

inference | Parts-Container Explained

The screenshot displays the inference software interface, which is designed for data analysis and reporting. It consists of several main components:

- Excel Spreadsheet:** A window titled "IN07 | R Analysis of Michelson Data.xlsx" showing a table with columns "Speed", "Run", and "Expt". The data is as follows:

Expt	Speed	Run
1	850	1
2	740	2
3	900	3
4	1070	4
5	930	5
6	850	6
7	950	7
8	980	8
9	880	9
10	880	10
11	1000	11
12	980	12
13	980	13
14	650	14
15	760	15
16	810	16
17	1000	17
18	1000	18
19	960	19
20	960	20
21	960	1
22	940	2
23	960	3
24	940	4
25	880	5
26	800	6
27	850	7
28	880	8
29	900	9
30	880	8
31	900	9
- Word Document:** A window titled "IN07 | R Analysis of Michelson Data.docx" showing a document titled "Analysis of Michelson Data". The document contains sections for "Background", "The Data Set", and "The Analysis".
- Task Panes:** Two "inference" task panes are visible. The left pane shows a "Parts Container" with a tree view of data sets and code blocks. The right pane shows "Edit Parts Container Properties" with a "Platform" dropdown set to "R" and a "References" list containing "xtable".

Paul van Eikeren, Ph.D.

www.inference.us

25 February 2009

inference | Parts-Container Explained

Encapsulates and manages structured elements of dynamic documents in Microsoft Office documents

Summary

Inference merges three core business processes—document publishing and reporting, software application development, and predictive analytics—into a single unified space. Inference works in conjunction with Microsoft Office by transforming standard Office documents into dynamic documents, turning Office document into software applications. Inference extends the capabilities of Office far beyond the functions of word processing, spreadsheet calculations and slide presentation.

Inference enables dynamic documents by extending Office document with a generic embedded Parts Container. The embedded Parts Container provides support for both free-form unstructured content and, structured information comprised of data and protocols for performing actions on the data. By embedding the inference | Parts-Container in Microsoft Office documents and linking the protocols to a predictive analytics engine, documents become applications.

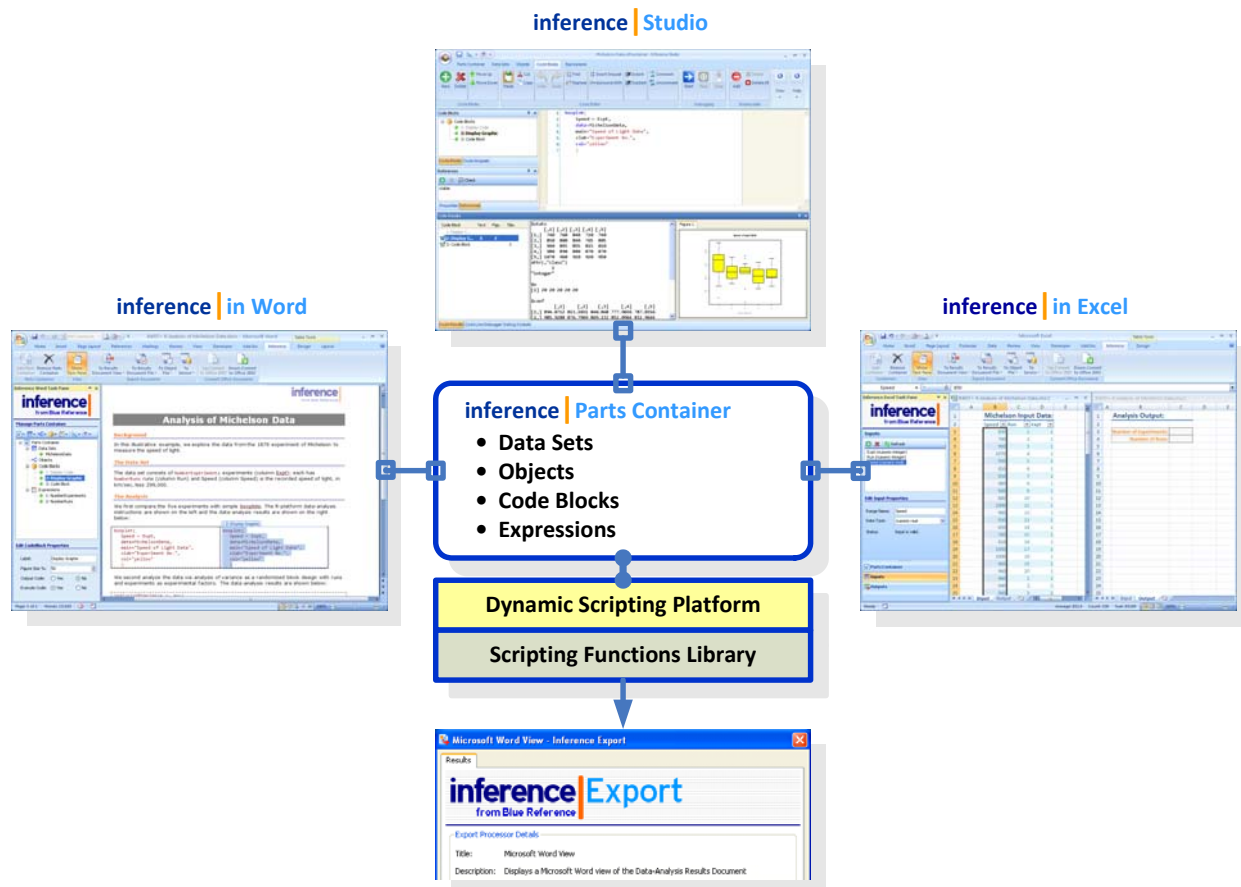


Table of Contents

Summary	2
Business Case for Dynamic Documents	4
Competing on Analytics Driven by Predictive Analytics Software	4
Documents Remain Central to Business Processes	4
BPM Drives Convergence of Documents and Predictive Analytics.....	4
The Document-as-Application: Dynamic Documents	5
Dynamic Documents Enabled by inference Parts-Container	5
Separate Structured from Unstructured Information	5
Store and Manage Structured Information in a inference Parts-Container	5
Working with the inference Parts-Container	6
Inserting the inference Parts-Container in Office Documents.....	6
Managing the inference Parts-Container Using the Inference Task Pane	7
Build Results Document with Export Document Services	8
Benefits of Inference Dynamic Documents	9
Bridge the World of Structured and Unstructured Information.....	9
Enable Document Indexing and Retrieval Based on Free-Form Text and Structured Fields	10
Increase Your Return from Investment in Dynamic Scripting Platforms	11
Speed-Up Predictive-Analytics Application Development.....	11
Dynamic Documents Enable Reusable Predictive Analytics	12
Dynamic Documents Implement Literate Predictive Analytics	13
For More Information	13

Business Case for Dynamic Documents

Competing on Analytics Driven by Predictive Analytics Software

Business luminaries have argued that the frontier for using data to make competitive decisions has shifted dramatically. There has been an emergence of a new form of competition based on extensive use of data, analysis, predictive modeling, and fact-based decision making. Companies do not just want to report on historic data, but want to know what the trends are, what will happen if the current trend continues and, based on that information, take appropriate actions.

Competing on Analytics uses a class of software application called predictive analytics, which employ a range of techniques from statistics and data mining that analyze current and historical data to make predictions about future events. In science and engineering, predictive analytics exploit laboratory data to capture relationships among many factors to allow assessment of performance or risk of failure to achieve performance, which then serves to guide appropriate operating conditions leading to success. In business, predictive analytics exploit patterns found in historical and transactional data to identify risks and opportunities. Application of predictive analytics ranges from improving product quality by identifying sources of variability and failure to using new tools to identify the most profitable customers and offer them the right price for a product. Some recognized leaders are using predictive analytics for performance improvement by using experimental designs to measure the overall impact or “lift” of intervention strategies like targeted price and product promotions.

Documents Remain Central to Business Processes

In spite of these new developments, simple documents remain central to business processes. Documents are portable, persistent and self-contained artifacts. They represent fully contextual views into information organized for the purpose of relating the disparate pieces into actionable knowledge. Many business processes are document centric and require the persistence and context of a document. Documents are used to transfer knowledge and information workers rely on documents to communicate information within organizations.

BPM Drives Convergence of Documents and Predictive Analytics

Business process management (BPM) is a method to align an organization with the wants and needs of customers. It is a holistic management approach that promotes business effectiveness and efficiency. BPM occurs best when the content, structure and knowledge in documents are brought together with data and predictive analytics in a relevant and consumable fashion. Business data is typically focused on the “what” of a business. Business documents tend to focus on the “why” and are unstructured and contextual. And, predictive analytic applications contain the “why” of the business. The reality is that business is done at the intersection of “what”, “why” and “how”---where facts and context meet actionable analysis.

Traditionally, the domains of documents, data and predictive analytic applications have been isolated from one another. Documents are stored in content management systems, shared file servers and local drives. Data is stored in data bases. And predictive analytic applications are high-end applications relegated to specialists. Dynamic documents provide a means to combine the traditional strength of

documents with the power and flexibility of predictive analytics applied to data. In fact, dynamic documents can act like software applications while preserving the features of established workflows and formats. Dynamic documents combine the best attributes of both applications and static documents.

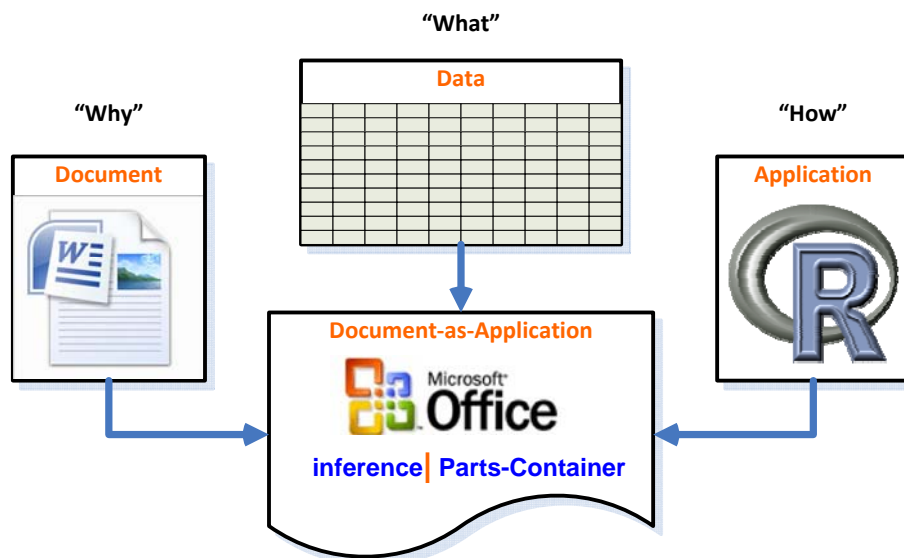
The Document-as-Application: Dynamic Documents

The lines are beginning to blur between publishing documents and performing predictive analysis. Many business processes require the persistence and context of a document and the dynamism and interactivity of a predictive analytics. They require structured data to come together with unstructured content—not only on the computer screen—but as part of a unified document. For many document-centric processes, the document becomes the new predictive analysis context. Organizations must no longer make the tradeoff between live data, the interactivity of predictive analytics, and the persistence and context of a document. The document is the application.

Dynamic Documents Enabled by inference| Parts-Container

Separate Structured from Unstructured Information

A persistent theme in the application of dynamic documents is required support for both free-form unstructured content (“why” elements) and, structured information comprised of data (“what” elements) and protocols for performing an action on the data (“how” elements). Inference resolves this tension by providing a universal container, the inference| Parts-Container, for storing and managing structured information. By embedding the inference| Parts-Container in the Microsoft Office document and linking the protocols to a predictive analytics engine, documents become applications.



Store and Manage Structured Information in a inference| Parts-Container

The unifying element of Inference is to transform Microsoft Office documents into applications—that is, turn them into “smart documents”—by embedding an inference| Parts-Container and linking it to

calculation engine of an application. Specifically, the embedded inference | Parts-Container is used to store and manage the following:

Data Sets. These are data objects captured in standardized formats that support data structures including scalars, vectors, matrices, data frames and lists. Illustrative examples of such data objects are Excel (*.xls and *.xlsx) files and Statistical Data Markup Language (StatDataML) files. Using standardized formats ensures interoperability between applications. For example, StatDataML provides a seamless method to exchange data between R, S-Plus, Matlab, Octave and python dynamic scripting platforms as well as between the Java and .NET (e.g., C#, VB.NET, IronPython, IronRuby, managed Jscript, dynamic VB) frameworks. Making Data Sets an extensible element future-proofs the inference | Parts-Container. For example, a developer can extend the inference | Parts-Container to store and manage additional (new, open or proprietary) file formats using the capabilities of the inference | SDK.

Objects. Any software object serialized as a file can be stored in the inference | Parts-Container. Object types are pluggable entities and therefore extensible using the inference | SDK. Particularly useful objects are executable instructions, either as ASCII text or binary data, for the dynamic scripting platform. Objects marked as loadable are instantiated by the dynamic scripting platform when the contents of the Part-Container are executed. For example, the following objects are currently supported: *.R scripts and *.RData files for R; *.py and *.pickle files for python; and *.js files for JScript.

Code Blocks. These are blocks of code written in the language of the dynamic scripting platform. A code block is comprised of one or more lines of code. Code blocks are interpreted and executed by the dynamic scripting platform during the Export Document function. Code blocks may be used to define new functions and/or instantiate software objects. Some code blocks produce textual and/or graphical output, which then becomes integrated into the results document.

Expressions. These are single lines of code written in the language of the dynamic scripting platform. Expressions are typically used to return simple character or numeric values. Expression results may be inserted into the running text of a paragraph like in inference | in Word results documents or may be inserted in cells in inference | in Excel results documents. Expressions are distinguished from Code Blocks in that Expressions are executed after all Code Blocks have been executed. This allows one to request the output of Expressions anywhere in the document, even if the Expression depends on prior execution of a Code Block.

Working with the inference | Parts-Container

Inserting the inference | Parts-Container in Office Documents

Assembling or managing an Office dynamic document entails two tasks. Task one consists of laying out the free-text or spreadsheet elements of the document. And, task two consists of assembling the necessary elements of the inference | Parts-Container and providing the corresponding placeholders for locating them in the dynamic document.

Assembling or managing an Office dynamic document requires starting with an Office document containing an embedded inference | Parts-Container, which can be obtained one of two ways:

1. **Making a clone of an existing Office dynamic document.** You can clone (make a copy) any Office dynamic document and use the embedded Parts-Container in the clone as a starting point for assembling a new dynamic document.
2. **Insert a Part-Container into an existing Office document.** You can take an existing Office document (e.g., new Word file or exiting Excel file) and insert a Part-Container by selecting Inference > Parts Container > Add Parts Container from the ribbon.

Managing the inference| Parts-Container Using the Inference Task Pane

An inference | Parts-Container embedded into an Office document has an associated Task Pane, which is used to manage its contents and properties. The Task Pane can be made visible/invisible by using the Inference > View > Show Task Pane toggle on the Office ribbon. The Inference Task Pane has two principal regions, which are illustrated in the figure below for inference | in Word:

Manage Parts Container Region. The hierarchical tree is a representation of the Parts Container, which is comprised of 4 part types (Data Sets, Objects, Code Blocks and Expressions). Within each part type are zero or more instances of the part type.

Edit Parts Properties Region. Selecting the Parts Container or any of the parts types in the tree provides an Edit Parts Container form in the Edit Parts Properties region. This is where you specify the dynamic scripting platform and accompanying references to packages/libraries you want to associate with the dynamic document. Selecting any instance of a part type in the tree provides an Edit “part-type” Properties form in the Edit Parts Properties region. This is where you set the properties for the particular instance of a parts type.

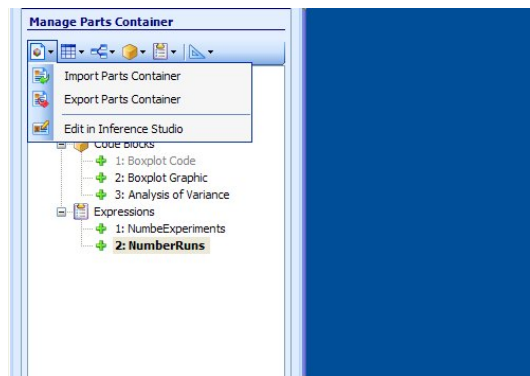
The screenshot shows the Microsoft Word interface with the Inference Task Pane open on the left and the Office Document Canvas on the right. The task pane is divided into two main sections: 'Manage Parts Container' and 'Edit CodeBlock Properties'. The 'Manage Parts Container' section shows a hierarchical tree of parts, including 'Parts Container', 'Data Sets', 'Objects', 'Code Blocks', and 'Expressions'. The 'Edit CodeBlock Properties' section shows a form for editing the selected part. The Office Document Canvas displays the content of the selected part, which is a document titled 'Analysis of Michelson Data'. The document contains text, a table, and code blocks. Red arrows point to the following regions:

- Inference Task Pane:** The top section of the task pane, containing the 'Manage Parts Container' and 'Edit CodeBlock Properties' sections.
- Manage Parts-Container:** The hierarchical tree of parts in the task pane.
- Edit Parts Properties:** The form for editing the selected part in the task pane.
- Office Document Canvas:** The main area of the Word document displaying the content of the selected part.
- Expressions:** A section of the document content, highlighted with a red box.
- Code Blocks:** A section of the document content, highlighted with a red box.

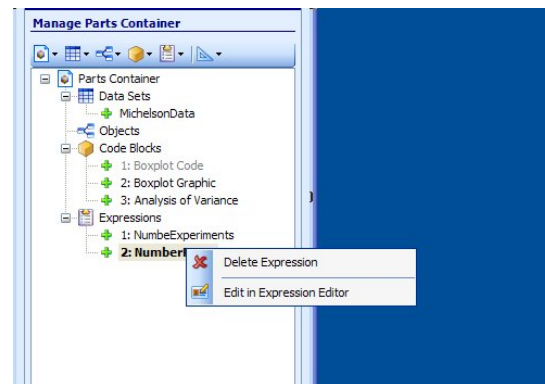
Many of the operations of the inference | Parts-Container can be initiated in the hierarchical tree in the Manage Parts-Container region. Principal initiation points include the following:

Manage Document Menu. This provides drop-down menus for initiating operations on the parts container, data sets, objects, code blocks and expressions.

Right-Click Element Menus. Right-clicking on the container, part types and part-type instances in the hierarchical tree provides menus for initiating appropriate operations.



Manage document menu



Right-click menu on tree

Build Results Document with Export Document Services

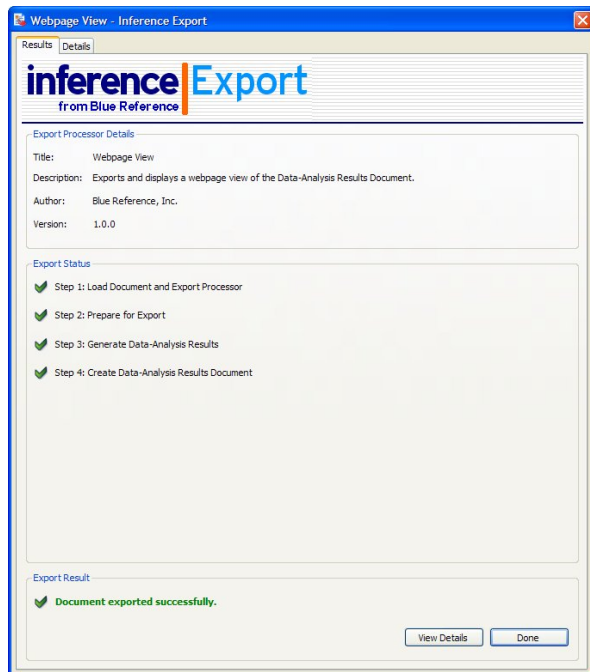
Executing dynamic documents on-the-fly to build results documents with Export Document services is a central function of dynamic documents. Executing a dynamic document is accomplished by selecting the Inference > Export Document function from the Office ribbon. This function provides a choice of four output destinations for results documents:

- **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 Office 2003 or 2007 document files, 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.
- **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.

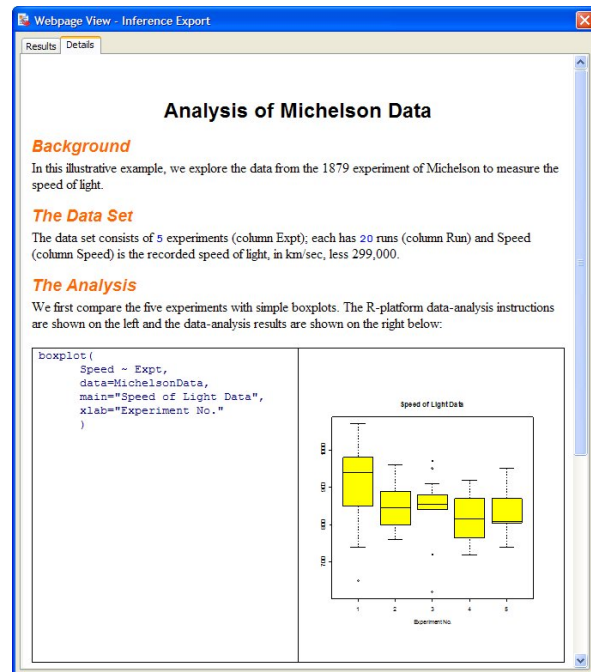
Export Document is an independent process that operates in its own thread independent of inference | in Excel. Operation of Export Document is illustrated in the figure below. Export Document is a highly flexible process that can be extended to include additional and custom destinations by using the inference | SDK.

When the user calls the Export Document function, execution is performed in the following order:

- 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



Export to webpage view: results



Export to webpage view: details

Note that Expressions are evaluated last, which ensures that they are evaluated correctly after any dependent code blocks are evaluated. Hence expressions may be placed anywhere in the document and they will be evaluated correctly. Also, the display of expressions may be aliased to their labels to improve readability of the dynamic source document.

Benefits of Inference Dynamic Documents

Bridge the World of Structured and Unstructured Information

Currently, the information technologies are divided into serving two worlds:

1. The world of unstructured information comprised of free-form text documents stored in document management system such as SharePoint, Documentum, Electronic Lab Notebooks (ELNs) and project/study management systems.
2. The world of structured information comprised of tabular/hierarchical data managed in relational/object databases in conjunction with Laboratory Information Management Systems (LIMS), transactional databases, data web services and hardware devices (e.g., instruments, sensors)

Users need to work within both worlds. Users should not have to choose one or the other. Inference provides a bridge that enables the user to operate within both worlds simultaneously through a single familiar user interface without losing any of the benefits of applying unstructured information and without losing any of the value by enables integration, data mining and reporting on structured information.

Enable Document Indexing and Retrieval Based on Free-Form Text and Structured Fields

The Parts Container in an Inference document is a structured object maintained independently from the formatted free text in the document. For example, a Word 2007 file, which is comprised of zipped package of files, the parts container is held as an XML file in a subfolder entitled "customXml." Given the structured nature of the parts container defined by an XML schema, its contents can be extracted into fields without loss of semantic value. This capability can be the basis for a strategy of indexing and retrieving MS Office documents on the basis of free-form text indexing and structured fields indexing. For example, the open source Lucene indexing engine can be used in conjunction with the open source Tika Content Analysis Toolkit, which is a toolkit for detecting and extracting metadata and structured text (e.g., the inference parts container) from documents using existing parser libraries. The extracted metadata can then be indexed as fields along side with unstructured text indexed as free-form text. This enables the user to query and retrieve documents on the basis of free-form text and structured fields.

Increase Your Return from Investment in Dynamic Scripting Platforms

Organizations have invested in best-of-breed dynamic scripting platforms like R, S-Plus, Matlab and Python, and development frameworks like .NET (C#, VB.NET, IronPython, IronRuby, managed Jscript, and dynamic VB). However, the expertise required to use these platforms and frameworks is confined to a few experts in the organization. Their use is generally restricted to statisticians, programmers and analysts, who develop scripts that generate data extracts and reports for their business user customers. This situation creates a lot of routine work for statisticians, programmers and analysts, but more importantly, it increases the cycle time from data to insight to action on business critical issues. With dynamic documents based on Microsoft Office, you increase your return from investment in dynamic scripting platforms and development frameworks for reasons including the following:

Leverage the Investment in Dynamic Scripting Platforms. Business users can perform predictive analytics on their data in the familiar Microsoft Word environment using statistical and machine learning methods developed by your statistician or analyst on the best-of-breed scripting platforms for predictive analytics.

Reduce Dependence on Statisticians and Analysts. You no longer need to be a statistician or analyst to develop and run a new predictive analytics report. Dynamic documents can walk a user through the process of modifying a dynamic predictive analytics document to the requirements of their objectives and data without assistance from a statistician or analyst.

Extend the Use of Predictive Analytics to Business Users. Dynamic documents put actionable predictive analytics into the hands of business users by using the familiar Microsoft Office environment and exposing a level of detail and complexity which is appropriate for the task and to the end user.

Speed-Up Predictive-Analytics Application Development

In the rapidly changing environment of R&D, there is a need for timely “rapid prototyping” and development of “quick and dirty” predictive analytic applications. Despite the extensibility, flexibility and rich collection of packages available for dynamic scripting platforms, development and deployment of such new predictive analytic applications remains complex and time consuming for reasons including the following:

- requires experts who are greatly outnumbered by non-expert users and whose time commitments are already stretched;
- requires development capability in multiple dynamic scripting platforms because each platform is optimized for restricted application domains;
- requires development of custom data entry and conversion methods, which are typically very time consuming;
- requires rapid-response application deployment, which is typically not consistent with the priorities of organizational IT; and
- requires development of documentation and training materials, which adds to the application-development cycle.

Using dynamic documents in Microsoft Office speeds-up predictive-analytics application development for reasons including the following:

Rapid Application Prototyping Using Pre-Build Predictive Analytic Elements. Applications can be assembled from standard inference | Parts-Container elements—that is, data sets, objects, code blocks and expressions—and laid out on the Office application canvas using the familiar WYSIWYG editor capabilities of Microsoft Office. To assist in the development of custom Parts-Container elements, a fully-featured, graphical integrated development environment (IDE), Inference | Studio, is provided for construction, execution and debugging of dynamic scripting logic in the multiple languages of dynamic scripting platforms.

Simple Data-Entry and Preparation in Excel. Data-entry and preparation can be achieved by tight integration with Microsoft Office applications like Excel.

Painless Application Deployment and Maintenance. Simple installation and maintenance is an important criterion in selecting software solutions. Dynamic document solutions can be deployed as regular documents from a centralized server. Upon opening the dynamic document, all necessary associated components are automatically included in the dynamic document. Optionally, access to the dynamic scripting platform and its associated packages/libraries may be achieved through a low-cost web service.

Single User Interface for Multiple Dynamic Scripting Platforms. Since business processes are highly interdisciplinary, predictive analytic solutions are framed by the strengths of and limitations imposed by the underlying dynamic scripting platform. Accordingly, predictive analytic solutions can benefit from using multiple dynamic scripting platforms. Thus, using a common, familiar client environment like Microsoft Office across multiple scripting platforms is extremely welcome.

Ability to Develop Custom Solutions Using the Inference SDK. Although Inference uses a common Task Pane across Office applications, some organizations may want to design and construct a custom Task Pane to meet their specific requirements. Towards that end, Inference provides an SDK comprising of a pluggable interface for addition of new data-set variants, objects, data-analysis platforms and export services.

Dynamic Documents Enable Reusable Predictive Analytics

In research and development, there is a pervasive need for guided, narrowly-focused, and often single-purpose predictive analytics. Since dynamic documents based on Inference tie together application instructions, data access, data preparation, data analysis, data visualization and results reporting for a specific task, they can be reused by the non-expert user to get the right information, apply the right calculations, create the right visualizations and generate the appropriate report.

Dynamic documents for predictive analytics are easily created in Inference by assembling Inference parts. Such dynamic documents can serve as the basis for standardized and repeatable predictive analytics. Execution of such dynamic documents reliably generates the requisite results document in the choice of format.

Dynamic Documents Implement Literate Predictive Analytics

A principal challenge in the effective use of dynamic scripting platforms is communicating the results of predictive analytics to data owners and stakeholders. Communication is conventionally achieved through reports, generated in two separate steps comprising (1) analyzing the data in specialized applications, and (2) assembling the report with the analytic results (tables, graphs, figures) via copy-and-past to a word processor application.

Dynamic documents integrate and automate predictive analysis and report document generation into a single step. This approach is often referred to as “literate programming”, “literate statistical practice” and “literate data analysis.” Using dynamic documents, result documents are dynamic in that the contents, including figures, tables, etc., can be recalculated each time a view of the document is generated. Additionally, documents can be exported in different forms (e.g., HTML, Word) or can be directed to different receivers (e.g., printer, database, file, e-mail, web service).

For More Information

Visit our websites at

- www.BlueReference.com
- www.InferenceForR.com
- www.InferenceForMATLAB.com
- www.InferenceForDotNET.com
- www.InferenceForQbD.com

or contact Blue Reference via e-mail at

- Info@BlueReference.com
- Sales@BlueReference.com
- Partner@BlueReference.com

or call us via telephone at

- 1-541-316-2343