

Using Inference in Excel

The screenshot displays the Microsoft Excel interface with the Inference Excel Task Pane open on the left. The task pane includes sections for 'Inputs', 'Edit Input Properties', and 'Parts Container'. The 'Inputs' section lists 'Expt (numeric-integer)', 'Run (numeric-integer)', and 'Speed (numeric-real)'. The 'Edit Input Properties' section shows 'Range Name: Speed' and 'Data Type: numeric-real'. The 'Parts Container' shows 'Inputs' and 'Outputs'.

The main Excel window shows two worksheets: 'Michelson Input Data' and 'Analysis Output'. The 'Michelson Input Data' worksheet contains the following data:

	Speed	Run	Expt
1			
2			
3	850	1	1
4	740	2	1
5	900	3	1
6	1070	4	1
7	930	5	1
8	850	6	1
9	950	7	1
10	980	8	1
11	980	9	1
12	880	10	1
13	1000	11	1
14	980	12	1
15	930	13	1
16	650	14	1
17	760	15	1
18	810	16	1
19	1000	17	1
20	1000	18	1
21	960	19	1
22	960	20	1
23	960	1	2
24	940	2	2
25	960	3	2

The 'Analysis Output' worksheet contains the following data:

	Analysis Output:
1	
2	
3	Number of Experiments:
4	Number of Runs:
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	

Paul van Eikeren, PhD

Blue Reference, Inc.

1 August 2008

Using Inference in Excel

Enable dynamic scripting in Microsoft Excel by using the capabilities of the Inference Parts-Container

Summary

Inference in Excel brings the benefits of dynamic scripting in platforms like R®, S-Plus®, MATLAB®, Python®, IronPython® and IronRuby® for .NET® to your Microsoft Excel® spreadsheets. Inference in Excel is an essential tool for business and technical professionals who regularly use Excel and need to expand its computational, graphic and automation capabilities. It provides a way for business analysts, financial analysts, line-of-business managers, scientists, engineers, educators and researchers to take advantage of Microsoft Excel to perform flexible, interactive exploration, analysis and visualization of multidimensional data by leveraging the power of these dynamic scripting platforms.

With Inference in Excel, you can have it all: the power and sophistication of the dynamic scripting platform while retaining the familiarity and convenience of Excel. Inference in Excel bridges Excel and dynamic scripting platforms, giving you the best of both worlds. By using data preparation in Excel in conjunction with high-level dynamic scripts, technical analysts can answer key questions, solve technical and business problems, and present results in ways that help drive decision making. The self-service nature of Inference in Excel means that users do not depend on IT support to get the job done. This document provides an introduction to using Inference in Excel.

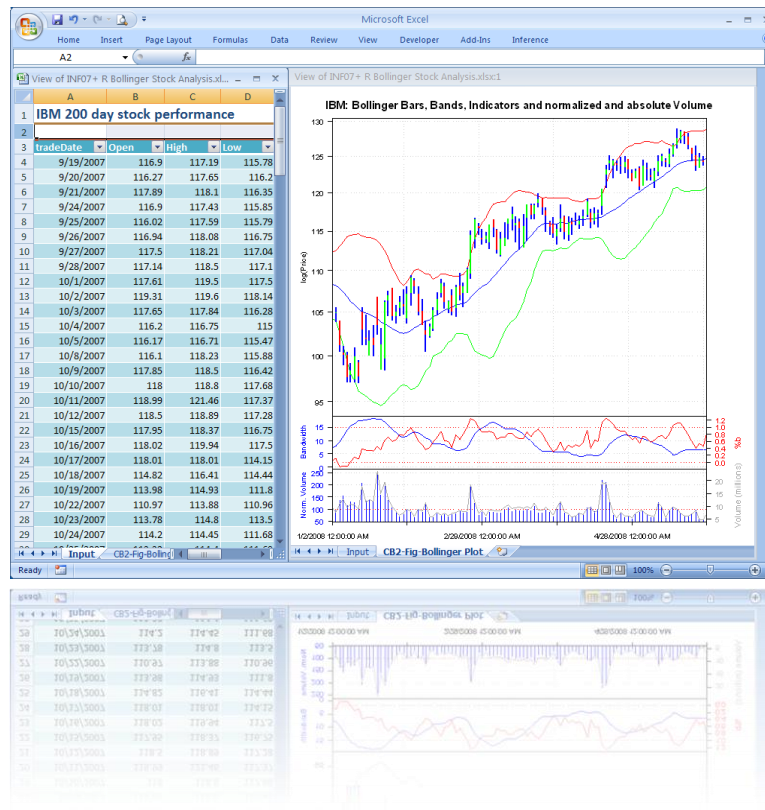


Table of Contents

Summary	2
Table of Contents	3
Excel Dynamic Documents	4
Dynamic Documents Based on Inference Parts-Container	4
Excel Named Ranges Linked to Parts-Container Elements	4
Assembling and Using an Excel Dynamic Document	5
Dynamic Document Embeds an Inference Parts-Container in Excel	6
Instantiate an Inference Parts-Container	6
Assemble and Manage Using the Inference in Excel Task Pane	6
Steps to Assemble and Manage Contents of Inference Parts Container	6
Step 1: Specify Inputs	6
Create Excel Input Sheet	6
Map Excel Range to Input Variable	6
Results of Mapping	7
Step 2: Specify Outputs	7
Create Excel Output Sheet	7
Map Excel Cell to Output Expressions	7
Results of Mapping	8
Step 3: Specify Scripting Instructions	8
Execute the Dynamic Document via Export Document	8
Overview of Export Document	8
Order of Execution in Dynamic Documents	9
Appendix A: Alternative Approaches to Mapping Excel Ranges	9
Disclaimer	10
Credit Notice	Error! Bookmark not defined.

Excel Dynamic Documents

Dynamic Documents Based on Inference Parts-Container

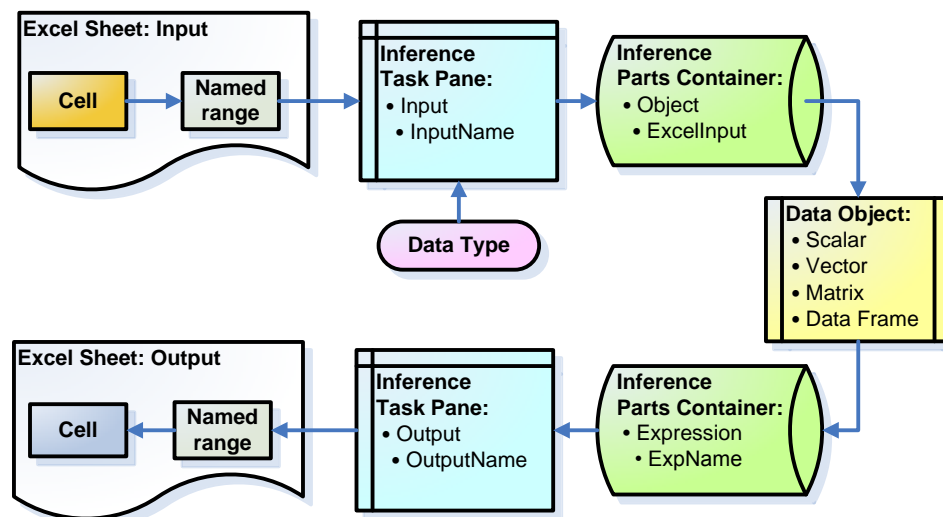
Inference in Excel extends Microsoft Excel by providing a bridge to powerful dynamic scripting solutions, which enables you to access the capabilities of R, S-Plus, MATLAB, Python and dynamic scripting for .NET from an Excel spreadsheet. Inference in Excel uses dynamic documents, which combine the features and capabilities of conventional Excel documents with interactive scripting to enable dynamic creation of just-in-time results documents. With Inference in Excel you can exchange data between dynamic scripting solutions and Excel, taking advantage of the familiar Excel interface while accessing the powerful computational and visualization capabilities of the R, S-Plus, MATLAB and .NET environments. Now you can have the best of both worlds. Inference in Excel brings the powerful capabilities of the high-end technical scripting platforms to the ease-of-use of the Excel spreadsheet.

Inference in Excel is a member of the Inference solution platform, a tightly integrated suite of applications based on Microsoft Office, which operate on a common Inference Parts-Container linked to an underlying dynamic scripting platform and the Inference Export framework. An Inference Parts-Container provides a structured entity for holding and managing data sets, software objects, code blocks and expressions. An Inference Parts-Container can be serialized as a file, as it is for Inference Studio, or directly embedded in Microsoft Office documents, as it is for Word and Excel.

Excel Named Ranges Linked to Inference Parts-Container Elements

Input and output values in Excel cells are linked to the Inference Parts-Container, which in turn is linked to the underlying dynamic scripting platform, via the mapping scheme illustrated in the figure below.

Notable elements of this mapping scheme include the following:



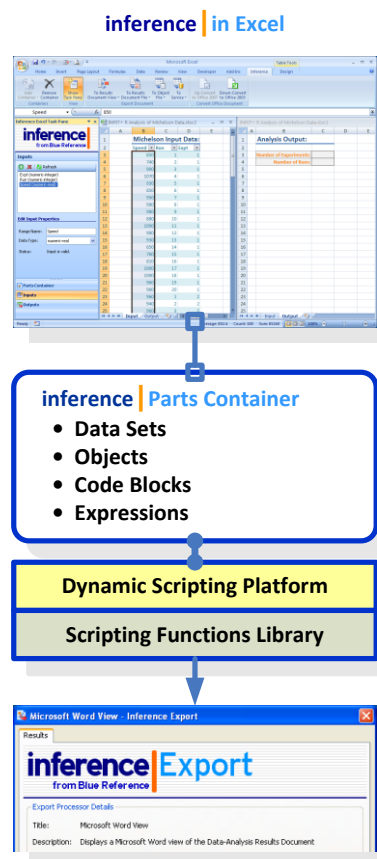
- Input cell values are assigned to Excel named ranges on the sheet labeled “Input.”
- Excel named ranges are mapped to variables names and displayed on the Inference Task Pane. Optionally, variable names can be strictly typed (character, integer, real, logical, categorical or date-time), where data typing is ultimately passed to the data object.

- The collection of input data and associated attributes resides in the inference Parts-Container as an object under the name “ExcellInput.” The object is dynamically created when the workbook is saved.
- Data structures supported in Inference in Excel include scalars, vectors, matrices and data frames.
- Output values are generated by executing an Expression, which is comprised of single lines of code that return scalar, vector, matrix or data frame values.
- Output values are mapped to Excel named ranges corresponding to assigned cells on the sheet labeled “Output.”

Assembling and Using an Excel Dynamic Document

Assembling an Inference in Excel dynamic document consist of the following sequence of three steps, which is illustrated in the figure below:

1. Embed an Inference Parts-Container in the Excel document
2. Assemble the dynamic document by specifying the contents of the Inference Parts-Container and perform the necessary Input and Output mapping to Excel via Excel named ranges
3. Execute the dynamic document via Document Export to generate the results document.



The sequence of steps to assemble and use an Excel dynamic document is described in detail below.

Dynamic Document Embeds an Inference Parts-Container in Excel

Instantiate an Inference Parts-Container

When Inference is selected in the Excel ribbon, you are offered a choice of two types of containers to insert: a Parts Container or a DataFrame-Container. The Parts-Container is used to assemble dynamic documents; the DataFrame-Container is used to assemble a dataset that has the properties and attributes of a data frame.

Assemble and Manage Using the Inference in Excel Task Pane

Upon inserting a Parts-Container, you will see a new Task Pane comprised of three sections:

- The **Parts-Container Section** is used to assemble and manage the elements of the dynamic document including data sets, objects, code blocks and expressions.
- The **Input Section** is used to assemble and manage the mapping between cells in the Excel sheet labeled “Input” and the object entitled “ExcelInput”, which is dynamically created in the Parts-Container.
- The **Output Section** is used to assemble and manage the mapping between placeholder cells in the Excel sheet labeled “Output” and the Expressions specified in the Parts-Container.

Steps to Assemble and Manage Contents of Inference Parts-Container

Step 1: Specify Inputs

Create Excel Input Sheet

First create an Excel sheet and label it “Input” (without quotes). The Input sheet is where an Inference in Excel dynamic document stores the collection of input values, which get mapped to the scripting platform. We will use Excel Named Ranges as identifiers for the location on input values.

Map Excel Range to Input Variable

Next map an Excel range of cells to an input variable in the following manner¹:

- Add data to cells in the Excel sheet labeled “Input.”
- Select a rectangular range of cells² containing the data.
- Using the Input Section of the Task Pane, select the “Map New Input to Selected Cell” icon.
- Enter a valid Excel Range Name comprised of letters, number and underscores.

¹ It is possible to map Excel ranges to input variables by alternate means. In fact, you can use any of the features provide by Excel in the Formula section of the ribbon to Define Names on the Input sheet. This is explained further in Appendix A.

² To map a scalar value, select a single cell. To map a vector, select a column of cells or a row of cells. To map a matrix, select a rectangular grouping of cells. Note that the selection must always be rectangular and contiguous.

- The result of the above actions is twofold. One, the variable name now shows up on the Task Pane and is assigned a character data type by default. If you make any changes to the variable you may need to refresh the Task Pane. And two, selecting the variable name on the Task Pane triggers selection of the corresponding range on the Input sheet.
- Use the Edit Input Properties on the Task Pane to change the name of the variable or to change the assigned data type.

Results of Mapping

The action of saving the Excel workbook dynamically creates an object entitled “ExcelInput.” If you move to the Parts Container section of the Task Pane, you will see that a new object entitled “ExcelInput.” This object is created dynamically every time the Excel workbook is saved to a disk file. You can examine the contents of the object by right clicking on the object and selecting the Edit in File Editor. Do not make changes in the file as the synchronized mapping between and Excel and the scripting platform will be altered.

Step 2: Specify Outputs

Create Excel Output Sheet

Create an Excel sheet and label it “Output” (without quotes). The Output sheet is where values resulting from Expression calculations will be collected upon the action of Export Document. In an analogous manner, we will use Excel Named Ranges as placeholders for the Expression output values.

Map Excel Cell to Output Expressions

Next map an Excel cell to an output Expression³ in the following manner:

- Select the cell in Output sheet where you want the results of the Expression calculation to be placed. The chosen cell will correspond to the first value in a column vector, the first value in a row vector, or the upper-left corner of a matrix.
- Using the Output Section of the Task Pane, select the “Add Expression Output to Selected Cell” icon.
- Enter a valid Expression Label comprised of letters, number and underscores.
- The result of the above actions is twofold. One, the Expression label name now shows up on the Task Pane. If you make any changes to the Expression you may need to refresh the Task Pane. And, two, selecting the Expression name on the Task Pane triggers selection of the corresponding cell on the Output sheet.
- Use the Edit Expression Properties on the Task Pane to change the name of the Expression or to change whether you want to use the alias code text.

³ In the procedure outlined, an Excel cell is mapped to a new Expression name. An alternative method involves first creating the Expressions in the Parts-Container section. Then for each Expression shown in the Output section, map the Expression to a selected cell in the Output sheet using the “Map Expression Output to Selected Cell” icon.

Results of Mapping

The action of saving the Excel workbook dynamically creates a collection of Expressions. If you move to the Parts Container section of the Task Pane, you will see new Expressions corresponding to the mapping in the Output section. You can now specify the instructions for the mapped Expressions by right clicking on the Expression and selecting Edit in Expression Editor. Upon the action of Export Document, the Expression instructions are executed and the output is added to the Output sheet and can be used in subsequent Excel calculations.

Step 3: Specify Scripting Instructions

The instruction specifying how the dynamic scripting platform is to use the Excel data are specified in the Inference Parts-Container as code blocks. Scripting instructions are specified by creating one or more code blocks in the Inference Parts-Container section and adding the requisite scripting instructions using the CodeBlock Editor⁴. There are four principal uses of scripting instructions in code blocks:

1. To instantiate the collection of objects needed to perform the requisite calculations and analyses.
2. To define custom functions which are executed as Expressions.
3. To specify the textual output to be created. This will be placed in a separate sheet during Export Document.
4. To specify the graphical output to be created. This will be placed in a separate sheet while performing the Export Document function.

Execute the Dynamic Document via Export Document

Overview of Export Document

Executing a dynamic Excel document is comprised of performing the Export Document function of Inference in Excel. This function provides results documents for four possible output destinations:

- **To a Results Document View.** Essentially provides a non-persistent preview of the results document. Users can select from an Excel view, a webpage view, or a PDF view.
- **To a Result Document File.** Provides the results document in a persistent form as a file. Users can select from an Excel 2003 or 2007 document file, a PDF document file, an HTML document file, a web page archive file, or an XPS document file.
- **To an Object File.** Provides a means to capture software objects instantiated by the scripting platform as persistent files. Users can select from an Excel file with an embedded DataFrame container or a StatDataML file.

⁴ An alternative to using the CodeBlock Editor for assembling the scripting instructions is to use the Inference Studio integrated development environment (IDE) that is supplied with Inference in Excel. To use Inference Studio simply right click on the Parts Container node in the tree and select Edit with Inference Studio.

- **To a Service.** Provides a means to assemble the elements in the Parts Container as individual files collected in a folder, which can be used directly by the native dynamic scripting application.

Export Document is an independent process that operates in its own thread outside of Inference in Excel. Export Document is a highly flexible process that can be extended to include additional and custom destinations by using the Inference SDK tool.

Order of Execution in Dynamic Documents

When the user calls the Export Document function, execution occurs in the following sequence:

- the scripting runtime engine is instantiated
- the called out References (packages, libraries) are loaded in the order specified
- the Data Sets are loaded in the order they appear in the Parts-Container
- the Objects are loaded in the order they appear in the Parts-Container
- the Code Blocks are executed in the order they appear in the Parts-Container
- the Expressions are executed in the order they appear in the Parts-Container

Appendix A: Alternative Approaches to Mapping Excel Ranges

Mapping of Excel ranges to input variables can be achieved by alternate means. You can use any of the features provide by Excel for defining and using names in formulas by using the Formula > Define Names section of the ribbon. For detailed help go to the Excel Home > Formula and Name Basics > Working with Names section of the Excel Help. For example, you can use the Excel ribbon to:

- Define a name for a cell or cell range on a worksheet
- Define a name by using a selection of cells in the worksheet. This is especially useful if you want to map all the columns in a DataFrame-Container in a single action.
- Define a name by using the New Name dialog box
- Manage names by using the Name Manager dialog box
- Change a name
- Delete one or more names

This document was developed prior to the product's release to manufacturing, and as such, we cannot guarantee that all capabilities and features described herein will conform exactly to those found in the shipped product. The information contained in this document represents the current view of Blue Reference, Inc. as of the date of publication. Because Blue Reference must respond to changing market conditions, this document should not be interpreted to be a commitment on the part of Blue Reference, and Blue Reference cannot guarantee the accuracy of any information presented after the date of publication. The information represents the product at the time this document was printed and should be used for planning purposes only. Information is subject to change at any time without prior notice. This document is for informational purposes only. BLUE REFERENCE MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

Blue Reference, Inc. may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Blue Reference, Inc., the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property. Contact us at info@bluereference.com for more information.

Blue Reference, Inference, inference |, Parts Container, DataFrame Container and InterViewer are either registered trademarks or trademarks of Blue Reference, Inc. in the United States and/or other countries. Excel, Word, Office, .NET, JScript, IronPython, IronRuby, and Visual Basic are registered trademarks or trademarks of Microsoft Corporation in the United States and other countries. Matlab is a registered trademark of The MathWorks, Inc.™ Python and CPython are registered trademarks or trademarks of the Python Software Foundation. S-PLUS is a registered trademark of Insightful Corporation in the United States and/or other countries. R is a copyright of the Free Software Foundation, Inc. Other product and company names mentioned herein may be trademarks or registered trademarks or copyrights of their respective companies and are the property of their respective owners.

© 2007-2008 Blue Reference, Inc. All rights reserved