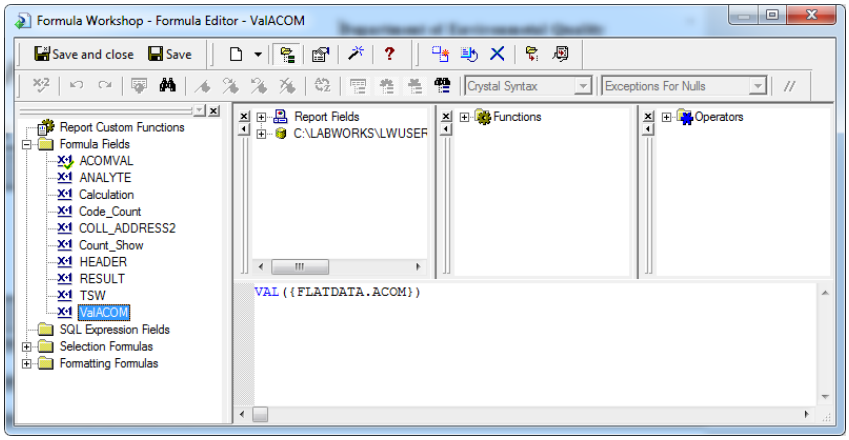


2. Click **New** in the pull-down menu.



3. Enter the name of the formula you will create in the **Formula Name** dialog box.

4. Click **OK**.



Formula editor sections

The Formula Editor contains four main sections.

Report Fields – contain all database fields accessible for your report and any formulas or groups already created for the report

Functions – pre-built procedures that return values; perform calculations such as average, sum, count, sin, trim, and uppercase

Operators – actions; describe an operation or an action to take place between two or more values; e.g., add, subtract, less than, greater than

Formula text window – area where you create a formula

Selecting syntax

The drop-down list in the top right corner of the Formula Editor allows you to select either Crystal or Basic syntax for creating the formula.

Changing the syntax from Crystal syntax to Basic syntax or vice versa changes the list of functions in the Functions window and the list of operators in the Operators window. The functions and operators differ from syntax to syntax.

Since the report fields are available to either syntax, the available report fields remain the same.

Entering formula components

The Report Fields, Functions, and Operators contain the primary formula components. Double click any component to add this component to your formula.

For example, if you set the syntax to Basic Syntax and double click the Operators > Control Structures > Multi-Line If in the Operators list, the following text is transferred to the Formula text window with the cursor between the If and Then:

If | Then ElseIf Then Else

End If

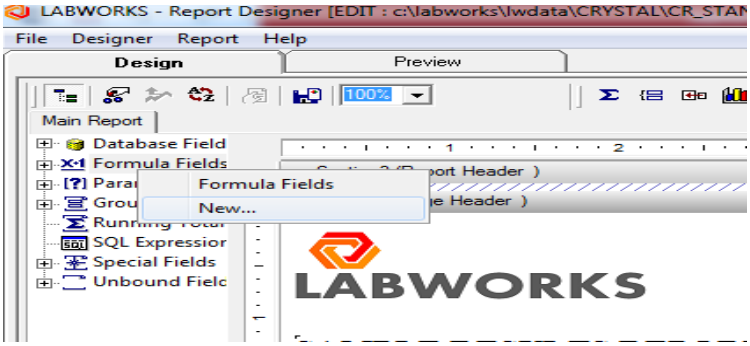
The above text helps you organize the parts needed to write your formula.

Creating and Modifying Formulas

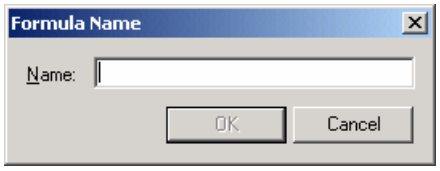
Creating and inserting a formula in a report

To create a formula:

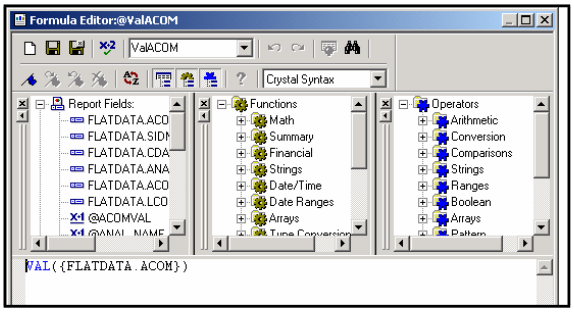
1. Right click on **Formula Fields**.



2. Click **New** in the pull-down menu.



3. Enter the name of the formula you will create in the **Formula Name** dialog box.
4. Click **OK**.



5. Select either **Crystal** or **Basic** syntax.
6. Enter the formula by typing in the components or selecting them from the component lists.
7. Click Check to identify any errors in the formula.
8. Correct any syntax errors the Formula Checker identifies.
9. Click **Save and Close** when the formula has the correct syntax.

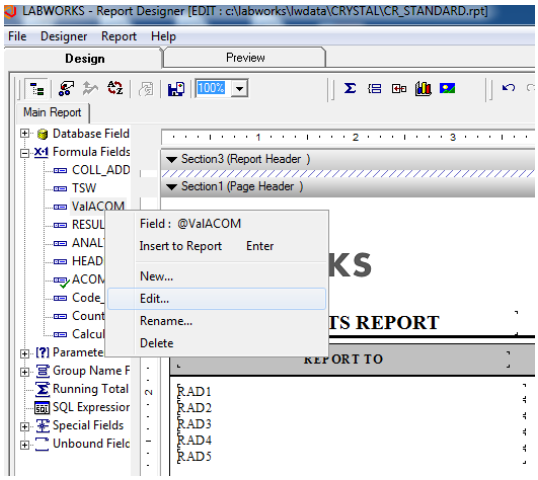
To insert the formula in the report:

1. Double click the formula in the Formula fields list.
2. Place the cursor where you want the formula to appear on your report.
3. Click once to place the field in the desired position.

NOTE: A formula placed on a report is indicated by @ symbol.

Modifying formulas

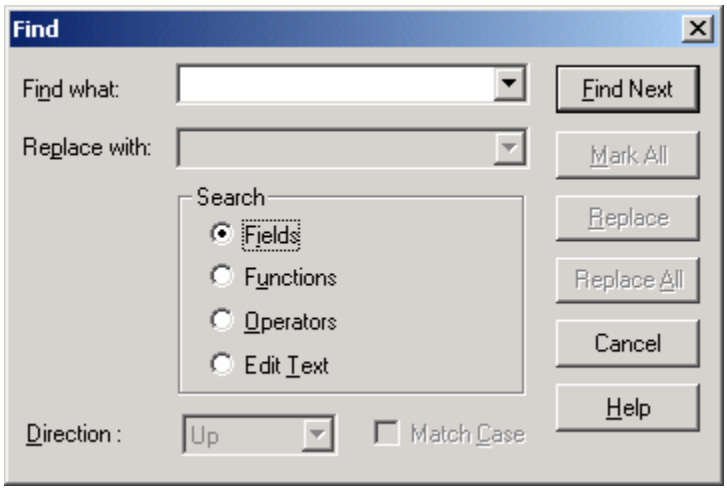
1. Right click on the formula you want to edit in the **Formula Fields** list.



2. Click **Edit**. The Formula Editor appears.
3. Edit the formula.
4. Click Check to identify any errors in the formula.
5. Correct any syntax errors the Formula Checker identifies.
6. Click **Save and Close** when the formula has the correct syntax.

Searching and replacing text

1. Right click on the formula you want to edit in the **Formula Fields** list.
2. Click **Edit**. The Formula Editor appears.
3. Click the Find/Replace **icon** on the menu bar of the Formula Editor dialog box. The Find dialog box appears.



4. Enter the text to search for the in the **Find what** field.
5. Enter the text to replace the found text in the **Replace with** field.
6. Click on **Search** option to direct the search to a specific list in the Formula Editor.
7. Click Mark All to mark all occurrences of the search text.
8. Click Replace All to replace all occurrences of the search text with the contents of the **Replace with** text box.

NOTE: You can also search but not replace within any of the Formula Editor lists. The Mark All, Replace, and Replace All buttons become inactive when you specify a search within a list box.

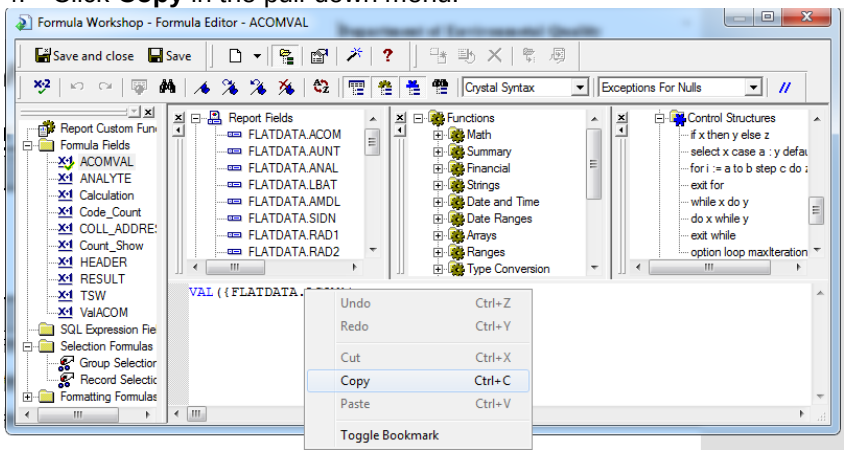
Copying Formulas from One Report to Another

To copy a formula field from one report to another:

1. Select the formula field you want to copy in the report.
2. Right click on the formula field and click **Copy** in the pull-down menu.
3. Open the report to which you want to copy the formula.
4. Right click and click **Paste** from the pull-down menu.
5. Drag the formula to where you want it on the report.
6. To make changes to the formula, right click on the formula field and click **Edit Formula** from the pull-down menu. The Formula Editor appears.
7. Delete the old values and type in the new values, or select them from the Fields, Functions, and/or Operators tree.
8. Click **Save and Close** when finished.

To copy a formula from one report to another:

1. Right click on the formula field in the **Formula Fields** list from which you want to copy a formula.
2. Click **Edit**. The Formula Editor appears.
3. Select and right click on the formula you want to copy.
4. Click **Copy** in the pull-down menu.



5. Open the report to which you want to copy the formula.
6. Create the new formula; click in the **Formula** text window of the Formula Editor and click **Paste** to

insert the formula.

OR

Use the edit function to add the formula to an existing formula field in the report by pasting the formula in the Formula Editor.

Key points for editing a copy of a formula

When making changes, use the following points as a guide:

- All fields, formulas, and group fields referenced in the formula copy must actually exist in the new report. This means that any database referenced in the original formula or a database with the same structure, field names, and alias must be active in the new report.
- If such a database is not active, you must change the field, formula, and group field references in the formula copy to correspond to elements in your new report.
- If the formula contains conditional elements, make certain that the conditions apply to the data in the new report. For example, if the formula in your old report performed an action when the quantity was greater than 100, make sure that the greater than 100 condition makes sense in the new formula. When modifying a formula, you may find that greater than 10 or greater than 2000 makes more sense with your new data.
- If you are using the formula with new data, and if your report contains statements similar to the following:

  If {file.FIELD} = "text string"

  Make sure that the text strings used in the formula match values that actually exist in the new data.

Deleting Formulas

When a formula is created and added to a report, the Report Designer stores the specification for creating the formula, using the name you assigned to it. It places a working copy of that formula at the point you specify in the report. A working copy is any occurrence of the formula in the report.

To delete formulas completely, you must delete the specification and all working copies of the formula. You cannot delete the specification without deleting all working copies of the formula.

To remove working formulas from reports:

1. In the report, right click the **formula field** you want to delete.
2. Click **Delete** in the pull-down menu.

NOTE: After deleting the working copies of a formula from the report, the formula specification remains unchanged and listed in the Formula Fields list on the left portion of the LABWORKS Report Designer. The formula field remains available for you to insert it in the report again if needed.

To delete formula specifications:

1. Right click the formula you want to delete in the **Formula Fields** list.

2. Click Delete in the pull-down menu.

NOTE: A dialog box appears if this formula is currently in use in a report. If you delete this formula, you will delete all references of it in reports.

3. Click **Yes** to delete.

Creating Formulas with Basic Syntax

Basic Syntax Overview

When creating formulas, you have the option of using either Crystal or Basic syntax. Syntax rules are used to create correct formulas. Almost any formula written with one syntax can be written with the other. Reports can contain formulas that use Basic syntax as well as formulas that use Crystal syntax.

If you are familiar with Microsoft Visual Basic or other versions of Basic, then Basic syntax may be more familiar to you. In general, Basic syntax is similar to Visual Basic except that it has specific extensions to handle reporting.

Using Basic syntax does not slow report processing. Reports using Basic syntax formulas can run on any computer that LABWORKS Report Designer runs on. Also, using Basic syntax formulas does not require distributing any additional files with your reports.

Tips on Basic Syntax

If you have no programming experience

In many cases you may not need to use the formula language. LABWORKS Report Designer includes several Experts that automatically handle situations where formulas could be used. These include the Select, Search, Running Totals and Highlighting Experts, and the Insert Summary and Insert Grand Total dialog boxes. Before creating formulas, check to see if you can use one of these tools.

However, you may need to create formulas without the help of an Expert. Read this section to learn about Basic syntax and the rules you need to follow to create a formula.

If you know Microsoft Visual Basic, VBScript, or another version of Basic

In order to create formulas using Basic syntax, you need to understand the following:

- How a Basic syntax formula refers to other fields in the report such as database fields, parameter fields, summary fields, running total fields and other formula fields.
- How to return a value from a formula by setting the special variable named formula.
- How to use functions specific to report processing such as ReportTitle and OnFirstRecord.
- How to use the type system.

Basic syntax is strongly typed, similar to using Option Explicit in Visual Basic and there is no Variant type. Basic syntax supports separate Date, Time and DateTime types unlike just the Date type in Visual Basic. It also supports range types, such as
10 to 20, to allow for ranges of values that are common in many reporting applications.

Familiar features

Many Basic syntax functions work in the same way as their counterparts in Visual Basic. This includes string functions such as Len, Mid and Filter, math functions such as Abs, Rnd and Sin, financial functions such as PV, programming shortcut functions such as IIF and date functions such as DateSerial, DateAdd and DateDiff. Most operators supported by Visual Basic are also in Basic syntax.

For example, string concatenation (&) and date-time literals (#...#).Most statements and control structures use the same syntax as in Visual Basic. This includes the If, Select, Do While, Do Until, While and For/Next statements.

The overall look of the formula will be unmistakably like Basic. For example, Basic style comments and line continuation characters are supported as is the Basic language use of new lines, colons, and the equal sign.

Basic Syntax Fundamentals

Formula results

The result of a formula, or the value that is printed when the formula is placed in a report, is called the value returned by the formula. Every formula in LABWORKS Report Designer must return a value. Basic syntax does this by setting the value of the special variable formula. For example, here is a simple Basic syntax formula that returns the value 10:

```
formula = 10
```

The value returned by a formula can be one of the seven simple data types supported: Number, Currency, String, Boolean, Date, Time, and DateTime. LABWORKS Report Designer also supports range types and array types, but these cannot be returned by a formula.

Restrictions on changing formula variable data types

The formula variable can be set several times within a single formula. For example, suppose a company has a shipping policy in which orders over $1,000 are insured, but orders below that amount are not insured:

```
Rem A formula that returns a String value
If {Orders.Order Amount} >= 1000 Then
    formula = "Insured Shipping"
Else
    formula = "Regular shipping"
End If
```

The text following the keyword Rem is a comment for the user reading the formula and is ignored by the Basic syntax compiler.

The above formula returns the text string value "Insured shipping" if the value of the database field {Orders.Order Amount} is greater than or equal to 1000; otherwise, it returns the text string value "Regular Shipping." Text strings are usually just referred to as strings. Notice that the formula variable appears twice in the above example.

If the formula variable is set to a value of one type, it cannot be set to a value of another type later in the same formula. For example, replacing the String "Regular shipping" in the above example with the Number 10 would result in an error since the special variable formula was first set to the String value "Insured shipping."

The reason for this restriction is that LABWORKS Report Designer needs to know in advance what the return type of a formula will be so that it can allocate enough storage for the returned values. This is because different types have different storage requirements. Another reason is that the formatting options available for a formula field depend on its type. For example, a Number field has Number formatting options, such as the number of decimals to display, which do not make sense for a String field.

The special variable formula should not be declared, unlike other variables used in a Basic syntax formula.

How formulas relate to functions in Microsoft Visual Basic

Consider the following Basic syntax formula:

```
Rem A formula that returns a String value
Rem The function Rnd returns a random number
Rem between 0 and 1
If Rnd > 0.9 Then
    formula = "You won!"
Else
   formula = "Sorry, try again."
End If
```

The above formula returns the text string value "You won!" if the random number returned by Rnd is greater than 0.9 and the text string value "Sorry, try again." otherwise.

The use of the formula variable is similar to writing a function named formula in Visual Basic.

For example, the above formula could be written as a Visual Basic function as follows:

```
Rem The following code is in Visual Basic
Function formula()
If Rnd > 0.9 Then
formula = "You won!" Else
formula = "Sorry, try again." End If
End Function
```

Variable formula must be assigned a value

If the variable formula is not assigned a value, it is not a complete Basic syntax formula. Some examples in this section are not complete Basic syntax formulas but rather just fragments intended to explain a particular feature.

Sometimes you may want to write a formula that just declares and initializes some global variables. These formulas are commonly inserted into the report header section of a report. In such cases, assign any value to the special variable formula. Every formula must return a value, even if you are not interested in using that value. For example:

```
Rem Some Global variable declarations
Rem Remember to set the value of 'formula' Global x As String, y As
Number, z As DateTime x = "hello"
y = 10.5
z = #Aug 6, 1976#
formula = 10
```

Basic syntax is not case-sensitive

What this means is that formula, Formula, and FORMULA are all considered the same. This is true of all variable names, functions, and keywords used in a Basic syntax formula.

The only exception to this rule is for strings. The string "Hello" is not the same as the string "hello".

Comments (Basic syntax)

Formula comments are notes included with a formula to explain its design and operation. Comments do not print and they do not affect the formula; they appear only in the Formula Editor. Use comments to explain the purpose of a formula or explain the steps involved in writing it.

Comments work as in Visual Basic. Begin comments with a Rem or an apostrophe.

NOTE: A comment beginning with a Rem is a separate statement and must either start on a new line or be separated from the previous statement by a colon.

```
Rem This is a comment
Rem This is another comment
formula = 10 'So is any text after an apostrophe
formula = 20 : Rem This is also a comment
'Comments can occur after the formula text
```

Fields (Basic syntax)

Many of the fields you use when creating your report can also be referred to in your formulas. For example, database, parameter, running total, SQL expression, summary, and group name fields can all be used in a formula. You can also refer to other formula fields in your formula.

The easiest way to insert a field into your report is to double-click a field's name in the Report Fields tree. This ensures that the correct syntax for the field is used.

How fields appear in formulas

Database, parameter, formula, running total and SQL expression fields have their names surrounded by braces. Database field names are taken from the database. For example:
database field: {Employee.Last Name}

- **Parameter, formula, running total, and SQL expression** field names are specified when the fields are created. Parameter fields also includes a question mark: {?my parameter field}.
- **Formula fields** include an at sign: {@another formula }.
- **Running totals fields** include a pound sign: {#my running total}.
- **Sql expression fields** include a percent sign: {%my sql expression}. Summary and group name fields look like function calls but are really shorthand notation for a report field.
- **Sum summary field**: Sum({Orders.Order Amount}, {Orders.Ship Via}).
- Group name field: GroupName({Orders.Ship Via}).

Sample formulas using fields

The formula in this example uses the Example database. To find out how many

days it takes to ship the product from the date when the order was placed, subtract the ship date database field from the order date database field:

```
Rem A formula that uses database fields
formula = {Orders.Ship Date} - {Orders.Order Date}
```

To find the total dollar amount of a given product that was ordered, multiply its unit price by the quantity ordered:

```
formula = {Orders Detail.Unit Price} * _
        {Orders Detail.Quantity}
```

The example uses the line continuation character " _ " (space underscore).

To calculate a sale price of 80 percent of the original unit price:

```
formula = {Orders Detail.Unit Price} * 0.80
```

Statements (Basic syntax)

A Basic syntax formula consists of a sequence of statements. Each statement must be separated from the previous statement by either a new line or a colon. Typically, each statement takes one line, but you can continue a statement onto the next line by using the line continuation character, which is a space followed by an
underscore.

For example:

```
'Declare a variable x to hold a number
Dim x As Number
'Assign the value of 30 to x
x = 10 + 10 + 10
'This also assigns the value of 30 to x
x = 10 + _
10 + 10
'Line continuation characters _
can also be used in comments
Dim y as String
'Three statements separated by two colons
y = "Hello" : x = 30 : formula = True
```

Assignment and Data Types (Basic syntax)

Assignment

Use the equal sign (=) when making assignments. The keyword Let can be optionally included as well. For example:

```
x = 10
Let y = 20
```

Simple Data Types (Basic syntax)

The simple data types in LABWORKS Report Designer are Number, Currency, String, Boolean, Date, Time and DateTime.

Number (Basic syntax)

Enter numbers without any comma separators or currency symbols. (Generally, you would want to have formatted numbers appearing as the result of a formula and
not in the formula itself.)

Examples

10000

-20

1.23

Currency (Basic syntax)

Use the CCur function to create a Currency value. The initial C in CCur stands for convert and it can be used to convert Number values to Currency values.

Examples

CCur (10000)

CCur (-20)

CCur (1.23)

String (Basic syntax)

Strings are used to hold text. The text must be placed between double quotation marks (") and cannot be split between lines. If you want to include double quotes in a string, use two consecutive double quotation marks.

"This is a string."

"123"

"The word ""hello"" is quoted."

You can extract individual elements or substrings from a string by specifying the character position or a range of character positions. Negative values are allowed; they specify the position starting from the end of the string.

"hello" (2) 'Equal to "e" "hello" (-5) 'Equal to

"h"

"604-555-1234" (1 to 3) 'Equal to "604"

"abcdef" (-3 to -1) 'Equal to "def"

You can also extract substrings from a string using the Left, Right and Mid functions.

Boolean (Basic syntax)

The Boolean values are:

True
False
Yes can be used instead of True and No instead of False.

Date, Time, and DateTime (Basic syntax)

The DateTime type can hold date-times, dates only or times only. The Date type holds dates only and the Time type holds times only. The Date and Time types are more efficient than the DateTime type, and so can be used in situations where the added functionality and flexibility of the DateTime type is not needed.

Visual Basic does not support separate types for holding dates only or times only. The Basic syntax DateTime type is similar to Visual Basic's Date type.

You can create DateTime values directly using the date-time literal construction. It is formed by typing in the date-time between 2 pound (#) signs. Many different formats are supported, as in Visual Basic.

NOTE: Date-time literals cannot be split between lines.

Examples

#8/6/1976 1:20 am#
#August 6, 1976#
#6 Aug 1976 13:20:19#
#6 Aug 1976 1:30:15 pm#
#8/6/1976#
#10:20 am#

Even though #10:20 am# looks like it could have the Time type and #8/6/1976# looks like it could have the Date type, they do not. They both have the DateTime type, as do all date-time literals. For example, you can think of #10:20 am# as a DateTime value with a null date part. To convert it to the Time type use CTime (#10:20 am#).

Instead of using date-time literals, you can use CDateTime to convert a String to a DateTime. For example:

CDateTime ("8/6/1976 1:20 am") CDateTime ("10:20 am")

However, there is one key difference between using date-time literals and the above usage of CDateTime. Date-time literals always use U.S. English date formats rather than settings from the locale of the particular computer on which LABWORKS Report Designer is running. Thus, the date-time literal examples above would work on all computers. On the other hand, on a French system, you could use constructions like:

CDateTime ("22 aout 1997") 'Same as #Aug 22, 1997#
Date values can be constructed with CDate and Time values with CTime: CDate ("Aug 6, 1969")
CDate (1969, 8, 6) 'Specify the year, month, day
'Converts the DateTime argument to a Date
CDate (#Aug 6, 1969#)
CTime ("10:30 am")
CTime (10, 30, 0) 'Specify the hour, minute, second

CTime (#10:30 am#)

Range data types (Basic syntax)

Ranges are designed to handle a spectrum of values. Range types are available for all the simple types except for Boolean. That is: Number Range, Currency Range, String Range, Date Range, Time Range and DateTime Range. You can generate ranges using the To, _To, To_, _To_, Is >, Is >=, Is < and Is <= keywords. In general, To is used for ranges with 2 endpoints, and Is is used for open ended ranges (only one endpoint). The underscores are used to indicate whether or not the endpoints are in the range.

Examples of Number Range values

The range of numbers from 2 to 5 including both 2 and 5

2 To 5

The range of numbers from 2 to 5, not including 2 but including 5

2 _To 5

All numbers less than or equal to 5

Is <= 5

All number less than 5

Is < 5

Examples of DateTime Range values

#Jan 5, 1999# To #Dec 12, 2000#

Is >= #Jan 1, 2000#

Using Ranges in Formulas (Basic syntax)

There are twenty-seven functions in LABWORKS Report Designer that specify date ranges. For example, the function LastFullMonth specifies a range of date values that includes all dates from the first to last day of the previous month. So if today's date is September 15, 1999 then LastFullMonth is the same as the range value CDate (#Aug 1, 1999#) To CDate (#Aug 31, 1999#).

Ranges are often used with If or Select statements. The following example computes student letter grades based on their test scores. Scores greater than or equal to 90 receive an "A", scores from 80 to 90, not including 90 receive a "B" and so on.

```
Rem Compute student letter grades
Select Case {Student.Test Scores}
Case Is >= 90
    formula = "A"
Case 80 To_ 90
    formula = "B"
Case 70 To_ 80
    formula = "C"
Case 60 To_ 70
    formula = "D"
```

```
    Case Else
        formula = "F"
    End Select
```

The above example uses the Select statement which is discussed in more detail in Control structures (Basic syntax). You can check if a value is in a range by using the In operator. For example:

```
formula = 5 In 2 To 10 'True
formula = 5 In 2 To_ 5 'False
formula = 5 In 2 To 5 'True
```

The Maximum and Minimum functions can be used to find the endpoints of a range:
formula = Maximum (2 To 10) 'Returns 10

Array Data Types (Basic syntax)

Arrays in LABWORKS Report Designer are ordered lists of values that are all of the same type. These values are known as the array's elements. The elements of an array can be any simple type or range type. One way to create an array is using the Array function.

Examples

An array of 3 Number values. The first element is 10, second 5, and third 20.

```
Array (10, 5, 20)
```

An array of 7 String values:

```
Array ("Sun", "Mon", "Tue", "Wed", "Th", "Fri", "Sat")
```

An array of 2 DateTime Range values (note the line continuation character which is used for readability):

```
Array (#Jan 1, 1998# To #Jan 31, 1998#, _
#Feb 1, 1999# To #Feb 28, 1999#)
```

You can extract individual elements out of an array using parentheses containing the index of the element you want. This is called subscripting the array:

```
Array (10, 5, 20) (2) 'Equal to 5
```

Arrays in Basic syntax are indexed from 1 (this means the first element has index 1). This is unlike in Visual Basic where arrays are indexed from 0 by default. However, in Visual Basic, arrays can be indexed from 1 by using the Option Base statement.

Number ranges can also be used to subscript arrays. The result is another array. For example:

```
Array (10, 5, 20) (2 To 3) 'Equal to Array (5, 20)
```

Arrays are most useful when used with variables. Using variables, you can change the individual elements of an array and resize the array to accommodate more elements. This capability significantly expands the capabilities of the formula language to do complex calculations.

For example, you can accumulate database field values into a global array variable in a detail level formula, and then use a formula in a group footer to perform a calculation based on those values. This allows you to perform a wide variety of customized summary operations.

Variables (Basic syntax)

This section describes the key components of variables and shows you how to create variables and assign values to them.

Variable overview

A variable represents a specific data item, or value, and acts as a placeholder for that value. When a formula encounters a variable, the formula searches for the value of the variable and uses it in the formula. Unlike a constant value, which is fixed and unchanging, a variable can be repeatedly assigned different values. You assign a value to a variable and the variable maintains the value until you later assign a new value. Because of this flexibility, it is necessary for you to declare variables before you use them so that LABWORKS Report Designer is aware of them and understands how you intend to use them.

Example of a variable

If you wanted to report on customers by area code, you could create a variable that extracts the area code from a customer fax number. The following is an example of a variable called areaCode:

```
Dim areaCode As String
areaCode = Left ({Customer.Fax}, 3)
Rem could also use: areaCode = {Customer.Fax} (1 To 3)
```

The first line of the variable example is the variable declaration; it gives the variable a name and type. The database field {Customer.Fax} is a String field and the Left function extracts the first 3 characters from its current value. The variable areaCode is then assigned this value.

Variable declarations using dim (Basic syntax)

Before using a variable in a formula, you must declare it. A variable can hold values of a given type. The allowed types are the seven simple types (Number, Currency, String, Boolean, Date, Time and DateTime), the six range types (Number Range, Currency Range, String Range, Date Range, Time Range and DateTime Range) and variables that hold arrays of the previously mentioned types. This gives a total of 26 different types that a variable can have. When you declare a variable, you also specify its name. A variable cannot have the same name as any function, operator, or other keyword that is valid for Basic syntax. For example, your variable cannot be named Sin, Mod, or If because Sin is a built in function, Mod is a built in operator, and If is a built in keyword. When typing formulas in the formula editor, the names of the built-in functions, operators, and other keywords are highlighted in a different color. This makes it easy to check if the variable name conflicts.

Once a variable is declared, it can be used in the formula. For example, you might want to assign it an initial value:

```
Dim x As Number 'Declare x to be a Number variable
x = 10 'Assign the value of 10 to x
```

Variables can only hold values of one type (Basic syntax)

A variable can only hold values of one type. For example, if a variable holds a Number value you cannot later use it to hold a String.

Example

```
Dim y As String
y = "hello"
'OK - the Len function expects a String argument
formula = Len (y)
```

```
'Error- y can only hold String values
y = #Jan 5, 1993#
'Error- y can only hold String values
y = Array ("a", "bb", "ccc")
'Error- the Sin function expects a Number argument
formula = Sin (y)
```

You can declare more than one variable per statement by separating their declaration by commas:

```
Dim x As Number, y as String, z as DateTime Range
x = 10 : y = "hello"
z = #Jan 1, 1999# To #Jan 31, 1999#
```

Declaring variables without immediately specifying their type (Basic syntax)

In general, the type of a variable does not need to be explicitly given when declaring it. In such cases, the variable's type is determined by the first assignment that is made to it. This is similar to the special variable formula. This is different from in Visual Basic where a variable whose type is not given at declaration automatically has the Variant type. However, in practice, it means that you can write formulas in a similar style to what you would do if using a Variant in Visual Basic.

```
Dim p 'The type of p is not known yet
p = "bye" 'The type of p is now set to be String
Dim q 'The type of q is not known yet
q = Array ("hello", p) 'q is a String Array
'Error- p is a String variable and cannot hold a Number
p = 25
Dim r
'r is a Number variable, and holds the value 5
r = (10 + 5) / 3
'The types of a and c are not known yet
Dim a, b As Boolean, c
b = False
'The type of a is now set to Boolean
'and its value is False
a = b
'The type of c is now set to Number and its value is 17
c = 2 + 3 * 5
```

Examples of declaring and initializing range variables

```
Dim gradeA, quarter
'The type of gradeA is set to Number Range
gradeA = 90 To 100
'The type of quarter is set to Date Range
quarter = CDate (1999, 10, 1) To CDate (1999, 12, 31)
```

Variable Scope (Basic syntax)

Variable scopes are used to define the degree to which variables in one formula are made available to other formulas. There are three levels of scope: local, global and shared. Every variable has a scope, and this scope is specified when the variable is declared.

Local variables

Variables with local scope, also known as local variables, are declared using either the Dim or Local keywords. For example, all the declarations in the previous section using Dim were declaring local variables. Another example:

```
Local x As Number 'equivalent to Dim x As Number
```

Local variables are restricted to a single formula and a single evaluation of that formula. This means that you cannot access the value of a local variable in one formula from a different formula.

Example

```
Rem Formula A
Local x as Number
x = 10
formula = x
Rem  Formula  B
EvaluateAfter ({@Formula A})
Local x as Number
formula = x + 1
```

The function call EvaluateAfter ({@Formula A}) ensures that Formula B will be evaluated after Formula A is evaluated. Formula A returns a value of 10 and Formula B returns a value of 1. Formula B does not have access to Formula A's x and thus cannot use the value of 10 and add 1 to it; instead, it uses the default value for the uninitialized local variable x found in Formula B, which is 0, and adds 1 to it to get 1.

You can also create local variables with the same name but different types in different formulas. For example, the type declarations in formulas A and B do not conflict with:

```
Rem Formula C
Local x as String x = "hello" formula = x
```

Local variables are the most efficient of the three scopes. In addition, they do not interfere with one another in different formulas. For these reasons, it is best to declare variables to be local whenever possible.

Global variables (Basic syntax)

Global variables use the same memory block to store a value throughout the main report. This value is then available to all formulas that declare the variable, except for those in subreports. Declare a global variable as in the following example:

```
Global y As String
```

Since global variables share their values throughout the main report, you cannot declare a global

variable in one formula with one type and then declare a global variable with the same name in a different formula with a different type.

Example

```
Rem Formula A
Global z As Date
z = CDate (1999, 9, 18)
formula = 10
Rem Formula B
Global z As Number
formula = True
```

In this case, if you enter and save Formula A first, LABWORKS Report Designer will return an error when you check or try to save Formula B. This is because the declaration of the Global variable z as a Number conflicts with its earlier declaration in Formula A as a Date.

Using Global variables

Global variables are often used to perform complex calculations where the results of a formula depend upon the grouping and page layout of the actual printed report. This is accomplished by creating several formulas, placing them in different sections of the report, and having the different formulas interact via global variables.

Here is an example of the sort of effects that can be produced:

```
Rem Formula C
Global x as Number
x = 10
formula = x
Rem Formula D
'call the function WhileReadingRecords
WhileReadingRecords
Global x as Number
x = x + 1
formula = x
```

If Formula C is placed in the Report Header and then Formula D is placed in a detail section, Formula C will be evaluated before Formula D. Formula C will be evaluated once and then Formula D will be evaluated for each record appearing in the detail section. Formula C returns 10. For the first detail record, Formula D returns 11. This is because the value 10 of x is retained from when it was set by Formula C. Formula D then adds 1 to this value, setting x to 11 and then returns 11. For the second detail record, formula D return 12, adding 1 to the previously retained value of x which was 11. This process continues for the remaining detail records.

The call to WhileReadingRecords tells LABWORKS Report Designer to re-evaluate Formula D as it reads in each record of the report. Otherwise, since the formula does not contain any database fields, the program will evaluate it only once before reading the records from the database. The formula will then return the value 11 instead of 11, 12, 13,... as the successive records are processed.

If the statement x = x + 1 is replaced by x = x + {Orders Detail.Quantity}, you create the effect of a running total based on {Orders Detail.Quantity}, although it is one starting at 10 rather than 0 because

of Formula C. In this case, you can omit the call to WhileReadingRecords, since it will automatically occur because the formula contains a database field.

Shared variables (Basic syntax)

Shared variables use the same memory block to store the value of a variable throughout the main report and all of its subreports. Thus, shared variables are even more general than global variables. To use a shared variable, declare it in a formula in the main report as in the following example:

```
Shared x As Number
x = 1000
```

and declare it in a formula in the subreport as in the following example:

```
Shared x as Number
```

In order to use shared variables the variable must be declared and assigned a value before it can be passed between the main report and the subreport.

Declaring Array Variables (Basic syntax)

There are several different ways of declaring array variables. The first way is to use empty parentheses and explicitly specify the type of the array:

```
'Declare x to be a Global variable
'of Number Array type
Global x () As Number
'Initialize x
x = Array (10, 20, 30)
'Declare y to be a Shared variable
'of String Range Array type
Shared y () As String Range
'Initialize y
y = Array ("A" To "C", "H" To "J")
```

The second way is to declare the variable without specifying that it is an array and without giving its type and waiting for the first assignment to the variable to completely specify its type:

```
'Declare y to be a Local variable
'but do not specify its type
Dim y
'The type of y is now set to be a String Array
y = Array ("Sun", "Mon", "Tue", "Wed", "Th", _ "Fri", "Sat")
```

The third way is to declare that the variable is an array but not specify its type fully until the first assignment. Assuming the declaration of y above:

```
'Declare z to be a Local variable that is an Array
Local z()
'z is set to Array ("Mon", "Tue") and is a String Array
z = y(2 to 3)
```

The fourth way is to specify explicitly the size of the array during the declaration. If you use this technique, the array is automatically created and default values are used to fill the array. For example,

for a Number Array, each element is initialized to 0 and for a String array each element is initialized to the empty string "". Since this type of declaration actually creates the array, you must specify its type with the As clause so that LABWORKS Report Designer knows how much storage space to reserve for the array.

```
Dim a(2) As String
'Assign a value to the first element of the array a
a(1) = "good"
a(2) = "bye"
'The & operator can be used to concatenate strings
'the formula returns the String "goodbye"
formula = a(1) & a(2)
```

Arrays and For/Next loops

Arrays are commonly used with For/Next loops. The following example creates and then uses the array Array (10, 20, 30, ..., 100) using a For/Next loop. See For/Next loops (Basic syntax) for more details.

```
Dim b (10) As Number
Dim i
For i = 1 To 10
b(i) = 10 * i
Next i
formula = b(2) 'The formula returns the Number 20
```

Using array variables (Basic syntax)

You can assign values to elements of an array and use the values of the elements for other computations:

```
Global x() As String
x = Array ("hello", "bye", "again")
'Now x is Array ("hello", "once", "again")
x (2) = "once"
'The statement below would cause an error if not
'commented out since the array has size 3
'x (4) = "zap"
'The formula returns the String "HELLO"
formula = UCase (x (1))
```

The Redim and Redim Preserve keywords can be used to resize an array, which is useful if you want to add extra information to it. Redim erases the previous contents of the array first before resizing it whereas Redim Preserve preserves the previous contents.

```
Dim x () As Number
Redim x (2) 'Now x is Array (0, 0)
x (2) = 20 'Now x is Array (0, 20)
Redim x (3) 'Now x is Array (0, 0, 0)
x (3) = 30 'Now x is Array (0, 0, 30)
```

```
Redim Preserve x (4) 'Now x is Array (0, 0, 30, 0)
formula = "finished"
```

Default Values for Simple Types (Basic syntax)

An uninitialized variable will have the default value for its type. In general, it is not a good programming practice to rely on the default values of types. For example, initialize all local variables in your formula, initialize all global variables in a formula placed in the Report Header, and initialize all shared variables in a formula placed in the Report Header of the main report.

When an array is resized using the Redim keyword, the entries are filled with default values for the type.

Default values

Number

0

Currency

CCur (0)

String

"" 'The empty string

Date

CDate (0, 0, 0) 'The null Date value

Time

The null Time value. Value held by an uninitialized Time variable.

DateTime

The null DateTime value. Value held by an uninitialized DateTime variable.

NOTE: It is not recommended that your formulas rely on the values of uninitialized range or array variables.

Automatic Type Conversions (Basic syntax)

Generally, in Crystal Reports, values of one type cannot be used where values of another type are expected without explicitly supplying a type conversion function. For example:

```
Dim postalCode as String
'Error- assigning a Number value to a String variable
postalCode = 10025
'OK- use the type conversion function CStr
'to create "10025"
postalCode = CStr (10025, 0)
```

However, there are a few conversions that are made automatically:

- Number to Currency
- Date to DateTime
- Simple type to Range value of the same underlying simple type

For example, the following assignments are correct:

```
Dim cost As Currency
'Same as: cost = CCur (10)
cost = 10
Dim orderDate As DateTime
'Same as: orderDate = CDateTime (1999, 9, 23, 0, 0, 0)
orderDate = CDate (1999, 9, 23)
Dim aRange As Number Range
'Same as: aRange = 20 To 20
aRange = 20
Dim aRangeArray () As Number Range
'Same as :
'aRangeArray = Array (10 To 10, 20 To 25, 2 To 2)
aRangeArray = Array (10, 20 To 25, 2)
```

The opposite conversions are not allowed. For example:

```
Dim num As Number
num = 5 + CCur (10) 'Error
'OK- convert to Number type using the CDbl function
num = CDbl (5 + CCur (10))
```

5 is converted to CCur (5) and added to CCur (10) to make CCur (15). However, this Currency value cannot be automatically assigned to the Number variable num since automatic conversions from Currency to Number are not allowed. Similarly, functions accepting a Currency argument can be supplied a Number argument instead, and the Number argument will be converted to a Currency, whereas functions accepting a Number argument cannot be supplied a Currency argument without first explicitly converting the Currency to a Number using CDbl.

Functions (Basic syntax)

When using a function in a formula, type the name of the function and supply the arguments required. For example, the Len function requires a String argument and computes the length of the string.

```
Dim x As String
x = "hello"
formula = Len (x) 'The formula returns the Number 5
```

Supplying arguments of the incorrect type required by the function produces an error. For example, calling Len (3) would produce an error since Len does not accept a Number argument. Functions sometimes can accept different numbers of arguments or types of arguments. For example, the CDate function could accept a single String argument to form a Date value or 3 Number values holding the year, month and day respectively and form a Date value from them. See Date, Time, and DateTime

(Basic syntax).

Example with the Mid function

```
Dim x as String
x = "hello"
'Start at position 2, go to the end of the string
formula = Mid (x, 2) 'formula is now "ello"
'Start at position 2, extract 1 character
formula = Mid (x, 2, 1) 'formula is now "e"
```

The classes of functions are: Math, Summary, Financial, String, Date/Time, Date Range, Array, Type Conversion, Programming Shortcuts, Evaluation Time, Print State, Document Properties and Additional Functions. There are also some functions specific to conditional formatting formulas.

Functions similar to visual basic (Basic syntax)

The Math, Financial, String, Date/Time, Type Conversion, and Programming Shortcuts groups consist mainly of functions that are familiar to Visual Basic users. Most of the functions are intended to work in the same way as the Visual Basic function of the same name.

Sometimes functions will have more overloads than are available in Visual Basic. For example, the CDate function supports the Visual Basic overload of creating a Date value from a String value, such as CDate ("Sept 18, 1999") but it also supports an overload of creating a Date value by supplying the year, month and day as Number arguments e.g. CDate (1999, 9, 18). The overloads are indicated in the Functions tree.

Some functions that are supported by Basic syntax are not listed in the Basic syntax Functions tree. This is because they are equivalent to Basic syntax functions that are already listed in the tree. For example, the Len function is the normal Basic syntax and Visual Basic function for finding the length of a string. However, the Length function is listed in the Functions tree and works the same as the Len function. Length is the traditional Crystal syntax function for this action and it is included for the convenience of Crystal syntax users migrating formulas to Basic syntax.

Summary functions (Basic syntax)

The Summary function group provides functions for creating summary fields such as:

```
Sum({Orders.Order Amount}, {Orders.Ship Via})
```

Summary fields are normally created using the Insert Summary or Insert Grand Total dialogs. They then appear in the Report Fields tree, and can be used in a formula by double clicking there. However, they do not need to be created in this way. You can create a summary field exclusively for use by your formula by filling in the arguments to one of the functions in the Summary functions section. However, any groups that refer to summary fields must already exist in the report.

Date ranges (Basic syntax)

For additional information refer to Range data types (Basic syntax).Date ranges produced by these functions depend on the current date. For example, if today's date is September 18, 1999, then LastFullMonth is the Date Range value:

```
CDate(#Aug 1, 1999#) To CDate(#Aug 31, 1999#)
```

This functionality is often useful, but if you want to determine a date range based on a database field such as {Orders.Order Date}? The Date/Time functions can be used instead.

Example

```
Dim d As Date
d = CDate ({Orders.Order Date})
Dim dr As Date Range
dr = DateSerial (Year(d), Month(d) - 1, 1) To _
DateSerial (Year(d), Month(d), 1 - 1)
'At this point dr is the Date Range value holding
'the last full month before {Orders.Order Date}
```

The DateSerial function makes this easy because you don't have to worry about special cases. It never lets you create an invalid date. For example, DateSerial (1999, 1 - 1, 1) is December 1, 1998. Note that in the above example, {Orders.Order Date} is actually a DateTime field and so the CDate function is used to convert it to a date by truncating the time part.

Array functions (Basic syntax)

The array functions compute summaries of an array's elements. For example, the Sum function when applied to an array returns the sum of the elements of the array.

Example

The following formula returns 100:

```
formula = Sum (Array (10, 20, 30, 40))
```

Evaluation time functions (Basic syntax)

These are the report specific functions: BeforeReadingRecords, WhileReadingRecords, WhilePrintingRecords and EvaluateAfter. You can use these functions to guide LABWORKS Report Designer as to when your formula should be evaluated.

Should the formula be evaluated before retrieving the records from the database, while reading the records from the database but before the records have been grouped, sorted and summarized, or while printing the report, when the records are grouped, sorted and summarized? In general, LABWORKS Report Designer sets an appropriate evaluation time for your formula, based on how much information the formula needs. For example, if a formula uses a database field, then it cannot be evaluated before the records are read from the database. However, you sometimes need to force a later evaluation time than normal to get the desired effect. See Using Global variables for an example.

Normally, the returned value of a function is used further in a formula. However, evaluation time functions are called to change the internal behavior of LABWORKS Report Designer and their return value is not used. They can be called by just placing their name in a separate statement, optionally preceded by the keyword Call.

```
WhilePrintingRecords
Call WhilePrintingRecords
```

Print state functions (Basic syntax)

These are once again reporting-specific functions. For example, the notation {Orders.Order Date} refers to the value of the field in the current record where PreviousValue

({Orders.Order Date}) refers to the value in the immediately preceding record. NextValue ({Orders.Order Date}) refers to the value in the next record. IsNull ({Orders.Order Date}) checks if the field's value is null.

Other examples are PageNumber and TotalPageCount. These can be used to access pagination information about your report.

Document properties functions (Basic syntax)

These are reporting specific functions that refer to the report document as a whole. For example, PrintDate and ReportTitle.

Additional functions (Basic syntax)

These functions are in User Function Libraries (UFLs). A UFL is a separate dynamic link library or Automation server that you create and LABWORKS Report Designer uses to add your own customized functions to the formula language. Writing a UFL is more involved than writing a formula using Basic or Crystal syntax. See LABWORKS Report Designer Developer's Help for more information.

NOTE: Using UFLs makes your reports less portable because you must distribute your UFL along with the report.

Conditional formatting functions (Basic syntax)

When writing a conditional formatting formula, you may want to use the additional functions that appear at the top of the Functions tree.

Example

If you wanted to format the {Customer.Last Year's Sales} field so that sales of more than $100,000 are printed in green and sales of less than $15,000 are printed in red and all else are printed in black.

```
Rem Conditional formatting example 1
If {Customer.Last Year's Sales} > 100000 Then
    formula = crGreen
ElseIf {Customer.Last Year's Sales} < 15000 Then
    formula = crRed
Else
    formula = crBlack
End If
```

Since this is a font color formatting function, the list of Color Constants appears in the Functions tree. This example uses three: crGreen, crRed and crBlack. You could have used the actual numeric values of the color constants instead. For example,
crRed is 255 and crGreen is 32768. However, the formula is easier to understand using the color constants. All constant functions in Basic syntax have the "cr" prefix.

Some formatting attributes do not use constant functions. For example, if you wanted to not print {Customer.Last Year's Sales} values if the sales were less than
$50,000, you could write the following conditional formatting formula for the suppress attribute:

```
Rem Conditional formatting example 2
```

```
If {Customer.Last Year's Sales} < 50000 Then
    formula = True 'suppress the value
Else
    formula = False 'do not suppress the value
End If
```

Or more simply:

```
Rem Conditional formatting example 3 - Rem equivalent to
example 2
    formula = {Customer.Last Year's Sales} < 50000
```

If the last year's sales are less than $50,000, then the expression

{Customer.Last Year's Sales} < 50000

is True, and so the formula returns True. On the other hand, if the last year's sales are greater than or equal to $50,000, then

{Customer.Last Year's Sales} < 50000

is False and so the formula returns False

General purpose conditional formatting functions (Basic syntax)

There are three general purpose conditional formatting functions. These functions are displayed at the top of the Functions tree when appropriate.

CurrentFieldValue – can be used for any formatting formula where you are formatting a field value.

DefaultAttribute – can be used for any formatting formula

GridRowColumnValue – can be used for formatting a formula where you are formatting a field value in a Cross-Tab or OLAP grid.

The CurrentFieldValue enables you to conditionally format Cross-Tab or OLAP grid cells based on their value. The GridRowColumnValue enables you to conditionally format the cells of a Cross-Tab or OLAP grid based on row or column headings values. These two functions are essential in some situations as there is no other way in the formula language to refer to these fields.

Example

If you wanted Cross-Tab cells to be suppressed if the values are less than 50,000:

```
Rem Conditional formatting example 4
formula = CurrentFieldValue < 50000
```

Operators (Basic syntax)

Arithmetic operators

The arithmetic operators are addition (+), subtraction (-), multiplication (*), division (/), integer division (\), modulus (Mod), negation (-) and exponentiation (^). Arithmetic operators are used to combine numbers, numeric variables, numeric fields, and numeric functions to get another number.

Examples

```
'Outstanding preferred stock as a percent of
'common stock
formula =  ({Financials.Preferred Stock} / _
{Financials.Common Stock}) * 100
'The square root of 9, Sqr(9), is 3.
'The formula returns 17.
formula =  7 + 2 * 3 - 2 + Sqr(6 + 3) * Len("up")
```

Order of precedence

When you create arithmetic expressions that involve several operators, the order that the program evaluates the various elements of the expression becomes important.

In general, the program evaluates expressions in the following order: from left to right – follows the rules of precedence from basic math.

Example

Multiplication and division are performed first from left to right. Then addition and subtraction are performed. For example, 5 + 10 * 3 = 5 + 30 = 35. You can change this order of precedence by using parentheses. For example, (5 + 10) * 3 = 15 * 3
= 45. If you are unsure of the order of precedence, it is a good idea to clarify your intentions with parentheses.

The arithmetic operators in LABWORKS Report Designer have the same order of precedence as in Visual Basic. Here is the list, from highest precedence to lowest:

Exponentiation (^)
Negation (-)
Multiplication and division (*, /)
Integer Division (\)
Modulus (Mod)
Addition and subtraction (+, -)

Comparison operators (Basic syntax)

The comparison operators are equal (=), not equal (<>), less than (<), less than or equal to (<=), greater than (>) and greater than or equal to (>=).

Comparison operators are usually used to compare operands for a condition in a control structure such as an If statement. Comparison operators as a group all have lower precedence than the arithmetic operators. For example, expressions like 2 + 3
< 2 * 9 are the same as (2 + 3) < (2*9).

Boolean operators (Basic syntax)

The Boolean operators are, in order of precedence from greatest to lowest: Not, And, Or, Xor, Eqv and Imp. Boolean operators are typically used with comparison operators to generate conditions for control structures. Boolean operators as a
group have lower precedence than the comparison operators. Thus for example, the expression 2 < 3 And 4 >= -1 is the same as (2 < 3) And (4 >= -1).

Null fields and how to use IsNull (Basic syntax)

The {Product.Color} field contains both basic colors such as "red" and "black" and more descriptive two word colors such as "steel satin" and "jewel green". Here is an example of a formula that writes out "basic" for the basic colors and "fancy" for the others.

```
If InStr({Product.Color}, " ") = 0 Then
    formula = "basic"
Else
    formula = "fancy"
End If
```

The function call to InStr searches the {Product.Color} string for a space. If it finds a space, it returns the position of the space, otherwise it returns 0. Since basic colors are only one word with no spaces, InStr will return 0 for them.

Dealing with null values (Basic syntax)

For some products, such as the Guardian Chain Lock, a color value was not recorded and so the {Product.Color} field has a null value in the database for that record. In general, when LABWORKS Report Designer encounters a null valued field in a formula, it immediately stops evaluating the formula and produces no value. Thus, the Guardian Chain Lock record does not have any word printed beside it. If you want to handle null field values in your formula, you must explicitly do so using one of the special functions designed for handling them: IsNull, PreviousIsNull or NextIsNull. Here is an example of how to fix the above example using IsNull:

```
If IsNull({Product.Color}) Or _
InStr({Product.Color}, " ") = 0 Then
   formula = "basic"
Else
    formula = "fancy"
End If
```

Relating to operators, when LABWORKS Report Designer evaluates the condition:

```
IsNull({Product.Color}) Or _
InStr({Product.Color}, " ") = 0
```

It first evaluates IsNull ({Product.Color)), and when it determines that this is True, it knows that the whole condition is True, and does not need to check whether

```
InStr({Product.Color}, " ") = 0
```

In other words, LABWORKS Report Designer will stop evaluating a Boolean expression when it can predict the results of the whole expression. In the following example, the formula guards against attempting to divide by zero in the case that denom is 0:

```
Dim num As Number, denom As Number
...
If denom <> 0 And num / denom > 5 Then
...
```

NOTE: Visual Basic does not support this technique, since all parts of a Boolean expression in Visual Basic are evaluated, even if not necessary.

Control Structures (Basic syntax)

Formulas without control structures execute each statement in the formula only once. When this happens the formula is evaluated. The statements are executed in a sequential fashion, from the first statement in the formula to the last. Control structures enable you to vary this rigid sequence. Depending upon which control structure you choose, you can skip over some of the statements or repeatedly evaluate some statements depending on certain conditions. Control structures are the primary means of expressing business logic and typical report formulas make extensive use of them.

Basic syntax supports many of the main control structures from Visual Basic with the same syntax. One of the advantages of the Basic language is it is easy to read block notation for control structures. This simplifies the writing and debugging of complex formulas.

If statements (Basic syntax)

The If statement is one of the most useful control structures. It enables you to evaluate a sequence of statements if a condition is true and evaluate a different sequence of statements if it is not true.

NOTE: When formatting with conditional formulas, always include the Else keyword; otherwise, values that don't meet the If condition may not retain their original format. To prevent this, use the DefaultAttribute function (If...Else formula = DefaultAttribute).

Example

A company plans to pay a bonus of 4 percent to its employees except for those who work in Sales who will receive 6 percent. The following formula using an If
statement would accomplish this:

```
Rem Multi-line If example 1
If {Employee.Dept} = "Sales" Then
    formula = {Employee.Salary} * 0.06
Else
    formula = {Employee.Salary} * 0.04
End If
```

In this example, if the condition {Employee.Dept} = "Sales" evaluates as true, then the

```
    formula = {Employee.Salary} * 0.06
```

statement is processed. Otherwise the statement following the Else, namely the

```
    formula = {Employee.Salary} * 0.04
```

is processed.

Suppose another company wants to give employees a 4% bonus, but with a minimum bonus of $1,000. The following example shows how. Notice that the Else clause is not included; it is optional, and not needed in this case.

```
Rem Multi-line If example 2
formula = {Employee.Salary} * 0.04
If formula < 1000 Then
    formula = 1000
End If
```

Now suppose that the previous company also wants a maximum bonus of $5,000. You now need to use an ElseIf clause. Notice that ElseIf is all one word. The following example has only one ElseIf

clause, but you can add as many as you need.

NOTE: There is a maximum of one Else clause per If statement.

The Else clause is executed if none of the If or ElseIf conditions are true.

```
Rem Multi-line If example 3
formula = {Employee.Salary} * 0.04
If formula < 1000 Then
    formula = 1000
ElseIf formula > 5000 Then
    formula = 5000
End If
```

Example

Suppose that a company wants to compute an estimate of the amount of tax an employee needs to pay and write a suitable message. Income below $8,000 is not taxed, income between $8,000 and $20,000 is taxed at 20%, income between
$20,000 and $35,000 is taxed at 29%, and income above $35,000 is taxed at 40%.

```
    Rem Multi-line If example 4
    Dim tax As Currency, income As Currency
    income = {Employee.Salary}
    Dim message As String
    If income < 8000 Then
        tax = 0
        message = "no"
    ElseIf income >= 8000 And income < 20000 Then
        message = "lowest"
        tax = (income - 8000)*0.20
    ElseIf income >= 20000 And income < 35000 Then
        message = "middle"
        tax = (20000 - 8000)*0.20 + (income - 20000)*0.29
    Else
        message = "highest"
        tax = (20000 - 8000)*0.20 + (35000 - 20000)*0.29 + _ (income -
        35000)*0.40
    End If
    Dim taxStr As String
    Rem use 2 decimal places
    Rem and use the comma as a thousands separator
    taxStr = CStr (tax, 2, ",")
        formula = "You are in the " & message & _ " tax bracket. "
        & _
        "Your estimated tax is " & taxStr & "."
```

Notice, the use of variables to simplify the logic of the computation. Also, notice that there are two statements that are executed when one of the conditions are met; one assigns the tax variable, and the other assigns the message variable. It is often useful to have multiple statements executed as a result of a condition.

Single-line and multi-line If statements (Basic syntax)

There are two kinds of If statement, the single-line if statement and the multi-line if statement. Starting on a new line after the first **Then** turns your If statement into a multi-line If statement. Otherwise it is a single-line If statement. The multi-line If statement always includes an End If whereas the single line If statement does not.

Because of the use of line-continuation characters, single-line If statements do not need to be on a single line. In general, it is preferable to use multi-line If statements since they have a clearer layout. However, for simple situations, the single-line If statement is sometimes used.

```
Rem Single-line If example 1
Rem Same result as multi-line If example 1
If {Employee.Dept} = "Sales" Then _
    formula = {Employee.Salary} * 0.06 _
Else _
    formula = {Employee.Salary} * 0.04
```

Here is an example showing various forms of single-line If statements:

```
Rem Single-line If example 2
Dim per As Number, extra As Boolean
per = 2 : extra = False
'An example with no Else clause
If {Employee.Dept} = "Sales" Then per = 10
'More than 1 statement in the Then or Else part can
'be included by separating them with colons
If {Employee.Dept} = "R&D" Then _
    per = 5 : extra = True _
Else _
    per = 3
```

Select statements (Basic syntax)

The Select statement is similar to an If statement. Sometimes however, you can write formulas that are clear and less repetitive using the Select statement. This example evaluates the {Customer.Fax} field to determine if the area code is for Washington state (206, 360, 509) or British Columbia, Canada (604, 250):

```
Rem Select example 1
Select Case Left ({Customer.Fax}, 3)
Case "604", "250"
    formula = "BC"
Case "206", "509", "360"
```

```
        formula = "WA"
    End Select
```

The expression right after the Select Case keywords is called the Select condition. In the above example it is Left ({Customer.Fax}[1 To 3]). The Select statement tries to find the first Case that matches the Select condition, and then executes the statements following it, up until the next Case.

```
    Rem Same effect as Select example 1
    Dim areaCode As String
    areaCode = Left ({Customer.Fax}, 3)
    If areaCode In Array ("604", "250") Then
        formula = "BC"
    ElseIf areaCode In Array ("206", "509", "360") Then
        formula = "WA"
    End If
```

Example

This formula groups the number of Oscar nominations a movie received into low, medium, high or extreme categories and in the process, shows some of the possibilities for the expression lists following the Case labels. Notice the optional Case Else clause. If none of the Case expression lists are matched by the preceding Case clauses, then the Case Else clause is matched. For example, in the following example, if {movie.NOM} is 11, then the formula returns "extreme".

```
    Rem Select example 2
    Select Case {movie.NOM}
    Case 1,2,3, Is < 1
        Rem Can have multiple statements in the
        Rem statement blocks
        formula = "low"
  Case 4 To 6, 7, 8, 9
        formula = "medium"
  Case 10
        formula = "high"
  Case Else
        formula = "extreme" End Select
```

For/Next loops (Basic syntax)

For/Next loops enable you to evaluate a sequence of statements multiple times. This is unlike the If and Select statements where the program passes through each statement at most once during the formula's evaluation. For/Next loops are best when you know in advance the number of times the statements need to be evaluated.

Sample 1

Suppose you want to reverse the {Customer.Customer Name} string. For example, "City Cyclists" becomes "stsilcyC ytiC".

```
    Rem Reverse a string version 1
    formula = ""
```

```
Dim strLen
strLen = Len ({Customer.Customer Name})
Dim i
For i = 1 To strLen
Dim charPos
charPos = strLen - i + 1
formula = formula & _
    Mid({Customer.Customer Name}, charPos, 1)
Next i
```

Let us examine how this formula works assuming that the current value of the field {Customer.Customer Name} is "Clean Air". The variable strLen is assigned to be the length of "Clean Air", namely 9. At this time it is also typed to be a Number variable.
The variable i is known as a **For counter variable** since its value changes with each iteration of the For loop. In other words, it is used to count the iterations of the loop. The For loop will iterate 9 times, during the first time, i is 1, then i is 2, then i is 3 and so on until finally i = 9. During the first iteration, the ninth character of {Customer.Customer Name} is appended to the empty special variable formula. As a result formula equals "r" after the first iteration. During the second iteration, the eighth character of {Customer.Customer Name} is appended to formula and so
formula equals "ri". This continues until after the ninth iteration, formula equals, "riA
naelC" which is the reversed string.

Sample 2

Here is a simpler version of the above formula, that uses a Step clause with a negative Step value of -1. For the "Clean Air" example, i is 9 for the first iteration, 8 for the second, 7 for the third and so on until it is 1 in the final iteration.

```
Rem Reverse a string version 2
formula = ""
Dim i
For i = Len ({Customer.Customer Name}) To 1 Step -1
    formula = formula + _
        Mid({Customer.Customer Name}, i, 1)
Next i
```

Sample 3

The simplest version is to use the built in function StrReverse:

```
Rem Reverse a string version 3
formula = StrReverse ({Customer.Customer Name})
```

The built in String functions in LABWORKS Report Designer 8.5 can handle many of the string processing applications which would traditionally be handled using a For/Next loop or some other kind of loop. However, For/Next loops provide the most flexibility and power in processing strings and arrays. This can be essential if the built-in functions do not cover your intended application

For/Next loop example (Basic syntax)

Here is a more detailed example of Crystal Reports' string processing capabilities. The Caesar cipher

is a simple code that is traditionally credited to Julius Caesar. In this code, each letter of a word is replaced by a letter five characters further in the alphabet. For example, "Jaws" becomes "Ofbx". Notice that "w" is replaced by "b". Since there are not 5 characters after "w" in the alphabet, it starts again from the beginning.

Here is a formula that implements applying the Caesar cipher to the field {Customer.Customer Name} in the Example database:

```
Rem The Caesar cipher
Dim inString 'The input string to encrypt inString =
{Customer.Customer Name}
Dim shift
shift = 5
formula = ""
Dim i
For i = 1 To Len(inString)
    Dim inC, outC
    inC = Mid(inString, i, 1)
    Dim isChar, isUCaseChar
    isChar = LCase(inC) In "a" To "z"
    isUCaseChar = isChar And (UCase (inC) = inC)
    inC = LCase(inC)
    If isChar Then
        Dim offset
        offset = (Asc(inC) + shift - Asc("a")) Mod _ (Asc("z") - Asc("a") + 1)
        outC = Chr(offset + Asc("a"))
        If isUCaseChar Then outC = UCase(outC)
                    Else


   outC = inC

    End If
      formula = formula & outC
      Next i
```

In the above example, there is a multi-line If statement nested within the statements block of the For/Next loop. This If statement is responsible for the precise details of shifting a single character. For example, letters are treated differently from punctuation and spaces. In particular, punctuation and spaces are not encoded. Control structures can be nested within other control structures and multiple statements can be included in the statement block of a control structure

Using the Exit For statement (Basic syntax)

You can exit from a For/Next loop by using the Exit For statement. The following example searches the Global array names for the name "Fred". If it finds the name, it returns the index of the name in the array. Otherwise it returns -1. For example, if the names array is:

Array ("Frank", "Helen", "Fred", "Linda")

Then the formula returns 3.

```
Global names () As String
'The names array has been initialized and filled
'in other formulas
Dim i
formula = -1
'The UBound function returns the size of its array
'argument
For i = 1 to UBound (names)
    If names (i) = "Fred" Then
        formula = i
        Exit For
    End If
Next i
```

Do Loops (Basic syntax)

Another looping mechanism is the Do loop. A Do loop can be used to execute a fixed block of statement an indefinite number of time.

Four different types of Do loops

Type of Do Loop	Explanation	Example
Do While ... Loop	The Do While ... Loop evaluates the condition, and if the condition is true, then it evaluates the statements following the condition. When it has finished doing this, it evaluates the condition again and if the condition is true, it evaluates the statements again. It continues repeating this process until the condition is false.	Do While condition statements Loop
Do Until ... Loop	The Do Until ... Loop is similar to the Do While ... Loop except it keeps evaluating the statements until the condition is true rather than while it is true.	Do Until condition statements Loop
Do ... Loop While	The Do ... Loop While evaluates the statements only once. It then evaluates the condition, and if the condition is true, evaluates the statements again. This process continues until the condition is false.	Do statements Loop While condition
Do ... Loop Until	Similar to Do ... Loop While except that it evaluates the statements until the condition is true.	Do statements Loop Until condition

The Do loops support an Exit Do statement to jump immediately out of the loop. The Exit Do statement is similar to the Exit For in For/Next loops.

Do While ... Loop formula example (Basic syntax)

The following example searches for the first occurrence of a digit in an input string. If a digit is found, it returns its position, otherwise it returns -1. In this case, the input string is set explicitly to a string constant. If preferred, it could be set to be equal to a String type database field instead. For example, for the input String, "The 7 Dwarves", the formula returns 5, which is the position of the digit 7.

```
Dim inString
inString = "The 7 Dwarves" Dim i, strLen
i = 1
strLen = Len (inString)
formula = -1
Do While i <= strLen And formula = -1
    Dim c As String
    c = Mid (inString, i, 1)
    If IsNumeric (c) Then formula = i
    i = i + 1
Loop
```

While loops (Basic syntax)

The While loop is similar to the Do While ... Loop except that it does not support an Exit statement. It uses While ... Wend instead of Do While ... Loop as its syntax.

```
While condition
    statements
Wend
```

Safety mechanism for loops (Basic syntax)

There is a safety mechanism to prevent report processing from hanging due to an infinite loop. Any one evaluation of a formula can have at most 30,000 loop condition evaluations per formula evaluation. For example:

```
Dim i
i = 1
Do While i <= 100000
    If i > {movie.STARS} Then Exit Do
    i = i + 1
Loop
formula = 20
```

If {movie.STARS} is greater than 30,000 then the loop condition (i <= 100000) will be evaluated more than the maximum number of times and an error message is displayed. Otherwise, the loop is OK.

The safety mechanism applies on a per formula base, not for each individual loop. For

example:

```
Dim i
i = 1
For i = 1 To 10000
    formula = Sin (i)
Next i
Do While i <= 25000
    i = i + 1
Loop
```

The above formula also triggers the safety mechanism since the 30,000 refers to the total number of loop condition evaluations in the formula and this formula will have 10001 + 25001 such evaluations.

Limitations (Basic syntax)

For reference purposes, here are the sizing limitations of the formula language:

- The maximum length of a String constant, a String value held by a String variable, a String value returned by a function or a String element of a String array is 254 bytes.
- The maximum size of an array is 1000 elements.
- The maximum number of arguments to a function is 1000. (This applies to functions that can have an indefinite number of arguments such as Choose).
- The maximum length of the text of a formula is 64K.
- The maximum number of loop condition evaluations per evaluation of a formula is 30,000. (See safety mechanisms for loops for the precise meaning of this).
- Date-time functions modeled on Visual Basic accept dates from year 100 to year 9999. Traditional LABWORKS Report Designer functions accept dates from year 1 to year 9999.

Creating Formulas with Crystal Syntax

Creating Formulas with Crystal Syntax

This section provides you with an overview of various parts of a formula. You learn about Crystal syntax and the techniques you can use when creating a formula.

If you are new to Crystal syntax, begin with these topics: Crystal

syntax overview

Crystal syntax fundamentals

When creating formulas, you have the option of using either Crystal or Basic syntax. Syntax rules are used to create a correct formula. Almost any formula written with one syntax can be written with the other. Reports can contain formulas that use Basic syntax as well as formulas that use Crystal syntax.

Crystal Syntax Fundamentals

The result of a formula, or the value that is printed when the formula is placed in a report, is called the value returned by the formula. Every formula in LABWORKS Report Designer must return a value. For example, here is a simple Crystal syntax formula that returns a value of 10:

```
10
```

The value returned by a formula can be one of the seven simple data types supported. These are Number, Currency, String, Boolean, Date, Time, and DateTime.

LABWORKS Report Designer also supports range types and array types, but these cannot be returned by a formula.

For example, suppose a company has a shipping policy in which orders over $1,000 are insured, but orders below that amount are not insured:

```
//A formula that returns a String value
If {Orders.Order Amount} >= 1000 Then
    "Insured shipping"
Else
        "Regular shipping"
```

The text following the two forward slashes is a comment for someone reading this formula and is ignored by the Crystal syntax compiler.

The formula returns the text string value "Insured shipping" if the value of the database field {Orders.Order Amount} is greater than or equal to 1000; otherwise, it returns the text string value "Regular Shipping" otherwise.

Crystal syntax is not case-sensitive

For example, the keyword Then could equivalently be typed in as then or THEN. This is true of all variable names, functions, and keywords used in a Crystal syntax formula.

The only exception to this rule is for strings. The string "Hello" is not the same as the string "hello".

Comments (Crystal syntax)

Formula comments are notes included with a formula to explain its design and operation. Comments do not print and they do not affect the formula, but they appear in the Formula Editor. You can use comments to explain the purpose of a formula or explain the steps involved in writing it.

Comments begin with two forward slashes (//) and are followed by the text of the comment. Everything that follows the slashes on the same line is treated as being part of the comment:

```
//This formula returns the string "Hello"
//This is another comment
"Hello" //Comments can be added at the end of a line
```

```
//Comments can occur after the formula text
```

Fields (Crystal syntax)

Many of the fields used in the construction of a report can be referred to in your formulas. For example, database, parameter, running total, SQL expression, summary, and group name fields can all be used in a formula. You can also refer to other formula fields in your formula.

The easiest way to insert a field into your report is to double-click a field's name in the Report Fields tree. This ensures that the correct syntax for the field is used.

How fields appear in formulas

Database, parameter, formula, running total, and SQL expression fields have their names surrounded by braces.

Database field names are taken from the database. For example:
database field: {Employee.Last Name}

Parameter, formula, running total, and SQL expression field names are specified when the fields are created.

- Parameter fields also include a question mark: {?my parameter field}.
- Formula fields include an at sign: {@another formula}.
- Running totals fields include a pound sign: {#my running total}.
- SQL expression fields include a percent sign: {%my SQL expression}.

Summary and group name fields look like function calls. However, they are really shorthand

notation for a report field.

- Sum summary field: Sum({Orders.Order Amount}, {Orders.Ship Via}).
- Group name field: groupName({Orders.Ship Via}).

Expressions (Crystal syntax)

An expression is any combination of keywords, operators, functions, and constant values that result in a value of a given type. For example:

```
//An expression that evaluates to the Number value 25
10 + 20 - 5
//An expression that evaluates to the String value
//"This is a string."
"This is a string."
```

A Crystal syntax formula consists of a sequence of expressions. The value of the final expression is the value returned by the formula and what gets printed. Each expression must be separated from the previous expression by a semicolon (;).

Multiple expressions (Crystal syntax)

Typically, each expression takes one line, but you can continue an expression onto the next line if you need more space.

The formula below consists of five expressions. It returns the Number value 25 since that is the value of the last expression in the formula.

Example

```
//Expressions example
//The first expression. Its value is the Number
//value 30
10 + 20;
//The second expression. Its value is the String
//"Hello World". It takes up two lines.
"Hello " +
"World";
//The third expression. Its value is of Number type
{Orders Detail.Quantity} * 2 - 5;
//The fourth expression. Its value is of String type
If {Orders Detail.Quantity} > 1 Then
    multiple units"
 Else
        "one unit";
//The fifth and final expression. Its value is the
//Number value 25
20 + 5
```

Placing a semicolon after the last expression in the formula is also allowed, but optional. For example, the above formula could have ended with:

```
20 + 5;
```

Some of the sample formulas in the preceding section do not have semicolons. This is because they consist of a single expression, and a semicolon is optional after the last expression. Many formulas in Crystal syntax can be written as a single expression.

Notice that there is no semicolon after the "multiple units" String. In fact, if you put a semicolon there, the program will report an error. This is because a semicolon separates expressions, and the

```
Else
        "one unit";
```

is not a separate expression. It does not stand alone apart from the If. In fact, it is an integral part of the If expression since it describes the value that the If expression will return under certain circumstances.

The example is not a practical example because the first four expressions in the formula did

not have any effect on the last expression

How earlier expressions affect later expressions (Crystal syntax)

The fact that a Crystal syntax formula is a sequence of expressions whose result is the value of the final expression is the most important concept in understanding Crystal syntax. This expression-based syntax allows you to write very short formulas with a lot of functionality.

Example

```
//First expression. It declares the Number variable x
//and then returns the value of an uninitialized
//Number variable, which is 0.
NumberVar x;
//Second expression. It assigns the value of 30 to x,
//and returns 30.
x := 30
```

The above formula would give an error if the first expression were omitted. This is because the second expression refers to the Number variable x, and the program needs to have x declared before it understands expressions involving it.

In general, you use variables to get the earlier expressions in a formula to affect the final expression. See Variables (Crystal syntax) for more information

Using the If expression (Crystal syntax)

The If expression is one of the most widely used features of Crystal syntax. It also provides insight into the nature of expressions. Consider the earlier If expression as a separate formula. Notice that because this formula is a single expression, it does not need a semicolon:

```
If {Orders Detail.Quantity} > 1 Then
   "multiple units"
Else
    "one unit"
```

Suppose you wanted to modify this formula so that it either prints "multiple units" or the number 1.//An erroneous formula

```
If {Orders Detail.Quantity} > 1 Then
   "multiple units"
Else
    1
```

This formula will result in an error. This is because the values in this expression are different types: "multiple units" is a String value and 1 is a Number value. LABWORKS Report Designer requires that the value of an expression always be of a single type.

This example can be corrected by using the CStr function to convert the Number 1 to a String value. For example, the Number 1 is converted to the string "1" by calling CStr (1, 0).

```
//A correct formula
If {Orders Detail.Quantity} > 1 Then
```

```
    "multiple units"
Else
    CStr (1, 0) //Use 0 decimals
```

Assignment (Crystal syntax)

The assignment operator is a colon followed by an equals sign (:=).

Example

```
//Assign the Number value of 10 to the variable x
x := 10;
//Assign the String value of "hello" to the
//variable named greeting
greeting := "hello";
```

The equality operator (=) is used to check when two values are equal. A common error is to use the equality operator when the assignment operator is actually intended. This can generate a mysterious error message or no error message at all since it is often syntactically correct to use the equality operator. For example:

```
greeting = "hello";
```

The above formula checks if the value held by the variable greeting is equal to the value "hello". If it is, then the expression's value is True, and if is not then the expression's value is False. In any case, it is a perfectly correct Crystal syntax expression (assuming that the greeting is a String variable).

Simple Data Types (Crystal syntax)

The simple data types in LABWORKS Report Designer are Number, Currency, String, Boolean, Date, Time, and DateTime.

Number

Enter numbers without any comma separators or currency symbols (generally you would want to have formatted numbers appearing as the result of a formula and not in the formula itself).

Example

```
10000
-20
1.23
```

Currency (Crystal syntax)

Use the dollar sign ($) to create a Currency value. Example

```
$10000
-$20
$1.23
```

You can also use the CCur function. The initial C in CCur stands for convert and it can be used to convert Number values to Currency values:

CCur (10000)

CCur (-20)

CCur (1.23)

String (Crystal syntax)

Strings are used to hold text. The text must be placed between double quotation marks (") or apostrophes (') and cannot be split between lines. If you want to include double quotation marks in a string delimited by double quotation marks, use

two consecutive double quotation marks. Similarly, if you want to include an apostrophe in a string delimited by apostrophes, use two consecutive apostrophes.

Example

"This is a string."

"123"

"The word ""hello"" is quoted."

'This is also a string.'

'123'

'Last Year''s Sales'

If you use double quotes for the left side of the string, you must use double quotes on the right side. Similarly for apostrophes. The following example is incorrect:

'Not a valid string."

You can extract individual elements or substrings from a string by specifying the character position or a range of character positions. Negative values are allowed; they specify the position starting from the end of the string.

"hello" [2] //Equal to "e"

"hello" [-5] //Equal to "h"

"604-555-1234" [1 to 3] //Equal to "604"

"abcdef" [-3 to -1] //Equal to "def"

You can also extract substrings from a string using the Left, Right and Mid functions.

Boolean (Crystal syntax)

The valid Boolean values are:

True

False

Yes can be used instead of True and No instead of False.

Date, Time, and DateTime (Crystal syntax)

The DateTime type can hold date-times, dates only, or times only and thus is rather versatile. The Date type holds dates only and the Time type holds times only. The Date and Time types are more efficient than the DateTime type, and so can be used in situations where the added

functionality and flexibility of the DateTime type is not needed.

You can create DateTime values directly using the date-time literal construction, which is formed by typing in the date-time between two pound (#) signs. Many different formats are supported.

NOTE: Date-time literals cannot be split between lines.

Examples

```
#8/6/1976 1:20 am#
#August 6, 1976#
#6 Aug 1976 13:20:19#
#6 Aug 1976 1:30:15 pm#
#8/6/1976#
#10:20 am#
```

Even though #10:20 am# looks like it could have the Time type and #8/6/1976# looks like it could have the Date type, they do not. They both have the DateTime type, as do all date-time literals. For example, you can think of #10:20 am# as a DateTime value with a null date part. To convert it to the Time type use CTime (#10:20 am#).

Instead of using date-time literals, you can use CDateTime to convert a String to a DateTime. For example,

```
CDateTime ("8/6/1976 1:20 am")
CDateTime ("10:20 am")
```

However, there is one key difference between using date-time literals and the above usage of CDateTime. Date-time literals always use U.S. English date formats rather than settings from the locale of the particular computer on which LABWORKS Report Designer is running. Thus, the date-time literal examples above would work on all computers. On the other hand, on a French system, you could use constructions like:

```
CDateTime ("22 aout 1997") //Same as #Aug 22, 1997#
```

Date values can be constructed with CDate and Time values with CTime:

```
CDate ("Aug 6, 1969")
CDate (1969, 8, 6) //Specify the year, month, day
//Converts the DateTime argument to a Date
CDate (#Aug 6, 1969#)
CTime ("10:30 am")
CTime (10, 30, 0) //Specify the hour, minute, second
CTime (#10:30 am#)
```

Range Data Types (Crystal syntax)

Ranges are designed to handle a spectrum of values. Range types are available for all the simple types except for Boolean. That is: Number Range, Currency Range, String Range, Date Range, Time Range and DateTime Range. You can generate ranges using the To, _To, To_, _To_, UpTo, UpTo_, UpFrom, and UpFrom_ keywords. In general, To is used for ranges with 2 endpoints, and UpTo and UpFrom are used for open ended ranges (only one endpoint). The underscores are used to indicate whether or not the endpoints are in the range.

Examples of Number Range values

The range of numbers from 2 to 5 including both 2 and 5

```
2 To 5
```

The range of numbers from 2 to 5, not including 2 but including 5

```
2 _To 5
```

All numbers less than or equal to 5

```
UpTo 5
```

All number less than 5

```
UpTo_ 5
```

Examples of DateTime Range values:

```
#Jan 5, 1999# To #Dec 12, 2000#
UpFrom #Jan 1, 2000#
```

Using ranges in formulas (Crystal syntax)

There are twenty-seven functions in LABWORKS Report Designer that specify date ranges. For example, the function LastFullMonth specifies a range of date values that includes all dates from the first to last day of the previous month. So if today's date is September 15, 1999 then LastFullMonth is the same as the range value CDate (#Aug 1, 1999#) To CDate (#Aug 31, 1999#).

Ranges are often used with If or Select expressions. The following example computes student letter grades based on their test scores. Scores greater than or equal to 90 receive an "A", scores from 80 to 90, not including 90, receive a "B" and so on.

```
//Compute student letter grades Select
{Student.Test Scores}
Case UpFrom 90 :
    "A"
Case 80 To_ 90 :
    “B"
Case 70 To_ 80 :
    "C"
Case 60 To_ 70 :
    "D"
Default :
    "F";
```

The above example uses the Select expression which is discussed in more detail in the control structures section. You can check if a value is in a range by using the In operator. For example:

```
5 In 2 To 10; //True
5 In 2 To_ 5; //False
5 In 2 To 5; //True
```

The Maximum and Minimum functions can be used to find the endpoints of a range:

Maximum (2 To 10) //Returns 10

Array Data Types (Crystal syntax)

Arrays in LABWORKS Report Designer are ordered lists of values that are all of the same type. These values are known as the array's elements. The elements of an array can be any simple type or range type. Arrays can be created using square brackets ([ ]).

Examples

An array of 3 Number values. The first element is 10, the second is 5 and the third is 20.

[10, 5, 20]

An array of 7 String values:

["Sun", "Mon", "Tue", "Wed", "Th", "Fri", "Sat"]

An array of 2 DateTime Range values:

[#Jan 1, 1998# To #Jan 31, 1998#,
#Feb 1, 1999# To #Feb 28, 1999#]

You can extract individual elements out of an array using square brackets containing the index of the element you want. This is called subscripting the array:

[10, 5, 20] [2] //Equal to 5

Number ranges can also be used to subscript arrays. The result is another array. For example:

[10, 5, 20] [2 To 3] //Equal to [5, 20]

Arrays are most useful when used with variables and so will be discussed in more detail in the Variables (Crystal syntax) section. Using variables, you can change the individual elements of an array and resize the array to accommodate more
elements. For example, you can accumulate database field values into a global array variable in a detail level formula, and then use a formula in a group footer to perform a calculation based on those values. This enables you to perform a wide variety of customized summary operations.

Variables (Crystal syntax)

This section describes the key components of variables and shows you how to create variables and assign values to them.

Variables overview

A variable represents a specific data item, or value, and acts as a placeholder for that value. When a formula encounters a variable, the formula searches for the value of the variable and uses it in the formula. Unlike a constant value, which is fixed and unchanging, a variable can be repeatedly assigned different values. You assign a value to a variable and the variable maintains the value until you later assign a new value. Because of this flexibility, it is necessary for you to declare
variables before you use them so that LABWORKS Report Designer is aware of them and understands how you intend to use them.

Example

If you wanted to report on customers by area code, you could create a variable that extracts the area code from a customer fax number. The following is an example of a variable called

areaCode:

```
Local StringVar areaCode;
areaCode := {Customer.Fax} [1 To 3];
```

The first line of the variable example is the variable declaration; it gives the variable a name and type. The database field {Customer.Fax} is a String field and [1 To 3] extracts the first 3 characters from its current value. The variable areaCode is then assigned this value.

Variable declarations (Crystal syntax)

Before using a variable in a formula, you must declare it.

A variable can hold values of a given type. The allowed types are the seven simple types (Number, Currency, String, Boolean, Date, Time and DateTime), the six range types (Number Range, Currency Range, String Range, Date Range, Time Range and DateTime Range) and variables that hold arrays of the previously mentioned types. This gives a total of 26 different types that a variable can have.

When you declare a variable, you also specify its name. A variable cannot have the same name as any function, operator or other keyword that is valid for Crystal syntax. For example, your variable cannot be named Sin, Mod or If because Sin is a built in function, Mod is a built in operator and If is a built in keyword. When typing formulas in the formula editor, the names of the built in functions, operators and other keywords are highlighted in a different color and so it is easy to check if the variable name conflicts.

Once a variable is declared, it can be used in the formula. For example, you might want to assign it an initial value:

```
Local NumberVar x; //Declare x to be a Number variable
x := 10; //Assign the value of 10 to x
```

The keyword for declaring the Number variable has a Var at the end. This is true for all the variable types in Crystal syntax.

A variable can only hold values of one type. For example, if a variable holds a Number value, you cannot later use it to hold a String.

Example

```
Local StringVar y;
y := "hello";
//OK- the Length function expects a String argument
Length (y);
//Error- y can only hold String values
y := #Jan 5, 1993#;
//Error- y can only hold String values
y := ["a", "bb", "ccc"];
//Error- the Sin function expects a Number argument
Sin (y);
```

You can combine declaring a variable and assigning it a value in a single expression. For example:

```
Local NumberVar x := 10 + 20;
Local StringVar y := "Hello" + " " + "World";
```

```
Local DateVar z := CDate (#Sept 20, 1999#);
Local NumberVar Range gradeA := 90 To 100;
```

This is a good practice because it is more efficient and helps prevent the common mistake of having incorrectly initialized variables.

Here are some more examples of declaring and initializing range variables:

```
Local NumberVar Range gradeA;
Local DateVar Range quarter;
gradeA := 90 To 100;
quarter := CDate (1999, 10, 1) To CDate (1999, 12, 31);
```

Variable scope (Crystal syntax)

Variable scopes are used to define the degree to which variables in one formula are made available to other formulas. There are three levels of scope in Crystal Reports: local, global and shared. Every variable has a scope, and this scope is specified when the variable is declared.

Local Variables

Variables with local scope, also known as local variables, are declared using the Local keyword followed by the type name (with the Var suffix) followed by the variable name as in the above examples.

Local variables are restricted to a single formula and a single evaluation of that formula. This means that you cannot access the value of a local variable in one formula from a different formula.

Example

```
//Formula A
Local NumberVar
x; x := 10;
//Formula B
EvaluateAfter ({@Formula A})
Local NumberVar x;
x := x + 1;
```

The function call EvaluateAfter ({@Formula A}) ensures that Formula B will be evaluated after Formula A is evaluated. Formula A returns a value of 10 and Formula B returns a value of 1. Formula B does not have access to Formula A's x and thus cannot use the value of 10 and add 1; instead, it uses the default value for the uninitialized local variable x found in Formula B, which is 0, and adds 1 to it to get 1.

You can also create local variables with the same name but different types in different formulas. For example, the type declarations in formulas A and B do not conflict with:

```
//Formula C
Local StringVar x := "hello";
```

Local variables are the most efficient of the three scopes. They also do not interfere with one

another in different formulas. For these reasons, it is best to declare variables to be local whenever possible.

Global variables (Crystal syntax)

Global variables use the same memory block to store a value throughout the main report. This value is then available to all formulas that declare the variable, except for those in subreports. You declare a global variable as in the following example:

```
Global StringVar y;
```

You can also omit the Global keyword which creates a Global variable by default:

```
StringVar y; //Same as: Global StringVar y;
```

However, even though global variables are easy to declare, it is recommended that you use them only when local variables do not suffice.

Since global variables share their values throughout the main report, you cannot declare a global variable in one formula with one type and then declare a global variable with the same name in a different formula with a different type.

Example

```
//Formula A
Global DateVar
z;
z := CDate (1999, 9, 18)
//Formula B
NumberVar
z; z := 20
```

In this case, if you enter and save Formula A first, LABWORKS Report Designer will return an error when you check or try to save Formula B. This is because the declaration of the Global variable z as a Number conflicts with its earlier declaration in Formula A as a Date.

Using global variables

Global variables are often used to perform complex calculations where the results of a formula depend upon the grouping and page layout of the actual printed report. This is accomplished by creating several formulas, placing them in different sections of the report, and having the different formulas interact via global variables.

Example

```
//Formula C
Global NumberVar x;
x := 10;
//Formula D
//Call the function WhileReadingRecords
WhileReadingRecords;
Global NumberVar x;
x := x + 1
```

If Formula C is placed in the Report Header and then Formula D is placed in a detail section,

Formula C will be evaluated before Formula D. Formula C will be evaluated once and then Formula D will be evaluated for each record appearing in the detail section. Formula C returns 10. For the first detail record, Formula D returns 11. This is because the value 10 of x is retained from when it was set by Formula C. Formula D then adds 1 to this value, setting x to 11 and then returns 11. For the second detail record, formula D return 12, adding 1 to the previously retained value of x which was 11. This process continues for the remaining detail records.

The call to WhileReadingRecords tells LABWORKS Report Designer to re-evaluate Formula D as it reads in each record of the report. Otherwise, since the formula does not contain any database fields, the program evaluates it only once before reading the records from the database. The formula will then return the value 11 instead of 11, 12, 13, ... as the successive records are processed.

If the expression x := x + 1 is replaced by x := x + {Orders Detail.Quantity}, you create the effect of a running total based on {Orders Detail.Quantity}, although it is one starting at 10 rather than 0 because of Formula C. In this case, you can omit the call to WhileReadingRecords, since it will automatically occur because the formula contains a database field.

Shared variables (Crystal syntax)

Shared variables use the same memory block to store the value of a variable throughout the main report and all of its subreports. Thus shared variables are even more general than global variables. To use a shared variable, declare it in a formula in the main report as in the following example:

```
Shared NumberVar x := 1000;
```

and declare it in a formula in the subreport as in the following example:

```
Shared NumberVar x;
```

In order to use shared variables, the variable must be declared and assigned a value before it can be passed between the main report and the subreport.

You can declare array variables by following the type name with the keyword Array.

Example

```
//Declare x to be a Global variable of
//Number Array type
Global NumberVar Array x := [10 , 20, 30];
//cost is a Global variable of Currency Array type
//It is automatically Global since the scope specifier
//(one of Local, Global or Shared) is omitted. CurrencyVar Array
cost := [$19.95, $79.50, $110.00,
$44.79, $223.99];
//payDays is a Global variable of Date Array type
Global DateVar Array payDays := [CDate(1999, 5, 15), CDate(1999,
5, 31)];
//y is a Shared variable of String Range Array type
Shared StringVar Range Array y := ["A" To "C", "H" To "J"];
//days is a Local variable of String Array type
Local StringVar Array days;
```

```
days := ["Sun", "Mon", "Tue", "Wed", "Th", "Fri", "Sat"];
```

Using array variables (Crystal syntax)

You can assign values to elements of an array and also use the values of the elements for other computations.

Example

```
StringVar Array x := ["hello", "bye", "again"];
x [2] := "once"; //Now x is ["hello", "once", "again"]
//The expression below would cause an error if not
//commented out since the array has size 3
//x [4] := "zap";
//The formula returns the String "HELLO"
UpperCase (x [1])
```

The Redim and Redim Preserve keywords can be used to resize an array if you want to add extra information to it. Redim erases the previous contents of the array first before resizing it whereas Redim Preserve preserves the previous contents.

```
Local NumberVar Array x;
Redim x [2]; //Now x is [0, 0]
x [2] := 20; //Now x is [0, 20]
Redim x [3]; //Now x is [0, 0, 0]
x [3] := 30; //Now x is [0, 0, 30]
Redim Preserve x [4]; //Now x is [0, 0, 30, 0]
“finished"

Local StringVar Array a;
Redim a [2];
//Assign a value to the first element of the array a
a[1] := "good";
a[2] := "bye";
//The & operator can be used to concatenate strings
a[1] & a[2] //The formula returns the String "goodbye"
```

Using Arrays with For loops

Arrays are commonly used with For loops. The following example creates and then uses the Array [10, 20, 30, ..., 100] using a For loop. See For loops (Crystal syntax) for more details.

```
Local NumberVar Array b;
Redim b[10];
Local NumberVar i;
For i := 1 To 10 Do
(
    b[i] := 10 * i
);
b [2] //The formula returns the Number 20
```

Default Values for the Simple Types (Crystal syntax)

An uninitialized variable will have the default value for its type. In general, it is not a good programming practice to rely on the default values of types. For example, initialize all local variables in your formula, initialize all global variables in a formula placed in the Report Header and initialize all shared variables in a formula placed in the Report Header of the main report.

When an array is resized using the Redim keyword, the entries are filled with default values for the type. It is useful to know about default values when using If and Select expressions.

Default values

Number

0

Currency

$0

String

"" //The empty string

Date

Date (0, 0, 0) //The null Date value

Time

The null Time value. Value held by an uninitialized Time variable.

DateTime

The null DateTime value. Value held by an uninitialized DateTime variable.

NOTE: It is not recommended that your formulas rely on the values of uninitialized range or array variables.

Automatic Type Conversions (Crystal syntax)

Generally, in Crystal Reports, values of one type cannot be used where values of another type are expected without explicitly supplying a type conversion function. For example:

```
Local StringVar postalCode;
//Error- assigning a Number value to a String
postalCode := 10025;
//OK - use the type conversion function CStr
//to create "10025"
postalCode := CStr (10025, 0);
```

However, a few conversions are made automatically:

- Number to Currency
- Date to DateTime
- Simple type to Range value of the same underlying simple type

For example, the following assignments are correct: Local
CurrencyVar cost;
//Same as: cost := $10
cost := 10;
Local DateTimeVar orderDate;
//Same as: orderDate := CDateTime (1999, 9, 23, 0, 0, 0)
orderDate := CDate (1999, 9, 23);
Local NumberVar Range aRange;
//Same as: aRange := 20 To 20
aRange := 20;
Local NumberVar Range Array aRangeArray;
//Same as : aRangeArray := [10 To 10, 20 To 25, 2 To 2]
aRangeArray := [10, 20 To 25, 2];

The opposite conversions are not allowed. For example:

Local NumberVar num;
num := 5 + $10; //Error
//OK- convert to Number type using the CDbl function num := CDbl
(5 + $10) //Could also use ToNumber
5 is converted to $5 and added to $10 to make $15. However, this Currency value cannot be automatically assigned to the Number variable num since automatic conversions from Currency to Number are not allowed. Similarly, functions accepting a Currency argument
can be supplied a Number argument instead, and the Number argument will be converted to a Currency, whereas functions accepting a Number argument cannot be supplied a Currency argument without first explicitly converting the Currency to a Number using CDbl.

Functions (Crystal syntax)

When using a function in a formula, type the name of the function and supply the arguments required. For example, the Length function requires a String argument and computes the length of the string.

Local StringVar x := "hello";
Length (x) //The formula returns the Number 5

Supplying arguments of the incorrect type required by the function produces an error. For example, calling Length (3) would produce an error since Length does not accept a Number argument. Functions sometimes can accept different numbers of arguments or types of arguments. For example, the CDate function which could accept a single String argument to form a Date value or 3 Number values holding the year, month and day respectively and form a Date value from them. See Date, Time, and DateTime (Crystal syntax).

Example with the Mid function Local
StringVar x := "hello";
Local StringVar y;
//Start at position 2, go to the end of the string
y := Mid (x, 2); //y is now "ello"
//Start at position 2, extract 1 character
y := Mid (x, 2, 1) //y is now "e"

The classes of functions are: Math, Summary, Financial, String, Date/Time, Date Range, Array, Type Conversion, Programming Shortcuts, Evaluation Time, Print State, Document Properties, and Additional Functions. There are also some functions specific to conditional formatting formulas.

Non reporting-specific functions (Crystal syntax)

The Math, Financial, String, Date/Time, Type Conversion, and Programming Shortcuts groups consist mainly of functions that are not specific to reporting, but may be found in any full featured programming environment. Many of the functions are similar in functionality to Visual Basic functions of the same or similar name.

NOTE: Some functions that are supported by Crystal syntax are not listed in the Crystal syntax Functions tree. This is because they are equivalent to Crystal syntax functions already listed in the tree.

For example, the Length function is the traditional Crystal syntax function for finding the length of a string. Crystal syntax also supports Len as a synonym. Len is the Visual Basic and Basic syntax function for this, its inclusion is for the convenience of Visual Basic, and Basic syntax users who want to write or modify Crystal syntax formulas.

Summary functions (Crystal syntax)

The Summary function group provides functions for creating summary fields such as:

```
Sum({Orders.Order Amount}, {Orders.Ship Via})
```

Summary fields are normally created using the Insert Summary or Insert Grand Total dialogs. They then appear in the Available Fields tree and can be used in a formula by double clicking there. However, they do not need to be created in this way. You can create a summary field exclusively for use by your formula by filling in the arguments to one of the functions in the Summary functions section appropriately. However, any groups in the report that the summary field refers to must already exist in the report.

Date ranges (Crystal syntax)

This category of functions was discussed in the section Range data types (Crystal syntax). One additional comment is that the Date ranges produced by these functions depend on the current date. For example, if today's date is September 18, 1999, then LastFullMonth is the Date Range value:

```
CDate(#Aug 1, 1999#) To CDate(#Aug 31, 1999#)
```

This functionality is often useful, but what if you want to determine a date range based on a database field such as {Orders.Order Date}? The Date/Time functions can be used instead. For example:

```
Local DateVar d := CDate ({Orders.Order Date});
Local DateVar Range dr;
dr := DateSerial (Year(d), Month(d) - 1, 1) To
        DateSerial (Year(d), Month(d), 1 - 1);
//At this point dr is the Date Range value holding
//the last full month before {Orders.Order Date}
```

The DateSerial function makes this easy because you don't have to worry about special cases. It never lets you create an invalid date. For example, DateSerial (1999, 1 - 1, 1) is

December 1, 1998.

In the above example, {Orders.Order Date} is actually a DateTime field and so the CDate function is used to convert it to a date by truncating the time part.

Array functions (Crystal syntax)

The array functions compute summaries of an array's elements. For example, the Sum function when applied to an array returns the sum of the elements of the array. For example, the following formula returns 100:

```
Sum ([10, 20, 30, 40])
```

Evaluation time functions (Crystal syntax)

These are the reporting specific functions BeforeReadingRecords, WhileReadingRecords, WhilePrintingRecords, and EvaluateAfter. You can use these functions to guide LABWORKS Report Designer as to when your formula should be evaluated.

Should the formula be evaluated before retrieving the records from the database, while reading the records from the database but before the records have been grouped, sorted and summarized, or while printing the report, when the records are grouped, sorted and summarized? In general, LABWORKS Report Designer sets an appropriate evaluation time for your formula, based on how much information the formula needs. For example, if a formula uses a database field, then it cannot be evaluated before the records are read from the database. However, you sometimes need to force a later evaluation time than normal to get the desired effect.

Print state functions (Crystal syntax)

These are once again reporting specific functions. For example, the notation {Orders.Order Date} is used to refers to the value of the field in the current record whereas Previous ({Orders.Order Date}) refers to the value in the immediately preceding record and Next ({Orders.Order Date}) in the next record. IsNull ({Orders.Order Date}) checks if the field's value is null.

Other examples are PageNumber and TotalPageCount that you can use to access pagination information about your report.

Document properties functions

These are specific functions, such as PrintDate and ReportTitle, that refer to the report document as a whole.

Additional functions (Crystal syntax)

These are functions in User Function Libraries or UFLs. A UFL is a separate dynamic link library or Automation server that you create and that LABWORKS Report Designer can use to add your own customized functions to the formula language. Writing a UFL is more involved than writing a formula using Basic syntax or Crystal syntax. See LABWORKS Report Designer Developer's Help for details.

NOTE: Using UFLs makes your reports less portable because you must distribute your UFL along with the report.

Conditional formatting functions (Crystal syntax)

When writing a conditional formatting formula, certain additional functions appear at the top of the Functions tree to help you with this. For example, you can format the {Customer.Last Year's Sales} field to print sales of more than $100,000 in green, sales of less than $15,000 in red, and all other sales in black.

Example

```
//Conditional formatting example 1
If {Customer.Last Year's Sales} > 100000 Then
    crGreen
Else If {Customer.Last Year's Sales} < 15000 Then
   crRed
Else
    crBlack
```

Since this is a font color formatting function, the list of Color Constants appears in the Functions Tree. This example uses three: crGreen, crRed and crBlack. You could have used the actual numeric values of the color constants instead. For example, crRed is 255 and crGreen is 32768. However, the formula is more understandable using the color constants. All constant functions in Crystal syntax can have the "cr" prefix.

Crystal syntax still supports constant functions from previous versions without the "cr" prefix. For example, you can use "Red" instead of "crRed". However, using the "cr" prefix organizes constant functions and is recommended.

Some formatting attributes do not use constant functions. For example, if you wanted to not print {Customer.Last Year's Sales} values if the sales were less than $50,000, you could write the following conditional formatting formula for the suppress attribute:

```
//Conditional formatting example 2
If {Customer.Last Year's Sales} < 50000 Then
    True //suppress the value
Else
    False //do not suppress the value
```

Or more simply:

```
//Conditional formatting example 3 -
//equivalent to example 2
{Customer.Last Year's Sales} < 50000
```

If the last year's sales are less than $50,000, then the expression

{Customer.Last Year's Sales} < 50000

is True, and so the formula returns True. On the other hand, if the last year's sales are greater than or equal to $50,000, then

{Customer.Last Year's Sales} < 50000

is False and so the formula returns False

General purpose conditional formatting functions (Crystal syntax)

There are three general purpose conditional formatting functions. These functions are displayed at the top of the Functions tree whenever appropriate.

- **CurrentFieldValue** – can be used for any formatting formula where you are formatting

a field value.

- **DefaultAttribute** – can be used for any formatting formula.
- **GridRowColumnValue** – can be used for any formatting formula where you are formatting a field value in a Cross-Tab or OLAP grid.

CurrentFieldValue allows you to conditionally format the cells of a Cross-Tab or OLAP grid based on their value, while GridRowColumnValue lets you conditionally format the cells of a Cross-Tab or OLAP grid based on the values of the row or column headings. These two functions are essential in this situation, since there is no other way in the formula language to refer to the values of these fields. For example, if you wanted Cross-Tab cells to be suppressed if the values are less than
50,000:

```
//Conditional formatting example 4
CurrentFieldValue < 50000
```

Operators (Crystal syntax)

Arithmetic operators

The arithmetic operators are addition (+), subtraction (-), multiplication (*), division (/), integer division (\), percent (%), modulus (Mod), negation (-) and exponentiation (^). Arithmetic operators are used to combine numbers, numeric variables, numeric fields and numeric functions to get another number.

Examples

```
//Outstanding preferred stock as a percent of
//common stock
{Financials.Preferred Stock} %
{Financials.Common Stock};
//The square root of 9, Sqr(9) is 3
//The formula returns 17
7 + 2 * 3 - 2 + Sqr(6 + 3) * Length("up");
```

Order of precedence (Crystal syntax)

When you create arithmetic expressions that involve several operators, the order that the program evaluates the various elements of the expression becomes important. In general, the program evaluates expressions from left to right. However, it also follows the rules of precedence from basic math.

Example

Multiplication and division are performed first from left to right and then addition and subtraction are performed.

For example, 5 + 10 * 3 = 5 + 30 = 35. You can change this order of precedence by using parentheses. For example, (5 + 10) * 3 = 15 * 3 = 45. If you are unsure of the order of precedence, it is a good idea to clarify your intentions with parentheses.

List of arithmetic operators, from highest precedence to lowest

- Exponentiation (^)

- Negation (-)
- Multiplication, division, and percent (*, /, %)
- Integer Division (\)
- Modulus (Mod)
- Addition and subtraction (+, -)

Comparison operators (Crystal syntax)

The comparison operators are equal (=), not equal (<>), less than (<), less than or equal (<=), greater than (>) and greater than or equal (>=).

Comparison operators are usually used to compare operands for a condition in a control structure such as an If expression. Comparison operators as a group all have lower precedence than the arithmetic operators. Thus, expressions like 2 + 3 < 2 * 9 are the same as (2 + 3) < (2*9).

Boolean operators (Crystal syntax)

The Boolean operators are, in order of precedence from greatest to lowest: Not, And, Or, Xor, Eqv and Imp. Boolean operators are typically used with comparison operators to generate conditions for control structures. Boolean operators as a group have lower precedence than the comparison operators. Thus for example, the expression 2 < 3 And 4 >= -1 is the same as (2 < 3) And (4 >= -1).

Null fields and how to use IsNull (Crystal syntax)

The {Product.Color} field contains both basic colors such as "red" and "black" and fancy two word colors such as "steel satin" and "jewel green". Suppose that you want to write a formula that writes out "basic" for the basic colors and "fancy" for the others.

```
If InStr({Product.Color}, " ") = 0 Then
    "basic"
Else
    "fancy"
```

The function call to InStr searches the {Product.Color} string for a space. If it finds a space, it returns the position of the space, otherwise it returns 0. Since basic colors are only one word with no spaces, InStr will return 0 for them.

Dealing with null values (Crystal syntax)

For some products, such as the Guardian Chain Lock, a color value was not recorded and so the {Product.Color} field has a null value in the database for that record. In general, when LABWORKS Report Designer encounters a null valued field in a formula, it immediately stops evaluating the formula and produces no value. Thus, the Guardian Chain Lock record does not have any word printed beside it. If you want to handle null field values in your formula, you must explicitly do so using one of the special functions designed for handling them: IsNull, PreviousIsNull or NextIsNull.Here is how to fix up the previous example using IsNull:

```
If IsNull({Product.Color}) Or
    InStr({Product.Color}, " ") = 0 Then
    "basic"
Else
    "fancy"
```

How this relates to operators, is that when LABWORKS Report Designer evaluates the condition:

```
IsNull({Product.Color}) Or
InStr({Product.Color}, " ") = 0
```

It first evaluates IsNull ({Product.Color}), and when it determines that this is True, it knows that the whole condition is True, and thus does not need to check whether

```
InStr({Product.Color}, " ") = 0
```

In other words, LABWORKS Report Designer will stop evaluating a Boolean expression when it can deduce the results of the whole expression. In the following example, the formula guards against attempting to divide by zero in the case that denom is 0:

```
Local NumberVar num; Local
NumberVar denom;
...
If denom <> 0 And num / denom > 5 Then
...
```

Control structures (Crystal syntax)

Formulas without control structures execute each expression in the formula exactly once when the formula is evaluated. The expressions are executed in a sequential fashion, from the first expression in the formula to the last. Control structures enable you to vary this rigid sequence. Depending upon which control structure you choose, you can skip over some of the expressions or repeatedly evaluate some expressions depending on if certain conditions hold. Control structures are the primary means of expressing business logic and typical report formulas make extensive use of them.

If expressions (Crystal syntax)

The If expression is one of the most useful control structures. It allows you to evaluate an expression if a condition is true and evaluate a different expression otherwise.

NOTE: When formatting with conditional formulas, always include the Else keyword; otherwise, values that don't meet the If condition may not retain their original format. To prevent this, use the DefaultAttribute function (If...Else DefaultAttribute).

Example

A company plans to pay a bonus of 4 percent to its employees except for those who work in Sales who will receive 6 percent. The following formula using an If expression would accomplish this:

```
//If example 1
If {Employee.Dept} = "Sales" Then
    {Employee.Salary} * 0.06
Else
```

```
    {Employee.Salary} * 0.04
```

In this example, if the condition {Employee.Dept} = "Sales" evaluates as true, then the

```
{Employee.Salary} * 0.06
```

expression is processed. Otherwise the expression following the Else, namely the

```
{Employee.Salary} * 0.04
```

is processed.

Suppose another company wants to give employees a 4% bonus, but with a minimum bonus of $1,000. The following example shows how. Notice that the Else clause is not included; it is optional, and not needed in this case.

```
//If example 2
Local CurrencyVar bonus := {Employee.Salary} * 0.04;
If bonus < 1000 Then
bonus := 1000;
//The final expression is just the variable 'bonus'.
//This returns the value of the variable and is the
//result of the formula bonus
```

Another way of accomplishing example 2 is to use an Else clause:

```
//If example 3
Local CurrencyVar bonus := {Employee.Salary} * 0.04;
If bonus < 1000 Then
    1000
Else
    Bonus
```

Now suppose that the previous company also wants a maximum bonus of $5,000. You now need to use an Else If clause. The following example has only one Else If clause, but you can add as many as you need. Note, however, that there is a maximum of one Else clause per If expression. The Else clause is executed if none of the If or Else If conditions are true.

```
//If example 4
Local CurrencyVar bonus := {Employee.Salary} * 0.04;
If bonus < 1000 Then
    1000
Else If bonus > 5000 Then
   5000
Else
   bonus;
```

If example (Crystal syntax)

Suppose that a company wants to compute an estimate of the amount of tax an employee needs to pay and write a suitable message. Income below $8,000 is not taxed, income from $8,000 to $20,000 is taxed at 20% income from $20,000 to
$35,000 is taxed at 29% and income above $35,000 is taxed at 40%.

```
//If example 5
Local CurrencyVar tax := 0;
Local CurrencyVar income := {Employee.Salary}; Local
```

```
StringVar message := "";
If income < 8000 Then
(
     message := "no";
     tax := 0
)
Else If income >= 8000 And income < 20000 Then
(
    message := "lowest";
    tax := (income - 8000)*0.20
)
Else If income >= 20000 And income < 35000 Then
(
    message := "middle";
    tax := (20000 - 8000)*0.20 + (income - 20000)*0.29
)
Else
(
    message := "highest";
    tax := (20000 - 8000)*0.20 + (35000 - 20000)*0.29 + (income -
     35000)*0.40
);
//Use 2 decimal places and the comma as a
//thousands separator
Local StringVar taxStr := CStr (tax, 2, ",");
"You are in the " & message & " tax bracket. " &
Your estimated tax is " & taxStr & "."
```

The use of variables is to simplify the logic of the computation. Also, there are 2 expressions that are executed when one of the conditions are met; one assigns the tax variable, and the other assigns the message variable. It is often useful to have multiple expressions executed as a result of a condition.

More details on If expressions (Crystal syntax)

The If expression is an expression. In other words it evaluates to a value of a given type. If there is no Else clause, and the condition is not true, then the value is the default value for the type. For example:

```
If Length ({Employee.First Name}) < 5 Then
"short"
```

The above If expression returns a String value. The string value is "short" if the Employee's first name has fewer than 5 letters and the empty String "" otherwise. Consider the formula:

```
If Year({Orders.Order Date}) >= 1995 Then
{Orders.Order Date}
```

For order dates before 1995, the above If expression returns the null DateTime value. It is a DateTime value rather than a Date value since {Orders.Order Date} is a DateTime database

field. The null DateTime value is not printed by LABWORKS Report Designer so if the above formula is placed in a report, the formula field would be blank for order dates before 1995. Null Time values and null Date values behave similarly.

Here is an example that illustrates the use of parentheses to have more than one expression executed as the outcome of an If condition. A company charges a 5 percent fee for orders shipped within 3 days and a 2 percent fee otherwise. It wants to print messages such as "Rush shipping is $100.00" or "Regular shipping is
$20.00" as appropriate.

```
Local StringVar message; Local
CurrencyVar ship;
If {Orders.Ship Date} - {Orders.Order Date} <= 3 Then
(
    message := "Rush";
    //A semicolon at the end of the next line
    //is optional
    ship := {Orders.Order Amount} * 0.05
) //A semicolon cannot be placed here
Else
(
    message := "Regular";
    ship := {Orders.Order Amount} * 0.02;
);
//The preceding semicolon is required to separate the
//If expression from the final expression below message & "
shipping is " & CStr (ship)
```

When expressions are grouped together with parentheses, the whole group is considered as a single expression, and its value and type are the value and type of the final expression inside the parentheses.

```
//The parentheses group expression as a whole has
//Currency type
(
    //The first expression in the parentheses has
    //String type
    message := "Rush";
    //The second and final expression in parentheses
    //has Currency type
    ship := {Orders.Order Amount} * 0.05;
)
```

Thus, for example, the following formula gives an error. The reason is that the Then part of the If expression returns a Currency value while the Else part returns a
String value. This is not allowed, since the If expression is an expression and so must always return a value of a single type.

```
//An erroneous formula
Local StringVar message;
Local CurrencyVar ship;
```

```
If {Orders.Ship Date} - {Orders.Order Date} <= 3 Then
(
    message := "Rush";
    ship := {Orders.Order Amount} * 0.05
)
Else
(
     //The following 2 lines were interchanged ship :=
     {Orders.Order Amount} * 0.02;
     message := "Regular";

);
message & " shipping is " & CStr (ship)
```

One way of fixing up the erroneous formula without being careful about expression order is just to make the If expression return a constant value of the same type in every branch. For example, the If expression now returns the Number value 0:

```
//Repaired the erroneous formula
Local StringVar message; Local
CurrencyVar ship;
If {Orders.Ship Date} - {Orders.Order Date} <= 3 Then
(
    message := "Rush";
    ship := {Orders.Order Amount} * 0.05;
    0
)
Else
(
     ship := {Orders.Order Amount} * 0.02;
     message := "Regular";
     0
);
message & " shipping is " & CStr (ship)
```

Select expressions (Crystal syntax)

The Select expression is similar to an If expression. Sometimes however, you can write clearer and less repetitive formulas using the Select expression. For example,
to evaluate the {Customer.Fax} field to determine if the area code is for Washington state (206, 360, 509) or British Columbia, Canada (604, 250):

```
//Select example 1
Select {Customer.Fax}[1 To 3]
Case "604", "250" :
     "BC"
Case "206", "509", "360" :
     "WA"
Default :
```

```
    "";
```

The expression right after the Select keyword is called the Select condition. In the above example it is {Customer.Fax}[1 To 3]. The Select expression tries to find the first Case that matches the Select condition, and then executes the expression following the colon for that Case. The Default case is matched if none of the preceding cases matches the Select condition. Notice that there is also a colon after the Default.

```
//Same effect as Select example 1
Local StringVar areaCode := {Customer.Fax}[1 To 3];

If areaCode In ["604", "250"] Then
    "BC"

Else If areaCode In ["206", "509", "360"] Then
    "WA"

Else
    "";
```

Example

This formula groups the number of Oscar nominations a movie received into low, medium, high or extreme categories and in the process, shows some of the possibilities for the expression lists following the Case labels:

```
//Select example 2
Select {movie.NOM} Case 1,2,3,
Is < 1 :
(
    //Can have expression lists by using
    //parentheses
    10 + 20;

    "low"
)
Case 4 To 6, 7, 8, 9 : "medium"
Case 10 :
    "high"
Default :
    "extreme"
```

The Default clause of the Select expression is optional. If the Default clause is missing and none of the cases are matched, then the Select expression returns the default value for its expression type. For example, if in the above example the Default clause were omitted and {movie.NOM} = 11, it would return the empty string "". The Select expression is an expression, and similar comments as in the More details on If expressions section apply to it as well.

For loops (Crystal syntax)

For loops enable you to evaluate a sequence of expressions multiple numbers of times. This is unlike the If and Select expressions where the program passes through each expression at most once during the formula's evaluation. For loops are best when you know the number of times that the expressions needs to be evaluated in advance.

The syntax of the For loop through examples

Example 1

Suppose you want to reverse the {Customer.Customer Name} string. For example, "City Cyclists" becomes "stsilcyC ytiC".

```
//Reverse a string version 1
Local StringVar str := ""; Local
NumberVar strLen :=
    Length ({Customer.Customer Name});
Local NumberVar i;
For i := 1 To strLen Do
(
                Local NumberVar charPos := strLen - i + 1;
                str := str + {Customer.Customer Name}[charPos]
  );
  str
```

Examine how this formula works assuming that the current value of the field {Customer.Customer Name} is "Clean Air". The variable strLen is assigned to be the length of "Clean Air", namely 9. The variable i is known as a For counter variable since its value changes with each iteration of the For loop. In other words, it is used
to count the iterations of the loop. The For loop will iterate 9 times, during the first time, i is 1, then i is 2, then i is 3 and so on until finally i equals 9. During the first iteration, the ninth character of {Customer.Customer Name} is appended to the empty string variable str. Thus str equals "r" after the first iteration. During the second iteration, the eighth character of {Customer.Customer Name} is appended to str and so str equals "ri". This continues until after the ninth iteration, str equals,
"riA naelC" which is the reversed string.

Example 2

Here is a simpler version of the above formula that uses a Step clause with a negative Step value of -1. For the "Clean Air" example, i is 9 for the first iteration, 8 for the second, 7 for the third and so on until it is 1 in the final iteration.

```
//Reverse a string version 2
Local StringVar str := "";
Local NumberVar strLen :=
    Length ({Customer.Customer Name});
Local NumberVar i;
For i := strLen To 1 Step -1 Do
(
    str := str + {Customer.Customer Name}[i]
);
str
```

Example 3

The simplest version is to use the built in function StrReverse:

```
//Reverse a string version 3
StrReverse ({Customer.Customer Name})
```

The built in String functions in LABWORKS Report Designer 8.5 can handle many of the string processing applications that would traditionally be handled using a For loop or some other kind of loop. However, For loops provide the most flexibility in processing strings and also power in processing arrays, which can be essential if the built-in functions do not cover your intended application.

For loop example

Here is a more full featured example of Crystal Reports' string processing capabilities. The Caesar cipher is a simple code that is traditionally credited to Julius Caesar. In this code, each letter of a word is replaced by a letter five characters further in the alphabet. For example, "Jaws" becomes "Ofbx". Notice that "w" is replaced by "b"; since there are not 5 characters after "w" in the alphabet, it starts again from the beginning. Here is a formula that implements applying the Caesar cipher to the field {Customer.Customer Name} in the Example database:

```
//The Caesar cipher
```

```
//The input string to encrypt
Local StringVar inString := {Customer.Customer Name};
Local NumberVar shift := 5;
Local StringVar outString := "";
Local NumberVar i;
For i := 1 To Length(inString) Do
(
    Local StringVar inC := inString [i];
    Local StringVar outC;
    Local BooleanVar isChar :=
    LowerCase(inC) In "a" To "z";
    Local BooleanVar isUCaseChar :=
   isChar And (UpperCase (inC) = inC);
   inC := LCase(inC);
   If isChar Then
(
    Local NumberVar offset :=
    (Asc(inC) + shift - Asc("a")) Mod
    (Asc("z") - Asc("a") + 1);
    outC := Chr(offset + Asc("a"));
    If isUCaseChar Then outC := UpperCase(outC)
)
Else
    outC := inC;
    outString := outString + outC
);
outString
```

In the above example there is an If expression nested within the expression block of the For loop. This If expression is responsible for the precise details of shifting a single character. For example, letters are treated differently from punctuation and spaces. In particular, punctuation and spaces are not encoded. The general points here are that control structures can be nested within other control structures and that multiple expressions can be included in the (parentheses enclosed) expression blocks of other control structures

Using Exit For (Crystal syntax)

You can exit from a For loop by using Exit For. The following example searches the Global array names for the name "Fred". If it finds the name, it returns the index of the name in the array. Otherwise it returns -1.For example, if the names array is:

```
["Frank", "Helen", "Fred", "Linda"]
```

Then the formula returns 3.

```
Global StringVar Array names;
//The names array has been initialized and filled
//in other formulas
Local NumberVar i;
```

```
Local NumberVar result := -1;
//The UBound function returns the size of its array
//argument
For i := 1 to UBound (names) Do
(
    If names [i] = "Fred" Then
    (
        result := i;\
        Exit For
    )
);
result
```

When considered as an expression, the For loop always returns the Boolean value True. Thus you will almost never want a For loop to be the last expression in a formula, since then the formula will then just display the value True rather than your intended result.

While loops (Crystal syntax)

Another looping mechanism is the While loop. A While loop can be used to execute a fixed block of statement an indefinite amount of time.

Two different types of While loops

Type of Do Loop	Explanation	Example
While ... Do	The While ... Do loop evaluates the condition, and if the condition is true, then it evaluates the expression following the Do. When it has finished doing this, it evaluates the condition again and if the condition is true, it evaluates the expression following the Do again. It continues repeating this process until the condition is false.	While condition Do expression
Do ... While	The Do ... While loop evaluates the expression once no matter what. It then evaluates the condition, and if the condition is true, evaluates the expression again. This process continues until the condition is false.	Do expression While condition

The While loops support an Exit While statement to jump immediately out of the loop. Its use is analogous to the use of Exit For in For loops.

As with the For loop, the While loop when considered as an expression always returns the Boolean value True.

While ... Do loop example (Crystal syntax)

The following example searches for the first occurrence of a digit in an input string. If a digit is found, it returns its position, otherwise it returns -1. In this case, the input string is set explicitly to a string constant, but it could be set equal to a String type database field instead. For example, for the input String, "The 7 Dwarves", the formula returns 5, which is the position of the digit 7.

```
Local StringVar inString := "The 7 Dwarves";
Local NumberVar strLen := Length (inString);
Local NumberVar result := -1;
Local NumberVar i := 1;
While i <= strLen And result = -1 Do
(
    Local StringVar c := inString [i];
    If NumericText (c) Then
        result := i;
    i := i + 1;
);
result
```

Safety Mechanism for Loops (Crystal syntax)

There is a safety mechanism to prevent report processing from hanging due to an infinite loop. Any one evaluation of a formula can have at most 30,000 loop condition evaluations per formula evaluation. This will be explained by the example below. For example:

```
Local NumberVar i := 1; While i <=
100000 Do
(
    If i > {movie.STARS} Then
    Exit While;
    i := i + 1
);
20
```

If {movie.STARS} is greater than 30,000 then the loop condition (i <= 100000) will be evaluated more than the maximum number of times and an error message is displayed. Otherwise, the loop is OK.

The safety mechanism applies on a per formula base, not for each individual loop. For example:

```
Local NumberVar i := 1; For i := 1 To
10000 Do
(
```

```
    Sin (i);
);
While i <= 25000 Do
(
    i := i + 1;
)
```

The above formula also triggers the safety mechanism since the 30,000 refers to the total number of loop condition evaluations in the formula and this formula will have 10001 + 25001 such evaluations.

Limitations (Crystal syntax)

For reference purposes, here are the sizing limitations of the formula language:

- The maximum length of a String constant, a String value held by a String variable, a String value returned by a function or a String element of a String array is 254 bytes.
- The maximum size of an array is 1000 elements.
- The maximum number of arguments to a function is 1000. (This applies to functions that can have an indefinite number of arguments such as Choose).
- The maximum length of the text of a formula is 64K.
- The maximum number of loop condition evaluations per evaluation of a formula is 30,000.
- Date-time functions modeled on Visual Basic accept dates from year 100 to year 9999. Traditional LABWORKS Report Designer functions accept dates from year 1 to year 9999.

Parameter Fields

About Parameter Fields

The purpose of parameters is to prompt a report user to enter information. The responding information determines what appears in the report. By using parameter fields in the report, formulas, and selection formulas, you can create a single report that can be modified as needed. This chapter explains parameter fields are and how they can be used to create a report that accesses different types of data depending on the user's needs.

Parameter fields support the following data types:

Boolean – requires a yes/no or true/false answer.

Currency – requires a dollar amount.

Date – requires an answer in a date format.

DateTime – requires both date and time.

Number – requires a numeric value.

String – requires a text answer.

Time – requires an answer using a time format.

Parameter field prompting text can be up to four lines long with approximately 60-70 characters per line; depending on character width, up to the 254-character limit. Text over one line in length automatically wraps.

You can create a pick list for the user to select the parameter value rather than having them enter it manually.

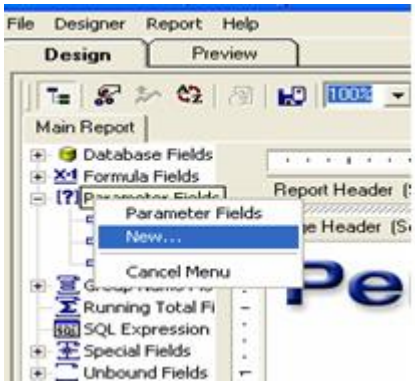
A parameter field does not have to be placed in a report to be used in a record or a group selection formula. You create the parameter field and then enter it in your formula as you would any other field.

Creating a Parameter Field

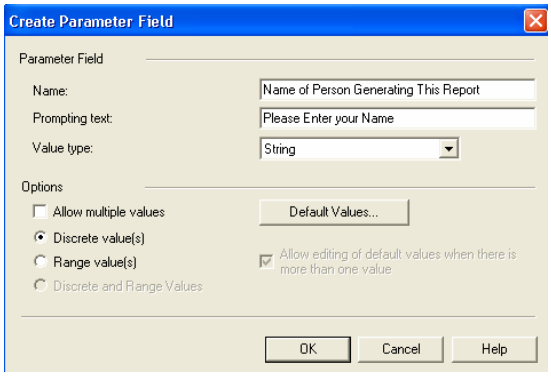
To create a parameter field:

1. Open the report to which you want to add a parameter field in LABWORKS Report Designer.

2. Right click Parameter Fields in the list on the left side of the screen and click New.



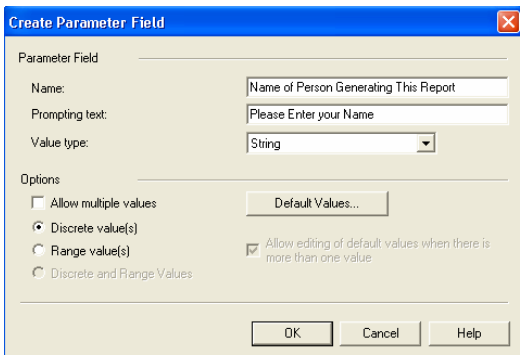
3. Notice that the Create Parameter Field dialog box appears.



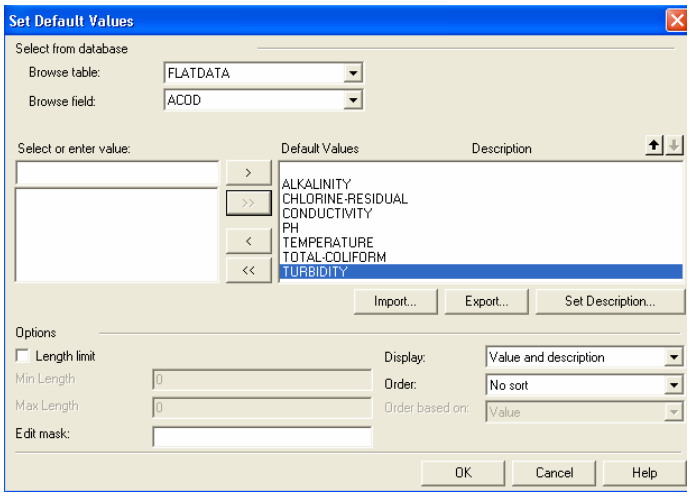
4. Enter a name for the parameter in the **Name** field – name may be up to 255 alphanumeric characters.

5. Enter the desired prompting text in the **Prompting text** field – may be up to 255 alphanumeric characters. This text appears in the Enter Parameter Values dialog box when the report is refreshed.

6. Select the appropriate **Value type** from the list. When creating a parameter whose Value type is either Date or DateTime, you can change the date format as needed.



7. Click **Set Default Values**. The Set Default Values dialog box appears.

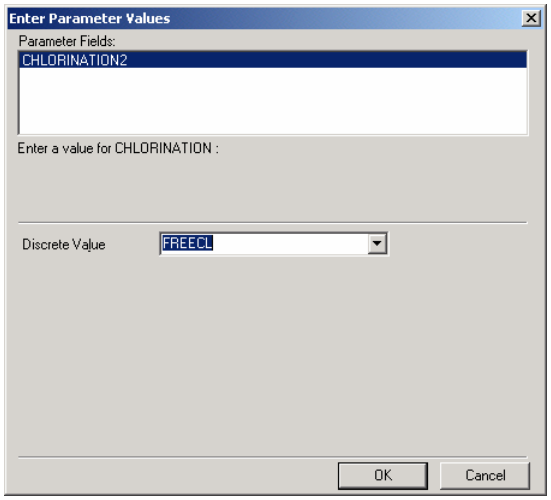


8. Select the database in the **Browse Table** field.
9. Select the field from the **Browse field** drop-down list.
10. Click >> to move all of the fields to the **Default Values** area.
11. Click **OK**. The Create Parameter Field dialog box appears.
12. Click **OK**.
13. Notice that the created parameter now appears in the **Parameter Fields** list.
14. Drag and drop the created parameter into your report. If you don't want to see the parameter field you dropped in your report, place it in a section you can suppress, such as a report header or footer.
15. Click **Close**.

To incorporate the parameter into the selection:

1. Right click on the report and click **Report** in the pull-down menu.
2. Click **Select Expert**. The Choose Field dialog box appears.
3. Select the desired field from the database table and then click **OK**. The Select Expert appears.
4. Select how the value is to function from the drop-down list, e.g., is equal to.
5. Select the parameter from the adjacent drop-down list.
6. Click **OK**.
7. Save the report.
8. Click **File** and click **Print Preview**.

9. Notice that the Enter Parameter Values dialog box appears.



10. Select a value in the **Discrete Value** field.

11. Click **OK**. The report appears with the information for Free Chlorine (FREECL).

Deleting Parameter Fields

There are several ways to delete parameters in a report. The type of parameter you are deleting determines the method.

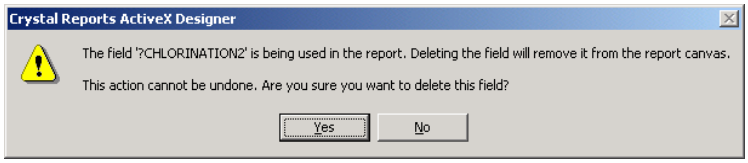
To delete a parameter that is not used in a formula:

1. Expand the **Parameter Fields** folder.

2. Right click on the parameter you want to delete.

3. Click **Delete**.

4. If the parameter is being used in the report, a deletion confirmation message appears.

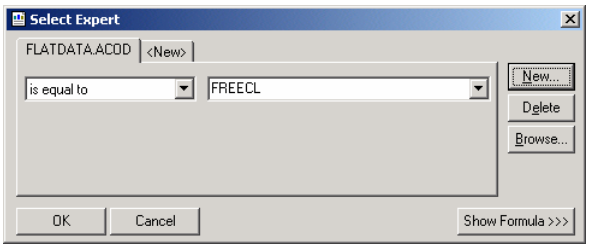


5. Click **Yes** to delete the parameter field. The parameter is removed from the Parameter Fields folder and from your report.

To delete a parameter used with the Select Expert:

1. Right click on the report and click **Report** in the pull-down menu.

2. Click **Select Expert**. The Select Expert dialog box appears.



3. Click on the tab whose selection criteria uses the parameter you want to delete.
4. Click the **Delete** button
5. Click **OK** to close the Select Expert.
6. Expand the **Parameter Fields** folder and click the parameter you want to delete.
7. Click **Delete**.

To delete a parameter that is used in a formula:

1. Select the formula that contains the parameter you want to delete in the **Formula Fields** folder.
2. Click **Edit** and delete the parameter field from the formula.

   NOTE: If the parameter is used in more than one formula, it must be deleted from each formula.
3. Close the Formula Editor.
4. Expand the **Parameter Fields** folder and click the parameter you want to delete.
5. Click **Delete**.

Advanced Parameter Features

Parameters may be used in a variety of ways within a report. This section explains some of the advanced methods of using parameters.

Creating a parameter with multiple values

To create a parameter with multiple values:

1. Select **Parameter Fields** and click **New**.
2. Enter a Name and Prompting text.
3. Select a **Value type**.
4. Select the **Allow multiple values** checkbox to enable more than one value to be entered into the parameter.
5. Click **Set default values** to determine the type of entries that can be entered. The Set

Default dialog box appears.

6. From the Browse **table** drop-down list, select the table for the default values.

7. From the Browse **field** drop-down list, select the field for the default values. The values associated with this table and field appear in the **Select or enter value to add** area.

8. Select the values listed, or create your own, and move them to the **Default Values** area using the arrow buttons.

9. Adjust the display, order, and length limit as required.

10. Click **OK**.

Applying conditional formatting using parameter fields

Parameter fields can be used to create conditional formatting formulas that can be customized when you refresh the report data. A conditional formatting formula may be used for color-flagging data that meets certain conditions.

You do not need to use parameter fields if the conditions for flagging these items never change. You can just use formulas for text flags or conditional formatting for border flags. You need to use parameter fields in formulas and conditional formatting formulas to change the conditions from report to report.

To apply conditional formatting using parameter fields:

1. Create the parameter field of the data type you need for the formula.

2. Create the formula and use the parameter field in place of the fixed value you would normally use.

3. Right click on the field and click **Format.** The Format Editor appears.

4. Click the **Conditional formula** button next to the **Color** property on the **Font** tab, and format the field using a conditional formatting formula like this:

```
If {customer.LAST YEAR'S SALES} > {?SalesTarget} Then
Red
Else
Black
```

In this example, when you refresh the data, the program prompts you for the value that triggers the color flag, known as the threshold value. It then runs the report and flags all the customers that had sales last year above the threshold figure. You can change the figure each time you run the report and the program will flag a different set of Customer Names.

Creating a report title using parameter fields

You can use parameter fields to create a report title that can be changed each time the report is refreshed.

To create a report title using parameter fields:

1. Select **Parameter Fields** and click **New**. The Create Parameter Field dialog box appears.
2. Type a name for the parameter field in the **Name** field.
3. Type in any prompting text you wish in the **Prompting text** field.
4. Select **String** from the **Value type** drop-down list.
5. Click Set default values. The Set Default Values dialog box appears.
6. Type the desired text in the **Select or enter value to add** field and click > to add the title to the **Default Values** area.
7. Continue adding titles as required.
8. Click **OK**. The Create Parameter Field dialog box appears.
9. Click **OK**. The new parameter appears in the Parameter Fields list.
10. Place the parameter field in the **Page Header** section of the report to have the title appear on every page.
11. Place the parameter field in the **Report Header** section if you want the title to appear on only the first page of the report. Now, when you refresh the data, the program prompts you for a report title. You can change the title each time you run the report, if desired.

Specifying single or ranges of values

To specify single or range of values:

1. Select the **Discrete value(s)** option if you want the parameter field to accept single values rather than ranges of values.
2. Select the **Range value(s)** option to specify that the parameter field accept a range of values.
3. If you select **Range value(s),** enter a start value and an end value when prompted for parameter values, and the report then displays all records within that range. For example, if you enter the values 5 and 10, the range is 5-10, and the report will display all records with values between 5 and 10. This also applies to string parameter fields. With a start value of A and an end value of E, the report displays all records within an alphabetical range of A-E.
4. Select the Allow multiple values and the Discrete value(s) field accept multiple single values.

NOTE: This allows you to enter more than one value, but these values are evaluated individually and are not interpreted as a range.

5. Select the Allow multiple values and **Range value(s**) options to have the parameter field accept multiple ranges.

6. If applicable, click **Set default values** to select the **Length limit** checkbox to designate the length of the field. If you select Length limit: for a Currency or Number parameter field, the Min Value and Max Value fields appear for a DateTime parameter field; the Start Date-time and End Date-time fields appear for a Date parameter field; the Start Date and End Date fields appear for a Time parameter field; the Start Time and End Time fields appear.

Incorporating a parameter into a formula

To incorporate a parameter into a formula:

1. Select **Parameter Fields** and click **New**. The Create Parameter Field dialog box appears.

2. Create a parameter field and save it.

3. Right click on **Formula Fields** and click **New**. The Formula Name dialog box appears.

4. Enter the name of the formula and click **OK**.

5. Create a formula using the parameter field as you would any constant value.

6. Double click the database field, press =, then double click the parameter. Identify parameter fields easily by looking for (?).

7. Click **Save** and close. The name of the formula you created is highlighted in the Formula list.

8. Drag and drop the formula into the report.

9. Click **File > Print Preview** to generate the report. A dialog box appears prompting you for values.

Subreports

What is a Subreport

A subreport is a report within a report. The process for creating a subreport is similar to the process of creating a regular report. A subreport can have most of the characteristics of a report, including its own record selection criteria. The only differences between a subreport and a primary report are that a subreport:

- Is inserted as an object into a primary report; it cannot stand on its own (although a subreport can be saved as a primary report).
- Can be placed in any report section and the entire subreport will print in that section.
- Cannot contain another subreport.
- Does not have Page Header or Page Footer sections.

There are four instances in which a subreport would typically be used:

- To combine unrelated reports into a single report.
- To coordinate data that cannot otherwise be linked.
- To present different views of the same data within a single report.
- To perform one-to-many lookups from a field that is not indexed on the lookup field.

Processing considerations for subreports

When using or considering the use of subreports, there are some considerations to keep in mind that will affect how fast and efficient your reports will process. Consider the following:

- Subreports will require additional resources.
- Make sure you have enough resources, such as free disk space and memory, for the size of the database and the amount of processing a report requires. When Crystal Reports processes subreports, it may call data from the database onto the client computer, depending on what functions are being used in the subreport's formulas. Client-side processing requires additional hard disk space and increases network usage. If there are not enough resources on the computer, the report may stop processing or become corrupt, or the computer may return an error.
- Where you place a subreport determines report processing time.
- The report section where you place a subreport affects the processing time of the whole report. For example, placing the subreport in the Details section of the main report causes the subreport to print once for each record returned on the main report. This slows report processing. If the subreport is placed in the Report Header, the subreport will only print once per report.
- Where you place a linked subreport determines what data is returned.

- In reports that contain groups, a subreport can be inserted into the Group Header or Footer. If the subreport is linked to the main report on the grouped field, each time the group changes the subreport reflects data for that group.

- If you are passing values from the subreport to the main report with shared variables, Crystal Reports must first evaluate the shared variable in the subreport before it calls it from the main report. To accomplish this place the subreport in a main report section above the section containing the formula that calls the shared variable.

- Page Headers belong to each individual page and therefore cannot span more than one page. This can be an issue when a subreport is placed in the Page Header section of a main report. If the subreport is too large to fit on one page, the following message is displayed: "The page size is not large enough to format the contents of an object".

How subreport linking works

When you link a subreport to a primary report, the program creates the link using a parameter field.

- When a subreport link field is selected the program creates:

- A parameter field in the subreport which is then used to retrieve values passed to it by the primary report.

- A record selection formula for the subreport using the parameter field.

- The selection formula limits the subreport to those records in which the link field is equal to the parameter field value.

When the reports run, the program finds the first primary field record it needs and passes the value in the link field to the parameter field in the subreport. The program then creates the subreport with record selection based on the parameter field value. He is an example:

- You create a report that shows customer data and a subreport that shows order data. Then you link the two reports using the Customer ID field.

- When you run the report, the program finds the first customer record it needs and passes the Customer ID value from that record to the subreport parameter field.

- The program runs the Orders subreport. Since the subreport selection formula selects only those records in which the Customer ID value is equal to the parameter field value, and since that parameter field value is equal to the Customer Id in the first record in the primary report, the subreport contains only those records that have the same customer ID. Namely, those records that are orders for the first customer.

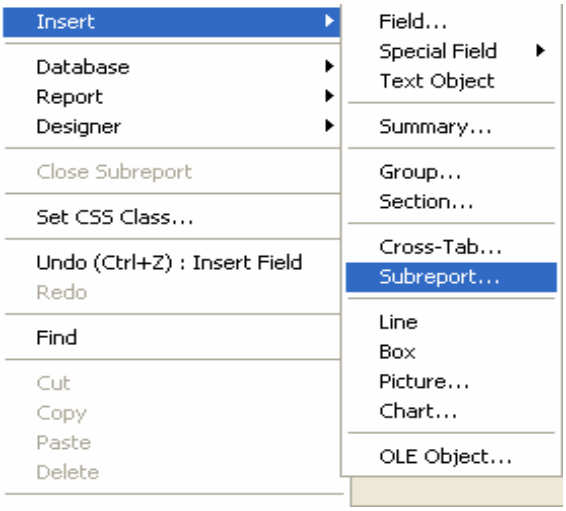
- When the subreport is finished the program locates the second record it needs in the primary report, prints the customer data, and then passes this customers ID number to the parameter field.

- The program then runs a subreport including only those order records for the second customer.

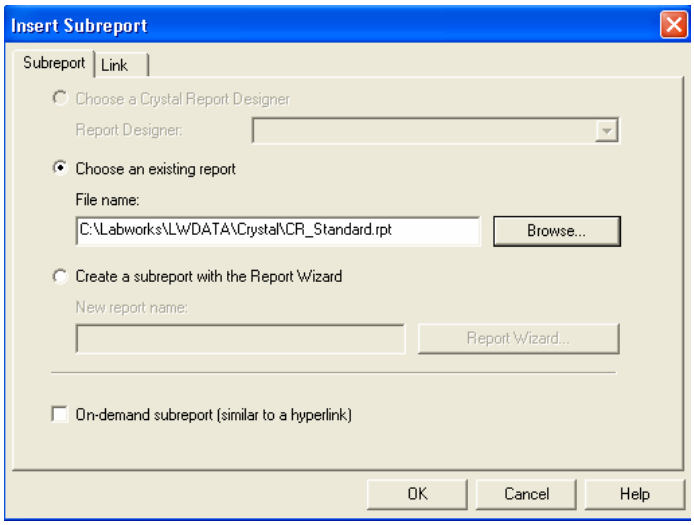
- The process continues until the report is finished.
- All of this parameter field manipulation takes place behind the scenes. You pick the fields that will link the primary report with the subreport and the program does the rest. The values are passed without the parameter field prompting you for a value.
- Using a pre-existing report as a subreport.
- You can insert an existing report into another report as a subreport. Open the primary report and follow the procedure below.

To use a pre-existing report as a subreport:

1. With the primary report open, right click **INSERT>SUBREPORT**. The Insert Subreport dialog box opens.



2. Click **Choose a report** and enter the name. If you do not know the name of the subreport, click the **Browse** button and locate it in the dialog box.

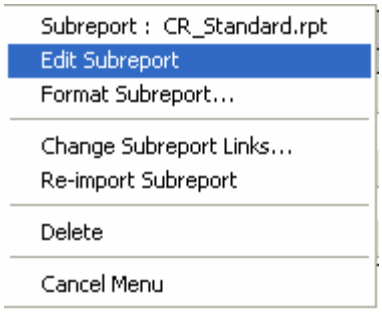


3. In the **Report Name** text box, enter the name and path of the report, or click **Browse** and choose the report from the dialog box that opens.

4. From the Insert Subreport dialog box, click **OK**. The program displays a rectangular object placement frame.

5. Move the top margin of the object frame into the section where you want the object and click once to place it. The program places the subreport object into that section and increase the size for the section vertically, if required. The program creates a subreport design tab labeled with the subreport name.

6. If you need to edit your report, right click on the subreport and modify your subreport as you would any other report. You can suppress any sections in the subreport that you do not want to display.



7. Preview your subreport. Two options exist when previewing your subreport. If you click Print preview while in the subreports Design tab, you can preview just the subreport. If you want to preview the subreport as part of the main report, click Print Preview from the primary reports Design tab.

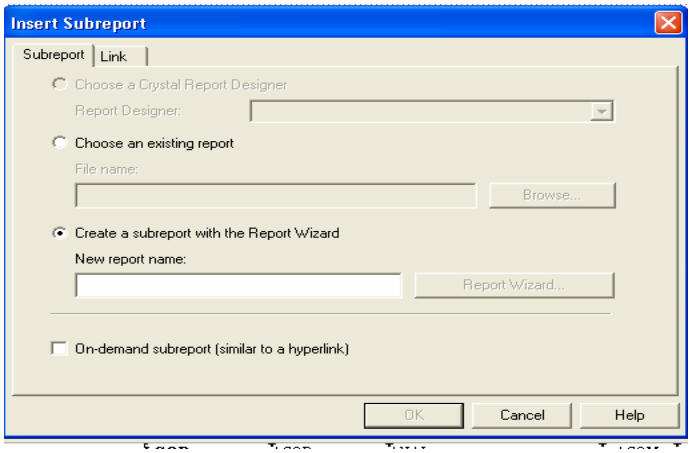
Note: Subreports do not contain Page Header of Page Footer sections.

Using the Report Wizard to create a subreport

You can use the Report Wizard to create and insert a subreport into another report. Open the primary report and follow the procedure below.

To use the Report Wizard to create a subreport:

1. With the primary report open, click **Insert>Subreport**. The Insert Subreport dialog box opens.



2. Click **Create a subreport with the Report Wizard** and enter a name in the **New report** name text box.
3. Click the **Report Wizard** button.
4. Follow the steps in the wizard to crate your subreport.
5. Click **OK** when you are finished with the wizard to return to the Insert Subreport dialog box.
6. From the Insert Subreport dialog box, click **OK**. The program displays rectangular object placement frame.
7. Move the top margin of the object frame into the section you want the object and click to place it. The program places the subreport object into that section and increases the size of the section vertically, if required. The program creates a subreport design tab labeled with the subreport name.
8. Preview your subreport. Two options exist when previewing your subreport. If you click Print Preview while in the subreport's Design tab, you can preview just the subreport. If you want to preview the subreport as part of the main report, click Print Preview from the primary report's Design tab.

Modifying a subreport

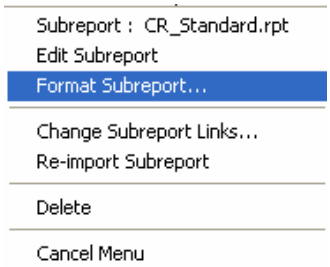
Once you have created and inserted your subreport, you may want to make some modifications to it. Applying object formatting or section formatting within the subreport is the same as with any other report. You can also add formatting to the subreport placeholder. Items such as the border, fonts and subreport labels can be formatted or changed. To edit a subreport, you access

the subreport design tab and make changes to the report just as you would any other report. Any of the sections and objects on the report can be formatted.

To format a subreport:

1. In either the design or preview tab of the main report, right-click the subreport placeholder.

2. From the shortcut menu, click **Format Subreport**. The Format Editor will open.



3. You can now apply formatting using the tabbed dialog.

4. QA/QC Module Technical Reference

Technical Prologue

Introduction

This chapter is designed for the technical specialist or designated system administrator of the LABWORKS LIMS. Generally the person(s) assigned these tasks will handle software installation, system configuration, integration into other tools, and database preparation (of analysis prefixes, SI forms, and pick lists). These tasks require computer / database technical skills outside of laboratory/environmental specific duties. This documentation is tailored for people handling these duties and is not designed for the operator of the software. If you plan on operating the QA/QC Module, please refer to the LABWORKS LIMS QA/QC Module Setup and Configuration Reference in the next chapter.

This document also assumes the Server and Client installs of LABWORKS LIMS has been completed successfully. Installation of LABWORKS is not covered here.

Overview

The QA/QC Module is an integral part of the LABWORKS LIMS. It is considered a core application, and therefore does not require a special control file license for usage. The only required license is a valid LABWORKS LIMS read/write Desktop license with the appropriate privilege to run the application. The privileges for QA/QC will be discussed later in the chapter.

A technical specialist who is in charge of QA/QC Module implementation and maintenance will have most of their work prior to initial usage. Once the lab begins using the application, general (System Manager related) maintenance may include tasks such as adding additional prefixes in SysMgr table or assisting with integration into other areas of LABWORKS (Chemical Inventory, Instrument Maintenance and Calibration, and User Training Module). For the most part, after initial setup the lab should be able to handle all daily tasks by following instructions in the QA/QC User Manual.

The QA/QC Module introduced in LABWORKS version 6.3 includes a system built using the latest LABWORKS architecture and Microsoft .NET framework. It includes a new UI with the look and feel of other core modules within the system. Users already familiar with the legacy QA/QC Module may have to get used to the new interface for daily tasks, but new users to the system will have a significantly lower learning curve to accomplish the same daily tasks.

QA/QC Module Execution

QA/QC Application Install / Upgrade Information

To use the latest QA/QC module, you must first be running at least LABWORKS 6.4. The LABWORKS 6.4 installation process installs the new QA/QC module programs. No additional installation is required.

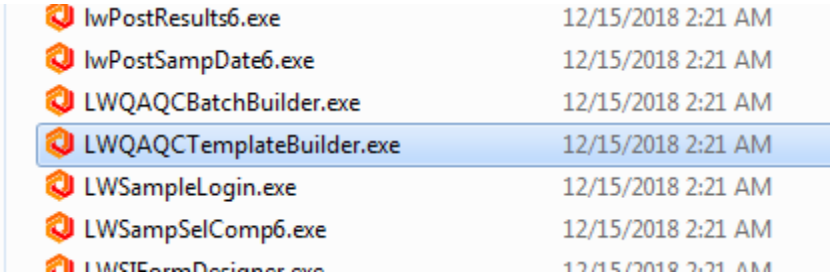
QA/QC Module Programs

If your lab has been using the legacy QA/QC program already, clicking the existing entry on the

LABWORKS desktop menu will continue to launch the old application even after installation. You will need to add a new link on your lab's Desktop for the latest application to launch (for both the QA/QC Template and Batch Builder applications) and remove the old tool.

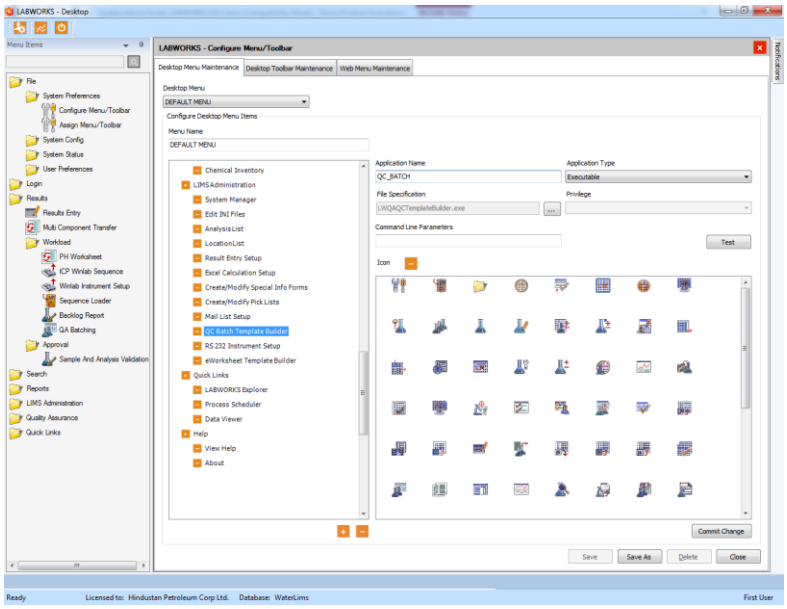
As with all LABWORKS LIMS programs, the QA/QC module programs are located in the **C:\LABWORKS\Client** path.

- **LWQAQCTemplateBuilder** – the QA/QC Template Builder program that designs the configuration to be used each time a specific batch of QA/QC samples are created in the system.
- **LWQAQCBatchBuilder** – the QA/QC Batch Builder program that creates the batches of QA/QC samples off of a given configurable template.



QA/QC Module Programs

Use the Configure Menu/Toolbar to change or remove the old Desktop menu option. Then you can publish the change to all the lab users using the Assign Menu/Toolbar option.



QA/QC Module Desktop Menu Option

No new control file to run the application is needed since the QA/QC module is a core LABWORKS application.

The LABWORKS.INI file, used by older legacy applications, is no longer used within the new QA/QC module. All configurations are done within the application itself or via System Manager keys.

QA/QC Application Privileges

The QA/QC Batch Builder program uses the same privileges as the legacy application. There is a privilege for execution of the application and a privilege for creation of batches. However, the new QA/QC Template Builder has 2 new privileges that must be added to the system.

- **QABATCHE_APP** – to run the batch builder (same as legacy)
- **BATCHSETUP** – for modification of existing batches (same as legacy)
- **QATEMP_APP** – to run the template builder
- **QATEMPSETUP** – to create new or modify existing templates

QA/QC Application Specifics

QA/QC Analyses Prefixes

The latest QA/QC Module supports the previous legacy QA/QC prefixes for test assignments. These are referred to as “special” analysis code prefixes.

These special analysis code prefixes are used by LABWORKS to designate common types of QA analyses that may be performed on a sample. Entry of these prefixes into the system is also the job of the System Administrator. These are entered in the System Manager area (included in the SysMgr database table).

There are built-in prefixes and custom prefixes. The custom prefixes can be used if the customer is not able to create all the required QA/QC tests needed for samples. The additional custom prefixes used in the system can be defined by **QAPREFIX_** in front of the System Manager key value (*SYSMGROPT* field in the SysMgr table).

The following are built-in prefixes.

- B_ – Blank Analysis
- A_ – Amount Spiked
- S_ – Spiked Result
- R_ – Recovery
- D_ – Duplicate Analysis
- P_ – Precision
- L_ – Lab Control Analysis
- I_ – Initial Calibration
- C_ – Continuous Calibration
- T_ – Trip Blank
- F_ – Field Blank
- #Q_ – Study Group

An example of when additional QA/QC prefixes may be needed is when more than one type of “L” test (Lab Control Analysis) is needed. The customer can define a _L1, _L2, _L3, etc. type of test all by inclusion of the QAPREFIX_ in front within SysMgr table. This will allow the additional prefixes to be available for usage and configuration in the system. You can have a maximum of 4 characters for a custom prefix. For example, a **QAPREFIX_TEST** entry in SysMgr would define the prefix **TEST**_ in the LIMS.

SYSMGROPT	SYSMGRVAL
QAPREFIX_BA	Blank Spk Amt
QAPREFIX_BR	Blank Spk Recovery
QAPREFIX_BS	Blank Spk
QAPREFIX_L3	LCS Recovery
QAPREFIX_LA	LCS Amt
QAPREFIX_P	Samp Dup Precision
QAPREFIX_ZZ	SOLUBLE

Example additional QA/QC prefixes defined in SysMgr

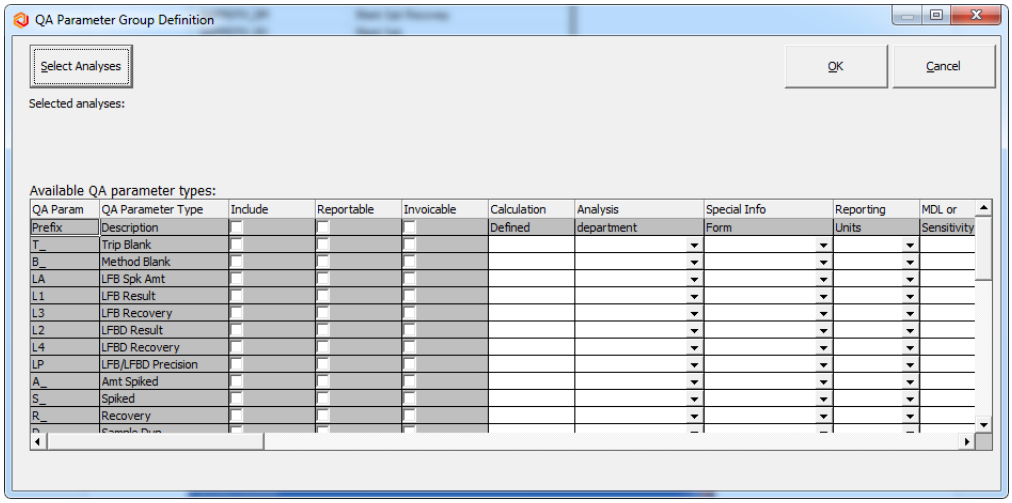
The purpose of defining prefixes is to handle the logged in QA/QC analyses in specific ways that may be slightly different from other regular analyses in LABWORKS. To facilitate the creation of these QA/QC test codes, the parameter group definition utility (**QCCFG6.EXE**) can be used.

The following section explains the usage of this QA/QC test group configuration utility.

QA/QC Test Group Configuration (QCCFG6)

In order to configure a QA/QC test group (#Q study group) with prefixes you defined using the SysMgr table (for custom specs) or any built-in prefix group, you can use the ***Parameter Group Definition*** utility. This program will save you a lot of time by creating all the required #Q and components for you from 1 simple configuration screen. It can be launched in the regular programs folder. The filename is **QCCFG6.EXE**.

Launching the utility will open a list of all standard built-in and custom prefixes available in the system. The columns in the list allow definition of special handling when samples of these types are logged into the system through their associated #Q group.



QA/QC Parameter Group Definition Utility

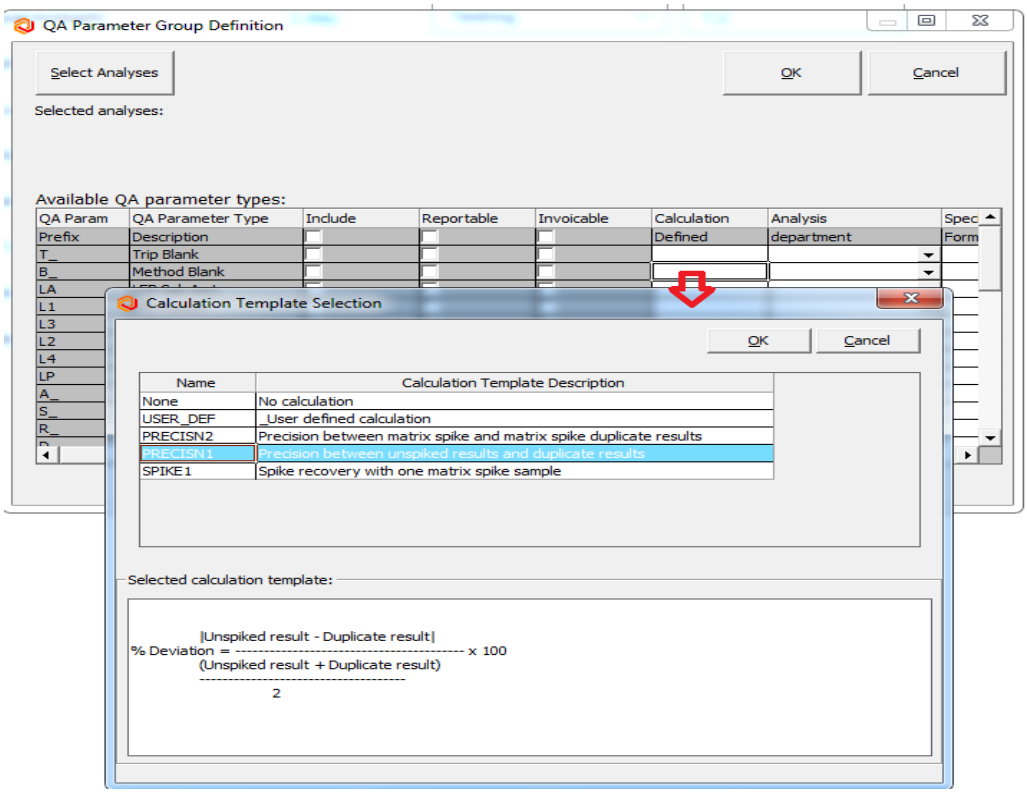
You must first select the analyses to configure above using the **Select Analyses** button at the top portion of the screen. You then define the individual QA parameter groups (rows) for all analyses logged in the system with these prefixes. Each row is a specific parameter group (prefix) defined. The built-in

groups are always towards the top, and the custom groups are placed at the bottom of the list. T he data entered in these cells will go into the automatic creation of the #Q group and its components.

The following are the available configuration options (columns) for each.

- **Include** – allows selection of which groups to include (to be saved) when OK is clicked. Only the groups checked to be included will have test codes created for them.
- **Reportable** – new tests of these groups will be marked reportable
- **Invoicable** – new tests of these groups will be marked invoicable
- **Calculation Defined** – allows the selection and definition of a specific calculation that will be assigned to the new tests of these groups.

Click into the cell to select or deselect a value.



Configuration Columns 1-4

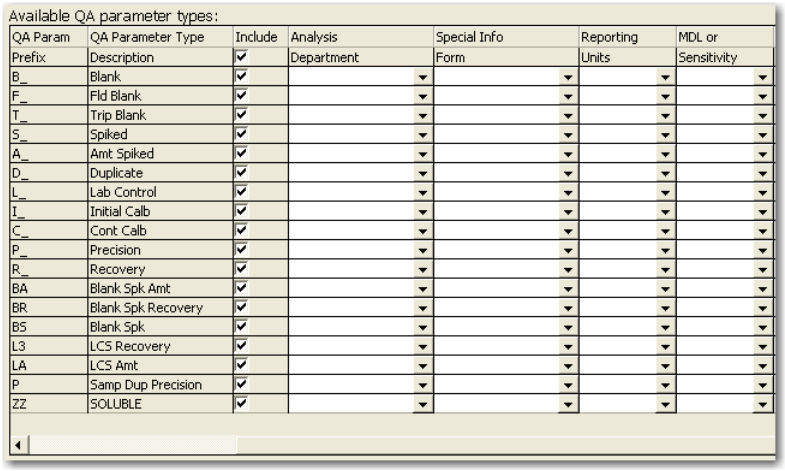
Click the arrow keys at the bottom of the screen to scroll over to the next set of columns.

Note that any of these columns can be set to a value of **_ANALYSIS** (included in the combo box) which selects whatever value is already assigned to the analysis code.

- **Analysis Department** – allows selection of a specific department that will be assigned to the new tests of these groups.
- **Special Info Form** – allows selection of a specific SI form that will be assigned to the new tests of these groups.
- **Reporting Units** – allows selection of a specific unit of measure for reporting that will be

assigned to the new tests of these groups.

- **MDL or Sensitivity** – allows selection of a specific MDL value that will be assigned to the new tests of these groups.

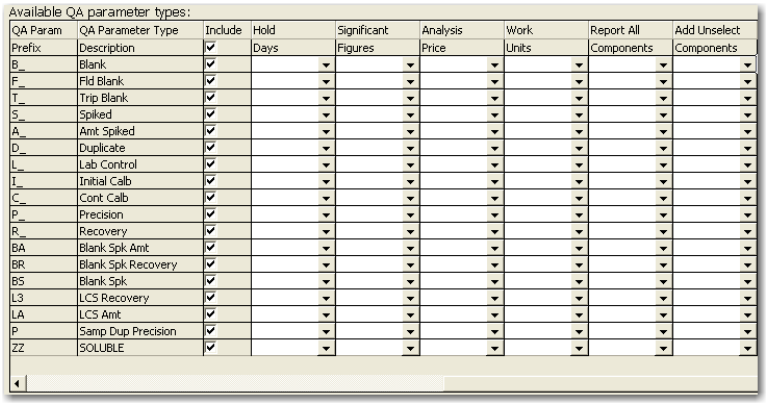

Available QA parameter types:

QA Param Prefix	QA Parameter Type Description	Include	Analysis Department	Special Info Form	Reporting Units	MDL or Sensitivity
B_	Blank	☑				
F_	Fld Blank	☑				
T_	Trip Blank	☑				
S_	Spiked	☑				
A_	Amt Spiked	☑				
D_	Duplicate	☑				
L_	Lab Control	☑				
I_	Initial Calb	☑				
C_	Cont Calb	☑				
P_	Precision	☑				
R_	Recovery	☑				
BA	Blank Spk Amt	☑				
BR	Blank Spk Recovery	☑				
BS	Blank Spk	☑				
L3	LCS Recovery	☑				
LA	LCS Amt	☑				
P	Samp Dup Precision	☑				
ZZ	SOLUBLE	☑				

Configuration Columns 5-8

Once again, click the arrow keys at the bottom of the screen to scroll over to the next set of columns.

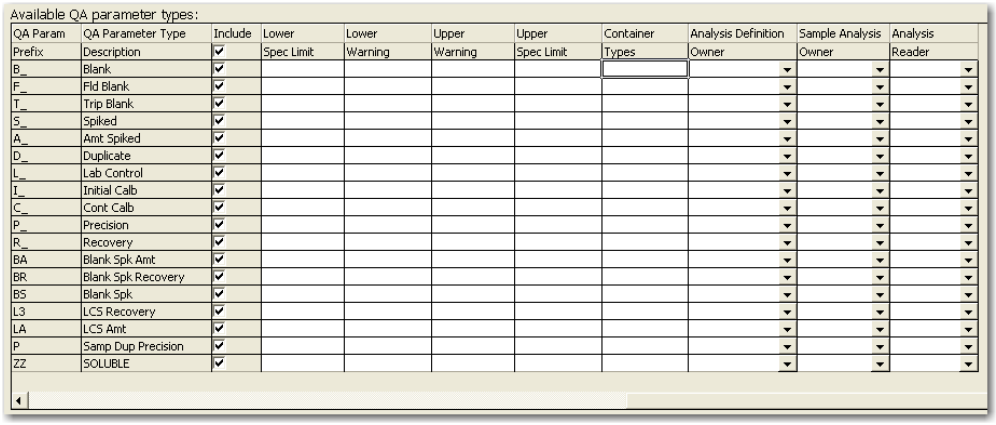
- **Hold Days** – allows selection of a specific number of hold days that will be assigned to the new tests of these groups.
- **Significant Figures** – allows selection of a specific number of significant figures for test results that will be assigned to the new tests of these groups.
- **Analysis Price** – allows selection of a specific price that will be assigned to the new tests of these groups.
- **Work Units** – allows selection of a specific “work” unit of measure that will be assigned to the new tests of these groups.
- **Report All Components** – a Boolean yes/no value that shows whether you want all components of the test reported or not and will be assigned to the new tests of these groups.
- **Add Unselect Components** – a Boolean yes/no value that shows whether you want to add unselected components to the login group or not and will be assigned to the new tests of these groups.


Available QA parameter types:

QA Param Prefix	QA Parameter Type Description	Include	Hold Days	Significant Figures	Analysis Price	Work Units	Report All Components	Add Unselect Components
B_	Blank	☑						
F_	Fld Blank	☑						
T_	Trip Blank	☑						
S_	Spiked	☑						
A_	Amt Spiked	☑						
D_	Duplicate	☑						
L_	Lab Control	☑						
I_	Initial Calb	☑						
C_	Cont Calb	☑						
P_	Precision	☑						
R_	Recovery	☑						
BA	Blank Spk Amt	☑						
BR	Blank Spk Recovery	☑						
BS	Blank Spk	☑						
L3	LCS Recovery	☑						
LA	LCS Amt	☑						
P	Samp Dup Precision	☑						
ZZ	SOLUBLE	☑						

Configuration Columns 9-14

Once again, click the arrow keys at the bottom of the screen to scroll over to the next set of columns. The remaining columns 15 through 22 define any special specifications (lower through upper specs) that apply to these groups or specific owners (analysis and sample owners).

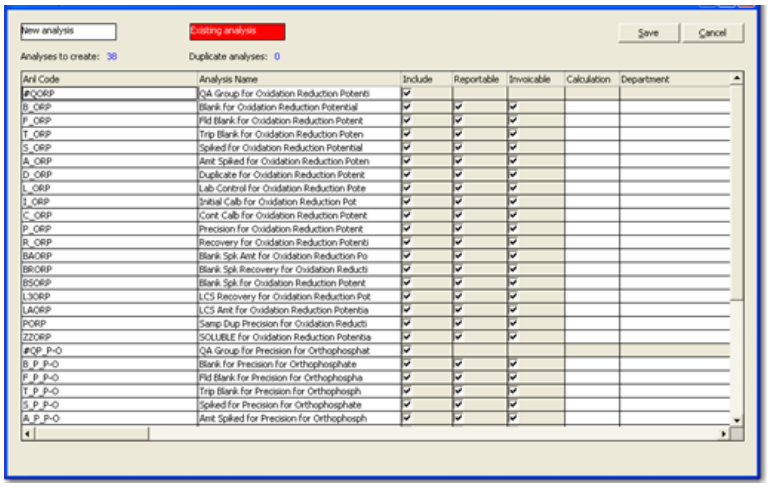

Available QA parameter types:

QA Param	QA Parameter Type	Include	Lower	Lower	Upper	Upper	Container	Analysis Definition	Sample Analysis	Analysis
Prefix	Description	☑	Spec Limit	Warning	Warning	Spec Limit	Types	Owner	Owner	Reader
B_	Blank	☑								
F_	Fld Blank	☑								
T_	Trip Blank	☑								
S_	Spiked	☑								
A_	Amt Spiked	☑								
D_	Duplicate	☑								
L_	Lab Control	☑								
I_	Initial Calb	☑								
C_	Cont Calb	☑								
P_	Precision	☑								
R_	Recovery	☑								
BA	Blank Spk Amt	☑								
BR	Blank Spk Recovery	☑								
BS	Blank Spk	☑								
L3	LCS Recovery	☑								
LA	LCS Amt	☑								
P	Samp Dup Precision	☑								
ZZ	SOLUBLE	☑								

Configuration Columns 15-22

When finished with the configuration of each of the group rows, click **OK** to save them or **CANCEL** to exit without saving.

After clicking OK, the system shows you the new QA parameters it is about to create for a final confirmation.

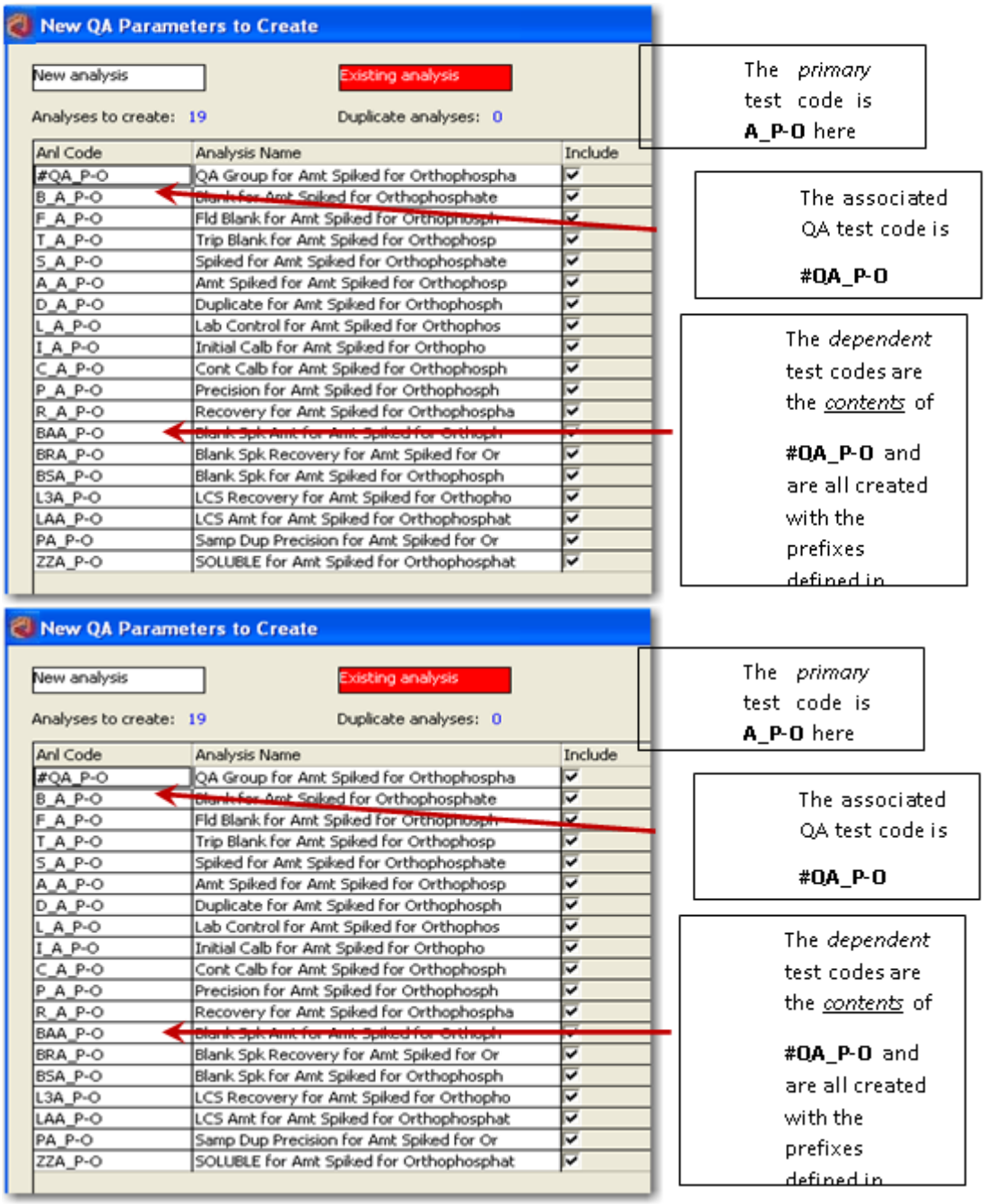

New analysis | Existing analysis | Save | Cancel

Analyses to create: 38 Duplicate analyses: 0

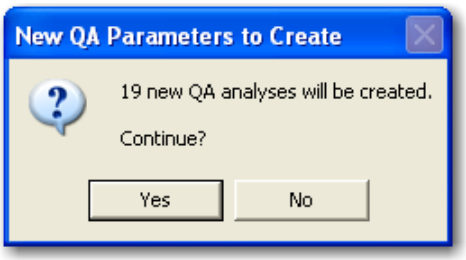
Anl Code	Analysis Name	Include	Reportable	Invoicable	Calculation	Department
#QORP	QA Group for Oxidation Reduction Potenti	☑				
B_ORP	Blank for Oxidation Reduction Potential	☑	☑	☑		
F_ORP	Fld Blank for Oxidation Reduction Potent	☑	☑	☑		
T_ORP	Trip Blank for Oxidation Reduction Poten	☑	☑	☑		
S_ORP	Spiked for Oxidation Reduction Potential	☑	☑	☑		
A_ORP	Amt Spiked for Oxidation Reduction Poten	☑	☑	☑		
D_ORP	Duplicate for Oxidation Reduction Potent	☑	☑	☑		
L_ORP	Lab Control for Oxidation Reduction Pote	☑	☑	☑		
I_ORP	Initial Calb for Oxidation Reduction Pot	☑	☑	☑		
C_ORP	Cont Calb for Oxidation Reduction Potent	☑	☑	☑		
P_ORP	Precision for Oxidation Reduction Potent	☑	☑	☑		
R_ORP	Recovery for Oxidation Reduction Potenti	☑	☑	☑		
BAORP	Blank Spk Amt for Oxidation Reduction Po	☑	☑	☑		
BRORP	Blank Spk Recovery for Oxidation Reducti	☑	☑	☑		
BSORP	Blank Spk for Oxidation Reduction Potent	☑	☑	☑		
L3ORP	LCS Recovery for Oxidation Reduction Pot	☑	☑	☑		
LAORP	LCS Amt for Oxidation Reduction Potentia	☑	☑	☑		
PORP	Samp Dup Precision for Oxidation Reducti	☑	☑	☑		
ZZORP	SOLUBLE for Oxidation Reduction Potentia	☑	☑	☑		
#QP_P-O	QA Group for Precision for Orthophosphat	☑				
B_P_P-O	Blank for Precision for Orthophosphate	☑	☑	☑		
F_P_P-O	Fld Blank for Precision for Orthophospha	☑	☑	☑		
T_P_P-O	Trip Blank for Precision for Orthophosph	☑	☑	☑		
S_P_P-O	Spiked for Precision for Orthophosphate	☑	☑	☑		
A_P_P-O	Amt Spiked for Precision for Orthophosph	☑	☑	☑		

New QA Parameters to Create

This utility will basically save you time from the manual creation of the same QA/QC tests in Analysis List Maintenance by building the primary and dependent analyses all for you automatically. For example, if the A_P-O test code was assigned and configured, the utility will build the special #Q group code and all associated analysis codes for inclusion.



Clicking **SAVE** on this final screen will generate the all the codes for you. If you want some created, but not others, you can use the **Include** column to unselect specific codes from the list.



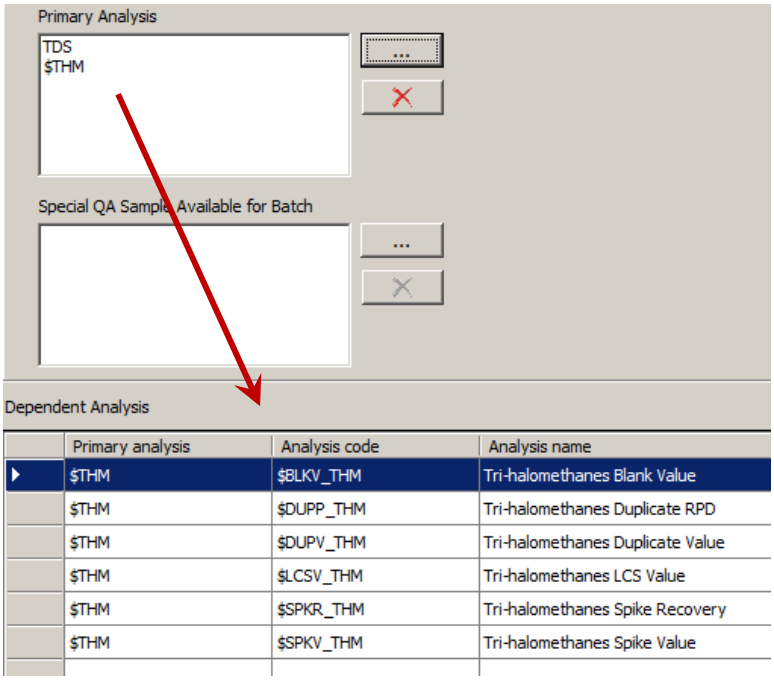
New QA Parameters to Create

The following section explains further on the setup and usage of the primary and dependent analyses.

QA/QC Primary and Dependent Analyses

Using the system prefixes above will allow users to follow standard QA/QC analysis code abbreviations within the system. In a QA/QC template, a primary analysis code can be assigned with a list of dependent analyses associated with it. In order to associate a QA/QC dependent analysis list, you must create a test group with the same analysis code with a #Q in front.

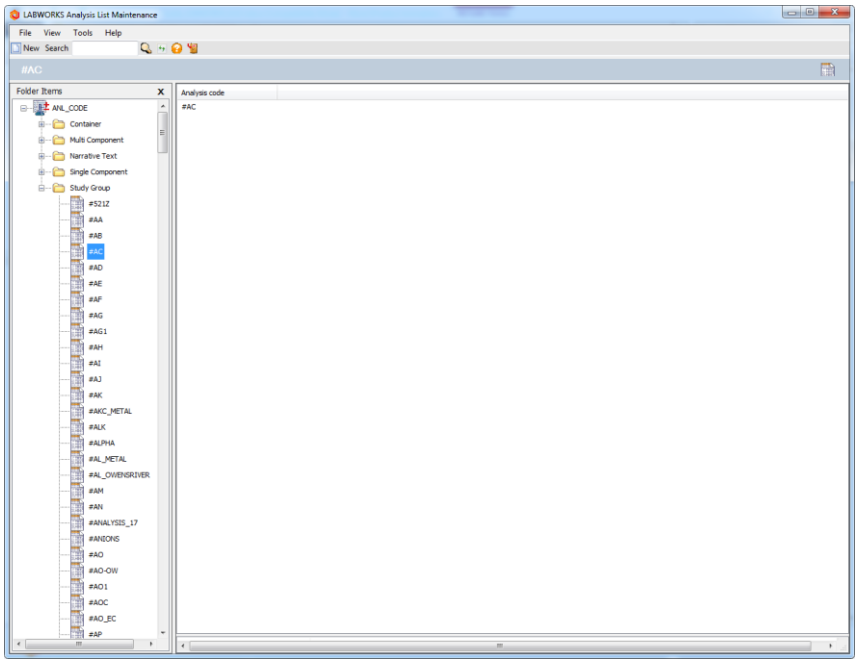
To provide a simple illustration of how the "dependent" #Q tests work, let's take a real-world example. If we have an analysis code for Tri-halomethanes ($THM) defined in LABWORKS, we can associate QA/QC-only tests by creating another test code called **#Q$THM**. The **#Q** in the front of this code tells the system that the contents of this study group test are *dependent analyses* specifically for **$THM**. Therefore, test **$THM** is the ***primary test*** while the contents of **#Q$THM** become the ***dependent tests***.



Example of #Q dependent tests within a QA/QC Template

This provides the capability for chemists to test QA/QC tests completely separate from the normal samples they run for the same test (does not conflict with the analytes already assigned to the test).

Define the QA/QC dependent study group lists from either within the ***Analysis Maintenance*** module or the ***QA Parameter Group Definition*** utility (**QCCFG6.EXE**) explained in the previous section.



LABWORKS Analysis List Maintenance - #Q group code example

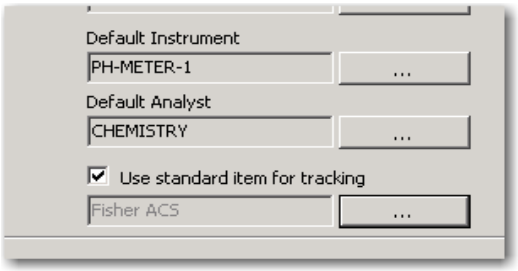
Optional Module Integration and SysMgr Keys

There are three optional LIMS modules that can be tied into the QA/QC Module.

- Chemical Inventory
- Instrument Maintenance and Calibration (IM&C)
- Personnel Training

These three modules include maintenance of other types of “objects” within LABWORKS that are able to be state managed, tracked, reported, etc. You are able to tie the objects from these modules into a given QA/QC sample batch by assigning them to a QA/QC Template.

On the far right-side of a QA/QC template is where the links for these optional modules are placed. An ellipse button to the right of each item will select these records from the optional modules.

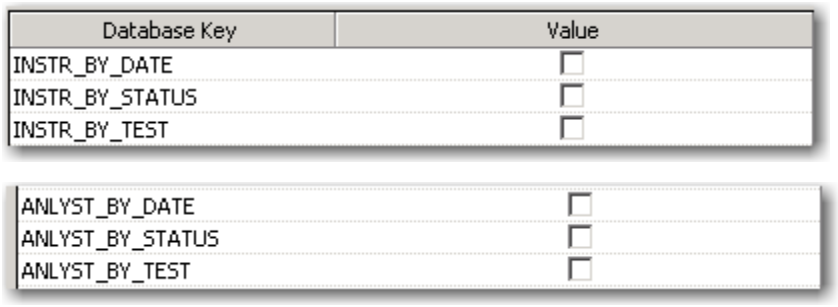


Optional Module Section in a QA/QC Template

The **Default Instrument** option opens a list of values taken from the *IM&C* module. The **Default Analyst** option opens a list of values taken from the *Personnel Training* module. The checkbox to **use standard item for tracking** links an item from the *Chemical Inventory* module.

The Chemical Inventory list will show the list of all inventory items you have available in the system. However, the instrument and analyst list can be filtered by the use of SysMgr keys.

The Boolean SysMgr keys that affect the list of instruments and analysts within the QA/QC module are as follows.



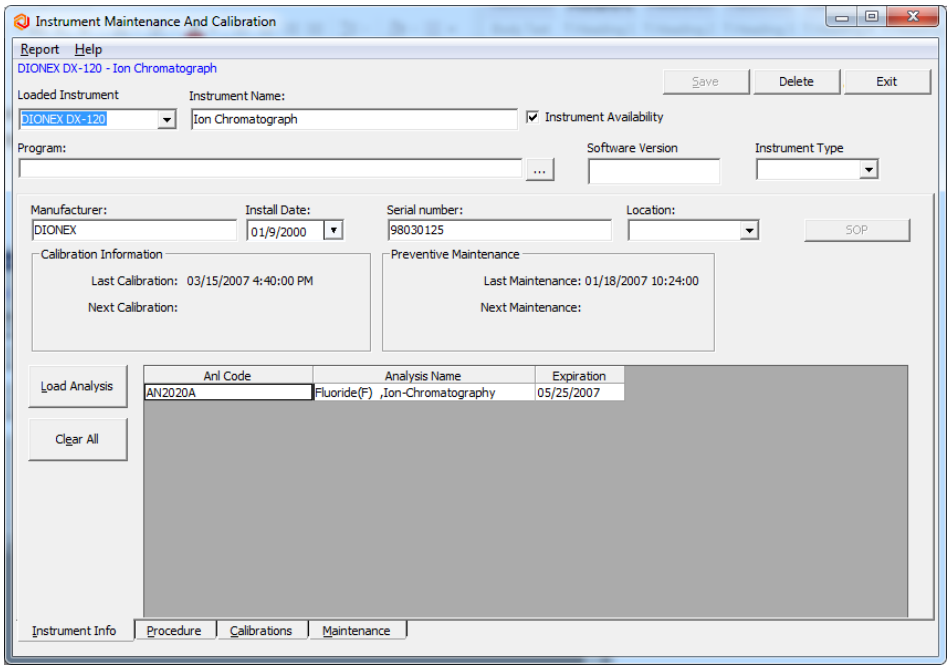
Database Key	Value
INSTR_BY_DATE	☐
INSTR_BY_STATUS	☐
INSTR_BY_TEST	☐

ANLYST_BY_DATE	☐
ANLYST_BY_STATUS	☐
ANLYST_BY_TEST	☐

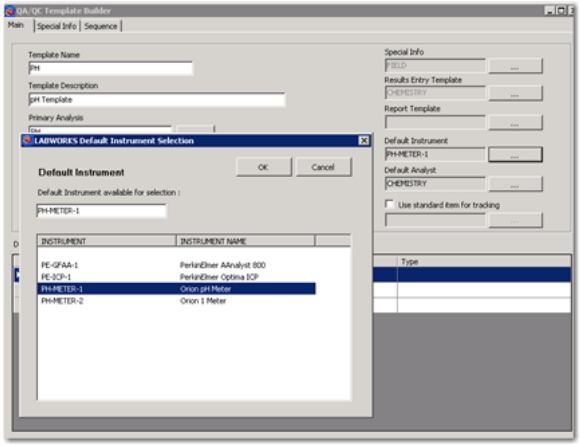
- **INSTR_BY_DATE** – shows a list of instruments filtered by install date
- **INSTR_BY_STATUS** – shows a list of instruments with data (calibration history) – will not show items that are not in active use
- **INSTR_BY_TEST** – shows a list of instruments with assigned tests that match the QA/QC template's assigned test – will prevent users from assigning instruments to a QA/QC template that it is not designated or properly calibrated to test

- **ANLYST_BY_DATE** – shows a list of analysts filtered by hire date
- **ANLYST_BY_STATUS** - shows a list of analysts with training data (history that shows training happened) – will not show users that were not trained
- **ANLYST_BY_TEST** – shows a list of analysts with assigned tests that match the QA/QC template's assigned test – will prevent users from taking on QA/QC testing they were not specifically trained for

To illustrate how this can be utilized, we can take the following example. With INSTR_BY_STATUS turned on, an instrument that was never calibrated or had any data associated with it would not be selectable.

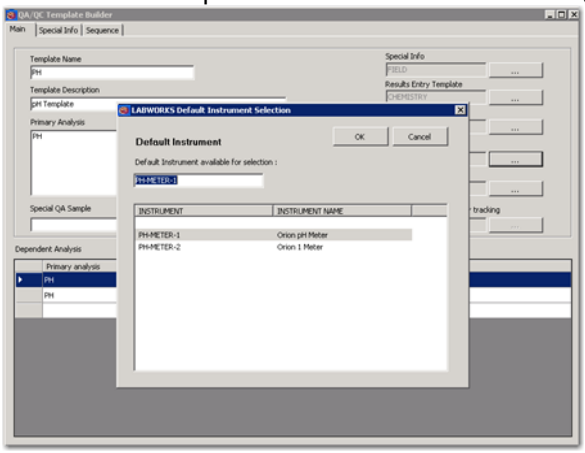


PE-GC-1 Instrument has no data associated with it



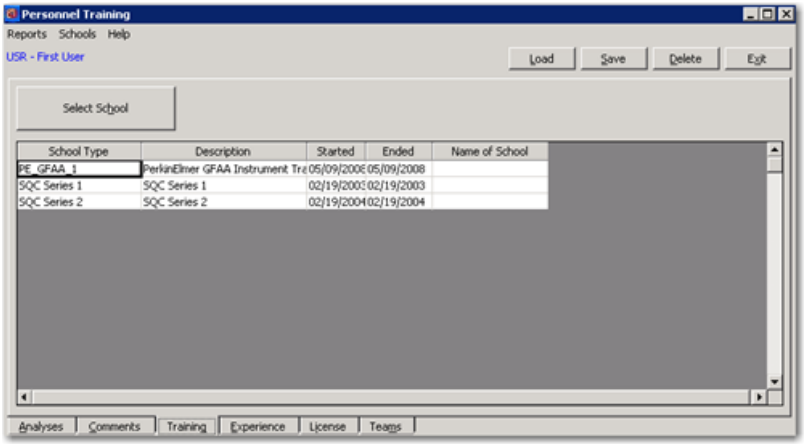
The PE-GC-1 Instrument is not selectable within QA/QC

Another similar example is if we had INSTR_BY_TEST turned on and was sampling for PH. When you are creating a QA/QC batching template that has a primary analysis of PH, only those instruments selected to sample PH will be selectable within QA/QC.



Only PE testing Instruments are selectable within QA/QC for PE

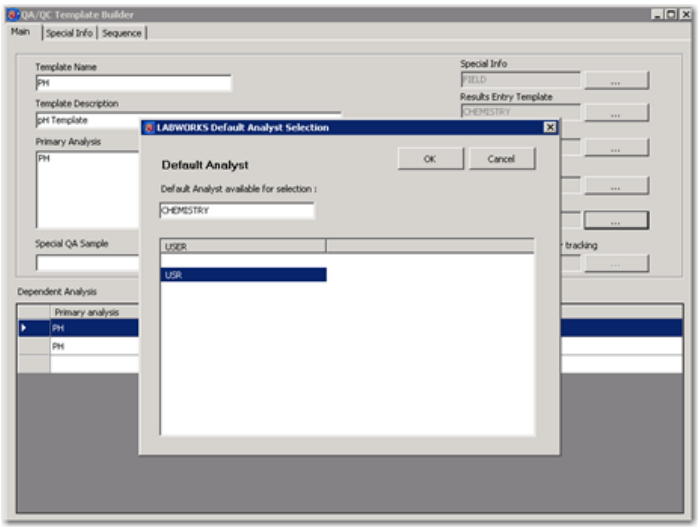
Likewise with Personnel Training if ANLST_BY_STATUS is turned on, only those users that have had training and in good status will be selectable within QA/QC.



Only the USR user has had all the proper training required

In our example, USR has had all the required training.

Opening a analyst list within a QA/QC Template will only show that one user who has had all the proper training required to do QA/QC testing.



Only the USR user has had all the proper training required

Also with analysts, you can assign tests as well and switch the filter to use ANLST_BY_TEST. In our previous example where we were sampling with PH, only those users trained to sample PH will be selectable.

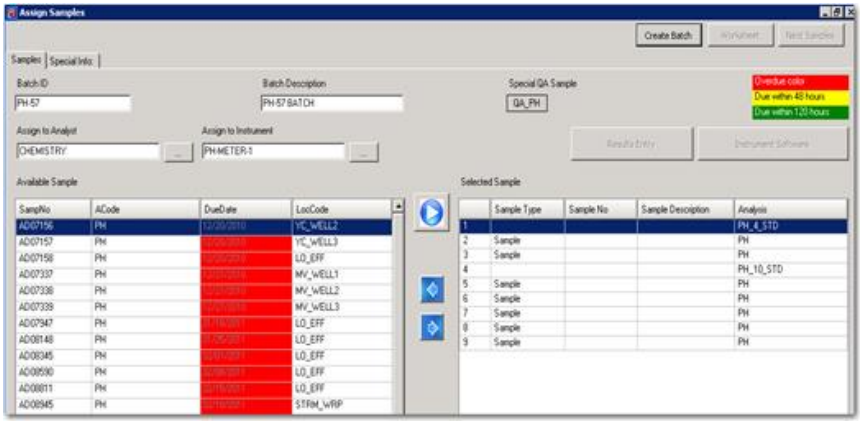
Other QA/QC SysMgr Keys

Besides the 6 filtering keys defined in the previous section, there are also 3 SysMgr keys that define the color scheme within the QA/QC Batch Builder program.

The following keys define the QA/QC Batch Builder color scheme. The color scheme is used to define which samples are due or overdue in an existing or new batch after the **Next** button is clicked from the main screen.

- WARN_DUE_COLOR – these are samples in the batch list that are close to becoming due (they will be due soon within 120 hours). The default is usually **green**.
- DUE_COLOR – these are samples due NOW or within 48 hours. The default is usually **yellow**.
- OVERDUE_COLOR – these are samples that are overdue or late. The default is usually **red**.

A legend of the color scheme can be viewed on the far right side of the batch.



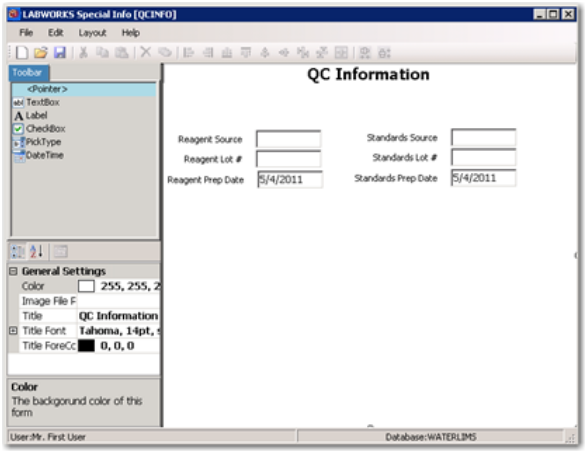
Color Scheme in QA/QC Batching

SI and User Fields in QA/QC

There are no user field requirements in the system. The system uses the user fields assigned to specific location codes as normal.

Special Info (SI) fields, however, can be assigned to the batch via the QA/QC Template. This is done using the **Special Info** selection on the **Main** tab of the template.

Special Info



Example QC Information SI Form

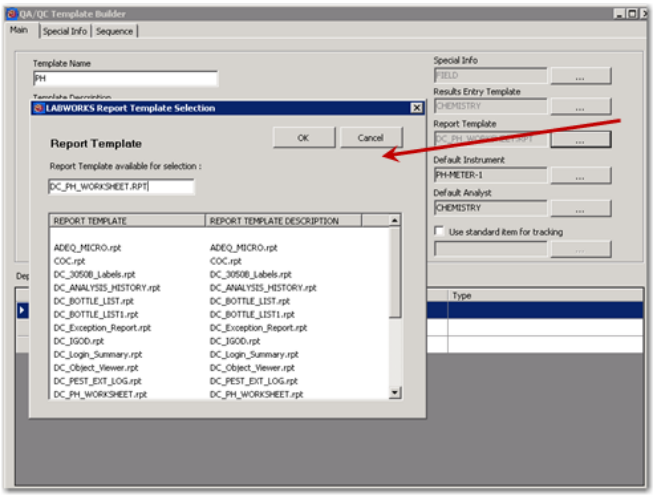
Default information can be entered within the SI fields of the assigned form. This information is then tracked at the *batch level*. This provides a more high-level form of special information tracking above *both* the analysis and sample levels that are stored at sample login.

QA/QC Crystal Worksheets

The Crystal worksheets work the same as they did in the legacy application. They are export reports that launch the EXPTDATA6 application for extraction of batch data (into export tables FLATDATA_QA and FLATDATA_QABATCH).

You must first have a valid QA/QC report with a CEF (Custom Export File) associated selected for exporting the QA/QC data as an option.

Once you have a valid report, and it is in an ACTIVE state in Object Publisher, it can be selected within a QA/QC batching template.



QA/QC Template Report selection

If a valid report was assigned to the QA/QC Template, when the batching is performed a **Worksheet** button will be highlighted within the QA/QC Batch Builder application. The user can click this button to fire off the worksheet.

Note that this is designed only for basic worksheet information only. For more detail (sample result information, for example), it may be best to select a custom report from the LABWORKS Desktop instead. However, if results are entered from the batch builder (by launching Results Entry), the selected template should reflect this data immediately if stored. How your worksheet performs depends on your report template built within the LABWORKS Report Designer tool.

Common QA/QC Problems

As part of the final section of this document, we wanted to provide some support assistance by listing the most common errors on QA/QC turned into support. These can be used as a means to catch problems before they occur.

Issue #1: no analysts in my selection list – you may have filtering turned on, so check your SysMgr keys.

Issue #2: no instruments in my selection list – you may have filtering turned on, so check your SysMgr keys.

Issue #3: batches were not created properly or template is not selectable – check and make sure your template is in an ACTIVE state in Object Publisher.

Issue #4: can't add items to chemical inventory for selection in QA/QC – you must have Chemical Inventory set up properly. It has its own set of SysMgr keys, privileges, and a set up process. Check your Chemical Inventory manual.

Issue #5: no color scheme in my batches or colors are invalid – check your SysMgr settings for the keys WARN_DUE_COLOR, DUE_COLOR, and OVERDUE_COLOR and make sure they are correct.

Issue #6: my worksheets won't print – make sure your worksheet is a QA/QC report template meaning it is pointing to the QA/QC tables in LABWORKS. You can test this by opening the LABWORKS Report Designer and selecting the template in design mode. When the template is open, check the database connection information.

Issue #7: no worksheets are selectable in my QA/QC template – make sure the report template is in an ACTIVE state in Object Publisher.

For other support related issues, contact LABWORKS Support.

5. QA/AC Module Setup and Configuration Reference

Overview and Setup

Introduction

The Quality Assurance / Quality Control (QA/QC) application for creating batch samples for the evaluation of quality has been an integral part of LABWORKS LIMS for many years. With the release of LABWORKS 6.4, a new QA/QC batch sampling application was introduced that encompasses a new updated user interface and feature-rich design that was the result of many years of customer feedback. The new application, designed using the latest LABWORKS architecture and Microsoft .NET Framework, is now a powerful tool in quality control sampling.

Quality control is important for several reasons. Laboratories need to make sure that certain criteria are met to ensure good data is reported. Whether a lab is regulated by ISO standards or needs good documentation for customers to ensure sample quality, a good comprehensive QA/QC process is always beneficial. QA/QC testing helps to catch potential issues such as instrumentation calibration issues, problems with received samples, or issues in sampling procedures to name a few. The latest QA/QC module is better integrated into other LABWORKS applications to better track this data over the previous legacy tool. For example, QA/QC batches can be assigned to specific instruments or equipment from the Chemical Inventory module or Instrument Maintenance and Calibration. Also specific employees from the Personnel Training module can be assigned to a batch template. Special Information forms can be assigned to track specific custom fields for batches as well.

This chapter was written with the end-user in mind and will cover everything from initial setup and basic configuration for daily usage. For technical documentation please reference the QA/QC Technical Reference chapter.

Module Overview

The process of “batching” consolidates and simplifies sample handling, quality control testing, and results entry for logged samples. You can group samples into a batch for a particular analysis, designate a QA/QC sample for the batch, and automatically assign a user-defined group of QA/QC tests to that sample’s work order. From the main QA/QC Batching screen, you have the option to create, modify, delete, and/or print QA/QC batches. The module provides two separate applications to handle this process.

QAQCTemplateBuilder.exe

Sets up templates to be used in batching

Defines specific details of how a batch will be done

QAQCBatchBuilder.exe

Creates sample batches for QA/QC

Performs specific assignments that were defined in the template

Enter the specifics for your QA/QC batch by creating a “template” first. This template is where you define specific information on how the batch will be processed later. You can define special information (custom fields), define analyses, specific results-entry and reporting templates, default instruments or analysts, or even link to an item defined in your Chemical Inventory application (sold separately). All the information in the template will be used when you create your batch later.

After creating a QA/QC batch you have a number of capabilities available to you in LABWORKS such as worksheet printing, batch results entry, automatic calculations for recovery and precision (with limit checking), etc. The QA/QC Report function also allows you to generate and print a detailed report that displays information for all QA/QC results associated with selected batched samples. Crystal report templates can be used to design your own reports as well.

Before a user-defined group of QA/QC tests can be assigned to a QA/QC sample, analysis codes must be set up. You can set up an analysis code by using the **Analysis List Maintenance** module under the **Maintenance** menu of the LABWORKS Desktop (see the LABWORKS User Manual or Help for more details). The System Administrator is generally the person who sets up analysis codes, so this is beyond the scope of this chapter. If your analysis codes for QA/QC purposes are not yet entered in the system, please consult your System Administrator.

Special analysis code prefixes are used by LABWORKS to designate common types of QA analyses that may be performed on a sample. Entry of these prefixes into the system is also the job of the System Administrator. See the following section for more information on these special analysis codes.

QA/QC Common Terminology

The following are common QA/QC terms used in both LABWORKS LIMS and in the industry of Quality Assurance sampling.

LABWORKS LIMS Specific Terms

Special analysis code prefixes – these are used by LABWORKS to designate common types of QA analyses that may be performed on a specific sample. These prefixes are entered in the System Manager area of the system by the Administrator.

These are the prefixes used in the system.

B_ – Blank Analysis

A_ – Amount Spiked

S_ – Spiked Result

R_ – Recovery

D_ – Duplicate Analysis

P_ – Precision

L_ – Lab Control Analysis

I_ – Initial Calibration

C_ – Continuous Calibration

T_ – Trip Blank

F_ – Field Blank

#Q_ – Study Group

Sample Designation – provides sample selection by entering specific LABWORKS sample numbers, location codes, or first user defined fields in the text field on the screen.

Login record file – a sample selection login batch number for designation of a specific group of samples which were logged in and available for QC sample designation.

Other file – a sample selection option which provides a designated text file containing a list of valid LABWORKS sample ID numbers.

Cross Reference – a sample selection option which provides a list of samples found by the user's most recent cross reference search. This option is only active if a Cross Reference Search has previously been performed by the current user.

QA/QC Batch – a sample batch designation related to a specific QC sample set. A QA/QC batch can also be used throughout LABWORKS during sample selection to retrieve this batch as needed. Batch retrieval can be used to modify those sample batches at will or for reporting purposes.

Unbatched Pending – refers to a list of analyses that are not currently batched.

Pending Batched – refers to a list of analyses that are currently batched.

Completed – refers to a list of analyses that have already been analyzed from the batch. These analyses are finished.

Analyses assigned to samples – refers to all the analyses assigned to the samples within the QA/QC batch.

Analyses from selected department – refers to all the analyses assigned to a specific department within the QA/QC batch.

Clone batch – refers to the process of duplicating batches and using QA specifications for more than one sample. This can be particularly important if you have a small sample set.

Assign instrument for batch – the process of assignment of instruments to a specific QA/QC batch which can later reference additional data from the Instrument Maintenance and Calibration module.

Assign analyst for batch – the process of assignment of an analyst to a specific QA/QC batch which can later reference additional data from the user Training module.

Login special QA sample – refers to the selection of a "special" QA sample that will automatically assign special analyses as defined previously.

Range of batch dates – refers to batches created within a specified date range

Range of batch numbers – refers to batches from a specified range of batch numbers

On-line QA/QC batches – refers to "available" or active QA/QC batches. Batches can be marked inactive where they are taken out of the QA/QC test loop. These on-line batches are considered within the workflow. They are in the system to be analyzed and reported.

See section at the end of the chapter to see many more common terms related to QA/QC.

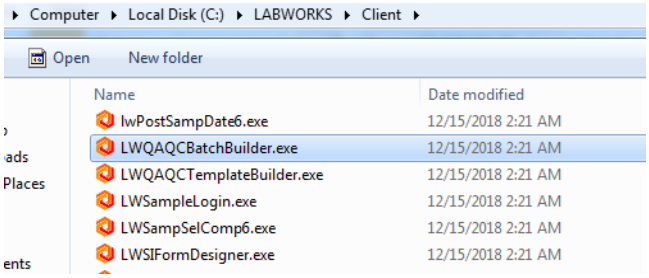
QA/QC Application Information

To use the latest QA/QC module, you must first be running at least LABWORKS 6.4. The LABWORKS 6.4 installation process installs the new QA/QC module programs. No additional installation is required. The System Administrator will be responsible for installing the proper LABWORKS server and client applications, running any needed service packs and database scripts, and configuring the system for laboratory use.

Launching the Module

If you are using the legacy QA/QC program already, clicking the existing entry on your desktop menu will cause the old application to launch. You will need to add a new link on your LABWORKS Desktop for the latest application to launch.

Remember that the QA/QC Template Builder application is separate from the QA/QC Batch Creation utility. You will need two separate links on your desktop to run each program. These are located in the **C:\LABWORKS\Client** path.



QA/QC Module Programs

No new control file to run the application is needed since the QA/QC module is a core LABWORKS application.

Consult your System Administrator to add the new adjustments to your LABWORKS Desktop menu under the **Configure Menu/Toolbar** area.

The LABWORKS.INI file, used by older legacy applications, is no longer used within the new QA/QC module. All configurations are done within the application itself or via System Manager keys.

QA/QC Template Builder

Introduction

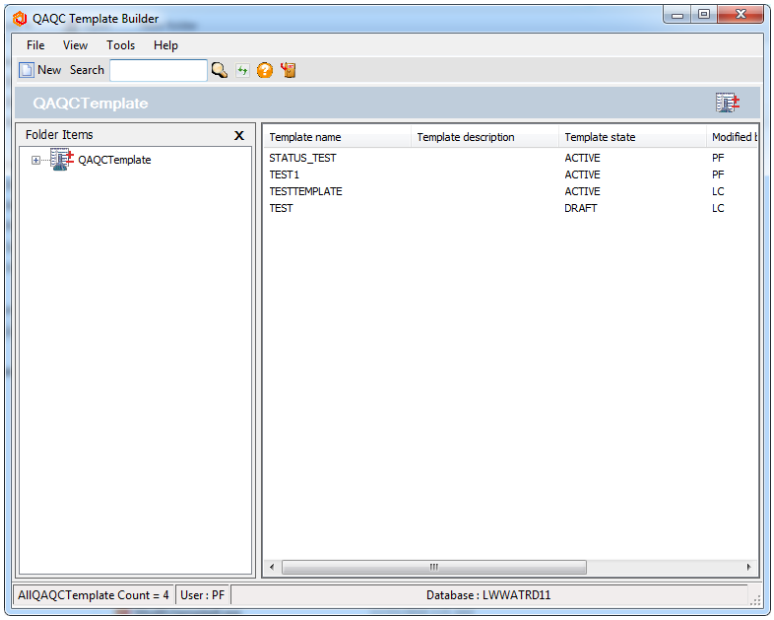
The purpose of the Template Builder application is for configuration of a QA/QC batch. In the previous legacy application, certain batch configuration detail was kept in the LABWORKS.INI file. The INI file is no longer used for QA/QC program configuration. The configuration of a QA/QC batch is exclusively done within the creation of a *batch template*.

The program filename is **LWQAQCTemplateBuilder.exe** and is located in the C:\LABWORKS\Client\ folder.

Note that all QA/QC batching templates created are *objects* in LABWORKS and can be state managed. Initially all templates are in DRAFT mode. If state management is turned on in LABWORKS, you must *approve* a batching template before it can be used.

Template Builder UI

The program's user interface (UI) should look familiar to users that have used any of the maintenance modules for analysis and location codes. It is a standard Windows-Explorer looking screen layout with a main menu across the top, a toolbar, a tree-view along the left with a configurable listing on the right. A status bar along the bottom of the screen displays the logged in user, database, etc.

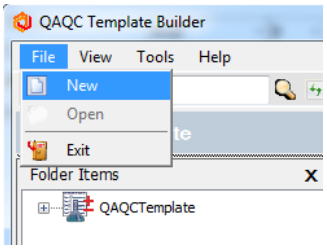


QA/QC Template Builder Main Screen

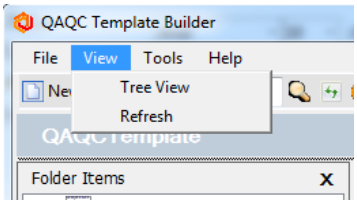
Main Menu

The main menu across the top of the main Template Builder screen includes the following.

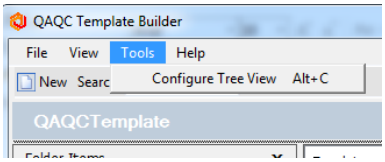
File Menu – click here to create New templates, Open existing templates, or Exit the application



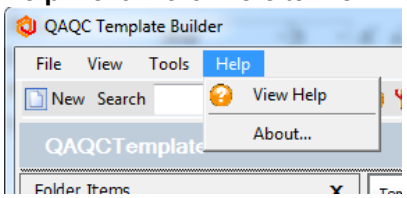
View Menu – click here to toggle the display of the tree-view or refresh the screen



Tools Menu – click here to configure the tree view

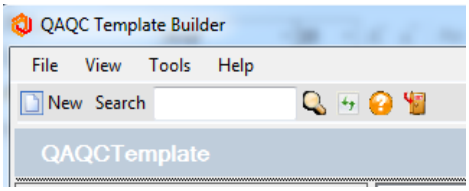


Help Menu – click here to **View Help** or show the **About** screen



Toolbar

The main screen toolbar is located below the main menu. It includes the following.



New button – create a new template

Search – search for a template in the list or tree

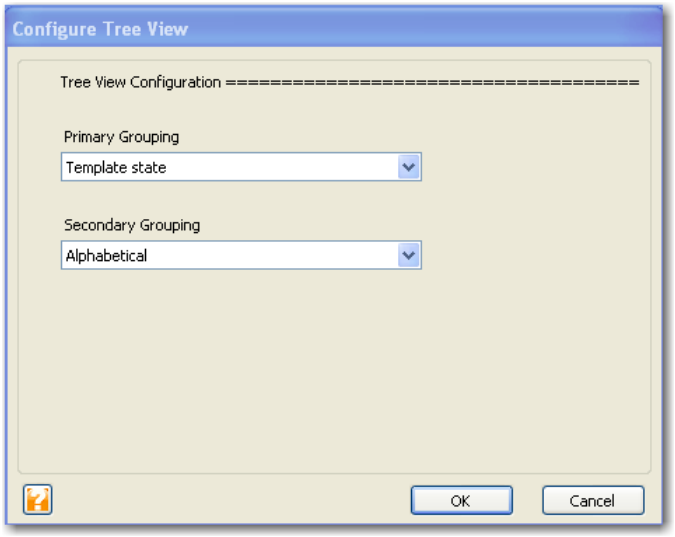
Refresh – refresh the screen for changes (will show changes other users have done recently on the network, for example)

Help – launch the help

Exit – exit the module

Template Builder UI Configuration

The tree view can be customized for displaying QA/QC templates either by state (draft, approved, etc.) or alphabetically by template name. This customization can be performed by selecting the **Configure Tree View** option under the Tools menu. The hot key of **Alt-C** can also be clicked to open the screen as well. The tree configuration screen is opened after selection.



Configure Tree View Screen

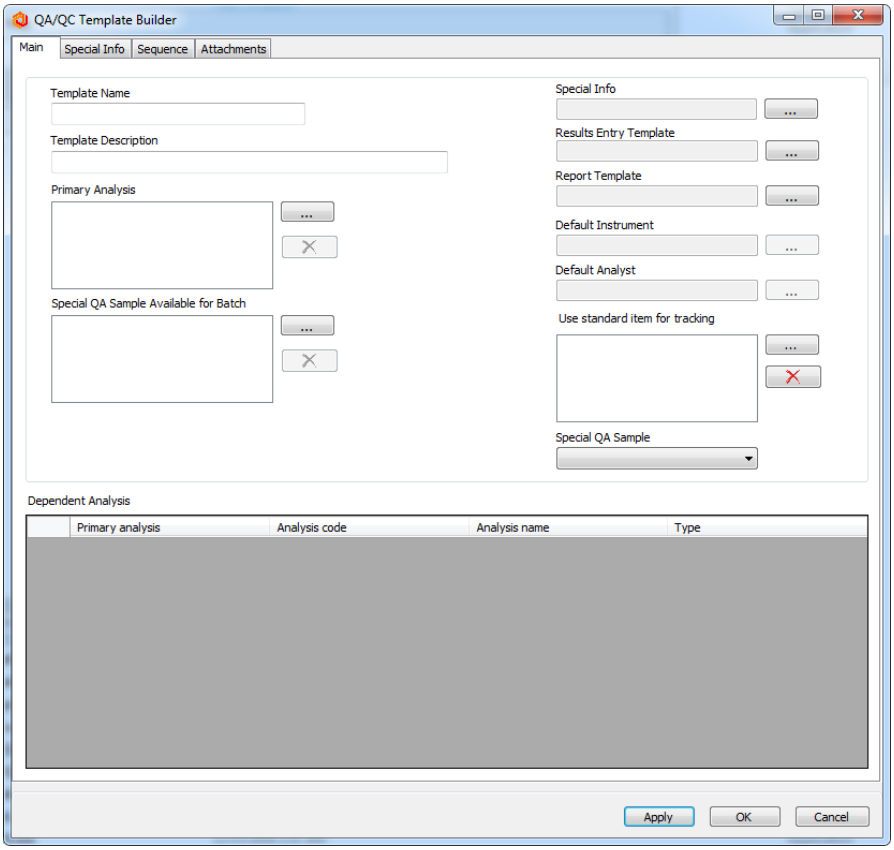
There are two levels allowed for display, **Primary** and **Secondary**, for displaying a tree listing of templates. The Primary refers to the first branch of the tree while the secondary is the sub-branch. Each branch can be sorted either by **template state** or **alphabetically**.

Select how to sort each branch, and then click **OK** to save and exit. Click **Cancel** to exit without saving.

Creating a New QA/QC Batching Template

The purpose of this application is to create a new template specifically for batching QA/QC samples. In order to do this we can select **New** under the **File** menu of the main screen, or click **New** from the **toolbar** as well.

This will open the new QA/QC Template screen.



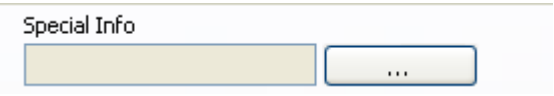
Initially the screen is empty with the cursor on the **Template Name** field. The template name field must be entered before anything else, and it is the key field for identification, searching and sorting from the main screen.

New QA/QC Template Screen

1. Enter the **Template Name** field first (required database primary key 40 characters in length). This value must be unique to only this template.

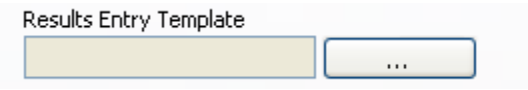
2. Enter the **Template Description** field (optional field 60 characters in length). This simply gives you more detail about this template and its purpose.

3. Enter the **Special Info** form (optional field). This defines which special info form to assign to this template for further custom information tracking specifically for each batch created with this template. Click the ellipse button to the right to select a special info form.

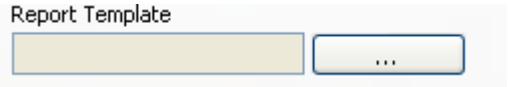


Note: use the Special Info Form Designer tool to create new Special Info forms.

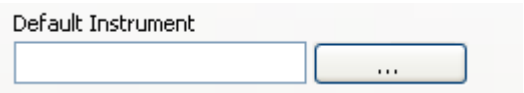
4. Enter the **Results Entry Template** (optional field). This defines what result entry custom form to use when entering results specifically for batches created with this QA/QC template. If one is not defined, the assigned default form will be used in the Results Entry module. Click the ellipse button to the right to select a Results Entry form template.



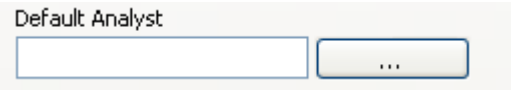
5. Enter the **Report Template** field (required field for worksheet printing). You will need to pre-configure a Crystal report worksheet template to be used here. Consult your System Administrator for report creation. This will require using the Report Designer module for template creation, then approval of the template for production use. Click the ellipse button on the right to select from available reports (that are in the *approved* state for usage).



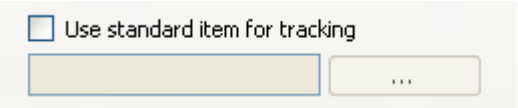
6. Enter the **Default Instrument** (optional field). This list is from the Instrument Maintenance and Calibration module (**IMandC6.exe**). Consult your System Administrator for usage of this module. Click the ellipse button on the right to select from available instruments (that are in the *approved* state for usage).



7. Enter the **Default Analyst** (optional field). This list is from the Personnel Training module (**Training6.exe**). Consult your System Administrator for usage of this module. Click the ellipse button on the right to select from available analysts.

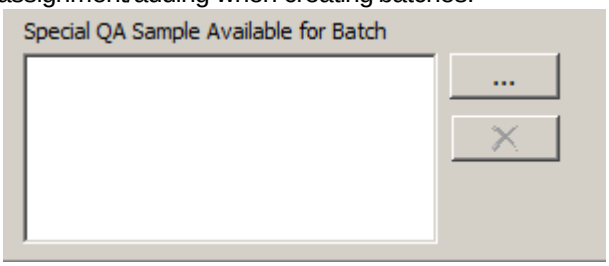


8. Toggle the **Use standard item for tracking** option (optional field) if using the Chemical Inventory module (**LWChemInventory.exe**). This will show a listing of chemical inventory for selection during batch creation if doing QA/QC sampling for inventory. Consult your System Administrator for usage of this module. Click the ellipse button on the right to select from available inventory items.

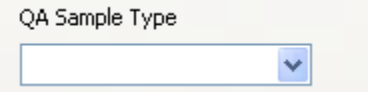


9. Enter any **Primary Analysis** (optional box) using the ellipse button to the right of box. You can enter as many analyses here as needed. The ellipse button will open a list of available analyses for selection from the Analysis List Maintenance module. The red **X** button will delete selected primary analyses from the box.

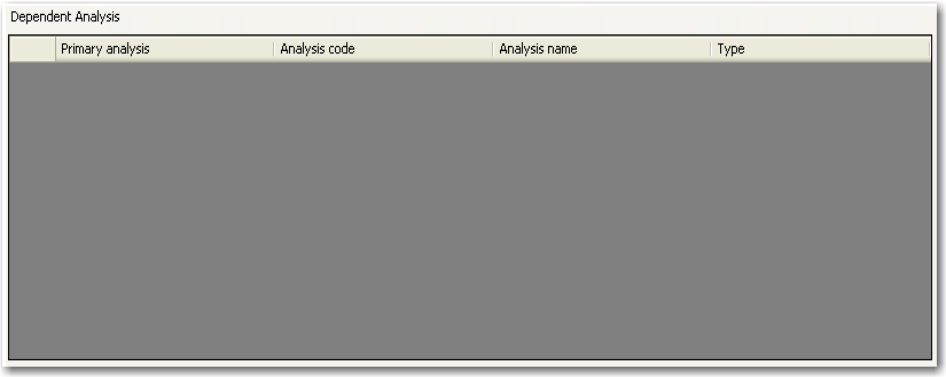
10. Special QA Sample Available for batch is a list of QA Location Codes that will be available for assignment/adding when creating batches.



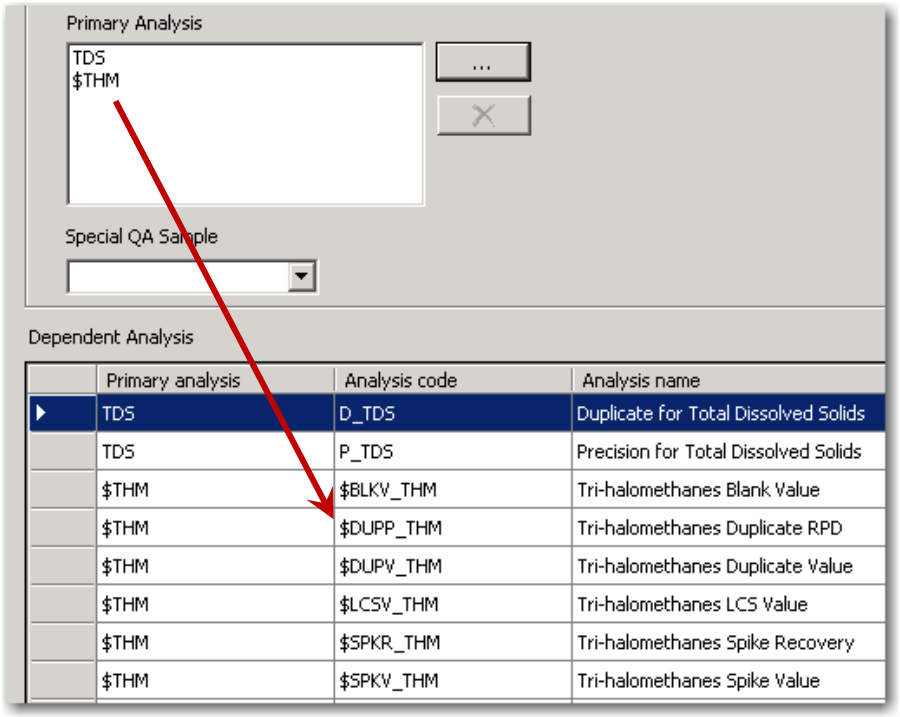
11. Enter a **QA Sample Type** (optional field). If you would like to assign a specific QA Location code (those beginning with 'QA_'), select it from this drop down menu. This will automatically login a new sample when a Batch is created.



12. Finally (on the initial tab of the screen), enter any Dependent Analyses in the box below (optional field). This section will be populated with the possible QA Analysis codes (those defined in the #Q Analysis study group) for the analyses selected in the '**Primary Analysis**' section.



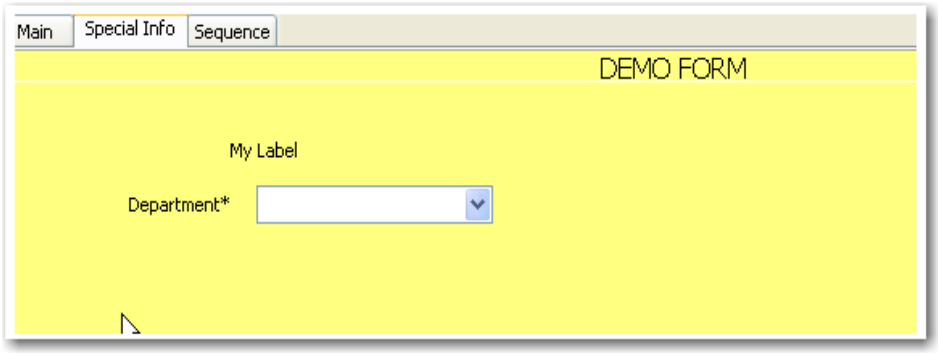
To provide a simple illustration of how the "dependent" #Q tests work, let's take a real-world example. If we have an analysis code for Tri-halomethanes ($THM) defined in LABWORKS, we can associate QA/QC-only tests by creating another test code called **#Q$THM**. The **#Q** in the front of this code tells the system that the contents of this study group test are *dependent analyses* specifically for **$THM**. Therefore, test **$THM** is the ***primary test*** while the contents of **#Q$THM** become the ***dependent tests***.



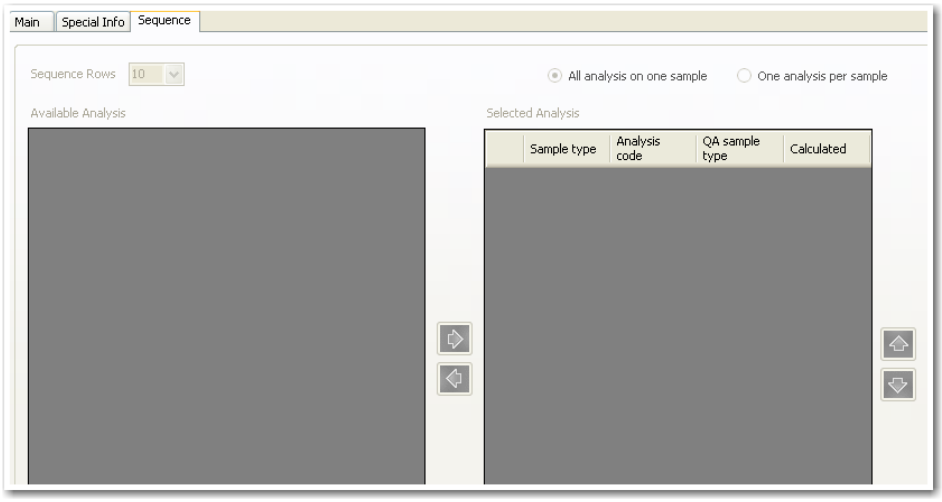
An example of #Q dependent tests

This provides the capability for chemists to test QA/QC tests completely separate from the normal samples they run for the same test (does not conflict with the analytes already assigned to the test).

13. Now click the **Special Info** tab (assuming a special info form was entered on the first tab) to enter any default values enter your assigned custom form.

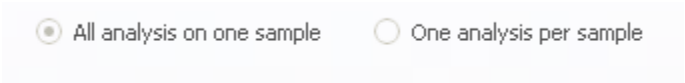


14. Finally, click the **Sequence** tab. This is where we will define the order of the analyses within the batch. Possible selections will be on the left, selected items will show on the right. The *arrow keys* in the center move analyses back and forth while the *arrow keys* on the right define the order.



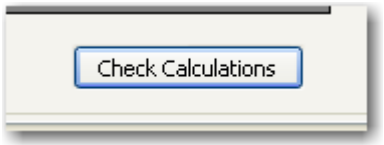
Note: Sequence Rows (top left) define the number of allowed samples in the batch.

There are two different options on the right side of this tab that defines how more than 1 analysis selected for the batch is processed. If '***All analysis on one sample***' is selected, then all of the selected analyses will be on a sample-by-sample basis. If '***One analysis per sample***' is selected, then each analysis will have its own record.

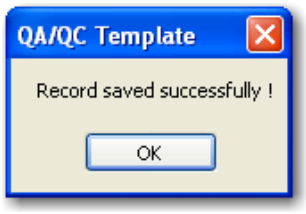


For example, let's assume a lab has defined all of their metals as single components. They used to run them on an AA, but are now running on an ICP. We can include all of the single component metals in the same batch and have all of the samples on the same line by choosing '***All analysis on one sample***'.

At the bottom right portion of this tab is a button to **Check Calculations** used if some of the selected analyses have calculations associated with them. It will fire the calculation and check for problems.

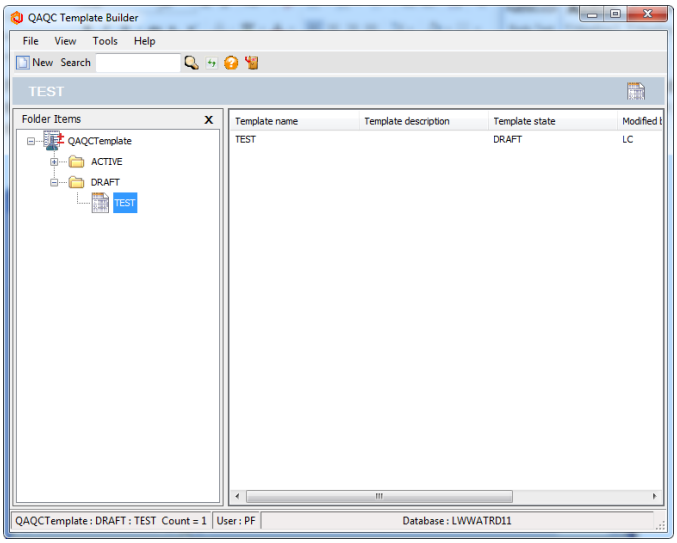


15. When finished entering data, click **Apply** to save or **OK** to save and exit. A message box should tell you the new QA/QC template was stored successfully to the database.



16. **Cancel** will exit without saving the new template change.

If you saved the new template successfully and exited, the new template will be displayed in the main screen tree-view. It will be displayed based on how you configured the tree in the previous section (by state or by name alphabetically).



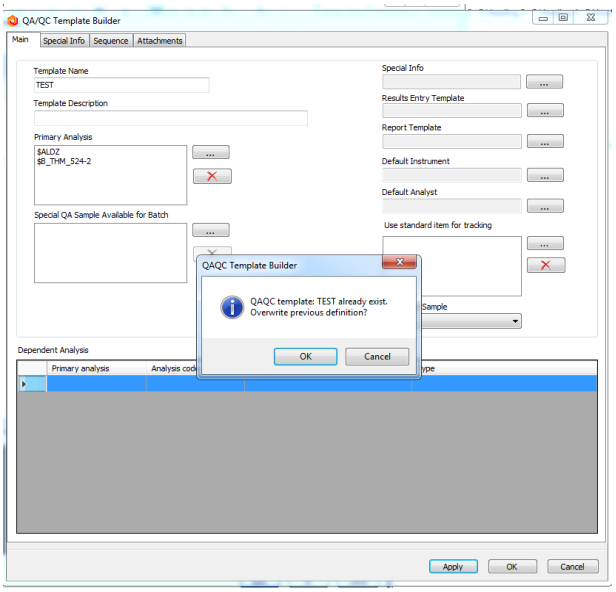
QA/QC Template Builder Main Screen – after new template entry

Remember this template is still in DRAFT status as shown above in the tree. If you have state management turned on in your system, you must have a supervisor approve this template before use. All state managed objects in LABWORKS must be in an *approved* ACTIVE state before they can be used.

Modifying Existing QA/QC Batching Templates

To modify a template from the main QA/QC Template listing, simply double-click on it.

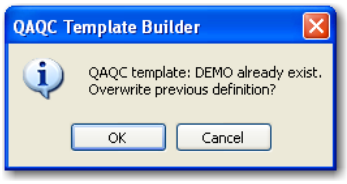
The same QA/QC Template screen should open showing all the details on the template record allowing you to change any field, add additional analyses, SI fields, etc.



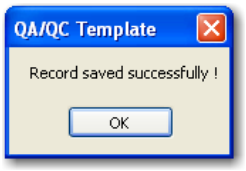
QA/QC Template Screen – after modification

For a list of screen fields and how they are used, reference the previous section for entering new templates.

After making all your necessary template adjustments, click the **Apply** button to save it. There will be a verification box. Click **OK** to save or **Cancel** to not save your changes.



Clicking **OK** will confirm the save was successful (or not).



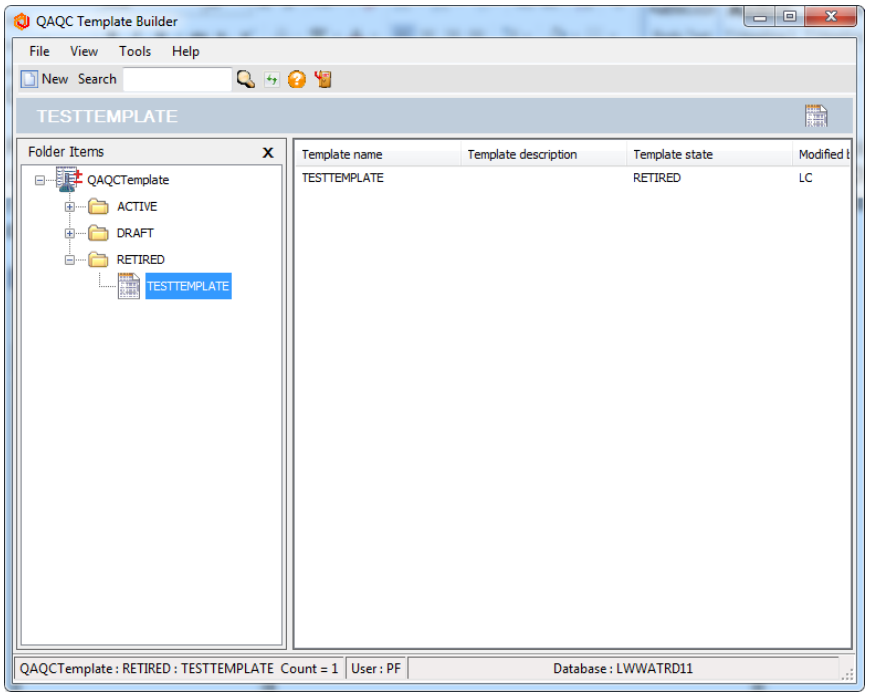
Click OK again to return back to the main screen.

Note that once again your template has been moved to a **DRAFT** status. If there are LABWORKS users that rely on this specific batching template, remember you will still need to go *approve* this template into an active status before it can be used.

Retiring / Removing Existing QA/QC Batching Templates

To retire (remove) a template from the LABWORKS QA/QC sample batching process, we must select to retire it in the LABWORKS **Object Publisher**. You must have the proper privileges to retire a template. Once it is retired, it is no longer selectable for batching.

Retired batching templates show up under the RETIRED tree in the QA/QC Template Builder. This is the *only* place they can be viewed again once retired from the system outside of the **Object Publisher** (where they can be moved back to active if needed). This gives the user the capabilities of moving an old template back into circulation if needed.

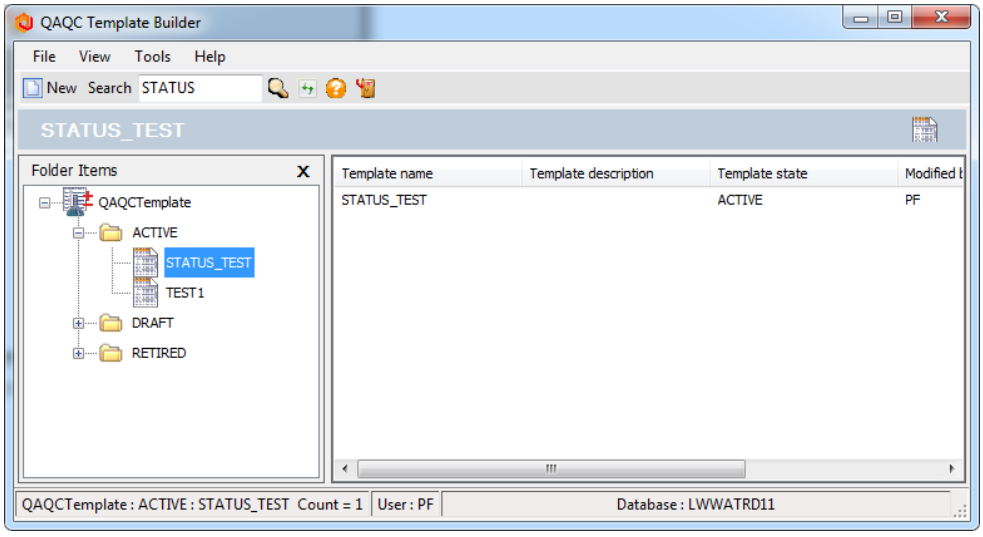


QA/QC Template Builder Main Screen – active and retired templates shown

Searching for QA/QC Batching Templates

Use the Search box at the top of the main QA/QC Template Builder screen within the toolbar to find specific templates to modify.

Simply click in the search box, enter the template name value (all or part), then hit the [Enter] key or click the magnifying glass to the right of the box. The system will search for that template and highlight it on the left within the tree. You can then double-click to open the template record for modification.



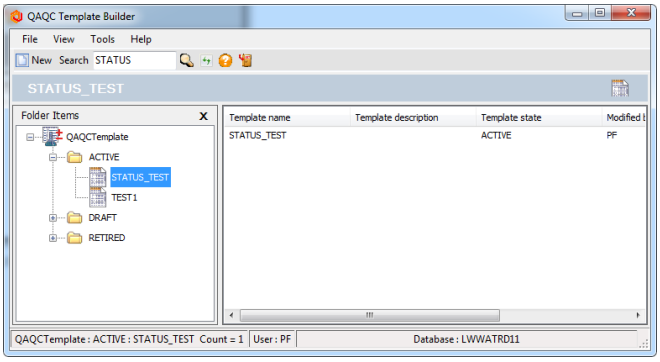
QA/QC Template Builder Main Screen – searching for templates

Approving QA/QC Batching Templates for use

The very last step before you use a QA/QC Batching Template in production is to have someone with an appropriate security privilege *approve* it from a **DRAFT** to **ACTIVE** state.

We simple need to right-click on our new **DRAFT** template and select the new state. This will allow the transition into the next state. If it is an automatic state transition, it will move into the state wi thout providing the choice. Manual state transitions will require you to select them first from the right-click menu.

Once our new template is approved for use, it is ready for sample batching in LABWORKS. The next time we open the QA/QC Template Builder, we can see our new template has made the transition into the approved production state of **ACTIVE**.



QA/QC Template Builder after Template Approval to ACTIVE

Note that it only shows the state because we selected it to be displayed under **Tools-Configure Tree View**.

QA/QC Batch Builder

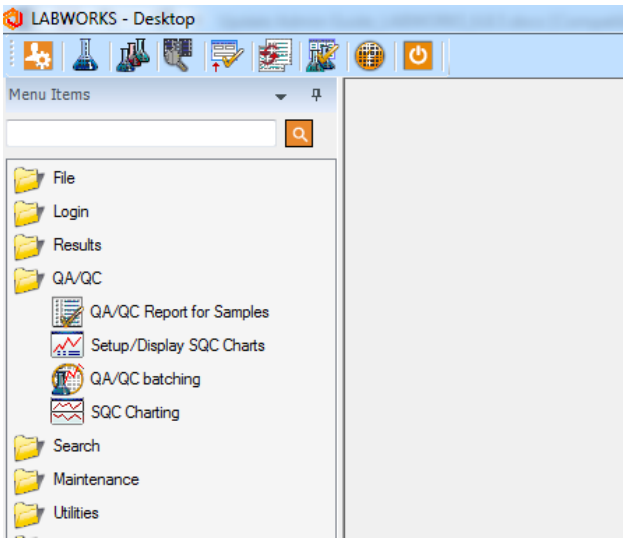
Introduction

The purpose of the Batch Builder application is to run a QA/QC template to produce a QA/QC “batch” of samples. It is the tool that does the batch execution defined from template creation in the previous chapter. In the previous legacy application, certain batch configuration detail was kept in the LABWORKS.INI file. The INI file is no longer used for QA/QC program configuration. The configuration of a QA/QC batch is exclusively done within the creation of a *batch template*. Reference the previous chapter for creating new QA/QC templates or modifying existing ones.

The program filename is **LWQAQCBatchBuilder.exe** and is located in the C:\LABWORKS\Client\ folder.

Note that all QA/QC batching templates created are *objects* in LABWORKS and can be state managed. Initially all templates are in DRAFT mode. If state management is turned on in LABWORKS, you must *approve* a batching template before it can be used. Only approved active batches show up in the QA/QC Batch Builder tool.

As mentioned in the first section of this chapter, the two programs that make up the new QA/QC module must be added as links on the desktop if upgrading from a previous version of LABWORKS. Otherwise, the old legacy application will launch instead.

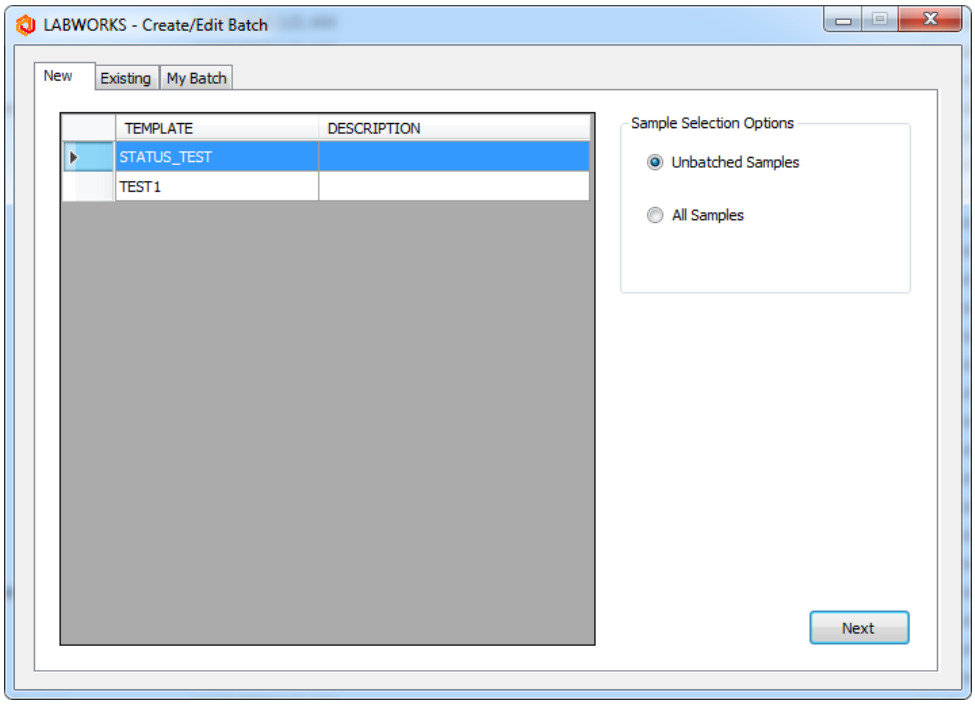


LABWORKS Desktop with Quality Assurance folder example

In the example screenshot above, both programs were added under a new *Quality Assurance* folder along with some of the applications that integrate with batch templates (Chemical Inventory, Personnel Training, Instrument Maintenance and Calibration, etc.).

Batch Builder UI

The program's user interface (UI) is much like a wizard. Initially the active batch templates are displayed for selection.



QA/QC Batch Builder Initial Screen

There are two lists displayed in two separate tabs for batch template selection. The **New** tab shows a list of *active* templates for new batch creation. You are able to select from **unbatched samples** (samples that haven't been batched previously) or **all samples**. The **Existing** tab shows a listing of batches already created from a designated date value. This tab allows you to select existing batches for modification.

Clicking the **Next** button will move to the next step in the batching process, assigning sample numbers.

The following sample assignment screen is a listing of all available samples on the left with a selection of samples for the batch on the right. The defaults entered for the selected template are then applied to the following screen. This next screen is where batch creation occurs.

The following section will detail the steps in the batch creation process.

Creating a New QA/QC Batch

The following are the steps in creating new QA/QC batches from a selected template.

1. From the QA/QC Batch Builder initial screen, select the desired QA/QC template from the list under the **New** tab. Click to select it.
2. On the right side of the tab, select one of the following **Sample Selection** options
   a. **Unbatched Samples** – for samples not previously batched
   b. All Samples
3. Click **Next** to continue. The following screen is where the assignment of samples occurs.

QA/QC Batch Builder – Assign Samples screen

The left-side listing shows an available sample listing for either unbatched or all samples (as chosen from the previous screen) while the right-side is a listing of the samples in your batch ***sequence***. If you selected a template of 10 samples in your sequence, then 10 available sample "slots" will be open on the right. You can then scroll down the left side, find samples, and then use the arrow keys to assign them on the right.

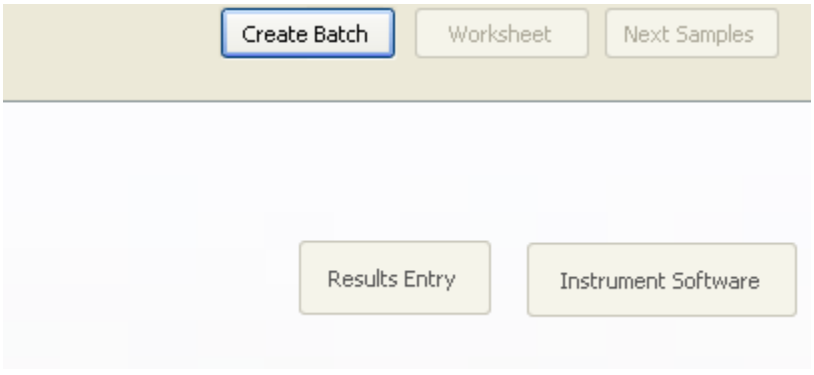
Create Batch button – this will create the batch once we are ready.

Worksheet button – currently grayed out but will activate once the batch is created at a later step. The worksheet is our Crystal report template assigned to this selected QA/QC template.

Next Samples button – currently grayed out but will activate once the batch is created. If there are more samples in the backlog than our batch size, we can batch those by choosing **Next Samples**.

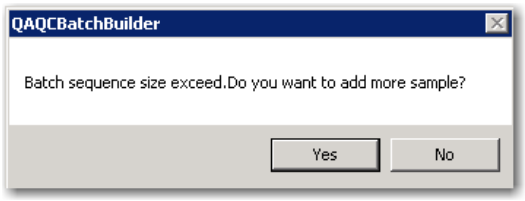
Results Entry button – currently grayed out but will activate once the batch is created. This will take us directly into the Results Entry module (using our defined template) for this batch for immediate entering of results.

Instrument Software button – currently grayed out but will activate once the batch is created. This will open the link to the program we specified in the QA/QC Template.

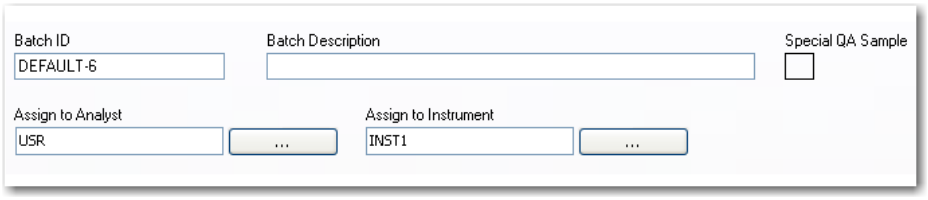


4. Proceed to use the arrow keys in the center of the lists to assign samples to your sequence on the right.

NOTE: If you need more samples than assigned to the batch, simply select the last row and click the right arrow. There will be a prompt to add more samples to the sequence. Click **Yes** to add another row.



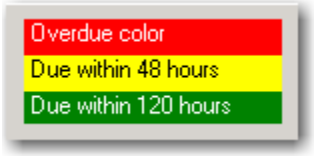
5. Enter the **Batch ID** – the template name (by default) followed by the batch number.
6. Enter the **Batch Description** – the template description (by default) of this batch but it can be modified or added to (up to 60characters).
7. Enter the **Special QA Sample** – if a **QA**_ location code was assigned to the QA/QC template, it will show here.
8. Enter the **Assign to Analyst** – the analyst (from Personnel Training) assigned to the template is shown here by default, but we can choose someone else.
9. Enter the **Assign to Instrument** – the Instrument (from Instrument Maintenance and Calibration) is shown here by default, but we can pick a new instrument.



10. (Optional) enter any special information for this batch under the **Special Info** tab. The default values enter into the QA/QC template should be displayed allowing you to modify them to enter more information.
11. When finished entering special information, click back to the **Samples** tab.

Note: You can use System Manager (*SysMgr*) keys to filter the lists of analysts and instruments. Consult the System Administrator Reference or QA/QC Module Technical Reference chapters for more detail if list filtering is needed.

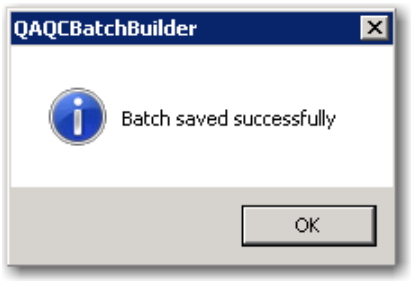
Note: The color scheme in the left-side list that shows overdue samples can be modified as well using System Manager keys.



Once again, consult your System Administrator Reference or QA/QC Module Technical Reference chapters for more detail if you need to change this record list color scheme.

The large blue arrow in the center is the **Auto Fill** button. This will take all of the available samples and move them to the right (This is limited to our batch size). It will also assign the QA samples to the first sample number it moves across. Once again, as in step 4 above, if you need to expand the sequence list on the right, it will prompt you to do so.

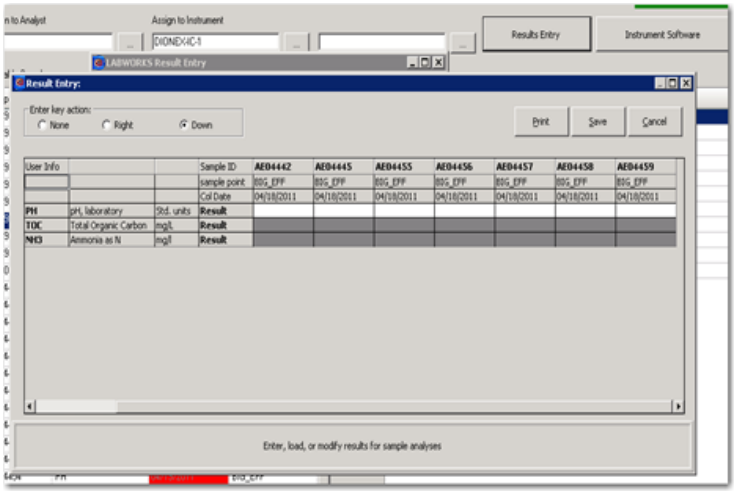
12. If everything looks correct, click the **Create Batch** button to proceed. The batch is then created and everything saved. A message will display when the batch has completed the creation process was successful (or not).



Note: If you did not have a Crystal report assigned to the QA/QC template, the **Worksheet** button will remain grayed out.

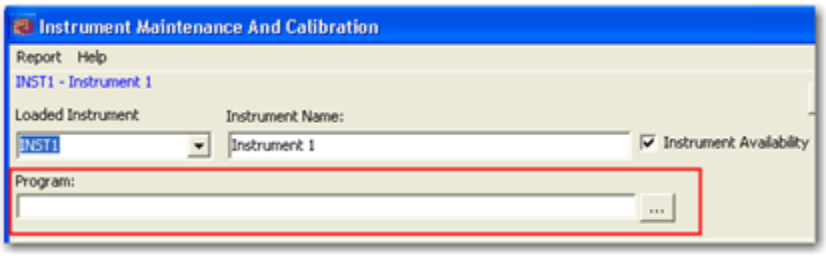
Note: The Result entry and Instrument Software buttons are now active.

(Optional) click the **Results Entry** button to enter results now.



Results Entry directly from QA/QC Batch Builder

13. Click **Save** to store the results and close Results Entry.
14. Click Instrument Software to launch the program associated in Instrument Maintenance and Calibration (if assigned).

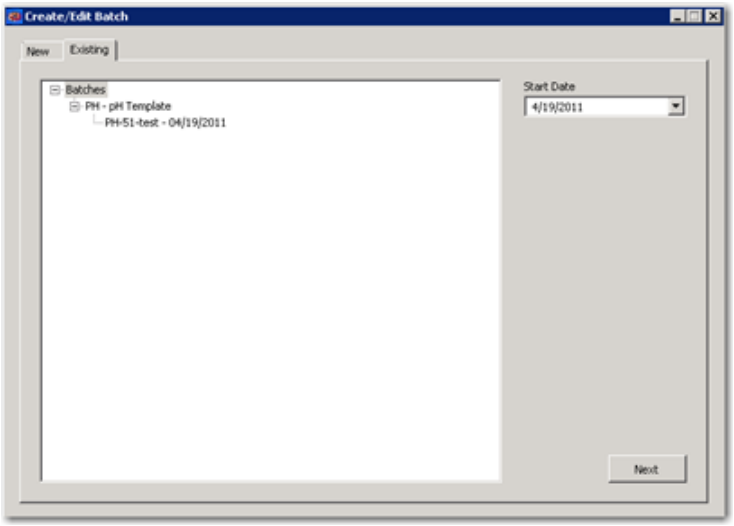


15. At this point, the batch is ready for reporting and validation. Click **Next Samples** to display additional samples from the backlog that are available for batching if larger than our batch size.
16. When finished batching any additional samples, we can close this screen to return to the main batching screen. This concludes the creation of **new batches**.

Modify Existing QA/QC Batches

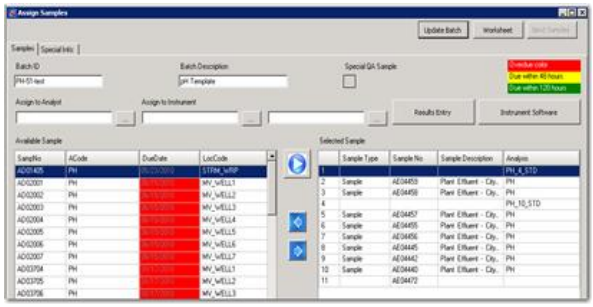
When opening the QA/QC Batch Builder main screen, you can select to create new batches (previous section) or modify existing batches. To modify existing batches, click the **Existing** tab.

The QA/QC batch we just created (in the previous section) will be displayed in the list of existing batches.



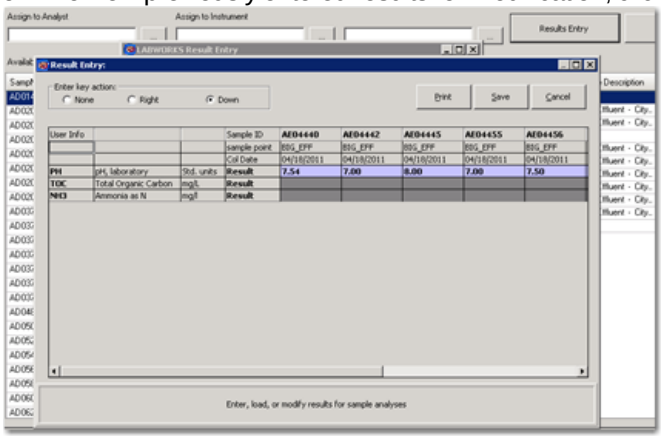
QA/QC Batch Builder – Existing Batches

1. Click to select the batch in the list to modify.
2. Click **Next** to open the batch. The Assign Samples screen opens.



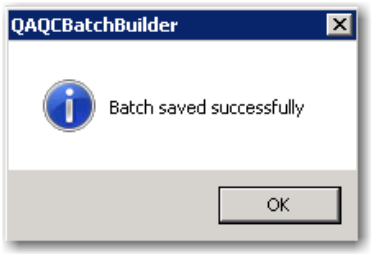
QA/QC Assign Samples screen – Selected Batch Modification

3. Modify any fields necessary or add additional samples to the batch as needed.
4. Click the Special Info tab to modify any special information fields as needed.
5. To view previously entered results for modification, click **Results Entry**.



Results Entry editing directly from QA/QC Batch Builder

6. When finished making modifications click the **Update Batch** button.



7. You can now exit the screen to return to the main QA/QC Batch Builder screen. You batch has been successfully modified.

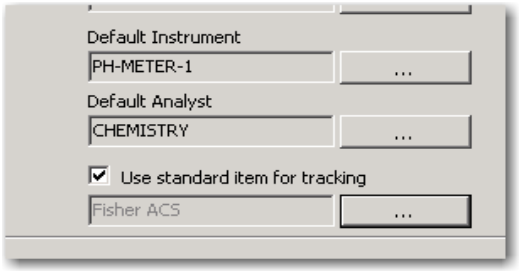
QA/QC Batching and Other LIMS Modules

Introduction

QA/QC Batching is now fully integrated with other optional modules such as Chemical Inventory, Instrument Maintenance and Calibration, and Personnel Training. This chapter will provide useful tips on using them together.

It is a common practice to apply periodic QA/QC procedures to chemicals stored in a laboratory's inventory (to ensure quality and validity) and instrumentation used in the lab for sampling (to ensure proper calibration). It is also common for the actual Chemists to complete the proper training prior to doing quality assurance testing (QA/QC) procedures. Since other modules within LABWORKS track this data, it was important that the new QA/QC module have the capability of pulling this data for tracking within a given QA/QC batching template.

On the far right-side of a QA/QC template is where the links for these optional modules are placed. An ellipse button to the right of each item will select these records from the optional modules.

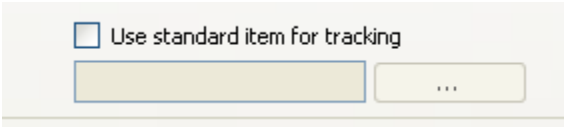


Optional Module Section in a QA/QC Template

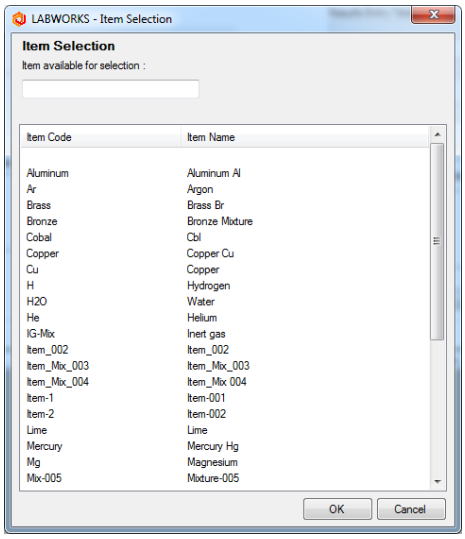
The following sections will cover the basic usage of these other optional modules specifically as it relates to QA/QC. Consult the individual module chapters for more detail for usage of each optional application. Consult your System Administrator if there is a module that is not operating or setup properly within your LIMS.

Chemical Inventory

When creating a QA Batch Template, we have the option to '**Use standard item for tracking**'.



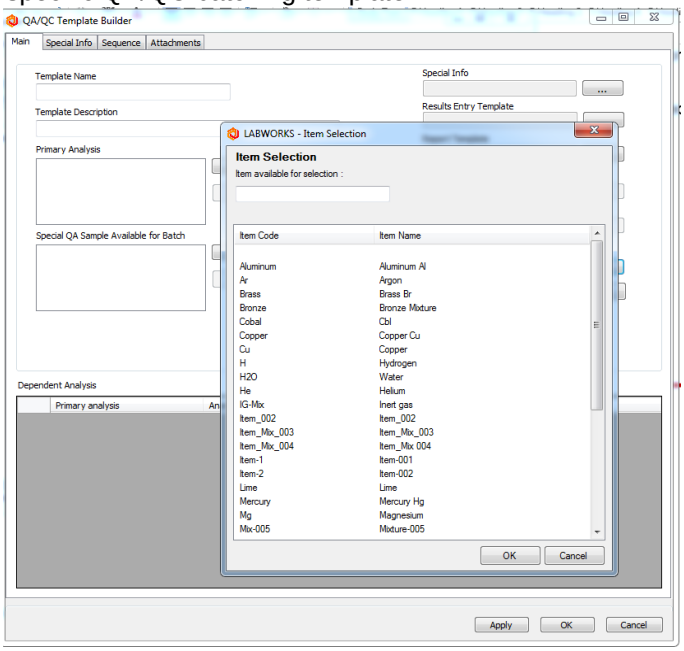
After we've selected to use a standard item for tracking, we can then pick from available items in Chemical Inventory (if any).



Item Selection from Chemical Inventory within a QA/QC Template

If nothing is showing within the list initially, you can open Chemical Inventory and define items within that module. Chemical Inventory is not a core LABWORKS application and will require a new control file to unlock the module along with some initial setup time for data entry and configuration.

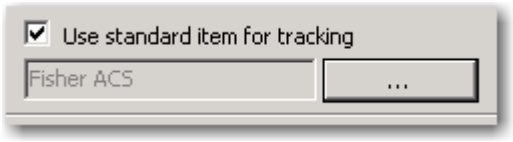
Once you have data in Chemical Inventory, any inventory item can be selected and assigned to a specific QA/QC batching template.



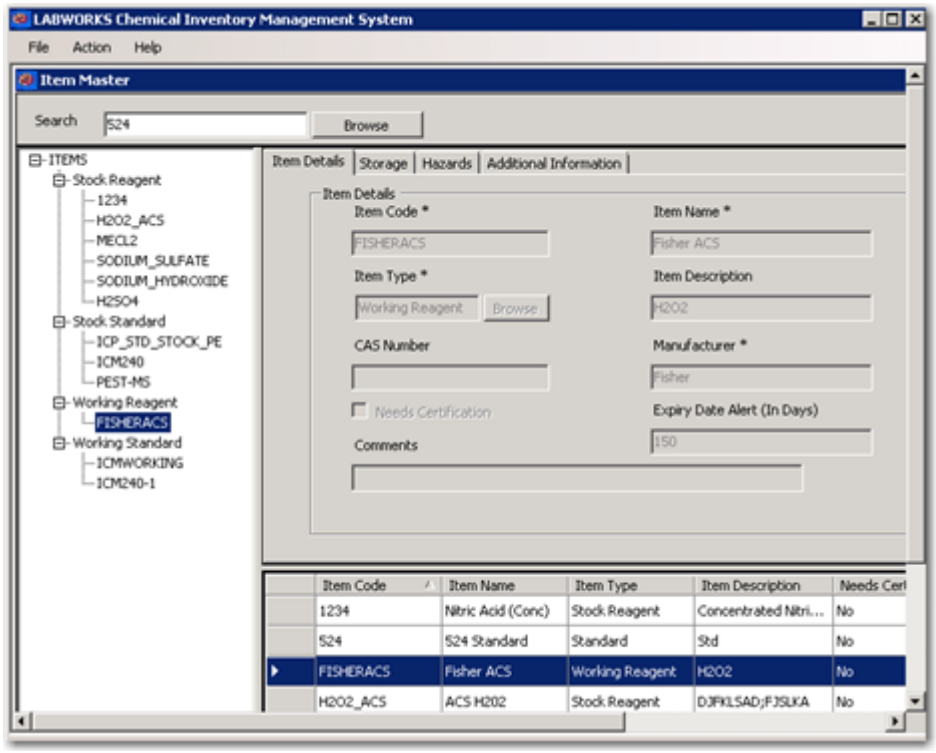
Item Selection from Chemical Inventory within a QA/QC Template

Once a chemical inventory item is assigned and saved with the QA/QC template, all QA/QC batches created from that template will be specific to that inventory item. This is a great method of tracking sampling done over time for inventory items. This data will then be available for reports showing the sampling performed on specific inventory items over a given date range, for example.

Once the inventory item is selected within the template and saved, it will be displayed on the right-side of the screen.



Within the Chemical Inventory module all item-specific detail is stored. Reporting on inventory can be done from within this module directly. However, sample results reporting on QA/QC specifics for an item can be accomplished from the LABWORKS LIMS Desktop customized to the needs of the laboratory.



Same QA/QC Item Selected within Chemical Inventory

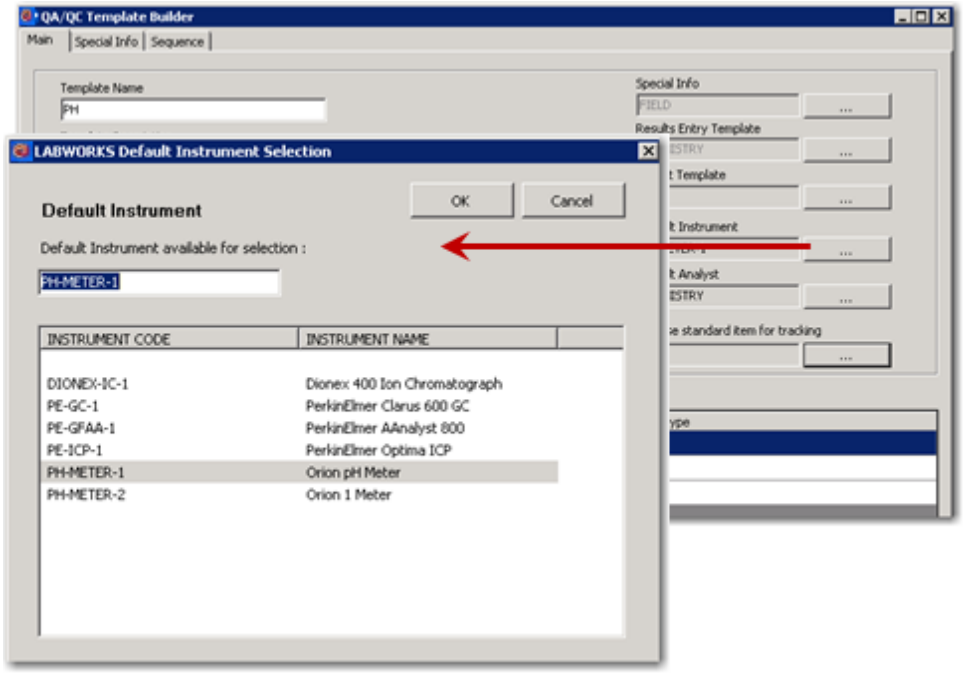
The integration of these two modules can merge the data from inventory item detail to QA/QC sample results reporting into custom Desktop reports.

Consult your System Administrator for custom reporting needs. The LABWORKS Report Designer module can be used to build these specific reports.

Instrument Maintenance and Calibration

When creating a QA Batch Template, we have the option to select a '**Default Instrument**'. This will open the list from the Instrument Maintenance and Calibration (IM&C) module.

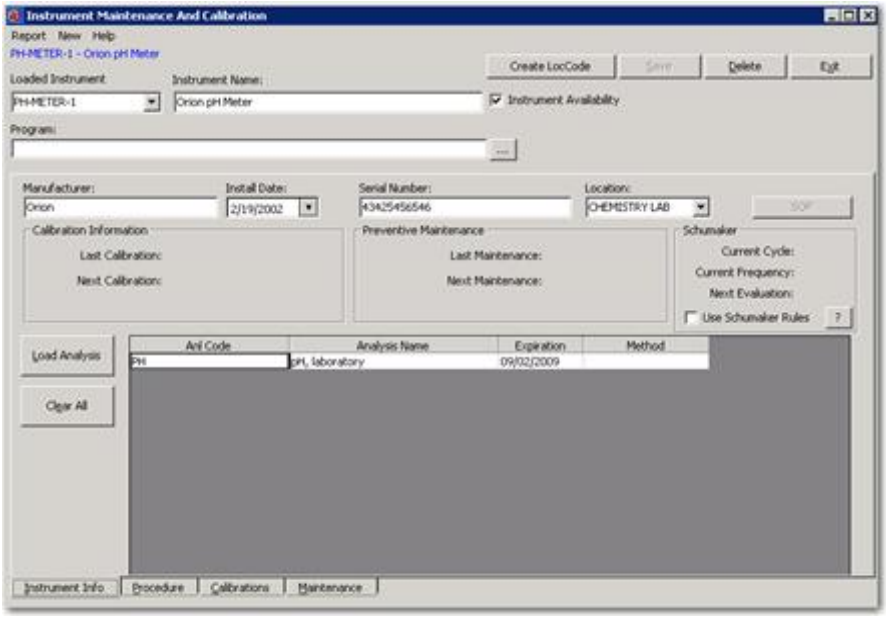
Once you have data in the IM&C module, any instrument can be selected and assigned to a specific QA/QC batching template.



Item Selection from IM&C within a QA/QC Template

Once an instrument is assigned and saved with the QA/QC template, all QA/QC batches created from that template will be specific to instrument. This is a great method of tracking calibration checking as proof of sampling accuracy or for laboratory certification requirements. This data will then be available for reports showing the sampling performed on specific instruments over a given date range, for example.

The instrument record itself within the IM&C module will have all the necessary calibration procedures and analyses assigned while the QA/QC template is basically the blueprint for assuring the calibration testing is done exactly the same each time. By the use of a QA/QC batching template for a given instrument, we know the QA/QC sample batching is done correctly each time calibration testing is required.



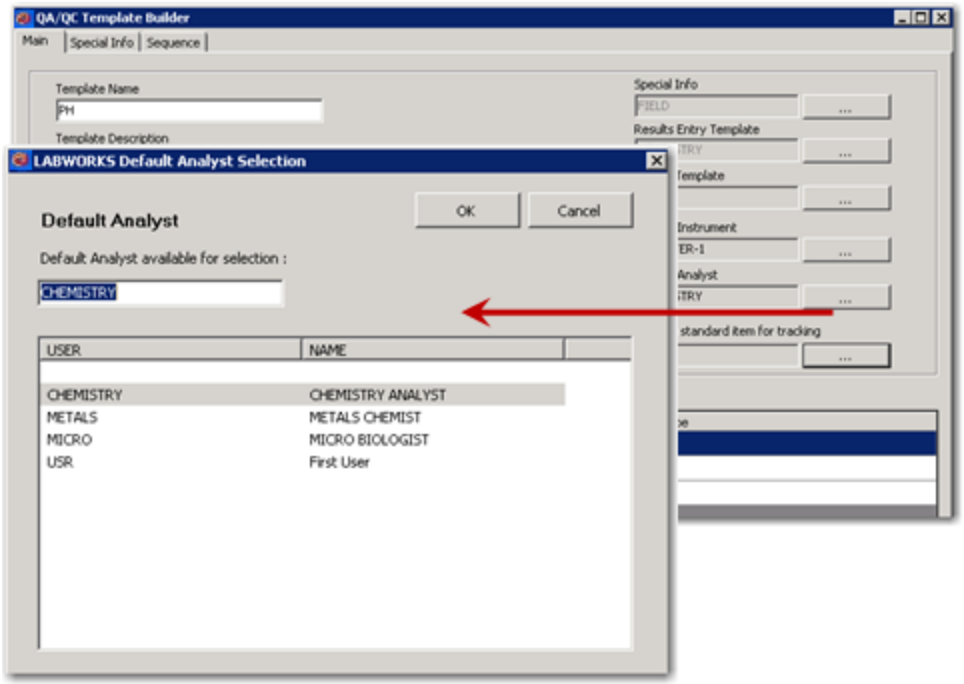
Same Instrument Selected from within the IM&C module

Consult your System Administrator for custom reporting needs. The LABWORKS Report Designer module can be used to build specific instrument calibration sample results reports.

Personnel Training

When creating a QA Batch Template, we have the option to select a '**Default Analyst**'. This will open the list from the Personnel Training module.

Once you have data in the Personnel Training module, any user can be selected and assigned to a specific QA/QC batching template.

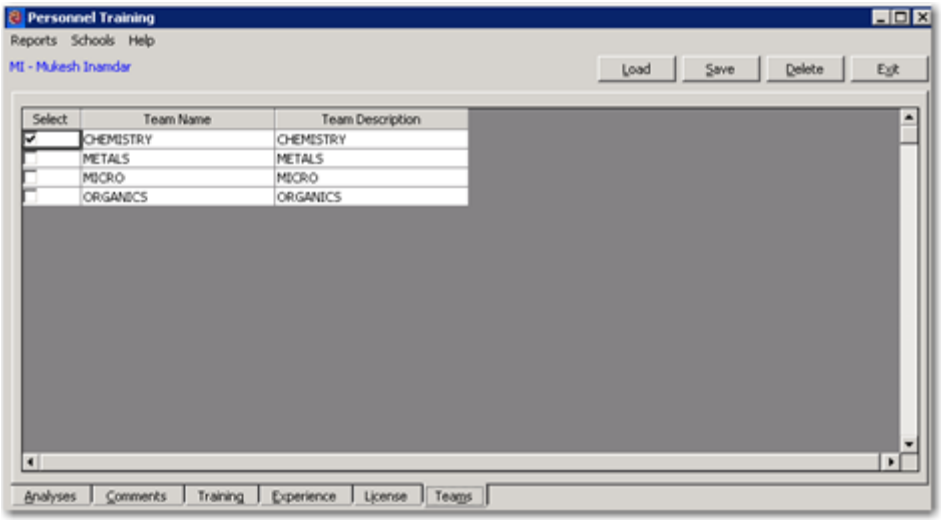


Item Selection from Personnel Training within a QA/QC Template

Once a user from Personnel Training is assigned and saved with the QA/QC template, all QA/QC batches performed by that user are tracked. If there are any problems in how testing is done or how an instrument was calibrated, for example, a Lab Manager can easily observe who performed the work.

Integration of Personnel Training can be used for the assignment of chemists to do QA/QC sampling that are ***qualified*** only. In other words, specific chemists may have gone through required calibration training while others have not. You may want to ensure that only those specifically qualified individuals are performing the quality assurance sampling. It may also be a requirement for laboratory certification as well that only those qualified individuals perform these tasks.

Before a user will be selectable within a QA/QC template, they must be assigned to a specific team. Teams are important for tracking purposes. From within Personnel Training, click **Load** to open a user. Once the user is selected, you can click the **Teams** tab to assign a team to that user. Then click **Save** to store it. Once the team assignment is completed, that user should show up within the QA/QC module.



User Selected from Personnel Training – Teams Tab

Consult your System Administrator to assign new, modify, or remove teams from the Personnel Training module.

Consult your System Administrator for personnel custom reporting needs. The LABWORKS Report Designer module can be used to build specific sample results reports specific to individual users within the Personnel Training module.

Additional QA/QC Common Terminology

Acceptance Criteria – specified limits placed on characteristics of an item, process, or service defined in requirements documents.

Accuracy – a measure of the closeness of an individual measurement or the average of a number of measurements to the true value. Accuracy includes a combination of random error (precision) and systematic error (bias) components that are due to sampling and analytical operations; the EPA recommends using the terms "precision" and "bias", rather than "accuracy," to convey the information usually associated with accuracy.

Activity – an all-inclusive term describing a specific set of operations of related tasks to be performed, either serially or in parallel (e.g., research and development, field sampling, analytical operations, equipment fabrication), that, in total, result in a product or service.

Analyte – a specific chemical that can be detected by a given analytical method.

Assessment – the evaluation process used to measure the performance or effectiveness of a system and its elements. As used here, assessment is an all-inclusive term used to denote any of the following: audit, performance evaluation (PE), management systems review (MSR), peer review, inspection, or surveillance.

Audit (quality) – a systematic and independent examination to determine whether quality activities and related results comply with planned arrangements and whether these arrangements are implemented effectively and are suitable to achieve objectives.

Audit of Data Quality (ADQ) – a qualitative and quantitative evaluation of the documentation and procedures associated with environmental measurements to verify that the resulting data are of acceptable quality.

Authenticate – the act of establishing an item as genuine, valid, or authoritative.

Background Sample – a sample taken from an area close to the site of investigation where chemicals present in the ambient media are assumed to be present at concentrations not associated with the investigation site.

Bias – the systematic or persistent distortion of a measurement process, which causes errors in one direction (i.e., the expected sample measurement is different from the sample's true value).

Blank – a sample subjected to the usual analytical or measurement process to establish a zero baseline or background value. Sometimes used to adjust or correct routine analytical results. A sample that is intended to contain none of the analytes of interest. A blank is used to detect contamination duri ng sample handling preparation and/or analysis. There are many types of blanks, each with a specific purpose including:

Equipment Blanks – monitor for potential contamination from decontamination procedures of field gear or from other sources of equipment contamination like oil or other lubricants.

Field Blank – a blank used to provide information about contaminants that may be introduced during sample collection, storage, and transport. A clean sample, carried to the sampling site, exposed to sampling conditions, returned to the laboratory, and treated as an environmental sample.

Laboratory Blanks – samples that are used to identify potential sources of contamination that are generated during the processing and analysis of samples in the laboratory.

Method Blank – a blank prepared to represent the sample matrix as closely as possible and analyzed exactly like the calibration standards, samples, and quality control (QC) samples. Results of method blanks provide an estimate of the within-batch variability of the blank response and an indication of bias introduced by the analytical procedure.

Trip Blank – a clean sample of a matrix that is taken to the sampling site and transported to the laboratory for analysis without having been exposed to sampling procedures.

Calibration – a comparison of a measurement standard, instrument, or item with a standard or instrument of higher accuracy to detect and quantify inaccuracies and to report or eliminate those inaccuracies by adjustments.

Calibration Curves – curves that are generated by regression of the response of a detector against the known concentrations of analytes of interest. There are several types of calibration curves including:

Continuing Calibration – a mid-range concentration standard that is analyzed on a daily basis and is compared with the continuing calibration of fidelity to the regression relationship established using the continuing calibration. The continuing calibration is often referred to as the Daily Calibration.

Initial Calibration – a multi-concentration, generally multi-analyte series of known concentrations analyzed periodically to establish a regression relationship that established the lower and upper concentration limits.

Calibration Drift – the deviation in instrument response from a reference value over a period of time before recalibration.

Certification – the process of testing and evaluation against specifications designed to document, verify, and recognize the competence of a person, organization, or other entity to perform a function or service, usually for a specified time.

Collocated Samples – two or more portions collected at the same point in time and space so as to be considered identical. These samples are also known as field replicates and should be identified as such.

Comparability – a measure of the confidence with which one data set or method can be compared to another.

Completeness – a measure of the amount of valid data obtained from a measurement system compared to the amount that was expected to be obtained under correct, normal conditions.

Confidence Interval – the numerical interval constructed around a point estimate of a population parameter, combined with a probability statement (the confidence coefficient) linking it to the population's true parameter value. If the same confidence interval construction technique and assumptions are used to calculate future intervals, they will include the unknown population parameter with the same specified probability.

Confidentiality Procedure – a procedure used to protect confidential business information (including proprietary data and personnel records) from unauthorized access.

Data Quality Assessment (DQA) – the scientific and statistical evaluation of data to determine if data obtained from environmental operations are of the right type, quality, and quantity to support their intended use. The five steps of the DQA Process include: 1) reviewing the DQOs and sampling design, 2) conducting a preliminary data review, 3) selecting the statistical test, 4) verifying the assumptions of the statistical test, and 5) drawing conclusions from the data.

Data Quality Indicators (DQIs) – the quantitative statistics and qualitative descriptors that are used to interpret the degree of acceptability or utility of data to the user. The principal data quality indicators are bias, precision, accuracy (bias is preferred), comparability, completeness, representativeness.

Data Quality Objectives (DQOs) – the qualitative and quantitative statements derived from the DQO Process that clarify study's technical and quality objectives, define the appropriate type of data, and specify tolerable levels of potential decision errors that will be used as the basis for establishing the quality and quantity of data needed to support decisions.

Data Quality Objectives (DQO) Process – a systematic strategic planning tool based on the scientific method that identifies and defines the type, quality, and quantity of data needed to satisfy a specified use. DQOs are the qualitative and quantitative outputs from the DQO Process.

Data Validation – a well-defined procedure for assessing whether or not a set of environmental data have met acceptability criteria defined in the data quality objective process. This procedure may follow existing guidelines, for example those define by EPA (Guidelines for data validation; EPA New England data validation guidelines) or they can be defined by the user as part of the DQO process. These user - defined criteria may be either more or less strict than the EPA guidelines.

Deficiency – an unauthorized deviation from acceptable procedures or practices, or a defect in an item.

Demonstrated Capability – the capability to meet a procurement's technical and quality specifications through evidence presented by the supplier to substantiate its claims and in a manner defined by the customer.

Design – the specifications, drawings, design criteria, and performance requirements. Also, the result of deliberate planning, analysis, mathematical manipulations, and design processes.

Design Change – any revision or alteration of the technical requirements defined by approved and issued design output documents and approved and issued changes thereto.

Design Review – a documented evaluation by a team, including personnel such as the responsible designers, the client for whom the work or product is being designed, and a quality assurance (QA) representative but excluding the original designers, to determine if a proposed design will meet the established design criteria and perform as expected when implemented.

Detection Limit (DL) – a measure of the capability of an analytical method to distinguish samples that do not contain a specific analyte from samples that contain low concentrations of the analyte; the lowest concentration or amount of the target analyte that can be determined to be different from zero by a single measurement at a stated level of probability. DLs are analyte- and matrix-specific and may be laboratory-dependent. Some of the more commonly used definitions are described below.

Instrument Detection Limit (IDL) – the lowest concentration or mass an instrument can detect above background instrument noise under ideal conditions. Sample preparation is not considered in the determination of an IDL.

Method Detection Limit – a statistically derived estimate of the lowest concentration or mass detectable under method conditions at the concentration evaluated. A series of standards at an estimated limit of detection is analyzed multiple times (usually 7), a standard deviation of these seven replicate analyses is determined and the standard deviation is multiplied by the Student's t-distribution statistic at 6 degrees of freedom (3.14). See Reporting Limit.

Practical Quantitation Limit (PQL) – a measure of the lowest limit of detection under the conditions of a particular method. The PQL is often determined by multiplying the MDL by a factor of between 3 and 10.

Reporting Limit (RL) – for a target analyte, the reporting limit is instrument dependent and based on the lowest point on the calibration curve.

Distribution – 1) the appointment of an environmental contaminant at a point over time, over an area, or within a volume; 2) a probability function (density function, mass function, or distribution function) used to describe a set of observations (statistical sample) or a population from which the observations are generated.

Document Control – the policies and procedures used by an organization to ensure that its documents and their revisions are proposed, reviewed, approved for release, inventoried, distributed, arch ived, stored, and retrieved in accordance with the organization's requirements.

Duplicate Analysis – a measure of precision determined by analyzing samples twice or by analyzing a second sample taken from the same source at the same time and analyzed under identical conditions. There are several different types of duplicate samples that provide information on the precision of specific types of environmental data operations

Field Duplicates – independent samples that are collected as close as possible to the same point in time and space. They are two separate samples taken from the same source, stored in separate containers and analyzed independently. These types of duplicates are useful in characterizing the precision of the sampling process.

Matrix Duplicates – an intralaboratory split sample that is used to document the precision of a method in a given sample matrix.

Split Samples – two or more representative portions taken from one sample in the field or in the laboratory and analyzed by different analysts or laboratories. Split samples are quality control (QC) samples that are used to assess analytical variability and comparability.

Duplicate Samples – two samples taken from and representative of the same population and carried through all steps of the sampling and analytical procedures in an identical manner. Duplicate samples are used to assess variance of the total method, including sampling and analysis. See also collocated sample.

Environmental Conditions – the description of a physical medium (e.g., air, water, soil, sediment) or a biological system expressed in terms of its physical, chemical, radiological, or biological characteristics.

Environmental Data – any parameters or pieces of information collected or produced from measurements, analyses, or models of environmental processes, conditions, and effects of pollutants on human health and the ecology, including results from laboratory analyses or from experimental systems representing such processes and conditions.

Environmental Data Operations – any work performed to obtain, use, or report information pertaining to environmental processes and conditions.

Environmental Monitoring – the process of measuring or collecting environmental data.

Environmental Processes – any manufactured or natural processes that produce discharges to, or that impact, the ambient environment.

Environmental Programs – an all-inclusive term pertaining to any work or activities involving the environment, including but not limited to: characterization of environmental processes and conditions; environmental monitoring; environmental research and development; the design, construction, and operation of environmental technologies; and laboratory operations on environmental samples.

Environmental Technology – an all-inclusive term used to describe pollution control devices and systems, waste treatment processes and storage facilities, and site remediation technologies and their components that may be utilized to remove pollutants or contaminants from, or to prevent them from entering, the environment. Examples include wet scrubbers (air), soil washing (soil), granulated activated carbon unit (water), and filtration (air, water). Usually, this term applies to hardware-based systems; however, it can also apply to methods or techniques used for pollution prevention, pollutant reduction, or containment of contamination to prevent further movement of the contaminants, such as capping, solidification or vitrification, and biological treatment.

Equipment Blanks – monitor for potential contamination from decontamination procedures of field gear or from other sources of equipment contamination like oil or other lubricants.

Estimate – a characteristic from the sample from which inferences on parameters can be made.

Evidentiary Records – any records identified as part of litigation and subject to restricted access, custody, use, and disposal.

Expedited Change – an abbreviated method of revising a document at the work location where the document is used when the normal change process would cause unnecessary or intolerable delay in the work.

Environmental Technology – an all-inclusive term used to describe pollution control devices and systems, waste treatment processes and storage facilities, and site remediation technologies and their components that may be utilized to remove pollutants or contaminants from or prevent them from entering the environment. Examples include wet scrubbers (air), soil washing (soil), granulated activated carbon unit (water), and filtration (air, water). Usually, this term will apply to hardware-based systems; however, it will also apply to methods or techniques used for pollution prevention, pollution reduction, or containment of contamination to prevent further movement of the contaminants, such as capping, solidification or vitrification, and biological treatment.

Extraction – the process of releasing compounds for a sample matrix prior to analysis.

Field Analysis – analyses that are performed external to a fixed laboratory. Field analysis can be qualitative or quantitative depending on the instruments used to perform the analyses and the level of instrument calibration.

Field Blank – a blank used to provide information about contaminants that may be introduced during sample collection, storage, and transport. A clean sample, carried to the sampling site, exposed to sampling conditions, returned to the laboratory, and treated as an environmental sample.

Field Duplicates – independent samples that are collected as close as possible to the same point in time and space. They are two separate samples taken from the same source, stored in separate containers and analyzed independently. These types of duplicates are useful in characterizing the precision of the sampling process.

Field Screening – analysis in the field using portable instrumentation that may include hand-held devices with general detectors (e.g., photoionization detectors (PID) or portable gas chromatographs equipped with an appropriate detector (flame ionization detector, mass spectrometer, etc.).

Field Blank – a blank used to provide information about contaminants that may be introduced during sample collection, storage, and transport. A clean sample, carried to the sampling site, exposed to sampling conditions, returned to the laboratory, and treated as an environmental sample.

Field (Matrix) Spike – a sample prepared at the sampling point (i.e., in the field) by adding a known mass of the target analyte to a specified amount of the sample. Field matrix spikes are used, for example, to determine the effect of the sample preservation, shipment, storage, and preparation on analyte recovery efficiency (the analytical bias).

Field Split Samples – two or more representative portions taken from the same sample and submitted for analysis to different laboratories to estimate interlaboratory precision.

Financial Assistance – the process by which funds are provided by one organization (usually governmental) to another organization for the purpose of performing work or furnishing services or items. Financial assistance mechanisms include grants, cooperative agreements, and governmental interagency agreements.

Finding – an assessment conclusion that identifies a condition having a significant effect on an item or activity. An assessment finding may be positive or negative, and is normally accompanied by specific examples of the observed condition.

Flame Ionization Detector (FID) – a universal detector that uses a flame to burn a gas stream usually from a chromatographic column.

Goodness-of-Fit Test – the application of the chi square distribution in comparing the frequency distribution of a statistic observed in a sample with the expected frequency distribution based on some theoretical model.

Grade – the category or rank given to entities having the same functional use but different requirements for quality.

Graded Approach – the process of basing the level of application of managerial controls applied to an item or work according to the intended use of the results and the degree of confidence needed in the quality of the results. See also Data Quality Objectives (DQO) Process.

Guidance – a suggested practice that is not mandatory, intended as an aid or example in complying with a standard or requirement.

Guideline – a suggested practice that is not mandatory in programs intended to comply with a standard.

Hazardous Waste – any waste material that satisfies the definition of hazardous waste given in 40 CFR 261, "Identification and Listing of Hazardous Waste."

Holding Time – the period of time a sample may be stored prior to its required analysis. While exceeding the holding time does not necessarily negate the veracity of analytical results, it causes the qualifying or "flagging" of any data not meeting all of the specified acceptance criteria.

Identification Error – the misidentification of an analyte. In this error type, the contaminant of concern is unidentified and the measured concentration is incorrectly assigned to another contaminant.

Independent Assessment – an assessment performed by a qualified individual, group, or organization that is not a part of the organization directly performing and accountable for the work being assessed.

Initial Calibration – a multi-concentration, generally multi-analyte series of known concentrations analyzed periodically to establish a regression relationship that established the lower and upper concentration limits.

Inspection – the examination or measurement of an item or activity to verify conformance to specific requirements.

Instrument Detection Limit (IDL) – the lowest concentration or mass an instrument can detect above background instrument noise under ideal conditions. Sample preparation is not considered in the determination of an IDL.

Internal Standard – a standard added to a test portion of a sample in a known amount and carried through the entire determination procedure as a reference for calibrating and controlling the precision and bias of the applied analytical method.

Laboratory Blanks – samples that are used to identify potential sources of contamination that are generated during the processing and analysis of samples in the laboratory.

Laboratory Split Samples – two or more representative portions taken from the same sample and analyzed by different laboratories to estimate the interlaboratory precision or variability and the d ata comparability.

Limit of Quantitation – the minimum concentration of an analyte or category of analytes in a specific matrix that can be identified and quantified above the method detection limit and within specified limits of precision and bias during routine analytical operating conditions.

Management – those individuals directly responsible and accountable for planning, implementing, and assessing work.

Management System – a structured, nontechnical system describing the policies, objectives, principles, organizational authority, responsibilities, accountability, and implementation plan of an organization for conducting work and producing items and services.

Management Systems Review (MSR) – the qualitative assessment of a data collection operation and/or organization(s) to establish whether the prevailing quality management structure, policies,

practices, and procedures are adequate for ensuring that the type and quality of data needed are obtained.

Mass Spectrometer – an instrument detector that is used to generate mass spectra. Most often in routine environmental analysis a quadrupole mass analyzer is used to generate mass spectra, however other types of mass spectrometers can be used (e.g., magnetic sector instruments).

Mass Spectrometry – the science of using the principles of thermodynamics generate and interpret mass to charge ratios called mass spectra.

Mass Spectra – the plural of mass spectrum.

Mass Spectrum – the result of mass spectrometry. A mass spectrum is a characteristic pattern of mass to charge fragments generated from the bombardment of individual compounds in the gas phase. Mass spectra of unknown compounds can be compared to the spectra from standard mass spectral libraries.

Matrix – the component or substrate (e.g., surface water, drinking water, soil, sediment, tissue) that contains that analyte if interest.

Matrix Duplicates – an intralaboratory split sample that is used to document the precision of a method in a given sample matrix.

Matrix Spike – a sample prepared by adding a known mass of a target analyte to a specified amount of matrix sample for which an independent estimate of the target analyte concentration is available. Spiked samples are used, for example, to determine the effect of the matrix on a method's recovery efficiency.

Mean (arithmetic) – the sum of all the values of a set of measurements divided by the number of values in the set; a measure of central tendency.

Mean Squared Error – a statistical term for variance added to the square of the bias.

Measurement and Testing Equipment (M&TE) – tools, gauges, instruments, sampling devices, or systems used to calibrate, measure, test, or inspect in order to control or acquire data to verify conformance to specified requirements.

Memory Effects Error – the effect that a relatively high concentration sample has on the measurement of a lower concentration sample of the same analyte when the higher concentration sample precedes the lower concentration sample in the same analytical instrument.

Method – a body of procedures and techniques for performing an activity (e.g., sampling, chemical analysis, quantification), systematically presented in the order in which they are to be executed.

Method Blank – a blank prepared to represent the sample matrix as closely as possible and analyzed exactly like the calibration standards, samples, and quality control (QC) samples. Results of method blanks provide an estimate of the within-batch variability of the blank response and an indication of bias introduced by the analytical procedure.

Method Detection Limit – a statistically derived estimate of the lowest concentration or mass detectable under method conditions at the concentration evaluated. A series of standards at an estimated limit of detection is analyzed multiple times (usually 7), a standard deviation of these seven replicate analyses is determined and the standard deviation is multiplied by the Student's t-distribution statistic at 6 degrees of freedom (3.14). See Reporting Limit.

Mid-Range Check – a standard used to establish whether the middle of a measurement method's calibrated range is still within specifications.

Mixed Waste – a hazardous waste material as defined by 40 CFR 261 Resource Conservation and Recovery Act (RCRA) and mixed with radioactive waste subject to the requirements of the Atomic Energy Act.

Must – when used in a sentence, a term denoting a requirement that has to be met.

Nonconformance – a deficiency in a characteristic, documentation, or procedure that renders the quality of an item or activity unacceptable or indeterminate; nonfulfillment of a specified requirement.

Objective Evidence – any documented statement of fact, other information, or record, either quantitative or qualitative, pertaining to the quality of an item or activity, based on observations, measurements, or tests that can be verified.

Observation – an assessment conclusion that identifies a condition (either positive or negative) that does not represent a significant impact on an item or activity. An observation may identify a condition that has not yet caused a degradation of quality.

Organization – a company, corporation, firm, enterprise, or institution, or part thereof, whether incorporated or not, public or private, that has its own functions and administration.

Organization Structure – the responsibilities, authorities, and relationships, arranged in a pattern, through which an organization performs its functions.

Outlier – an extreme observation that is shown to have a low probability of belonging to a specified data population.

Parameter – a quantity, usually unknown, such as a mean or a standard deviation characterizing a population. Commonly misused for "variable," "characteristic," or "property."

Peer Review – a documented critical review of work generally beyond the state of the art or characterized by the existence of potential uncertainty. Conducted by qualified individuals (or an organization) who are independent of those who performed the work but collectively equivalent in technical expertise (i.e., peers) to those who performed the original work. Peer reviews are conducted to ensure that activities are technically adequate, competently performed, properly documented, and satisfy established technical and quality requirements. An in-depth assessment of the assumptions, calculations, extrapolations, alternate interpretations, methodology, acceptance criteria, and conclusions pertaining to specific work and of the documentation that supports them. Peer reviews provide an evaluation of a subject where quantitative methods of analysis or measures of success are unavailable or undefined, such as in research and development.

Percent Recovery – used as a measure of accuracy.

Performance Evaluation (PE) – a type of audit in which the quantitative data generated in a measurement system are obtained independently and compared with routinely obtained data to evaluate the proficiency of an analyst or laboratory.

Performance Evaluation (PE) Sample – a sample of known composition provided to a laboratory to determine laboratory compliance with a predetermined set of quality assurance and quality control criteria.

Pollution Prevention – an organized, comprehensive effort to systematically reduce or eliminate pollutants or contaminants prior to their generation or their release or discharge into the environment.

Practical Quantitation Limit (PQL) – a measure of the lowest limit of detection under the conditions of a particular method. The PQL is often determined by multiplying the MDL by a factor of between 3 and 10.

Precision – a measure of mutual agreement among individual measurements of the same property, usually under prescribed similar conditions expressed generally in terms of the standard deviation.

Procedure – a specified way to perform an activity.

Process – a set of interrelated resources and activities that transforms inputs into outputs. Examples of processes include analysis, design, data collection, operation, fabrication, and calculation.

Project – an organized set of activities within a program.

Qualified Data – any data that have been modified or adjusted as part of statistical or mathematical evaluation, data validation, or data verification operations.

Qualified Services – an indication that suppliers providing services have been evaluated and determined to meet the technical and quality requirements of the client as provided by approved procurement documents and demonstrated by the supplier to the client's satisfaction.

Quality – the totality of features and characteristics of a product or service that bears on its ability to meet the stated or implied needs and expectations of the user.

Quality Assurance (QA) – an integrated system of management activities involving planning, implementation, assessment, reporting, and quality improvement to ensure that a process, item, or service is of the type and quality needed and expected by the client.

Quality Assurance Program Description/Plan - (see Quality Management Plan)

Quality Assurance Project Plan (QAPP) – a formal document describing in comprehensive detail the necessary quality assurance (QA), quality control (QC), and other technical activities that must be implemented to ensure that the results of the work performed will satisfy the stated performance criteria. The QAPP components are divided into four classes: 1) Project Management, 2) Measurement/Data Acquisition, 3) Assessment/Oversight, and 4) Data Validation and Usability. Requirements for preparing QAPPs can be found in EPA QA/R-5.

Quality Control (QC) – the overall system of technical activities that measures the attributes and performance of a process, item, or service against defined standards to verify that they meet the stated requirements established by the customer; operational techniques and activities that are used to fulfill requirements for quality. The system of activities and checks used to ensure that measurement systems are maintained within prescribed limits, providing protection against "out of control" conditions and ensuring the results are of acceptable quality.

Quality Control (QC) Sample – an uncontaminated sample matrix spiked with known amounts of analytes from a source independent of the calibration standards. Generally used to establish intra - laboratory or analyst-specific precision and bias or to assess the performance of all or a portion of the measurement system.

Quality Improvement – a management program for improving the quality of operations. Such management programs generally entail a formal mechanism for encouraging worker recommendations with timely management evaluation and feedback or implementation.

Quality Management – that aspect of the overall management system of the organization that determines and implements the quality policy. Quality management includes strategic planning, allocation of resources, and other systematic activities (e.g., planning, implementation, and assessment) pertaining to the quality system.

Quality Management Plan (QMP) – a formal document that describes the quality system in terms of the organization's structure, the functional responsibilities of management and staff, the lines of authority, and the required interfaces for those planning, implementing, and assessing all activities conducted.

Quality System – a structured and documented management system describing the policies, objectives, principles, organizational authority, responsibilities, accountability, and implementation plan of an organization for ensuring quality in its work processes, products (items), and services. The quality system provides the framework for planning, implementing, and assessing work performed by the organization and for carrying out required quality assurance (QA) and quality control (QC).

Radioactive Waste – waste material containing, or contaminated by, radionuclides, subject to the requirements of the Atomic Energy Act.

Readiness Review – a systematic, documented review of the readiness for the start-up or continued use of a facility, process, or activity. Readiness reviews are typically conducted before proceeding beyond project milestones and prior to initiation of a major phase of work.

Record (quality) – a document that furnishes objective evidence of the quality of items or activities and that has been verified and authenticated as technically complete and correct. Records may include photographs, drawings, magnetic tape, and other data recording media.

Recovery – the act of determining whether or not the methodology measures all of the analyte contained in a sample.

Relative Standard Deviation (RSD) – a measure of precision calculated by: RSD = S/x X 100.

Remediation – the process of reducing the concentration of a contaminant (or contaminants) in air, water, or soil media to a level that poses an acceptable risk to human health.

Repeatability – the degree of agreement between independent test results produced by the same analyst, using the same test method and equipment on random aliquots of the same sample within a short time period.

Reporting Limit (RL) – for a target analyte, the reporting limit is instrument dependent and based on the lowest point on the calibration curve.

Representativeness – a measure of the degree to which data accurately and precisely represent a characteristic of a population, a parameter variation at a sampling point, a process condition, or an environmental condition.

Reproducibility – the precision, usually expressed as variance, that measures the variability among the results of measurements of the same sample at different laboratories.

Requirement – a formal statement of a need and the expected manner in which it is to be met.

Research (applied) – a process, the objective of which is to gain the knowledge or understanding necessary for determining the means by which a recognized and specific need may be met.

Research (basic) – a process, the objective of which is to gain fuller knowledge or understanding of the fundamental aspects of phenomena and of observable facts without specific applications toward processes or products in mind.

Research Development/Demonstration – the systematic use of the knowledge and understanding gained from research and directed toward the production of useful materials, devices, systems, or methods, including prototypes and processes.

Round-Robin Study – a method validation study involving a predetermined number of laboratories or analysts, all analyzing the same sample(s) by the same method. In a round-robin study, all results are compared and used to develop summary statistics such as interlaboratory precision and method bias or recovery efficiency.

Ruggedness Study – the carefully ordered testing of an analytical method while making slight variations in test conditions (as might be expected in routine use) to determine how such variations affect test results. If a variation affects the results significantly, the method restrictions are tightened to minimize this variability.

Sampling and Analysis Plan – a document consisting of a description of the quality assurance, field sampling steps and laboratory analysis for a given environmental data gathering effort.

Scientific Method – the principles and processes regarded as necessary for scientific investigation, including rules for concept or hypothesis formulation, conduct of experiments, and validation of hypotheses by analysis of observations.

Self-Assessment – the assessments of work conducted by individuals, groups, or organizations directly responsible for overseeing and/or performing the work.

Sensitivity – the capability of a method or instrument to discriminate between measurement responses representing different levels of a variable of interest.

Service – the result generated by activities at the interface between the supplier and the customer, and the supplier internal activities to meet customer needs. Such activities in environmental programs include design, inspection, laboratory and/or field analysis, repair, and installation.

Shall – a term denoting a requirement that is mandatory whenever the criterion for conformance with the specification permits no deviation. This term does not prohibit the use of alternative approaches or methods for implementing the specification so long as the requirement is fulfilled.

Significant Condition – any state, status, incident, or situation of an environmental process or condition, or environmental technology in which the work being performed will be adversely affected sufficiently to require corrective action to satisfy quality objectives or specifications and safety requirements.

Software Life Cycle – the period of time that starts when a software product is conceived and ends when the software product is no longer available for routine use. The software life cycle typically includes a requirement phase, a design phase, an implementation phase, a test phase, an installation and check-out phase, an operation and maintenance phase, and sometimes a retirement phase.

Source Reduction – any practice that reduces the quantity of hazardous substances, contaminants, or pollutants.

Span Check – a standard used to establish that a measurement method is not deviating from its calibrated range.

Speciation – the ability of a chemical to exists in several forms (e.g., valence state, complexation). Speciation is generally controlled by redox conditions and pH.

Specification – a document stating requirements and referring to or including drawings or other relevant documents. Specifications should indicate the means and criteria for determining conformance.

Spike – a substance that is added to an environmental sample to increase the concentration of target analytes by known amounts; used to assess measurement accuracy (spike recovery). Spike duplicates are used to assess measurement precision.

Split Samples – two or more representative portions taken from one sample in the field or in the laboratory and analyzed by different analysts or laboratories. Split samples are quality control (QC) samples that are used to assess analytical variability and comparability.

Standard Deviation – a measure of the dispersion or imprecision of a sample or population distribution expressed as the positive square root of the variance and has the same unit of measurement as the mean.

Standard Operating Procedure (SOP) – a written document that details the method for an operation, analysis, or action with thoroughly prescribed techniques and steps and that is officially approved as the method for performing certain routine or repetitive tasks.

Standard Reference Material – a quality control tool that is a matrix-specific, single or multi-component mixture of analytes that has been certified by an entity external to the laboratory performing the analysis for a particular concentration. This quality control element is used to monitor accuracy.

Supplier – any individual or organization furnishing items or services or performing work according to a procurement document or a financial assistance agreement. An all-inclusive term used in place of any of the following: vendor, seller, contractor, subcontractor, fabricator, or consultant.

Surrogate Spike or Analyte – a pure substance with properties that mimic the analyte of interest. It is unlikely to be found in environmental samples and is added to them to establish that the analytical method has been performed properly.

Surrogate Standard – a standard of known concentration added to environmental samples for quality control purposes. A surrogate standard is unlikely to be found in environmental samples but has similar properties. Surrogates are intended to monitor for recovery differences, problems during the extraction phase of the analysis and for any potential matrix interferences.

Surveillance (quality) – continual or frequent monitoring and verification of the status of an entity and the analysis of records to ensure that specified requirements are being fulfilled.

Target Analyte – a compound associated with a particular analytical method.

Technical Review – a documented critical review of work that has been performed within the state of the art. The review is accomplished by one or more qualified reviewers who are independent of those who performed the work but are collectively equivalent in technical expertise to those who performed th e original work. The review is an in-depth analysis and evaluation of documents, activities, material, data, or items that require technical verification or validation for applicability, correctness, adequacy, completeness, and assurance that established requirements have been satisfied.

Technical Systems Audit (TSA) – a thorough, systematic, on-site qualitative audit of facilities, equipment, personnel, training, procedures, record keeping, data validation, data management, and reporting aspects of a system.

Tentatively Identified Compound (TIC) – compounds that are detected in environmental samples that are not target analytes. TICs are identified generally as a result of using mass spectrometry techniques. When a TIC is identified, it can be definitively identified by analyzing an authentic standard of the putative unknown.

Total Petroleum Hydrocarbons (TPH) – TPH is an operationally defined parameter with no universally accepted definition, therefore it is incumbent upon the ultimate data user to establish a working definition of TPH on a project specific basis. Some useful definitions are presented here.

MCP Definition – the MCP defines TPH as all compounds containing carbon and hydrogen only from nC9 through nC36.

Traceability – the ability to trace the history, application, or location of an entity by means of recorded identifications. In a calibration sense, traceability relates measuring equipment to national or international standards, primary standards, basic physical constants or properties, or reference materials. In a data collection sense, it relates calculations and data generated throughout the project back to the requirements for the quality of the project.

Trip Blank – a clean sample of a matrix that is taken to the sampling site and transported to the laboratory for analysis without having been exposed to sampling procedures.

Validation – confirmation by examination and provision of objective evidence that the particular requirements for a specific intended use have been fulfilled. In design and development, validation concerns the process of examining a product or result to determine conformance to user needs.

Variance (statistical) – a measure or dispersion of a sample or population distribution.

Verification – confirmation by examination and provision of objective evidence that specified requirements have been fulfilled. In design and development, verification concerns the process of examining a result of a given activity to determine conformance to the stated requirements for that activity.

6. Discharge Monitor Technical Reference

Technical Prologue to DMR Usage

Introduction

This chapter is designed for the technical specialist or designated system administrator of the LABWORKS 6.4 Discharge Monitoring system. Generally the person(s) assigned these tasks will handle software installation, system configuration, and database preparation (of permits, discharges/outfalls, and requirements). These tasks require computer / database technical skills outside of laboratory/environmental specific duties. This documentation is tailored for people handling these duties and is not designed for the operator of the software. If you plan on operating the software to generate monthly DMR reports, please refer to the LABWORKS Discharge Monitor User Reference in the next chapter.

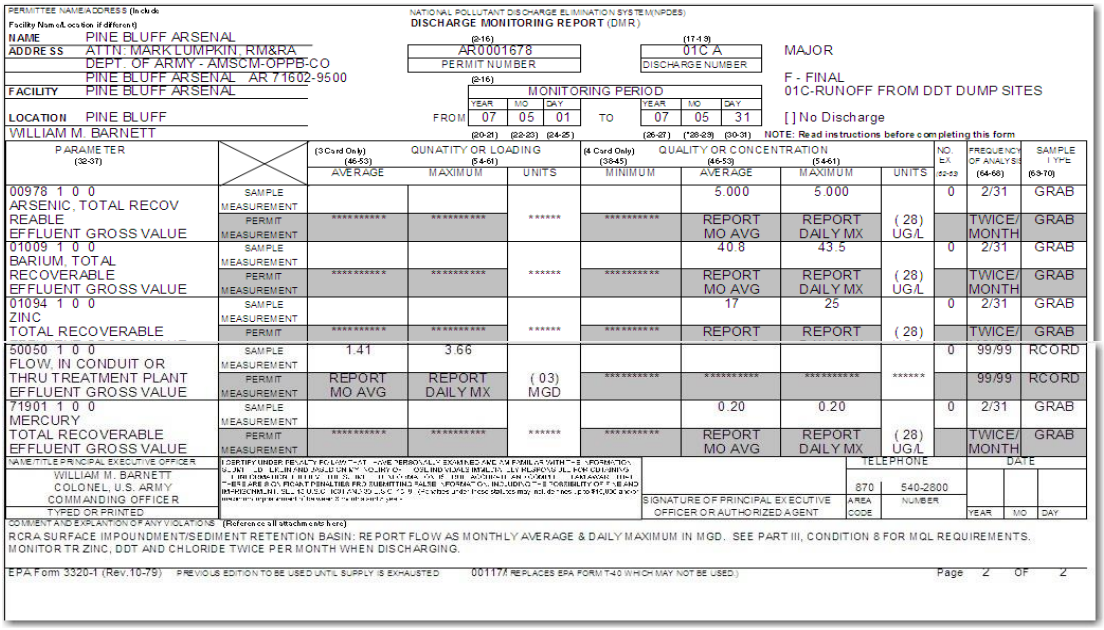
This document also assumes the Server and Client installs of LABWORKS 6.4 has been completed successfully. Installation of LABWORKS is not covered.

Overview

A technical specialist who is in charge of DMR implementation is obviously not required to know all the specifics of the business. However, a good understanding of the purpose/goal of the system along with why it is needed is always good for those responsible for keeping it running properly. The following is a short background and overview of the business and DMR module.

Water pollution degrades surface waters making them unsafe for drinking, fishing, swimming, and other activities. In the United States, authorized by the Clean Water Act, the National Pollutant Discharge Elimination System (NPDES) permit program controls water pollution by regulating point sources that discharge pollutants into public bodies of water. Point sources are discrete conveyances such as pipes or man-made ditches. Individual homes that are connected to a municipal system, use a septic system, or do not have a surface discharge do not need an NPDES permit; however, industrial, municipal, and other facilities must obtain permits if their discharges go directly to surface waters. In most cases, the NPDES permit program is administered by individual authorized states. Since its introduction in 1972, the NPDES permit program is responsible for significant improvements to the nation's water quality. Other nations outside of the United States are taking close observance to the NPDES permit system of water quality, and many are beginning to adopt very similar systems.

Different state governments may prefer samples reported differently (calculations of average results, minimums, maximums, for the week, day, etc.), but the end result is always a similar-looking spreadsheet-type report. Below is an example page from a final monthly DMR report.



NATIONAL POLLUTANT DISCHARGE ELIMINATION SYSTEM (NPDES)
DISCHARGE MONITORING REPORT (DMR)

PERMITTEE NAME/ADDRESS (Include Facility Name/Location if different)
NAME PINE BLUFF ARSENAL
ADDRESS ATTN: MARK LUMPKIN, RM&RA
DEPT. OF ARMY - AMSCM-OPPB-CO
PINE BLUFF ARSENAL AR 71602-9500
FACILITY PINE BLUFF ARSENAL
LOCATION PINE BLUFF
WILLIAM M. BARNETT

(2-16) PERMIT NUMBER: AR0001678
(17-19) DISCHARGE NUMBER: 01C A
MAJOR
F - FINAL
01C-RUNOFF FROM DDT DUMP SITES
[ ] No Discharge

(2-16) MONITORING PERIOD
FROM YEAR 07 MO 05 DAY 01 TO YEAR 07 MO 05 DAY 31
(20-21) (22-23) (24-25) (26-27) (28-29) (30-31) NOTE: Read instructions before completing this form

PARAMETER (32-37)		QUANTITY OR LOADING (3 Card Only) (45-53) AVERAGE	QUANTITY OR LOADING (54-61) MAXIMUM	QUANTITY OR LOADING UNITS	QUALITY OR CONCENTRATION (4 Card Only) (38-45) MINIMUM	QUALITY OR CONCENTRATION (45-53) AVERAGE	QUALITY OR CONCENTRATION (54-61) MAXIMUM	QUALITY OR CONCENTRATION UNITS	NO. EX. 62-63	FREQUENCY OF ANALYSIS (64-68)	SAMPLE TYPE (69-70)
00978 1 0 0 ARSENIC, TOTAL RECOV REABLE	SAMPLE MEASUREMENT					5.000	5.000		0	2/31	GRAB
EFFLUENT GROSS VALUE	PERMIT MEASUREMENT	**********	**********	******	**********	REPORT MO AVG	REPORT DAILY MX	(28) UG/L		TWICE/ MONTH	GRAB
01009 1 0 0 BARIUM, TOTAL RECOVERABLE	SAMPLE MEASUREMENT					40.8	43.5		0	2/31	GRAB
EFFLUENT GROSS VALUE	PERMIT MEASUREMENT	**********	**********	******	**********	REPORT MO AVG	REPORT DAILY MX	(28) UG/L		TWICE/ MONTH	GRAB
01094 1 0 0 ZINC TOTAL RECOVERABLE	SAMPLE MEASUREMENT					17	25		0	2/31	GRAB
	PERMIT MEASUREMENT	**********	**********	******	**********	REPORT	REPORT	(28)		TWICE/ MONTH	GRAB
50050 1 0 0 FLOW, IN CONDUIT OR THRU TREATMENT PLANT	SAMPLE MEASUREMENT	1.41	3.66						0	99/99	RCORD
EFFLUENT GROSS VALUE	PERMIT MEASUREMENT	REPORT MO AVG	REPORT DAILY MX	(03) MGD	**********	**********	**********	******		99/99	RCORD
71901 1 0 0 MERCURY TOTAL RECOVERABLE	SAMPLE MEASUREMENT					0.20	0.20		0	2/31	GRAB
EFFLUENT GROSS VALUE	PERMIT MEASUREMENT	**********	**********	******	**********	REPORT MO AVG	REPORT DAILY MX	(28) UG/L		TWICE/ MONTH	GRAB

NAME/TITLE PRINCIPAL EXECUTIVE OFFICER
WILLIAM M. BARNETT
COLONEL, U.S. ARMY
COMMANDING OFFICER
TYPED OR PRINTED

I CERTIFY UNDER PENALTY OF LAW THAT I HAVE PERSONALLY EXAMINED AND AM FAMILIAR WITH THE INFORMATION SUBMITTED IN THIS DOCUMENT AND ALL ATTACHMENTS AND THAT, BASED ON MY INQUIRY OF THOSE INDIVIDUALS IMMEDIATELY RESPONSIBLE FOR OBTAINING THE INFORMATION, I BELIEVE THAT THE INFORMATION IS TRUE, ACCURATE AND COMPLETE. I AM AWARE THAT THERE ARE SIGNIFICANT PENALTIES FOR SUBMITTING FALSE INFORMATION, INCLUDING THE POSSIBILITY OF FINE AND IMPRISONMENT FOR KNOWING VIOLATIONS.

SIGNATURE OF PRINCIPAL EXECUTIVE OFFICER OR AUTHORIZED AGENT

TELEPHONE: AREA CODE 870; NUMBER 540-2800

DATE: YEAR; MO; DAY

COMMENT AND EXPLANATION OF ANY VIOLATIONS (Reference all attachments here)
RCRA SURFACE IMPOUNDMENT/SEDIMENT RETENTION BASIN: REPORT FLOW AS MONTHLY AVERAGE & DAILY MAXIMUM IN MGD. SEE PART III, CONDITION 8 FOR MQL REQUIREMENTS. MONITOR TR ZINC, DDT AND CHLORIDE TWICE PER MONTH WHEN DISCHARGING.

EPA Form 3320-1 (Rev.10-79) PREVIOUS EDITION TO BE USED UNTIL SUPPLY IS EXHAUSTED 00117A REPLACES EPA FORM T-40 WHICH MAY NOT BE USED.) Page 2 OF 2

DMR Module History/Background (the “Story”)

In the mid-90’s, a growing number of LABWORKS customers which included industrial and commercial facilities, municipalities and wastewater treatment plants, etc. needed a method of an automatic generation of the required monthly NPDES report. Local (state) governments require a specifically formatted report with results for specific analyses (called *requirements*) from each designated outfall (called *discharges*). These outfalls, or discharges, as described in the above overview, are specific locations, or *point sources*, where pollution into a public body of water could potentially occur. These samples on required outflow analyses (such as PH, BOD, CBOD, TSS, etc.) were generally taken on a daily or weekly basis and stored in LABWORKS, but our customers needed a way to take these samples and transfer them onto a specially formatted report to meet state government requirements. Failure to meet these requirements could result in a large fine or the eventual closure of the facility. The NPDES report is generally downloaded from the website in Microsoft Excel (XLS) format and was very detailed and time-consuming to prepare manually. Pressure by LABWORKS customers continued for a solution to the time-consuming preparation of the NPDES permit report. This lead to the development of the DMR, *Discharge Monitoring Report*, module.

The DMR module was never a core-application. It was initially written as a custom application with its own Microsoft Access database (\LWDATA\DMR\DMR97.MDB) populated by individual calls to the core LABWORKS database via general middle tier queries for sample information.

DMR reporting is generally done for a single month. Customers use the DMR module to pole all samples in LABWORKS for a given month and parse that data into an Excel report. Samples are designated for DMR by the use of user fields. The DMR application looks for any samples with a matching permit number (generally a user field called PERMIT_NO) and discharge number (generally a user field called DICHARGE_NO). These samples will be pulled into DMR for the specific permit being printed for that date range selected. The old application would obtain this sample list through SQL queries, then copy the samples and corresponding tests into its own Access database file (DMR97.MDB) while building calculations and formatting records for report preparation. After this “calculation” process, the final Excel DMR for the month can be generated and turned into the state government.

The old DMR module has had numerous support challenges over the years mostly from the following.

Incompatible Field Sizes – LABWORKS core database fields that change with new updates do not match within the old DMR Access database forcing IT to manually update DMR Access tables

Calculation Challenges – customers in a specific state needing values calculated a little differently from customers in other states forced LABWORKS to provide custom calculation functions and maintain them for different accounts

Date Challenges – some customers require 'partial weeks' or weeks that are at the beginning or end of a month containing less than a full 7 days while other customers may allow an expansion into the following or previous month to obtain full weeks

Configuration Problems – location codes change and no longer match permit or discharge data, analysis codes change and no longer match requirement data because no integration with the core database exists

Database Corruption – any problem with the separate Access database would make the system unusable adding work for the IT department to maintain a separate database file

Clunky User Interface – training new users of legacy application has always been a problem for our customers since the system has no standard Windows user interface and requires going through many different screens to achieve the end result (the DMR report)

The new DMR application, called *DischargeMonitor*, has been introduced as an optional product within the LABWORKS 6.2 package using the Microsoft.NET framework. These problems that have plagued the legacy application for years has been engineered out of this new module.

Generally only commercial or waste/drinking water labs are DMR customers. However, the users of the application rely heavily on the tool not only for regulatory reporting but also (in the case of commercial labs) for contract report preparation for their own lab customers as a significant source of revenue.

The LABWORKS Support department gets more calls on the legacy DMR application than nearly any other module, and it is also the most challenging to support. A support call on DMR will generally always be forwarded to development taking even longer to provide a solution back to the customer. For customers currently running the legacy DMR tool, the benefits of upgrading to the new *DischaregeMonitor* module could greatly improve lab productivity and report turn-around along with the added benefits of less maintenance and problems for IT.

DMR Common Terminology

The following is a list of some common DMR terminology that may be second nature to folks involved in DMR preparation, but certainly not for the system administrator or IT professional who maintains the system. This information can be used as a reference to educate administrators to speak the same lingo as those operating the DMR system.

Effluent

The outflow or discharge itself.

Composite Sample

A sample that is based on a collection (or *composite*) of continuous samples from a discharge over a given time period. For domestic wastewater, a sample made up of a minimum of three effluent portions collected in a continuous 24-hour period or during the period of daily discharge if less than 24 hours, and combined in volumes proportional to flow, and collected no closer than two hours apart. For industrial wastewater, a composite sample is a sample made up of a minimum of three effluent portions collected in a continuous 24-hour period or during the period of daily discharge if less than 24 hours, and combined in volumes proportional to flow, and collected no closer than one hour apart. Various state governments may define the composite differently.

Grab Sample

A sample that is collected as-is only once (hence the term “*grab*” which suggests the analyst simply grabs the single sample) from the discharge.

7-day Average Concentration

The arithmetic average of all effluent samples, composite or grab as required by the permit, within a period of one calendar week, Sunday through Saturday (or how ever their state government defines their reporting calendar week).

Concentration Average

Daily average concentration is the arithmetic average of all effluent samples, composite or grab as required by the permit, within a period of one calendar month, consisting of at least four sepa rate representative measurements.

For domestic wastewater treatment plants - When four samples are not available in a calendar month, the arithmetic average (weighted by flow) of all values in the previous four consecutive month period consisting of at least four measurements shall be utilized as the daily average concentration.

For all other wastewater treatment plants - When four samples are not available in a calendar month, the arithmetic average (weighted by flow) of all values taken during the month shall be utilized as the daily average concentration.

Fecal coliform bacteria concentration - the number of colonies of fecal coliform bacteria per 100 milliliters effluent. The daily average fecal coliform bacteria concentration is a geometric mean of the values for the effluent samples collected in a calendar month. The geometric mean shall be determined by calculating the nth root of the product of all measurements made in a calendar month, where n equals the number of measurements made; or, computed as the antilogarithm of the arithmetic mean of the logarithms of all measurements made in a calendar month. For any measurement of fecal coliform bacteria equaling zero, a substituted value of one shall be made for input into either computation method.

Concentration Maximum

Daily maximum concentration - Daily maximum concentration is the maximum concentration measured on a single day, by the sample type specified in the permit, within a period of one calendar month.

7-day average concentration - 7-day average concentration is the arithmetic average of all effluent samples, composite or grab as required by this permit, within a period of one calendar week, Sunday through Saturday.

Fecal coliform bacteria concentration - The 7-day average for fecal coliform bacteria is the geometric mean of the values for all effluent samples collected during a calendar week.

Concentration Minimum

The lowest measured value during the monitoring period is to be reported in this field unless otherwise specified in the permit. Examples of parameters with minimum values to be reported are pH, Dissolved Oxygen, Chlorine, and any other parameters with minimum limits listed in a permit.

Concentration Units

Unit of measure is specified in the permit and cannot be changed.

Flow Terminology

2-hour Peak Flow - Applies to domestic wastewater treatment plants: The maximum flow sustained for a two-hour period during the period of daily discharge. Multiple measurements of instantaneous maximum flow within a two-hour period may be compared to the permitted 2-hour peak flow.

Annual Average Flow - The arithmetic average of all daily flow determinations taken within the preceding twelve consecutive calendar months. The annual average flow determination shall consist of daily flow volume determinations made by a totalizing meter, charted on a chart recorder and limited to major domestic wastewater discharge facilities with a 1 million gallons per day or greater permitted flow.

Daily Average Flow - The arithmetic average of all determinations of the daily discharge within a period of one calendar month. The daily average flow determination shall consist of determinations made on at least four separate days. If instantaneous measurements are used to determine the daily discharge, the determination shall be the arithmetic average of all instantaneous measurements taken during that month. Daily average flow determination for intermittent discharges shall consist of a minimum of three flow determinations on days of discharge.

Daily Maximum Flow - The highest total flow for any 24-hour period in a calendar month.

Instantaneous Flow - The measured flow during the minimum time required to interpret the flow measuring device.

Maximum 2-hour Peak Flow - Applies to domestic wastewater treatment plants: The highest 2-hour peak flow for any 24-hour period in a calendar month.

Frequency of Analysis

Minimum frequency of analysis is specified in the permit. Actual frequency of analysis could be more frequent. The default value for this field is the frequency specified in the permit, but it can be changed.

When the actual frequency of analysis for sampling does not match what is required by the permit, choose the appropriate frequency from the drop-down list located on the edit parameter page. If you have sampled at a frequency which does not have a corresponding description/code available in the drop-down list, select "OTHER (XX/XX)". If you choose this option, you must include a description of the actual frequency of analysis for the parameter in the comments field on the edit parameter page.

No Data Reason

A brief explanation of why measurements cannot be submitted. Several options are available in the drop-down list. If you cannot find a corresponding explanation in this list, choose "8 - OTHER". A comment must be supplied when a No Data Indicator is chosen. For a complete list of the available codes see PCS No Data Indicator Codes.

Number of Exceedances

For DMR reporting, the definition of exceedance is number of sample measurements during the monitoring period that exceed maximum (and/or minimum or 7-day average as appropriate) permit requirement for each parameter. If none, enter "0". DO NOT include monthly average or daily average violations in this field. Permittees with continuous pH, or temperature monitoring requirements should consult the permit for what constitutes an exceedance and report accordingly.

Parameter

Specified in the permit as effluent characteristics for each outfall, a parameter can be a chemical, metal, constituent, property, or other factor which is measured, analyzed, and evaluated to determine effluent (or sludge) quality or quantity.

Parameter Status

After validation of the measurement data the parameter will have one of these status types:

Error - Errors were found when the parameter data was validated. It cannot be submitted until the errors are corrected.

Valid - No errors or warnings were found when the parameter data was validated. The parameter can be submitted.

Warn - Warnings were found when the parameter data was validated. The parameter can be submitted. Warnings include things like Quantity Average and Concentration Average are the same or when measurements have exceeded the limit values.

Submitted - The parameter data has been submitted but not yet processed. The data is an official TCEQ record but has not yet been processed. The parameter cannot be edited or deleted in this state.

Processed - The parameter data has been processed and loaded into the EPA system. It can be edited or deleted in this state.

Permit Limits

Limits as specified in the permit for each parameter.

Quantity Average

Daily average loading (lbs/day) - Daily average loading (lbs/day) is the arithmetic average of all daily discharge loading calculations during a period of one calendar month. These calculations must be made for each day of the month that a parameter is analyzed. The daily discharge, in terms of mass (lbs/day), is calculated as (Flow, MGD x Concentration, mg/l x 8.34).

Annual average flow - Annual average flow is the arithmetic average of all daily flow determinations taken within the preceding 12 consecutive calendar months. The annual average flow determination shall consist of daily flow volume determinations made by a totalizing meter, charted on a chart recorder and limited to major domestic wastewater discharge facilities with a one million gallons per day or greater permitted flow.

Daily average flow - Daily average flow is the arithmetic average of all determinations of the daily flow within a period of one calendar month. The daily average flow determination shall consist of determinations made on at least four separate days. If instantaneous measurements are used to determine the daily flow, the determination shall be the arithmetic average of all instantaneous measurements taken during that month. Daily average flow determination for intermittent discharges shall consist of a minimum of three flow determinations on days of discharge.

Quantity Maximum

Daily maximum loading - Daily maximum loading (lbs/day) is the highest daily discharge, in terms of mass (lbs/day), within a period of one calendar month.

Daily maximum flow - Daily maximum flow is the highest total flow for any 24-hour period in a calendar month. 2-hour peak flow (domestic wastewater treatment plants) is the maximum flow sustained for a two-hour period during the period of daily discharge. The average of multiple measurements of instantaneous maximum flow within a two-hour period may be used to calculate the 2-hour peak flow.

Maximum 2-hour peak flow - Maximum 2-hour peak flow (domestic wastewater treatment plants) is the highest 2-hour peak flow for any 24-hour period in a calendar month.

Quantity Units

Unit of measure is specified in the permit and cannot be changed.

Sample Type

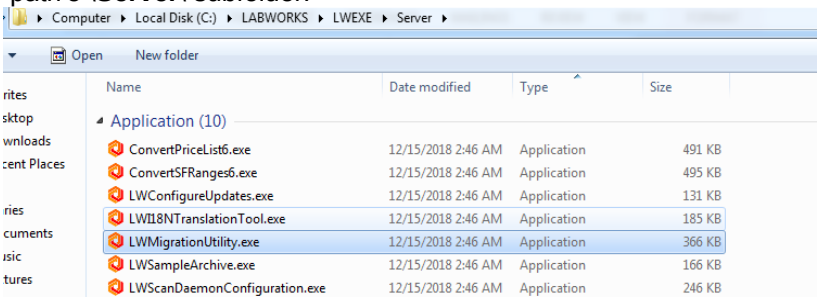
Sample Type is specified in the permit. The default value for this field is the sample type specified in the permit, but it can be changed if the actual sampling method used is different.

Existing Users: Legacy Migration

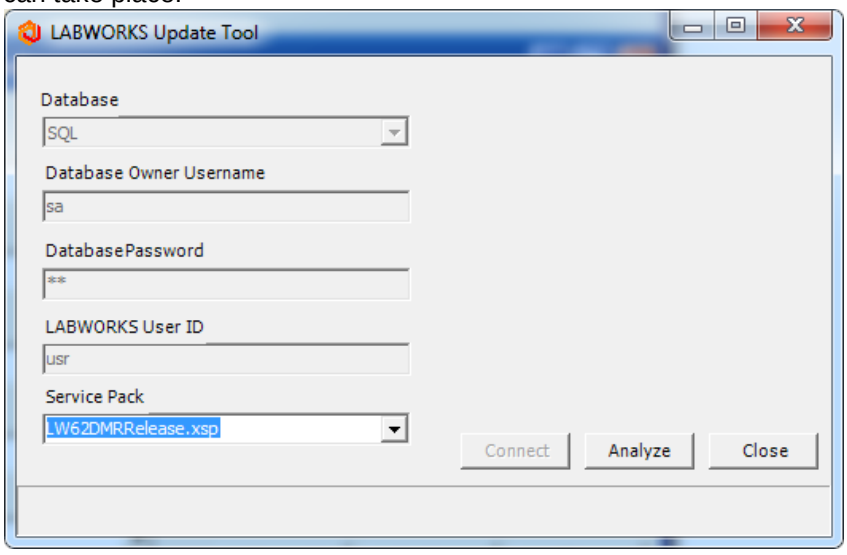
Migrating old Access Database

After LABWORKS 6.2 is installed (with a proper control file license for DMR as well), the next step before the new system can be used is to migrate the data from the legacy system into the core database.

The tool to perform this migration is called the **LWMigrationUtility** and is located under the license path's **Server**\ subfolder.

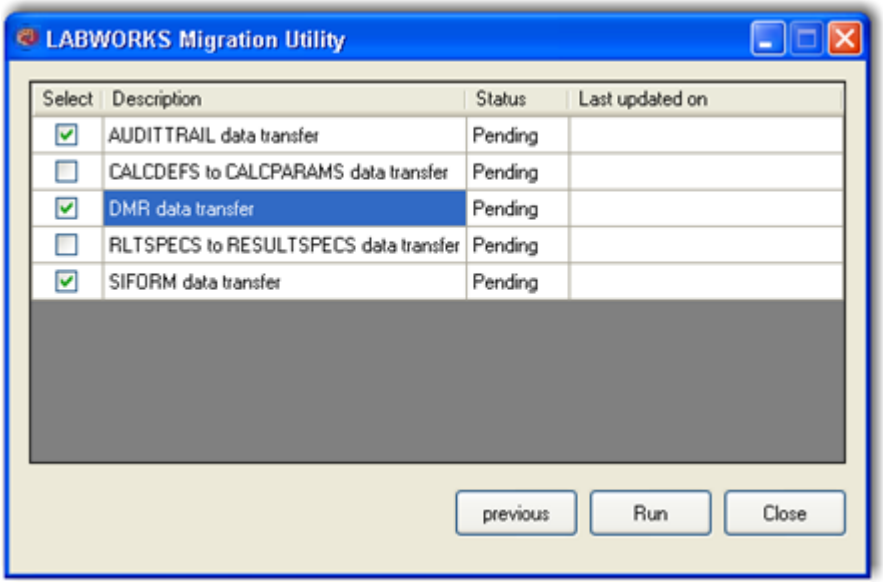


When the migration module is launched, it will initially prompt you to run the **UpdateTool** program. If you have not already done this and executed the LW62 and DMR related service packs, you must complete this first. This is required in order to add the necessary tables to your database before data migration can take place.

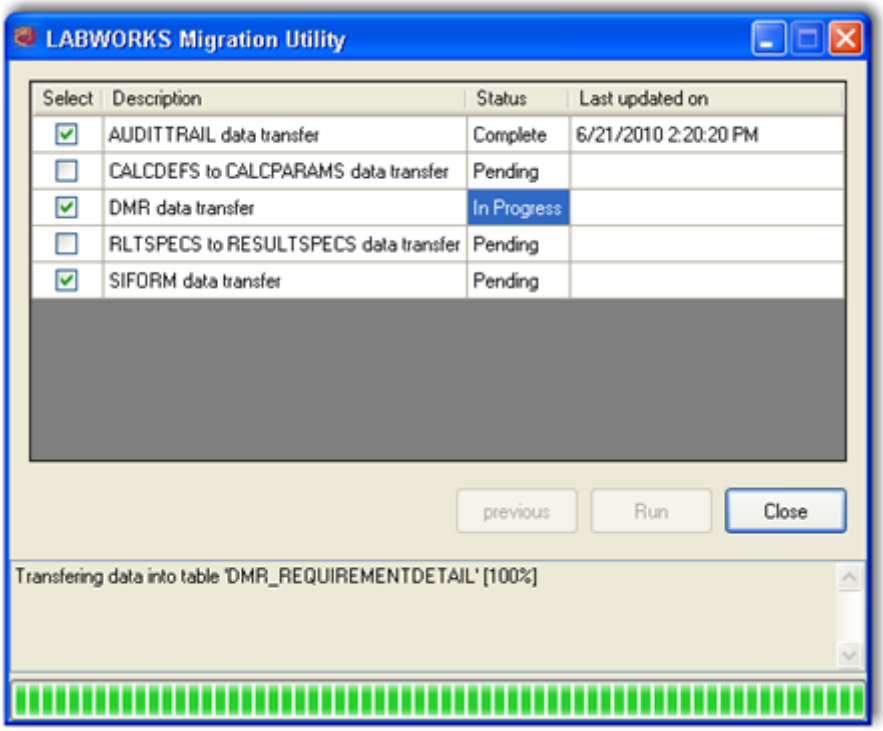


After connecting to the database, make sure and successfully complete all service packs, but in particular **LW62Release.xsp** and **LW62DMRRelease.xsp**.

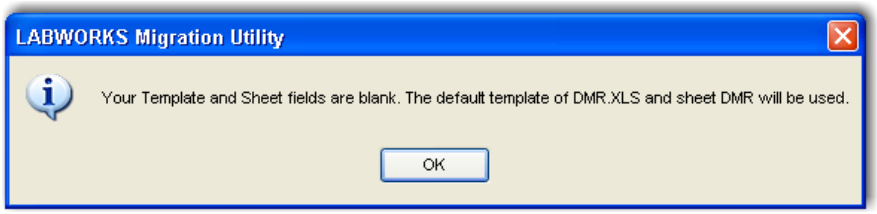
When the **UpdateTool** service packs have successfully executed, the migration tool can continue. There are a few different legacy data migration processes that can be selected (and probably should), however the only migration required for new DMR is the **DMR data transfer** option. Make sure at least this option is selected, then click **Run**.



The data migration will then begin.

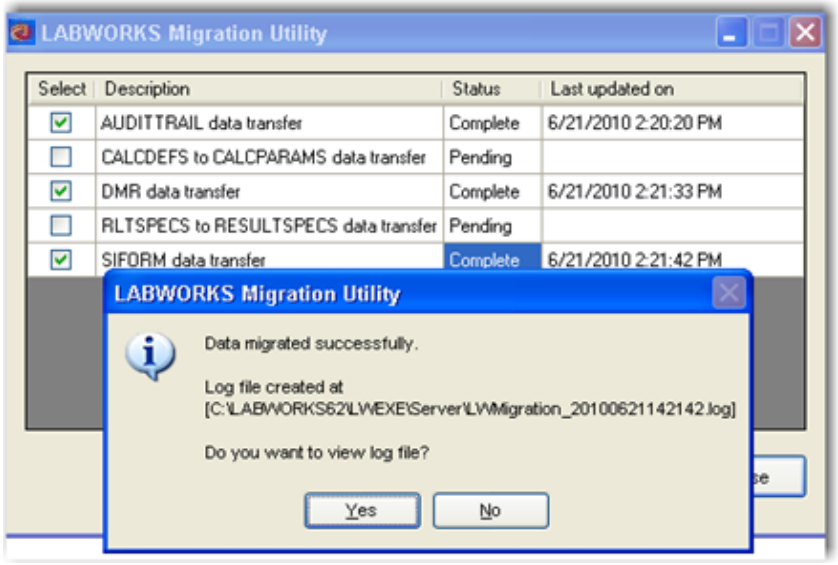


Note that during the migration process, you may get a prompt telling you that your template and sheet fields are blank and that the default Excel template of DMR.XLS will be used.



This is not an error. It is merely a notice. Early versions of legacy DMR did not give you the option of template selection. Only one Excel template, called DMR.XLS, was required. However, later versions of the legacy tool (LABWORKS 5.8) was enhanced to provide the capability of template selection giving customers the ability to have more than one specific final DMR report format. However, if the user never defined a specific template, it would always use the default DMR.XLS template and never prompt the user. In the new *DischargeMonitor* application, the user can still select from multiple templates at print time, but a designated default template is required in the database. so the migration tool automatically adds a reference to the DMR.XLS template file if no template is specified in the legacy tables.

If the migration completes successfully, it will show ***Complete*** under the **Status** column and provide an option to view the log file which can be printed for more technical detail on data changes that were performed.



User Field Requirements

After data migration, there is an important LABWORKS user field database modification that you also may be required to do before using the new application.

The legacy application supported several different user field versions of the DMR Permit Number field (PERMIT_NUM, PERMIT_NO, PERMITNO, etc.) as well as the DMR Discharge/Outfall Number (OUTFALL_NUM, DISCHNO, DISCHARGE_NO, etc.). However, in the new *DischargeMonitor* application, there is only one specific user field that is accepted for each of these.

PERMIT_NO	varchar(15)
DISCHARGE_NO	varchar(10)

PERMIT_NO – required user field for permit number

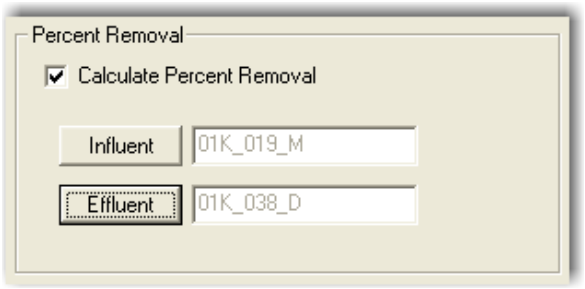
DISCHARGE_NO – required user field for outfall/discharge number

If your database tables (**USERFLDS** and **SUSERFLDS**) have any other spelling for these two user fields, you *must* change them. The new *DischargeMonitor* application will not recognize any other user field. The calculation of final records for the DMR report will not find any samples without correct user fields.

Requirement Record Changes

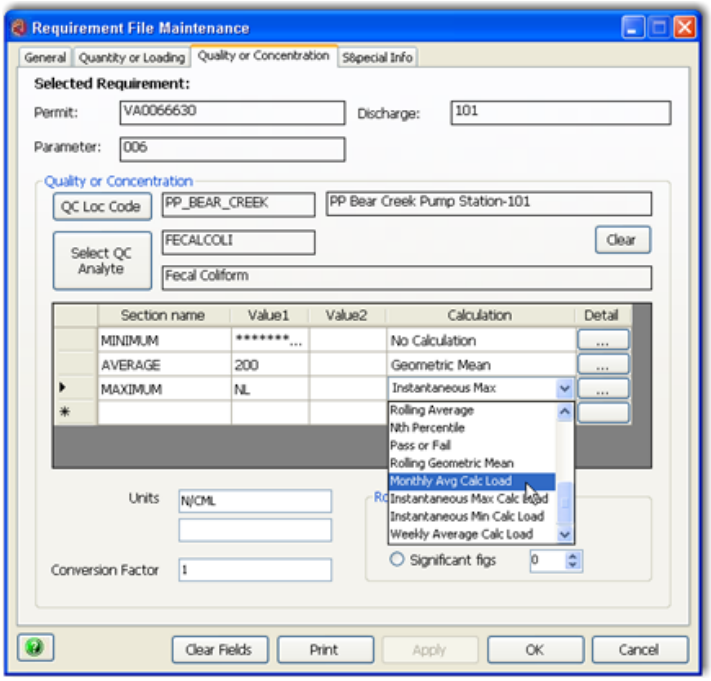
At this point, you should be able to open DMR and run the application. After opening the new module, you will see a nice Windows Explorer-type user interface. All permits, discharges, and requirements should open into the tree on the left with a listing of records on the right based on your tree selection. The system may run okay for maintenance, but there could be minor changes that need to happen within some requirement records before the report will accurately print.

The legacy application had an option (checkbox) for percent removal. With this option checked, certain calculations are adjusted during the calculation of final records for the report. For example, when a Monthly Average calculation is used with Calculate Percent Removal checked, it will use a designated loading (FLOW) value to apply to those samples. Certain permits require a load-balanced calculation. In other words, the calculation must be based on measured flow values (more / less velocity and quantity of discharge). Obviously, the same amount of coliform in a large outfall will show a much more diluted result than a small outfall.



In the new DMR application, the load-based calculations were added as selectable calculations themselves instead of a check box.

For example, if you want a standard monthly average calculation, you could select **Monthly Average** in the Requirement Maintenance screen. If you want this load-based, you would select the **Monthly Avg Calc Load**.



This must be done for each load-based requirement.

As a final note, you may see a different calculation list than displayed in the screen shot above. This is because the new *DischargeMonitor* application uses calculation DLL files to define this list. Each calculation is a separate function inside the DLL files with designated inputs and outputs used by DMR. This allows for the simple installation of custom DMR calculation DLLs. See the next section for more detail.

DMR Calculation Installations

DMR DLL Files

The new *DischargeMonitor* application performs all detailed processing of records via external system files. There is a standard and custom DLL file that defines the list of available calculations selectable within the Requirements screen. If you open a requirement within the new DMR, and there are no calculations available for selection, then you are missing these files.

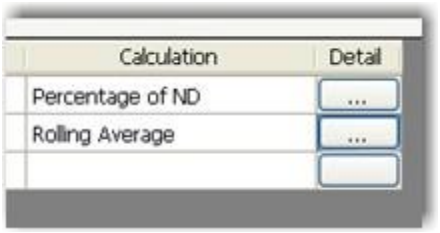
The application can accept only two calculation DLL files, one is "**STANDARD**" and other is "**CUSTOM**". If any client needs an extra custom calculation than the provided one, then **System Manager** or a **Field person** has to replace the client's current custom DLL with a new file. However, the standard DLL must always be included in the system. This is the base list of calculations most common in DMR.

Adding Custom DLL in Client Environment

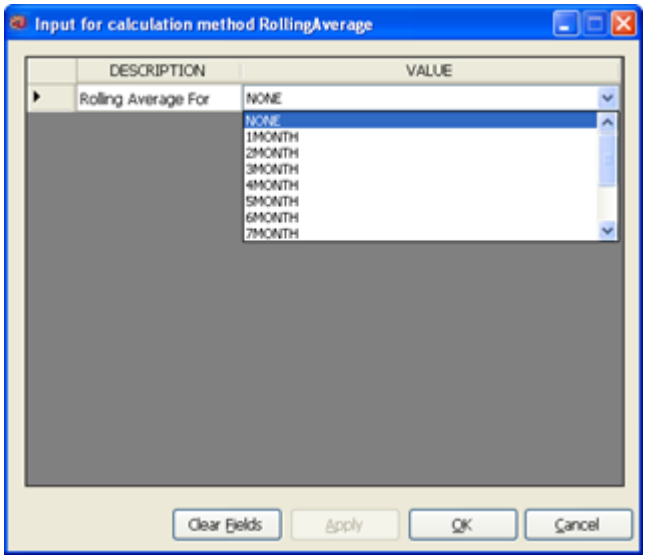
1. Copy custom DLL and add it into the "**SYSTEM**" folder of LABWORKS client i.e. **C:\LABWORKS\Client\System**].
2. Only replacing existing custom DLL with new one will not work. A little more is required. To make them available follow the steps below-

a. The new DMR calculations require three tables "**DMR_METHOD**", "**DMR_METHODDETAIL**" and "**DMR_METHODSUBDETAIL**". These tables contain details of each and every calculation i.e. "**STANDARD**" as well as "**CUSTOM**".
b. The new DMR calculations are dynamic, to understand it take some examples of custom calculation.

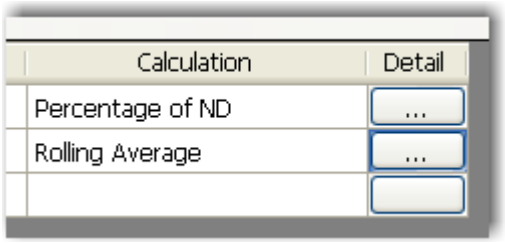
Example 1:- "**Rolling Average**"**:** Image below shows a Requirement form; "**Rolling Average**" custom calculation is set for "**MAXIMUM**" section.



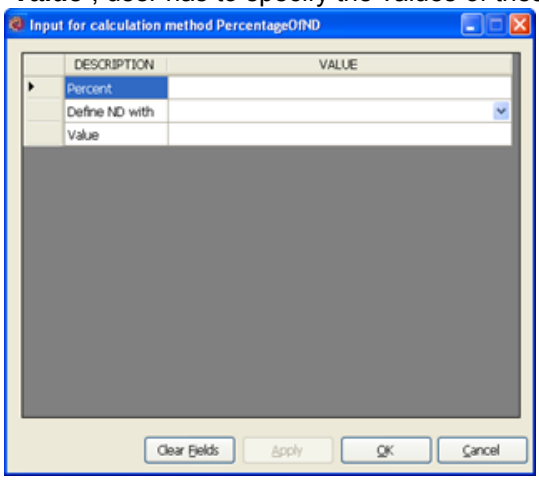
The "**Rolling Average**" calculation requires extra input such as **"Rolling average for n months".** User has to specify the value of "**n**". Refer image below-



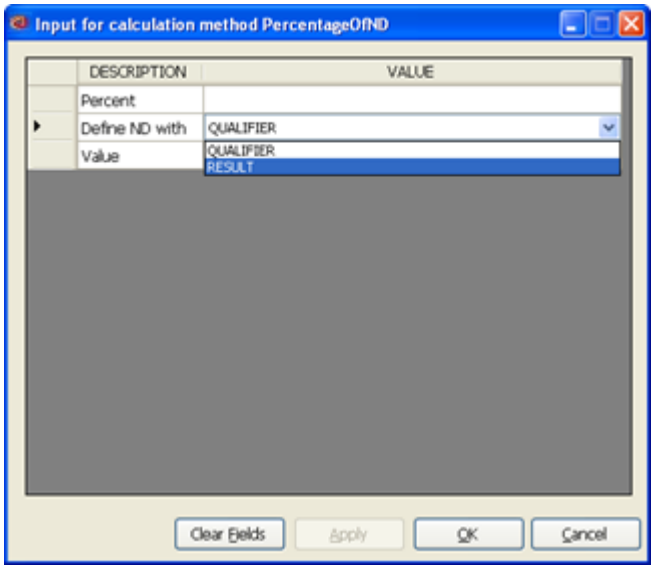
Example 2:- "**Percentage of ND**"**:** Image below shows a Requirement form; "**Percentage of ND**" custom calculation is set for "**AVERAGE**" section.



The "**Percentage of ND**" calculation requires extra input such as "**Percent**", "**Define ND with**" and "**Value**"; user has to specify the values of these fields. Refer image below -



The image below shows predefined values for "**Define ND with**"



The above examples demonstrate that both calculations require an extra input. Some of the extra inputs have predefined values. A single DLL cannot hold these values. To fulfill this requirement we have introduced three tables.

- **DMR_METHOD** – master table which lists the actual calculations
- **DMR_METHODDETAIL** – first level detail table for each calculation
- **DMR_METHODSUBDETAIL** – second level detail table for each calculation

Calculation Detail in New DB Tables

Image below shows calculations entry in "**DMR_METHOD**" table. This table contains information about calculations, not for extra input. Check selected entries for "**Rolling Average**" and "**Percentage of ND**" calculations. A "**HASPARAMETER**" column has "**-1**" value for both of these calculations, because they requires an extra input. The extra input information is saved in "**DMR_METHODDETAIL**" table.

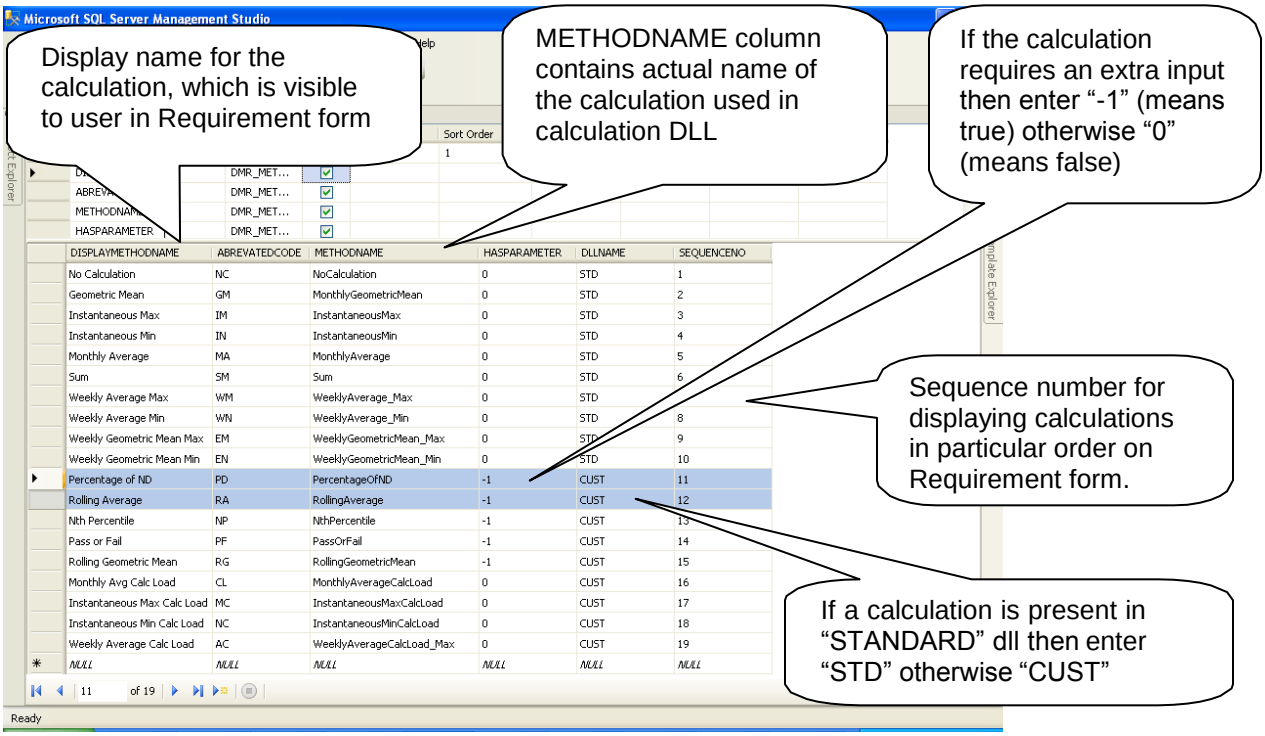
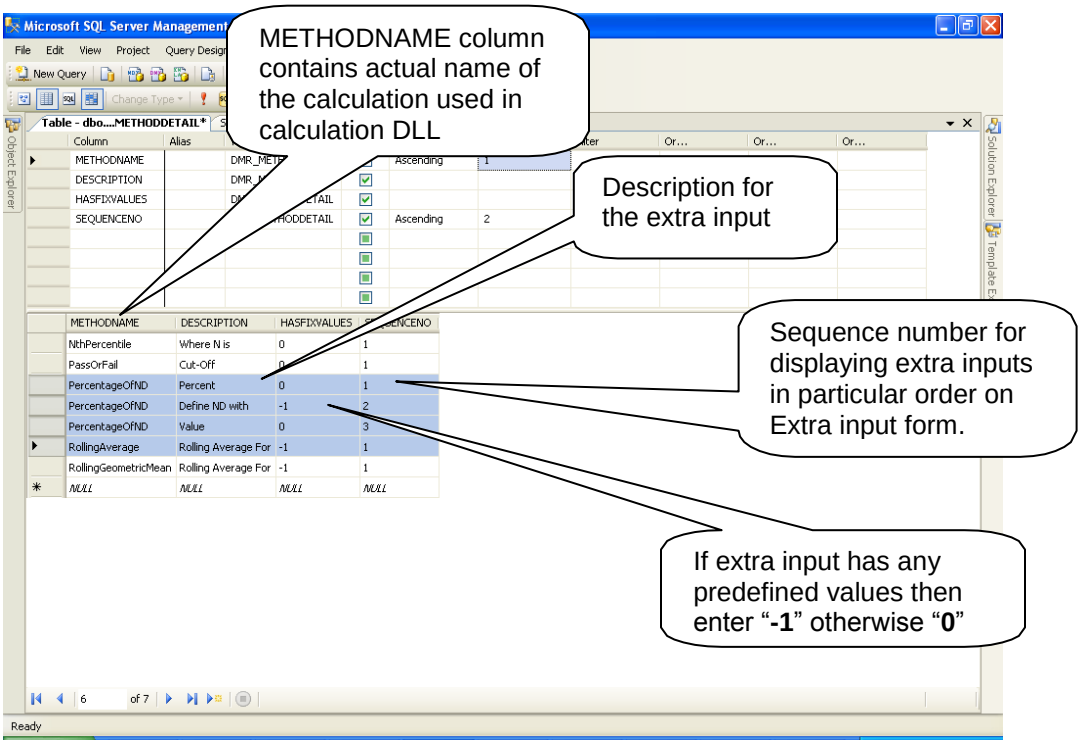
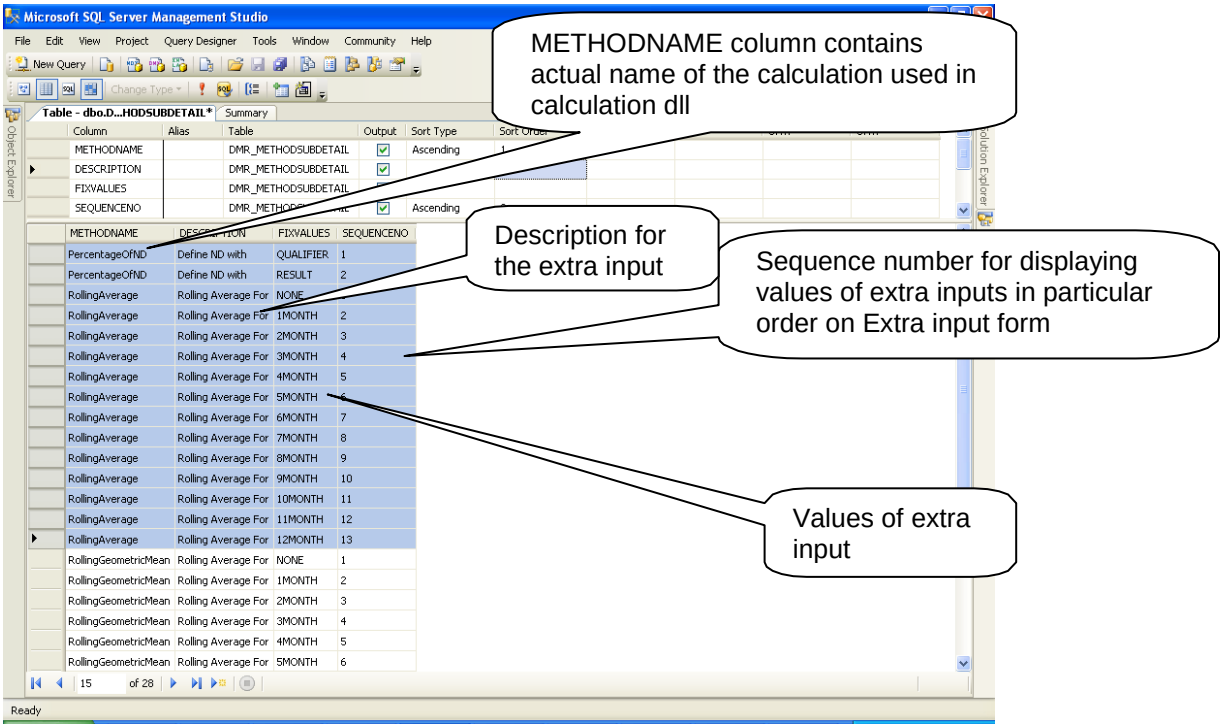


Image below display extra input information in “**DMR_METHODDETAIL**” table. “**Percentage of ND**” has three entries of extra input and “**Rolling Average**” one entry of extra input. The “**Define ND with**” of “**Percentage of ND**” and “**Rolling Average for**” of “**Rolling Average**” has “**-1**” entry in “**HASFIXVALUES**” column.



The actual entries of fix values are in "**DMR_METHODSUBDETAIL**" table. The image below shows fix value entries.



Selected are the fix values for the extra inputs of "**Percentage of ND**" and "**Rolling Average**".

Once the necessary entries for the calculations are done in the above mentioned tables, then calculations are visible in Requirement form.

Available Calculations in DLL Files

The calculation DLL file names must be as follows:

- The "**STANDARD**" calculation DLL must start with "**LWDMRSTD"**
- The "**CUSTOM**" calculation DLL must start with "**LWDMRCUSTOM"**

Within these DLL files reside the following list of calculations.

Calculation DLL File	No.	Calculation Name
LWDMRSTDCalculation.dll	1	MonthlyGeometricMean
	2	InstantaneousMax
	3	InstantaneousMin
	4	MonthlyAverage
	5	Sum
	6	WeeklyAverage_Max

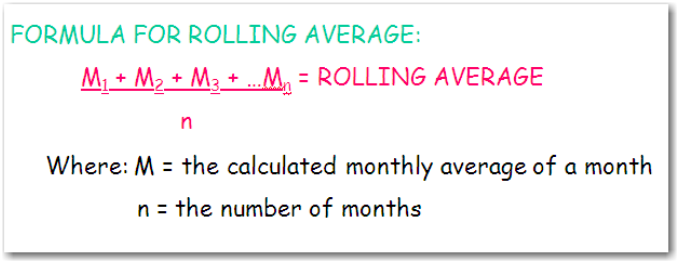
	7	WeeklyAverage_Min
	8	WeeklyGeometricMean_Max
	9	WeeklyGeometricMean_Min
LWDMRCustomCalculation.dll	1	PercentageOfND
	2	RollingAverage
	3	NthPercentile
	4	PassOrFail
	5	RollingGeometricMean
	6	MonthlyAverageCalcLoad
	7	InstantaneousMaxCalcLoad
	8	InstantaneousMinCalcLoad
	9	WeeklyAverageCalcLoad_Max

Custom Formula Usage (Details)

The following are formulas that were identified as “custom” since the majority of DMR users do not use them. Each of the following calculations are included in the custom DLL by default, and they are described below. The user can decide if they need these or not. If not, the System Admin can remove the DLL and associated records to keep the calculation list down to only the standard list.

Percentage of ND (**PercentageOfND**) – in LABWORKS, any analysis can have a ND (non-detected) value which means the sample result is below the required reportable value and is insignificant. The insignificant value is often reported as “ND” or “not detected” in LABWORKS. Occasionally, the governing state may want to know the percentage of these non-detects. For example, if 100 samples were reported for the month and 12 of them had results of ND, you would report a value of “12%” on the report. To report this, you’ll need to define what constitutes a non-detect result in LABWORKS. When selecting this, the user must be able to enter this value (“ND”, “NOT DETECTED”, etc.) that will define a non-detected result in LABWORKS. Each value with this result will be counted as a non-detected result that factors into the calculation.

Rolling Average (RollingAverage) – This calculation is really the same as a monthly average, but it also looks at previous month averages as part of the calculation.



Note: Calculation history is checked to perform this calculation, so performance will be slightly affected during final records processing when this calculation is selected. If n=4, then 4 months of previous history must be pulled into the formula where M1 is the current month.

Nth Percentile (NthPercentile) – A calculation where you report the result within a *designated percentile*. For example, if you are required to report the 90th percentile and there are 100 results for the month (in order from smallest to largest), you would report the 90th result value in the list of data from LABWORKS.

Pass or Fail (**PassOrFail**) – A Boolean result requirement. This calculation can either look at a specific text value (true or false, yes or no, 1 or 0), or a designated point to define whether or not a result passes or fails the requirement (for example, >30 is pass, <=30 fails). The requirement with this designated calculation will evaluate each result for the month for qualification as a “pass” or “fail” value. These

values are totaled for the month to determine the final pass or fail value based on the cut-off value. For example, for 100 samples within the reporting period, if 70 of these results were “pass” and 30 of them were “fail”, and the cut-off value was 40 (>40 passes and <=40 fails), then you would report “PASS” on the report because 70 > 40. To report this, you’ll need to define what this cut-off value is within the DMR application. This is important because it is not always 50/50. A requirement that passes 60% of the time, for example, may still fail in the eyes of the government. The user has the ability to define how strict the requirement is (the higher the cut-off value, the more strict the requirement). The cut-off value defines the percentage of failure or what percentage this requirement must overcome to pass. When the user selects this calculation, the UI (in the requirement screen) request this cut-off value. This is required for the calculation to work properly.

Rolling Geometric Mean (RollingGeometricMean) – This calculation is really the same as a geometric mean, but it also looks at previous month’s geo means as part of the calculation. See the **Rolling Average** calculation above for more detail.

Monthly Average Calc Load (MonthlyAverageCalcLoad) – This calculation is the same as the corresponding Monthly Average calculation except the CONTROL value (FLOW) is factored. During the calculate final records process, this “Loading” field is calculated for each requirement. Loading is calculated differently depending on how the designated CONTROL record is entered. The “Loading” is used in the Monthly Average CalcLoad calculation instead of the “Result” which is used in the regular Monthly Average calculation.

Weekly Average CalcLoad (WeeklyAverageCalcLoad_Max) – This is the same as the Weekly Average calculation but (just as above) the calculated “Loading” fields are used instead of the raw result from LABWORKS (see above for details).

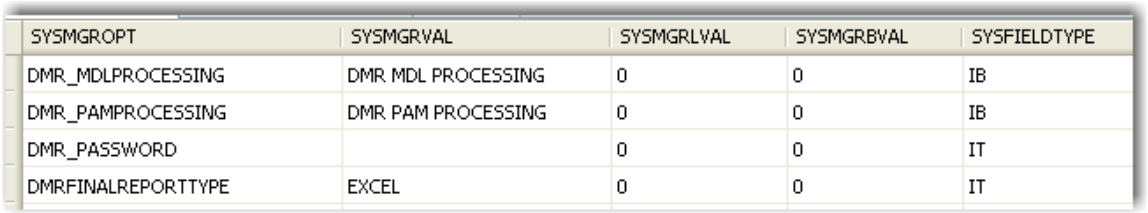
Instantaneous Max/Min CalcLoad (InstantaneousMinCalcLoad and **InstantaneousMaxCalcLoad)** – This is the same as the Instantaneous Min and Max calculations but (just as above) the calculated “Loading” fields are used instead of the raw result from LABWORKS (see above for details).

DMR Configuration Settings

SysMgr Keys

No support of the old legacy LABWORKS.INI file exists within the new application. The new DischargeMonitor application uses SysMgr keys exclusively. However, the majority of system settings are configured under the options or control screens within the application itself. That is documented within the Discharge Monitor User Reference chapter.

There are basically only 4 SysMgr keys supported within the new application.



SYSMGROPT	SYSMGRVAL	SYSMGRLVAL	SYSMGRBVAL	SYSFIELDTYPE
DMR_MDLPROCESSING	DMR MDL PROCESSING	0	0	IB
DMR_PAMPROCESSING	DMR PAM PROCESSING	0	0	IB
DMR_PASSWORD		0	0	IT
DMRFINALREPORTTYPE	EXCEL	0	0	IT

These include:

- DMR_MDLPROCESSING – a Boolean value that when set to false (0) will not factor in standard LABWORKS MDL processing which will add a “<” character into the result (and into the report) in front of the MDL value if the result from LABWORKS is less than the MDL**. Keep in mind that if you are already entering the results into LABWORKS like this (<4.0, for example), then make sure and turn this off. If it is on, and it encounters a result character instead of a numeric value, the DMR system will assume it is a zero.**
- DMR_PAMPROCESSING – a Boolean value that when set to false (0) will not factor in PAM, Project Analyte Management module, values. PAM is a separate module and set of database tables that is designed to evaluate results based on reporting limits. If you are already using PAM and evaluating the results prior to DMR, then this should be turned off.
- DMR_PASSWORD – a system password that is required before a DMR user can change any system settings within DMR.
- DMR_FINALREPORTTYPE – defines the default report type (EXCEL, CRYSTAL, or EDMR).

Common DMR Problems

As part of the final section of this document, we wanted to provide some support assistance by listing the most common errors on DMR final reports turned into governing agencies. These can be used as a means to catch problems before they occur.

Issue #1: Incorrect Weekly Averages – people reporting or setting up their weeks incorrectly

Incorrect Weekly Average - Partial weeks at the end of the month carry over to the next month for WEEKLY AVERAGE reporting purposes ONLY. Remember - for DMR reporting purposes, a calendar week runs from Sunday through Saturday) *(Tip – At the beginning of each year make note of which weeks are carried over on your calendar – see attachment)*

Issue #2: Exceptions are not tallied and reported properly (actually a rare problem, but this does happen)

Incorrect reporting of violations (a.k.a. exceedances) - Report all violations in the No. of Exceedances box. Some people have reported just one violation when they have multiple violations. *(See Chapter 1, Page 2 of the 2005 DMR Guidelines. Tip – Make a list of all violations [daily, weekly and monthly] on a separate piece of paper such as a cover letter, and use to correctly complete the No. of Ex boxes)*

Issue #3: Loadings (for Weekly Average) not calculated properly – this is a good example of the CONTROL record not being used properly or the wrong formula/calculation is selected.

Average weekly loading averages figured incorrectly. Each day's loading must be calculated separately - then averaged. Do not use the average flow and concentration for calculating the weekly average. (*See Chapter 3 Page 2 the 2005 DMR Guidelines for guidance*)

Issue #4: Loadings (for Monthly Average) not calculated properly – so we can see the complexity of factoring in the loading calculation is a common source of problems.

Average monthly loading averages figured incorrectly. Each day's loading must be calculated separately – then averaged. Do not use the average flow and concentration for calculating the monthly average. (*See Chapter 3 Page 2 the 2005 DMR Guidelines for guidance*)

Issue #5: Not filling out all required fields – don't forget to fill all fields.

Not completing all reporting boxes - All boxes must be completed. Failure to complete all sample measurement boxes, without providing an explanation on the DMR or on a cover letter attached to the DMR, will result in EPA considering the omission a non-reporting violation. *(See Chapter 1 Page 4 the 2005 DMR Guidelines for guidance. Tip - take last month's DMR and compare with DMR you just completed to see if you missed anything/ also have someone else review the DMR for omissions)*

Issue #6: Calculation of Monthly Average for FLOW. A correct week designation within the reporting period and correct results entered in LABWORKS should prevent this.

Incorrect daily flow averages - If you did not discharge every day of the month, divide total monthly flow by the number of days you discharged, not the number of days in the month. (*See Chapter 3 Page 2 the 2005 DMR Guidelines for guidance*)

Issue #7: No explanation for violations – most government agencies require an explanation at the bottom of the DMR report when a violation (exception) is entered on the report. Sometimes customers forget to do this, so a possible future enhancement would be to highlight a violation box on the Excel template when any exception is found.

No explanation for violation in comments box at bottom of DMR or in a cover letter - An explanation must be provided for all violations *(Provide an explanation for the violation with the violation list you made for reporting the number of violations. See Chapter 1 Page 6 of the 2005 DMR Guidelines)*

Issue #8: Geo Mean is not correct

Incorrect Geometric Mean – All samples <1 are considered as zero. If any sample is zero, "1" must be added to all samples before calculation, and one subtracted from the final answer. *(Tip – Use DES help sheet. See Chapter 1 Page 4&5 and Chapter 3 Page 4&5 of the 2005 DMR Guidelines*)

Other problems reported are late reports being filed or reports turned in unsigned, etc. which have nothing to do with the DMR application.

As your lab begins using the new DMR application, *DischargeMonitor*, we recommend meeting with the key lab supervisors, environmental managers or engineers responsible for DMR report production to discuss these commonly reported issues. If they understand the problems often caught at the agency level, they can identify procedural methods to prevent them before they occur. About 85% of reported problems by our software support department are specifically due to user mistake or configuration. Hopefully you, as the DMR System Administrator, will be able to use this documentation to identify non-system problems before both lab and IT resources are tied up researching issues that may not actually be problems.

7. Discharge Monitor User Reference

Labworks-Discharge Monitor

Introduction

The LABWORKS Discharge Monitoring module is a replacement for the old DMR module for tracking/reporting on discharges into water systems that may ultimately contribute to pollution.

Water pollution causes degradation of surface waters making them harmful for drinking, swimming, fishing and other activities. In the United States, authorized by the Clean Water Act, the National Pollutant Discharge Elimination System (NPDES) permit program controls water pollution by regulating point sources that discharge pollutants into public bodies of water. Point sources usually are pipes or man-made ditches.

Industrial, Municipal, and other facilities must get the NPDES permits if their discharges go directly to surface waters. Reports regarding the Water Quality have to submitted to the NPDES by the agencies which hold these permits.

A need to automatically generate the required monthly NPDES report lead to the development of the DMR - **D**ischarge **M**onitoring **R**eport module.

The LABWORKS-DMR module allows a user to record all the details regarding the Permit, Discharge, Requirements, and Calculations done on the samples and print the form or submit the form online to NPDES.

Overview of Labworks-Discharge Monitor

Hardware and Software Requirements

Hardware Requirements:

For stand-alone system:

- Processor should be Pentium or it's comparable.
- Minimum 32 MB RAM.
- At least 100 MB Free Disk Space.
- CD-ROM.
- Printer.

For LAN network:

- Computers along with addition of LAN cards and an Ethernet cable.

Features of Labworks-DMR

- Single click access to the most common functionalities.
- Minimal keyboard or mouse operations required for entries.
- Pre-populated Speciality specific information.

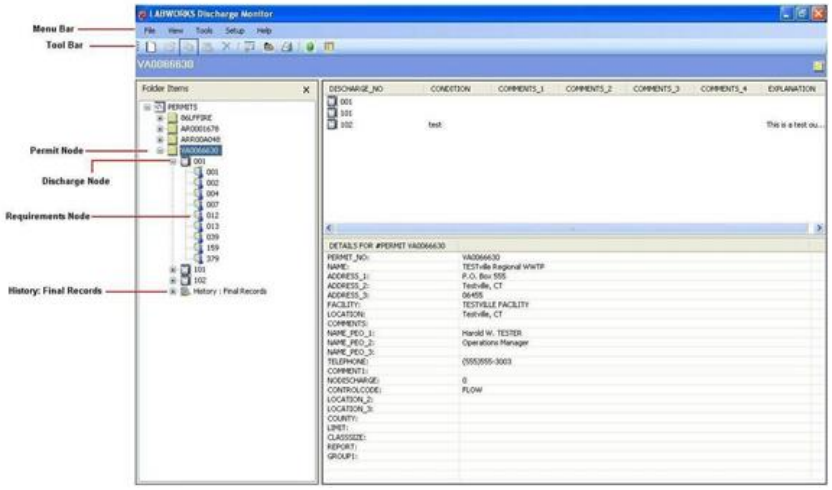
- Easy to use Graphical User Interface.
- Users, Roles and Workflow management.
- Effective Search Facility to Search any type of Information

Getting Started

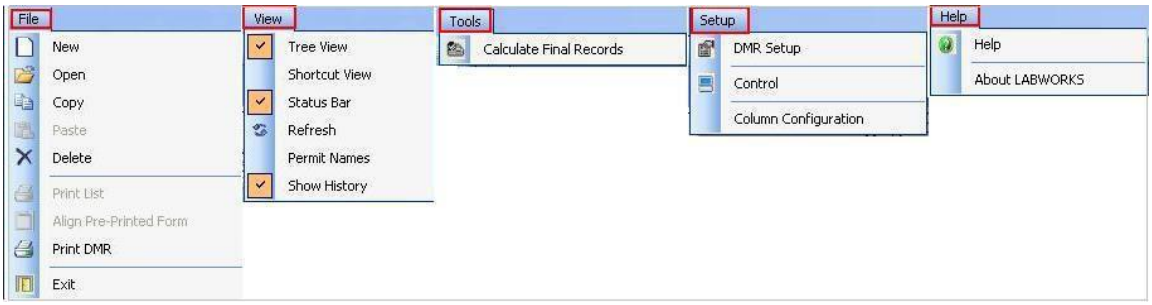
Quick Start

This section describes how to start the LABWORKS-Discharge Monitor Application.

1. **Double click** the  icon installed on the desktop.
2. It will open the **Login** screen.
3. Select the database from the **Available Database** drop down list.
4. If only one database is in use, then the **Available Database** list box is not available for selection.
5. Enter the user name in the **User Name** text field.
6. Enter the password in the **Password** text field.
7. Click the **OK** button.
8. The **LABWORKS Discharge Monitor** page will open.



The buttons defined in the old main screen are replaced by toolbar buttons and/or menu options at the top of the form

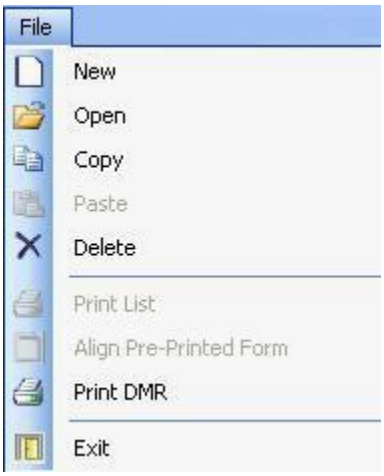


Menu Options

The Menu Options available in the application are described below.

File Menu Item

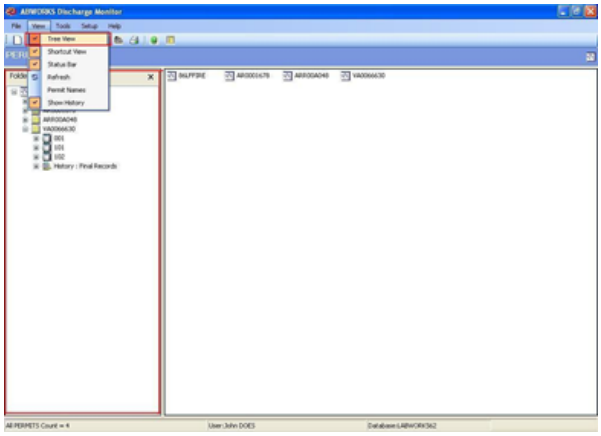
Following is the information about various File Menu items.



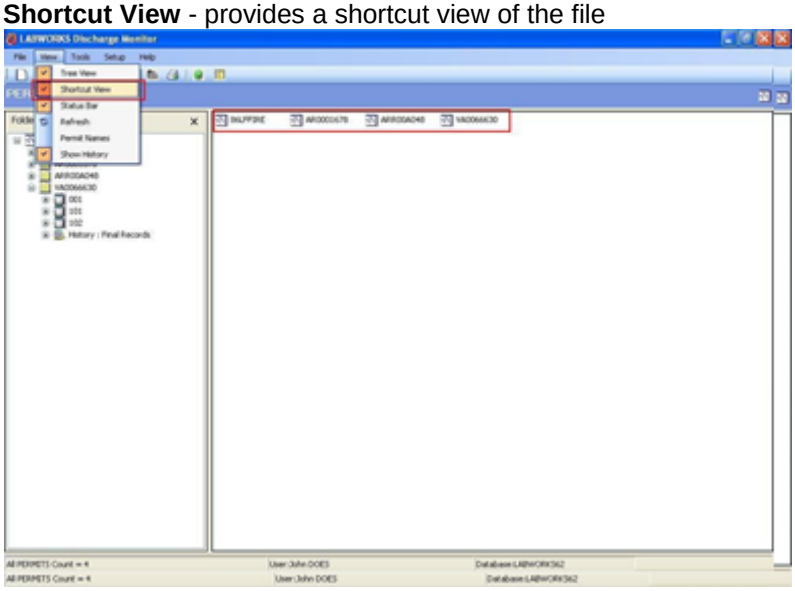
1. Click **New** to open a new file.
2. Click **Open** to open a file.
3. Click **Copy** to copy the file.
4. Click **Paste** to paste a file from another location.
5. Click **Delete** to delete the file.
6. Click **Print** List to print a list.
7. Click **Align Pre-Printed Form** to align the form before printing.
8. Click **Print DMR** to print the form.
9. Click **Exit** to close the application.

View Menu Item

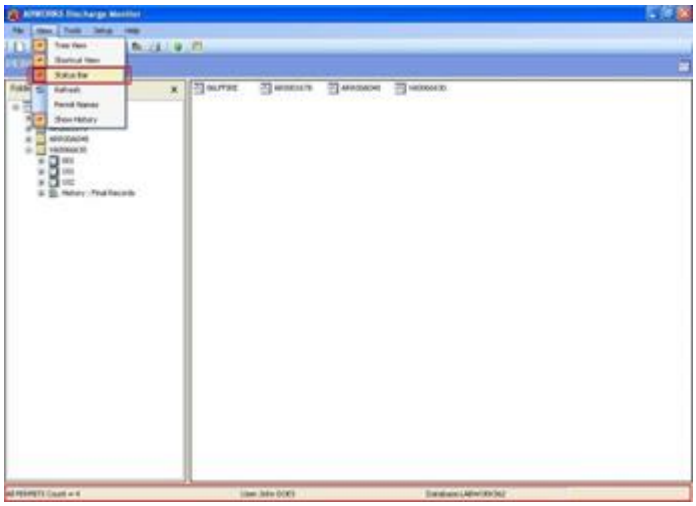
Tree View - provides a tree view of the file



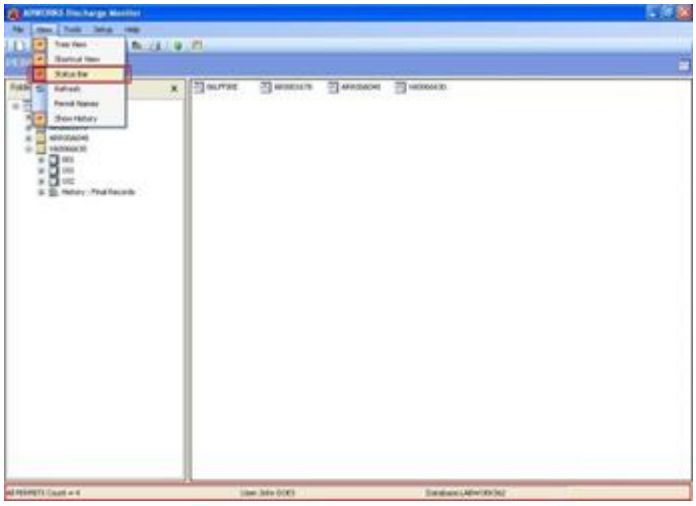
Shortcut View - provides a shortcut view of the file



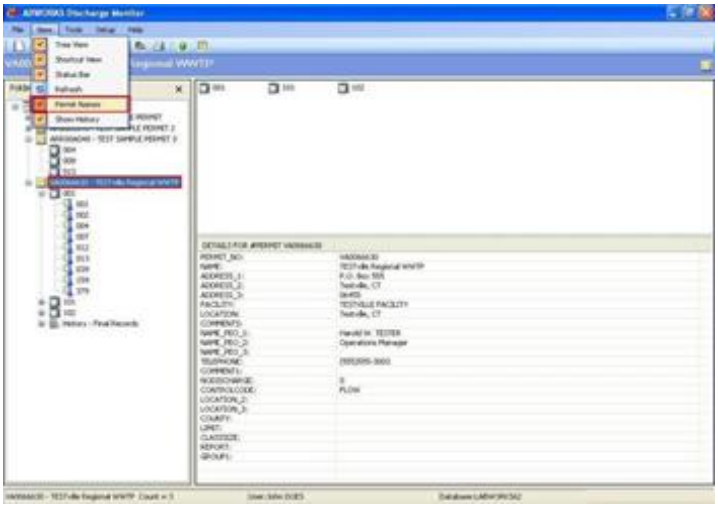
Status Bar- shows the Status bar at the bottom of the screen



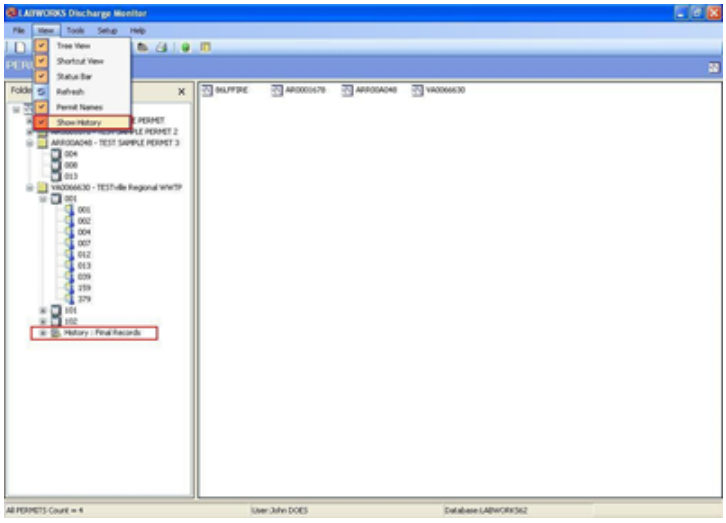
Refresh- will refresh the information



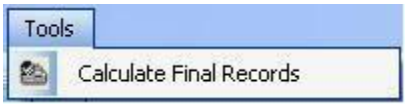
Permit Names-will show the Permit names



Show History-will show the history of the file

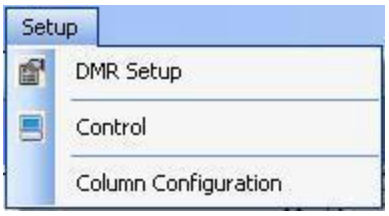


Tools menu item



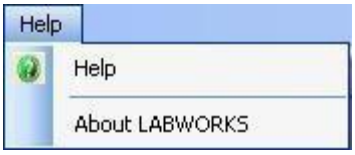
Calculate Final Records- clicking this item will open up a Calculate Final Records form.

Setup Menu item



1. DMR Setup- Clicking this item will open up a **DMR Setup** form.
2. Control- Clicking this item will open up a **Control File Maintenance** form.
3. Column Configuration- clicking this item will open up a **ListView column configuration** form.

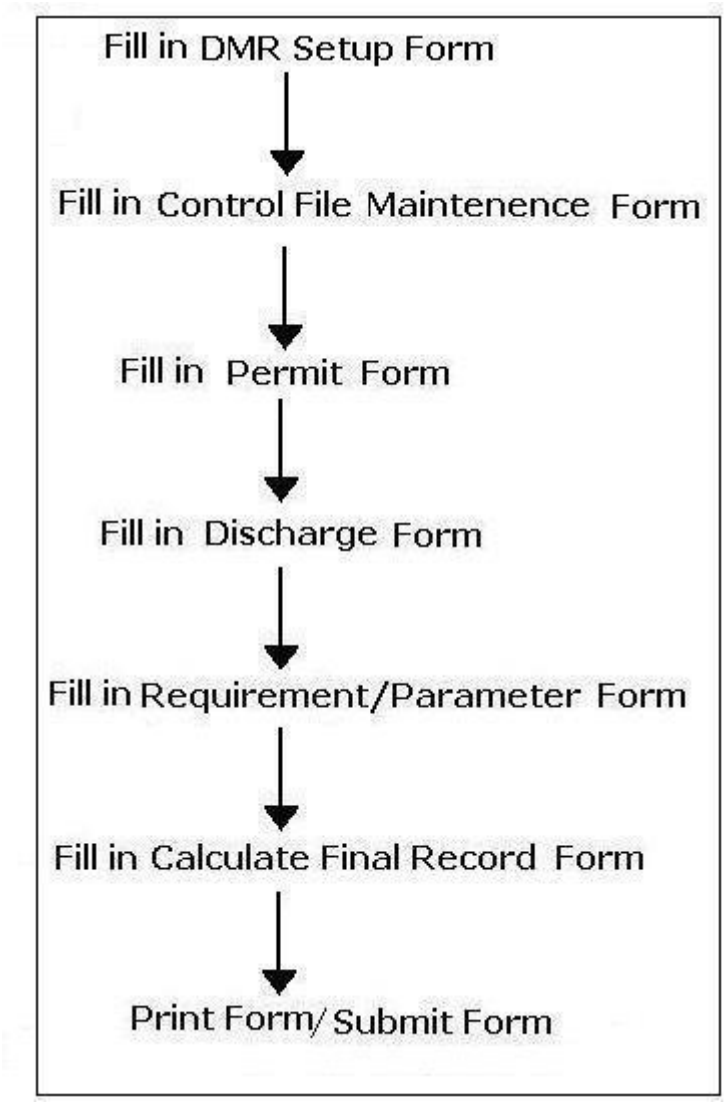
Help Menu Item



1. **Help** - Clicking this item will open up the **Help** menu
2. **About LABWORKS** - Clicking this item will open up the About Box with information about Labworks.

Workflow

This section describes the workflow of LABWORKS- Discharge Monitor module.



The image above shows the general workflow of the LABWORKS-DMR Module.

1. The User first fills up the **DMR Setup Form** which sets up the date, week details for report submission.
2. The second step is filling up the **Control File Maintenance Form** which will set up the controls.
3. The third step is filling in the **Permit Maintenance Form** which will set up the permit details.

4. The fourth step is filling in the **Discharge Form**/s related to the permit.

5. The fifth step is filling in the **Requirement/Parameter Form** to fill in the details of the parameters to be tested in the samples.

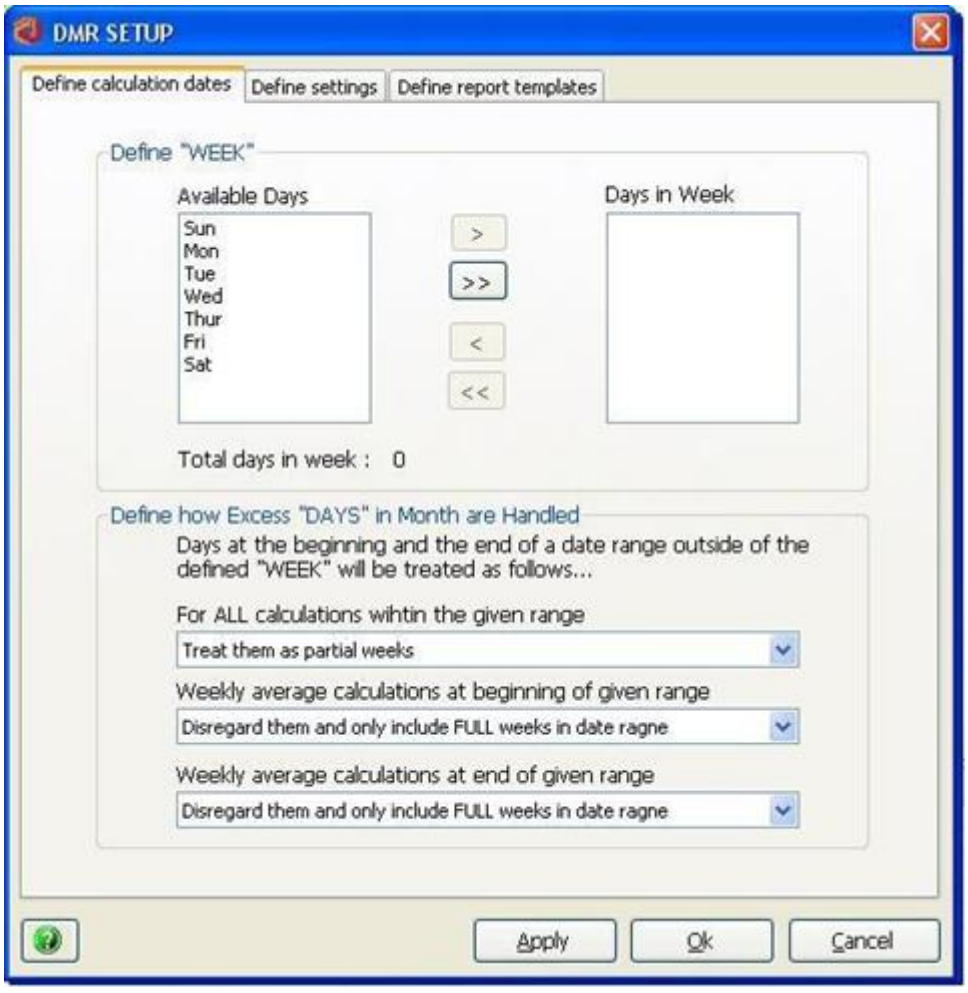
6. The sixth step is filling in the **Calculate Final Record Form** which will add details of the calculations done on the samples.

7. The final step is Printing or submitting the form to NPDES.

DMR Setup form

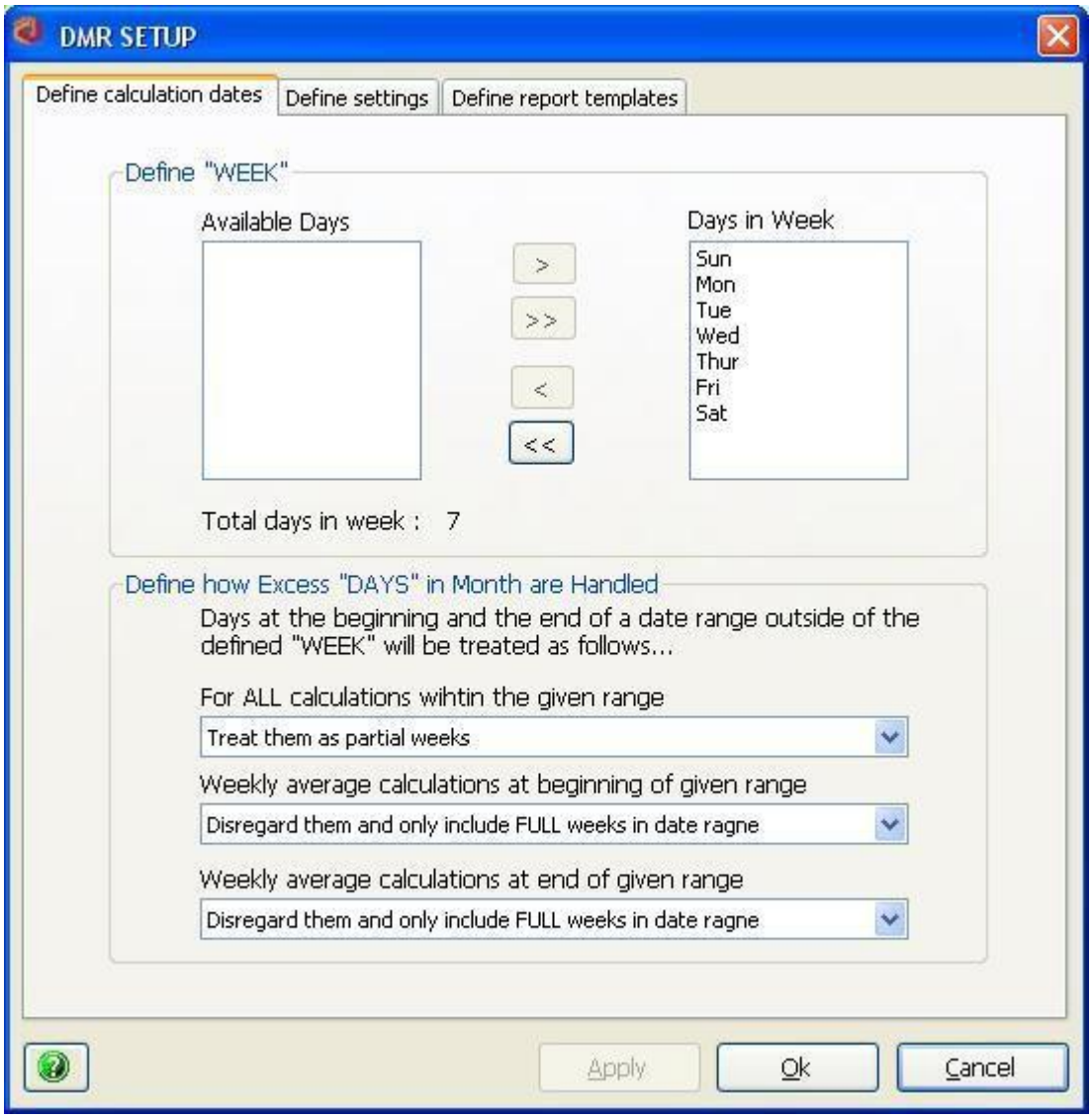
Filling in the DMR Setup form which is the first step in the LABWORKS-Discharge Monitor Module is described below.

Through this form the user can set the days of the week when the tests have to be done and also set up the week schedule for the reports.



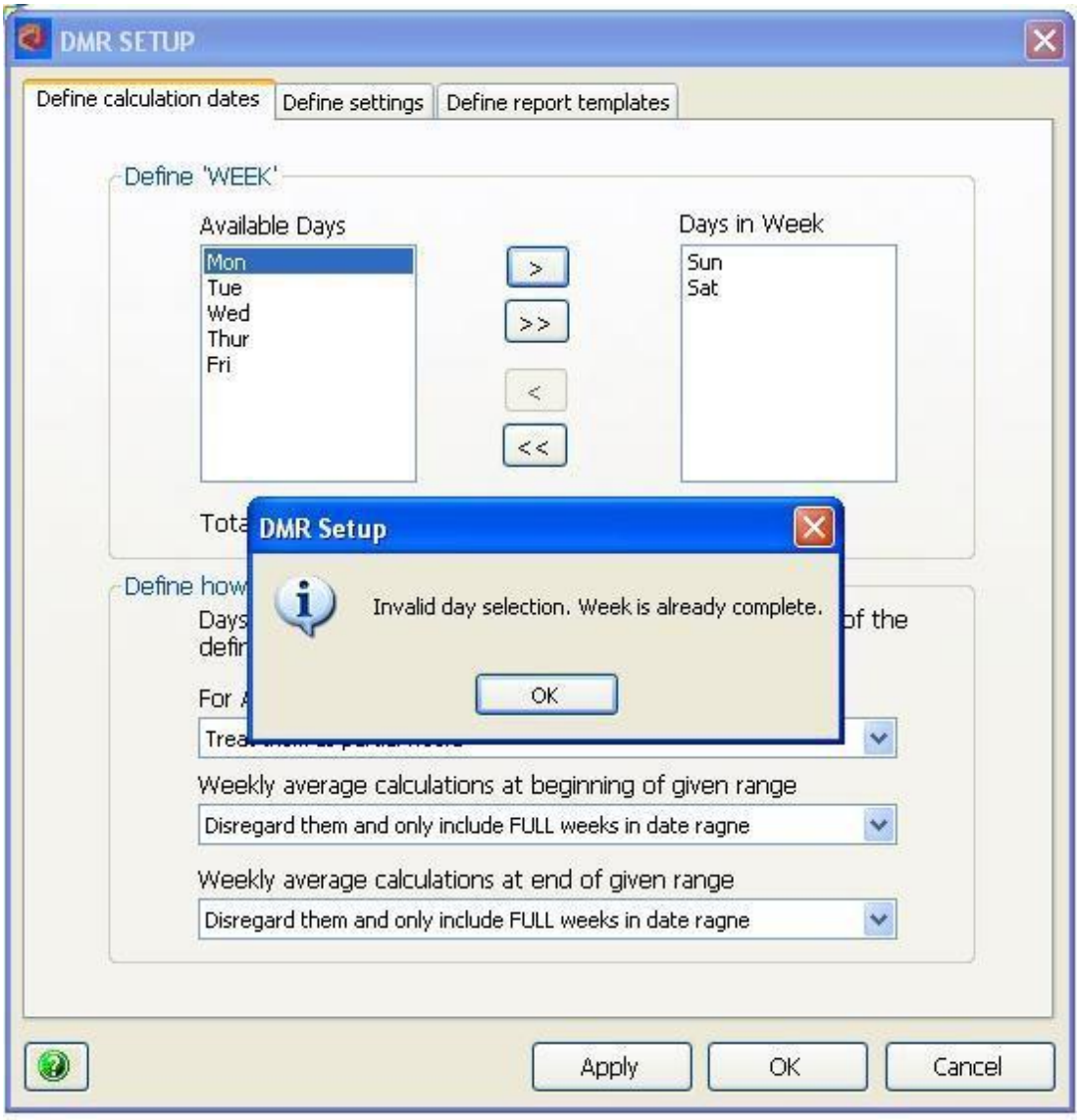
Define Calculation Dates tab

1. On the Menu Bar Go to Setup**>DMR Setup.**
2. The DMR Setup form will open.
3. Click the first tab **Define calculation dates**.

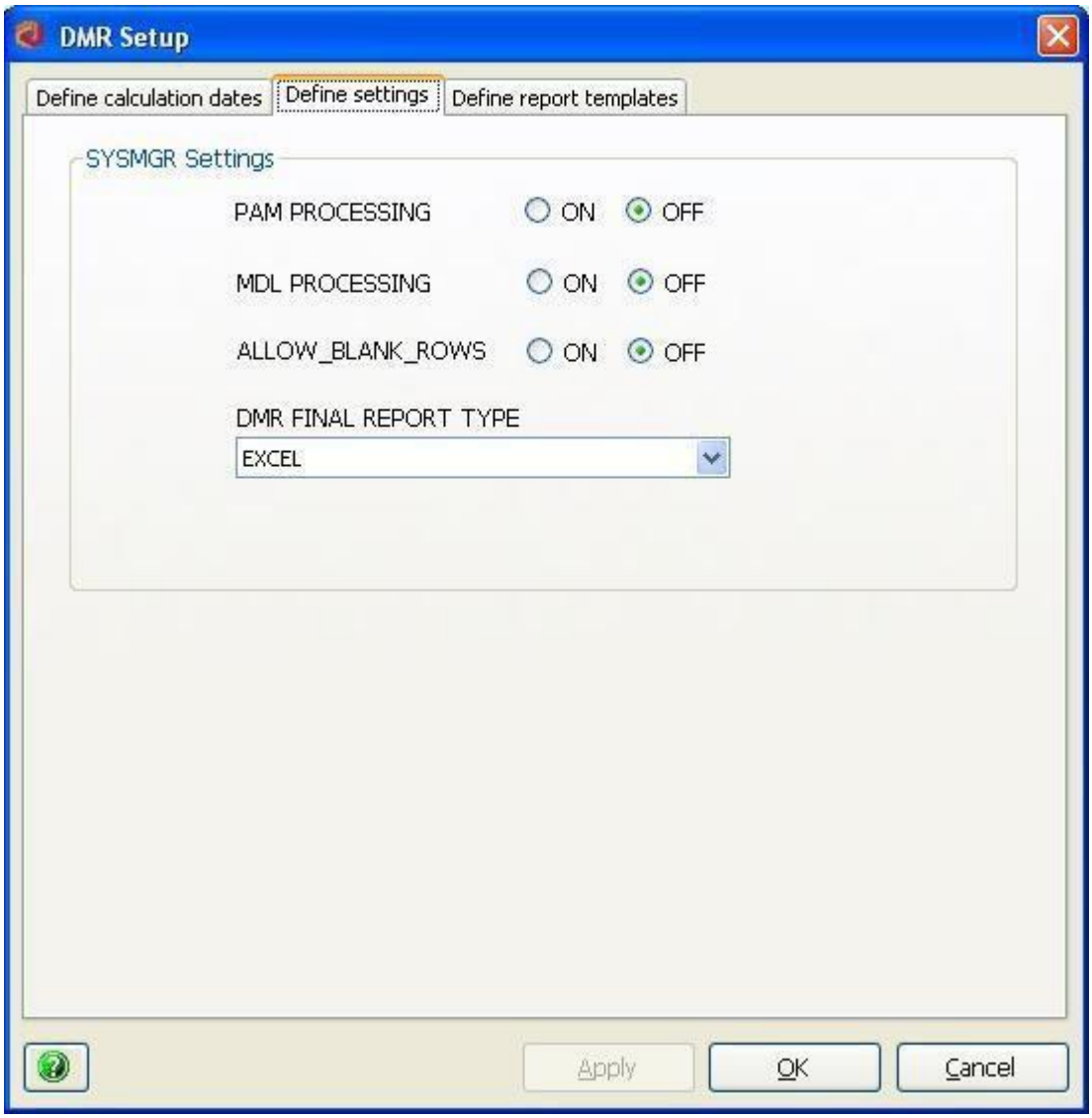


4. Select the days from the **Available Days** list.
5. Click the 

button to add the days in the **Days in Week** box.
6. The days will be added to the Days in Week box.
7. If the week is already over, then a message will appear as shown in the following example.



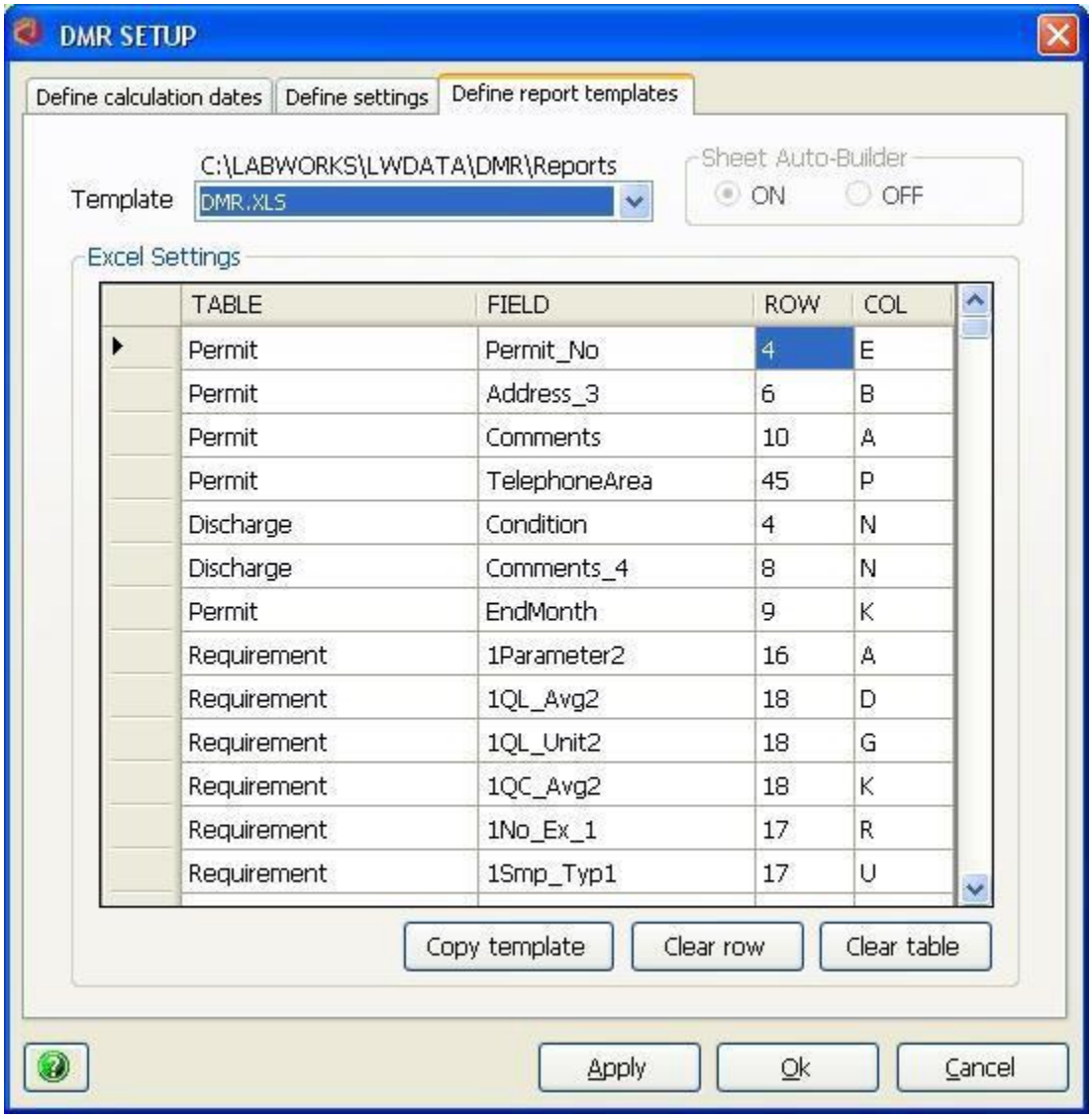
Define Settings tab



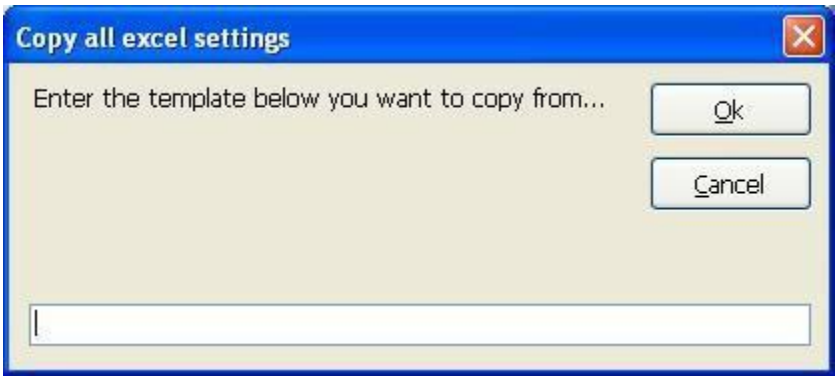
1. From the **PAM Processing** select the **ON** or **OFF** radio button. If this is **ON,** then **the** MDL value of **the** sample is replaced by the reporting limit of the sample.
2. From the **MDL Processing** select the **ON** or **OFF** radio button. If this is **ON**, the Result of the sample gets replaced by its MDL value, only when the result is less than MDL value.
3. From the **Allow_Blank_Rows** select the **ON** or **OFF** radio button.
4. Select an option from the **DMR Final Report Type** drop down list.
5. Click the **Apply** button to save the changes.

6. Click the **OK** button to close the form.

Define Report Templates tab



1. Select an option from the Templates drop down list.
2. Details of the sheet will be shown in the Excel Settings box.
3. Click the **Copy Template** button to select a template to copy.
4. It will open a new dialog box as shown in the following example.



1. Type an option in the text filed and click the **OK** button to copy the template.
2. Click **Cancel** to go to the previous screen.
3. Click the **Clear row** button on Figure 5 to clear a row.
4. Click the **Clear table** button on Figure 5 to clear the table.
5. Click the **Apply** button to save the changes.
6. Click the **OK** button the close the form.
7. Click **Cancel** to go to the previous screen.
8. Click the  button to get related **Help** information.

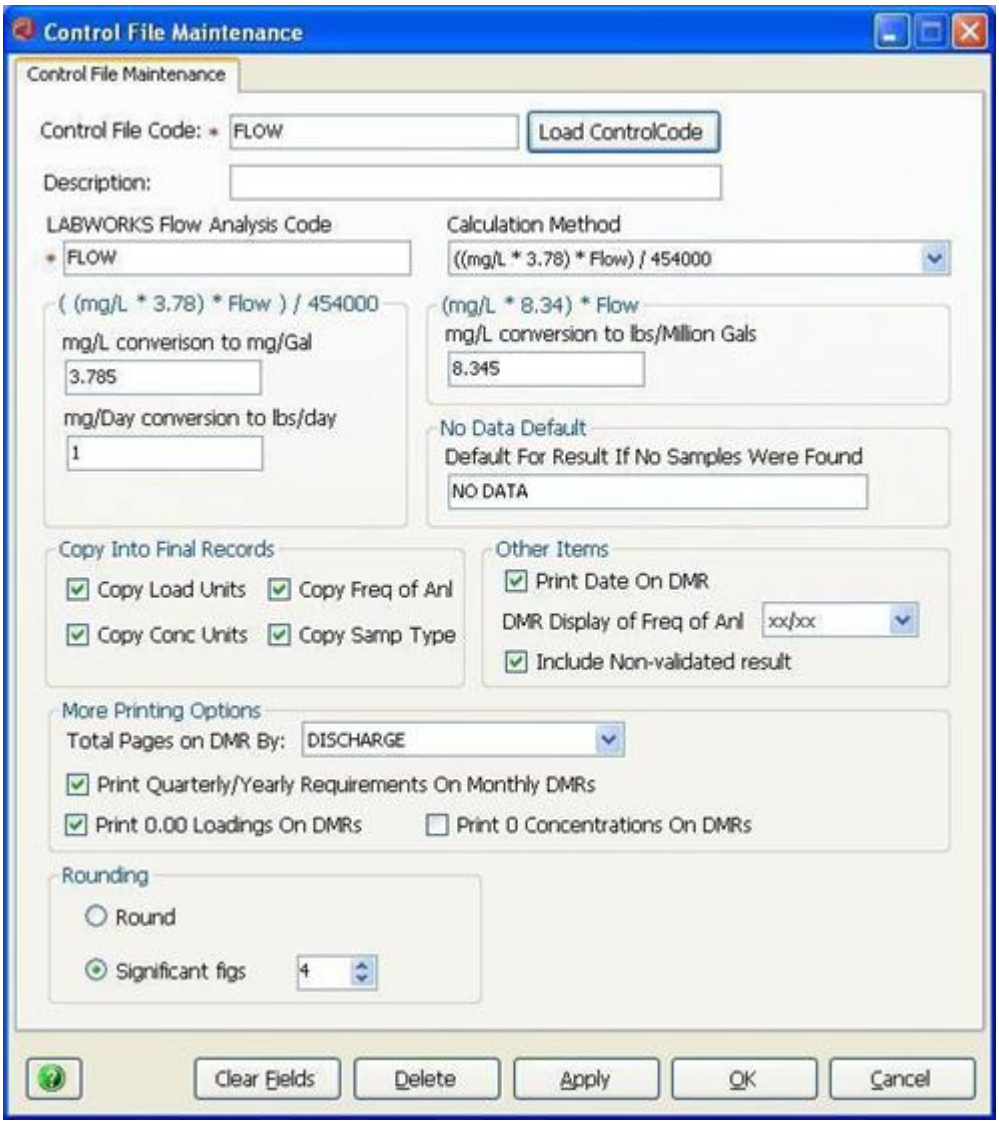
Control file maintenance form

This Control table contains global information for a permit. Multiple Control records can be created. Control entries can then be related to the Permit records.

This section describes how to fill in the control file maintenance form.

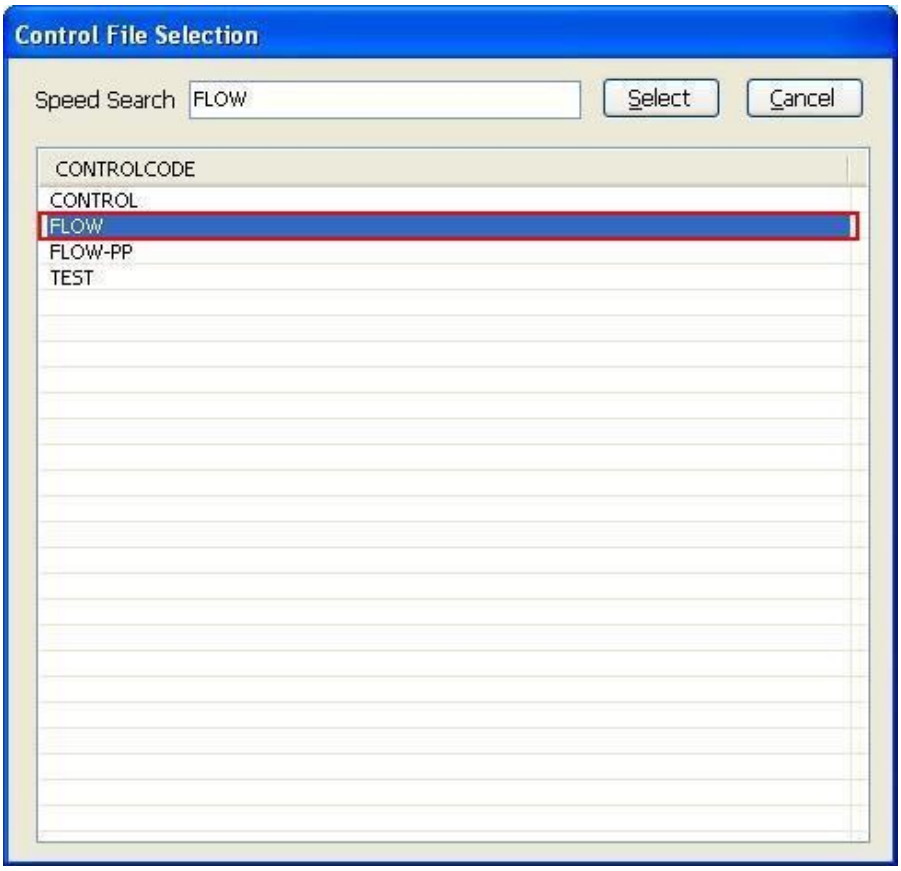
1. On the Menu Bar Go to Setup**>Control.**

2. It will open a **Control File Maintenance** form shown in the following example.



3. Click the **Load ControlCode** button to enter the code.

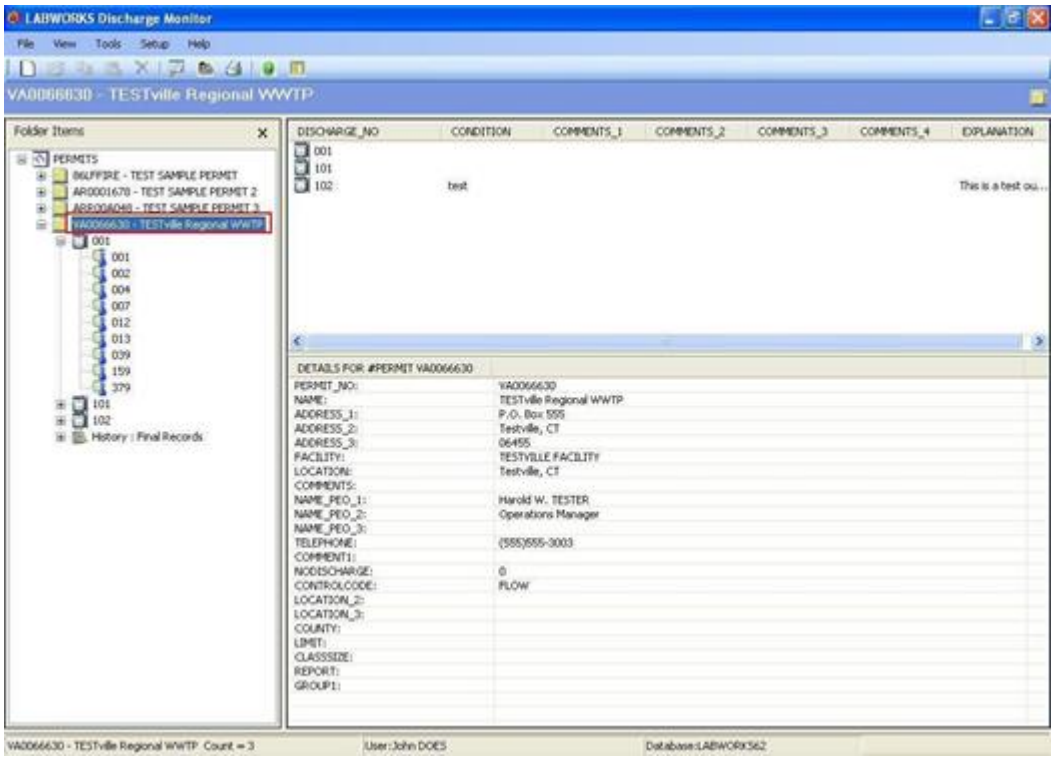
4. It will open up a new dialog box as shown here.



5. Select the option from the drop down (Outlined in Red above).
6. Click the **Select** button.
7. The information will be updated on the **Control File maintenance** form.
8. Fill in other details in the **Control File Maintenance** Form.
9. Click the **Apply** button to save the changes.
10. Click the **Delete** button to delete the information.
11. Click the **OK** button to close the form.
12. Click the **Cancel** button to go to the previous screen.
13. Click the **Clear Fields** button to clear all fields.

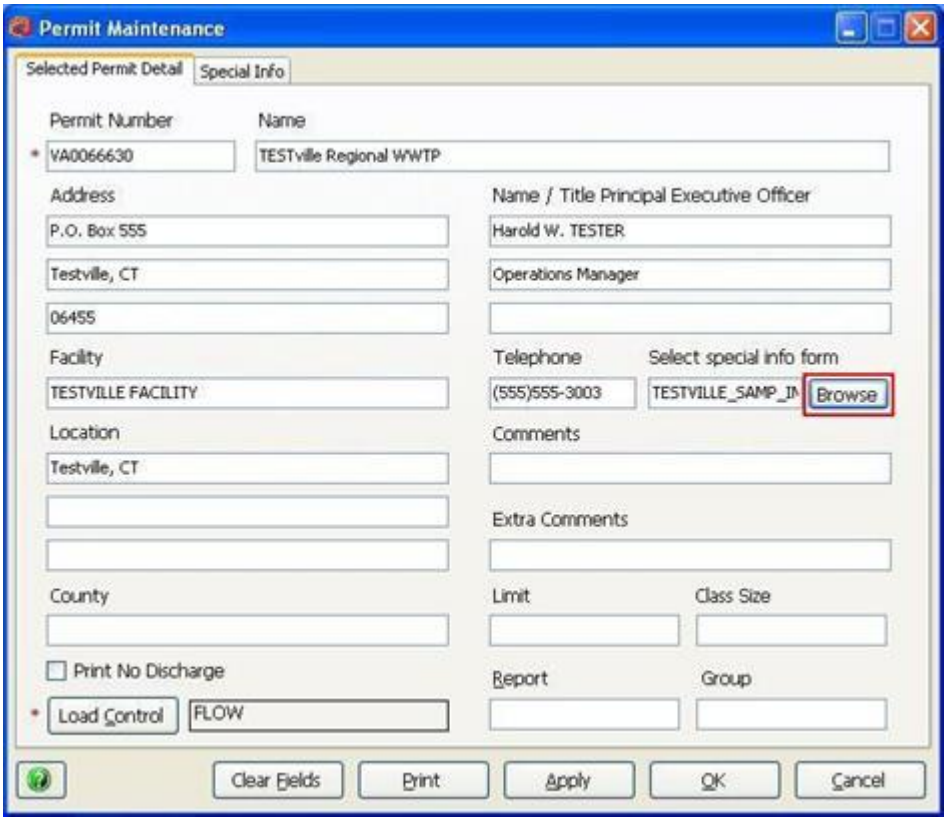
Permits

This section describes how to fill in the Permit details.



1. The screen shows **Permits** as the Main node on the left hand side.
2. The right hand side shows details of the selected item in the Tree view.
3. Double click the **Permit** Node to open the permit form (Outlined in Red).

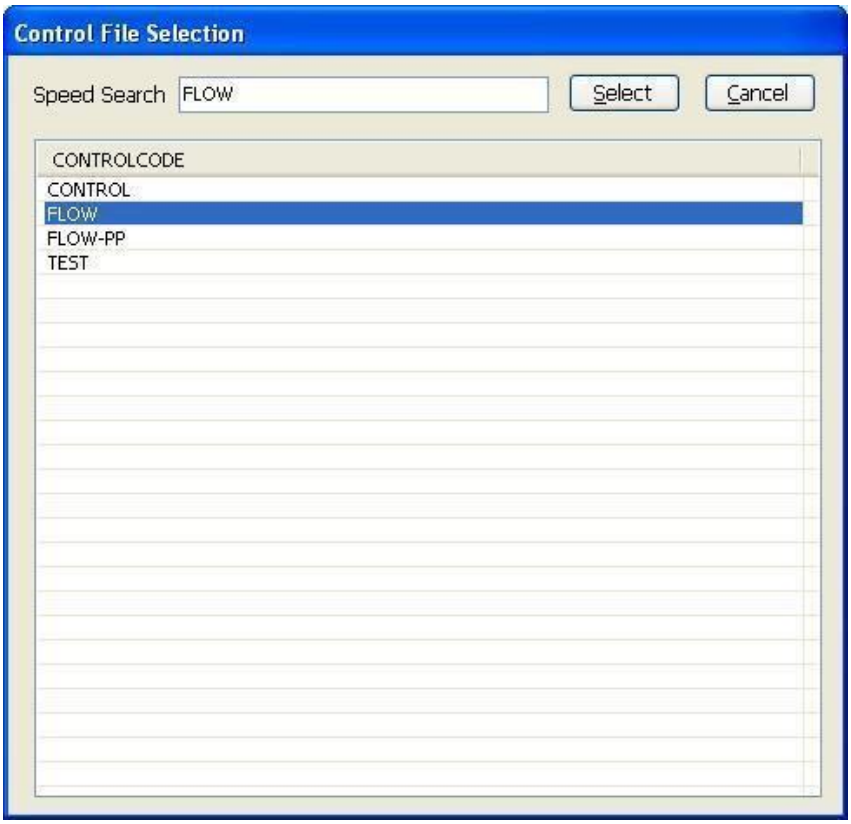
It displays the Permit Maintenance form as shown below.



Selected Permit Detail tab:

1. Fill in all the respective text fields.
2. Click the 

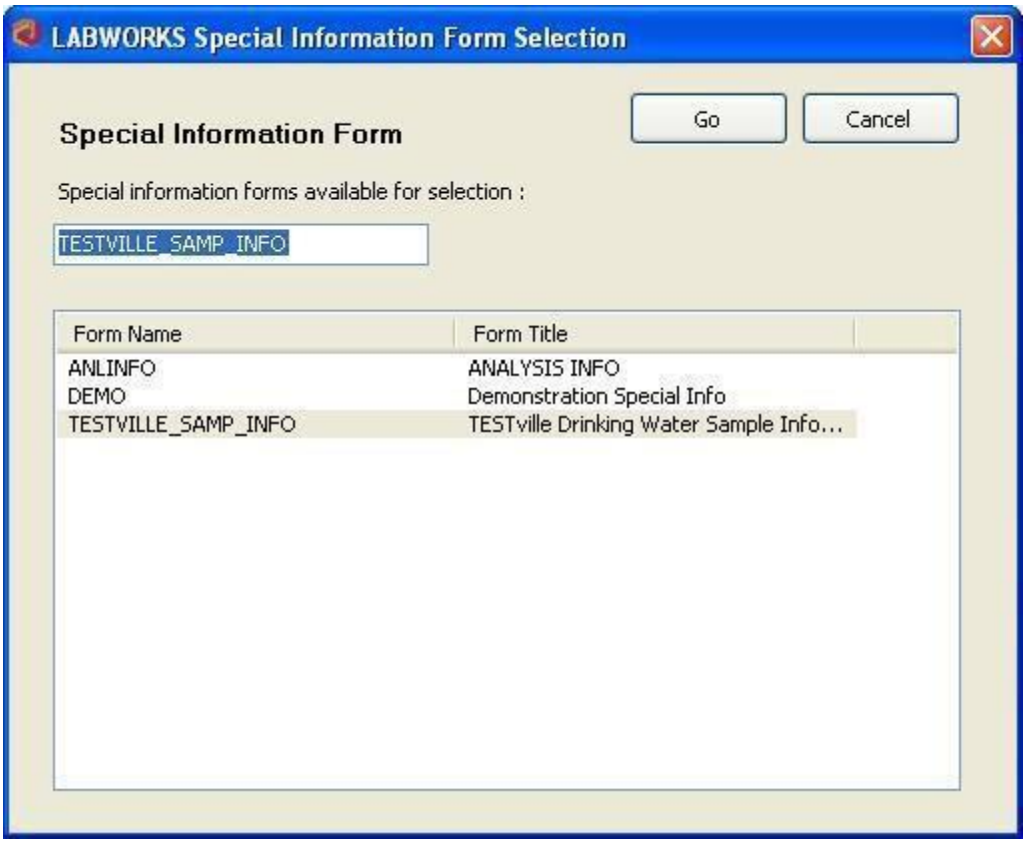
button.
3. A new **Control File Selection** form will open as shown in the following example.



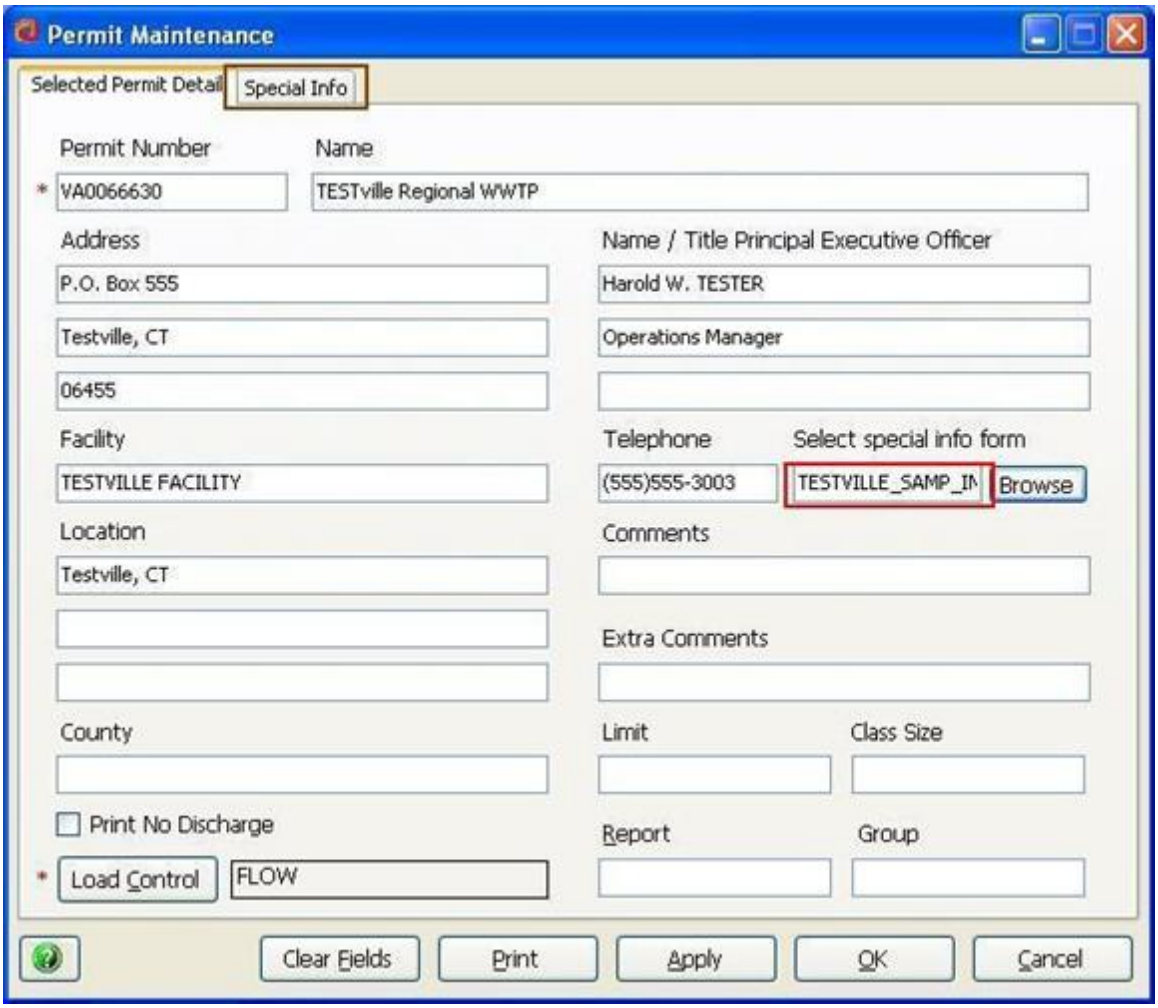
1. Select an option from this form.
2. Click the **Select** button shown above. The form will close.
3. The selected option will be updated on the **Permit Maintenance** form.
4. Click the **Apply** button on the **Permit Maintenance** form to save the changes.
5. Click the **OK** button to close the form.
6. Click **Clear Fields** button to clear all the fields.
7. Click the **Print** button to print the form.
8. Click the **OK** button to close the form.
9. Click the **Cancel** button to go to the previous screen.
10. Click the  button to get related Help information.

Special Info tab:

1. Click the Browse button (outlined in red) from the **Permit Maintenance** form.
2. It will open a Special Information Form Selection window.

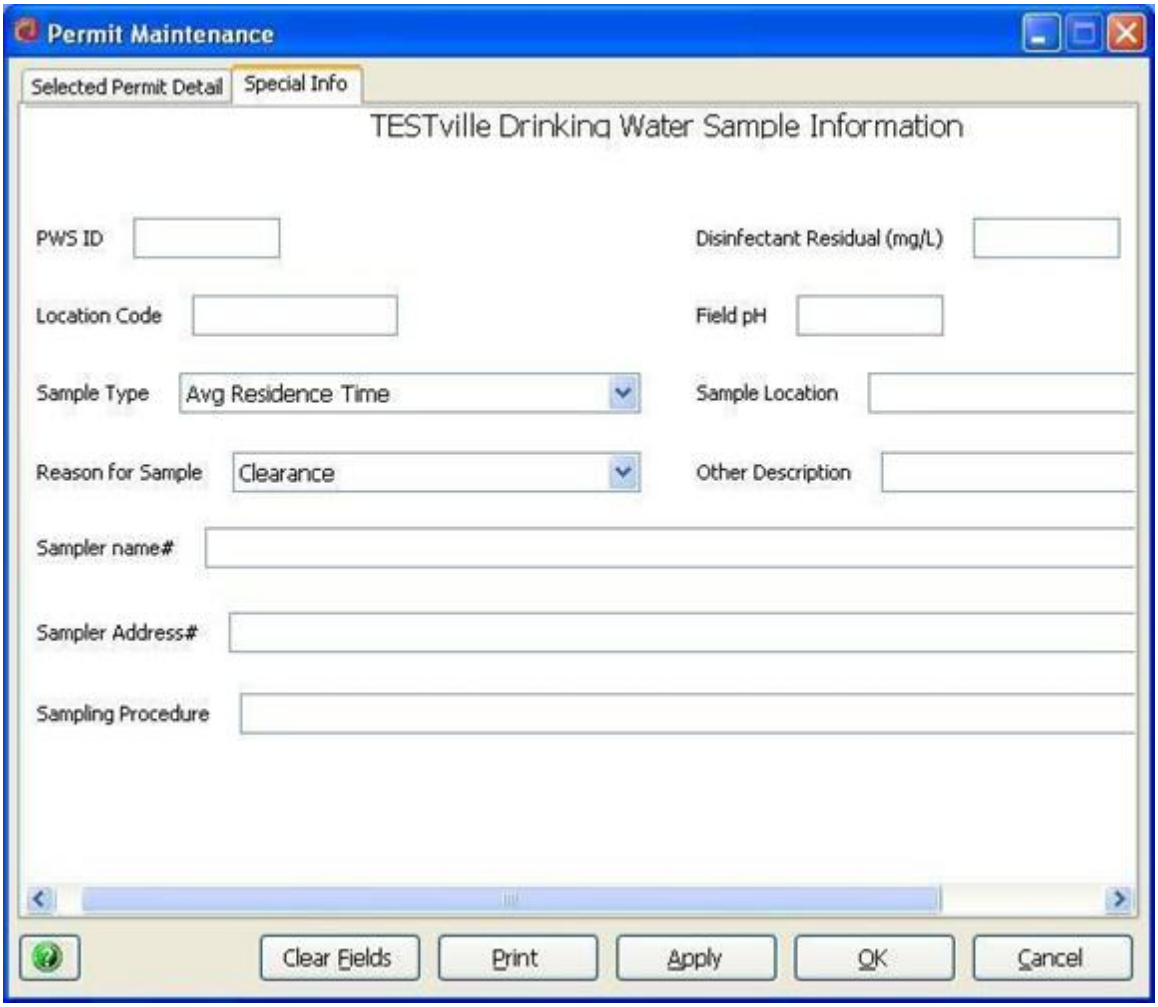


3. Select the required **Form Name** from the displayed list.
4. Click the **OK** button.
5. The selection will be updated on the Permit maintenance form (outlined in red).



6. Click the **Special Info** tab.

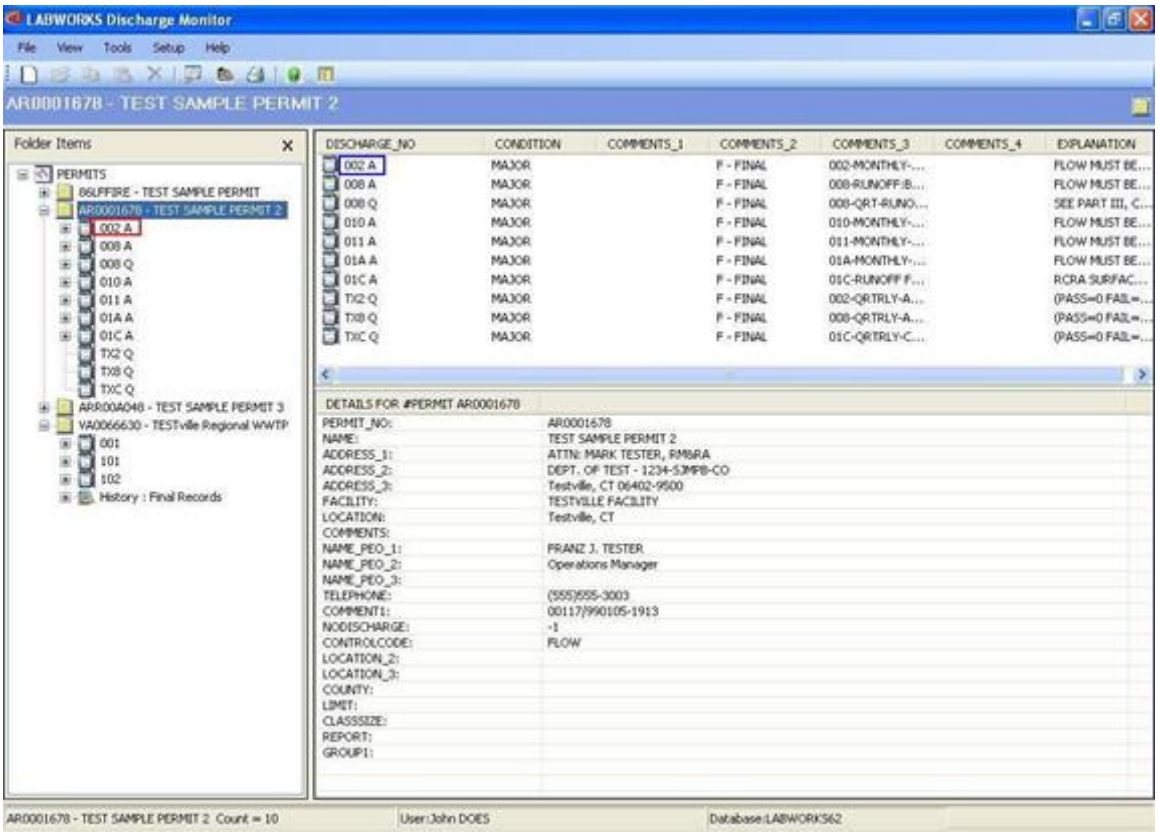
7. A new form will open shown in Figure 6 with details of the selected form displayed.



1. Fill in all the details in the respective text fields.
2. Click the **Apply** button to save the changes.
3. Click the **OK** button to save changes.
4. Click the **Clear Fields** button to clear the fields.
5. Click the **Print** button to print the form.
6. Click the **Cancel** button to reset the form.

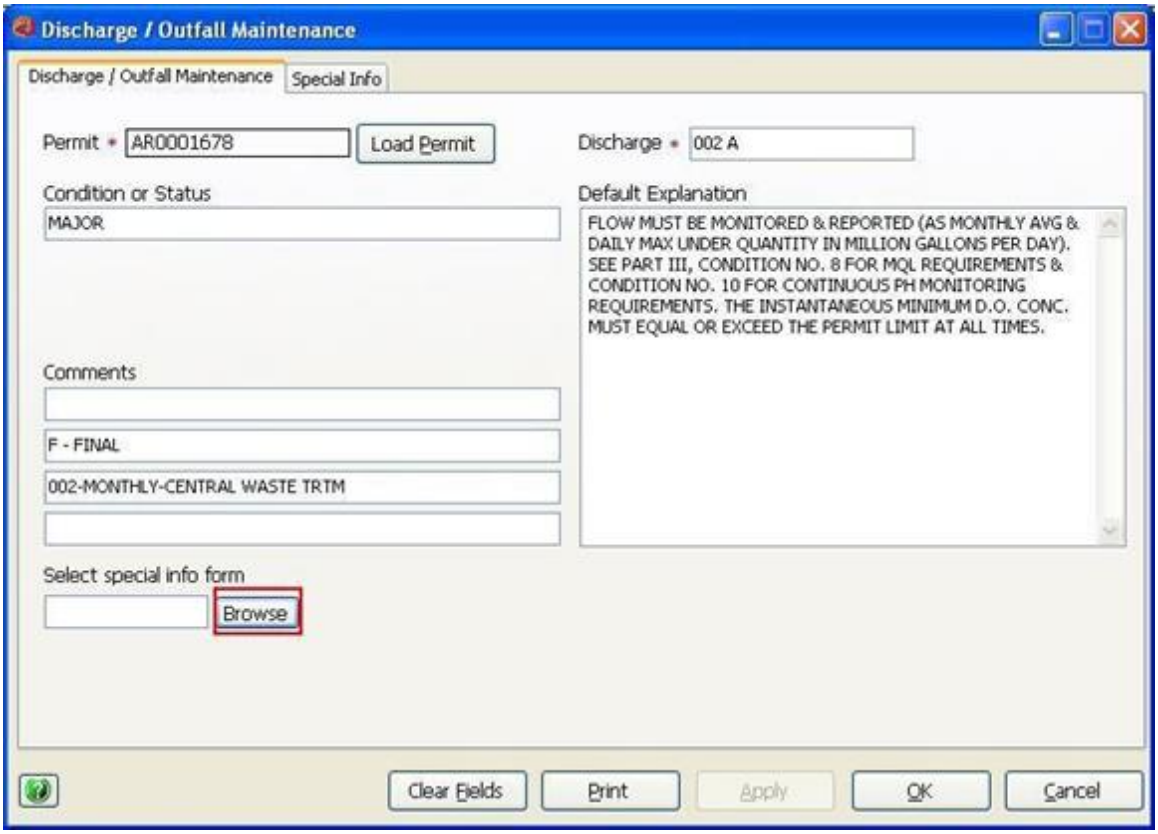
Discharge maintenance form

This section describes how to fill in the Discharge Maintenance form.

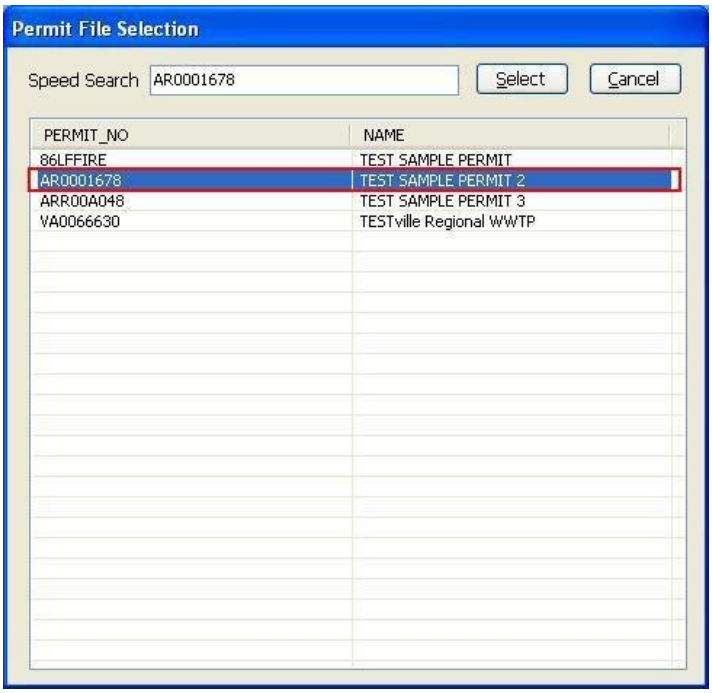


1. Click the **Discharge** node (Outlined in Red) **OR** (Outlined in Blue) to open the Discharge/Outfall Maintenance form.

2. The form is shown below.



3. Click the **Load Permit** button to open the **Permit File Selection** form.

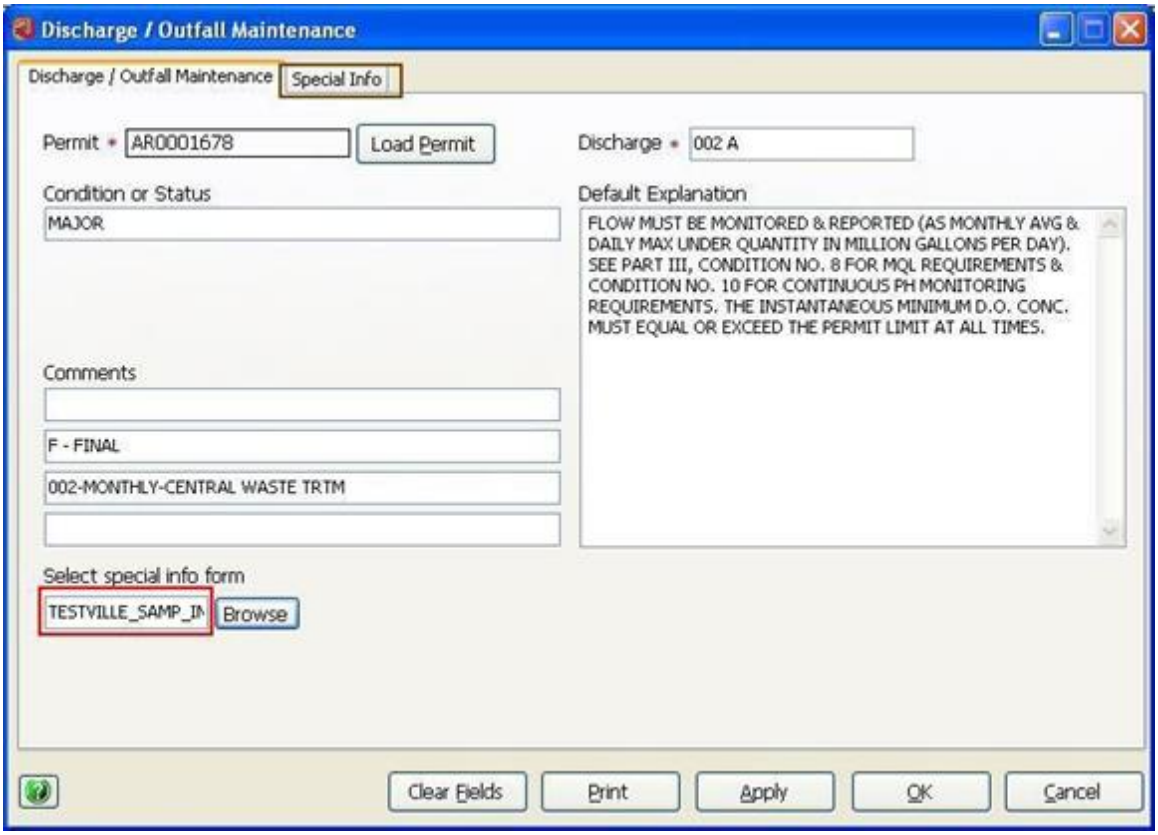


4. Select an option from the drop down list (Outlined in Red).
5. Click the **Select** button.
6. The information will be updated in the **Discharge/Outfall Maintenance** form. (Outlined in Red).
7. Fill in the other details.
8. Click the **Apply** button to save the changes.
9. Click **Clear Fields** button to clear all the fields.
10. Click the **Print** button to print the form.
11. Click the **OK** button to close the form.
12. Click the **Cancel** button to go to the previous screen.
13. Click the  button to get related **Help** information.

Special Info tab:

1. Click the Browse button shown in the **Discharge/Outfall Maintenance** form (outlined in red).
2. A new form will open as shown in the following example.

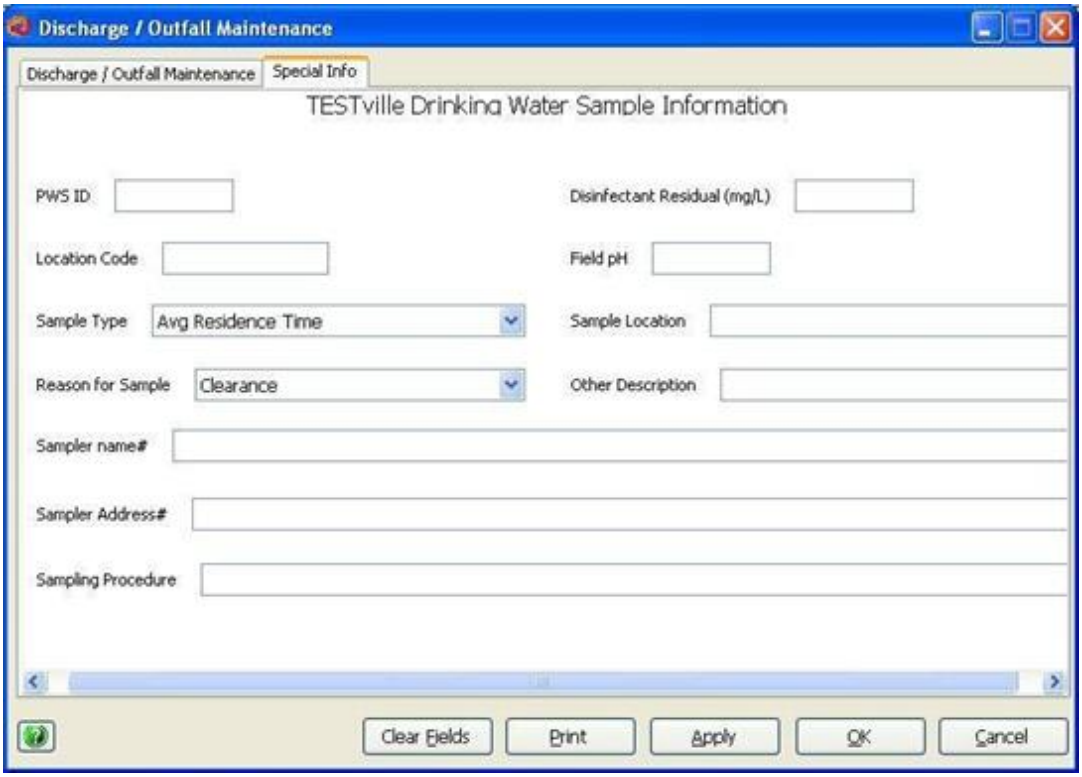
3. Select the required **Form Name** from the displayed list.
4. Click the **OK** button.
5. The form information will be updated on the Discharge/Outfall Maintenance form (outlined in red).



6. Click the **Apply** button to save the changes.

7. Click the **Special Info** tab (outlined in brown).

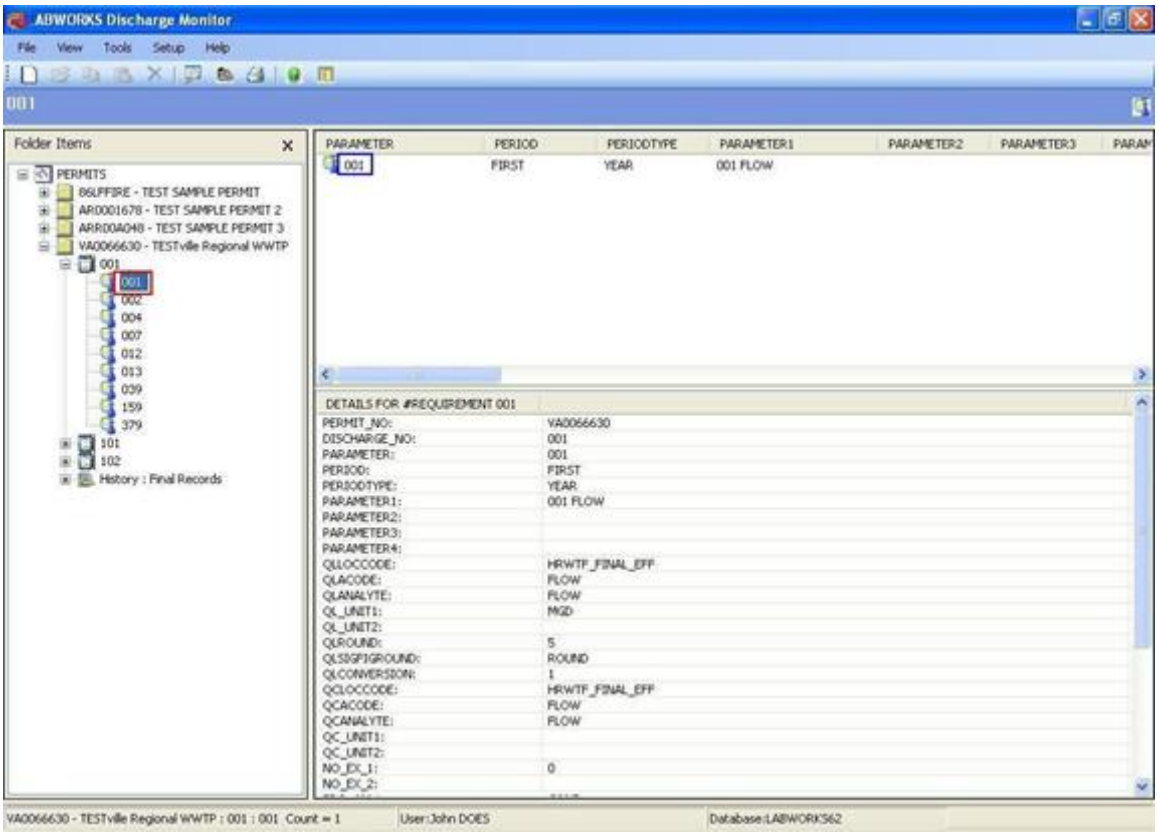
8. A new page will open as shown in the following example.



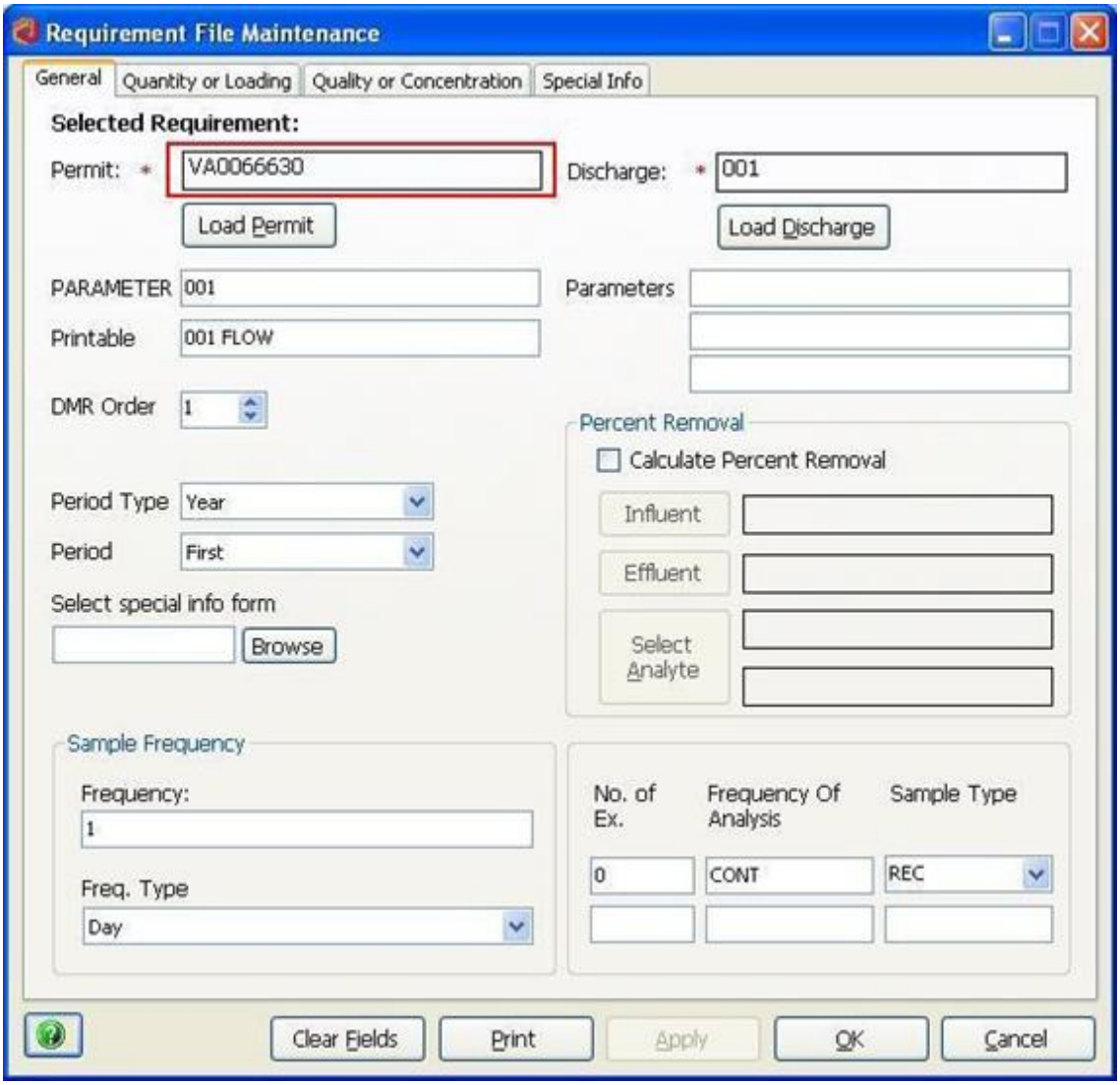
9. Fill in the details in the respective text fields.
10. Click the **Apply** button to save the changes.
11. Click the **OK** button to save the changes and close the form.
12. Click the **Print** button to print the form.
13. Click the **Cancel** button to reset the form.
14. Click the **Clear Fields** button to clear the fields.

Requirement / Parameters form

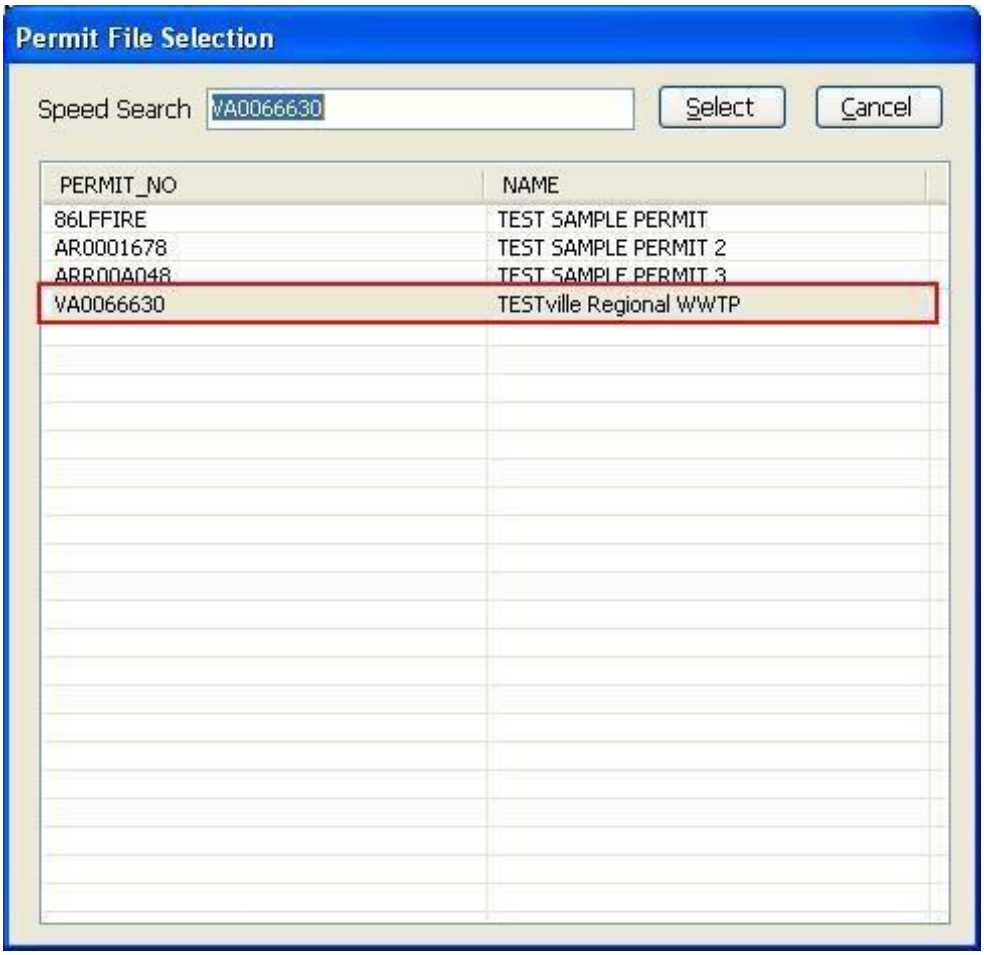
This section describes how to fill in the Requirements / Parameters form.



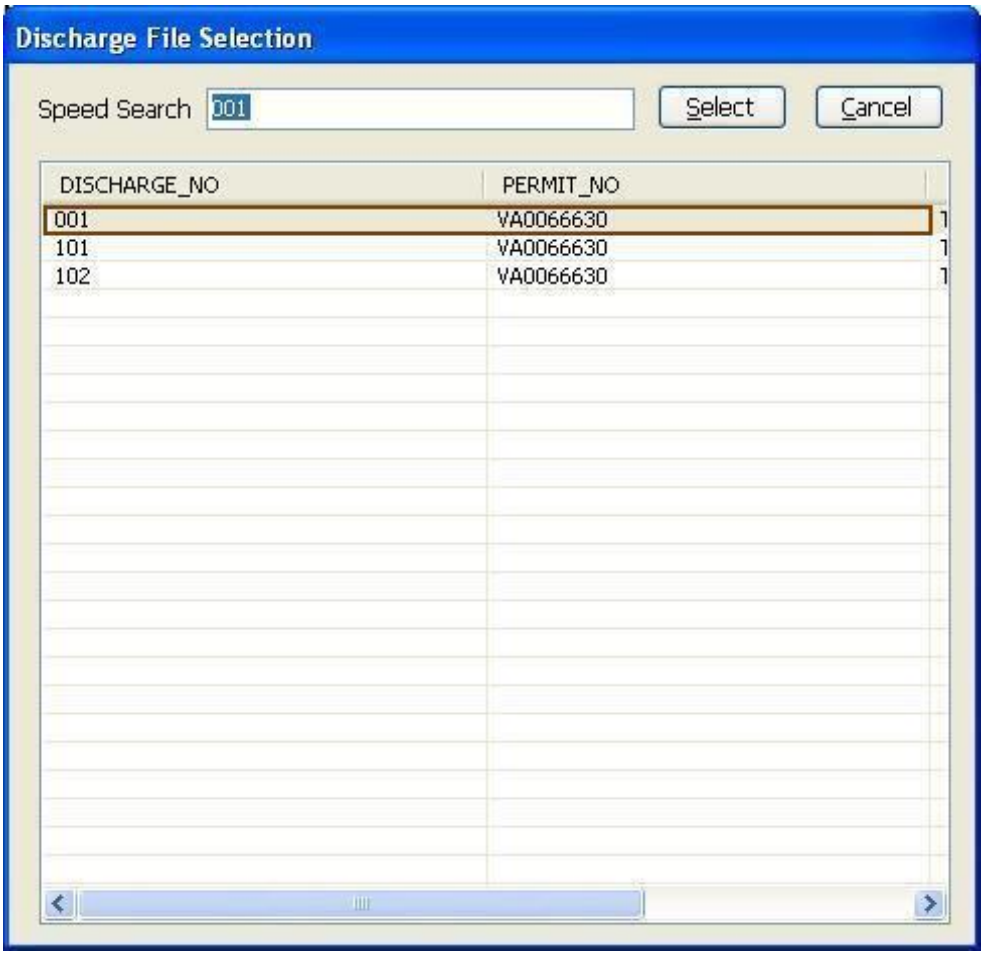
1. Click the Requirement/Parameters node Outlined in Red **OR** Outlined in Blue.
2. This will open the Requirement File Maintenance form.



3. Click the **Load Permit** button which will open the Permit File maintenance form.



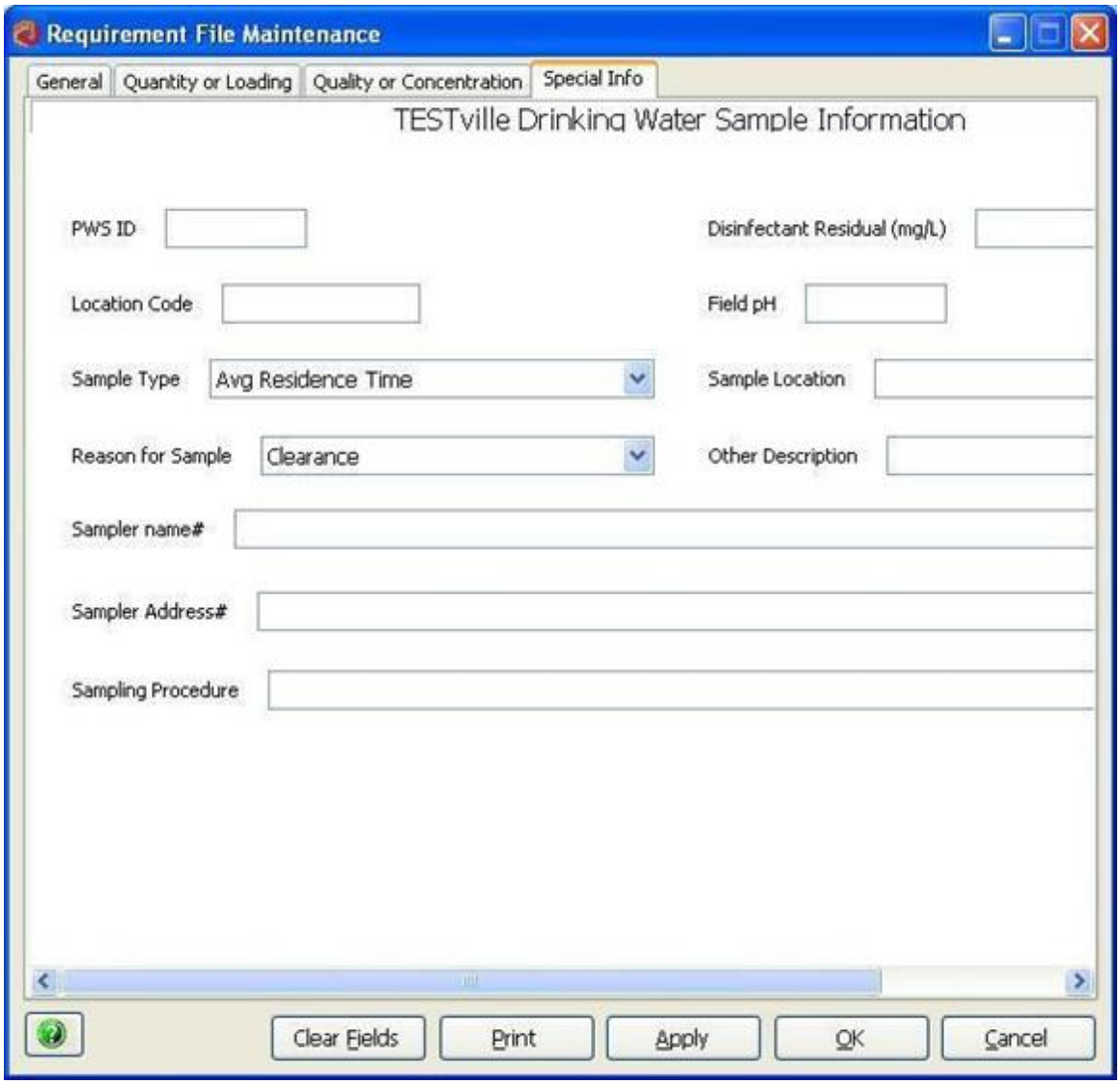
4. Select an option from the list in Figure 3 (Outlined in Red).
5. Click the **Select** button.
6. The information will be updated in the **Permit** field in the Requirement File Maintenance form. (Outlined in Red).
7. Click the **Load Discharge** button on the Requirement File Maintenance form to open the Discharge File Selection form.



8. Select an option from the drop down list (Outlined in Brown).
9. Click the **Select** button.
10. The information will be updated on the **Requirement File Maintenance** form (Outlined in Brown at figure 4).
11. Fill in the other details in the **Requirement File Maintenance** form.
12. Click the **Apply** button to save the changes.
13. Click **Clear Fields** button to clear all the fields.
14. Click the **Print** button to print the form.
15. Click the **OK** button to close the form.
16. Click the **Cancel** button to go to the previous screen.
17. Click the  button to get related Help information.

Special Info tab:

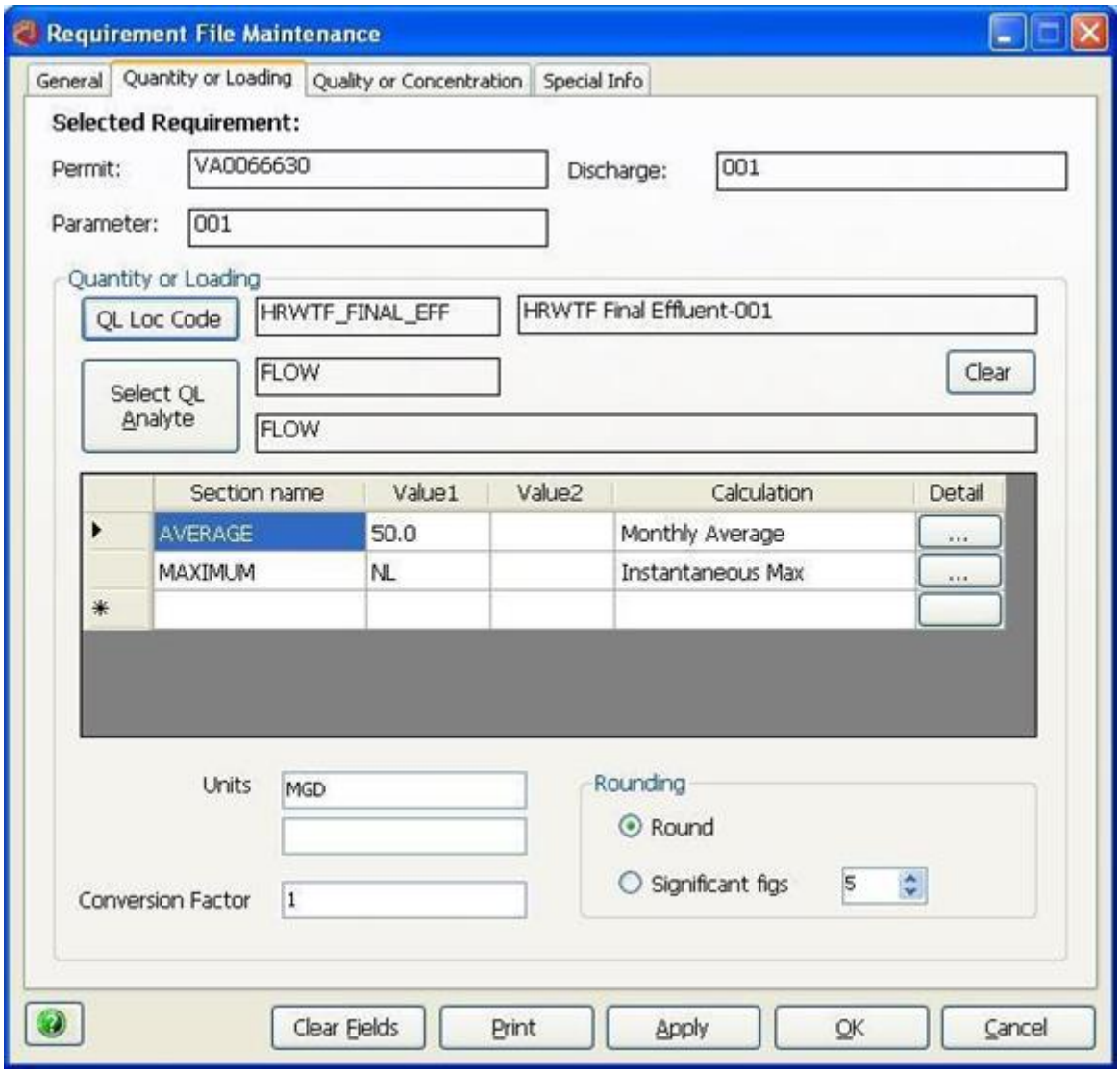
1. Click the **Browse** button on the Requirement File Maintenance form.
2. A new form will open.



3. Fill in the details in the respective text fields.
4. Click the **Apply** button to save the changes.
5. Click the **OK** button to save changes and close the form.
6. Click the **Cancel** button to reset the form.
7. Click the **Clear Fields** button to clear the fields.

Quality or Loading tab:

1. Clicking the **Quality or Loading** tab will open the screen shown in Figure 6 below.

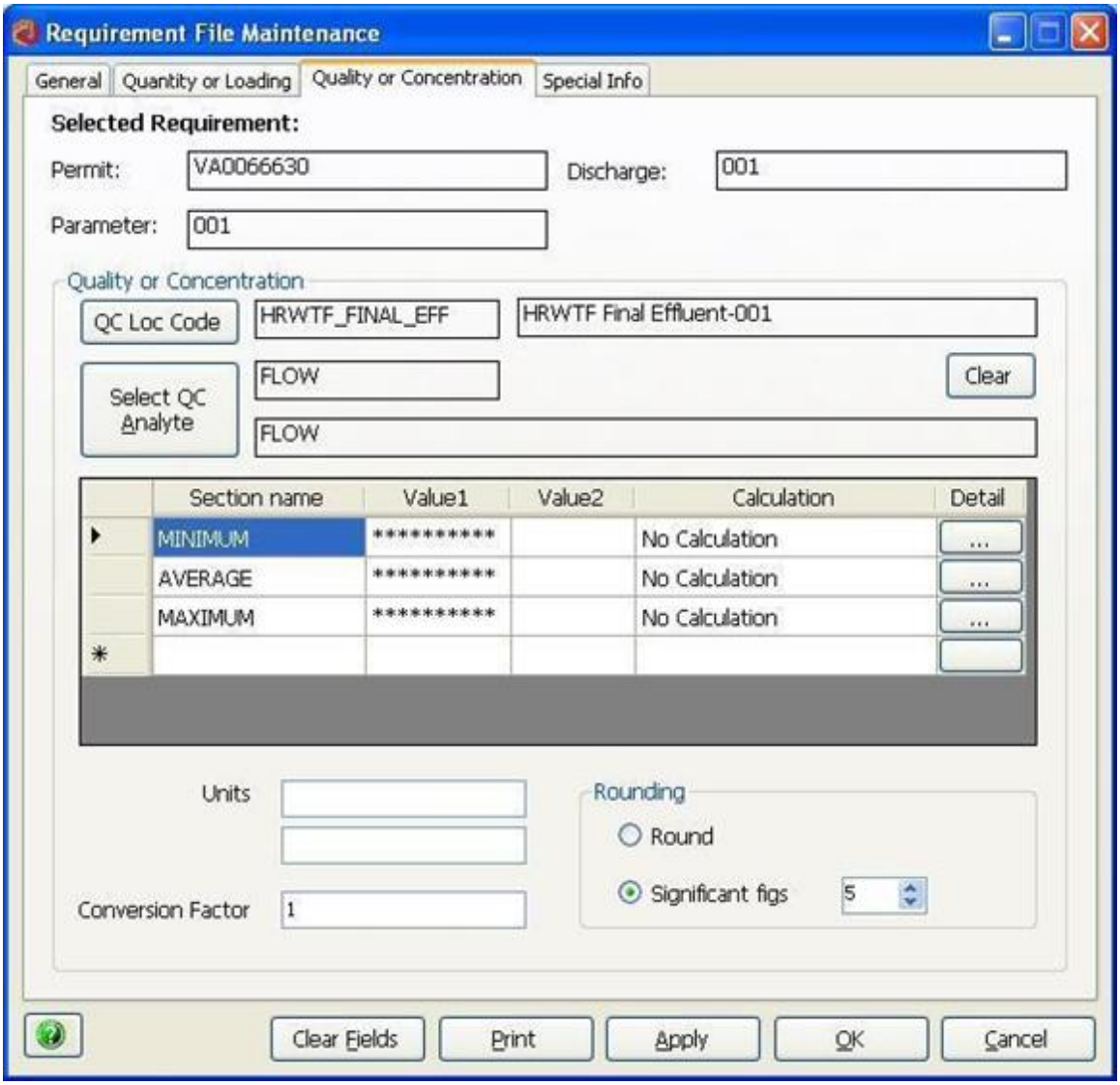


2. Fill in the respective fields.

3. In the **Section Name** grid section, extra rows can be added as needed by double clicking the row.

Quantity or Concentration tab:

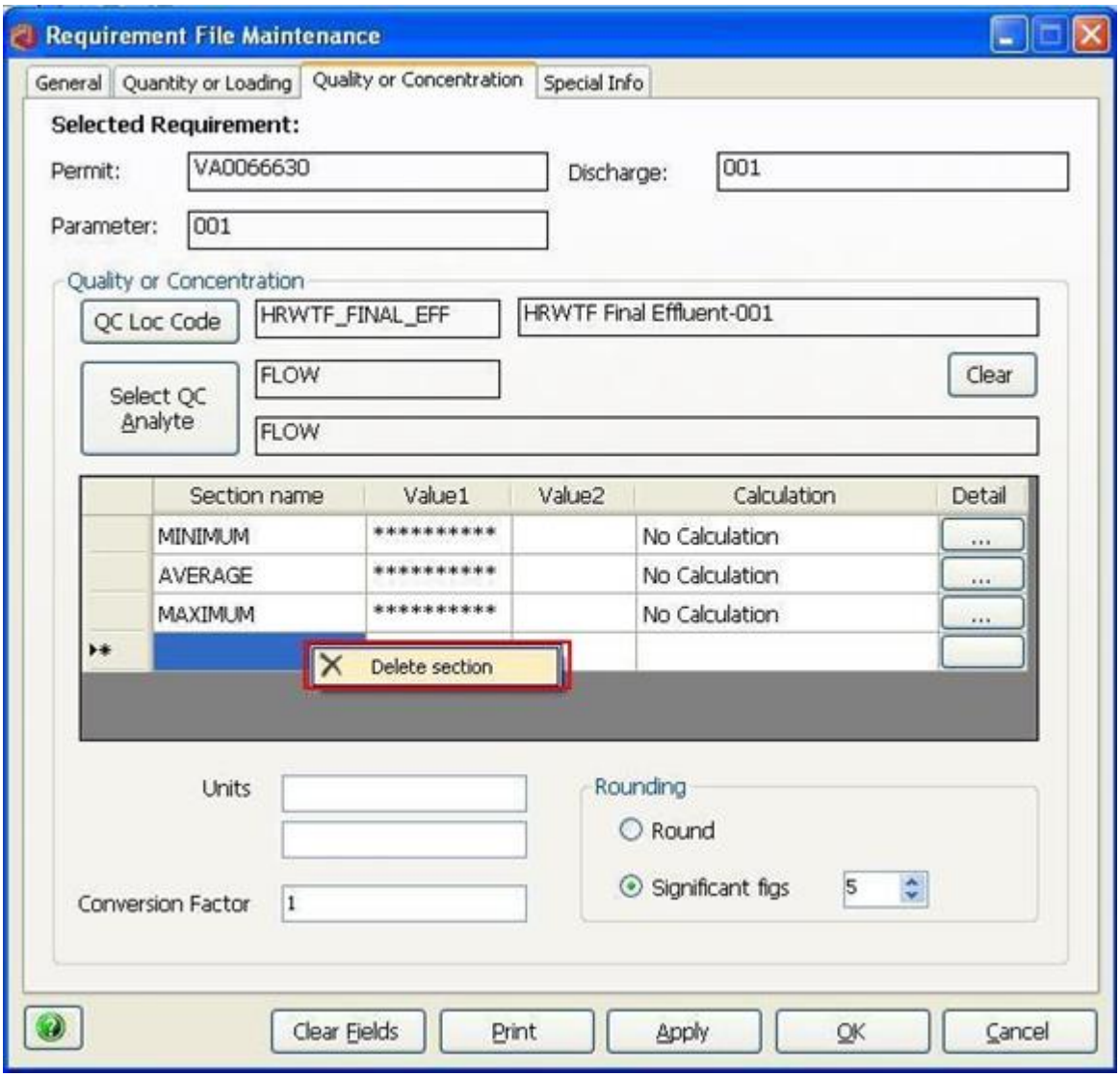
1. Clicking the **Quantity or Concentration** tab will open the screen as shown in the following example.



2. Fill in the respective fields.
3. In the **Section Name** grid section, extra rows can be added as needed by double clicking the row.

To Delete the row:

1. Right Click the row, a delete dialog box will open as shown in the following example (outlined in red).



2. Click the **Delete Section** to delete the row.

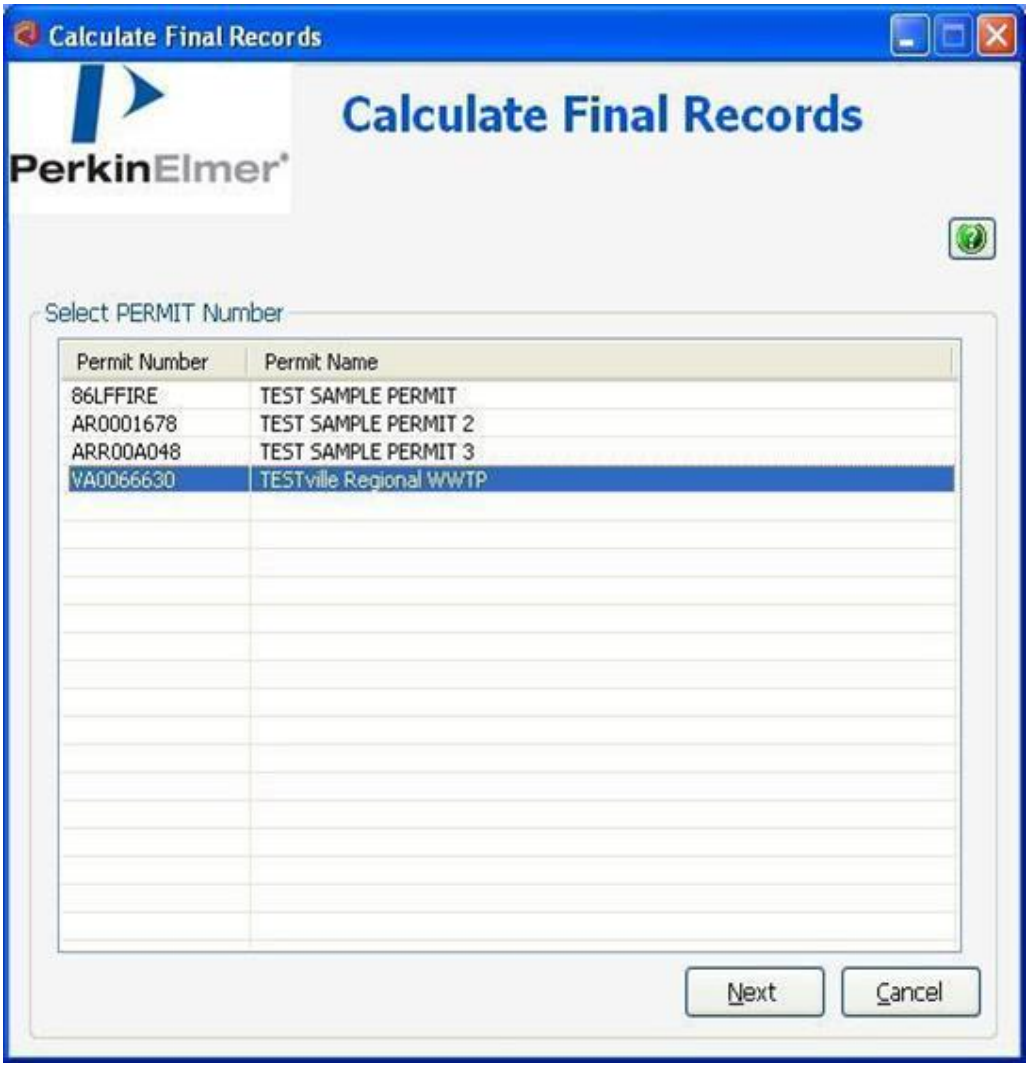
Calculate Final Record form

This section describes how to fill in the Calculate Final Record form.

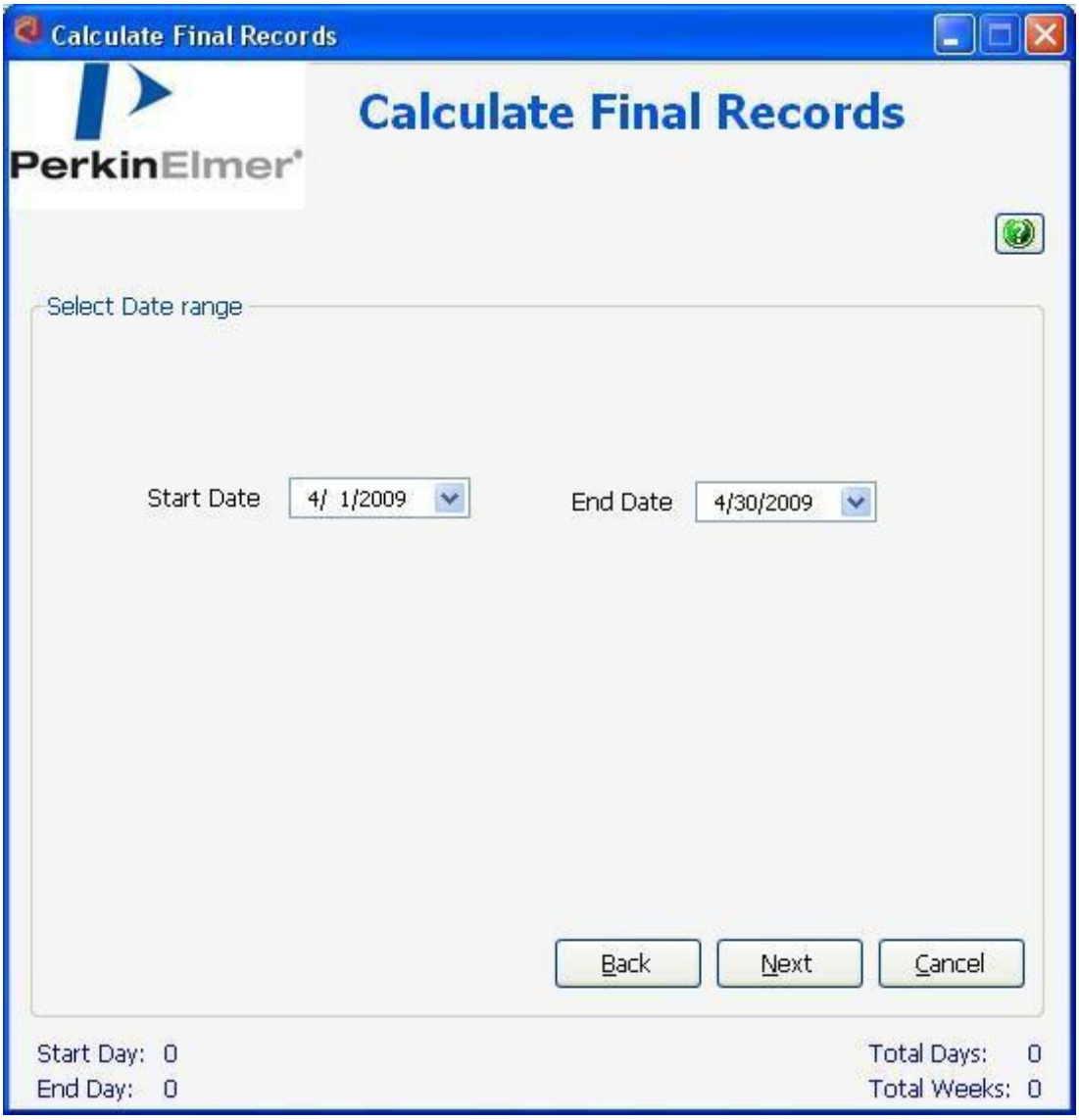
1. From the **Menu Bar** choose **Tools> Calculate Final Records OR**

2. From the **Tool Bar** click the icon to open the **Calculate Final Records** form.

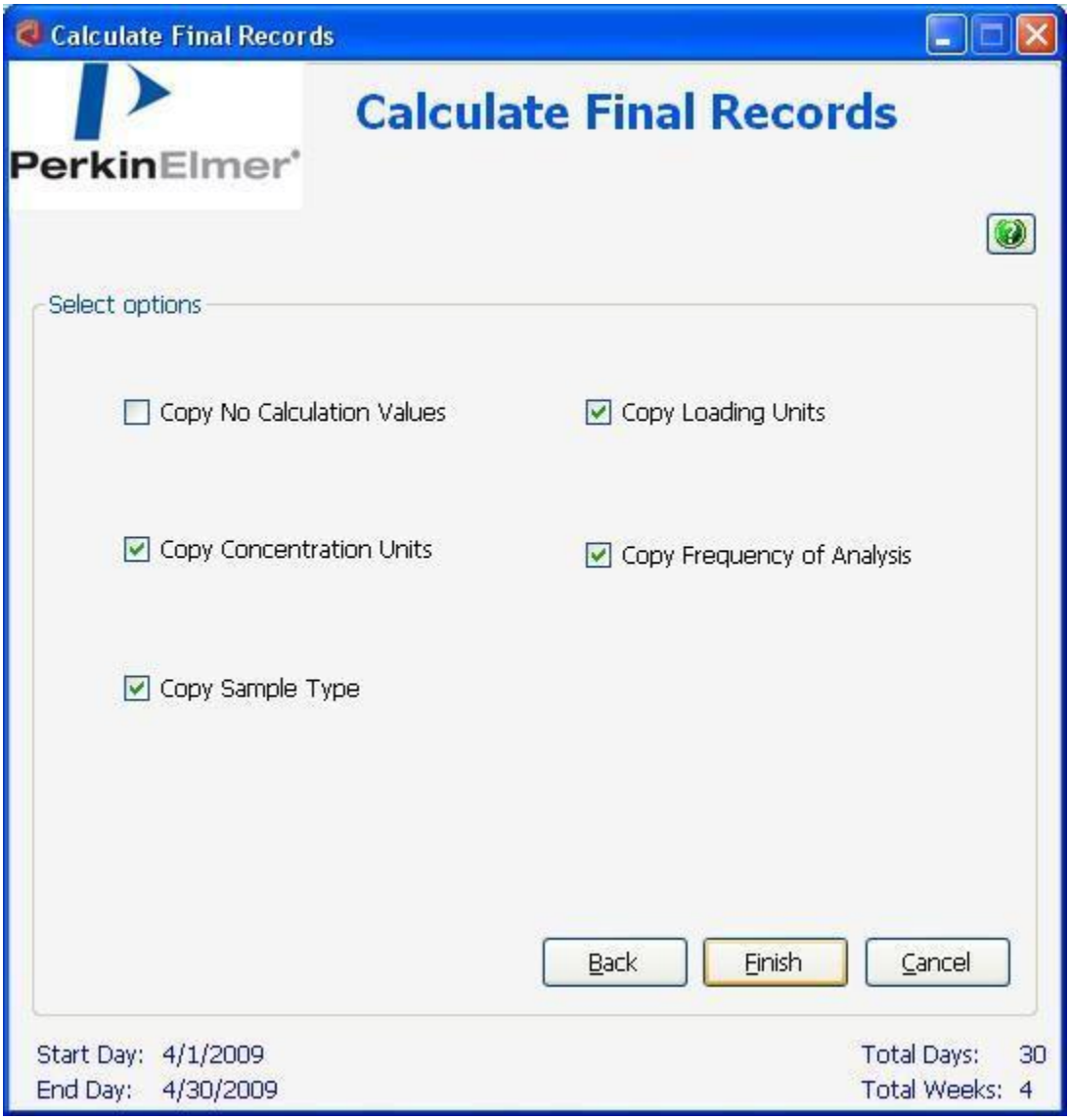
3. The Calculate Final Records form opens up as shown in the following example.



4. Select the **Permit Number** and **Permit Name** from the list.
5. Click the **Next** button.
6. The **Date Range** form will open as shown in the following example.



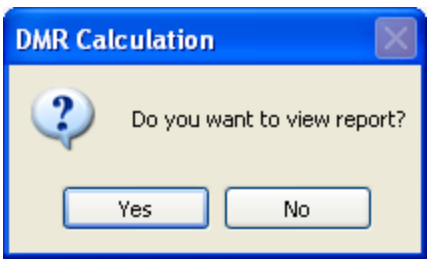
7. Select the **From Date** and **End Date** from the date picker.
8. Click the **Next** button.
9. The **Select** options form as shown in the following example will open.



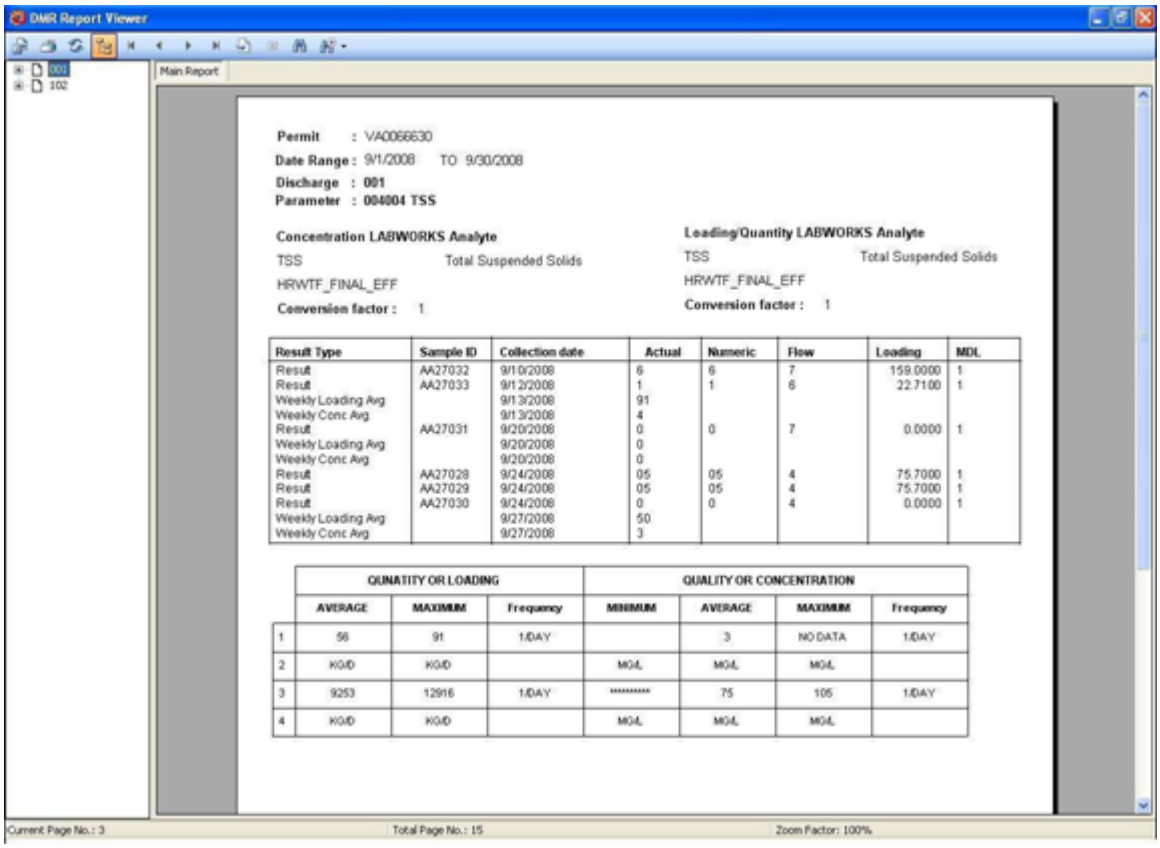
10. Check the required options and click **Finish**.

11. Application will do the calculations for the selected permit number.

12. After that the application will prompt the user with the message **"Do you want to view Report?"**



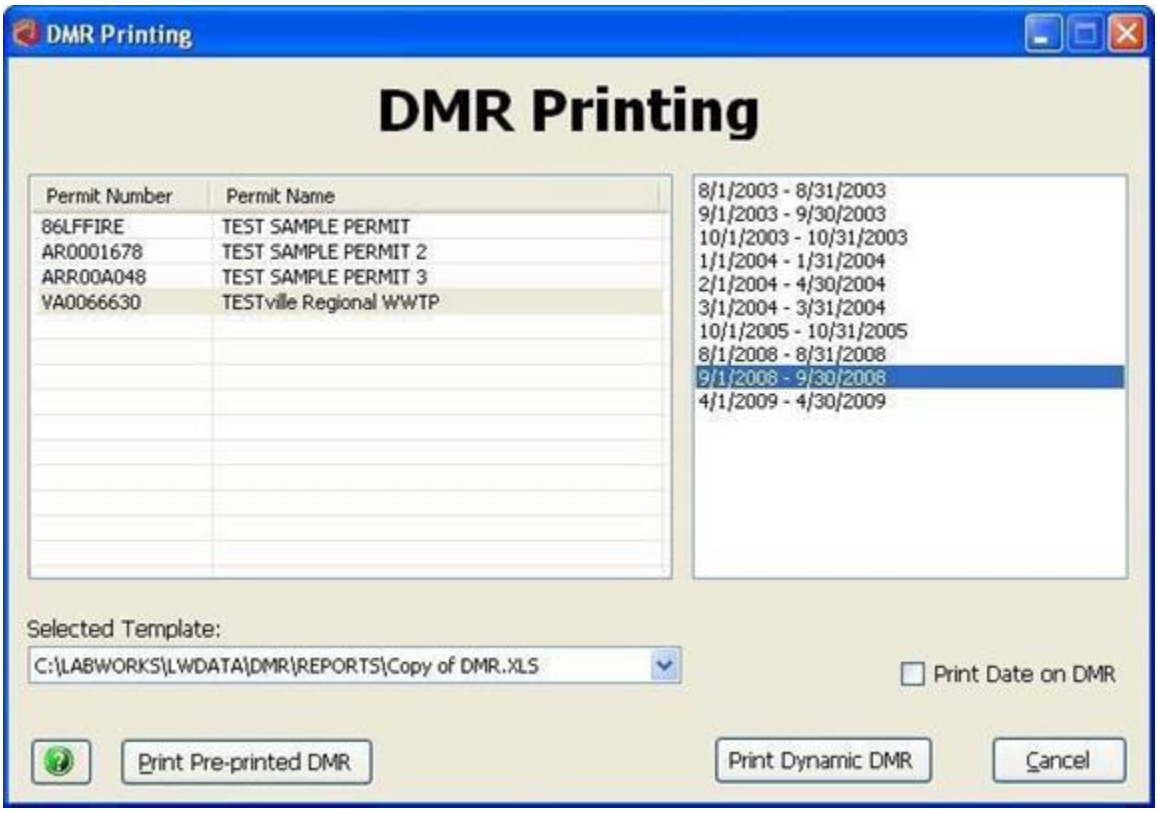
13. Click the **Yes** button to see the report shown in Figure 5. Click **No** to close the form.



Print DMR

The DMR printing form can be loaded by clicking the Print icon on the Tool Bar.

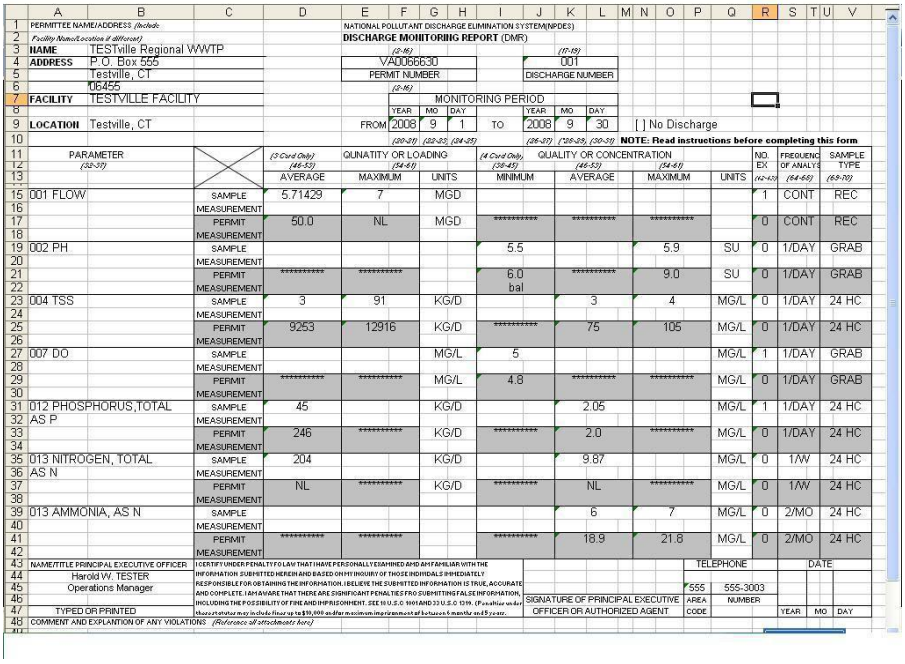
1. Click the  icon on the Tool bar.
2. A new page as shown in the following example will open.



3. Select the **Permit Number** and **Permit Name** from the displayed list.
4. Select the date range from the list.
5. From the **Selected Template** drop down, select the template.
6. Click the **Print Fix DMR** button **OR** the **Print Dynamic DMR** button to print the DMR report.

Print Fix DMR:

Clicking this button will allow the user to print the DMR report in fixed format based on the template selected. All the sections will not be covered if this option is selected, that means only standard sections will be printed. To print Customized section user has to select **Print Dynamic DMR** option.



NATIONAL POLLUTANT DISCHARGE ELIMINATION SYSTEM(NPDES)
DISCHARGE MONITORING REPORT (DMR)

NAME: TESTville Regional WWTP
ADDRESS: P.O. Box 555, Testville, CT 06455
FACILITY: TESTVILLE FACILITY
LOCATION: Testville, CT

VA0066630 (PERMIT NUMBER) | 001 (DISCHARGE NUMBER)

MONITORING PERIOD: FROM 2008 9 1 TO 2008 9 30 [ ] No Discharge

NOTE: Read instructions before completing this form

PARAMETER		QUANTITY OR LOADING AVERAGE	QUANTITY OR LOADING MAXIMUM	QUANTITY OR LOADING UNITS	QUALITY OR CONCENTRATION MINIMUM	QUALITY OR CONCENTRATION AVERAGE	QUALITY OR CONCENTRATION MAXIMUM	UNITS	NO. EX	FREQUENCY OF ANALYSIS	SAMPLE TYPE
001 FLOW	SAMPLE MEASUREMENT	5.71429	7	MGD					1	CONT	REC
	PERMIT MEASUREMENT	50.0	NL	MGD	**********	**********	**********		0	CONT	REC
002 PH	SAMPLE MEASUREMENT				5.5		5.9	SU	0	1/DAY	GRAB
	PERMIT MEASUREMENT	**********	**********		6.0 bal	**********	9.0	SU	0	1/DAY	GRAB
004 TSS	SAMPLE MEASUREMENT	3	91	KG/D		3	4	MG/L	0	1/DAY	24 HC
	PERMIT MEASUREMENT	9253	12916	KG/D	**********	75	105	MG/L	0	1/DAY	24 HC
007 DO	SAMPLE MEASUREMENT			MG/L	5			MG/L	1	1/DAY	GRAB
	PERMIT MEASUREMENT	**********	**********	MG/L	4.8	**********	**********	MG/L	0	1/DAY	GRAB
012 PHOSPHORUS,TOTAL AS P	SAMPLE MEASUREMENT	45		KG/D		2.05		MG/L	1	1/DAY	24 HC
	PERMIT MEASUREMENT	246	**********	KG/D	**********	2.0	**********	MG/L	0	1/DAY	24 HC
013 NITROGEN, TOTAL AS N	SAMPLE MEASUREMENT	204		KG/D		9.87		MG/L	0	1/W	24 HC
	PERMIT MEASUREMENT	NL	**********	KG/D	**********	NL	**********	MG/L	0	1/W	24 HC
013 AMMONIA, AS N	SAMPLE MEASUREMENT					6	7	MG/L	0	2/MO	24 HC
	PERMIT MEASUREMENT	**********	**********		**********	18.9	21.8	MG/L	0	2/MO	24 HC

I CERTIFY UNDER PENALTY OF LAW THAT I HAVE PERSONALLY EXAMINED AND AM FAMILIAR WITH THE INFORMATION SUBMITTED HEREIN AND BASED ON MY INQUIRY OF THOSE INDIVIDUALS IMMEDIATELY RESPONSIBLE FOR OBTAINING THE INFORMATION, I BELIEVE THE SUBMITTED INFORMATION IS TRUE, ACCURATE AND COMPLETE. I AM AWARE THAT THERE ARE SIGNIFICANT PENALTIES FOR SUBMITTING FALSE INFORMATION, INCLUDING THE POSSIBILITY OF FINE AND IMPRISONMENT. SEE 18 U.S.C. 1001 AND 33 U.S.C. 1319. (Penalties under these statutes may include fines up to $10,000 and maximum imprisonment of between 6 months and 5 years.)

NAME/TITLE PRINCIPAL EXECUTIVE OFFICER: Harold W. TESTER, Operations Manager (TYPED OR PRINTED)

SIGNATURE OF PRINCIPAL EXECUTIVE OFFICER OR AUTHORIZED AGENT

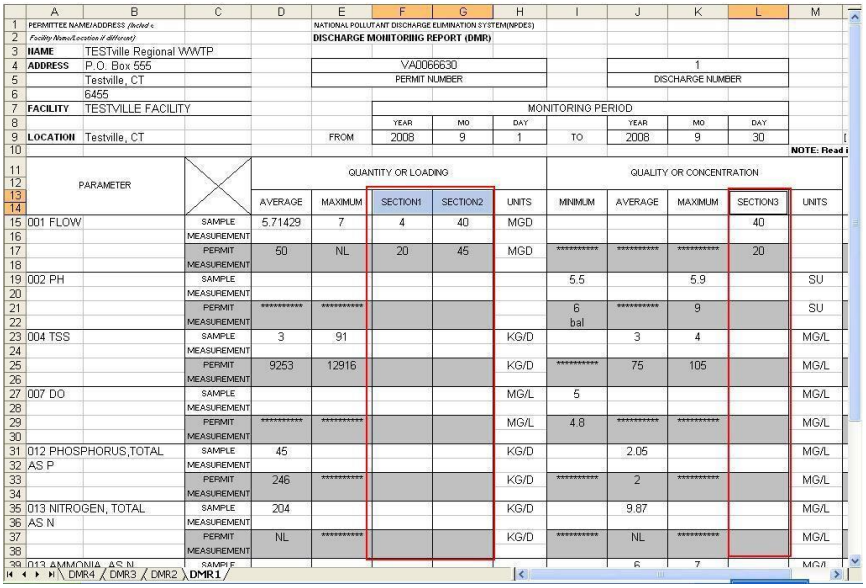
TELEPHONE: AREA CODE 555, NUMBER 555-3003

DATE: YEAR, MO, DAY

COMMENT AND EXPLANTION OF ANY VIOLATIONS (Reference all attachments here)

Print Dynamic DMR:

Clicking this button will allow the user to print the DMR form in dynamic format with all sections covered. A new template is made which will include all the sections.



NATIONAL POLLUTANT DISCHARGE ELIMINATION SYSTEM(NPDES)
DISCHARGE MONITORING REPORT (DMR)

NAME: TESTville Regional WWTP
ADDRESS: P.O. Box 555, Testville, CT 6455
FACILITY: TESTVILLE FACILITY
LOCATION: Testville, CT

VA0066630 (PERMIT NUMBER) | 1 (DISCHARGE NUMBER)

MONITORING PERIOD: FROM 2008 9 1 TO 2008 9 30

NOTE: Read i

PARAMETER		QUANTITY OR LOADING AVERAGE	MAXIMUM	SECTION1	SECTION2	UNITS	QUALITY OR CONCENTRATION MINIMUM	AVERAGE	MAXIMUM	SECTION3	UNITS
001 FLOW	SAMPLE MEASUREMENT	5.71429	7	4	40	MGD				40	
	PERMIT MEASUREMENT	50	NL	20	45	MGD	**********	**********	**********	20	
002 PH	SAMPLE MEASUREMENT						5.5		5.9		SU
	PERMIT MEASUREMENT	**********	**********				6 bal	**********	9		SU
004 TSS	SAMPLE MEASUREMENT	3	91			KG/D		3	4		MG/L
	PERMIT MEASUREMENT	9253	12916			KG/D	**********	75	105		MG/L
007 DO	SAMPLE MEASUREMENT					MG/L	5				MG/L
	PERMIT MEASUREMENT	**********	**********			MG/L	4.8	**********	**********		MG/L
012 PHOSPHORUS,TOTAL AS P	SAMPLE MEASUREMENT	45				KG/D		2.05			MG/L
	PERMIT MEASUREMENT	246	**********			KG/D	**********	2	**********		MG/L
013 NITROGEN, TOTAL AS N	SAMPLE MEASUREMENT	204				KG/D		9.87			MG/L
	PERMIT MEASUREMENT	NL	**********			KG/D	**********	NL	**********		MG/L
013 AMMONIA, AS N	SAMPLE							6	7		MG/L

DMR4 | DMR3 | DMR2 | DMR1

The section outlined in red are the customized sections which are printed only on clicking the **Print Dynamic DMR** button.

8. Automated Results Posting

Introduction to Automated Results Posting

Automated posting in LABWORKS is a means of entering results into the LIMS without the use of the Results Entry module itself. There are many activities going on in the background when a user enters results. These same activities happen when another application posts results into the system through an automated process.

Tools used for posting results automatically up to LABWORKS 6.1 has been the following:

1. PostSCRO (5.8) and PostSCR6 (6.0 and 6.1) - used for command-line single component results posting
2. PostMCRO (5.8) and PostMCR6 (6.0 and 6.1) - used for command-line multi-component results posting

In LABWORKS 6.2, the automatic posting of results was replaced by PostResults6. PostResults6 is a new command-driven module for posting both single and multi component tests through either the legacy input (CSV) format from PostSCRO / PostMCRO or via a new Excel input format that can be predefined with several additional input values other than only results. PostResults6 has the capabilities of importing not only results but also data such as qualifiers, dilution factors, MDL, PQL, start/end dates, etc. This gives an external lab, for example, the ability to adjust key "result related" fields automatically (if anything is changed or needs adjustment during results entry) without the destination lab's involvement.

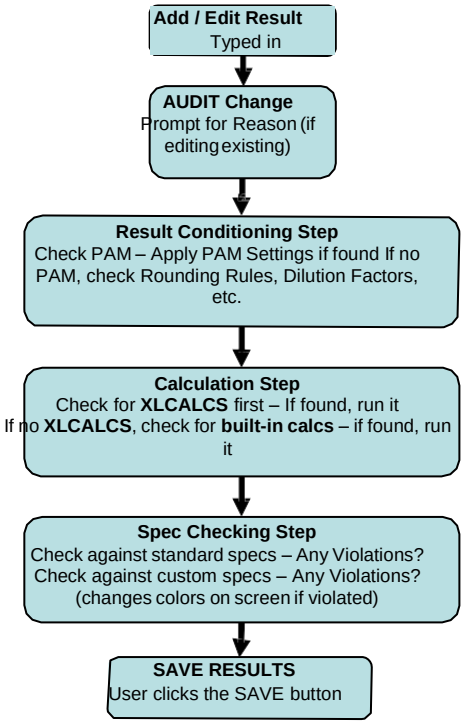
Launching and configuring PostResults6 is best through the user interface (UI) tool, LWPostResults6 which will be covered initially in this chapter. This tool greatly simplifies both the setup and launching of the command-line tool by the user of a step-by-step setup wizard.

Firing off the posting of reports can also happen by launching the UI tool, LWPostResults6, with the AR:1 parameter after it. Or, you can configure a shortcut with the proper command-parameters to launch PostResults6 directly from the LABWORKS Desktop, Process Scheduler, or Scan Daemon.

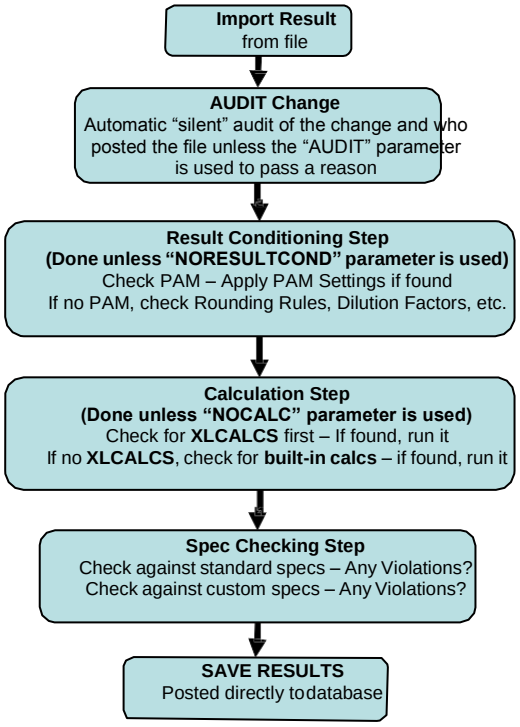
Diagram of Results Posting Process

Whether the process is done manually in the Results Entry module or automatically via an automated posting tool such as PostResults6, the steps that occur from results data input to database storage is the same (for the most part). The following diagram shows the process from both the manual and automated side in a side-by-side comparison.

Results Entry Module Process



PostResults6 Process



As diagramed, there are a series of steps that the result is evaluated against (inorder) before anything is posted to the database.

1. Audit Step - If a change is made to an existing result, the system will audit that change using both automated or manual methods. If the audit occurs within automated posting, there will be no prompt for a reason unless PostResults6 is passed the AUDIT parameter.
2. Result Conditioning Step - If PAM, Project Analyte Maintenance, data is found which will evaluate the entered or imported result against other values (which could change the result value), it is executed. If not, the system will continue to check other areas of the system for rounding rules, standard "text results", or dilution factor values which all could also change the result value before posting to the database.
3. Calculation Step - After result is "conditioned", if XLCALCS exist, calculations which are Microsoft Excel based, the system will fire these calculations and NOT execute the built-in calculations. If there are no XLCALCS for the specific result, then the built-in calculations are executed. This step could also change the result, but only if it is an OUTPUT parameter result.
4. Specification Checking Step - If standard or custom specs are entered for the analyte being evaluated, it will determine if a violation exists based on the conditioned or calculated result value. If a violation or warning occurs, it will change the color within the UI if manually entering the value in the Results Entry module. If automatic posting, it will simply store the violation in the database. This step will NOT change the result value itself.

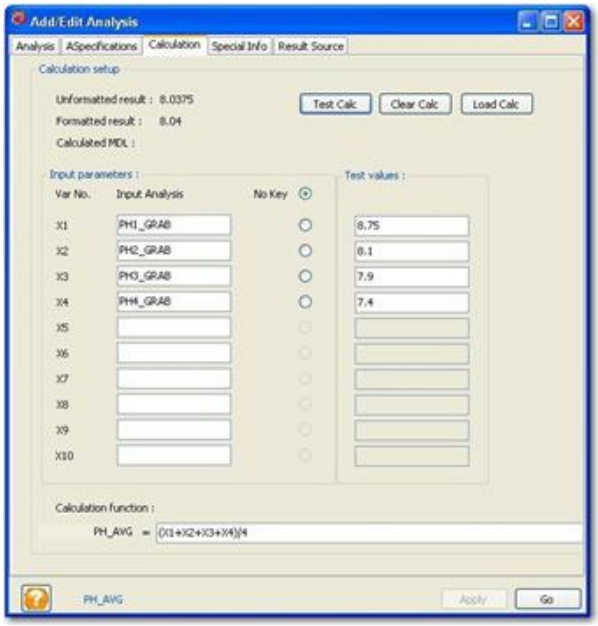
Regardless what method the user uses to enter results into LABWORKS (in build 6.2 or later), this is the process. Please note that steps 2 and 3 can change the original result value while steps 1 and 4 do not.

Results Posting ProcessExample

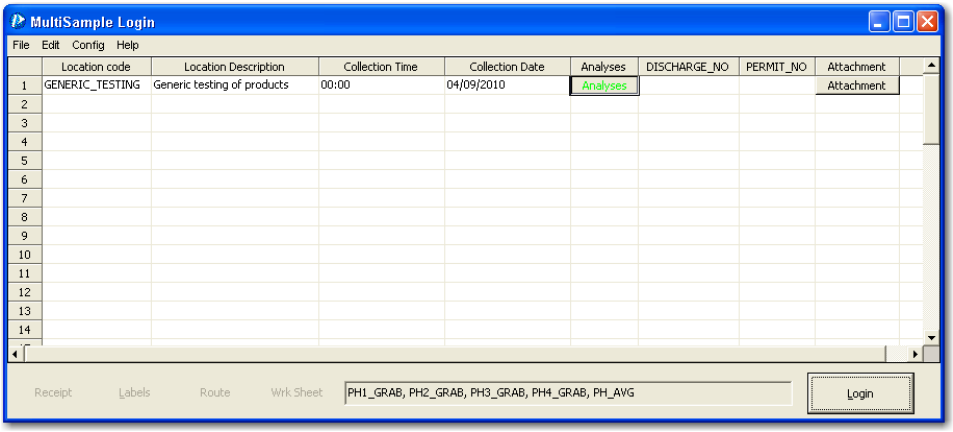
To understand the process and the 4 steps, a simple example will be used.

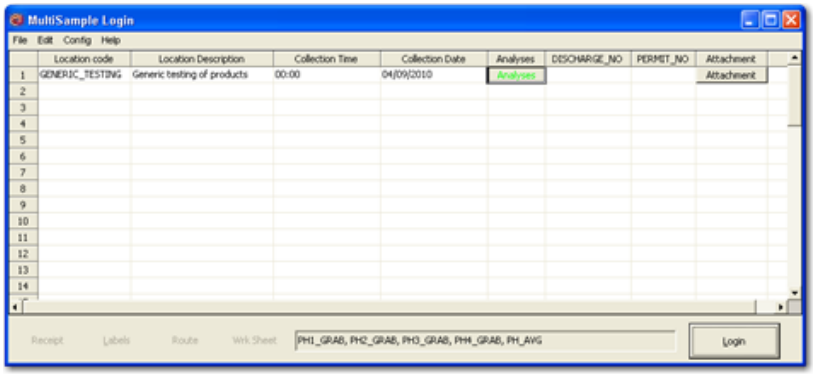
First of all, four analysis codes for PH is created for water "grab" samples done four times during the day. Another code, PH_AVG, is used to determine the average of those four samples. We will show both the internal calculations and Excel calculations.

For built-in calculations, we access the Calculation tab within the analysis code itself that will be storing the output of the calculation. Here we can add the four inputs (PH1 - PH4) for the final PH_AVG output. Then, we can test to make sure it is correct. When we are done, we can click the APPLY button to commit the changes. W e can adjust the specifications on the ASpecifications tab or within a location code that the analysis code is assigned to (if desired).

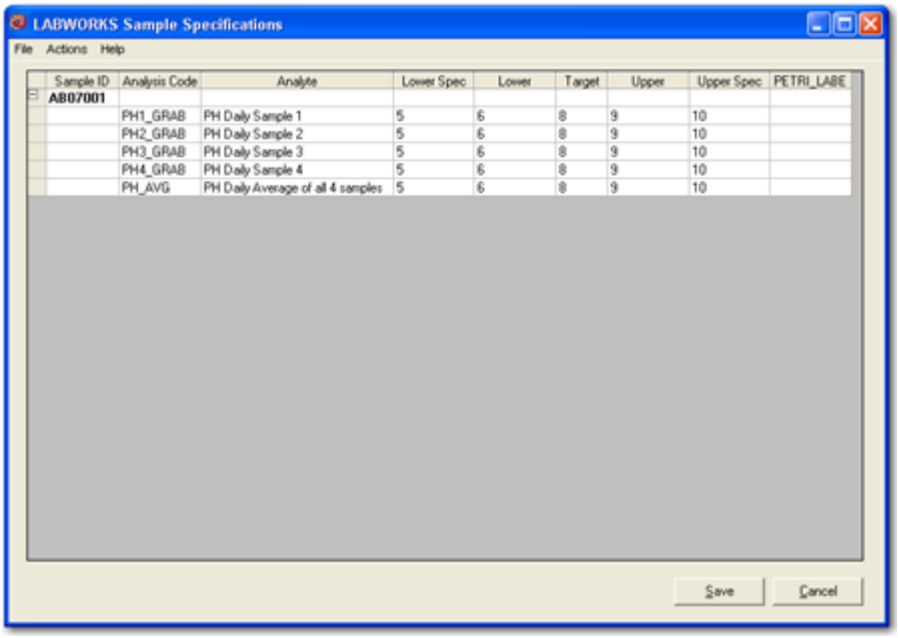


Now we can log in some samples that include all three of the test codes.

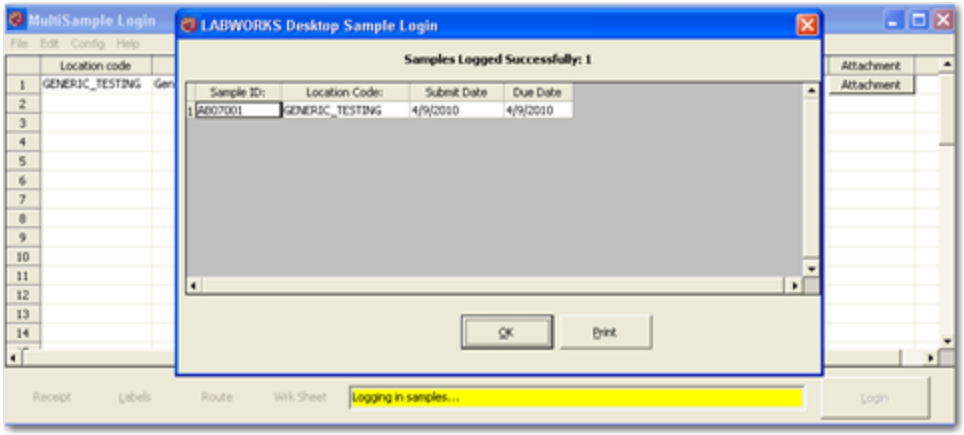




If we plan to adjust specifications at sample login (which is sometimes required because specs could change from customer to customer in contract labs), we can use the SampSpecLogin6 program under Post Login Settings.

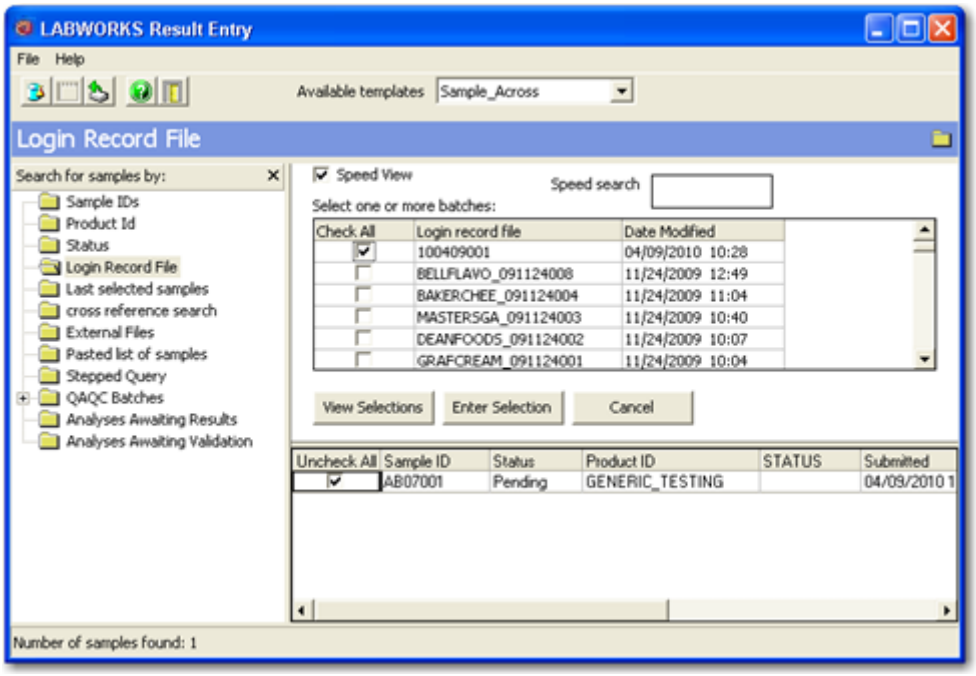


Specifications are important and will be evaluated within "step 4" of the results posting process. Now, proceed to login the samples.



We now have a sample (AB07001) in our system ready for results posting (manually or automated). Let's proceed to do the manual posting to visually demonstrate each step as defined.

Open Results Entry and select our sample.

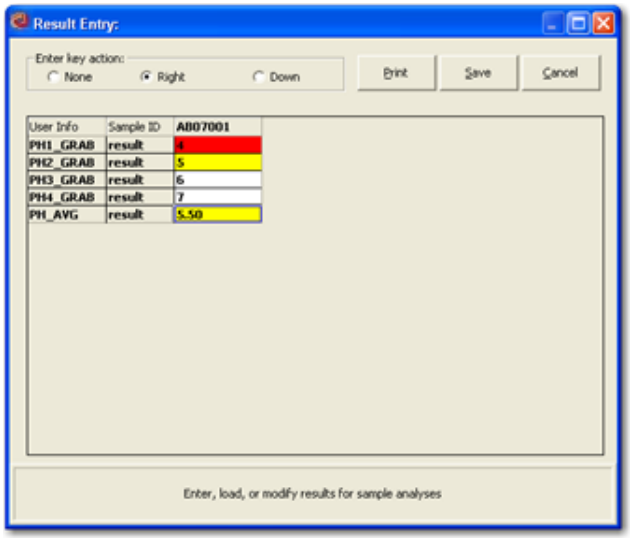


Now we enter the results.

As we proceed to enter the "input" results, the calculation that we defined in PH_AVG is executed once all the input

values are entered. Also notice the specification step is executed as well since we entered specifications at login. Once we did this, the values posted to the database after login had the specifications that we entered. These login specifications may not be the same as the default specifications. However, what we logged in (if we changed them), will take precedence.

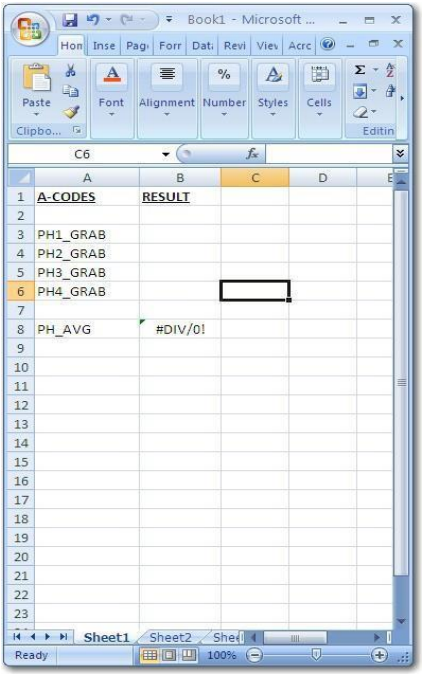
We can see that immediately following the final "input" value entry (PH4_GRAB), the PH_AVG is correctly calculated.



Since this is a new result posting, the AUDIT step was skipped. If this was an existing (already posted) set of results, the AUDIT prompt would have launched for each value entered (assuming the audit trail was turned on and operational). The Results Conditioning step was also skipped because we did not have PAM data or any other means of changing the result value entered into the system. The Calculation step (3), however, did execute. The Specifications Checking step (4), also executed properly since we see the cells were colored showing the violations.

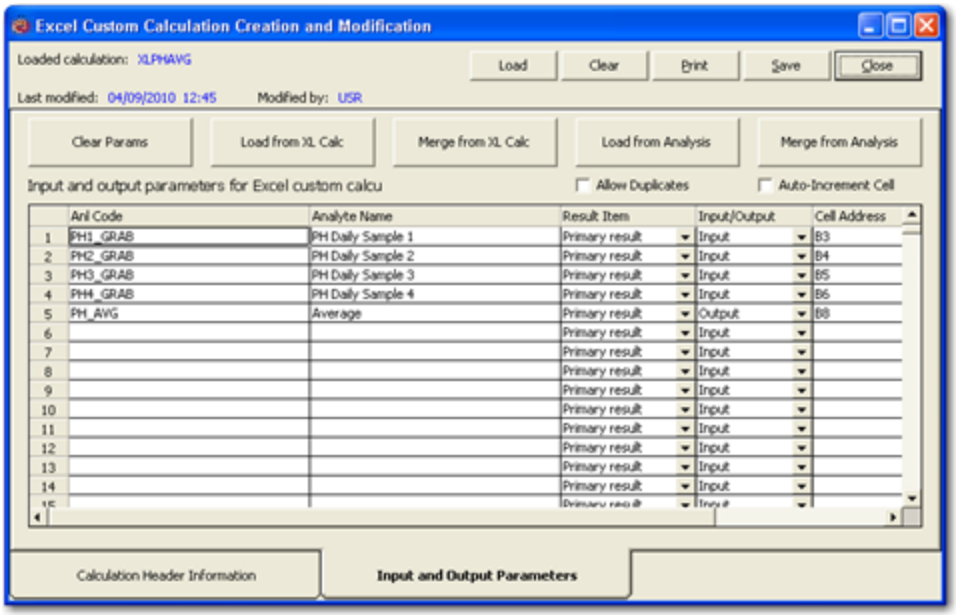
We can now post these results by clicking SAVE. All four steps defined in the previous section executed successfully within the Results Entry module for each of these results. Note that Excel calculations were checked first, but since we did not have any in the system for these analysis codes, the built-in calculations were executed.

Now we can proceed to enter an Excel calculation by opening the XLCALC module (XLCALC6.EXE). First, we create an Excel sheet with the calculations needed to produce the same result.

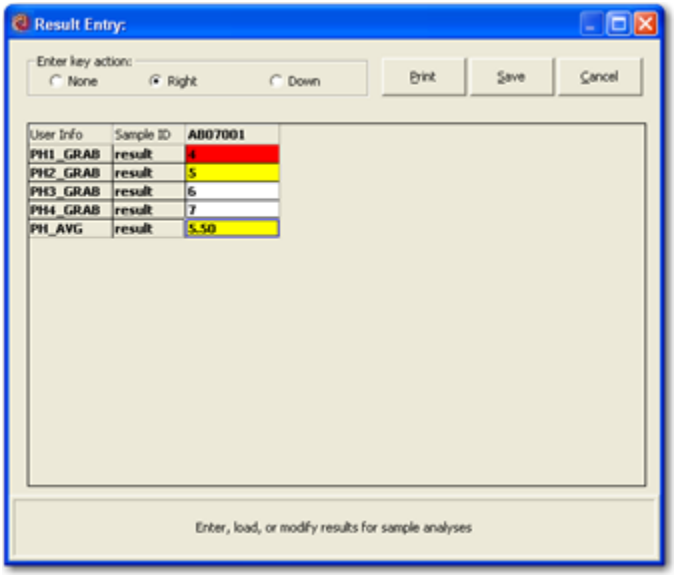


We then proceed to save the Excel sheet within the XLCALC folder of the data path.

We now launch the XLCALC module and enter a new XL Calculation defining our inputs and 1 output.



Going back into Results Entry, we see the same result. The only difference is now we notice our Excel sheet is launched, accessed for the "output" cell, then closed.



Note that as long as an Excel Calculation exists, the built-in calculation will NOT get executed. This is the case in both the manual and automated process. W e must remove any Excel Calculations first before built-in calculations are executed.

As before, we "post" the results to the database with the SAVE button.

The same series of steps happen during the automated posting of results as well just without the UI.

The following section will now discuss how the automated posting modules (LWPostResults6 and PostResults6) are configured and used for automated results posting (doing the same process in our previous example automatically from a file import). The remaining chapters will cover the above steps as it is related to only automated posting of results.

Automated Results Posting Tools: Configuration and Usage

LwPostResults6 and PostResults6

This section will walk you through a step-by-step process for using the Automated Results Posting module (PostResults6) within the User Interface (lwPostResults6) to its full potential.

The user interface tool, lwPostResults6, can be executed in one of three ways.

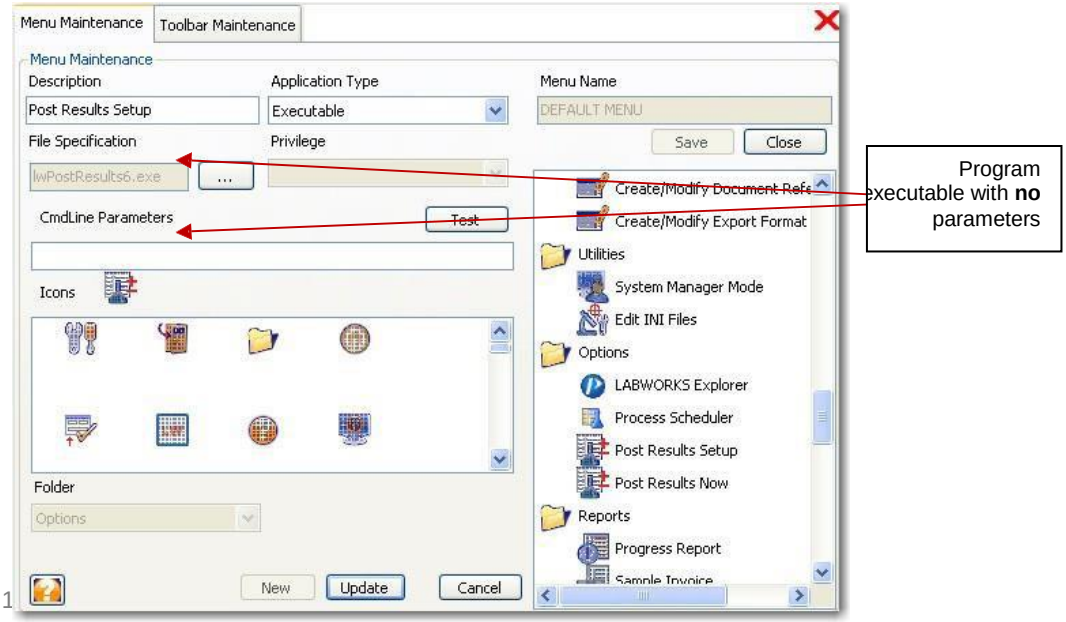
1. Stand-alone execution (without parameters) to launch the Setup/Configuration Wizard.
2. Execution from MultiSample Login (with Login Batch selected and an optional path/filename) to build the input file with all samples and tests. This file will be formatted exactly as defined in the Setup/Configuration Wizard for later importing results. It is designed to be sent off (to an external lab, for example) for data entry of results and/or other related fields. When this file is returned, it will be imported (posted) - see below.
3. Importation (of results) by execution with the parameter “PR:1” to fire off the results posting process (which calls PostResults6 with all configured settings defined during the Setup/Configuration Wizard) and imports all data returned in the input file

We will walk through each of the three uses in the following sections.

lwPostResults6: Stand-Alone Launch – for Setup/Configuration of Process (the "Wizard")

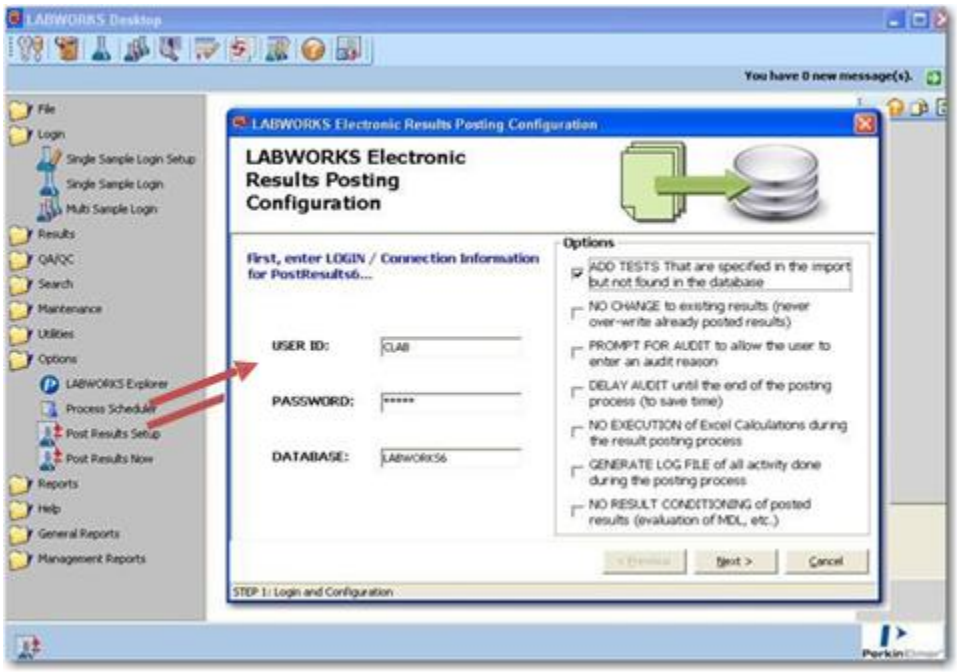
You can add a link on your LABWORKS Desktop for lwPostResults6 (with no command-parameters) in order to launch the Setup/Configuration Wizard whenever needed. This allows you to quickly configure the automated results posting process at any time.

Example Menu Maintenance for a “Post Results Setup” Option



Launching lwPostResults6 without command parameters (CmdLine Parameters) will launch the setup process in the form of a Wizard. We called this menu option Post Results Setup. When finished, save the new Desktop menu configuration and apply it to all users before beginning the steps below.

1. Click the Post Results Setup link in the LABWORKS Desktop. The first “step” in the setup process is displayed.



2. Enter a USER ID, PASSWORD, and DATABASE that will define exactly how the PostResults6 program will log into the database for results posting.

   Note: Generally you would not want to use a specific user’s account in your lab for doing this. We suggest setting up a contract lab or separate user account in LABWORKS specifically for this process to show (within the audit trail) the specific results posted directly from this process separate from any actual user account.

3. There are a number of options on the right-side of this step, select (click) all that apply.

   - **ADD TESTS** Option - This parameter is used to specify that tests can be added later. If the parameter is not specified then it will not allow adding new analysis in the database when specified in the input file.

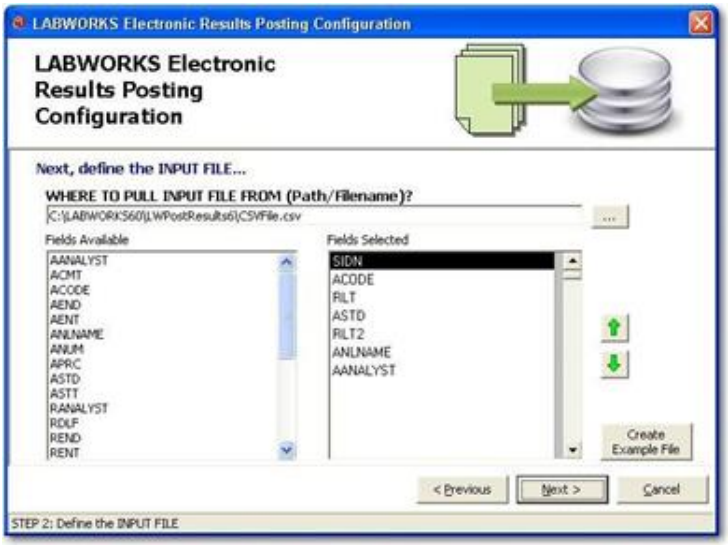
   - **NO CHANGE** Option - This parameter is used to disallow overwriting previously entered results. If this parameter is not specified then overwriting is allowed as default behavior.

   - **PROMPT FOR AUDIT** Option - This parameter is used to specify an audit reason. Audit reasons are specified as the value of the parameter. If this parameter is not specified then it will perform a silent audit with a default reason.

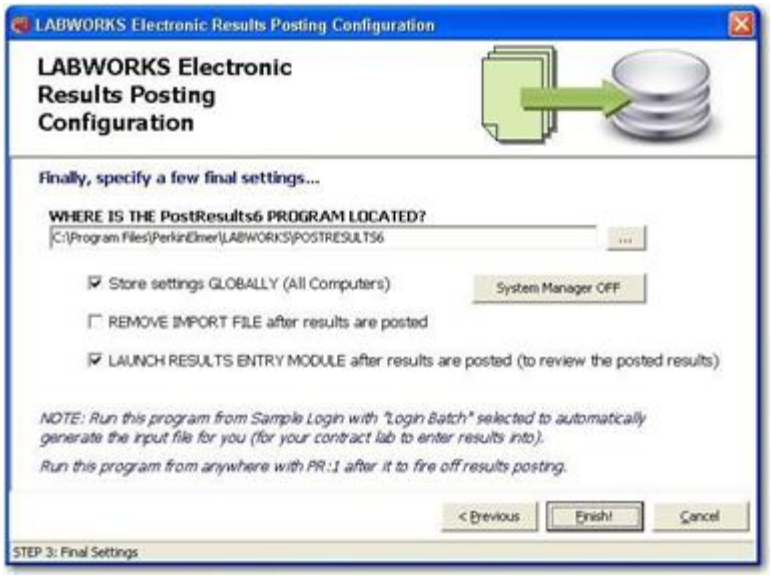
   - **DELAY AUDIT** Option - This parameter is used for saving audit information in a temporary file and posts all this information into the database once all result posting is completed. If the parameter is not specified then it will not save audit information in the temporary file.

- **NO EXECUTION of Excel Calcs** Option - This parameter is used to disallow performing calculations on the results. If this parameter is not specified it will perform the calculations by default. When this parameter is specified no calculations will be done on results.
- **GENERATE LOG FILE** Option - This parameter will create a DAT file showing everything done during the posting process (good for troubleshooting).
- **NO RESULT CONDITIONING** Option - This parameter will cause PostResults6 to skip the result conditioning step completely (PAM, etc.) and post raw results as-is to the database.

4. Click Next> to proceed to the next step. The step to define the input file is displayed.



5. Enter the input file path and filename at the box on the top. PostResults6 will automatically check for the existence of this file each time the results posting process is initiated. If you have an example file already in a specific folder, you can select it to the right (the "…" button).
6. Now define the file contents by selecting (double-clicking) the available fields to the left. The selected fields for the input file are displayed on the right. Use the green arrow keys to move fields up and down to define the order of columns in the file. You can click Create Example File to see what the file will look like after it's created at sample login.
7. Click Next> to move to the final step.



8. Enter the path to the PostResults6 program at the top box.

   Note: the actual .NET PostResults6 program is installed separately for LABWORKS 6.1 installations. By default it is installed in the client's C:\Program Files\PerkinElmer\LABWORKS\PostResults6\ folder. However, if you are running a 6.2 setup with a folder (\LABWORKS62\) that includes the PostResults6.exe file in the same folder as your other 6.2 applications, you still need to enter that path here. Use the "…" button to browse for this program to be sure the path and filename (PostResults6.exe) is correct.

9. Select the final few options below if applicable.

   - Store settings GLOBALLY Option – This parameter, when checked, will store all settings as System Manager keys in the database for all users. If this parameter is not checked, it will store settings locally on the PC (in the Windows registry). If multiple users are posting results, it is a good idea to store the settings globally. If only 1 PC is doing the posting (manually or through a scheduled process, for example), then a local settings storage should be sufficient.

     Note: you will have to initiate System Manager mode (the button to the right of the option) before the system will write any new system manager keys. Before you can globally store these, you must first know the System Manager password.

   - REMOVE IMPORT FILE Option – This parameter, when checked, will remove the import file automatically after results posting is complete.
   - LAUNCH RESULTS ENTRY MODULE Option – This parameter, when checked, will launch the Results Entry module after posting completes to review the results just posted.

10. Click Finish to complete the setup process and store all settings. You are now ready to run the process. Note that you can rerun the setup process at any time to revise these settings.

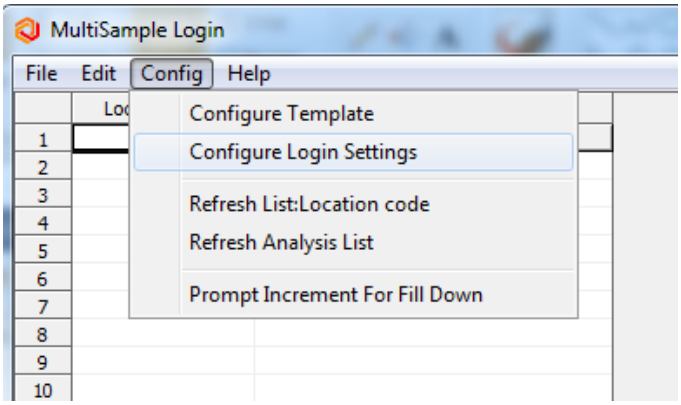
lwPostResults6: Execution from MultiSample Login for Input-File Generation

From within the MultiSample Login module, you can add lwPostResults6 under login settings to generate an input file automatically for you. It will create the CSV file exactly as you defined it in the

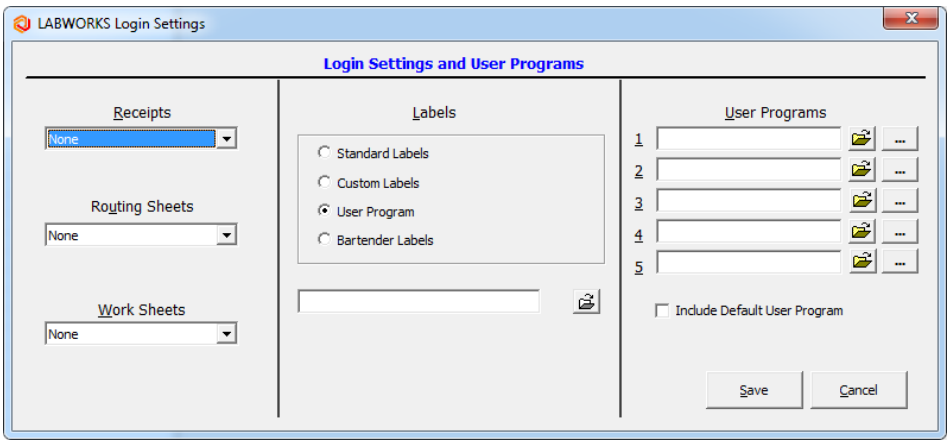
setup process (in the previous section) and enter all the samples and analysis codes into the file for you with default data. You can then send this file to your contract lab for results entry. W hen you receive the file back with results included, you can then post the results (in the next section).

The following steps show the setup in MultiSample Login.

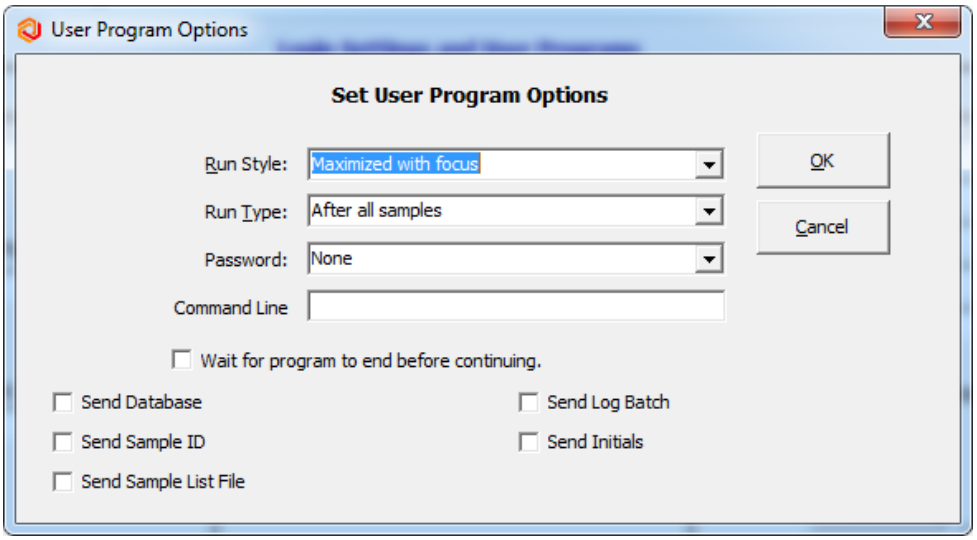
1. Open MultiSample Login and select the Configure Login Settings menu option under the Config menu.



2. On the LABWORKS Login Settings screen, add the lwPostResults6.exe program to the list.



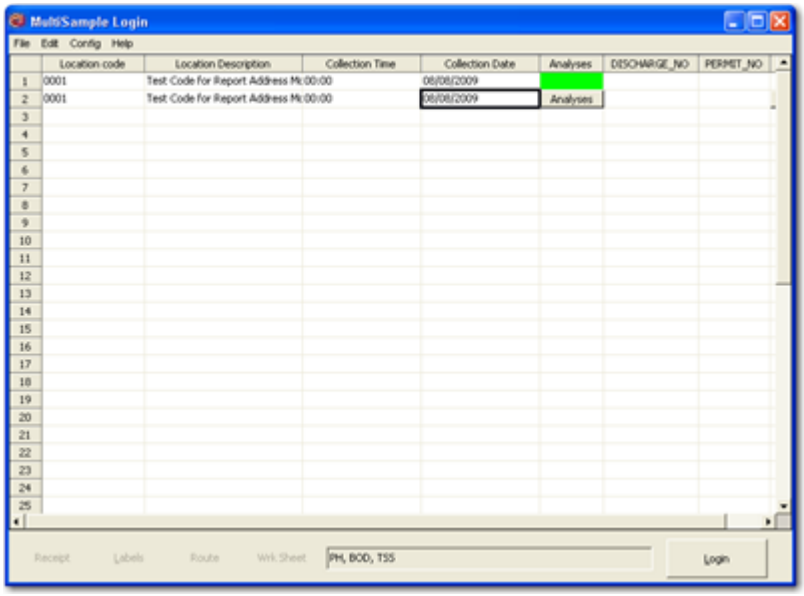
3. Click the ellipse (…) button to the right to configure settings.



Note: make sure you select to Send Log Batch, this is required. It is also recommended that you select to wait for program to end before continuing since it could take some time to generate the file. You can also add an IF: parameter to enter the path and filename of where you want this input file created. This is optional, but without it the system will create a file that is date/time stamped in the user folder. This parameter allows you to control the file's name and where it is generated to be easily picked up and sent to the contract lab.

Note #2: we recommend you keep the file names consistent to reduce confusion. Although not required, it is easier to keep the same filename here that you defined in your Setup/Configuration process in the previous section. If you don't, then when it is returned from the contract lab later, you will need to rename it before PostResults6 will find it during the posting process (next section).

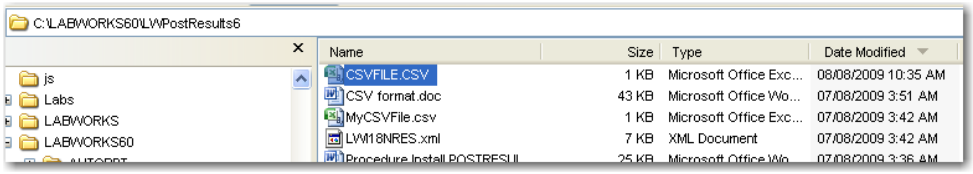
4. Click OK, then SAVE the settings to exit back to MultiSample Login.
5. Now login a few samples.



Note: for the purposes of this example, we are logging in 2 samples with PH, BOD, and TSS

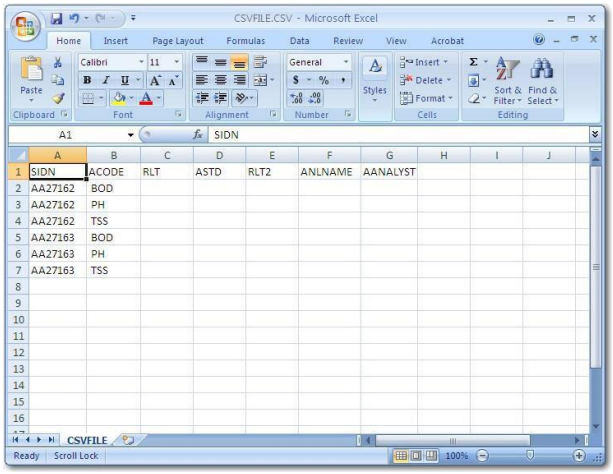
as the tests for each. Once the LOGIN button is clicked, the CSV file should have been created with these samples and tests included… ready to be sent to the contract lab.

6. Check for the creation of the CSV file.



Note: we see it was created successfully exactly where we told it to be saved.

7. We can double-check the contents of this file before sending it off to the contract lab.



Note: the file looks good! Exactly how we defined it in the previous setup process.

Keep in mind if you define the "IF:" parameter it will get replaced each time a login is performed. If you don't define this parameter, CSV files with date/time stamps will accumulate into each user's USER path. It will NOT delete these created files for you. Instead of defining this file, you could instead consolidate all the CSV files from the USER paths and send 1 file at the end of the day, for example.

lwPostResults6: Posting Results

From the LABWORKS Desktop (or a shortcut on the Window Desktop), you can define a link to the lwPostResults6 program with the PR:1 command parameter in order to fire off the results posting process when selected.

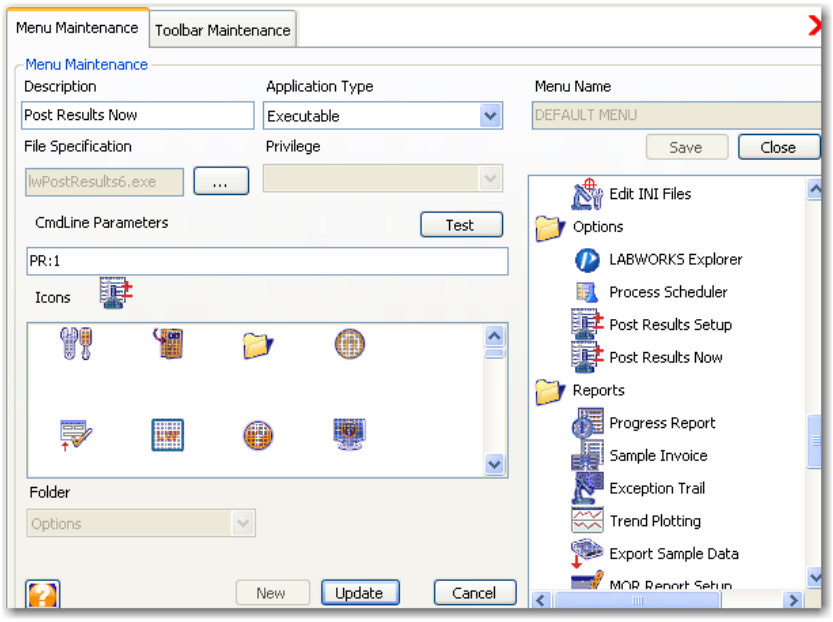
Generally you would ONLY fire off this process after the previous 2 sections have been completed successfully.

1. You have successfully completed the Setup/Configuration process in the previous section
2. You have received a completed INPUT FILE back from the contract lab with results (created from Sample

Login in the previous section) and have placed it in the correct folder where the input file was designated.

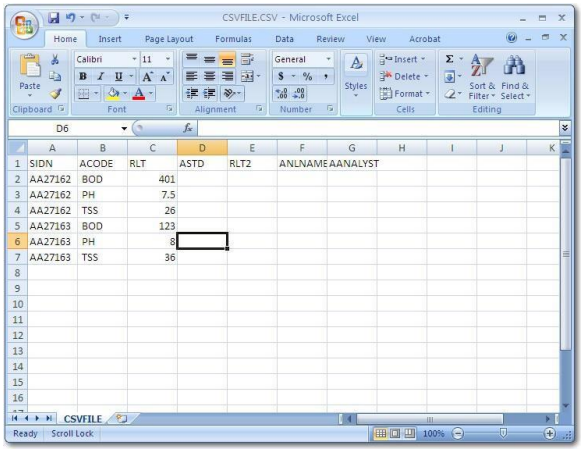
When this is done, and the file is in place and ready for importing, you can fire off this process of results posting.

For the purposes of this example, we added a desktop option called Post Results Now. We placed this in the Options folder below the Post Results Setup. You may want to add these links to the Results Entry section, but that is up to the user. You simply need to add the PR:1 parameter (CmdLine Parameters) to make this link fire off the posting process.



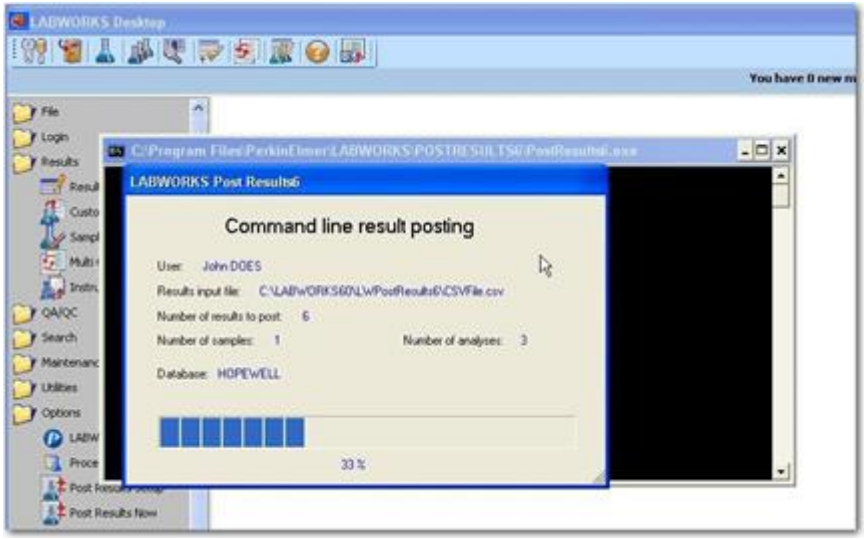
The following will define the process of posting results.

1. First, after you receive your correctly formatted CSV input file back from the contract lab with results, you need to drop it into the folder you defined in the setup process in section 1.

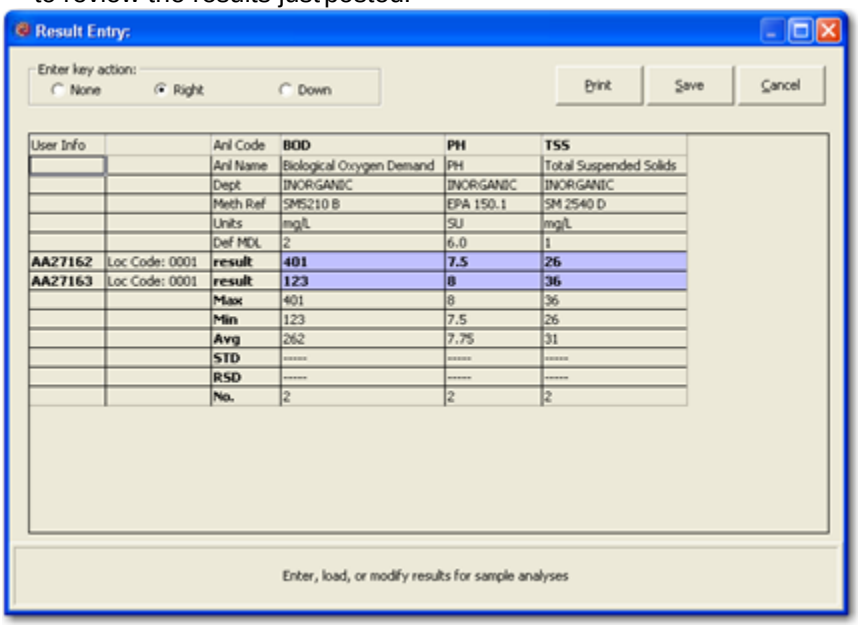


(Note: obviously you don't have to open the file returned if you trust your source. However, it may be a good process to give it a quick check by opening it and browsing the contents before posting just in case someone made some data-entry typos - entered data in a wrong column, for example)

2. Now simply click the Post Results Now link (or whatever you called it) that launches lwPostResults6 with the "PR:1" parameter and watch the process begin. You should see a status box showing the progress.



3. If you selected to launch Results Entry after posting (last step option during the Setup/Configuration process explained previously), then Results Entry will launch allowing you to review the results just posted.



User Info		Anl Code	BOD	PH	TSS
		Anl Name	Biological Oxygen Demand	PH	Total Suspended Solids
		Dept	INORGANIC	INORGANIC	INORGANIC
		Meth Ref	SM5210 B	EPA 150.1	SM 2540 D
		Units	mg/L	SU	mg/L
		Def MDL	2	6.0	1
AA27162	Loc Code: 0001	result	401	7.5	26
AA27163	Loc Code: 0001	result	123	8	36
		Max	401	8	36
		Min	123	7.5	26
		Avg	262	7.75	31
		STD	-----	-----	-----
		RSD	-----	-----	-----
		No.	2	2	2

Note: now you can fix any problems that you notice directly from Results Entry. However, keep in mind that the

AUDIT trail step will now be fired because you are now editing results that have already been posted.

You are allowed to run the process again if any error or problem occurred unless you elected to remove the input file in the Setup/Configuration process. If you selected that option, the input file will be deleted from the folder.

Rerunning the posting process will not hurt anything. It will simply over-write existing results unless you specified the NO CHANGE option in the Setup/Configuration process. The NO CHANGE option will not allow over-writing of already posted results. Also keep in mind that rerunning the process will fire off an AUDIT since you are modifying already posted results.

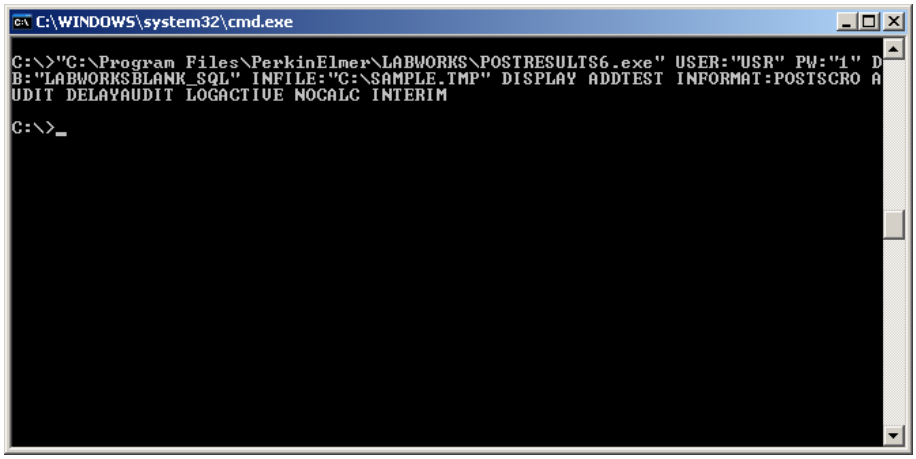
Additional Features of PostResults6 (Outside of lwPostResults6 UI Tool)

As mentioned previously, the PostResults6.exe application is a command-line results posting tool. It does not have to be executed through lwPostResults6, the user interface app, although it is much easier to do for the novice user.

If you wish to run the command-line application itself, PostResults6, and add the parameters to a shortcut (in the LABWORKS Desktop, Process Scheduler, Scan Daemon, etc.), a batch file, or other means, you can.

Keep in mind that the UI tool, lwPostResults6, only works with the new Excel input format. If you want to use PostResults6 for importing legacy formats (from PostSCRO/PostSCR6 or PostMCRO/PostMCR6), then you must use the command-line process.

Command line screen example with parameters



The following are all available command parameters for PostResults6.

Parameter: USER

The USER parameter is used for specifying the LABW ORKS User ID.

It is an optional parameter. If LABW ORKS Session is not running and parameter USER is not specified in the Command

line then the LABW ORKS authentication window will appear and these credentials can be entered. Format for the parameter USER: "LABW ORKS User Id".

Parameter: PW

The PW parameter is used for specifying the LABW ORKS Password.

It is an optional parameter. If LABW ORKS Session is not running and parameter PW is not specified in the Command line then the LABW ORKS authentication window will appear and these credentials can be entered.

Format for the parameter PW : "LABW ORKS Password".

Parameter: DB

The DB parameter is used to specify the target Database name. It is an optional parameter. If the LABW ORKS Session is not running and parameter DB is not specified in the Command line, error will be logged.

Format for the parameter DB: "Valid Target Database Name".

Parameter: ADDTEST

The ADDTEST parameter is used to specify that tests can be added later. It is an optional parameter. If parameter ADDTEST is not specified in the Command line then it will not allow adding new analysis in the database, else when ADDTEST is specified in the Command line it will allow add new analysis in the database.

Parameter: NOCHANGE

The NOCHANGE parameter is used to disallow overwriting previously entered results. If this parameter is not specified on command line then overwriting is allowed as default behavior. It is an optional parameter.

Parameter: AUDIT

The AUDIT parameter is used to specify audit reason. Audit reason is specified as the value of the parameter. It is an optional parameter. If parameter AUDIT is not specified in the Command line then it will perform silent audit with default reason.

Parameter: DELAYAUDIT

The DELAYAUDIT parameter is used for saving audit information in temporary file and posts all these information into the database once all result posting to database is completed. It is an optional parameter. If parameter DELAYAUDIT is not specified on the Command line then it will not save audit information in the temporary file.

Parameter: INFORMAT

The INFORMAT parameter is used to specify the input file mode in POSTSCRO, POSTMCRO and EXCEL format. It is an optional parameter. If input value of INFORMAT is POSTSCRO then it will post a single component result in database. If the input value of INFORMAT is POSTMCRO it will post a multiple component result in database. If input value of INFORMAT is not specified then it will use EXCEL format as default.

Format for the parameter INFORMAT: "POSTSCRO"

Parameter: DISPLAY

The DISPLAY parameter is used to enable displaying the status dialog, as shown in Figure 4. It is an optional parameter. If DISPLAY parameter is not specified on the Command line then it will not display the status dialog screen. W hen parameter DISPLAY is specified on the Command line it will display the status dialog screen.

Parameter: INTERIM

The INTERIM parameter if specified saves the output data to INTERIM.dat file. It is an optional parameter. If parameter

INTERIM is not specified on the Command line, it will not save the output data.

Parameter: INFILE

The INFILE parameter is used to specify the input results file name with path. This parameter is mandatory. Format for the parameter INFILE: "sample.tmp".

Parameter: LOGACTIVE

The LOGACTIVE parameter if specified creates the activity log file. It is an optional parameter. If parameter LOGACTIVE is not specified on the Command line then it will not generate the log file.

Parameter: NOCALC

The NOCALC parameter is used to disallow performing calculations on the results. It is an optional parameter. If parameter NOCALC is not specified on the Command line then it will perform the calculations by default. W hen NOCALC parameter is specified in the Command line then no calculations will be done on results.

After entering all the required parameters execute the command line. Following Result posting screen will appear only if

"DISPLAY" parameters is included in Command line.

INPUT FILE: The Different Input File Types

In the previous section of this chapter, only an Excel input file was discussed. This is the only supported format if used through the lwPostResults6 user tool. If you want to use a legacy format (previous input files used by POSTSCRO or POSTMCRO), then you are required to run PostResults6 from the command-line or from another tool besides lwPostResults6.

Example: POSTSCRO file contents

"AA22336","MYTEN","056","1","1/1/2008","5:30:00","2/2/2008","TOM"
"AA22336","MYtemp","34","2","10/12/2008","16:27","08/11/2010","PETER"

Example: POSTMCRO file contents

AA22337

$MULTITEST

05:31

01/12/2008

06:31

Michael

C:\GCDATA\PE_WINLAB32\STP0818.X

New result loaded from scanned instrument interface

2 kcn

5000

2 magnesium

6500

3

Example: Excel file contents

SIDN	ACODE	RLT	ASTD	RLT2
AA22336	KANAL	6781	16-Dec-08	3
AA02665	KANAL	3	28-Nov-08	2
AA02655	MYANA	4	28-Nov-08	1
AA22105	SINGLE1	5	30-Nov-08	2
AA22107	SQLTEST	645	28-Nov-08	3
AA22337	MYANA	33	23-Dec-08	233

List of all supported header fields for Excel input format

Specifier	Property
SIDN	Sample ID
ACODE	Analysis code
ASTD	Analysis start date
ASTT	Analysis start time (start date must also be passed)
AEND	Analysis end date
AENT	Analysis end time (end date must also be passed)
AANALYST	Analysis analyst (do not default)
APRC	Analysis price
ACMT	Analysis comment (append, not replace)
ANUM	Analysis number
ANLNAME	Analyte name
AVALDATE	Analysis validate date
AVALUSER	Analysis validation user
REPNO	Replicate number (default to 1)
RLT	Analyte primary result
RLT2	Analyte secondary result
RQUAL	Analyte qualifier
RRAW	Raw result
RDLF	Analyte Dilution factor (default to 1)
RMDL	Analyte MDL
RPQL	Analyte PQL
RUNT	Analyte unit
RSTD	Analyte start date
RSTT	Analyte start time (start date must also be passed)
REND	Analyte end date
RENT	Analyte end time (end date must also be passed)
RANALYST	Analyte analyst (do not default)
RNUM	Analyte result number
RRLTSRC	Analyte result source

Summary of the Output for PostResults6

- After execution of the program, the data is posted into target database (ANALYSIS and RESULTS table)

- The Audit trail file generated on DELAYAUDIT parameter is saved as <<InputFileName>>_AuditTrail.xml to User Path.
- All errors are trapped in ERRORLOG table. Pre-authentication errors are saved in ERRORLOG.xml file
- The Log file generated on LOGACTIVE parameter is saved to User Path as POSTRESULTS6.log
- INTERIM.DAT generated on INTERIM parameter is saved at LW DATA\ RESULTS location
- The result of operation in terms of number of success and failure is saved in MESSAGEQUEUE table as DESKTOP type of message

Contents of the error log file generated by PostResults6

If the database name is not entered correctly at the Command line then following error message will occur. Example: Contents of error log file is shown below.

```
<LWERROR>
 <ERROR>
  <ErrorDescription>The requested database LABWORKSB is not defined in configuration.</ErrorDescription>
  <ErrorDateTime>12/12/2008 12:40:31 PM</ErrorDateTime>
  <SecurityLevel>HIGH</SecurityLevel>
  <LogType>SECURITY</LogType>
  <UserID>0</UserID>
  <UserInitials>
  </UserInitials>
  <AssemblyName>Microsoft.Practices.EnterpriseLibrary.Data, Version=2.0.0.0, Culture=neutral,
PublicKeyToken=null</AssemblyName>
 <AssemblyVersion>2.0.0.0</AssemblyVersion>
 <ApplicationName>Microsoft.Practices.EnterpriseLibrary.Data</ApplicationName>
 <ClassName>DatabaseConfigurationView</ClassName>
 <MethodName>
 </MethodName>
 <WorkStationID>SVR006</WorkStationID>
</ERROR>
<ERROR>
 <ErrorDescription>Authentication     failed!</ErrorDescription>
 <ErrorDateTime>12/12/2008 12:40:31 PM</ErrorDateTime>
 <SecurityLevel>MEDIUM</SecurityLevel>
 <LogType>APPLICATION</LogType>
 <UserID>0</UserID>
 <UserInitials>
 </UserInitials>
 <AssemblyName>
 </AssemblyName>
 <AssemblyVersion>
 </AssemblyVersion>
 <ApplicationName>POSTRESULTS6</ApplicationName>
 <ClassName>Program</ClassName>
 <MethodName>Main</MethodName>
 <WorkStationID>SVR006</WorkStationID>
</ERROR>
```

Contents of the Process Log file Generated by PostResults6

The Log file displays the UserID, database name, Log datetime, result input file path, number of results posted, number of samples posted, number of analyses posted, number of results posted successfully, and number of results posted failure after the posting process has completed. This file is simply a general process log that can be used to troubleshoot any problems that happened during a posting.

Example: Contents of log file is shown below.

User : USR
Database name : LABWORKSBLANK
Log datetime : 12/10/2008 7:42:43 PM
Result input file path : C:\L1342139_POSTMCRO.TMP
Number of result to be posted : 3
Number of samples posted : 1
Number of analyses posted : 3
Number of results posted successfully : 3
Number of results posted failure : 0

Contents of INTERIM.dat file Created by PostResults6

This optional DAT file is used as an interim step to display the formatted data prior to database storage.

Example :Contents of INTERIM.dat file is shown below.

"AA22336","GR1","","09/12/2008","14:43:10","09/12/2008","14:43:10","10/12/2008","03:48:34","TA1","TA1","056660","1","SUNIL","NONE" .

Contents of Message Queue after PostResults6 Posting Process

PostResults6 will write data to the message queue displaying the number of results posted successfully and number of failures if any. W hen users log into the LABW ORKS Desktop, they will receive this message (if elected to) showing that results were just posted.

Example: Contents of message queue is shown below.

POSTRESULTS6: C:\SAMPLE.TMP.Number of results posted successfully: 1 and number of failures: 0

The AUDIT Step: Automated Results Posting

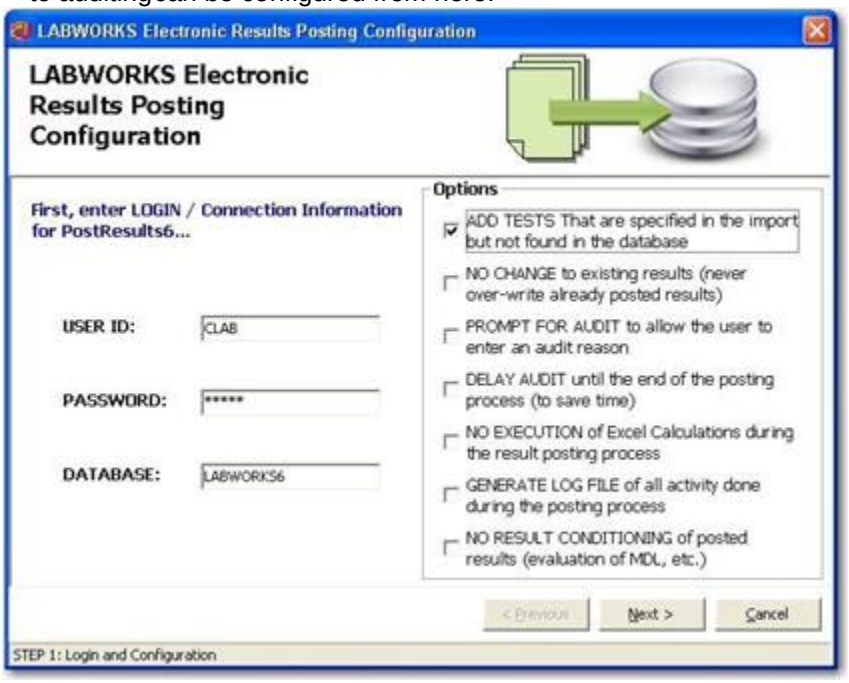
Posting to the audit trail happens whenever existing results change. Unless you have your audit trail turned off in your System Manager key settings, it will always post an audit record showing what changed, the date/time, who changed it (posted the result file), and why (a reason). A generic message for the reason will be posted to the audit trail that will notify the user that the change happened during an automated result posting process (by the user account you defined within the **Setup/Configuration** of **lwPostResults6** or the user account entered in the **command-parameters** of **PostResults6** itself). You are allowed to enter a reason using the “AUDIT” parameter.

As mentioned in the introductory section, we suggest adding a separate user account to the system to easily distinguish between results posted by the automated process (PostResults6) or by actual “real” users. Some people do this by entering a user account with their contract lab’s name, for example.

AUDIT User Setup: from lwPostResults6

After adding a new user account specifically for automated results posting (we will call this new user **CLAB** for our testing purposes). Proceed to the following steps.

1. Select the **Post Results Setup** link (or whatever you called it) from the LABW ORKS Desktop that launches the lwPostResults6 application with no command-parameters (or run it stand-alone from the programs path). The initial screen of the W izard is launched. Everything related to auditingcan be configured from here.



2. Enter the contract lab user account and password on the left along with the name of your database (which should automatically default within the box to the database currently logged into).

3. Set any AUDIT-Specific parameters accordingly.

- **NO CHANGE** Option - This parameter is used to disallow overwriting previously entered results. If this parameter is **NOT** checked then overwriting is allowed as default behavior along with auditing. If you check this option, no auditing will ever take place because no existing results will ever be touched.

- **PROMPT FOR AUDIT** Option - This parameter is used to specify an audit reason. Audit reasons are specified as the value of the parameter that is prompted to the user. If this parameter is not specified then it will perform a silent audit with a default reason.

- **DELAY AUDIT** Option - This parameter is used for saving audit information in a temporary file and posts
  all this information into the database once all result posting has completed. If the parameter is not specified then it will not save audit information in the temporary file.

4. Click **Next**, keep settings as defined in **Chapter II**, then **Finish** to store them.

Now the next time the process is fired for posting results, your audited values (if exists) can be viewed from within the screen for displaying audit records (from the Desktop). You can search for audit records specifically from the contract lab user account entered (**CLAB** in this example) to see all audit records from only this process.

AUDIT User Setup: from PostResults6 Console (Command-Level)

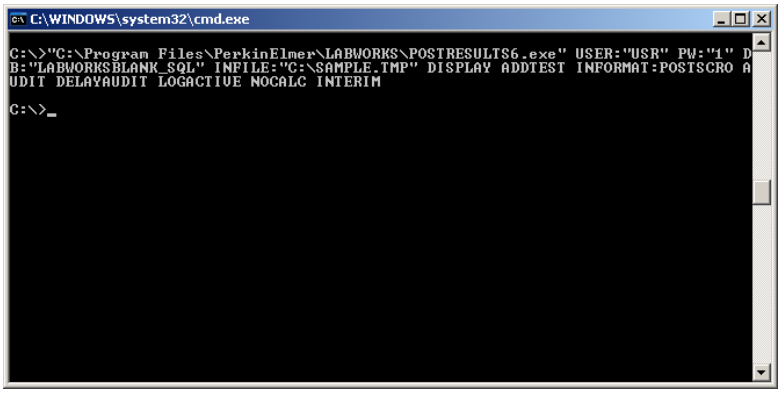
After adding a new user account specifically for automated results posting. The PostResults6.exe console application can be launched with the same parameters as defined in the previous section.

Define the user account that will be audited during the posting process using the **USER:** and **PW:** parameters.

Exactly as described in the previous section, the **AUDIT** parameter, if included within the command parameter string, will cause a prompt to be displayed to the user. This will obviously break the automation part, so if this tool is scheduled within another application for silent posting of results that parameter needs to be omitted.

Also as mentioned previously, the **DELAYAUDIT** parameter can also be added within the command parameter string to save audit information in a temp file before posting to the database (quicker usually).

The **NOCHANGE** parameter, if included in the command parameter string, will not allow changes to existing (already posted) results. If you add this within the command parameter string, it will not fire off any audit because no modification of existing results will be allowed.



The Result Conditioning Step: Automated Results Posting

Introduction: Business Rules Component

Numeric value rounding and result conditioning are separate but closely related overlapping areas of LIMS functionality. As already explained previously, result values generally make their way into LABW ORKS through one of a few different methods: from manual entry, by calculation, by importation from an instrument interface, or by posting from a command line routine through an externally provided file. New results, regardless of their origin need to be evaluated for possible result conditioning or rounding. As with specification checking and calculation triggering, rounding and conditioning need to be applied in a uniform and reliable manner in all applications that generate or post results, whether generated via the web or W indows.

In regulated laboratory settings, rounding rules can be very specific and strenuously enforced. Mandated analytical protocols may specify rounding rules that vary from analyte to analyte and even within concentration levels of the analyte. Legacy result conditioning and rounding functionality has been implemented through a variety of add-on “user” programs and application code that was never uniformly applied and not well understood, leading to high potential for incorrect behavior and results. Tools in LABW ORKS 6.2 have since been designed to address this problem.

To provide uniform rounding and result conditioning functionality to all LABW ORKS applications, a new business rules component was developed specifically for calculations and rounding.

To form an understanding of the new component for results conditioning and how LABW ORKS evolved from a legacy standpoint, the various types of result conditioning and rounding functionality present in legacy LABW ORKS applications will be described and discussed followed by an enumeration of problems, limitations, and complaints associated with that functionality.

Abbreviations and Terminologies

The following abbreviations appear throughout this chapter. Some of these terms have already been discussed previously.

PAM: Project Analyte Management

SF: Significant figures

MDL: Minimum detection limit

BDL: Below detection limit

ND: Not detected

RL: Reporting limit

MRL: Minimum reportable level

Although this section is not intended to cover any of these terms individually, the following is a description to understand what each are and how they are used.

Project Analyte Management (PAM): A result conditioning scheme which strips component analytes that are not of interest from samples and then replaces numeric results that fall below specified "reporting limits" with preconfigured text values. PAM has the highest precedence of all result conditioning and rounding functionality in legacy LABW ORKS.

PAM functionality has been implemented via special code in LABWORKS results entry and multicomponent result transfer applications.

Significant Figure Range Rounding (SF): A result conditioning scheme in which rules are implemented on an analyte specific basis for dealing with results falling in specific numeric ranges. A text string can be specified for replacing primary result values falling numerically below a certain MRL (minimum reportable level). Rounding rules can be specified for up to 10 numeric ranges (higher than the MRL). Those rounding rules can specify the number of significant figures to retain or a range interval to which to round off to the closest one of. SF Range rounding is implemented in legacy LABW ORKS as a post results entry "user" program, SFRound6.exe. This program is launched by command line and must be passed a file containing the sample numbers and analysis codes of assigned analyses with results to review for SF rounding.

**** IMPORTANT NOTE ON RANGES:** Keep in mind when entering these ranges that order matters. The first statement that evaluates to TRUE causes the system to refrain from continuing its evaluation. For example, if you are evaluating a result that is "< 100" after evaluating between "1 and 10", then the result will never make it to the second statement if the result is 5 because it is already evaluated as TRUE.

Odd-to-Even Rounding: A rounding method, also sometimes called "round-to-even", intended to yield perfectly neutral results over time that display neither positive nor negative bias. Use of this rounding method is often required by agencies regulating laboratories and bodies that promulgate methods for performing analyses and handling data. This rounding is conducted in the following manner:

- Decide which digit is the last (least significant) to keep.
- Increase it by 1 if the next digit (most significant digit to discard) is 6 or more, or a 5 followed by one or more non-zero digits.
- Leave it the same if the next digit is 4 or less
- Otherwise, if all that follows the last digit is a 5 and possibly trailing zeroes; then change the last digit to the nearest even digit. That is, increase the rounded digit if it is currently odd; leave it if it is already even.

All rounding that is currently performed in LABW ORKS uses this method.

"Below Detection Limit" and "Not Detected" Result Conditioning: When results are imported from instruments they sometimes have a positive value that falls below the MDL of the particular analyte. In that range, the analyte is assumed to be present but determining its exact amount is deemed impossible. In such cases the analyte is said to be "below detection limits" and a replacement of the numeric value with some special text may be required.

When the results reported for an analyte are 0 (zero) or some negative number or, alternatively, the instrument reports nothing at all for an analyte that the instrument method was capable of measuring, the analyte is said to be "not detected". In this case also, some special text may be required to report the result for the analyte.

The text to be inserted (substituted for the instrument numeric result) as the analyte result in not detected and below detection limit cases are currently specified by settings in the [Spread_Result_Entry] section of LABW ORKS.INI (see example below) for use in legacy

applications.

```
[Spread_Result_Entry]
    'specifies result labels to be used for 0 and below MDL in single component instrument imports
    SCND = Not detected
    SCBDL = Present < MDL

    'specifies result labels to be used for 0 and below MDL in multicomponent instrument imports
    MCND = Not detected
    MCBDL = Present < MDL
[EndSpread_Result_Entry]
```

Note: The settings that appear above are used by Mcxfer6.exe, ScanDaemon6.exe, and Postmcr6.exe as well as
ResultsEntry6.exe and potentially other legacy tools and user programs.

“Standard Results” Substitution: A process that replaces abbreviations from one to four characters entered or imported into results fields with text looked up from a pick list (RLTCODES) in the database. The lookup and replacement occurs immediately upon leaving the field containing the abbreviated entry within Results Entry (when manually entering results). For automated posting, the string is inserted after the audit step and before calculations are executed. If a field is text (non numeric), calculations are aborted in the following step in the process.

This feature was created to make it easier and quicker to manually enter commonly used text result values in “standard”
format. A few typical examples are:

Abbreviation	Standard Result
+	Positive p Present
nd	Not detected d Done
<	Below MDL

Previous (Legacy) Rounding Processes

Rounding and result conditioning processes in legacy applications are carried out by a mixture of application code and post run programs. Behavior differs depending upon whether the results are imported, calculated, or manually entered and whether the results are for single component analyses or analytes of multicomponent analyses. This lack of consistency has been the source of many user complaints and has consumed lots of sustaining engineering time.

LABWORKS 6.2+ Rounding/Conditioning Engine

The following were **requirements** of the new LABW ORKS business rules component that handled all results rounding and conditioning.

- Preserve as much current functionality as possible while achieving a cleaner more logical design.
- Do not change or complicate existing data structures any more than absolutely necessary and simplify them if possible.
- Simplify configuration and specification of rounding behavior with minimal changes to

existing UI.

- Move away from INI file settings for specifying result conditioning and rounding behavior.
- Provide a choice of number of significant figures rounding, number of decimal places rounding, or no rounding at all on an analyte by analyte (not analysis) basis.
- Perform substitutions for standard result abbreviations.
- Perform text replacement when specified for result quantities of zero or less, less then detection limits, and less that PAM reporting limits.
- Perform magnitude (range) dependent significant figures rounding when specified.
- Yield identical rounding results in all applications and on all supported platforms.
- Perform rounding in real time, providing immediate feedback to interactive users.
- Retain all unrounded (or prerounded) result values for later inspection and reporting.
- Provide granular control to apply (or not apply) rounding to each of the following types of results:
    - Results imported from instruments
    - Calculated results
    - Manually entered results
    - Results posted at login time
- Do not depend upon user programs for performing rounding.

Business Component Approach

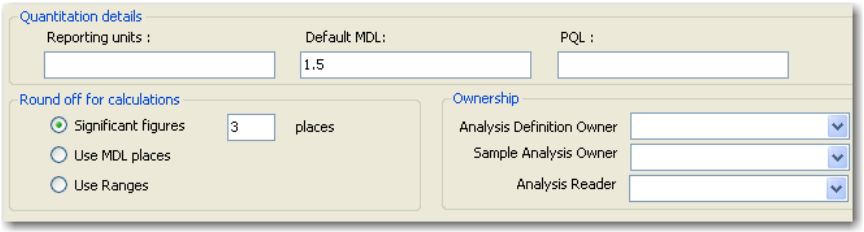
The approach to be applied for result conditioning and rounding in LABW ORKS 6.2+ will consist of specifying all rounding behavior within the LABW ORKS database (System Manager settings, Analysis List Maintenance, etc.) instead of configuration files as required in previous legacy modules. The internal XML-based LABW ORKS tool that handles this, controlled by the LW PlatformService W indows service, is called the SIMC, Sample Information Management Component.

Note: All SIMC component analytes possess mean average and first replicate result parts by definition. In cases where more than one replicate measurement is specified for an SIMC Component Analyte, calls to the engine will be made only when all replicate results have been entered or imported.

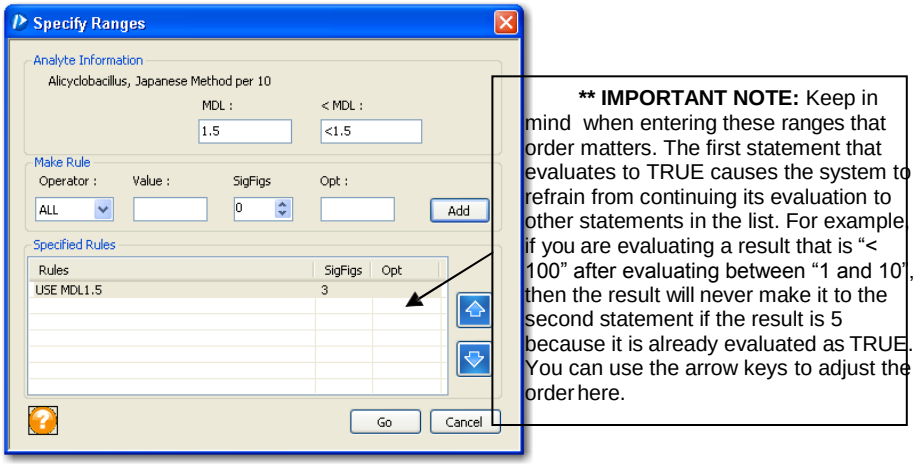
Three levels of rounding behavior specification will consist of hard coded default behavior that can be overridden by global rules and policies which, in turn, may be overridden by special case exceptions to global policies.

All global rounding policies are stored in System Manager settings. Rounding settings that are for specific analytes are stored in table **SFRANGES** and set using the Analysis List Maintenance application.

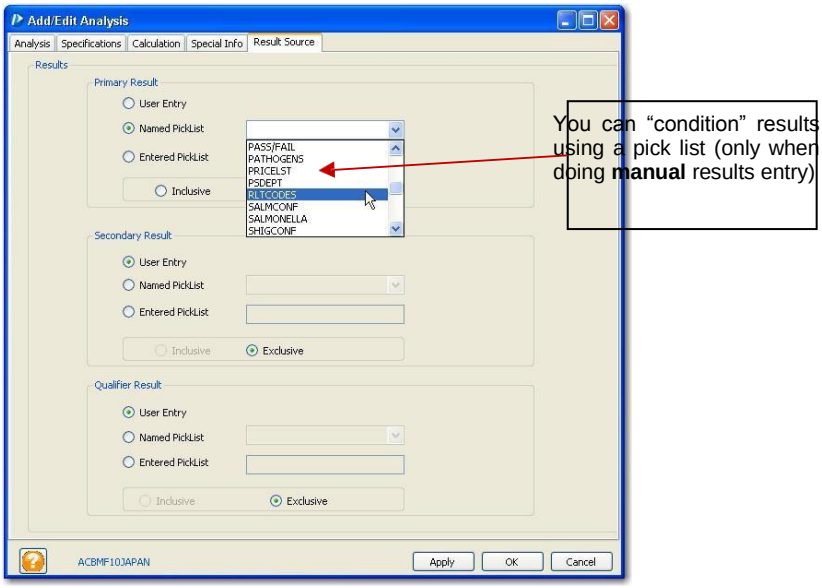
Opening any analysis code within Analysis List Maintenance will display the rounding information at the bottom of the first tab. The user is then allowed to adjust whether they want rounding for this analyte by **significant figures**, by places that match the assigned **MDL**, or defined by custom **ranges**.



If **Use Ranges** is selected, the user will have several options to define how the result for this analysis code is rounded.



To define the conditioned values selectable at Result Entry time (text values or whatever), the user can select the **Result Source** tab. However, any pick list assigned will not be available during any Automated Posting process.



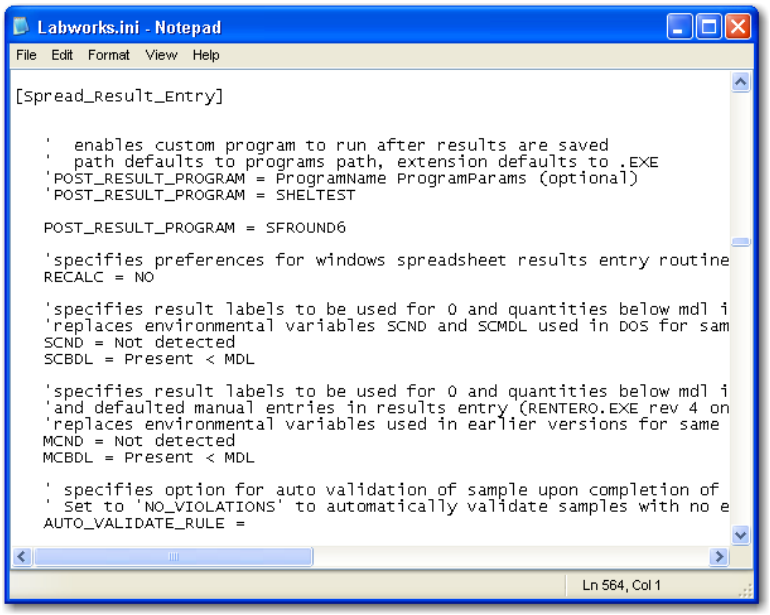
The default is no rounding for all analytes initially.

A new SIMC component field, RawResult**,** is used to store all results of any origin before rounding is applied to yield the(primary) Result. If no rounding is specified, the primary and raw results will be identical.

Rounding will no longer be implemented by “user” add-on programs.

Legacy SFROUNDO / SFROUND6: Where Does it Fit?

Support for this legacy tool was not removed although in LABW ORKS 6.2 it is no longer required. The legacy usage of this tool was for the assignment of rounding rules / ranges within a text file (SFRanges.txt) located within the LABW ORKS Data path. The user could then add SFROUNDO.EXE (for version 5.8 or prior) or SFROUND6.EXE (for versions 6.0 – 6.4) in the LABWORKS.INI’s [Spread_Result_Entry] section as a defined post-results entry program (using the POST_RESULT_PROGRAM parameter). It will then cause the program to fire after results entry (manual posting) and any ranges text file found will be implemented on the results entered.



If a LABWORKS user implements version 6.2 and wants to continue using text files that define these rounding ranges, that’s fine. SFROUND6.EXE will continue to work. Otherwise, the user has the option of getting rid of the ranges text file, removing the reference to SFROUND6.EXE in the LABW ORKS.INI, then simply enter these rounding ranges within the individual Analysis List Maintenance application. The choice is the user’s to determine. Either way is supported. However, the original legacy module used for 5.8 and previous versions, SFROUNDO.EXE is **not** supported. The user will, as a minimum, need to replace the SFROUNDO program reference in their LABW ORKS.INI with the newer SFROUND6 program before any of the previous legacy ranges defined in the SFRanges.txt file will work.

Keep in mind that SFROUND6, although supported, is only launched from legacy tools. In order to use automated results posting (PostResults6), the rounding information in the SFRanges.txt text file must be entered into the individual analysis codes in Analysis List Maintenance instead. This information must be in the LABW ORKS database, not text files, for the automated posting to utilize it.

Global Rounding System Manager Settings

System **global rounding defaults** is governed by a set of new system manager settings. Those settings are defined in the following list.

Setting Name: DEFAULT_ROUNDING_METHOD
Type: Text
Valid Values: N (none) / S (significant figures) / D (decimal places)
Comment: Default (hard coded is None)

Setting Name: DEFAULT_ROUNDING_STYLE
Type: Text

Valid Values: S (simple) / E (round to even -default)
Comment: The choice of “simple” rounding where 5 is always rounded up and “round-to-even” rounding is selectable on a system global basis.

Setting Name: DEFAULT_ROUNDING_SIGFIGS
Type: Integer
Valid Values: 1 – 10
Comment: Applies only if DEFAULT_ROUNDING_METHOD = S

Setting Name: DEFAULT_ROUNDING_ DECIMALS
Type: Integer
Valid Values: 0 – 10
Comment: Applies only if DEFAULT_ROUNDING_METHOD = D

Setting Name: DEFAULT_REPLACE_BDL
Type: Boolean
Valid Values: True (do replace number < MDL with value of DEFAULT_BDL_REPLACE_VALUE) / False (do not replace numbers < MDL)

Setting Name: DEFAULT_BDL_REPLACE_VALUE
Type: Text
Valid Values: Literal text string up to 14 characters long
Comment: Text to replace numeric result falling between 0 and the minimum detection limit. Run time replacement variables can be embedded in literal text. Applies only if DEFAULT_REPLACE_BDL = True
Example: “Less than %MDL%”

Setting Name: DEFAULT_REPLACE_ND
Type: Boolean
Valid Values: True (do replace number <= 0 with value of DEFAULT_ND_REPLACE_VALUE) / False (do not replace numbers <= 0)

Setting Name: DEFAULT_ND_REPLACE_VALUE
Type: Text
Valid Values: Literal text string up to 14 characters long
Comment: Text to replace numeric result falling at or below 0. Run time replacement variables can be embedded in literal text. Applies only if DEFAULT_REPLACE_ND = True
Example: “Less than %MDL%”

Setting Name: ROUND_MANUAL_RESULTS
Type: Boolean
Valid Values: True (do round off manually entered results) / False (do not round off manually entered results)
Comment: This functionality was only available via identity calculation definition in legacy.

Setting Name: ROUND_CALCULATED _RESULTS
Type: Boolean
Valid Values: True (do round off results of calculations) / False (do not round off results of calculations)

Setting Name: ROUND_IMPORTED _RESULTS
Type: Boolean
Valid Values: True (do round off results imported from instruments) / False (do not round off results imported from instruments)

Setting Name: ROUND_LOGGED _RESULTS
Type: Boolean
Valid Values: True (do round off results posted at login time) / False (do not round off results posted at login time)

Comment: New command line application for logging in samples will allow results to be posted too.

Setting Name: ROUND_SPECIFIED_RESULTS
Type: Boolean
Valid Values: True (do round off results filled from location specifications) / False (do not round off results filled from location specifications)
Comment: Default results for filling at result entry time can be optionally specified as a type of user defined "specification" in the 6.1 location list maintenance application.

All "global" system rounding settings will be overridden for "specific" analytes settings defined within Analysis List
Maintenance using the "Specify Ranges" dialog.

Results Substitution and PAM Specifics

Standard results substitution serves to provide shortcuts and standardization for manually entering non-numeric result values. The lookup and replacement of manually entered abbreviations from pick list **RLTCODES** takes place at the initial result-entry level and is performed by the background process engine before any other rounding or result conditioning is attempted. Lookup and replacement of abbreviations with "standard" results will be **restricted to manually entered results only**. In other words, during the automated results posting process, PostResults6 will not evaluate an imported result of "ND" as Not Detected". Calls to the process engine to perform this substitution will be made as soon as any manual entry has been made in a results entry UI control.

Project analyte management (**PAM**) is separate from but overlaps with rounding issues. As mentioned previously, PAM restricts which component analytes of an analysis can have results recorded, how (which text label is used for) results that are numerically below the specified reporting limits for the assigned project (PAM group), and how qualifiers are assigned for three possible analyte result classifications (below MDL, below reporting limit, at or above reporting limit). PAM functionality related to this process works as follows.

- PAM groups and analyte settings continue to be created and modified with the PAM configuration utility.
- Because PAM is project specific (affects certain samples only), **PAM result conditioning** (for any analytes of PAM assigned analyses) **will trump all other rounding settings.** No further conditioning will ever be attempted after PAM has been applied to results for an analyte.
- PAM result conditioning and qualifier assignment is performed by the backend process engine at result entry time.
- PAM analyte filtering (removal of component analytes from analyses) is applied at the time of analysis assignment to samples.
- SIMC assigned analyses has a **PAMLimits** property that is set if the analysis is controlled as part of a PAM group assignment to a sample. If the **PAMLimits** property of the assigned analysis is set, only SIMC analytes will appear under the assigned analysis.
- Each analyte that is under PAM control will have properties set that specify how the result will be conditioned and how qualifiers shall be assigned.

LABWORKS Rounding Methods

The result conditioning step within the process of results posting shall expose the following public methods.

PerformRoundingForAnalyte: Performs result conditioning, rounding, and PAM operations on a passed SIMC Component Analyte object. Result conditioning is performed on only the Mean Average result part (SIMC Component Result object) of the component analyte. This method is called every time a Component Result for a replicate is entered or modified but no operations are performed until values exist for all specified replicates.

This method is passed a **SIMC Component Analyte object** and it returns the same object either modified or not.

Note: The term “Component Analyte” refers to the lone analyte of a single component assigned analysis or to any one of the analytes of a multicomponent analysis. At the SIMC level, all component analytes are considered to be equivalent, regardless of the type of analysis that contains them.

PerformStandardResultSubstitution: Checks a replicate result to determine if a substitution is specified (in pick list RLTCODES). If so, the substitution is performed. This method is called every time a replicate result is manually entered or modified.

This method is passed a **SIMC Component Result** object and it returns the same object either modified or not.

The Calculation Step: Automated Results Posting

Introduction

LABWORKS has provided two types of configurable calculations since its earliest versions, **"built-in" calculations** and **Excel calculations**.

When samples have analyses that are included within calculation definitions and are loaded into the results entry application or automatically posted, the calculations are automatically activated without the need to set any specific properties of the sample or the application.

LABW ORKS calculations are now performed within a **calculation engine** in order to extend calculation capabilities to web based applications as well as those running on W indows workstations.

These two basic types of LABW ORKS calculations, built-in and Excel, are utilized within the 3rd step defined within the "results posting" process immediately following results conditioning. Both of these methods are discussed in the following sections.

Terminology

Trigger: Every calculation defined in LABW ORKS has one or more analyses specified that provide input values for the calculations to be performed. These analyses are referred to as "triggers" because entering or changing their result values may cause the calculation to "fire" (be performed).

Target: Every calculation defined in LABW ORKS has at least one analysis specified that receives the output (or result) of the calculation. Analyses receiving the results of calculations are referred to as calculation "targets".

Fire: When a result value is entered for the last pending trigger analysis of a calculation or the result of one trigger analysis is changed after all triggers have values, the calculation is "fired" (or performed).

Cascading Calculation: An analysis specified as the target of one calculation may itself be a trigger for one or more other calculations. When the output of one calculation fires additional calculations, the additional calculation is referred to as a "cascading calculation".

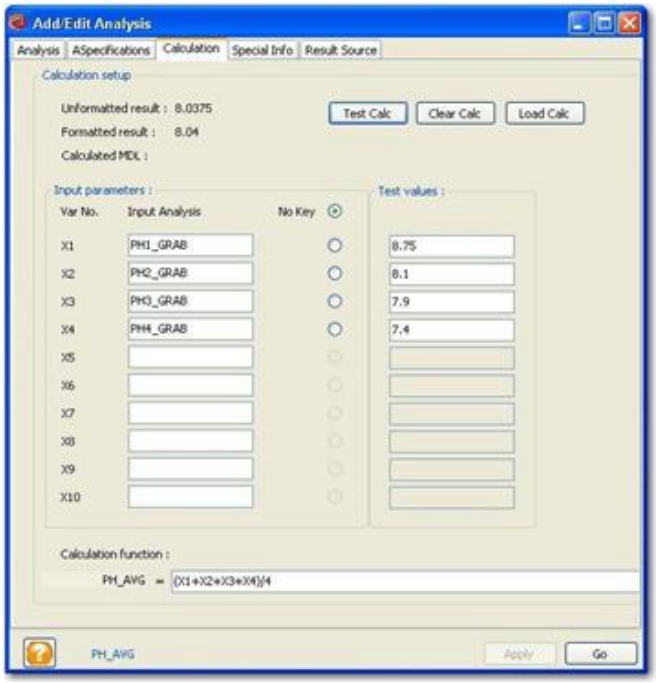
Built-In Calculations

Built-in calculations are performed by LABW ORKS code without the need for any external tool. Legacy built-in calculations may have from one to ten trigger analyses specified whose primary result values form the inputs to the calculation. Built-in calculations always have exactly one target analysis specified, whose primary result will be the output of the calculation.

Built-in calculations are configured in the **Analysis List Maintenance** application (under the **Calculation** tab of an opened analysis code record). The analysis is defined by specifying the trigger analyses (**inputs**) and an algebraic formula relating the triggers to the target (**output**).

A built-in calculation can be fired only when all of its triggers have primary result values assigned. This is true for both manual and automated result posting processes.

Any calculation that is entered for an analysis code within Analysis List Maintenance can be tested as well right from the **Calculation** tab by entering example input and output parameters. Since LABW ORKS 6.2+ does these calculations through a calculation engine (not in separate applications), if a calculation performs properly within this test screen, it should also perform the same throughout LABW ORKS.



After a calculation is performed, its result is rounded according to specified criteria then checked for exceptions. If the calculated analysis is itself configured as a trigger for one or more other calculations, those calculations are also triggered, resulting in cascading" calculations. Calculations continue until no calculation output changes the value of any trigger.

Data Layouts for Built-In Calculations: The definitions of built-in calculations are stored in table **CALCDEFS**, whose primary key is the analysis code of the target analysis. This restricts any LABW ORKS analysis to having no more than one built-in calculation defined for it. That table also contains columns for identifying up to 10 input trigger analyses, a column for specifying which input (if any) is the key input, a column specifying whether or not to calculate the MDL of the target analysis along with its primary result, and a column specifying how to handle nonnumeric key input values.

Table **CALDEFS** is queried by the legacy results entry program to determine which analyses in results entry grid are triggers for calculations and which are targets of calculations so that flags can be set to launch calculations as trigger values are entered or modified.

Key Input Analysis: One of the input analyses for a calculation can be optionally designated as the "key input". A key input is special because:

- Its start date-time, ending date-time, and analyst ID can be copied to the calculated analyte. If no key input is designated, the current clock time is assigned to the start and

ending date-times of the calculated analyte and the current LABW ORKS user ID is assigned to the analyst ID property of the calculated analyte.

- Options can be specified for how to perform the calculation if the key input value is non-numeric. Those options are:
    - Do not perform the calculation leaving the result value empty
    - Do not perform the calculation but assign the value of the key input to the calculation result.
      Example: The value of the key input result was “Not detected”. The calculated result is also set
      to “Not detected”.
    - Do not perform the calculation but assign the MDL value of the calculated output as its primary result. Example: The value of the key input result was “Not detected” and its MDL value was “0.10”. The calculated result is set to “0.10”.
    - Do not perform the calculation but assign < MDL value of the calculated output as its primary result. Example: The value of the key input result was “Not detected” and its MDL value was “0.10”. The calculated result is set to “< 0.10”.
    - Do not perform the calculation but assign a value of zero to the output as its primary result.
      Example: The value of the key input result was “Not detected”. The calculated result is set to
      “0”.
    - Perform the calculation substituting a value of zero for the key input result. Example: The value of the key input result was “Not detected”. The calculation is performed using 0 as the key input result.

An MDL value can also be used to recalculate the MDL of the calculated analyte by applying the same formula and other input values as was used for calculating the result. In this optional calculation, the initial value of the MDL for the key analyte is substituted for the primary result value in the calculation inputs, leaving all other inputs unchanged. After the calculation is performed, the result of the calculation is used as the “corrected” MDL of the calculated analyte.

Calculating Multicomponent Results: The results of both single and multicomponent analyses can be generated by built-in calculations. To calculate multicomponent results, at least one of the trigger analyses must also be a multicomponent analysis that has some component analytes in common with the target analysis. A calculation is attempted for each component analyte of the target analysis. That calculation for the analyte is completed only if result values are present for that analyte in every multicomponent trigger analysis.

For example:

- Assume we want to calculate results for a multicomponent analysis, **$TESTD.**
- Assume that $TESTD has 10 defined component analytes, Analyte01, Analyte02, …, Analyte10.
- Assume that the calculation has one single component trigger, **TESTB**, and two multicomponent triggers, **$TESTA** and **$TESTC**.
- Assume that **$TESTA** has the same ten analytes as $TESTD (namely Analyte01, Analyte02, …, Analyte10).
- Assume that **$TESTC** has only three analytes, Analyte01, Analyte04,

and Analyte06.

The overall calculation could be summarized as:

$TESTD = function ($TESTA, TESTB, $TESTC)

An attempt would be made to calculate results for each of the ten component analytes of $TESTD. The value of the single component trigger, TESTB, would be used for all ten attempts.

The component calculation for Analyte01 would be summarized as:

$TESTD.Analyte01 = function ($TESTA.Analyte01, TESTB, $TESTC.Analyte01)

The component calculation for Analyte02 would be summarized as:

$TESTD.Analyte02 = function ($TESTA.Analyte02, TESTB, $TESTC.Analyte02)

However, because there is no result for $TESTC.Analyte02, the calculation would not be completed.

In the end, only three analytes results would be calculated for $TESTD, those that were present in all of the multicomponent triggers, namely Analyte01, Analyte04, and Analyte06.

Recalculation: If a target analysis already has a previously calculated result stored in the database and one of its trigger analyses has a result changed, the default behavior is to **not** recalculate the output. This behavior can be modified by means of the setting in SYSMGR.

RECALC - If True then automatic recalculation of results when input values are changed. Otherwise recalculation is not performed.

If this setting is present, previously saved calculated results will be recalculated when the result of a trigger analysis is updated.

Formulas and Operators: Formulas for built in calculations consist of place holder variables, literal numbers, operators, optional parentheses for explicit specification of the order of operation processing, and optional spaces to aid readability.

A formula may contain up to 10 different place holder variables, each of which will be replaced at evaluation time with result values from specified input analyses. The following place holders variables are recognized:

X1, X2, X3, X4, X5, X6, X7, X8, X9, and X10

A given place holder may appear more than once within a formula.

Literals used in calculation formulas may be integer or real numbers, expressed in decimal or scientific notation.

Note: The formula parser must be able to distinguish between the “+” and “-“ characters used in the exponent part of a scientific notation literal value and the same characters used as addition and

subtraction (or negation) operators. It does this by checking for the proximity of the “E” character that is also part of the exponent in a scientific notation literal.

The operators that can be used in built-in calculations are classified as unary and binary. Unary operators accept one input value (operand). The following unary operators are recognized in LABW ORKS built-in calculation formulas:

Operator	Name	Usage Example	Note
-	negation	-X1	no space before operand
INT	integer	INT(X1)	integer part of number
ABS	absolute value	ABS(X1)	
SQR	square root	SQR(X1)	
LOG	log to base 10	LOG(X1)	
LN	natural log	LN(X1)	
EXP	exponential	EXP(X1)	e raised to power of X1
SIN	sine	SIN(X1)	X1 expressed as pi radians
COS	cosine	COS(X1)	
TAN	tangent	TAN(X1)	
ATN	arctangent	ATN(X1)	

Binary operators accept two input values (operands). The following binary operators are recognized in LABWORKS built- in calculation formulas:

Operator	Name	Usage Example	Note
+	addition	X1 + X2	
-	subtraction	X1 - X2	
/	division	X1 / X2	
*	multiplication	X1 * X2	
^	raise to power	X1 ^ X2	X1 raised to X2 power
MIN	minimum value	X1 MIN X2	returns lower of X1 and X2
MAX	maximum value	X1 MAX X2	returns higher of X1 and X2

Operators are processed from right to left within a given level of parentheses. There is no implicit order of operator precedence. Pairs of parentheses should be used when necessary to explicitly specify the order of operations. Operators contained within parentheses are carried out deepest (surrounded by the largest number or parentheses) first.

Examples:

X1 + X2 *X3

X1 is added to X2 then their sum is multiplied by X3

X1 + (X2 * X3)

X2 is multiplied by X3 then their product is added to X1

SQR((X1 ^ 2) + (X2 ^ 2))

Distance formula: The square root of X1 squared plus X2 squared

((ABS(X2 – X1)) / ((X1 + X2) / 2)) * 100

% Precision calculation: The absolute value of the difference of two measurements is divided by their average and multiplied by 100

(X1 + X2 + X3) / 3

Average of three values

Excel Calculations

LABWORKS Excel calculations are performed by opening an instance of Microsoft Excel on the workstation (but leaving it hidden), opening a specified workbook file, selecting a specified worksheet, and then copying input (for the calculation) data values from LABW ORKS into specified cell addresses. After copying the input values into the worksheet,
LABW ORKS calls the Excel calculate method. Calculation output values are then copied from specified cell addresses
back into the LABW ORKS application and the workbook is closed without saving any modifications.

Unlike built-in calculations, Excel calculations may return multiple output values. Also, Excel calculations may return nonnumeric results, and the input and output values can be properties other than just primary results. Sample properties as well as analysis component result properties can be specified as input of Excel calculations.

Many LABW ORKS Excel calculations can be defined that all use the same workbook and spreadsheet. Extremely complex calculations and decision logic can be implemented using all the functions provided by Excel, any add-in libraries, and any VBA code programmed into the workbook. The workbook macro code can even contain database table calls for external lookup values if needed.

LABW ORKS Excel calculations leverage the power of a very popular and mature spreadsheet application in a way that is easy to configure and virtually transparent to the end user. One of the factors that makes Excel calculations attractive to LABW ORKS users is that many of them already know how to set up and test complex data manipulations in spreadsheets. Excel spreadsheets are also very common tools generally already being used by both Chemists and Process Engineers in their daily operations. This gives users of Excel spreadsheets the ability to utilize these files within their LIMS simply by providing LABW ORKS what data items flow into and out of that spreadsheet.

Whether you are creating a new Excel spreadsheet or want to use an existing one, you will need to enter/setup an Excel
Calculation template within LABWORKS using the **XLCalc6.exe** application.

Specifics for Excel Calculations:

Workbook files to be used with LABW ORKS calculations must be stored in folder **XLCALCS** in

the main data path. Utility program **XLCalc6.exe** is used for defining and maintaining all LABW

ORKS Excel calculations.

Every LABW ORKS Excel calculations is defined by one header row in table **XLCALCHEAD** and two or more rows in table
XLPARAMS.

The header row specifies a unique code for identifying the calculation, a user friendly description for it, and the name of the Excel workbook file and worksheet name to be used in the calculations.

A row in table **XLPARAMS** is used to specify every input and output parameter of an Excel calculation. Columns in the row are used to specify the analysis code, analyte name, property code, and worksheet cell address of every input and output parameter. **Input parameters** can be **sample properties** (including user defined field and special information) **or** they can be **analyte result properties.** If the input is a sample property, "_SAMPLE_" is stored as the analysis code and the sample property code (for example "US03" for user defined field 3) is stored as the analyte name.

Only analyte result properties can be specified as output parameters of LABW ORKS Excel calculations. The analyte result properties that can be used as input and output parameters for Excel calculations are:

- Primary result
- Secondary result
- Qualifier
- MDL (minimum detection limit)
- PQL (practical quantitation limit)
- PAM dilution factor

Some differences between built-in calculations and Excel calculations are listed as follow:

- An analysis maybe the target (output) of only one built-in calculation but it can be a target of any number of Excel calculations.
- Only the analyte primary result properties can be inputs to built-in calculations. Input parameters for Excel calculations can be sample properties or several different analyte result properties. Sample properties can be useful for decision logic. For example, a state code (political jurisdiction) might be passed into an Excel calculation to determine which of several columns of contamination limits to apply.
- Built-in calculations can have only one output parameters which must be an analyte primary result. Excel calculations may have many output parameters which can each be any of several different analyte result properties.
- Built-in calculations can have a maximum of 10 input parameters. Excel calculation allowfor as many as1500 combined input and output parameters.

Order of Calculation Preference:
Often, a sample may have all of the trigger analyses and target analyses of more than one Excel calculation assigned to it. It may also have the triggers and output of built in calculations assigned to it. In such cases rules control which of several possible calculations to perform at run time. Those rules are:

- If a given analysis is the target of a built-in calculation and one or more Excel calculations (all of whose triggers are present in the sample), **an Excel calculation is performed rather than the built-in calculation**.

- If a given analysis is the target of more than one Excel calculation (all of whose triggers are present), **the Excel calculation with the greatest number of triggers present will be performed**.

Mixing Calculations: Outputs of built-in calculations may be triggers for Excel calculations and, conversely, outputs of Excel calculations may trigger built-in calculations. Therefore, cascading calculation schemes, using both types of calculations, can be configured. However, this practice is not encouraged due to the possibility of complex support problems if a calculation is incorrect or not performing properly.

Legacy Calculation Complaints and Deficiencies

In this section we will counteract several common complaints of legacy calculation functionality using the current LABW ORKS 6.2 calculation engine solution.

Legacy Compliant: Although there are several applications for entering and modifying LABW ORKS analysis results,
only one, Results Entry, performs calculations. Therefore, the samples must be opened in results entry, either manually or via command line to perform calculations after new results are posted by multicomponent transfer or ScanDaemon.

All calculation triggers and the calculation target must be available for a calculations to fire. Applications such as multicomponent transfer and ScanDaemon have no way of knowing if the analyses whose results are being loading are, in fact, triggers to any defined calculations, let alone what the state of any other triggers might be.

Solution: Calculations are now in a business rules component that can be called from any web or Windows based LABW ORKS application. After all pending calculations are performed, the calculation engine would pass back the SIMC Sample object (as XML) with the results of all calculations that were performed. Any application could call this engine after adding one or more new or updated results to a sample before saving the updated object back to the database. This process is standard in the automated results posting process as well as the manual results entry.

Legacy Complaint: The limitation of 10 inputs at most per calculation is too restrictive, sometimes requiring multiple cascading calculations to obtain one result (the 5 Day BOD calculations for an example). Excel calculations do not have this limitation. However, Excel calcs can't be launched by automatic results posting tools when pulling in CSV files or instrument interface files.

Solution: The LABW ORKS 6.2 calculation engine now fires Excel calculations for automated posting of results (via PostResults6). Also legacy memory restrictions for built in calculations was also removed because the calculation engine can easily query the parameters table to determine if assigned analyses in the SIMC sample object are calculation triggers or targets.

Legacy Complaint: Calculation behavior should not be dependent upon INI file settings which are insecure and subject to accidental or malicious changes. Furthermore, INI files settings are not appropriate for use by back-end business layer components.

Solution: Move all run time calculation settings into system manager settings where they can be secured and controlled and readily available.

Legacy Complaint: Only the designated “key” input of a LABW ORKS built-in calculation can nonnumeric and the calculation still be performed. Settings are provided to specify how to interpret nonnumeric key input values. If the value for any other input is nonnumeric, the calculation is not performed.

Solution: Additional properties to built-in calculations now exists to allow specification of how to handle nonnumeric values for inputs other than the key input. This is reflected in new data structures used for storing calculation definitions.

Legacy Complaint: Excel must be installed on every workstation that performs LABW ORKS Excel calculations. These installations can be expensive and time consuming. Besides, they cannot be used by web applications running in a browser. Remotely automating an Excel application instance on a web server is not feasible.

Solution: The calculation engine executes Excel calculations without the need for Microsoft Excel to be installed. The calculation engine performs LABW ORKS Excel calculations using the Excel calculation services library fromMicrosoft and shared workbook files published on a Microsoft SharePoint server. This option will not require multiple installations of Excel and it can be used by any application, whether web or W indows based. Despite its universal applicability, this option may not appeal to customers who are not hosting LABW ORKS web applications or who do not want to maintain SharePoint servers.

Legacy Complaint: Only intra-sample calculations can be performed. No inter-sample calculations are provided.

Solution: The calculation engine provides an extensible platform for adding future enhancements to calculation functionality. One can envision the future addition of a several new potential functions within this engine. However, the means will have to be provided to define and store the definitions of inter-sample calculations.

Additional Methods Provided By the Calculation Engine

In addition to methods for performing calculations, the calculation engine must also provide a method for setting four Boolean properties of every SIMC sample assigned analysis.

Those properties are:

IsCalcTrigger	(is a trigger for a built-in calculation)
IsCalcTarget	(is a target of a built-in calculation)
IsXLCalcTrigger	(is a trigger for an Excel calculation)
IsXLCalcTarget	(is a target of an Excel calculation)

This method is called once when an SIMC sample is loaded into a results entry application to set the properties which will then be used throughout the session by the SIMC and result entry application (or automated results posting process) to determine when calls to the methods for performing calculations need to be made.

The Specifications Check Step: Automated Results Posting

Introduction

A key step in the results posting process is the final step, when violations are checked. During the manual posting of results in the Results Entry module, violations are shown as color variances (yellow for a warning, red for a full violation). During the automated results posting process, there are no visual designation of violations until the posted results are reopened in Results Entry. However, the violations are posted into the database during the process.

Specifications used in the results posting process (manual or automated) can be defined in four possible places.

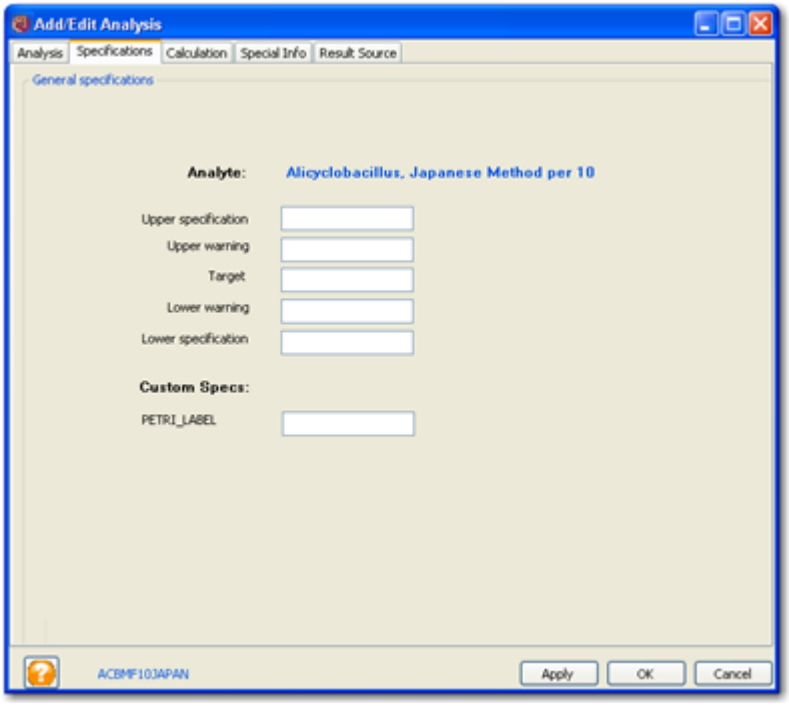
- Within Analysis Code (through **Analysis List Maintenance**)
- Within Location Code (through **Location List Maintenance**)
- Within Certificate of Analysis (using the program **COAPRN6**)
- At Sample Login (using post-login program **SampSpecLogin6**)

The Analysis Code level is the broadest range and is considered a global specification. The Location Code level takes precedence over the Analysis Code level as it is becoming more finely tuned or narrower in its use, and it's for a specific location code rather than a broad analysis code definition that can be used for any location code. The Certificate of Analysis level is the most specific / defined level of specifications generally for a specific client and/or product of the lab. These specifications take precedence of the previous two. However, any of these specifications can be modified after Sample Login using the **SampSpecLogin6** before the final posting to the database. Generally, this program is used to make any final specification modifications before final posting to the database in preparation for results entry (last minute spec changes at login).

Specifications Defined in Analysis List Maintenance

The first, most general, form of specification entry mentioned above is within the **Analysis List Maintenance** module. Keep in mind that specifications entered in other areas of the system take precedence over specifications entered here.

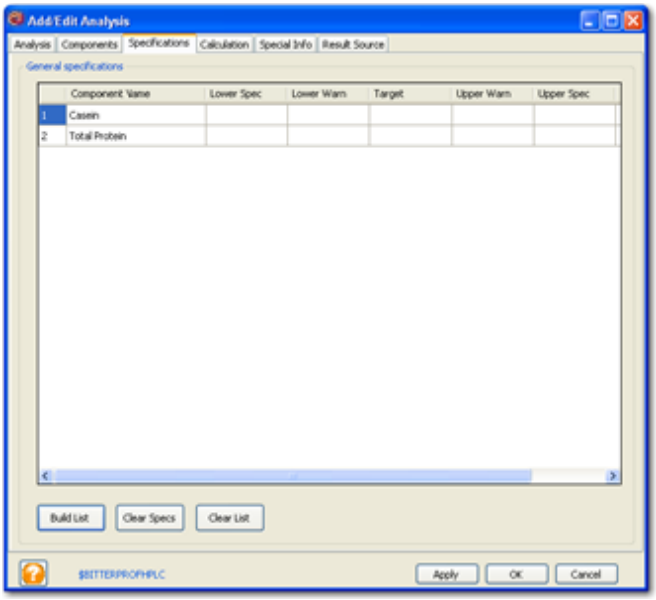
After opening a specific analysis code for modification, click the **Specifications** tab.



5 “**general specifications**” are displayed initially followed by any defined **Custom Specs** (fields added manually to the **RLTSPECS** table for additional spec checking needed) will displayed on this tab.

MultiComponent Specifications

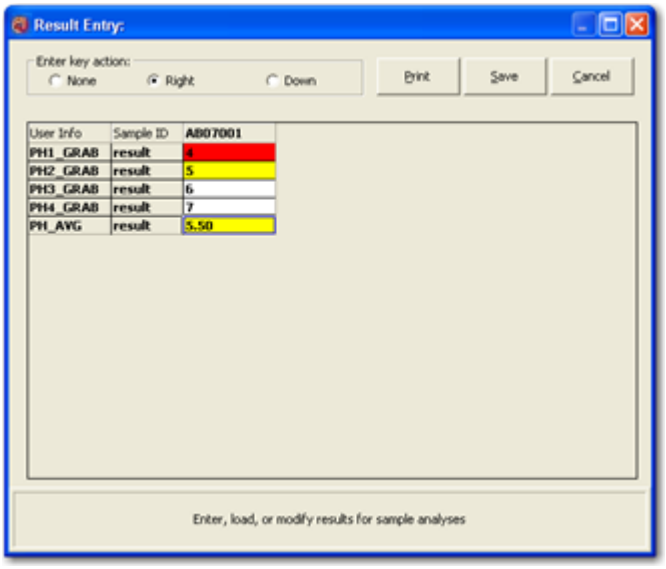
Multicomponent specifications are also entered within the **Specifications** tab although the components are listed in a tabular form and specs must be entered for each individual component (each row).



These “**general**” specs on this tab are defined as follows.

- Upper specification – an upper violation shown within the UI as **RED** when results are above this value.
- Upper warning – an upper warning shown within the UI as **YELLOW** when results are above this value yet below the upper specification limit defined above.
- Target – normally used for text type results. For example, PASS vs. FAIL. Numeric targets are not typically used outside of for informational purposes only.
- Lower warning – a lower warning shown within the UI as **YELLOW** when results are below this value yet above the lower specification limit defined below.
- Lower specification – a lower violation shown within the UI as **RED** when results are below his value.

When doing manual results posting within Results Entry module, these violations can be clearly viewed prior to saving the results to the database.



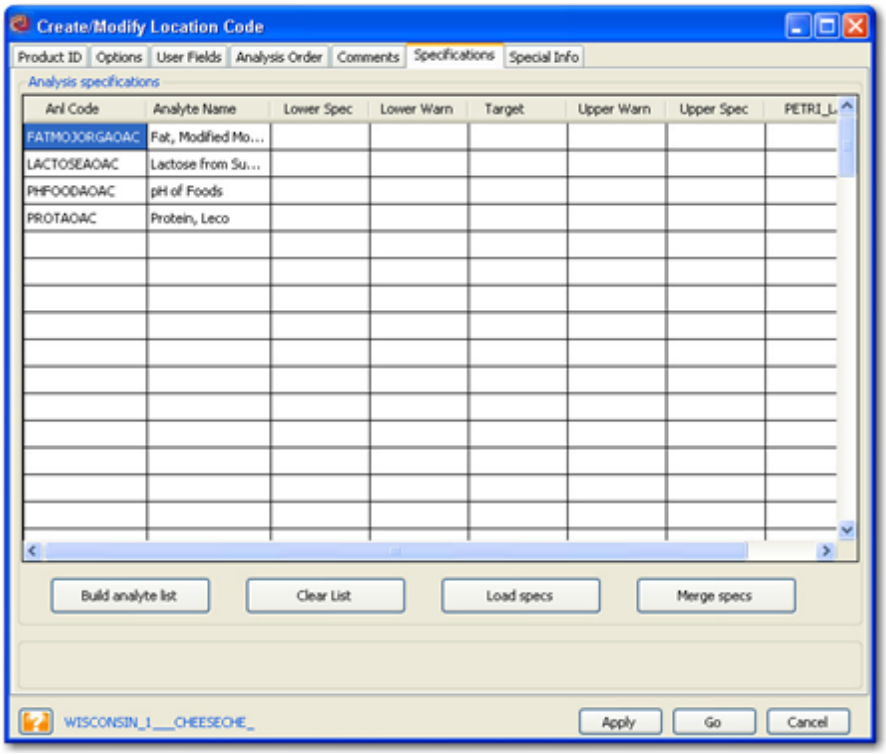
After the results are saved, or “posted”, the violations are stored in the database.

The color-coded UI within the manual results posting process is a good way of catching typos that can be corrected prior to clicking the SAVE button. However, when posting the results through an automated process (PostResults6), the user does not have the option to view the violations before the results are saved to the database. This opens potential data- entry issues that could exist within the input file. As mentioned earlier in this chapter, we always suggest a visual inspection of the input file prior to firing off any automatic posting process. Otherwise, any modifications that are made after-posting will fire off an audit which could itself create issues for any ISO-certified lab that is required to explain/document every result change after posting. A quick inspection of a result input file prior to an automated posting may save headaches later.

Specifications Defined in Location List Maintenance

The next form of specification entry mentioned which takes precedence over any specs entered within **Analysis List**

Maintenance module, is the Location List Maintenance module.



From a support standpoint, if someone is noticing a specification within an Analysis Code that is not behaving properly in Results Entry, it is usually because there are also specifications defined within the Location Code used.

The general and custom specs defined in the previous section work the exact same way (as result posting violations and warning colors) for Location Codes.

Specifications Defined in Certificate of Analysis (COA)

The next form of specification entry mentioned takes precedence over any specs entered within both **Analysis List Maintenance** and **Location List Maintenance** modules. As mentioned in the previous section, if specifications are not operating properly in either maintenance modules, always check the COA module.

Certificate of Analysis (COA) is an optional program for use within LABW ORKS. It is used to generate Certificates of Analysis (documentation regarding ingredients or composition of a material). It stores information about shipments of materials to customers that can be used to create quality control charts.

This COA option also has two modes for evaluating material, **Forward Search** and **Reverse Evaluate**.

- **Forward Search** takes a batch of material and evaluates all the customers who purchase that type of material to determine which customer's specifications the material satisfies.
- **Reverse Evaluate** looks for batches of material that satisfy a single customer's requirement.

Certificates of Analysis may be generated by establishing a client database for specifications on products. After setting up the option, laboratory identification numbers (Lab ID#) may be selected along with customer product numbers to extract all pertinent data from LABW ORKS for display. Information about items such as common carriers, trailer/car numbers, and other items may be entered as well. The COA displays the sample analysis versus customer specifications with any out-of-spec results highlighted. W hen the COA is approved, the program generates a Certificate of Analysis automatically. This shipment data is also stored so the customer shipment history can be reported.

The COA module requires some setup time. Customer "bases" are used to define client data where specifications can be entered. If there exists COA data for analysis codes or analytes selected within a results posting process, these will have precedence over any other specifications entered in LABW ORKS.

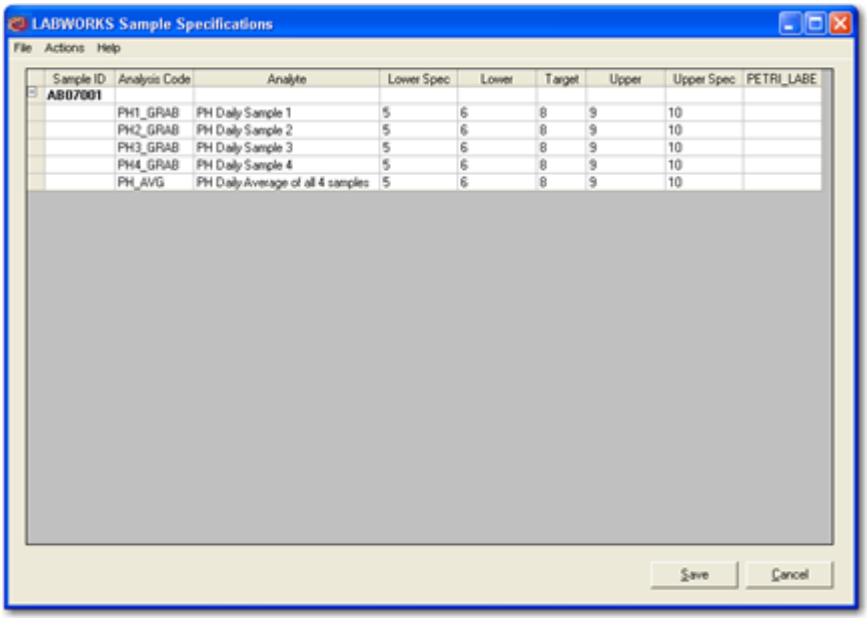
Specifications Defined at Sample Login

The last form of specifications entry prior to results posting is done through Sample Login using the **SampSpecLogin6.exe** program defined as a post-login user program.

Regardless which of the previous three forms of specifications take precedence, the SampSpecLogin6 program allows the user to make any final adjustments before the samples are posted to the database.

These final adjustments will be the final specifications used at results entry in both manual and automated processes.

Changing specifications at login is rarely done. However, it is sometimes required to be done by certain labs who need to make final adjustments as required by clients dropping off samples before analysis is performed. It provides a quick way to do that at Sample Login without requiring the user to exit Login and enter either of the Maintenance programs or the COA module (if used) to make that change.



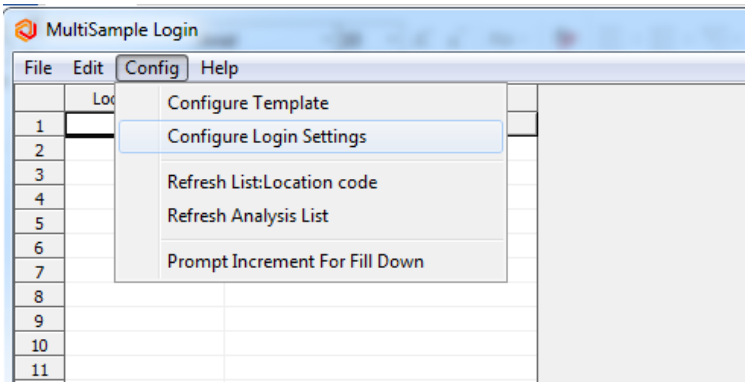
When the **SampSpecLogin6** program is launched from Sample Login, it will display all samples in tabular form with the corresponding analysis code, associated analytes, and all general and custom specs pulled from any of the three modules defined in the previous sections (whichever took precedence).

The user can either ignore this screen by clicking **Cancel** (which does nothing to the specs displayed), or the user can modify the specs and click **Save**. Saving specs changed here will over-write any previous specs. Specs saved here will be the final specs used during the results posting process.

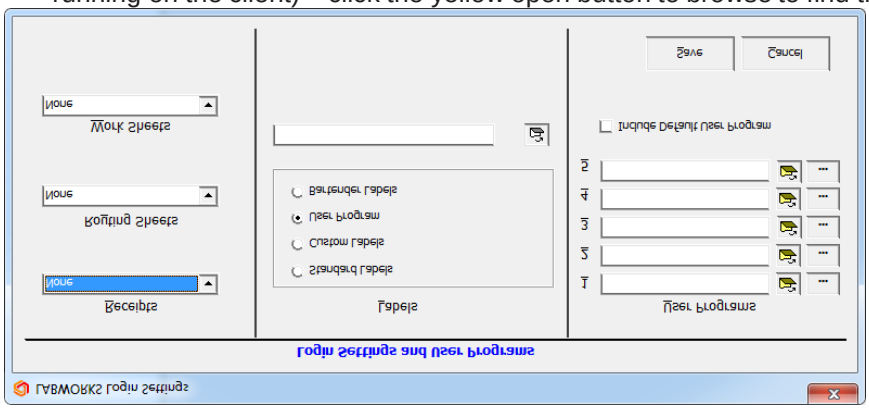
Setting up SampSpecLogin6

The process of setting up **SampSpecLogin6** to fire after Sample Login requires adding it to the list of post-login programs.

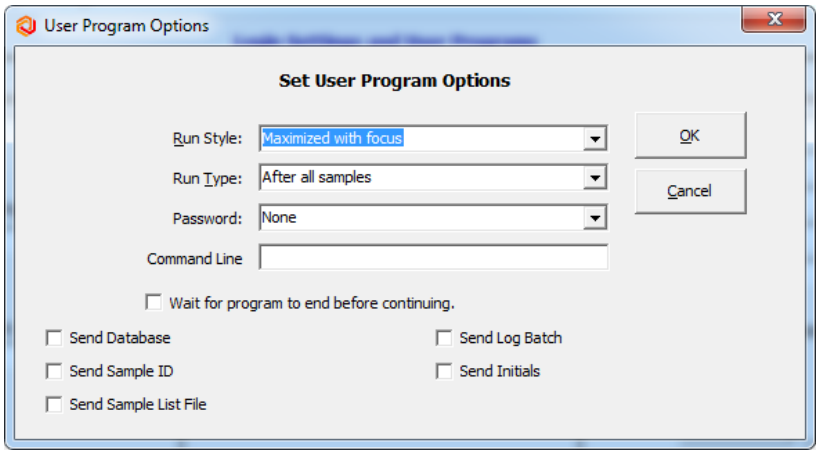
1. Open MultiSample Login
2. Click the **Config** menu
3. Select **Configure Login Settings**



4. Find a blank spot under the User Programs to enter the path to the SampSpecLogin6.exe program (in the applications path under **C:\LABWORKS\Client** or where ever programs are running on the client) – click the yellow open button to browse to find the program



5. Click the ellipse button (…) and enter the User Program Options



6. Enter all required settings

   a. Run Style: **Normal with focus**
   b. Run Type: **After all samples**
   c. Password: **None**
   d. Command Line: **EXPANDALL** (causes the expansion of sample/analysis list for quick viewing)
   e. Check "**Wait for program to end before continuing**", "**Send Sample List File**", and "**Send Log Batch**"
7. Click **OK**
8. Click **SAVE**

Now you will have the capabilities to make any final modification of specs at Sample Login prior to the results posting step.

Note: if you are also using **lwPostResults6** at Sample Login as well, it really makes no difference what order they are placed in.

Summary

Automated posting in LABW ORKS is a means of entering results into the LIMS without the use of the Results Entry module itself. There are many activities going on in the background when a user enters results. These same activities happen when another application posts results into the system through an automated process.

Automated posting of results in LABW ORKS 6.2 is done via the **PostResults6** tool which is a console application driven by command parameters. The process of configuration and execution of this tool can be made easier by the use of its UI counterpart, **lwPostResults6**. W hen **lwPostResults6** is executed stand-alone a **Setup Wizard** is launched. W hen launched with a **PR:1** parameter, it will fire off the console tool for posting results. Finally, **lwPostResults6** can be added to Sample Login for easy creation of the input file to be sent off to a contract lab for results entry (and for later importing).

The PostResults6 tool follows all the standard manual result entry posting process which includes auditing, results conditioning, calculations, rounding, and finally spec checking and violation posting. This is done via new results conditioning and calculation/rounding components in the new LABW ORKS 6.2 architecture making the process far superior over legacy results posting methods for several reasons. Some of these reasons include (but not limited to) performance, additional field posting ability, Excel calculations available without needing Excel installed, etc.

Within this manual, each of the steps within the results posting process both during manual entry and through automated tools in LABW ORKS 6.2 was analyzed and documented. Any additions or amendments to the process will be reflected within forthcoming versions of this documentation.

Appendix A: The Complete List of System Control Options in SYSMGR/System Manager

SYSMGROPT	SYSMGRVAL	SYSMGRLVAL	SYSMGRBVAL	SYSFIELDTYPE	Description
@CODE_IN_RESULTS		0	0	IB	SYSMGRBVAL = 0 then test codes with @ prefix (container test codes) will not be displayed in results entry but can be edited in container tracking module. SYSMGRBVAL = -1 and @test codes show up in results entry.
ACTIONPASSPW	AP	0	0		INTERNAL Use/any string value is ok
ACTIVITY_NOTIFY_TIME		10	0	IL	SYSMGRLVAL = # of minutes client without update before client is considered off line
ACTIVITYTIMEOUT	(long)	1800	0		SYSMGRLVAL = Not Used - migrated to ACTIVITY_NOTIFY_TIME
ADD_TO_PROCESS_SCHEDULER	PS add sample	0	0	IB	SYSMGRBVAL = 0 and samples logged into from Labworks Sample Login module will not be added to Process Scheduler. BVAL = -1 and samples will be added to Process Scheduler.
ADJUST_PQLS_FOR_DILFACTOR	Adjust PQL for Dilution Factor	0	-1	IB	SYSMGRBVAL = On data upload, should PQL's be adjusted for the dillution factor
ALLOW_AD_HOC_GROUP_PARAMETER S	allow ad-hoc group parameters	0	0	IB	SAMPLE LOGIN/Persisted Groups - allow ad-hoc group parameters
ALLOW_AD_HOC_GROUP_PROPERTIES	allow ad-hoc group properties	0	0	IB	SAMPLE LOGIN/Persisted Groups - allow ad-hoc group properties
ALLOW_BLANK_ROWS	DMR REPORT PRINT SETTING	0	-1	IB	DMR / DMR REPORT PRINT SETTING

SYSMGROPT	SYSMGRVAL	SYSMGRLVAL	SYSMGRBVAL	SYSFIELDTYPE	Description
ALLOW_CALC_RESULT_MANUAL_ENTR Y		0	-1	IB	SYSMGRBVAL = -1 allows a calculated result to be changed in Results Entry. = 0 and calculated result is locked for editing.
ALLOW_CUSTOM_REASON		0	-1	IB	Audit Trail - Allow user entered audit reasons
ALLOW_PREVIOUS_REASON		0	-1	IB	Audit Trail - Show previous reason list
ALLOWPASSWRDCHNG	(bool)	0	0	IB	Depricated
ANALYSIS_OWNERSHIP_ACTIVE	Analysis ownership tracking is active	0	0	IB	Analysis (Sample) ownership tracking is active
ANALYSIS_OWNERSHIP_PESSIMISTIC	Analysis ownership is pessimistic	0	0	IB	Analysis (Sample) ownership is pessimistic
ANL_LIST_HIST		0	0	IB	Obsolete - AuditTrail improved to keep track of analysis list changes
ANLCODE_OWNERSHIP_ACTIVE		0	0	IB	Analysis Code ownership tracking is active
ANLCODE_OWNERSHIP_PESSIMISTIC		0	0	IB	Analysis Code ownership is pessimistic
ANLYST_BY_DATE		0	-1	IB	Analysis Assignement - Check Analyst Date
ANLYST_BY_STATUS		0	-1	IB	Analysis Assignement - Check Analyst Status
ANLYST_BY_TEST		0	-1	IB	Analysis Assignement - Check Analyst Tests
AUDITON		0	-1	IB	SYSMGRBVAL = -1 activates audit trailing. = 0 and audit trail will not be active.
AUTO_VALIDATE_RULE		0	0	IB	SYSMGRVAL = "NO_VIOLATIONS" then sample is auto validated when all analyses complete without violation (PostResults6)

SYSMGROPT	SYSMGRVAL	SYSMGRLVAL	SYSMGRBVAL	SYSFIELDTYPE	Description
BACKLOG_CSV_FORMAT	5	0	0	IT	SYSMGRVAL = "5.8" or "6.2" | 6.2 adds adtional fields to CSV export
BANAME1	Invoicing address	0	0		Out of box name for Billing Address field name changed to value entered into SYSMGRVAL column (long name label)
BANAME2	invoicing address	0	0		Out of box name for Billing Address field name changed to value entered into SYSMGRVAL column (long name label)
BANAME3	InvAddr	0	0		Out of box name for Billing Address field name changed to value entered into SYSMGRVAL column (abbreviated name label)
BANAME4	Inv Addr	0	0		Out of box name for Billing Address field name changed to value entered into SYSMGRVAL column (abbreviated name label)
BARTENDER_DOMAIN	DomainName	0	0	IT	WebTop Bartender Printing - User Domain
BARTENDER_PASSWORD	Password	0	0	IT	WebTop Bartender Printing - User Password
BARTENDER_SHOWDIALOG	Always display print dialog	0	0	IB	WebTop Bartender Printing - Show Print Dialog
BARTENDER_USERNAME	UserName	0	0	IT	WebTop Bartender Printing - User Name
BATNUMBER		111	0		Next QC Batch Number sequence in SYSMGRLVAL column
CASNUMB	CAS#	0	0	IT	Display label for Analyte CAS (Chemical Abstract Number) out of box changed to value entered into SYSMGRVAL column)

SYSMGROPT	SYSMGRVAL	SYSMGRLVAL	SYSMGRBVAL	SYSFIELDTYPE	Description
CALIBRATION_NOTIFY_RECIPIENTS	Email address	0	0	IT	Configure email ids separated by comma to get calibration due date notification mails. In case, the default value is zero, email notifications will not be sent.
CALIBRATION_WARNING_DAYS		2	0	IL	Configure number of prior days to the calibration due date, on which email notifications should be sent. In case, the default value is zero, email notifications will not be sent.
CINV_AUTOLOT			0	IB	Chemical Inventory AutoIncrement Lot Number
CINV_LOTCOUNTER		0		IL	Chemical Inventory Lot Number
CINV_OVRDEXPDATE			0	IB	Chemical Inventory Override Expiry Date
COA_CERT_SAMP_SELECT_FIELDS	SSTA LCOD SUBDT COLDT	0	0	IT	Certificate of Analysis - Used by sample selection when more than one sample is entered for COA generation
COA_CUSTOMER_NAME		0	0	IT	Certificate of Analysis - Used to enable client specific code
COA_REV_EVAL_SAMP_FIELDS		0	0	IT	Certificate of Analysis - Reverse evaluation sample selection fields
COANUMBER		1	0		Certificate of Analysis - Next COA number
CONTAINER_FORMAT	123	0	0		Unique Container naming - Either 123 or ABC
CONTAINERID		2946	0		Unique Container naming - Number of next container
CRYSTAL_VERSION	10	0	-1		Crystal Reports Version - "XI" or any other value - "XI" uses Crystal XI and supports embedded designer. <> "XI" uses Crystal 2013 runtime version for display
CURRENT_LB_DAYS		3650	0	IL	Sample Selection Component, number of days prior to today for including Login Batch selection.

SYSMGROPT	SYSMGRVAL	SYSMGRLVAL	SYSMGRBVAL	SYSFIELDTYPE	Description
CURRENT_LOCATION		0	0	IT	It is a human-readable current location description (address/city/state/zip code). When opening maps (from Sample Login or from Location Maintenance), the map is positioned at this location if the address field is not defined.
CUSTINST_01_NAME	LEEMAN	0	0	IT	Custom (lwiiso.dll) interfaces CUSTINST_XX_Name/Type
CUSTINST_01_TYPE	MC	0	0	IT	Type is either MC or SC
DATABASEID	LABWORKS Water Demo Database	0	0	IT	Description displayed in Help/About
DATABASETYPE	SQLSERVER	0	0	IT	Depricated: Databasetype is read from LWSysconfig value
DATAREADPW	MASTERRO	0	0		Depricated: Foxpro password
DATAWRITEPW	MASTER	0	0		Depricated: Foxpro password
DBVERSION	6.9.0.0	0	0		Database version
DEFAULT_BDL_REPLACE_VALUE				IT	PostResults6 - Default Below Dection Limit Replacement Value
DEFAULT_ND_REPLACE_VALUE				IT	PostResults6 - Default Not Detected Replacement Value
DEFAULT_REPLACE_BDL			0	IB	PostResults6 - Yes/No Replace Below Detection Limit
DEFAULT_REPLACE_ND			0	IB	PostResults6 - Yes/No Replace Not Detected
DEFAULT_ROUNDING_DECIMALS		0		IL	PostResults6 - Number of default decimal values
DEFAULT_ROUNDING_METHOD	N			IT	PostResults6 - Rounding Method S/D S = SigFigs, D = Decimal point

SYSMGROPT	SYSMGRVAL	SYSMGRLVAL	SYSMGRBVAL	SYSFIELDTYPE	Description
DEFAULT_ROUNDING_SIGFIGS		0		IL	PostResults6 - Number of default sig figs
DEFAULT_ROUNDING_STYLE				IT	PostResults6 - Rounding Style - S/E E = Even/Odd (Default), S = Non-Even/Odd
DMR_MDLPROCESSING	DMR MDL PROCESSING	0	0	IB	DMR - Evaluate results with >, < prefixes
DMR_PAMPROCESSING	DMR PAM PROCESSING	0	0	IB	DMR - Get PAM (Project Analyte Management) Limits
DMR_PASSWORD		0	0	IT	Depricated
DMRFINALREPORTTYPE	EXCEL	0	0	IT	DMR - Report Type EXCEL/CRYSTAL
DNAME1	Sample Description	0	0		Out of box name for Sample Description field name changed to value entered into SYSMGRVAL column (long name label)
DNAME2	sample description	0	0		Out of box name for Sample Description field name changed to value entered into SYSMGRVAL column (long name label)
DNAME3	Descript	0	0		Out of box name for Sample Description field name changed to value entered into SYSMGRVAL column abbreviated name label)
DNAME4	Descript	0	0		Out of box name for Sample Description field name changed to value entered into SYSMGRVAL column abbreviated name label)
DUE_COLOR	GREEN	0	0	IT	Backlog - Color used for Due samples/analyses
DUE_HOURS		24	0	IL	Backlog - Hours used for Due samples/analyses
DV_DEFAULT_ACODE		0	0	IT	DataViewer - Acode user for interpretation
DV_LIST_OF_VALUES		0	-1	IB	DataViewer - List of values used for interpretation
DV_MOUSE	COMMENTS	0	0	IT	DataViewer - Mouse action COMMENTS or NARRATIVE

SYSMGROPT	SYSMGRVAL	SYSMGRLVAL	SYSMGRBVAL	SYSFIELDTYPE	Description
DV_NR_COLOR		0	-1	IB	DataViewer - (For future use)
DV_NUM_POINTS		0	0	IL	DataViewer - Number of Data points for history display
DV_NUMBER_OF_STATUS		0	0	IL	DataViewer - (For future use)
DV_PICKLIST_NAME		0	0	IT	DataViewer - Interpretation Picklist
DV_SHOW_CONTROLCHART	0	0	-1	IB	DataViewer - Include display of control chart in history
DV_USE_NR_VALUES		0	-1	IB	DataViewer - (For future use)
DV_USE_PICKLIST		0	-1	IB	DataViewer - Use Picklist in Narrative Interpretation Form
EXPLORER_GRAPH_DEFAULT		10	0	IL	Explorer - History- (For future use)
EXPLORER_GRAPH_POINTS		30	0	IL	Explorer - History - Number of points to include
EXPLORER_GRAPH_QUALIFIER		0	-1	IB	Explorer - History - Include Qualifier in table display
EXPLORER_GRAPH_RESULT2		0	-1	IB	Explorer - History - Include Results2 in table display
EXPLORER_GRAPH_SORTDATE	COL	0	0	IT	Explorer - History - Sort (COL/SUB/ANL - Collect/Submit/AnalysisStart
EXPLORER_GRAPH_TOP	EARLY	0	0	IT	Explorer - History - Sort EARLY/LATE

SYSMGROPT	SYSMGRVAL	SYSMGRLVAL	SYSMGRBVAL	SYSFIELDTYPE	Description
EXPORT_MDB_VER	(2, 97, 2000, 2007, 2010)	2000	0	IL	Export - Version of MDB < 2007 then "mdb". >= 2007 then "accdb"
FILTER_BY_MATRIX		0	0	IB	Sample Login, filter analysis codes by matrix
FORCE_AUDIT_REASON		0	0	IB	AuditTrail - Force user to enter a reason
FORCEANALYSISVAL		0	0	IB	Validation - Require Analyses to be validated before Samples can be validated.
FORCECHECKIN		0	0		Require user to enter password before data changes are saved
GOOGLE_MAPS_API_KEY		0	0	IT	It is a Google Maps API key that is used to map controls work correctly.
HASPAMDILFACTOR		0	0	IB	PAM/Project Analyte Management - Use Dilution Factor from PAM
HIDE_CHECKSUM_VIOLATION		0	0	IB	Do not display checksum violations to user, only record in errorlog
HIDE_MENU_ITEMS		0	0		Desktop - (For future use)
I18N_WORKSTATION		0	0	IT	Depricated - ID of Workstation for I18N Translation
IMCMODIFY	Create, Modify, Delete IM&C records	0	0		Depricated - Moved to PRIVS - Modify Instrument Maintenance Records
INSTR_BY_DATE		0	-1	IB	Qualify Instrument by Date
INSTR_BY_STATUS		0	-1	IB	Qualify Instrument by Status
INSTR_BY_TEST		0	-1	IB	Qualify Instrument by Analysis

SYSMGROPT	SYSMGRVAL	SYSMGRLVAL	SYSMGRBVAL	SYSFIELDTYPE	Description
INVNUMBER		32	0		Invoice Number - Next value
LABELPRINTERSEARCH	Microsoft Office	0	0	IT	LWSampleLogin - Bartender Label search value
LNAME1	Sample Point	0	0		Out of box name for LocationCode field name changed to value entered into SYSMGRVAL column (long name label)
LNAME2	sample point	0	0		Out of box name for LocationCode field name changed to value entered into SYSMGRVAL column (long name label)
LNAME3	SamplePt	0	0		Out of box name for LocationCode field name changed to value entered into SYSMGRVAL column (abbreviated label)
LNAME4	SamplePt	0	0		Out of box name for LocationCode field name changed to value entered into SYSMGRVAL column (abbreviated label)
LOCCODE_OWNERSHIP_ACTIVE	Location code ownership tracking is active	0	0	IB	Location code ownership tracking is active
LOCCODE_OWNERSHIP_PESSIMISTIC	Location code ownership is pessimistic	0	0	IB	Location code ownership is pessimistic
LOCK_VALID_SAMPS	Lock Sample modify after validation	0	0	IB	Lock Sample modify after validation
LOCKWAITSECS		60	0	IL	Amount of time query waits for table lock before generating error
LOGBATCHIDFORMAT	YYYY-MM-DD-XXX	0	0	IT	Login Batch Format - See document on supported formats
LOGIN_DEPT		0	0	IT	LWSampleLogin - Process Scheduler department value
LOGIN_INITIAL_STATUS	2	0	0	IT	LWSampleLogin - Process Scheduler initial status
LOGIN_PRIORITY		1	0	IL	LWSampleLogin - Process Scheduler initial

SYSMGROPT	SYSMGRVAL	SYSMGRLVAL	SYSMGRBVAL	SYSFIELDTYPE	Description
LOGINBATCHID_MODIFY			1	IT	Login Batch ID - User can modify
LOGINBATCHNAME	2014-02-03-004	0	0	IT	Record of most recent login batch
LRFNAME1	Log Group ID	0	0		Out of box name for Login Record File (aka Login Batch, Login Group, Sample Group, etc.) field name changed to value entered into SYSMGRVAL column (long name label)
LRFNAME2	log group ID	0	0		Out of box name for Login Record File (aka Login Batch, Login Group, Sample Group, etc.) field name changed to value entered into SYSMGRVAL column (long name label)
LRFNAME3	LogGrpID	0	0		Out of box name for Login Record File (aka Login Batch, Login Group, Sample Group, etc.) field name changed to value entered into SYSMGRVAL column abbreviated name label)
LRFNAME4	LogGroupID	0	0		Out of box name for Login Record File (aka Login Batch, Login Group, Sample Group, etc.) field name changed to value entered into SYSMGRVAL column abbreviated name label)
MAINTENANCE_NOTIFY_RECIPIENTS	Email Address	0	0	IT	Configure email ids separated by comma to get maintenance due date notification mails.
MAINTENANCE_WARNING_DAYS		2	0	IL	Configure number of prior days to the maintenance due date, on which email notifications should be sent.In case, the default value is zero, email notifications will not be sent.
MAX_SAMPLES	max samples to process in PR6	0	0	IL	Process Scheduler - If number of samples to display is greater than this value, then warn user performance could be slow
MC_NONPOS		0	0	IT	PostResults6 - For Future Use
MCBDL	<MDL	0	0	IT	PostResults6 - MultiComponent Below Dection Limit

SYSMGROPT	SYSMGRVAL	SYSMGRLVAL	SYSMGRBVAL	SYSFIELDTYPE	Description
MCND	ND	0	0	IT	PostResults6 - MultiComponent Not Dected
MCSEL_EDITAUDIT	Produces an audit when anything is edited AFTER Sample Login	0	0	IB	MultiComponent Select - Produces an audit when anything is edited AFTER Sample Login
MCSEL_LOGINAUDIT	Produces an audit at Sample Login	0	0	IB	MultiComponent Select - Produces an audit at Sample Login
ML_APP_NAME	Client Information	0	0	IT	Rename Mail List AppName
MOVETOARCHIVE		0	-1	IB	Move serviced messages to message archive
NOEXCEPTNOEMAIL		0	-1	IB	Exception Report - Only send email if exception
NOTIFY_REFRESH	1	0	0		Desktop - Refresh user messages
OVERDUE_COLOR	RED	0	0	IT	Backlog - Overdue Color
OWNERON		0	-1	IB	Sample ownership feature turned on (SYSMGRBVAL = -1) or turned off (SYSMGRBVAL = 0)
OWNERPESSIMISTIC		0	0	IB	Sample Ownership pessimistic
PASSWORDEXPIRES	(long)	0	0	IL	Depricated
PASSWORDLOCKOUT	(bool)	0	0	IB	Depricated
PASSWORDWARNING	(long)	0	0	IL	Depricated
PG_ALLOW_AD_HOC_GROUP_PARAME TERS		0	-1	IB	Persisted Group - Allow Ad-hoc Parameters

SYSMGROPT	SYSMGRVAL	SYSMGRLVAL	SYSMGRBVAL	SYSFIELDTYPE	Description
PG_ALLOW_AD_HOC_GROUP_PROPERTIES		0	-1	IB	Persisted Group - Allow Ad-hoc Properties
PGMANUMBER	PGMA GROUP ID COUNTER	0	-1	IL	Persisted Group - Next number
PJNAME1	Project	0	0		Out of box name for Project Code field name changed to value entered into SYSMGRVAL column (long name label)
PJNAME2	Project	0	0		Out of box name for Project Code field name changed to value entered into SYSMGRVAL column (long name label)
PJNAME3	project	0	0		Out of box name for Project Code field name changed to value entered into SYSMGRVAL column (abbreviated name label)
PJNAME4	PROJECT	0	0		Out of box name for Project Code field name changed to value entered into SYSMGRVAL column (abbreviated name label)
PONAME1	Comp/Grab	0	0		Out of box name for Purchase Order field name changed to value entered into SYSMGRVAL column (long name label)
PONAME2	Com/Grab	0	0		Out of box name for Purchase Order field name changed to value entered into SYSMGRVAL column (long name label)
PONAME3	comp/gra	0	0		Out of box name for Purchase Order field name changed to value entered into SYSMGRVAL column abbreviated name label)
PONAME4	COMP/GRAB	0	0		Out of box name for Purchase Order field name changed to value entered into SYSMGRVAL column abbreviated name label)

SYSMGROPT	SYSMGRVAL	SYSMGRLVAL	SYSMGRBVAL	SYSFIELDTYPE	Description
PSID		5340	0	IL	Next Process Scheduler ID in SYSMGRLVAL column
PSMAXRECORDS		3000	0		Process Scheduler maximum number of records to return on refresh
PSSHOWRESOURCE	Process Scheduler Resource Tab	0	-1		Process Scheduler Resource Tab
PSSHOWSHELFLIF		0	0	IB	Process Scheduler Show Shelf Life tab
PSUPDATETIME	Nov-23-2010 10:57	0	0		Most recent update of ProcessScheduler automatic sample login
PTMODIFY	Create, Modify, Delete Personnel Training records	0	0		Depricated
PS_USECOLLECTED		0	-1	IB	In LW Sample Login add “COLLECTED” status to initial status if PS_USECOLLECTED is checked.
QA_PARAM_ORDER	BLKV_ ICB_ ICV_ LCSV_ LCSA_ LCSR_ DUPV_ DUPP_ SPKV_ SPKA_ SPKR_ SPDV_ SPDR_ CCB_ CCV_	0	0	IT	QA Parameter order used in QAQC Setup
QA_PREFIX_BLKV_	BLANK	1	0		QA Prefix - User defined
QA_PREFIX_CCB_	CCB VALUE	14	0		QA Prefix - User defined
QA_PREFIX_CCV_	CCV VALUE	16	0		QA Prefix - User defined
QUOTENO	Quote Number	1	0	IL	Depricated - Next Quote Number

SYSMGROPT	SYSMGRVAL	SYSMGRLVAL	SYSMGRBVAL	SYSFIELDTYPE	Description
REMOVEINTERVAL		0	0	IL	Period of time for removal of serviced messages
REPBATNUMBER		1	0	IL	Email Report Batch Number
REPORT_OWNERSHIP_ACTIVE	Report template ownership tracking is active	0	0	IB	Report template ownership tracking is active
REPORT_OWNERSHIP_PESSIMISTIC	Report template ownership is pessimistic	0	0	IB	Report template ownership is pessimistic - don't include un-assigned
RESERVICEINTERVAL		0	-1	IB	Period before reserviceing messages that are in-process
RESULTSENTRY_DISABLE_TIMECHECK	Disable time check	0	0	IB	ResultsEntry6 - to disable time check
ROUND_CALCULATED_RESULTS			-1	IB	PostResults6 - Round calculated values
ROUND_IMPORTED_RESULTS			-1	IB	PostResults6 - Round Imported values
ROUND_LOGGED_RESULT			-1	IB	PostResults6 - For Future Use
ROUND_MANUAL_RESULTS			-1	IB	PostResults6 - For Future Use
ROUND_SPECIFIED_RESULT			-1	IB	PostResults6 - For Future Use
SAMPMODIFY_DUEDATE_RECALC		0	0	IT	Show/Hide prompt to recalculate due date in sample modify
SAMP_SPEC_SOURCE	SAMP_SPEC_TABLE	0	0	IT	Source for Sample Specs - SAMP_SPEC_TABLE/SAMP_SPEC_TABLE_ONLY/NULL
SAMPLE_OWNERSHIP_ACTIVE	Sample ownership tracking is active	0	-1	IB	Sample ownership tracking is active

SYSMGROPT	SYSMGRVAL	SYSMGRLVAL	SYSMGRBVAL	SYSFIELDTYPE	Description
SAMPLE_OWNERSHIP_PESSIMISTIC	Sample ownership is pessimistic	0	0	IB	Sample ownership is pessimistic - don't include un-assigned
SAMPLELOGIN_MAX_SAMPLES		50	0	IL	LWSampleLogin - Maximum number of samples in login batch
SAMPNUMBER		400336	0		Sample Number - Value of next number
SAMPQUERYSIZE		200	0	IL	Database Query Parameter, how many SAMPNOs to query in a single statement
SAVE_CUSTOM_REASON			-1	IB	Audit Trail - Save new reasons
SCAND_UPDATETIME	May-04-2010 15:14				ScanDaemon - Last updated time
SCBDL	<MDL	0	0	IT	PostResults6 - Single Component Below Dection Limit
SCHUMAKER		0	0	IB	Used with Custom program for Schumaker Instrument Validation
SCHUMAKER_AREF	SCHU	0	0		Used with Custom program for Schumaker Instrument Validation - Acode
SCNAME1	Sample Collector	0	0		Out of box name for Sample Collector field name changed to value entered into SYSMGRVAL column (long label)
SCNAME2	sample collector	0	0		Out of box name for Sample Collector field name changed to value entered into SYSMGRVAL column (long label)
SCNAME3	Collectr	0	0		Out of box name for Sample Collector field name changed to value entered into SYSMGRVAL column (abbreviated label)

SYSMGROPT	SYSMGRVAL	SYSMGRLVAL	SYSMGRBVAL	SYSFIELDTYPE	Description
SCNAME4	Collector	0	0		Out of box name for Sample Collector field name changed to value entered into SYSMGRVAL column (abbreviated label)
SCND	Not detected	0	0	IT	PostResults6 - Single Component Not Dected
SEC_RESUME_PROC	PROMPT_PW	0	0	IT	Used with SYSTEMTIMEOUT to control the type of prompt to display: PROMPT_PW or PROMPT_PW_USERID
SEC_VERIFY_PROC	PROMPT_PW_USERID	0	0	IT	Used with ForceCheckIn to control the type of prompt to display: PROMPT_PW or PROMPT_PW_USERID
SELECTPASSPW	AP	0	0		INTERNAL Use/any string value is ok
SEQUENTIAL_ANALYSIS_DUEDATE_RECALC			-1	IB	PostResults6 - Update duedates for Sequential Analyses
SHOWCRYSTALEXPORT		0	-1	IB	Show Crystal Report Export button on Webtop Crystal Report Viewer
SIGID		1	0		Quick Track - Signature ID Index
SSC_WARN_SAMPS		500	0	IL	Sample Selection Component, number of samples limit before possible delay warning.
STATE_MGMT_ON		0	0	IB	State Management On/Off
SYSTEMSHUTDOWN	(long)	0	0	IL	Period of time of inactivity before application closes
SYSTEMTIMEOUT	(long)	0	0	IL	Period of time of inactivity before application prompt for User Password before continuing

SYSMGROPT	SYSMGRVAL	SYSMGRLVAL	SYSMGRBVAL	SYSFIELDTYPE	Description
SYSWRITEPW	MASTER	0	0		Depricated
TC_PASSWORD	1234	0	0	IT	Total Chrome Password (Optional)
TC_USER_NAME	Mike Lehtola	0	0	IT	Total Chrome User Name (Optional)
TESTQUERYSIZE		200	0	IL	Database Query Parameter, how many ACODES to query in a single statement
UNLOCKMESSAGEINTERVAL		0	-1	IB	Unlock messages that are locked for period greater than interval
USE_62_PLATFORM		0	-1	IB	Use 6.2+ data storage and logic
USE_CONTAINER		0	-1	IB	Quick Track - Use Container
USE_IDS_MAIL	Use IDS Mail	0	-1	IB	Use IDS (3rd party component) email interface
USER_SET_LOC_OWN	Allow users to set location ownerships	0	-1	IB	Sample Modify - User can change owner
WARN_DUE_COLOR	YELLOW	0	0	IT	Backlog - Warning Color
WARN_DUE_HOURS		3	0	IL	Backlog - Warning Hours
USE_CALCULATION_ENGINE		0	-1	IB	If True then use LWCalculationEngine for calculation otherwise use Result Entry calculation logic for calculation
RECALC		0	0	IB	If True then automatic recalculation of results when input values are changed. Otherwise do not perform recalculation.

SYSMGROPT	SYSMGRVAL	SYSMGRLVAL	SYSMGRBVAL	SYSFIELDTYPE	Description
ADMINISTRATOR_EMAIL		0	0	IT	0
ALLOW_ANLCODE_DELETION	ALLOW_ANLCODE_DELETION	0	-1	IB	0
ALLOW_LOCCODE_DELETION	ALLOW_LOCCODE_DELETION	0	-1	IB	0
ALLOW_RAW_RESULT_MANUAL_ENTRY		0	0	IB	0
AUDIT_REASON_LATEST_AT_TOP	Allow latest reason to displayed at top.	0	-1	IB	0
DUE_COLOR_FONT	WHITE	0	0	IT	0
EXPORT_SAMP_SPEC		0	-1	IB	0
OVERDUE_COLOR_FONT	BLACK	0	0	IT	0
SHOW_STATUS	PS status	0	-1	IB	0
STATISTICAL_SIGFIG_VALUE		3	0	IL	0
USE_COL_SUB_TIME		0	0	IB	0
WARN_DUE_COLOR_FONT	BLACK	0	0	IT	0
WEB_REPORT_PATH	C:\LABWORKS\LW_REPORT	0	0	IT	0

Appendix B: The Complete List of LABWORKS.INI Settings

Module Name	Setting Name	Setting Value	Description
ALOGINO	LOGACTIVE	YES	specifies event and resource logging to lwuserpath\alogino.log
	PROJECT_ANALYTE_GROUP	PROJ	Specifies a project analyte reporting group for default (LCOD, SCOL, PORD, PROJ, USXX) Enterprise with sql compliant databases only
Analysis_Selection	FIELD	REF	Optional fields to display in analysis selection dialog (anlysis code and name are standard) Options are: DEPT (department) REF (reference) MDL UNIT PRICE
Analyte_Limiting	Remove	R_	This section allows you to remove test codes from the test selection in Zip login, Single sample login and Multi sample login. Remove = R_ - will remove any analysis code that starts with R_ Example: Remove = R_ -- will not list R_CD or R_AS
AUTO_MC_RLT_SPEC_KEYS			Specify function key (F2-F12) string substitutions for help topic "auto_mc_rlt_spec" keys are accessed by typing Ctrl-Fx combination at prompt key Ctrl+F1 is reserved for displaying current substitutions
BARTENDER_LABELS	LABEL_TYPE	1	This section is for labels using bartender and COC's using Access 'Label types:- 1 = all analyses on one record 2 = one record for each department 3 = one record for each bottletype. 4 = one record for each pscontainer 5 = one field for each analysis department
	SHOW_PRINT_DIALOG	NO	Setting for display print dialog.
CertOfAnalysis	COA_VERSION	NEW	Default settings for certificate of analysis option (DOS version) COA_VERSION specify as : NEW or OLD (default is NEW)
	CUST_BASE_DEFAULT	PROJ	Specifies a .cbf for default (LCOD, SCOL, PORD, PROJ, USXX)
	LOTNUMBER		How to build a lot number (NONE / BASF or SCOL / BASF1 / US01 / US02 / default)

Module Name	Setting Name	Setting Value	Description
			NONE (no lot number) SCOL or BASF (use sample collector) SIDN (use sample number as lot number) US01 (user field 1) US02 (user field 2) BASF1 (ponumber-project-coldate) default (numeric_part_of_sampno-project-coldate)
	PRINT_VIOL_WARNING	NO	Print notice of violation on standard format coa (yes / no (default)
	DISABLE_ON_VIO	NO	Disable printing of coa if violation (yes / no default)
	MULTISAMPLE		
	RESULT_REVIEW		Review / modify results on screen before printing coa
	HEADER_SAMP	LOWEST	Specify sample for coa header info if multisample - LOWEST (default) / HIGHEST / SELECT
	VIOL_IF_BLANK	Yes	Is a blank (empty or pending) result for a numeric or non-numeric spec a violation? (Yes / No Default is Yes)
	VIOL_IF_NOT_NUMBER	Yes	Is a non-numeric result for a numeric spec a violation? (Yes / No Default is Yes)
	CUSTOM_PREVIEW	NO	Preview custom format coa before printing (default = NO)
	ENVELOPE	NO	Pause for envelope printing (hp laser only) (default = NO YES or #)
	ENVELOPE_BIN	6	Printer bin for feeding envelopes (if ENVELOPE = YSE) 5 if automatic envelope loading is available on printer 6 if envelopes are manually fed into printer (default)
	ENVELOPE_SIZE	20	Envelope size option 20 Envelope #10 (4 1/8 x 9 1/2) default 19 Envelope #9 (3 7/8 x 8 7/8)
	METREFA		Method reference prefix substitutions for COA specify as: METREFx = PREFIX

Module Name	Setting Name	Setting Value	Description
			example: METREFA = ASTM- (changes "A" to "ASTM-")
	HIST_POINTS	10	Number of most recent results points to display default is 10 min is 3 max is 100
	HIST_SORT_ORDER	ASCENDING	Sort order for as shipped history display (ASCENDING / DESCENDING) default is ASCENDING
	EXCEL_COA_COPY_PATH		Path for saving excel workbook files after printing
Cross_Reference	SHOW_CURRENT_DATE	YES	Display current date when search routine is opened. If NO it will display last saved date in the search routine when re-run
Custody_Tracking	NULL_LOCATION	Not in custody	specifies report label for null (not entered) contained location (default = Unspecified)
	INITIAL_LOCATION	Receiving	Specifies default initial location for all container types
	PRELOGIN_LOCATION	Not in house	Specifies report description for location of sample containers before login Specifies initial locations for specific container codes which supercede default IL_@TCOL = Bact receiving IL_@WET = Chem receiving IL_@VOA = Orgn receiving IL_@VOAD = Orgn receiving IL_@VOAS = Orgn receiving IL_@VOAT = Orgn receiving
	INITIAL_REASON	New sample submitted to laboratory	Specifies reason to be associated with initial placement of all container types during login
	CURRENT_SAMPS	20	Specifies number of samples (default=20) considered current for tracking by sample designation of loccode or user field 1
	SELECT_CURRENT_SAMPS	NO	Specifies if automatically selects current samples (default=No)
	FORCE_MOVE_REASON	NO	Specifies if reason must be entered for moving sample container (default=No)
	SAVE_MOVE_REASON	YES	Specifies if entered reason for moving sample container is saved to list (default=Yes)
	SAVE_MOVE_DESTINATI	NO	Specifies if entered destination for sample container move is saved to list

Module Name	Setting Name	Setting Value	Description
	ON		(default=No)
DMR	SMP_TYP	COMP24 = 24HC	Entry to tell DMR to change COMP24 to 24HC
	Smp_Frq	Continuous = CONT	Smp_Frq = Continuous = CONT Smp_Frq = INSTAN = RCORD
Drive_S	PATH		Location of SQC data files... *.hed, *.dat, *.run (for LABWORKS DOS use only)
	NWA_VERSION	DOS	Specify the nwa application environment... WIN, DOS (default)
	LWSQCNET	NO	specifies if quality analyst is being run in a shared network mode ie, file qa.nwa in lwuserpath has data file path info => YES or go to lwsqcpath to launch quality analyst => NO (default)
	NO_UNITS	NO	Specifies if units should not appear with analyte name at tops of charts... YES / NO (default)
Email	EMAIL_SERVER		Specifies the name of the e-mail server in use at the workstation
	Exception_Report_Sender		Specifies the name of the exception report sender
EPA_CONTROL_CHART	MATRIX_FIELD	SAMPTYPE	The user field associated with the matrix
	NPOINTS	25	The default number of points in a chart group
	STDEV	3	The number of standard deviations for calculating control limits
	STARTDATE	1/1/95	The default starting data for looking for new datapoints
Exception_Report	'EXCEPTION_DATE	COLDATE	Specifies date/time to use for sorting and displaying exception trail Sql compliant, 32-bit only COLDATE (collection) SUBDATE (submittal) ANSDATE (anl start) ANEDATE (anl end) VIOLDATE (violation posting - default)
GPS	GPS_LAT_USERFIELD	11	The GPS data is stored in two user defined fields, these fields are defined below as GPS_LAT_USERFIELD and GPS_LON_USERFIELD. Set each keyword equal to the field number to use. Ex: GPS_LAT_USERFIELD = 3 will place the Latitude data in user field 3
	GPS_LON_USERFIELD	12	
History_Display	POINTS	10	Number of most recent results points to display default is 10 min is 3 max is 100

Module Name	Setting Name	Setting Value	Description
	SORTDATE	ANL	Date to sort history by (supercedes HISTSORT environmental variable) values (COL / SUB / ANL) coldate / subdate / analysis_date - default is COL
	TOP	EARLY	Display earliest or latest point at top (EARLY / LATE) default is EARLY
	RESULT2	NO	Display secondary result also (YES / NO) Enterprise only - default is NO
	QUALIFIER	NO	Display qualifier also (YES / NO) Enterprise only - default is NO
	PQL	NO	Display PQL also (YES / NO) - default is NO
	RAWResult	NO	Display raw result also (YES / NO) Enterprise only - default is NO
Interim_Result_Transfer	INTERIM_RESULT		Specifies if results being saved get sent to results directory for immediate transfer dos rev 3.40 and vb (dos 3.3 rev used l:lwks.ini for this feature)
	RESULT_EXPORT_FREQ	3	Settings for windows interim result export program
	RESULT_EXPORT_OUTFILE	Z:\DATA\LW300\ACCESS\SITEWIDE.DAT	
	RESULT_EXPORT_PROG	X:\APPS\LAB340\SITEWIDE.EXE	
Invoicing	PRN_FIELD	PROJ	Specifies additional fields to appear in invoices.prn line fields appear in order specified immediately after invoice number, before the other standard fields Options currently supported: PROJ, SCOL, SBAT, US01-Usxx PRN_FIELD = PROJ PRN_FIELD = SCOL PRN_FIELD = US01
LABWORKS_EXPLORER	SHOW_SQC	YES	
	SHOW_XREF	YES	
	SHOW_EXCEPTION	YES	
	SHOW_LOCDESCR	NO	Show Location Code Description
	SHOW_VALID_STATUS	NO	Show Pending Validation Status

Module Name	Setting Name	Setting Value	Description
Localization	TooltipLanguage		specify language for application tool tips TooltipLanguage = CtlNames TooltipLanguage = German
MC_Result_Import	DataSystem	GRF	Specifies installed data systems and defaults for windows mc result transfer. ' Examples: ' GRF - LABWORKS GRF format ' TURBO41 - PE Nelson TurboChrom 4.1 (.RST) ' TURBO40 - PE Nelson TurboChrom 4.0 (.RST) ' TURBO33 - PE Nelson TurboChrom 3.x (.RST) ' PEN2600 - PE Nelson 2600 (.ATB) ' QMASS - PE Q-Mass (.QPT) ' HPCHEM - HP PC Chemstation (.TXT) ' HPMSD - HP MSD (.REP) ' HPMS_CSV - HP MSD CSV (.CSV) ' HPMS_RR - HP MSD Reduced Result File (.RR) ' AC_SRP - AC PIONA/Reformulyzer (.SRP) ' ACSIMDIS - AC HT-750 SIMDIS (.REP) ' ACPIONA - AC PIONA Analyzer (.REP) ' ADA - Precision ADA (.???) ' DIONEXAI450 - Dionex AI-450 (.R??) ' EZCHROM - EZChrom (.SQC) ' LABQUEST - Tremetrics LabQuest (.SQC) ' VSTAR - Varian GC Star (.RPB) ' CHROMPERFECT - ChromPerfect (.Axx) ' MILLENNIUM - Waters Millennium ' VG_MINICHROM - VG MiniChrom
	ResultPath	c:\	

Module Name	Setting Name	Setting Value	Description
	DSType	GRF	
	NDResult	Not Detected	
	BDLResult	Below Det Lim	
	ResultType	A	
	DilFactor	1	
	RenameOption	N	
MC_Result_MDL	MDL_GRF	ANLIST	Specify whether to use instrument or analysis list MDL for result loading Options are: ANLIST, INSTR (default), HIGHER, LOWER specify as MDL_ + InstType = Option example: MDL_GRF = HIGHER this sets .grf file imports to uses the higher mdl
Process_Scheduler	SECURITY_MODE	GLOBAL	Settings for Security Mode GLOBAL or DISCRETE
	PREVIEW_TIME	59	Settings for optional process scheduler Preview time = # of minutes before sample requested time sample displays in list
	PREVIEW_HOURS_DAILY	12	Preview_Hours_Daily = Number of hours prior to schedule hourly samples will be displayed, max = 23
	PREVIEW_DAYS_WEEKLY	4	Preview_Days_Weekly = Number of days prior to schedule weekly samples will be displayed, max = 6
	PREVIEW_DAYS_MONTHLY	20	Preview_Days_Monthly = Number of days prior to schedule monthly samples will be displayed, max = 28
	RESTRICT_PREVIEW	NO	Restrict Preview for pre/logged in samples, Show all samples or Only Samples with request time in preview window
	REFRESH_INTERVAL	5	Refresh Interval is frequency data is updated
	RETAIN_WINDOW	1	Retain Window the number of hours a sample is displayed after Production OK
	VALIDATION_TYPE	SAMPLE	Validation Type, NONE = results finished only, SAMPLE = sample validation, LAB = manual lab validation
	ADD_TIMED_SAMPLES	YES	ADD_TIMED_SAMPLES, adds samples in time scheduled database, only one machine need to run this
	AUTOLOG_TIMED_SAMPLES	YES	AUTOLOG_TIMED_SAMPLES , automatically logs in Timed Samples

Module Name	Setting Name	Setting Value	Description
	AUTOLOG_PASSWORD	1	LOGIN PASSWORD , User password for Autologin of Samples
	PRIMARY_SYSTEM	PF-DELL650	PRIMARY_SYSTEM is the Labworks Machine ID for the system doing autologin and adding of timed samples
	PROGRESS_FORMAT	DEFAULT	Progress Report Format ' format for displaying progress report
	RESULT_FORMAT	RESULTS	Result Entry Format ' format for displaying results entry
	VIEW_PROGRESS	YES	View results format , if native then one sample at a time , No is native, YES = Progress Report, default
	USER1_DESC	Collect Type	User field descriptions / Leave blank to omit field
	USER2_DESC	Location	
	USER3_DESC	QC Type	
	USER4_DESC		
	NUSER1_DESC		
	NUSER2_DESC		
	USER1_ASSIGN	USR15	User field to Labworks Field Mappings
	USER2_ASSIGN	USR1	
	USER3_ASSIGN	USR20	
	USER4_ASSIGN		
	USER1_COMBO	YES	
	USER2_COMBO	NO	
	USER3_COMBO	NO	
	USER4_COMBO	NO	
	USER1_REQUIRED	NO	
	USER2_REQUIRED	NO	
	USER3_REQUIRED	NO	
	USER4_REQUIRED	NO	
	PROJ		
	DSCR		

Module Name	Setting Name	Setting Value	Description
	PORD		
	LBAT		
	RAD1		
	USR1 - USR20		
	LOGIN_PRIORITY	1	
	LOGIN_DEPT		
	LOGIN_NOPROMPT	YES	
	LOGIN_DEPT_FIELD	PROJ	
	LOGIN_INITIAL_STATUS	2	Options:- '0 = NOT READY '1 = READY TO COLLECT '1.5 = COLLECTED '1.8 = IN TRANSIT '2 = WAITING ANALYSIS
	LOGIN_USELRF	YES	
	SHOW_DESC	YES	Show Description displays the location code description instead of the location code in the main display.
	SHOW_REQ	YES	
	SHOW_COLOK	NO	
	SHOW_COL	NO	
	SHOW_WAITINGANALYSIS	NO	
	SHOW_ANALYSISCOMPLETE	NO	
	SHOW_LABOK	NO	
	SHOW_PRODOK	NO	
	SHOW_VIOLATION	YES	
	SHOW_PSID	NO	

Module Name	Setting Name	Setting Value	Description
	TOP	LATE	Display earliest or latest point at top (EARLY / LATE) default is EARLY
	USECOLLECTED	NO	Uses status collected between ready to collect and login
	LOGIN_AT_READYTOCOLLECT	NO	Logs sample in at Ready to collect for label printing
	COLLECTED_IN_BACKLOG	NO	Show Collected samples in Backlog, Default is NO, Sample is inactive until lab receipt
	SHOW_COLLECTED	NO	
	REQUIRE_PRODUCTIONOK	YES	REQUIRE_PRODUCTIONOK default is YES, Must be reviewed by requester, NO and Last status is LABOK
	DEFAULT_NO_LABELS	1	Default Number of Labels
	NO_PS_PROMPTS	NO	Eliminate Login OK Prompts / make barcoding easier
	SHOW_BARCODE	YES	Show Barcode Enhancements
	USE_IN_TRANSIT	NO	Include Status Transit to Lab
	USE_PSCONTAINER	NO	Use Bottle/Analysis Definition in PSCONTAINER table
	USE_BARTENDER	NO	Use Bartender Labeling Software
	PSAUDIT_DAYS_TO_KEEP	30	Days of Audit Trail to keep - Default is 30
	LEFT_CLICK	NO	For pen computers used left or right click
	COLLECTOR_REQUIRED	YES	Sample Collector is required by default when assigning LABWORKS SampleID Default COLLECTOR_REQUIRED = YES
	SHOW_COC	YES	Show Coc
	USE_MULTILOG	YES	Use Multi-Sample login for Sample Login
	USE_SINGLELOG	YES	Use Single-Sample login for Sample Login
	SHOW_PRELOGBATCH	NO	
	SHOW_LBPROMPT	YES	
	CUSTOM_LOGIN_RECORD_GROUP	SAMNUMB.EXE	To create a custom login record group set "Custom_Login_Record_Group" to the exe program name. be sure to include to entire path to the program.
	SHOW_LOGBATCH	YES	Show Login Record Group.
	SHOW_COLLECTION_DE	YES	Prompt for Collection information at login time, when used with

Module Name	Setting Name	Setting Value	Description
	TAIL		login_ready_collect=yes
	SHOW_SUBMIT_TIME	YES	prompt for sample submit time
	EDIT_LOCDESCR	YES	Permit editing of Location Description Field
	USE_ACCESS_FOR_COC	YES	Use access reporting for COC
	MAKE_PDA_FILE	YES	Create pda file YES/NO/ASK
	PDA_PATH	C:\LWUSER\PDA	Folder to put downloading PDA files to:
	PDA_ANALYSIS_ORDER	ANUM	Order for analyses for PDA, ANUM or ALPHA
	PDA_EXCEL_FORMAT	YES	Output PDA file in Excel Format
	SHOW_MASTER_STATUS	YES	Enable/Disable Master Status Display
	HISTORY_FROM_PICKLIST	YES	
	QAQC_BATCH_CREATION_OPTION	0	QAQC Batch Creation Option QAQC_BATCH_CREATION_OPTION = 0 '0 = QABatching6 '1 = LWQAQCBatchBuilder
	SHOW_MCSELECT	YES	Display MC Select menu item in Process Scheduler if flag SHOW_MCSELECT configured as YES.
Progress_Report	CURRENT_SAMPS	20	Specifies number of samples (default=20) considered current for progress report by sample designation of loccode or user field 1
	SELECT_CURRENT_SAMPS	YES	Specifies if automatically selects current samples (default=No)
	SAMP_DESG_ENTER_ACTION	OK	This setting specifies program response to enter key in sample designation text box OK (default) - pushes OK button NONE - no action SPACE - adds space to end of currently entered text used to allow multiple barcode scans into text box before loading samples SAMP_DESG_ENTER_ACTION = SPACE/NONE/OK
QA Batching	DEFAULT_BATCH_SIZE	15	Sets default batch size for all analyses (if not set here default is 10 samples)
	BATCH_SIZE_CD	20	Sets default batch size for a particular analysis

Module Name	Setting Name	Setting Value	Description
			'BATCH_SIZE_analcode = nn
RECALC_GENERAL_SPECIFICATION	MINIMUM_NO_OF_POINTS	5	MINIMUM_NO_OF_POINTS refers to the minimum number of results needed to recalculation specifications. Default is 5.
	NUMBER_OF_MOST_RECENT_POINTS	10	NUMBER_OF_MOST_RECENT_POINTS refers to the number of most recent points used to recalculate specifications. Default is 10
	PRECISION_CALCULATE_UPPER_WARNING	YES	PRECISION_CALCULATE_UPPER_WARNING indicates if the upper warning for precision is to be recalculated. PRECISION_UPPER_WARNING_SIGMA is the number of sigmas to recalculate the upper warning from the mean. Ex. - PRECISION_CALCULATE_UPPER_WARNING=YES. Default is Yes. Ex. - PRECISION_UPPER_WARNING_SIGMA=2. Default is 2.
	PRECISION_UPPER_WARNING_SIGMA	2	
	PRECISION_CALCULATE_UPPER_LIMIT	YES	PRECISION_CALCULATE_UPPER_LIMIT indicates if the upper warning for precision is to be recalculated. PRECISION_UPPER_LIMIT_SIGMA is the number of sigmas used to recalculate the upper warning from the mean. Ex. - PRECISION_CALCULATE_UPPER_LIMIT=YES. Default is Yes. Ex. - PRECISION_UPPER_LIMIT_SIGMA=2. Default is 2.
	PRECISION_UPPER_LIMIT_SIGMA	3	
	RECOVERY_CALCULATE_WARNINGS	YES	RECOVERY_CALCULATE_WARNINGS indicates if the upper and lower warnings are to be recalculated. RECOVERY_WARNING_SIGMA is the number of sigmas to used recalculate the upper and lower warnings Ex. - RECOVERY_CALCULATE_WARNINGS=YES. Default is Yes. Ex. - RECOVERY_WARNING_SIGMA=2. Default is 2.
	RECOVERY_WARNING_SIGMA	2	

Module Name	Setting Name	Setting Value	Description
	RECOVERY_CALCULATE_LIMITS	YES	RECOVERY_CALCULATE_LIMITS indicates if the upper and lower limits are to be recalculated. RECOVERY_LIMIT_SIGMA is the number of sigmas to used recalculate the upper and lower limits Ex. - RECOVERY_CALCULATE_LIMITS=YES. Default is Yes. Ex. - RECOVERY_LIMITS_SIGMA=3. Default is 3.
	RECOVERY_LIMIT_SIGMA	3	
	CALCULATE_SIGMA	SAMPLE	CALCULATE_SIGMA indicates to use either Sample or Population Statistics Ex. CALCULATE_SIGMA=POPULATION
Result_Conditioning	RULENAME	ANLCODE	specify result conditioning rule names for specific analysis codes when loading data from interfaced instruments specify as: RULENAME = ANLCODE
ResultScan	LOGACTIVE	YES	Post log entries during operation (default is 'NO')
	LOAD_FREQ	3	How often to load data in minutes
	LOAD_MODE	SINGLE	loading modes: "MULTI" parse all result files at once then post to database (default) "SINGLE" parse and post each result file one by one
	SCAN_INSTR	IAI_ASCII C:\data\lw300\user *.rlt	'SCAN_INSTR = inst_code data_path rlt_file_template addtest (optional) nochange (optional) norename (optional) '"NORENAME" option applies only to single load mode causes file to be renamed AFTER posting rather than before parsing (default) ' Philips PW-1480 XRF (imports PW1480 results from c:\results\xray*.lst files with result changes but no work order additions) 'SCAN_INSTR = IAI_PHILIPS_X40 D:\DATA1\ *.SAV 'SCAN_INSTR = IAI_CANNON_VISC D:\DATA2\ CAVDAT.DAT
	SCAN_INSTR	IAI_ASCII C:\data\lw300\user *.asc	

Module Name	Setting Name	Setting Value	Description
RouteSheets	PRINTER		variable - printer defaults to LPT1:
	BARCODE		barcode defaults to ON unless set to OFF, supercedes RBARCODE environmental variable
	SPECIAL_INFO	NO	Include special info fields, defaults to NO unless set to YES (new)
	COMMENT_LINES		free text comment to include, defaults to 0 unless set to positive number new)
Sample_Modify	Auto_Validate_Rule	No_Violations	Should sample be auto validated when no test is pending
	CURRENT_SAMPS	14	Specifies number of samples (default=20) considered current for modification by sample designation of loccode or user field 1
	SAMP_DESG_ENTER_ACTION	OK	This setting specifies program response to enter key in sample designation text box OK (default) - pushes OK button NONE - no action SPACE - adds space to end of currently entered text used to allow multiple barcode scans into text box before loading samples SAMP_DESG_ENTER_ACTION = SPACE/NONE/OK
	ADD_TEST_BASEDATE	LOCCODE	Specifies the base date option to use for calculating due dates of analyses added during sample modification 'ADD_TEST_BASEDATE = LOCCODE (default) COLDATE SUBDATE CURDATE
	SINGLE_SAMPLE_USER_PROG	SHELTEST	Specifies program to run after each modified sample in single sample modify include full path and extension or program path and .exe are assumed
	MULTI_SAMPLE_USER_PROG	SHELTEST	Specifies program to run after multisample modifications are saved include full path and extension or program path and .exe are assumed
SampleReceipt	SPECIAL_INFO		Receipt printing parameters for Windows Include special info fields, defaults to NO unless set to YES (new)
	COMMENT_LINES		Receipt printing parameters for Windows Free text comment to include, defaults to 0 unless set to positive number
SC_Result_Import	INSTRTYPE_NEG	ZERO	If instrument reports negative values change negative result to 0
	INSTRTYPE_NEG	OMIT	If instrument reports negative values don't report negative result
	INSTRTYPE_NEG	MDL	If instrument reports negative values report negative result as default MDL value for analysis

Module Name	Setting Name	Setting Value	Description
	INSTRTYPE_NEG	<MDL	If instrument reports negative values report negative result as 'less than ' default MDL value
	INSTRTYPE_NEG	RESULT	If instrument reports negative values report negative result as it is
	INSTRTYPE_NEG	other	If instrument reports negative values report negative result as 'other'
	INSTRTYPE_COL_WIDTH	14	how wide to make result review columns in spreadsheet (default = 8 chars)
Serial_Instruments	ANL_-TSSTARE	BALANCE_01	ANL & Analysis Name = Instrument_Name (in InstrumentLibrary) for results entry
	ANL_-TSSFWT	BALANCE_01	ANL & Analysis Name = Instrument_Name (in InstrumentLibrary) for results entry
	ANL_TAREWT	BALANCE_01	ANL & Analysis Name = Instrument_Name (in InstrumentLibrary) for results entry
	ANL_FINALWT	BALANCE_01	ANL & Analysis Name = Instrument_Name (in InstrumentLibrary) for results entry
	ANL_PH	PHMETER	ANL & Analysis Name = Instrument_Name (in InstrumentLibrary) for results entry
	SEQ_WEIGHT_PE_5100	BALANCE_05	SEQ_WEIGHT_ & INSTR CODE = Instrument_Name (in InstrumentLibrary) for sequence loading
Spread_Result_Entry	POST_RESULT_PROGRAM	SHELTEST	Enables custom program to run after results are saved, Path defaults to programs path, extension defaults to .EXE POST_RESULT_PROGRAM = ProgramName ProgramParams (optional)
	SCND	Not detected	Specifies result labels to be used for 0 and quantities below mdl in single component instrument imports, replaces environmental variables SCND and SCMDL used in DOS for same purpose
	SCBDL	Present < MDL	Specifies result labels to be used for 0 and quantities below mdl in single component instrument imports, replaces environmental variables SCND and SCMDL used in DOS for same purpose
	MCND	Not detected	Specifies result labels to be used for 0 and quantities below mdl in multicomponent instrument imports and defaulted manual entries in results entry (RENTERO.EXE rev 4 only), replaces environmental variables used in earlier versions for same purpose
	MCBDL	Present < MDL	Specifies result labels to be used for 0 and quantities below mdl in multicomponent instrument imports and defaulted manual entries in results entry (RENTERO.EXE rev 4 only), replaces environmental variables used in earlier versions for same purpose
	AUTO_VALIDATE_RULE		Specifies option for auto validation of sample upon completion of results entry, Set to 'NO_VIOLATIONS' to automatically validate samples with no exceptions
	CURRENT_SAMPS	20	This setting default to 20 and controls the number of the most recent samples considered current when selecting them by sample designation and entering

Module Name	Setting Name	Setting Value	Description
			either a location code or user field 1 value.
	SELECT_CURRENT_SAMPS	NO	This setting controls whether results entry will automatically bring up the current samples for a location code or user field 1 value, bypassing the sample selection screen.
	SAMP_DESG_ENTER_ACTION	OK	This setting specifies program response to enter key in sample designation text box, OK (default) - pushes OK button, NONE - no action, SPACE - adds space to end of currently entered text, used to allow multiple barcode scans into text box before loading samples “SAMP_DESG_ENTER_ACTION = SPACE/NONE/OK”
	ADJUST_PROJECT_REPORT_LIMITS	NO	specifies adjusting project analyte reporting (PAM) limits for analysis be dilution factor (sql compliant only). Values: Yes / No
	EXCEL_CALC_VISIBLE	YES	Specifies if Excel spreadsheet appears during calculations - YES (default) or NO
	SpecSound		Set SpecSound = ON to enable the playback of WAV files for spec violations (requires sound card and windows multimedia drivers)
	UpperSpec_Sound		Specify .WAV files for violation that may occur
	UpperWarning_Sound		Specify .WAV files for violation that may occur
	Target_Sound		Specify .WAV files for violation that may occur
	LowerWarning_Sound		Specify .WAV files for violation that may occur
	LowerSpec_Sound		Specify .WAV files for violation that may occur
	SortType	5	Change the sort type for tests based on the type of test code you create. 1 = Generic, 3 = Numeric, 5 = String (no case), 7 = String
Test_Sort	USER1	Seed	Only USER FIELDS support Incrementing.Each field must start with a seed. The seed will be used as the first number and will be incremented each time it is used.Seeds may include numbers and letters.Make sure you include enough digit positions in the seed to cover your needs. ex: if your seed is 1 then it will increment up to 9 and roll over back to 0. If you want to increment up to 10000 then make your seed equal to 00001.You can not use spaces at the beginning or end of the seed. Fields must be setup using the following syntax with the word Seed being replaced by your actual seed data. USER1 = AA-AAA-00091 USER2 = Seed
User_Field_Increments	'QAQC_SEARCH	ON	Parameters for windows sample validation program should a search be done for qa/qc results for validation samples (ON / OFF default is ON)

Module Name	Setting Name	Setting Value	Description
Validation	LOGACTIVE	YES	Specifies event logging to lwuserpath\validato.log
	SAMP_DESG_ENTER_ACTION	OK	This setting specifies program response to enter key in sample designation text box OK (default) - pushes OK button NONE - no action SPACE - adds space to end of currently entered text used to allow multiple barcode scans into text box before loading samples SAMP_DESG_ENTER_ACTION = SPACE/NONE/OK
	ACC_PATH	USER	ACC_PATH determines where the Access Report Writer support files will be located. Use USER for the user path, and DATA for the datapath. Any other desired path may be hard coded (i.e. C:\lwuser\) or if left blank, the datapath is used.
Windows_Desktop	Sound	OFF	Set Sound = ON to enable the playback of WAV files.
	Startup_Sound		Startup_Sound is the sound that will play when the desktop has first loaded, before the password prompt.
	Welcome_Sound		Welcome_Sound will play after the user has entered a correct password.
	Update_Time	5	Update_Time determines how often (in minutes) the Desktop should check for new Xref.dat files and changes to the location code list. Default time is 5 minutes - minimum time is 1 minute.
	Zip_Reports_Printer	1	Zip_Reports_Printer allows you to set the LPT port for zip advanced reports.
	MSWord_Null_Data_Replacement	"---"	
	UPDATE_HISTORY_FILE		To turn off saving entries into History Lists, set the following variable to No
Windows_Multi_Sample_Login	Button_Text		Text description to display on special button.Example: Button_Text = "Select Sample".
	Test_DueDate_BaseDate	COLDATE	To change the way due dates are calculated for tests, set this entry to One of the following: LOCCODE (default) COLDATE SUBDATE
	Special_Button		Special button assignment can be one of the following: "Sample Description", "Project Account Code", "User Field 1","User Field 2". Example: Special_Button = "Sample Description".
	Use_Login_Record_Group		Set "Use_Login_Record_Group" equal to "Yes" if you want to add/create a sample login group.
	CUSTOM_LOGIN_RECORD_GROUP		To create a custom login record group set "Custom_Login_Record_Group" to the exe program name. Be sure to include to entire path to the program.
	LOGIN_RECORD_FILE_TRANSFER		This entry is used to transport the login record group information from a custom program back to login. Setting it to a complete filespec will change the way the

Module Name	Setting Name	Setting Value	Description
			data is passed back to login. The filespec will be passed to the program as a command line parameter with " Result:" preceding the file name. If it is empty or missing, the transfer will take place as normal, through the keyboard.
	EXPORT_LOCATIONS		Setting this entry to YES will cause login to create a file containing the location codes that are about to be logged in.The filename will be passed as a command line parameter with "Export:" preceding the filename.
	FIELD1	Sample ID: = SIDN	Information to display when sample login completed. .
	FIELD2	Location Code: = LCOD	Information to display when sample login completed.
	FIELD3	Submit Date = SDAT	Information to display when sample login completed
	FIELD4	Due Date = DDAT	Information to display when sample login completed.
	FIELD5	User 1 = USR1	Information to display when sample login completed
	TDAT	Current Date	LABWORKS data item for Current Date.
	SIDN	Sample ID	LABWORKS data item for Sample ID
	LCOD	Location Code	LABWORKS data item for Location Code
	DSCR	Sample Description	LABWORKS data item for Sample Description
	SCOL	Sample Collector	LABWORKS data item for Sample Collector
	CDAT	Collection Date	LABWORKS data item for Collection Date
	CTIM	Collection Time	LABWORKS data item for Collection Time
	SDAT	Submit Date	LABWORKS data item for Submit Date
	STIM	Submit Time	LABWORKS data item for Submit Time
	DDAT	Due Date	LABWORKS data item for Due Date
	PORD	P.O. Number	LABWORKS data item for P.O. Number
	PROJ	Project Code	LABWORKS data item for Project Code
	RFOR	Report Format	LABWORKS data item for Report Format
	LUSR	Login User	LABWORKS data item for Login User
	LBAT	Login Batch	LABWORKS data item for Login Batch
	RAD1	Report Address 1	LABWORKS data item for Report Address 1
	RAD2	Report Address 2	LABWORKS data item for Report Address 2

Module Name	Setting Name	Setting Value	Description
	RAD3	Report Address 3	LABWORKS data item for Report Address 3
	RAD4	Report Address 4	LABWORKS data item for Report Address 4
	RAD5	Report Address 5	LABWORKS data item for Report Address 5
	BAD1	Billing Address 1	LABWORKS data item for Billing Address 1
	BAD2	Billing Address 2	LABWORKS data item for Billing Address 2
	BAD3	Billing Address 3	LABWORKS data item for Billing Address 3
	BAD4	Billing Address 4	LABWORKS data item for Billing Address 4
	BAD5	Billing Address 5	LABWORKS data item for Billing Address 5
	USR1	User Field 1	LABWORKS data item for User Field 1
	USR2	User Field 2	LABWORKS data item for User Field 2
	USR3	User Field 3	LABWORKS data item for User Field 3
	USR4	User Field 4	LABWORKS data item for User Field 4
	USR5	User Field 5	LABWORKS data item for User Field 5
	COM1	User Field 5	LABWORKS data item for Comment Line 1
	COM2	Comment Line 2	LABWORKS data item for Comment Line 2
	COM3	Comment Line 3	LABWORKS data item for Comment Line 3
	COM4	Comment Line 4	LABWORKS data item for Comment Line 4
	COM5	Comment Line 5	LABWORKS data item for Comment Line 5
	COM6	Comment Line 6	LABWORKS data item for Comment Line 6
	COM7	Comment Line 7	LABWORKS data item for Comment Line 7
	COM8	Comment Line 8	LABWORKS data item for Comment Line 8
	MAX_Samples	50	Indicates to MultiLog the maximum number of samples that can be logged in.
	ENTER_KEY_ACTION		ENTER_KEY_ACTION determines the movement of the cell pointer when the ENTER key is pressed. current options are "DOWN" and "NEXT".
	FIELD_LOCKED		FIELD_LOCKED allows data to only be selected using the history lists.Use the four character codes to specify which field is to be locked. In order for the lock to take place the field must be placed on the spreadsheet. USE USXX, (US01) for userfields

Module Name	Setting Name	Setting Value	Description
	FIELDS_FROZEN		FIELDS_FROZEN allows data to remain on screen when scrolling horizontally. Set the value equal to the number of rows to be frozen starting from the first editable column (not the row number column).
	Select_Field	USERFLDS.USER1	These are the fields for selecting location codes using the "Custom Selection Screen" up to five fields can be used. The first column is the field number, the second column is the description to display on screen, and the third column is the name of the field contained in the database using the syntax "TableName.FieldName" If the field name contains spaces then it must be surrounded with brackets "[]" such as USERFLDS.[Analysis number] ' Field Counter - Description - TableName.FieldName
	Select_Field	USERFLDS.USER2	
	Select_Field	LOCLIST.DESCR	
	ADD_TO_PROCESS_SCHEDULER	YES	Turn on adding samples to process scheduler
	USE_STORED_COLLECTION_TIMES	NO	Use times stored in pre-login file
	UPDATE_HISTORY_FILE		To turn saving entries into History Lists, set the following variable to No
Windows_Single_Sample_Login	Button_Text		Text description to display on special button. Example: Button_Text = "Select Sample".
	Special_Button		Special button assignment can be one of the following: "Sample Description", "Project Account Code", "User Field 1", "User Field 2". Example: Special_Button = "Sample Description".
	Use_Login_Record_Group	NO	Set "Use_Login_Record_Group" equal to "Yes" if you want to add/create a sample login group.
	CUSTOM_LOGIN_RECORD_GROUP		To create a custom login record group set "Custom_Login_Record_Group" to the exe program name.Be sure to include to entire path to the program.
	FIELD1 = Sample ID	SIDN	5 fields can be displayed when sample login completed.
	FIELD2 = Location Code	LCOD	5 fields can be displayed when sample login completed.
	FIELD3 = Submit Date	SDAT	5 fields can be displayed when sample login completed.
	FIELD4 = Due Date	DDAT	5 fields can be displayed when sample login completed.
	FIELD5 = User 1	USR1	5 fields can be displayed when sample login completed.
	ADD_TO_PROCESS_SCH		Turn on adding samples to process scheduler YES/NO

Module Name	Setting Name	Setting Value	Description
	EDULER		
	FIELD1		KEY WORDS or the line syntax must not be changed. All key words must be followed by an equal (=) sign.
Windows_Standard_Labels	FIELD2		
	FIELD3		
	FIELD4		
	FIELD5		
	PRINTER_NAME		
	BARCODES		Barcodes and barcode text can be toggled on or off depending on user preference.The parameter BARCODES turns the barcode on and off on the label, and the parameter 'BARCODE_TEXT turns the caption under a barcode on or off. Note that the barcode option for LABWORKS must be installed for these parameters to work. Default settings are ON for both these parameters.
	BARCODE_TEXT		Barcodes and barcode text can be toggled on or off depending on user preference.The parameter BARCODES turns the barcode on and off on the label, and the parameter 'BARCODE_TEXT turns the caption under a barcode on or off. Note that the barcode option for LABWORKS must be installed for these parameters to work. Default settings are ON for both these parameters.
	BARCODE	ON	Work sheet printing parameters for *** WINDOWS DESKTOP *** barcode defaults to ON unless set to OFF, supercedes WSBARCODE environmental variable (for STANDARD format worksheets only)
Worksheets	WS_SORT		sort options = 'ODUEDATE', 'SDUEDATE', 'ADUEDATE', 'SAMPNO', 'COLDATE', 'SUBDATE' Note: sort options do not overide sorting specified in Windows Backlog report when worksheets are printed there but do control which due date is printed ' WS_SORT = ODUEDATE - Overall Due Data.
	QA_SAMPLE_SUFFIX	None	how to print sample id on worksheets for qa parameters DESCR (default - prints SampleID + BLANK, DUPLICATE, PRECISION, etc.) ABBR (prints SampleID + _B, _D, _P, etc.) NONE (prints SampleID only)
	PORT		Select the com port number being used by the Zebra Printer. Valid options are: 1 or 2.
Zebra_Barcode_Printer			