

MSC-LIMS™ *Insights*

The source for news and tips of interest to users of MSC-LIMS,
an affordable laboratory information management system for small labs.

Issue No. 16

August 2011

Welcome

Welcome to **MSC-LIMS *Insights***.

This newsletter will help current MSC-LIMS users get the most out of their software, and will complement the product literature and downloadable demo that prospective users can find on our web site at www.msc-lims.com.



Join our mailing list for more information. Sign up at www.msc-lims.com/lims/maillist.html.

This newsletter is for and about MSC-LIMS users. We welcome your comments, and your suggestions for topics you would like to see addressed in upcoming issues. Please send your thoughts to newsletter@msc-lims.com. ▲

Compact LimsData Regularly to Improve Performance

Your LimsData database steadily increases in size over time as you continue to add new data. You can improve overall system performance by regularly compacting the database. We recommend you compact LimsData at least once a month but preferably once a week. In addition to minimizing the size of the LimsData database, compacting will defragment the file by rewriting it to contiguous space on the disk. Compacting also rewrites the database's tables in primary key order, recreates all table indexes, and updates database statistics, all of which can improve overall system performance and query performance in particular.

To compact your LimsData database, log on to the LIMS as a member of the Admins security role. Use the View Current Users option on the Admin menu to make sure you are the only one logged on to the LIMS. You must have exclusive control of the database to compact. Now use the Compact LimsData option on the Admin menu to exit the LIMS and begin the process. Follow the onscreen prompts and log on again to compact the database. Once database compacting completes all users can start the LIMS normally. ▲

In this Issue

<i>Welcome</i>	1
<i>Compact LimsData Regularly to Improve Performance</i>	1
<i>From the Developer</i>	2
<i>Microsoft Acknowledges Poor Access Performance on Windows 7</i>	2
<i>Creating a Basic Excel Export Template</i>	3
<i>Notes from Technical Support</i>	6
Query by Sample Status	6
Incomplete Samples with All Results	6
Sending Preliminary Reports with Messaging	7
<i>For Customers Only</i>	7
Knowledge Base	7
<i>Contact Us</i>	7

From the Developer

The art of software development can produce many challenges. Particularly trying are issues beyond our control such as those arising from the Microsoft Windows operating system or Microsoft Office, which are frequently the topic of technical support calls or emails. For example, one customer rolling out new Windows 7 workstations noticed slower query performance. We spent many days running tests and confirmed that an identical query on a local database is consistently fast on Windows XP but erratic and often slow on Windows 7. Microsoft has acknowledged the problem lies within the Jet database engine used by Microsoft Access and we are waiting for their solution. See *Microsoft Acknowledges Poor Access Performance on Windows 7* for more information.

When we recently began porting our software for MSC-LIMS 4.0, our next major release, we found several serious flaws in Access 2010. We discovered, and Microsoft has acknowledged, that all access keys (i.e. shortcut keystrokes using Alt plus the underlined letter) no longer work in any Access 2010 screen that uses tabs to organize the screen's fields. For many MSC-LIMS users, particularly those doing rapid keyboard data entry, it is crucial that familiar keystrokes continue to work with each MSC LIMS version so a solution is imperative.

In spite of these and other obstacles, we will persevere and do whatever it takes to keep MSC LIMS working effectively for our users and to move our product to its next Access platform. We are up to the challenge and we will keep you posted on our efforts.

Finally, please note our revised mailing address and update your records accordingly. All lines in our new address are required to ensure proper mail delivery:

Mountain States Consulting, LLC
970 W Broadway #471
PO Box 30000
Jackson, WY 83002



Rick Collard is the founder of Mountain States Consulting and the principal developer of the MSC-LIMS software. You can reach Rick by email at rcollard@msc-lims.com.

Microsoft Acknowledges Poor Access Performance on Windows 7

If you are considering migrating your MSC-LIMS workstations or workstations running any Access-based applications to Windows 7, you may want to postpone until Microsoft fixes this performance issue.

One MSC-LIMS installation recently noticed slow LIMS queries only on their new Windows 7 workstations. Their remaining Windows XP workstations continued to perform the identical queries with their normal fast speed. With a copy of their database, we were able to duplicate the slow queries on Windows 7 using both Access 2002 and Access 2010. After several days of testing it was apparent there was a serious problem with the Microsoft Access Jet database engine on Windows 7.

We logged a support request with Microsoft and provided a subset of the customer's database with a simple VBA procedure that timed repeated queries of a set of LIMS samples. The Microsoft support technician quickly duplicated the problem. Our VBA timing procedure demonstrated that the identical query ran consistently fast on Windows XP but was erratic and slow on Windows 7. Within a few weeks, we received notice that the support case was transferred to the Microsoft Access Escalation team. Later, a Senior Support Escalation Engineer acknowledged the problem:

(Continued on next page)

(Continued from previous page)

We have identified an issue in the Access source code that is contributing to the cause of the intermittent query performance problem you see in Microsoft Access when using Windows 7 machines.

Due to efficiencies built into the newer operating systems, as well as efficiencies in many newer machines, resource management is vastly different from the time when the database engine code for Microsoft Access was developed. The Microsoft Jet Database Engine code, on which the Access 2007 and Access 2010 Database Engine was based, was developed to work around resource limitations in older operating systems. Due to the recent changes in resource management, we are finding that the older approach to resource management is resulting in the intermittent query performance issues. Our developers are currently exploring an approach to resolving the

issue in the source code for the Access Database Engine.

With hundreds of MSC-LIMS users potentially affected by this problem, we requested the Microsoft Access developers consider a fix that could be provided to any MSC-LIMS installations with Windows 7 systems. We recently received the following update from Microsoft's Senior Support Escalation Engineer:

I hope you did not think I forgot about you. I just want to let you know that a fix for this issue is scheduled to be available late August/early September. I will update you if anything changes with regard to the timeframe. Thanks for your continued patience!

We are encouraged by Microsoft's progress and look forward to their fix. You can find the latest on the Windows 7 performance problem by following discussions we started at [Utter Access Forums](#) and Microsoft's [Access for Developers Forum](#). ▲

Creating a Basic Excel Export Template

This is the second in a series of *Insights* articles designed to provide a better understanding of the macros used in Excel templates. In our first installment (see Introduction to MSC-LIMS' Excel VBA Macros in [MSC-LIMS Insights No. 15](#)) we covered the location and names of the required macros in both MSC-LIMS export and import templates. In this article, we will expand on the capabilities of an export template's macros with a simple example.

New users implementing MSC-LIMS to automate their former paper-based records management often want a report equivalent to their original sample logbook. While a user-defined report (UDR) in MSC-LIMS is one solution, an Excel template offers better control of the report's layout and appearance. We will create a simple Excel sample log report to produce a record of samples logged daily, weekly, or for any other timeframe. The sample log report can list any sample characteristics such as sample ID, batch number, customer, project, collected date, etc. The report will not list any analyses or analytical results, as that will be the topic of a future article.

Often, the easiest way to create a new template is to copy and modify an existing template. We encourage you to explore the example templates in folder C:\MSC-LIMS\Examples. However, our goal in this series of *Insights* articles is to learn more about a template's macros so we will build our report from scratch.

As we learned in the previous article, every export template used with MSC-LIMS must have a LIMSData worksheet with macros BeforeTransferFromLIMS and AfterTransferFromLIMS. Since the example MSC-LIMS Export Template in folder C:\MSC-LIMS\Examples\Excel Export Templates, has only a LIMSData sheet with empty versions of the two required macros, we can use it as the basis for our template. Begin by making a copy of example file MSC-LIMS Export Template.xlt then rename the file Sample Log.xlt. Note that newer versions of Excel support additional file types and extensions. However, for compatibility you should continue to save all your MSC-LIMS templates in Excel 97-2003 format with an XLT extension.

(Continued on next page)

MSC-LIMS™ Insights

(Continued from previous page)

Open the Sample Log template, add a new worksheet named Report before the LIMSDData sheet, then close and save the template. Remember, to open any Excel template file (XLT extension) for editing either right-click the file in Windows Explorer and choose Open from the popup menu or open the file within Excel. Double-clicking an Excel template in Windows Explorer does not open the template file; rather it opens a new workbook created from the template.

We will design our template for MSC-LIMS' sample summary report. Query and preview a sample summary report containing several samples then use File | Export to MS Excel Template and select the new Sample Log template. In the resulting Excel workbook's LIMSDData worksheet, you will see all of the exported LIMS data with one row for each sample queried along with a header row of MSC-LIMS field names. Explore the data available for our report (see the example below).

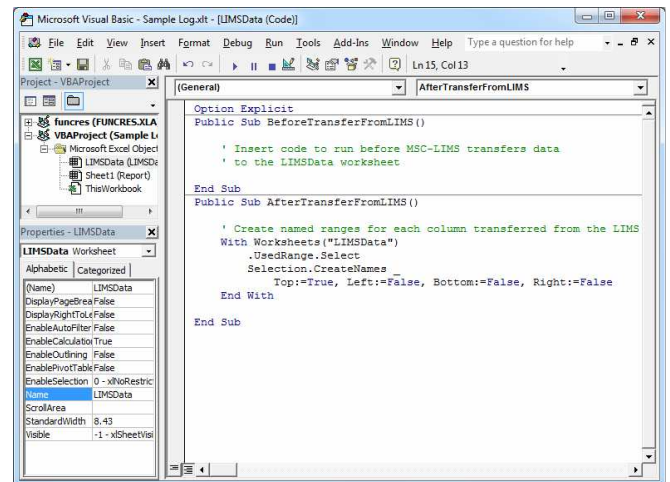
Project Name	Sample Type	Location	Sampler	Sample ID	Project ID	Sampler ID	Sample Type	Collected
Wastewater Grab	Site 1	Operators	100812L0C	6	15	27	8/12/2010	
Wastewater Grab	Influent	Operators	100812L0C	6	15	27	8/12/2010	
Wastewater Grab	Influent	Operators	100812L0C	6	15	27	8/12/2010	
Wastewater Grab	Influent	Operators	100812L0C	6	15	27	8/12/2010	
My Project N/A	N/A	Blow, Joe	100826181	17	21	31	8/26/2010	
My Project N/A	Site 1	Blow, Joe	100826181	17	21	31	8/26/2010	
My Project N/A	Site 3	Blow, Joe	100826181	17	21	31	8/26/2010	
My Project N/A	N/A	Blow, Joe	100828R04	17	21	31	8/28/2010	
My Project N/A	Site 1	Blow, Joe	100828R04	17	21	31	8/28/2010	
My Project N/A	Site 3	Blow, Joe	100828R04	17	21	31	8/28/2010	
Walk In	N/A	Blow, Joe	100828173	55	21	31	8/28/2010	
Walk In	N/A	Blow, Joe	100828173	55	21	31	8/28/2010	

Recall that MSC-LIMS runs macro BeforeTransferFromLIMS before writing the sample summary report's data to the LIMSDData sheet and macro AfterTransferFromLIMS after the data transfer. So that we may use simple Excel formulas to display sample data on our Report sheet that refer to LIMS data by field name, our AfterTransferFromLIMS will begin by creating named ranges for the data on the LIMSDData sheet using the row one labels for the names. For example, with named ranges we can use cell formulas like =INDEX(SampleID, 1) to display the first sample's ID and =INDEX(Project.Name, 2) to display the second sample's project.

Close the workbook without saving then open the Sample Log template. Use Alt+F11 to open the Visual Basic for Applications (VBA) editor. Double-click LIMSDData in the project explorer on the left then add the following lines to the AfterTransferFromLIMS macro:

```
' Create named ranges for each column transferred from the LIMS
With Worksheets("LIMSDData")
    .UsedRange.Select
    Selection.CreateNames _
        Top:=True, Left:=False, Bottom:=False,
Right:=False
End With
```

Use Debug | Compile VBA Project to check for errors. If there are no errors, close the VBA editor and return to the template. The screen below shows the current state of our macro. Note that lines that begin with an apostrophe (') and appear in green are simply comments used to add descriptive information about the macro's tasks.

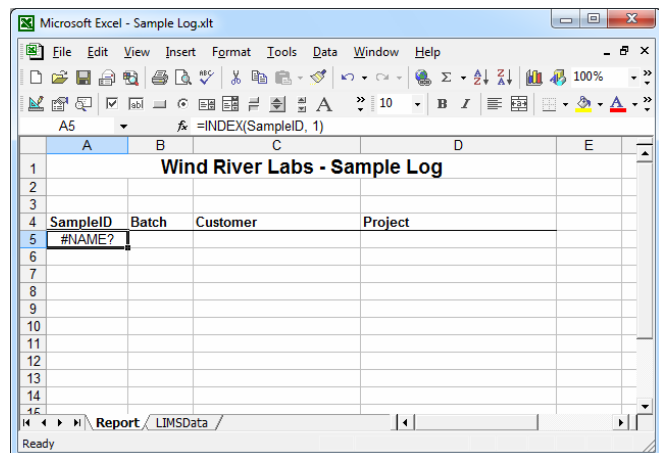


Switch to the Report sheet in the template and add an appropriate report title and column labels. So that Excel does not attempt to use our column labels instead of our named ranges in formulas use Tools | Options and disable the "Accept labels in formulas" setting on the Calculation tab. The example below shows column labels for Sample ID, Batch, Customer, and Project. In the cell directly below the Sample ID label, add the formula =INDEX(SampleID, 1) to test. We get the #NAME? error as expected because the SampleID named range does not exist and will only be created after our macro runs.

(Continued on next page)

MSC-LIMS™ Insights

(Continued from previous page)



Close the template saving your changes and export the sample summary report again. Switch to the Report sheet and notice that our cell with the test formula now correctly displays the sampleID for the first sample. We could copy our sample ID cell's formula to cells below changing the INDEX function's row number to 2, 3, etc. but that would be tedious since we would have to copy then edit the formula for the maximum number of samples we expect to ever export from the LIMS. An alternative is to change the formula so it is relative to the cell's current row. For example, since the sample ID formula is in cell A5, we can change it to =INDEX(SampleID, ROW() - 4). The ROW function returns the row number of the cell. When this formula is in cell A5 it is equivalent to =INDEX(SampleID, 1) and =INDEX(SampleID, 2) in cell A6. This formula is an improvement over our first attempt and it allows us to copy the formula to any number of rows below without editing. However, the formula will not have the desired result if we insert new title rows above row five. A better solution is to use a named range, as described in the following paragraph.

Following our example Report sheet above, select cells A5 through Z5, use Insert | Name | Define, enter the name "OneSample" without spaces then click the [OK] button. Although our example only uses cells A5 through D5, creating our named range through column Z allows us to add additional columns later without having to remember to update the named range. If we update our formula and make it relative to the row of the OneSample named range, we can freely insert rows above and our formulas will continue to work properly. Change the formula in cell A5 to =INDEX(SampleID, ROW() - ROW(OneSample) + 1).

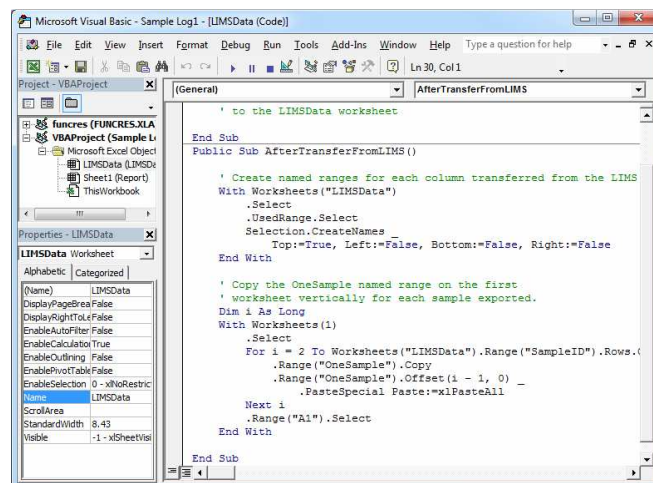
Now copy cell A5 to the other row five cells and change the SampleID field name in the formula to the column's appropriate LIMS field name (Batch, Customer.Name, Project.Name, etc.)

So that we do not have to anticipate the maximum number of samples ever exported and thereby copy our formulas to that many rows on the Report sheet, we can add a few lines of code to our AfterTransferFromLIMS macro to copy the formulas for us. Use Alt+F11 again to open the VBA editor and append the following lines below the existing lines in our macro:

```
' Copy the OneSample named range on the first
' worksheet vertically for each sample exported.
Dim i As Long
With Worksheets(1)
    .Select
    For i = 2 To
Worksheets("LIMSData").Range("SampleID").Rows.Count
    .Range("OneSample").Copy
    .Range("OneSample").Offset(i - 1, 0) _
    .PasteSpecial Paste:=xlPasteAll

    Next i
    .Range("A1").Select
End With
```

The screen below shows the updated AfterTransferFromLIMS macro. Use Debug | Compile VBA Project to check for errors. If there are no errors, close the VBA editor and return to the template.

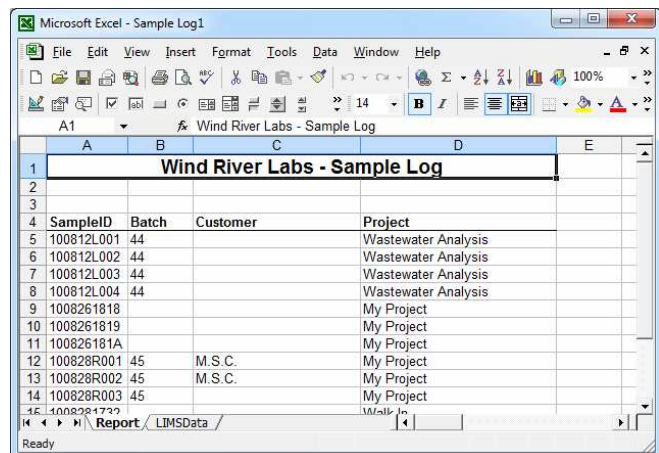


Close the template saving your changes and export the sample summary report again. Now the Report sheet is automatically visible after the macro completes and you should see one row of data for each sample exported.

(Continued on next page)


MSC-LIMS™ Insights

(Continued from previous page)



SampleID	Batch	Customer	Project
100812L001	44		Wastewater Analysis
100812L002	44		Wastewater Analysis
100812L003	44		Wastewater Analysis
100812L004	44		Wastewater Analysis
1008261818			My Project
1008261819			My Project
100826181A			My Project
100828R001	45	M.S.C.	My Project
100828R002	45	M.S.C.	My Project
100828R003	45		My Project

If any of the LIMS fields listed may be blank, such as batch and customer shown in the screen above, the result of our INDEX formula will display a zero. You can use Excel's option settings to suppress all zeros on the worksheet or you can use an IF function in your formulas to suppress zeros. For example, use the formula =IF(ISBLANK(expression), "", expression), where expression is the existing INDEX function.

The AfterTransferFromLIMS macro could also perform other tasks such as sorting the LIMSData sheet's data by customer. Now that you have a basic understanding of an Excel export template's purpose you may find it helpful to explore the macros in the MSC-LIMS example templates. Copying excerpts of code from a working template into your template is a good way to expand your Excel macro knowledge. 

Notes from Technical Support


Query by Sample Status

As one user noticed, sample status is not included in the query-by-example (QBE) controls, which is a situation we will correct in an upcoming version. In the meantime, there is a simple solution if you need to use a sample status criterion in your queries.

First, find the internal sample status numbers that uniquely identify each sample status. Preview the Sample Statuses report on the setup menu in the LIMS then export the report's data to the generic MSC-LIMS Export Template and note the sample status numbers in the resulting Excel workbook.

Next, add required "SampleStatus=N" expressions to the SQL Expressions screen, where N is the internal sample status number. For example, the default Normal, Rush, and Immediate statuses have internal sample status numbers 1, 2, and 3, respectively. Add the following expressions to the SQL Expressions screen on the Setup menu:

Name	Expression
Normal Samples	SampleStatus=1
Rush Samples	SampleStatus=2
Immediate Samples	SampleStatus=3

To query by sample status, simply select the appropriate expression from the SQL Expression list on the SQL tab of the QBE controls. 

Incomplete Samples with All Results


On more than one occasion, we have received a sample summary report as evidence of a sample with all analyses completed yet no completed date. For a site using MSC-LIMS Messaging to send completed results this posed a problem. How could the LIMS not mark the sample complete when all of its analyses were complete? Recently, more information brought the problem to light.

In one instance, a user was approving a batch of several hundred samples in the Results by Analyte screen in a large database with many concurrent users. When exiting the Results by Analyte screen, the LIMS had to process the completion event for every sample just completed, which included looking up and assigning costs to every analyte. In this case, with batch completion messaging enabled the LIMS also had to look up each sample's customer and project messaging configuration and add a completion message to Messaging's queue. The entire process takes time. Accustomed to fast system response when completing batches of just a few samples, the user thought the LIMS was hanging on the batch of several hundred samples. The user either rebooted or used Windows' Task Manager to end MSC-LIMS, which left some of the batch's samples without a completed date but with all results.

(Continued on next page)

MSC-LIMS™ *Insights*

(Continued from previous page)

The solution: be patient. Rebooting the workstation or using Task Manager to terminate an unresponsive MSC-LIMS should only be actions of a last resort. 

Sending Preliminary Reports with Messaging

A user sending final reports to customers automatically with MSC-LIMS Messaging recently asked:


I was wondering if there was any way that LIMS could send out a preliminary report. If a sample has more than one test being performed and one test is completed before another, is it possible for LIMS to send out a report with the one result completed and the other result still pending?

For Customers Only

This section of *MSC-LIMS Insights* is devoted to current users of MSC-LIMS. Here we briefly introduce only the most recent additions to MSC-LIMS.com Customers Only pages. Use your login name and password to log on to the Customers Only section of our website.


Knowledge Base

[Creating a New Excel Template Message Style in MSC-LIMS Messaging](#)

Summary: MSC-LIMS Messaging includes message styles that generate emails with attached files in a number of different formats including Excel and PDF files created from Excel. If you have created your own Excel final report template from one of the MSC-LIMS example templates, this article shows you how to add a new message style for your template so you can automatically send your final reports with MSC-LIMS Messaging. 

Yes, you can send a preliminary report via MSC-LIMS Messaging by using the resend button in the Results by Sample screen in the LIMS. The resend button is located immediately above the customer field. Prior to MSC-LIMS version 3.3, preliminary reports worked only for single-sample message styles. However, in version 3.3 the resend button now also works for batch message styles.

If you use an Excel template for your final report, consider using a formula in your template to label the report "Preliminary." For example, the formula below will insert "Preliminary" in the cell if at least one sample on the report is incomplete:

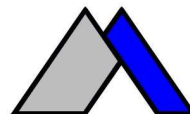
```
=IF(COUNTBLANK(CompletedDate) > 0, "Preliminary",  
" ") 
```

[Error: "Undefined function 'Replace\\$' in expression" when Importing from Excel](#)

Summary: A runtime error may occur in the results entry screens when importing results from Excel. Note that the error may be limited to a specific workstation in a multi-user installation and will only occur with MSC-LIMS version 3.3 or newer. This article describes how to troubleshoot and correct the error.

Contact Us

Questions, comments, suggestions?
Reach us at:



Mountain States Consulting, LLC
970 West Broadway #471
PO Box 30000
Jackson, Wyoming 83002 USA
Ph +1 307-733-1442
Fax +1 303-379-6850

info@msc-lims.com
www.msc-lims.com

Copyright © 2011 Mountain States Consulting, LLC.
All rights reserved.