

## XLR8 LabVIEW Toolkit Quick Reference Guide



DATA AHEAD **XLR8** is a toolkit to support direct read and write operations to Microsoft Excel® files in the “xls” and “xlsx” format from within LabVIEW. Use **XLR8** to read parameter data from xls and xlsx spreadsheets or to store reports that can be opened directly with Microsoft Excel® or the Microsoft Excel Viewer.

<b>XLR8 LabVIEW Toolkit Quick Reference Guide.....</b>	<b>1</b>
What XLR8 is.....	2
Supported functionality .....	2
Deployment .....	2
<i>Requirements.....</i>	<i>2</i>
<i>Building an application.....</i>	<i>3</i>
Warning .....	3
Installing XLR8.....	4
Quick introduction to the Microsoft Excel® spreadsheet format.....	5
Activating XLR8 .....	6
<b>Getting Started .....</b>	<b>9</b>
Write an “xls”/“xlsx” file .....	9
Read an existing file .....	9
<b>The XLR8 API.....</b>	<b>10</b>
Workbook operations (palette “Workbook“) .....	10
Sheet operations (palette “Sheet“).....	10
Read operations (palette “Read“) .....	11
Write operations (palette “Write“) .....	11
Edit Sheet operations (palette “Edit Sheet“).....	12
<i>Formatting .....</i>	<i>12</i>
<i>Utilities.....</i>	<i>13</i>
Applications / Examples („Write to xls file.vi“, Example finder).....	14
<b>Support and Feedback.....</b>	<b>15</b>
<b>VI Overview (LabVIEW) .....</b>	<b>16</b>

## What XLR8 is

**XLR8** is an API that offers a variety of read and write functions for Microsoft Excel® files in the “xls” and “xlsx” format, intended for the typical use in automated measurements and tests.

For this purpose, it provides a useful subset of a typical spreadsheet program’s functionality. While **XLR8** is based on a proven library “**NPOI 2.1**” and has been tested extensively, DATA AHEAD cannot guarantee full compatibility with all “xls” and “xlsx” files. As with any other piece of software, careful use is advised in order to prevent data loss. Make sure to have a backup of all relevant data files. Any corruption of data that results from the use of **XLR8**, DATA AHEAD cannot be held liable for.

## Supported functionality

- Microsoft Excel® 2007-2013 XLSX file format
- Microsoft Excel® 95-2003 XLS file format
- Reading, editing and writing of files
- Cell types: Strings, Numbers, Dates, Formulas
- Cell formatting operations:
  - o Font (type, style, color)
  - o Cell color
  - o Cell border style and color
  - o Data format
- Sheet formatting operations:
  - o Rows / columns
    - Width / height
    - Grouping
    - Merging
    - Hiding
  - o Inserting rows (retaining format)
  - o Deleting cells
  - o Deleting rows (retaining format)
  - o Splitting sheets
  - o Inserting images
  - o Zoom, gridline display
  - o Footer / header for printouts

## Deployment

### Requirements

If you want to deploy an application built with the **XLR8** components, please consider the following system requirements:

- Supported Operating Systems:
  - o Windows XP SP3
  - o Windows 7
  - o Windows 8
- Microsoft .NET Runtime 2.0 or newer
  - o Note that the Microsoft .NET 4.0 does not contain .NET 2.0
  - o On Windows XP please install .NET 3.5

### Building an application

**XLR8** adds all dependencies automatically. To deploy your application, ideally, build an installer from LabVIEW. If you choose to deploy manually please make sure the required files are present in the <data> folder of the application:

- ICSharpCode.SharpZipLib.dll
- NPOI.dll
- NPOI.OOXML.dll
- NPOI.OpenXML4Net.dll
- NPOI.OpenXMLFormats.dll

You will find these files in the following folder:

“C:\Program Files (x86)\National Instruments\LabVIEW xxxx\vi.lib\DATA AHEAD\XLR8\lib“

### Warning



Always create backup copies of any files you wish to edit with DATA AHEAD **XLR8**. Incorrect use of the **XLR8** API, as well as software or hardware crashes could lead to file corruption and loss of data!

## Installing XLR8

The easiest way to install **XLR8** is to find the Add-on in the LabVIEW Tools Network using the JKI VI Package Manager. Simply select “Install&Upgrade Packages” to install **XLR8** to your LabVIEW installation.

**XLR8** requires the JKI VI Package Manager 2014 or newer.

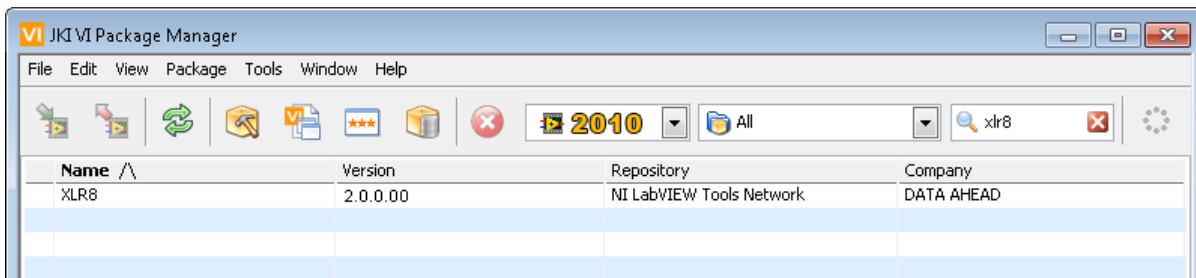


figure 1: JKI Package Manager to find and install XLR8

Upon successful installation, a new palette will be installed in LabVIEW that can be found in the “Programming” function palette of the block diagram.

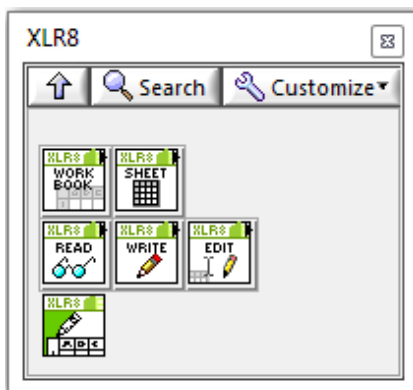


figure 2: XLR8 palette in "Programming"

Also, examples will be added to the LabVIEW Example Finder. Find them by selecting “Help” - > “Find Examples” (figure 3) in the LabVIEW menu. In the Example Finder, search for “XLR8”

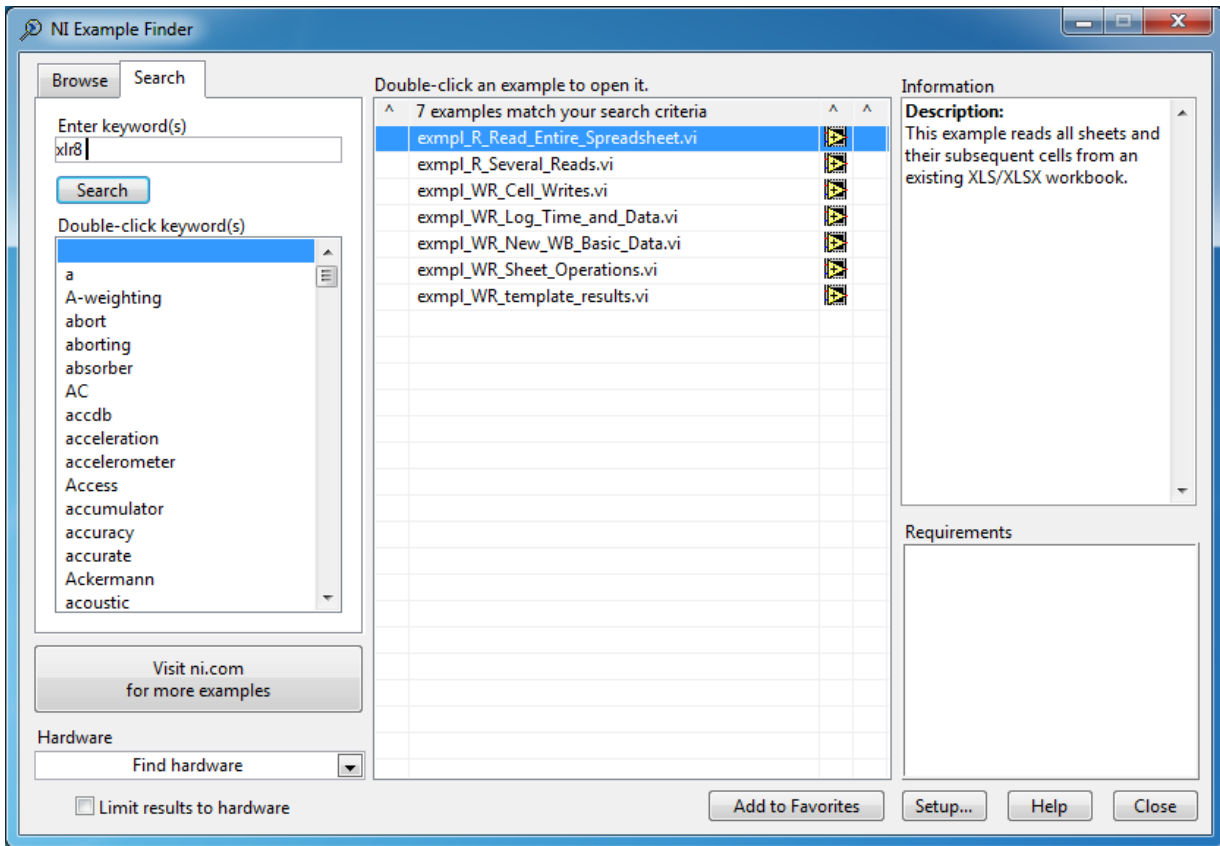


figure 3: XLR8 examples in the NI Example Finder

## Quick introduction to the Microsoft Excel® spreadsheet format

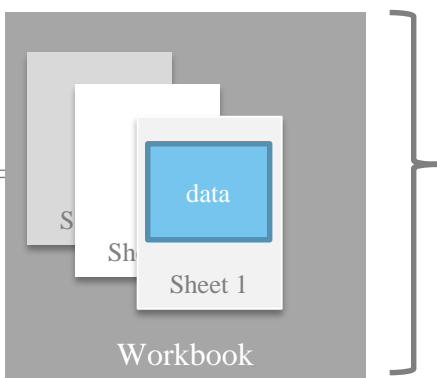


figure 4: xlsx workbook

A workbook represents the spreadsheet file and contains a number of sheets with data. In a spreadsheet program you can switch between sheets by selecting their corresponding tab.

## Activating XLR8

You can evaluate **XLR8** for 30 days. After this period you have to activate the **XLR8** add-on in LabVIEW. Upon purchasing a license from NI, you will receive an email titled “*Activation Information for your ni.com purchase: XLR8 - DATA AHEAD GMBH*” that contains your License ID and Password. In order to activate **XLR8**, your target system needs to be connected to the Internet. If there is no Internet connectivity available on the target computer, you can also activate **XLR8** through a web browser on a different computer (see step3 – alternative option).

**Step 1:** Open LabVIEW. Select “Help” in the menu bar and then on the menu point “Activate Add-ons” to carry out the function.

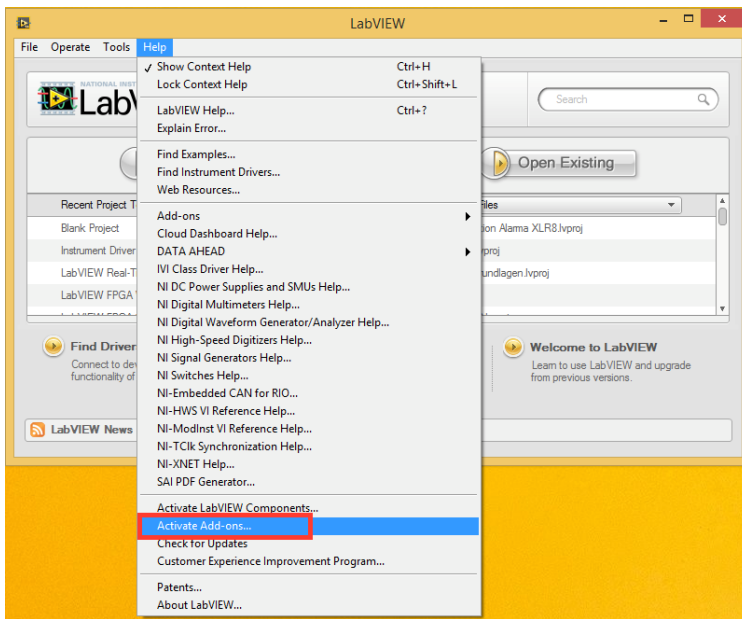


Figure 5: Activation Step 1

**Step 2:** The following pop-up window will appear and display your **XLR8** installation and activation status. Select **XLR8** “DATA AHEAD XLR8 x.x.x.xx” and click “next” or “Activate”.

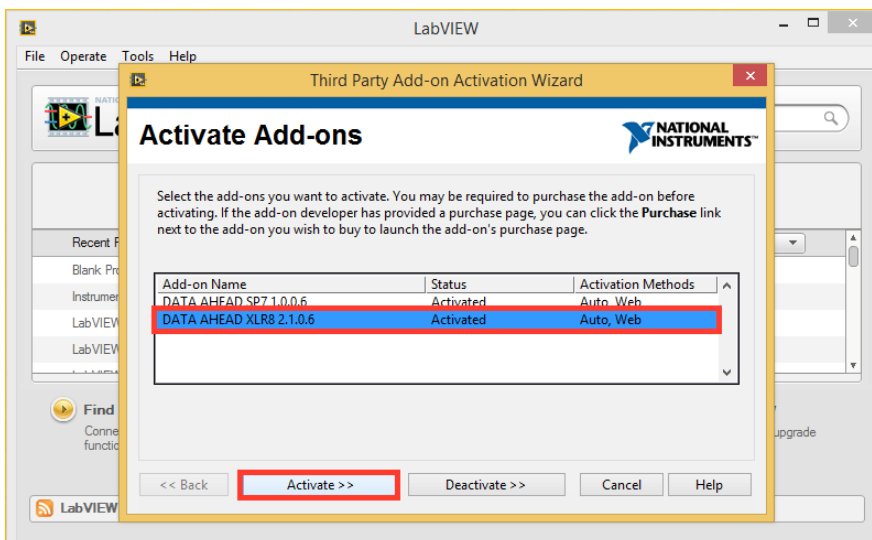
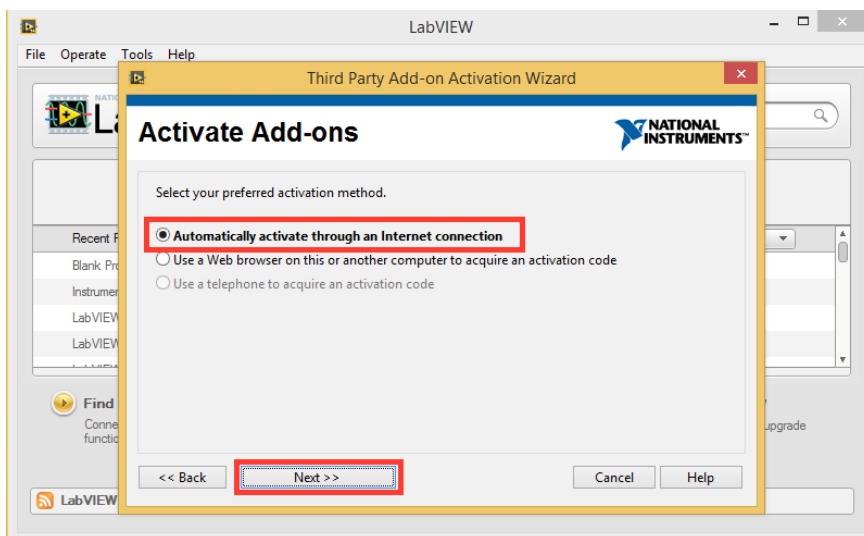


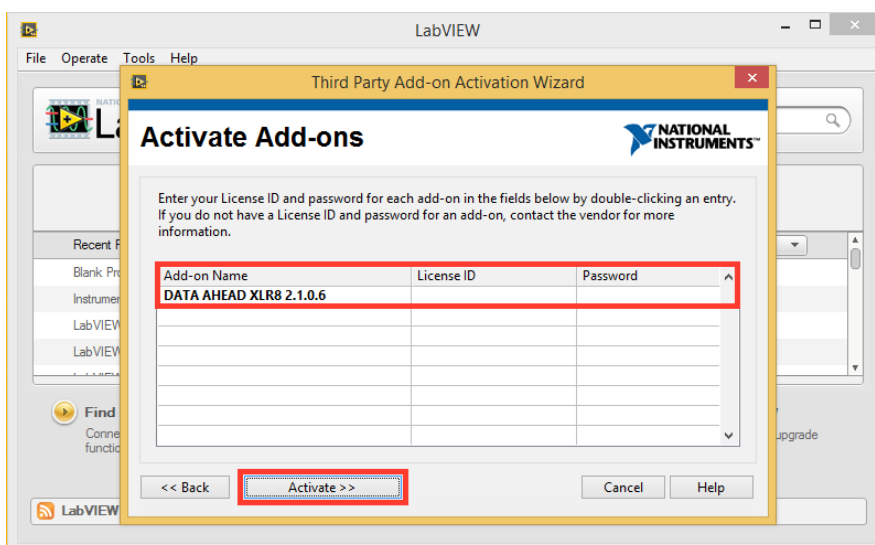
Figure 6: Activation Step 2

**Step 3:** Now select the method of activation. If the computer is connected to the Internet select the first option. If you do not have Internet access, see the alternative option below. To proceed, click “Next”.



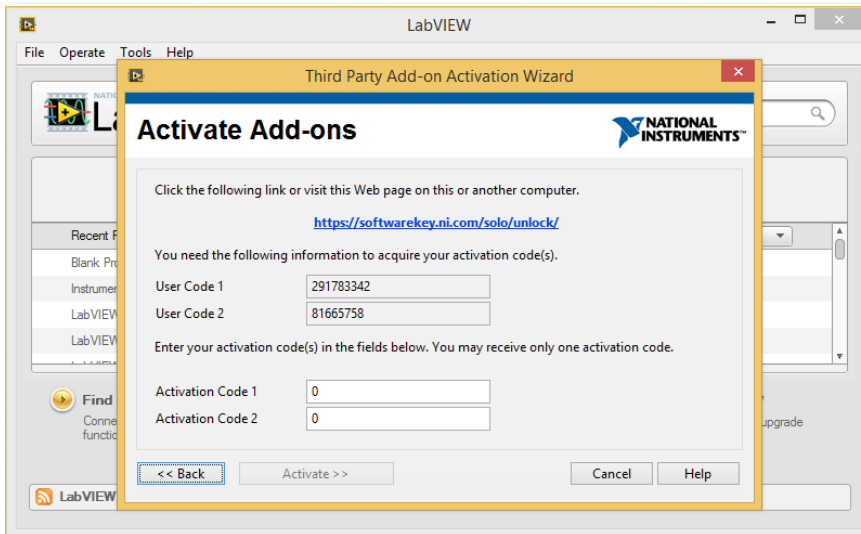
**Figure 7: Activation Step 3**

**Step 4:** To complete the activation, enter the provided License ID and Password that you received in the email from NI. Please mind capital letters. After entering the License ID and Password click “Activate”.



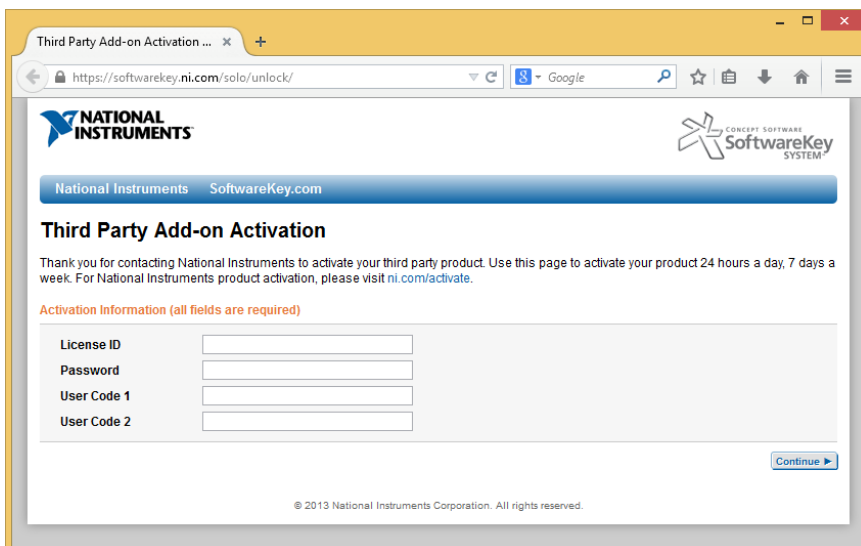
**Figure 8: Activation Step 4**

**Step 3 (alternative):** If no Internet connection is available, selecting the second option will direct you to National Instruments activation website where you will have to enter the user codes displayed in the dialog as well as your License ID and Password. This option allows you to activate the toolkit from another computer that is connected to the Internet.



**Figure 9: Activation Step 3, option 2: activate through a web browser – LabVIEW dialog**

**Step 4 (alternative):** Open the website <https://softwarekey.ni.com/solo/unlock/> on a computer with Internet connection. Enter the User Code 1 and User Code 2 on the website, as displayed in the dialog on the target computer. Enter License ID and Password from the activation e-Mail. Click “Continue”. Activation Code 1 and 2 will now be generated that you can enter in the dialog on the target computer. Finish the dialog by clicking “Activate”.



**Figure 10: Activation Step 3, option 2: activate through a web browser – activation website**



## Getting Started

To get started, please locate the examples that are shipped with the **XLR8** package. All of the examples include a short description in the VI documentation and explain the appropriate usage of the API VIs.

### Write an “xls”/“xlsx” file

In order to write a basic file in the “xls”/“xlsx” format, please open the example “exmpl\_WR\_New\_WB\_Basic\_Data.vi” (see front panel and code below). Select a new or existing file to save the data to, then run the example.

Locate the generated file and use Microsoft Excel® or a compatible spreadsheet program to check its contents.

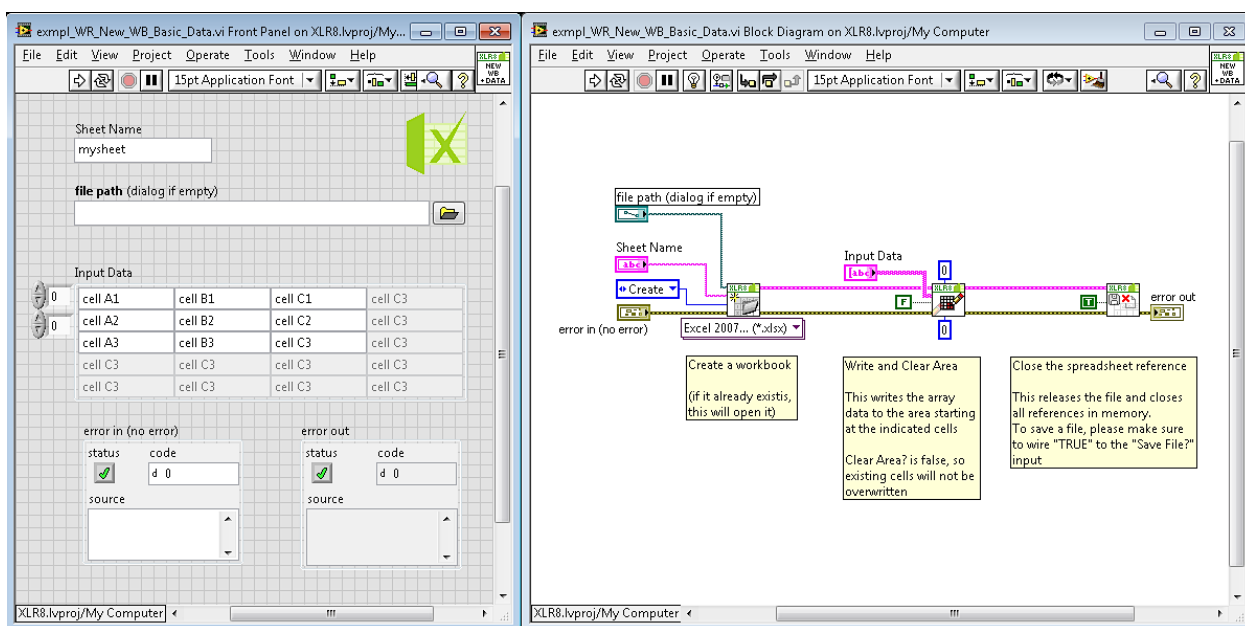


figure 11: basic write example

### Read an existing file

To show some basic operations that are required to read existing files, please open the example „exmpl\_R\_Several\_Reads.vi“. It will open a file that is shipped with **XLR8** and can be found in the folder <LabVIEW>\examples\XLR8\xls

Run it and examine its source code (figure 7)

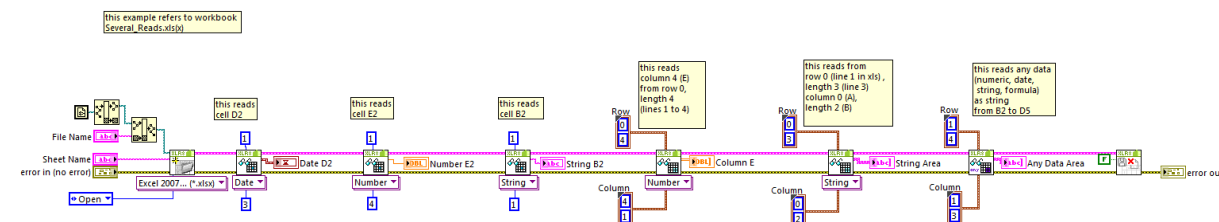


figure 6: basic read example

The other shipping examples demonstrate various basic and advanced functions of the **XLR8** API. Please refer to their respective VI documentations.

## The XLR8 API

### Workbook operations (palette “Workbook“)

These VIs refer to the workbook, i.e. the \*.xls/\*.xlsx spreadsheet file. The following operations can be performed:





VI	Name and description (see context help for more information)
	Open Create Workbook.vi Opens or creates a workbook
	Protect Workbook.vi Password protects a workbook
	Save Workbook.vi Saves a workbook
	Close Workbook.vi Closes a workbook. Can also save a workbook, if selected

table 1: workbook operations

### Sheet operations (palette “Sheet“)

These VIs refer to the active sheet, i.e. the tab that is currently shown in a spreadsheet program. Please see the example “exmpl\_WR\_Sheet\_Operations.vi“.









	Create Sheet.vi Creates a new sheet in the workbook
	Copy Sheet.vi Clones the active sheet.
	Edit Sheet Name.vi Renames the active sheet
	Delete Current Sheet.vi Deletes the currently active sheet
	Get Sheet Names.vi Lists all sheet names of the current workbook
	Hide Sheet.vi Hides the currently active sheet
	Set Current Sheet.vi Sets a sheet of the workbook active. All read/write operations refer to the active sheet
	Set Sheet Order.vi Places the active sheet to a new position in the workbook

table 2: sheet operations

## Read operations (palette “Read“)

These VIs are used to retrieve data from an existing sheet in a workbook. Depending on the data that you would like to read, select one of the following functions. See also examples that start with “exmpl\_R\_...” for Read function demonstrations.







	<b>Read Cell.vi (Polymorphic: Number / Date / String)</b> Reads a specified cell in the format selected
	<b>Read Area.vi (Polymorphic: Number / Date / String)</b> Reads a specified area in the format selected
	<b>Read Area (All Types).vi</b> Reads a specified area and returns string data
	<b>Read Current Sheet.vi</b> Reads a specified sheet and returns string data
	<b>Read All Sheets.vi</b> Reads all sheets of the workbook and returns a clustered string array
	<b>Read Named Range.vi (Polymorphic: Number / Date / String)</b> Reads a named range from a sheet

table 3: read operations

## Write operations (palette “Write“)

Use these VIs in order to write data to an existing sheet in a workbook. See also examples they start with “exmpl\_WR\_...” for write function demonstrations.









	<b>Write Cell.vi (Polymorphic: Number / Date / String)</b> Writes data in its respective format to a specified cell.
	<b>(Clear Sheet and) Write Area.vi (Polymorphic: Number / Date / String)</b> If selected, clears the sheet. Writes data to a specified area in the sheet.
	<b>Append Area.vi (Polymorphic: Number / Date / String)</b> Adds data in its respective format to the sheet, starting at the last row.
	<b>Write Hyperlink.vi</b> Writes a hyperlink to a specified cell.
	<b>Write Formula Cell.vi</b> Writes a formula to a specified cell.
	<b>Write Formula Area.vi</b> Writes formulas to a specified area of cells.
	<b>Append Formula Area.vi</b> Adds formulas to the sheet, starting at the last row.
	<b>Write To Named Range.vi (Polymorphic: Number / Date / String)</b> Writes to a named range in the selected sheet.

table 4: write operations

## Edit Sheet operations (palette “Edit Sheet”)

Use these VIs in order to format cells, cell ranges (“areas”) or certain sheet properties

### Formatting

	<b>Font Cell.vi</b> Adjusts the font settings of a specified cell.
	<b>Font Area.vi</b> Adjusts the font settings of a given area of cells.
	<b>Color Cell.vi</b> Adjusts the background color of a specified cell.
	<b>Color Area.vi (Polymorphic: single / multiple)</b> Adjusts the background color of an area of cells.
	<b>Borders Cell.vi</b> Adjusts the borders around a cell.
	<b>Borders Area.vi</b> Adjusts each cell’s border in a range of cells.
	<b>Frame Area.vi</b> Draws a border around a specified area.
	<b>Data Format.vi</b> Changes a cells data format, e.g. number representation, percent, etc.
	<b>Data Format Area.vi</b> Changes a cell range’s data format.
	<b>Row Height.vi</b> Changes the row height of a specified row.
	<b>Group Row.vi</b> Groups a number of rows.
	<b>Hide Row.vi</b> Hides a specified row.
	<b>Column Width.vi</b> Adjusts the width of a specified column.
	<b>Group Column.vi</b> Groups a number of columns.
	<b>Hide Column.vi</b> Hides a specified column.
	<b>Autosize Column.vi</b> Auto sizes a column.
	<b>Wrap Text.vi</b> Wraps the cell text to fit the column width.

## Utilities

	<b>Merge Cells.vi</b> Merges specified cells.
	<b>Delete Cell.vi</b> Removes a cell from a sheet. Can preserve formatting if desired.
	<b>Insert Row.vi</b> Inserts a number of new rows. Can preserve formatting if desired.
	<b>Delete Row.vi</b> Deletes a number of rows. Can shift up remaining rows if desired.
	<b>Get Last Row.vi</b> Gets last row of current sheet.
	<b>Zoom.vi</b> Zooms to a certain zoom level.
	<b>Display Gridline.vi</b> Toggles the display of grid lines.
	<b>Pane Freeze.vi</b> Freezes a window split to certain rows / columns.
	<b>Footer and Header.vi</b> Adds footer / header information for printing the document.
	<b>Wipe Area.vi</b> Wipes an area of cells, i.e. replaces all cells with blank cells.
	<b>Wipe Sheet.vi</b> Wipes an entire sheet, removes all existing data.
	<b>Create Named Range.vi</b> Creates a named range in the sheet.
	<b>Get Named Ranges.vi</b> Get all named ranges of current sheet as a string array.
	<b>Insert Image.vi</b> Inserts an image on current worksheet and cell. Supported formats: PNG, JPG (not supported for XLS), EMF, WMF, PICT, DIB (BMP).
	<b>Get Cellstyle.vi</b> Returns the Cell style of a selected cell.

## Applications / Examples („Write to xls file.vi“, Example finder)

XLR8 ships with a number of example programs that can be found using the LabVIEW example finder: Help -> Find Examples - Search for „XLR8“









	<p><b>Write to xls File.vi (Polymorphic: Double / Integer / String)</b>  Replaces the shipping “Write to spreadsheet file” VIs in the file palette(xls/xlsx).</p>
	<p><b>exmpl_R_Several_Reads.vi</b>  This VI demonstrates a number of read functions.</p>
	<p><b>exmpl_R_Read_Entire_Spreadsheet.vi</b>  This VI demonstrates how to read an entire xls spreadsheet.</p>
	<p><b>exmpl_WR_New_WB_Basic_Data.vi</b>  This VI demonstrates the typical API calls to create a simple new xls/xlsx file</p>
	<p><b>exmpl_WR_Sheet_Operations.vi</b>  This VI performs a variety of sheet operations</p>
	<p><b>exmpl_WR_Cell_Writes.vi</b>  This VI performs several cell writes using different data types and formats</p>
	<p><b>exmpl_WR_Log_Time_and_Data.vi</b>  This VI shows how to create a small logging document, creating a column for a time stamp and columns for measurement data</p>
	<p><b>exmpl_WR_template_results.vi</b>  This VI demonstrates how to fill out an existing Microsoft Excel® file, using it as a template and saving to a new file as a report</p>

table 5: applications and examples

## Support and Feedback

Please contact us at [xlr8@dataahead.de](mailto:xlr8@dataahead.de)

For latest news and support on our toolkits, go to  
<https://decibel.ni.com/content/groups/data-ahead-toolkit-support>

DATA AHEAD GmbH  
Prinzregentenufer 3  
90489 Nürnberg  
Germany

© Copyright 2015, DATA AHEAD

Microsoft Excel® is a registered trademark of Microsoft

XLR8 uses NPOI technology



<https://npoi.codeplex.com/>

## VI Overview (LabVIEW)

