

Meta5

PC Integration Tools

Version 4.3

Note

Before using this information and the product it supports, be sure to read the general information under “Notices”.

Third Edition (January 2013)

This edition applies to Version 4 Release 3 Modification Level 3 of Meta5, and to any subsequent releases until otherwise indicated in new editions or technical newsletters. Make sure you are using the correct edition for the level of the product.

Order publications through your Meta5 representative or the Meta5 branch office serving your locality.

When you send information to Meta5, you grant Meta5 a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Notices	v
Trademarks	vii

About This Book	ix
Who Should Read This Book	ix
We'd Like Your Comments	ix

Chapter 1. Microsoft Excel	
Integration	1
Excel Spreadsheet Icon	1
Examples of Use	2
Accessing Excel Spreadsheet data	2
Exporting an Excel Spreadsheet icon	3
Producing Formatted Excel Spreadsheets ..	4
Cell Referencing	5
Regions - Defining Region Names in Excel .	6
Using @-Variables with the Excel	
Spreadsheet Icon	7
Multiple inputs to an Excel Spreadsheet. . .	8
Passing data from a Meta5 Spreadsheet into	
an Excel Spreadsheet	10
Passing data from a Word Document into an	
Excel Spreadsheet	12
Passing data from a Query/Transformer tool	
into an Excel Spreadsheet	13
Calculations in an Excel Spreadsheet	13
Creating Totals	13
Performing Calculations	14
Passing Excel Spreadsheet data to an Excel	
Spreadsheet	14
Charts in Excel	15
Creating an Excel Chart	15
Exporting an Excel Chart to a Word	
Document	16
Excel Chart Naming	16
Iterating the Excel Spreadsheet	18
Creating Multiple Excel Spreadsheet	
Reports	18
iExcel Integrator Icon	22
Examples for Use	23
The iExcel Integrator Regions	23
Parameters of the iExcel Integrator	24
Creating a single sheet report	28
Creating a Multiple Sheet Report	34
Creating a Multiple Sheet Report	
Dynamically	34
Values passed to the controls of the Iterating	
Capsule	35
Template Sheets	36
Calculated values	37

Calculating Totals	38
Hyperlinks	39
Pasting Cells	39
Paste Comments	40
iExcel Controls	42

Appendix A. Advanced Commands for iExcel

45	
Command Line Parameters Quick Reference	
Chart	45
Argument / Function	45
Advanced Command Line Parameters	
Examples	46

Chapter 2. Microsoft Word Integration

51	
Overview	51
Word Document Icon	51
What can the Word Document do?	52
Examples of Use	52
Importing a Word Document	52
Exporting a Word Document icon	54
Producing Formatted Word Documents . . .	55
Regions - Bookmarks	55
Using @-Variables with the Word Document	
Icon	56
Multiple inputs to a Word Document	57
Passing data from a Text tool into a Word	
Document	59
Passing data from a Spreadsheet into a Word	
Document	60
Passing an Excel Chart into a Word Document	
63	
Excel Chart Naming	63
Iterating the Word Document	64
Creating Multiple Word Document reports .	64
iWord Integrator Icon	66
Examples of use	67
The iWord Integrator Regions	67
Parameters of the iWord Integrator	68
Examples of Use	70
Paste Bookmarks	72
Paste Picture	72
Including PC graphic files	72
Paste Object	73
Template	73
Data	74
Iterating the iWord Integrator	77
Modifying the Capsule to use @-Variables .	78

**Appendix B. Advanced Commands
for iWord 83**

Command Line Parameters Quick Reference
Chart 83
Command Line Parameter Examples 83

**Chapter 3. Meta5 Email Outbox and
Envelope 87**

Overview 87
Examples for Use 87
 Examples 87
Parameters 88
Envelope Attachments 89
 Examples for Use 89
Iterating Meta5 Capsule for Sending Multiple
Emails 90
Sending data into a region inside the body of the
Meta5 Envelope 92

Chapter 4. iDirList Integrator 95

Overview 95
 Examples for Use 95
 Parameters 96
 Output 96
 Examples for Use 97

Chapter 5. iWeb Integrator 101

The Web Page Scraper 101

iWeb Integrator Icon 101
 What can iWeb Integrator do? 101
 The iWeb Integrator Regions 101
 Parameters of the iWeb Integrator 102
 Example of use 103

Chapter 6. Brio Icon 105

Overview 105
Brio Icon Options 105
Using @variables in Limits in a Brio Report 106
Specifying an Output Section 106
Using the Brio Icon in a Capsule 107

Chapter 7. BusinessObjects Icon . 109

Overview 109
BusinessObjects Icon Options 109
Using @-variables in conditions in a
BusinessObjects Report 110
Specifying an Output Section 110
Using the BusinessObjects Icon in a Capsule 111

Meta5 Publications 113

Glossary 115

Index 117

Notices

References in this publication to Meta5 products, programs, or services do not imply that Meta5 intends to make these available in all countries in which Meta5 operates. Any reference to a Meta5 product, program, or service is not intended to state or imply that only that Meta5 product, program, or service may be used. Subject to Meta5's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the Meta5 product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by Meta5, are the responsibility of the user.

Meta5 may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Meta5, Inc.
122 West Main Street
Babylon, NY 11702
U.S.A.

Licensees of this program who wish to have information about it for the purpose of enabling (1) the exchange of information between independently created programs and other programs (including this one) and (2) the mutual use of the information that has been exchanged, should contact:

Meta5, Inc.
122 West Main Street
Babylon, NY 11702
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Various components of the Meta5 software system require the presence of one or more third party software products, including but not limited to Microsoft Excel for Meta5's Xlaunch, Xtract software product, Microsoft Word for Meta5's WordTool software product, Microsoft Outlook for Meta5's SendOut software product, Brio's BrioQuery for Meta5's AutoBrio software product, Business Objects' BusinessObjects for Meta5's BOConnect software product and/or IBM's Lotus Notes for SendNts software product. No license is granted by Meta5 to our customers for the use of these third party programs in conjunction with our software products. Furthermore, it is the obligation of our customers to ascertain

whether it has sufficient licenses from these third parties, for the third party software in order to utilize the above mentioned Meta5 software programs.

Trademarks

The following terms are trademarks of Meta5, Inc. in the United States or other countries or both:

Meta5

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Microsoft, Windows, Windows2000, Windows 98, Windows ME, Windows XP, Windows NT, Excel, Word, Outlook and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Brio and Brio Query are registered trademarks of Brio Software Inc.

BusinessObjects is a trademark of Business Objects SA.

Lotus Notes and Lotus 123 are Trademarks of Lotus Software, a subsidiary of IBM Corporation.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

About This Book

The PC Integration Tools user guide contains information about Meta5's integration capabilities with Microsoft Word, Excel and Outlook, and the BusinessObjects and BrioQuery tools. These PC Applications may be run from within a Meta5 Capsule enabling the user to create powerful Business Intelligence solutions by utilizing the best of breed tools.

Note: Keep in mind that although the Meta5 environment supports multi-threading of capsules via Capsule Services, the PC Applications we are relying on for integration typically do not support multi-threading.

Who Should Read This Book

The information and instructions contained in this book are written on the premise that you have:

- A working knowledge of PC use and the Microsoft Office suite
- A working knowledge of the Spreadsheet, Capsule, and Transformer tools
- A working knowledge of BusinessObjects and BrioQuery

We'd Like Your Comments

Your feedback is important to us in providing you with the most accurate and high-quality information. If you have any comments about this book or any other Meta5 documentation, please visit our Website at:

<http://www.meta5.com>

There you'll find a feedback page where you can enter comments and send them to us.

Chapter 1. Microsoft Excel Integration

Meta5 has developed an intelligent link with Microsoft's Excel product. Two new icons are introduced with Meta5 version 4.2: the Excel Spreadsheet icon and the iExcel Integrator icon. The Excel Spreadsheet is located in the * Blank Icons file drawer, while the iExcel Integrator is located in the * Integrators Icons file drawer. The *Integrator Icons file drawer is located inside New Icons as well as * Public File Drawers.



Figure 1. Excel Spreadsheet Icon



Figure 2. iExcel Integrator Icon

Some examples for Excel Spreadsheet are available in the Meta5 Sample Applications file drawer (inside New Icons), while examples for iExcel Integrator are located inside the * Integrator Icons file drawer and under the Base Examples folder.

The uses and differences between these two tools will become clearer, later in this chapter.

Excel Spreadsheet Icon

The Meta5 Excel Spreadsheet icon is a Microsoft Excel Spreadsheet residing on the Meta5 File Server. The internal Excel data is held on the Meta5 File Server and passed to Excel when the user double clicks on the Excel Spreadsheet icon.

Note: All workstations and Capsule Servers that need to run the Meta5 Excel Spreadsheet tool must have Microsoft Excel installed and be fully licensed to use

it. No license is granted or implied by Meta5 when using the Meta5 Excel Spreadsheet tool.

This Excel Spreadsheet is accessible from any desktop. This removes the limitation of the previous Meta5 version 4.1 Xlaunch tool of only having the spreadsheet accessible from the PC's local hard drive or mapped network drive.

To provide this Excel Spreadsheet to other non-Meta5 users, it must be "exported" from Meta5 to the PC by dragging and dropping it from the Meta5 desktop into the Windows desktop, by emailing it via Meta5's Envelope and Email Outbox, or by using the Meta5 PC Directory tool.

Examples of Use

- Accessing Excel Spreadsheet data.
- Exporting an Excel Spreadsheet icon
- Producing Formatted Excel Spreadsheets
- Passing data from a Meta5 Spreadsheet into an Excel Spreadsheet
- Passing data from a Word Document into an Excel Spreadsheet
- Passing data from a Query/Transformer tool into an Excel Spreadsheet
- Charting in Excel
- Exporting an Excel Chart to a Word Document
- Creating Multiple Excel Spreadsheet reports
- Creating Multiple Sheets in a Workbook

Accessing Excel Spreadsheet data.

The Excel Spreadsheet icon can be found in * Blank Icons within the New Icons file drawer.

To make use of an existing Excel Spreadsheet located in Windows, simply drag and drop the required Microsoft Excel icon from Windows Explorer or from the Windows desktop onto the Meta5 desktop and then into a Meta5 Capsule. This will result in a Meta5 Excel Spreadsheet icon, which is a copy of the Microsoft Excel Spreadsheet, being placed in the Capsule. From this point onwards, the Meta5 Excel Spreadsheet is stored on the Meta5 File Server and is not linked in any way to the original spreadsheet.

Alternatively, you may wish to automate your Capsule so that it dynamically picks up the contents of an Excel Spreadsheet and passes it to the next icon for further processing.

Placing a PC Directory tool in front of the Excel Spreadsheet icon and connecting them together can achieve this. Configure the PC Directory tool to access the

directory where your external data resides, and set the source region, on the arrow connecting them, to the name of the Excel spreadsheet you want. See Figure 3.

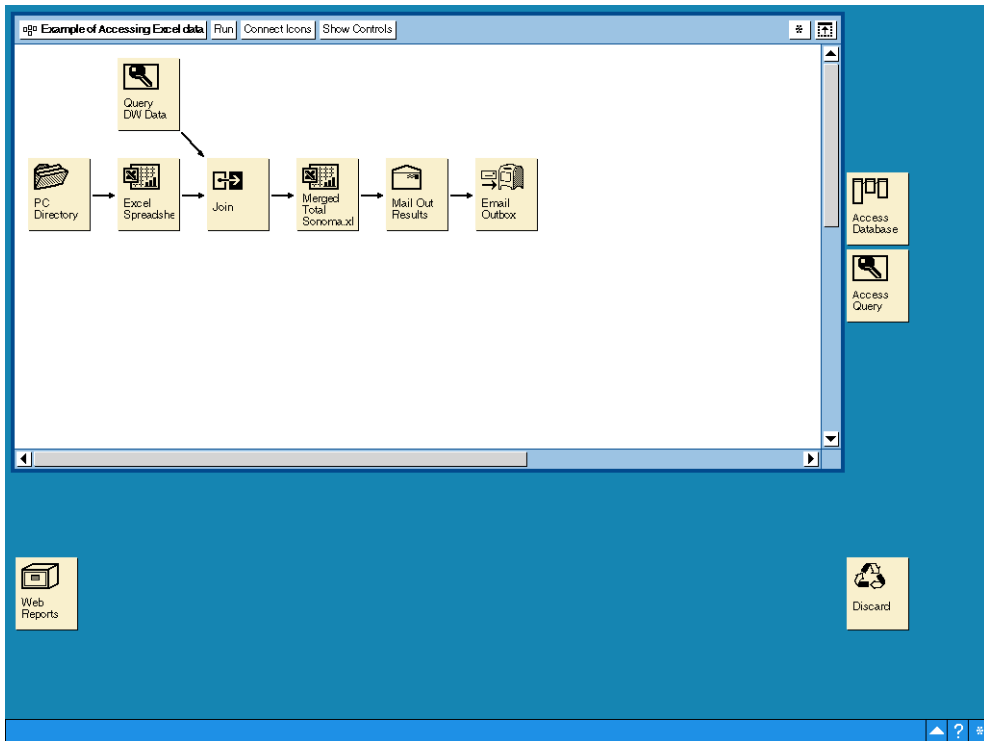


Figure 3. Accessing an Excel Spreadsheet via the PC Directory tool

The example application above joins the data on the Corporate Data Warehouse with data residing in an Excel Spreadsheet and, following some formatting, places the results in another Excel Spreadsheet, which is mailed, over the Corporate Email system, to the recipients.

This application demonstrates how a Meta5 Capsule can merge data between the corporate data warehouse and ad hoc data derived from the field or other source and distribute this over the corporate email system.

Exporting an Excel Spreadsheet icon

As mentioned earlier, the Meta5 Excel Spreadsheet icon resides on the Meta5 File Server. The example above, demonstrated how you could email a resultant Excel Spreadsheet out across your corporate email system to any recipient.

Another way, and perhaps more relevant if the spreadsheet is required to be processed further outside of Meta5, is to direct the Excel Spreadsheet through a PC Directory tool to your PC's hard drive (or networked drive).

Configure a PC Directory tool to address the directory where you wish to store the Excel Spreadsheet. Connect the Excel Spreadsheet to the PC Directory icon and configure the arrow connecting them to specify the resultant spreadsheet filename in the destination region. See Figure 4.

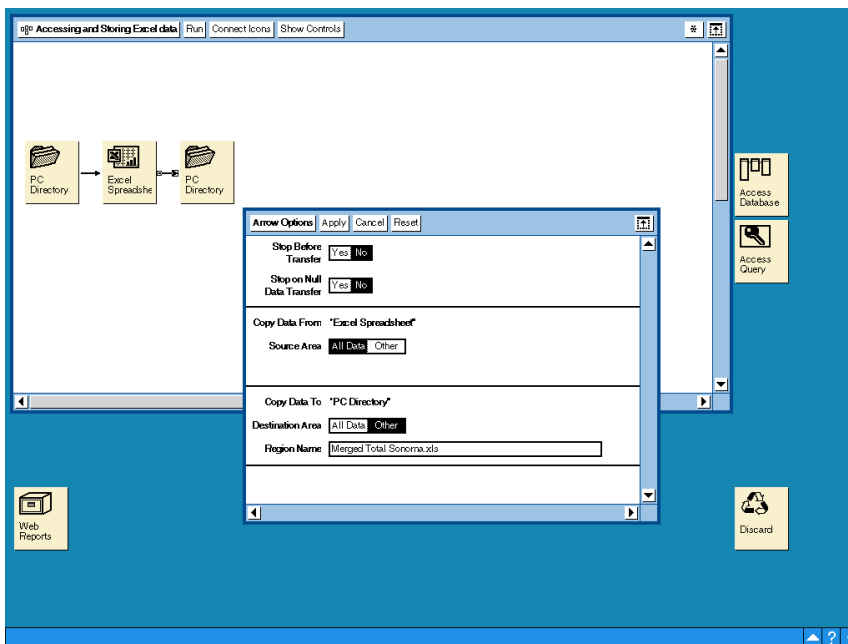


Figure 4. Placing a Meta5 Excel Spreadsheet on to the PC's hard drive

Producing Formatted Excel Spreadsheets

We have seen how easy it is to transfer data into Meta5 from MS Excel. If the destination is another Excel Spreadsheet then the contents of the source will be replicated in the target, including the formatting. If the destination is another Meta5 tool then the result is a raw dump of the data without the formatting.

Similarly, an Excel Spreadsheet that is created by Meta5 will have no formatting by default. The following section describes how to format your resultant Excel Spreadsheet for onward delivery to a recipient or for onward processing by another application.

Formatting of the Excel Spreadsheet is a one-time operation. Once the designer has completed the layout and formatting of a resultant Excel Spreadsheet it is ready to be used in a Capsule. The Capsule can be run any number of times without the need to revisit the formatting.

Cell Referencing

You can specify a cell reference or range in the region name of the arrow option either as input to an Excel Spreadsheet or output from it.

Examples:

A3	Single cell reference in the active sheet.
B3:F10	Range of cells in the active sheet.
Sheet3!C2:J250	Range of cells in Sheet3
Sheet2!B5	Cell B5 in Sheet2

Note: The active sheet is defined as the sheet that displays when you open an Excel Spreadsheet. In other words, the sheet that was last modified before the spreadsheet was closed.

In the case of exporting data out of an Excel Spreadsheet icon, the cell or range you specify limits the data cells that will be transferred.

If you specify a single cell reference in a Destination Region Name (importing data into an Excel Spreadsheet) then it is assumed that this is the top left starting point of where all the data in the Source Region should be located in the target.

You can restrict how much data is loaded to the target spreadsheet by specifying a range of cells to load to. If the range you specify is smaller than the data being transferred in, then the data is truncated. If the range is larger than the data being transferred, the extra cells are padded with #N/A's.

In the same way that formatting is preserved within a Defined Name, using a cell reference will also preserve formatting.

However, unlike a Defined Name, using a cell reference will not cause an error to be produced if existing data lies within the area to be populated. The data is simply overwritten.

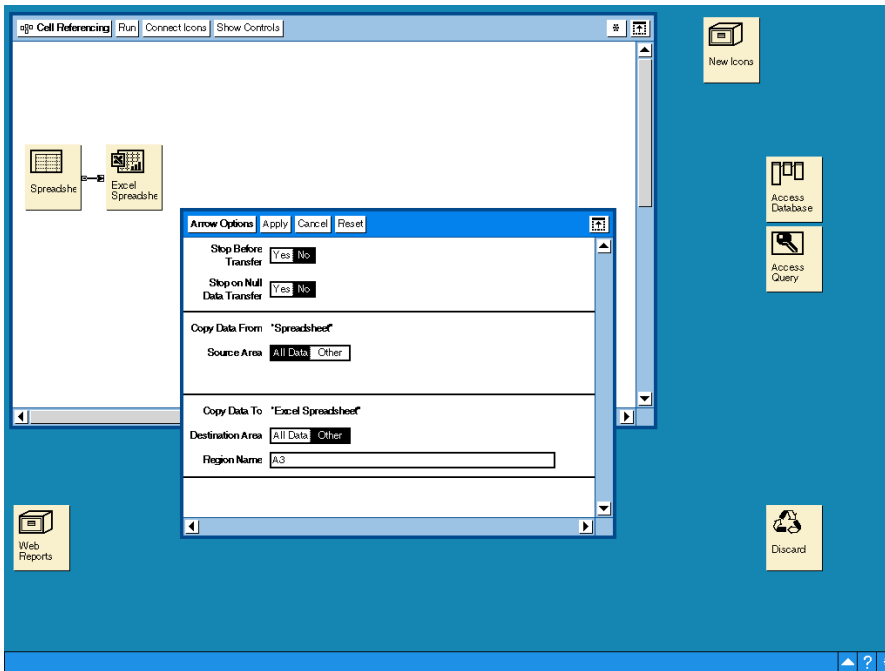


Figure 5. Using a Cell Reference in a Destination Region of the Arrow option

Regions - Defining Region Names in Excel

As with the Meta5 Spreadsheet, Text and other tools, the Meta5 Excel Spreadsheet understands the concept of regions as well.

The equivalent concept in the Excel Spreadsheet is the Excel Defined Name. This defined name can be set up in the Excel Spreadsheet by first, highlighting the rows and columns you want named and then by going to the Insert menu and selecting Name and then Define.

You are presented with the following window that allows you to give a name to the row/cell range you have selected.

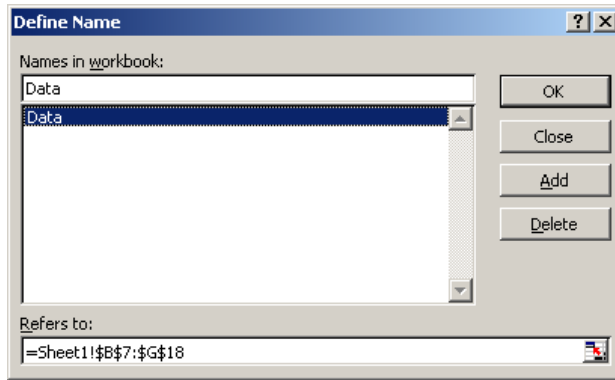


Figure 6. Inserting a Defined Name in Excel

Once an area is named in this way, you can address this area directly by using the Name in the source and/or destination regions of the arrows going in to/out of the Excel Spreadsheet.

By using the regions like this, Meta5 will preserve any formatting contained within the Defined Name and apply this formatting to any incoming data.

The Defined Name (Region) will expand to accommodate the data being transferred into it. Although the Defined Name expands to accommodate the data, any formatting specified within the Defined Name will be maintained but will not be extended to include any additional rows or columns of data. It is your responsibility to ensure that the maximum amount of data is catered for when formatting it, or you should consider using the iExcel Integrator for formatting dynamic amounts of data.

Note: The automatic expansion of a Defined Name will fail if there is an attempt to overwrite existing data in the spreadsheet. An error message will be produced to this effect. Consider using cell referencing to overwrite existing data.

Any data, including its format, which falls outside the Defined Name, will be maintained in the result. Headings, narrative and comments may be placed in this way.

Using @-Variables with the Excel Spreadsheet Icon

Meta5 @-variables can be used with the Excel Spreadsheet tool. They only have relevance to the Arrows coming into and out of an Excel Spreadsheet but it can prove to be a very useful facility.

You might consider using @-variables when choosing which location in the resultant spreadsheet to populate. In other words, an @-variable may be set to different Defined Names or Cell References/Ranges dependent on the logic within a Capsule. Perhaps negative profits need to be reported in a separate location or sheet within the Excel Spreadsheet.

Multiple inputs to an Excel Spreadsheet

The Excel Spreadsheet icon can accept input from a large number of Meta5 icons. These icons include, but are not limited to:

Tool	Typically Used For
Spreadsheet	Transferring in: Data, headings, narrative, comments and instructions.
Excel Spreadsheet	Transferring in: Data, headings, narrative, comments, instructions and graphics.
1-2-3 Spreadsheet	Transferring in: Data, headings, narrative, comments, instructions and graphics.
Query	Transferring in: Data
Brio Query	Transferring in: Data, headings, narrative, comments and instructions
Business Objects Query	Transferring in: Data, headings, narrative, comments and instructions
Reporter	Transferring in: Data
SQL Entry	Transferring in: Data
Word Document	Transferring in: Data, headings, narrative, comments and instructions
Transformers	Transferring in: Data, headings

We can therefore envisage the inputs to various parts of an Excel Spreadsheet as coming from a number of different sources. The headings may come from a Word Document whereas the data may come from the Data Warehouse. A title may be constructed from @-variable settings, for example the Product being reported on, and therefore a Spreadsheet is the most likely source. Lastly, instructions or comments may be included from another Excel Spreadsheet.

The following Capsule demonstrates this concept.

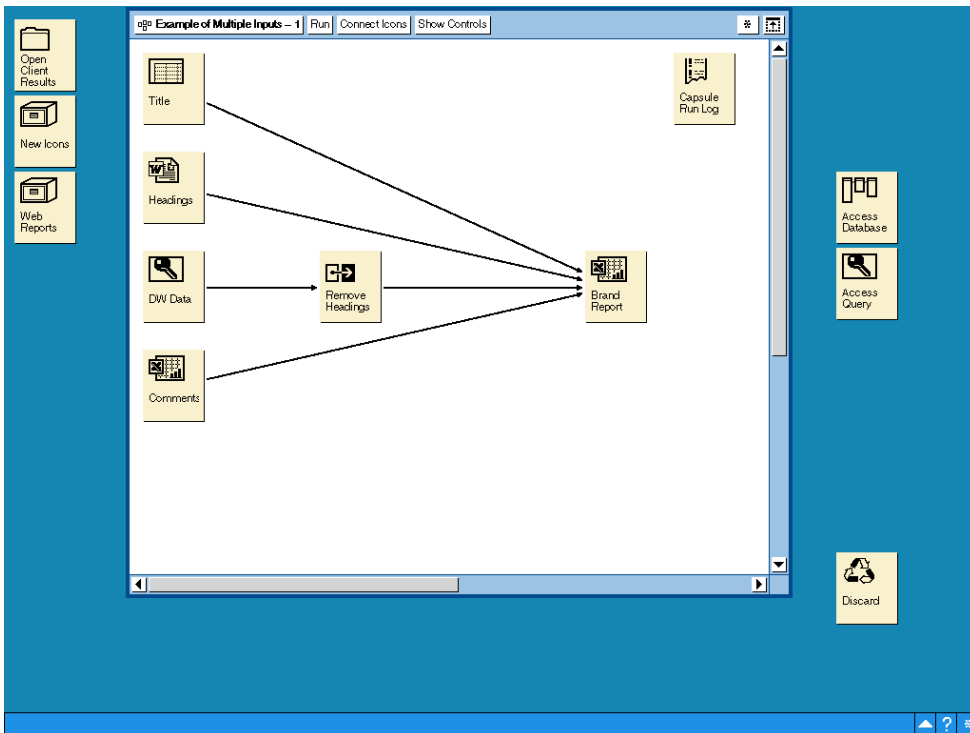


Figure 7. Example of multiple inputs to the Excel Spreadsheet icon

Let us next take a look at the resultant spreadsheet. This spreadsheet, as can be seen in Figure 8, contains a title, headings, data, calculated values, chart and textual comments (which includes a hyperlink to <http://www.OurCoffee.com>).

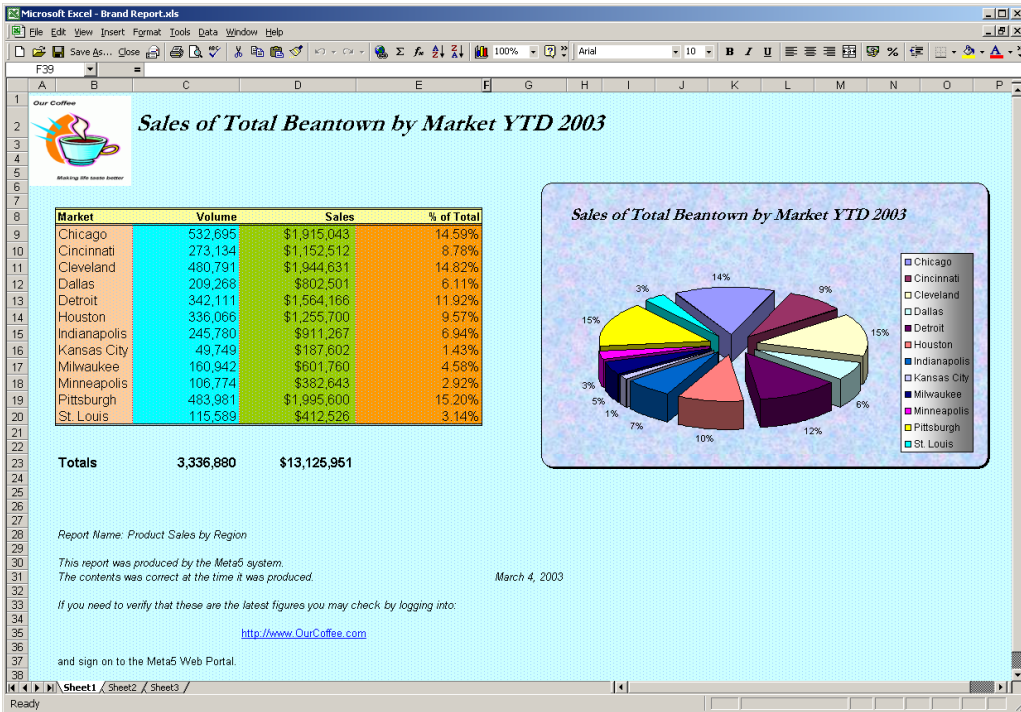


Figure 8. Resultant spreadsheet from the example above

Passing data from a Meta5 Spreadsheet into an Excel Spreadsheet

The title is calculated within a Meta5 spreadsheet (called "Title") from the @-variable containing the Product Description. This has therefore made this Capsule independent of the Product Description chosen. Refer to Figure 9 where the formula is displayed that generated the dynamic title.

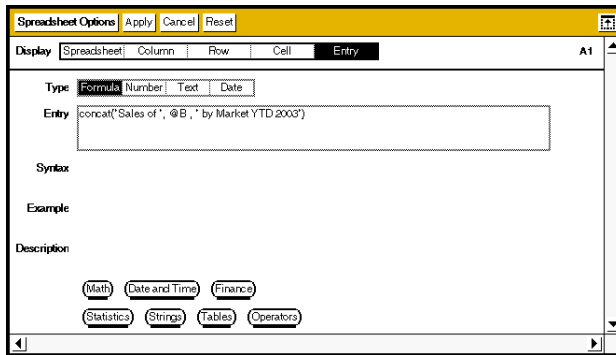


Figure 9. Building the variable title employing @-variables

Make sure you have set up the formatting required in the resultant spreadsheet in the location you wish to place the data.

Connect the spreadsheet to the Excel Spreadsheet. Ensure the data is going to the location you have chosen by specifying a Defined Name, Cell or Cell Range coordinates on the arrow's destination region name.

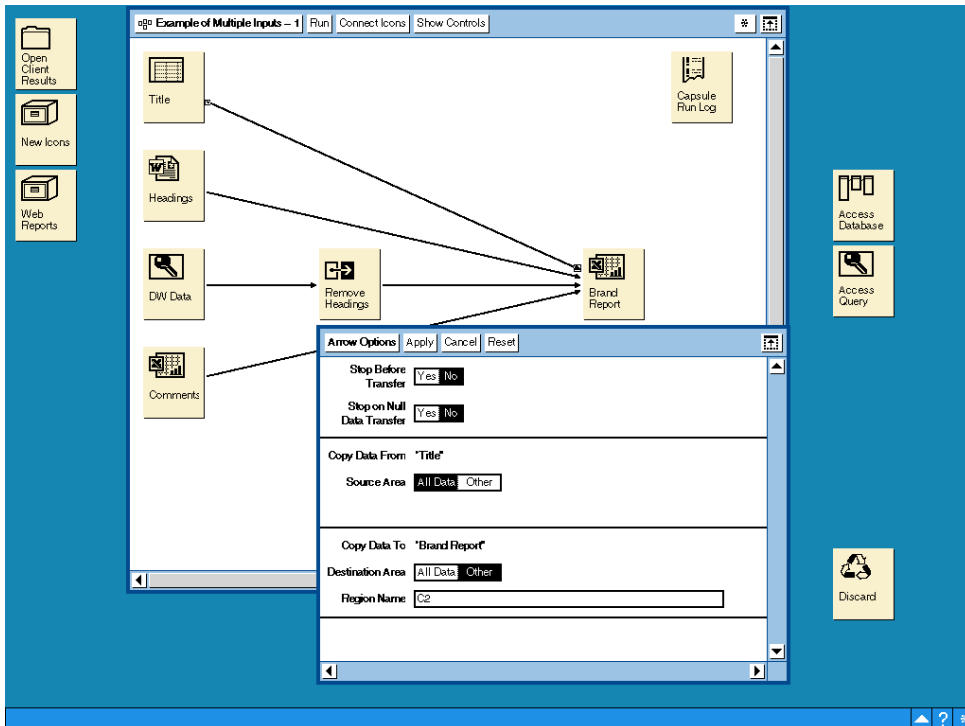


Figure 10. Specifying the location of where the title data should be located

Passing data from a Word Document into an Excel Spreadsheet

The headings have been derived from a Word document, which simply contains the headings separated by tabs.

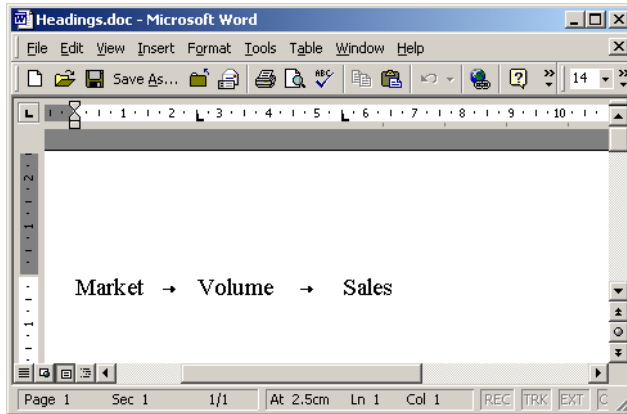


Figure 11. Tab separated values ensure the headings occupy different cells in the spreadsheet

This ensures that each column heading goes into a separate cell in the resultant spreadsheet starting at the Cell Reference specified on the destination region name of the arrow options.

The example, see Figure 12, uses a range of cells to limit the number of rows and columns to be transferred. In this case we have a finite number of cells to occupy so are able to fix the coordinates to A5:C5.

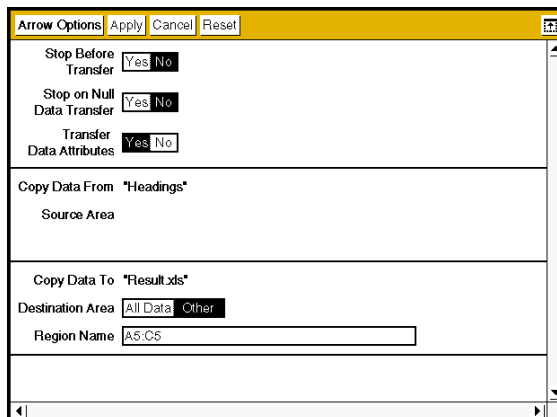


Figure 12. Specifying the location of where the headings data should be located

Passing data from a Query/Transformer tool into an Excel Spreadsheet

In the example we are using, the final spreadsheet will contain the column headings from the Word Document. The Query tool will pass the column names, derived from the WTDD, onto the next icon. We wish to use our own column headings and therefore must remove the Query generated headings. As this row of headings is the first row of data from the Query tool, we can simply strip or delete the first row using the Compress transformer.

Arrow Options	
Stop Before Transfer	<input checked="" type="radio"/> Yes <input type="radio"/> No
Stop on Null Data Transfer	<input checked="" type="radio"/> Yes <input type="radio"/> No
Transfer Data Attributes	<input checked="" type="radio"/> Yes <input type="radio"/> No
Copy Data From	"Remove Headings"
Source Area	<input checked="" type="radio"/> All Data <input type="radio"/> Other
Copy Data To	"Result.xls"
Destination Area	<input checked="" type="radio"/> All Data <input type="radio"/> Other
Region Name	<input type="text" value="Data"/>

Figure 13. Defined Name to identify the location of the Data Warehouse data

Notice, in Figure 13, that we have supplied a Defined Name called Data. This has been defined as cell range B8:D20 in the resultant spreadsheet. We have used a Defined Name here to allow for variable numbers of rows of data coming from our Query. The Defined Name will expand or contract to accommodate the data passed to it.

Calculations in an Excel Spreadsheet

Any calculation required in the result, such as the Totals row or % of Total column in our example in Figure 8, should be predefined and placed outside any location that will have data loaded into it.

Creating Totals

Since the data may have a variable number of rows, we must place the Totals row beyond the maximum row expected. Simply choose the row and create the summations needed using the Sigma key Σ on the Excel toolbar.

Performing Calculations

In our example we are passing the data into a defined name that spans columns B, C and D. We can therefore use column E to place our calculations of % of Total.

We cater for the variable length input by predefining as many calculations in column E as we expect rows of data from our data source. To avoid these calculations from displaying errors where a row is not populated from our source, we enclose them in an IF statement that will display the calculation if data exists in column B (Market) for that row.

e.g. `IF(B9="", "", D9/D23)`

Note that the divisor (Totals) is in a fixed location (`D23`). This means the Totals line must be in a fixed location. The iExcel Integrator tool is designed to be used for more sophisticated layouts.

Passing Excel Spreadsheet data to an Excel Spreadsheet.

In our example we are passing a number of cells from an Excel Spreadsheet into our resultant spreadsheet. We are interested in only the content of the Comments spreadsheet and do not desire the heading and author details. See Figure 14 for the content of the Comments spreadsheet.

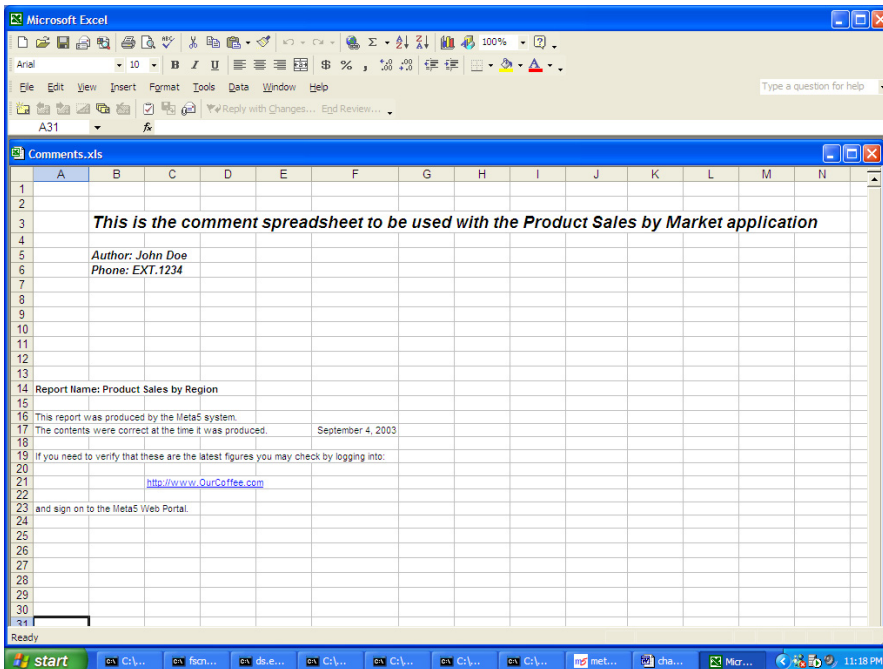


Figure 14. Contents of the Comments spreadsheet

We can therefore, restrict the input to just the cells A14:F23 by specifying this range on the Source Region Name of the Arrow options going out of the Comments spreadsheet.

We can further control the destination of these cells by specifying the cell range on the Destination Region Name on the Arrow options going into the resultant spreadsheet (see Figure 15).

Note the formatting for the hyperlink and date were performed on the resultant spreadsheet. The formatting of the original data (Comments spreadsheet) is not preserved when transferring into an Excel Spreadsheet.

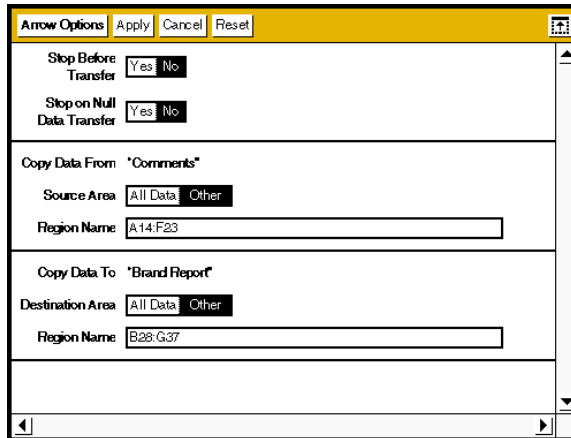


Figure 15. Specifying Cell Ranges on the source and destination region names

Charts in Excel

We have seen, from our example earlier, that the Excel Spreadsheet can contain charts.

There are several new features within Meta5 4.2 that allow the user to employ and manage Excel Charts within the Excel Spreadsheet and export them to other Meta5 tools.

Creating an Excel Chart

It is not the purpose of this manual to teach how to use Microsoft Excel. However it would be useful to include some basic information on how to build a simple chart and how charts can be used within Meta5.

If we use our earlier example, see Figure 8, we can see that the chart is based upon the data contained within the Defined Name Data. The easiest way to generate a chart from the data would be to populate the resultant spreadsheet with all of the data and text required, ensuring that inputs are directed to specific

locations within the spreadsheet to preserve any formatting you are about to apply.

Now we are ready to format the results. During this phase of the application development, we can set up the Excel chart. Highlight the rows and columns you wish to have included in your chart. Typically this includes the column headings and the data below it. Once the data is highlighted, click on the Excel Chart Wizard, located in the Excel toolbar. Follow the instructions to tailor the chart to your needs.

Once the chart is produced, after clicking Finish in the Chart Wizard dialogue, move the chart to the position you want it to be displayed.

Important note: The data is being passed into an Excel Defined Name called Data. This Defined Name will expand and contract to accommodate the size of the data coming in. Since we have included this Defined Name in the area to be included in our chart, the chart will adjust to accommodate the volume of data automatically.

Note that if you do not use a Defined Name in this way, then the Chart will remain static in terms of the rows and columns to be included. This may result in data being left out of the chart or the chart containing empty sections.

We will see later how the iExcel Integrator can be used to ensure charts are always complete.

Exporting an Excel Chart to a Word Document

The Excel Chart can be placed in a Word Document to enhance the appearance of a report.

Two things are needed to complete this action. First we need to know the name of the chart so that we can specify it in the Source Region Name of the arrow connecting the source Excel Spreadsheet to the Word Document.

Excel Chart Naming

Excel names its charts in the format "Chart x", where x is a number starting from 1 that denotes each chart in the workbook. The first chart in an Excel Workbook is named Chart 1, the second Chart 2 and so on. They are numbered sequentially as they appear in the Workbook starting from the top left cell in Sheet1 and working its way left to right and top to bottom of each sheet in turn.

Secondly we need to define a destination in the Word Document to contain this chart. We create a bookmark inside the Word Document at the location we would like the Excel chart to appear. The bookmark can be formatted but since it is a graphic we are restricted in what we can do to it. One formatting characteristic that may prove useful is the ability to center or right justify the bookmark. Generally though, any formatting you apply will be maintained and used by Meta5.

Lastly connect the two icons together and provide the name of the Chart as the Source Region Name and the bookmark name as the Destination Region Name in the Arrow options. Figure 16 demonstrates this procedure.

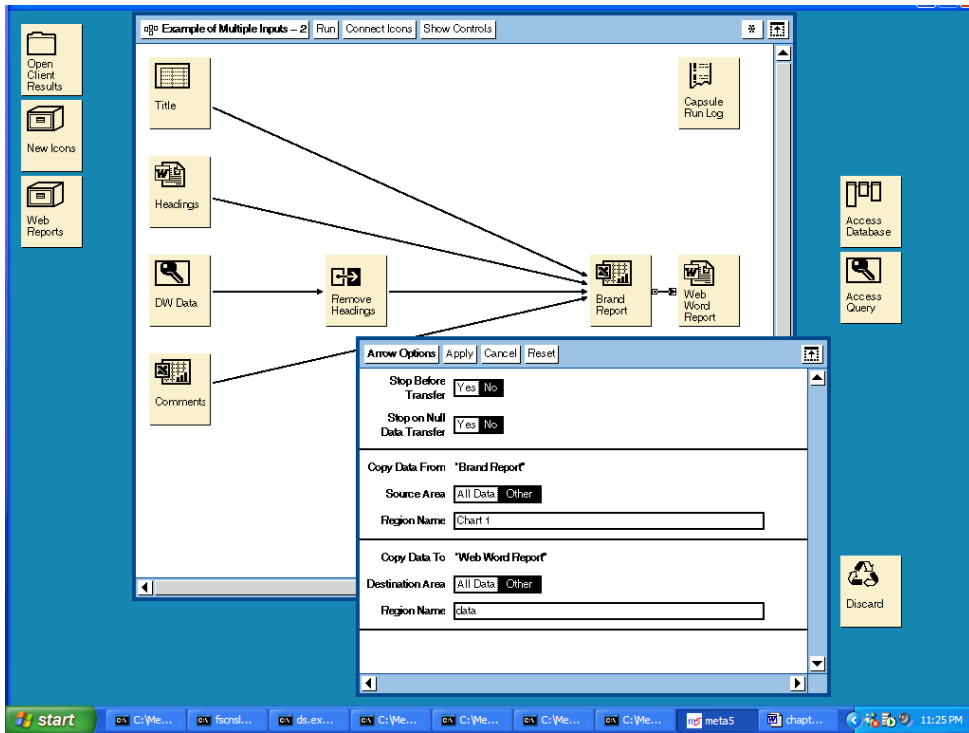


Figure 16. Directing an Excel Chart into a Word Document

The following figure shows the result of transferring Chart 1 from an Excel Spreadsheet into a Word Document's bookmark.

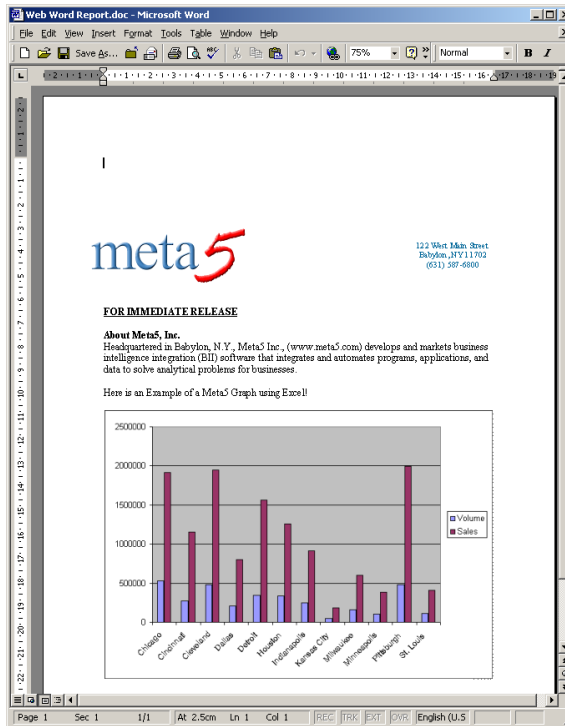


Figure 17. An Excel Chart inside a Word Document

Iterating the Excel Spreadsheet

The Meta5 Iteration capability is a very powerful mechanism to allow Capsules to run multiple times with a single click of the mouse. Iteration comprises a Capsule to be run multiple times and a controlling icon to pass the values to be used in each iteration of the Capsule. These values are passed through the Meta5 @-variables that are defined in the Capsule controls.

You may wish to consider two types of application of the iteration capability and the Excel Spreadsheet. Firstly, we can use iteration to produce multiple Excel Spreadsheets. Secondly we can use iteration to build multiple sheets within a single workbook.

Creating Multiple Excel Spreadsheet Reports

Let us take the example used earlier in this chapter to demonstrate the iteration capability of the Excel Spreadsheet.

The first step with any iterating Capsule is to make the Capsule dependent on @-variable assignments rather than constants. For example, constraints in a Query or Reporter, Headings and Titles, naming of the resultant Excel Spreadsheet and

recipients of an Email. All of these concepts are dealt with in detail in the Meta5 Capsule Users Guide.

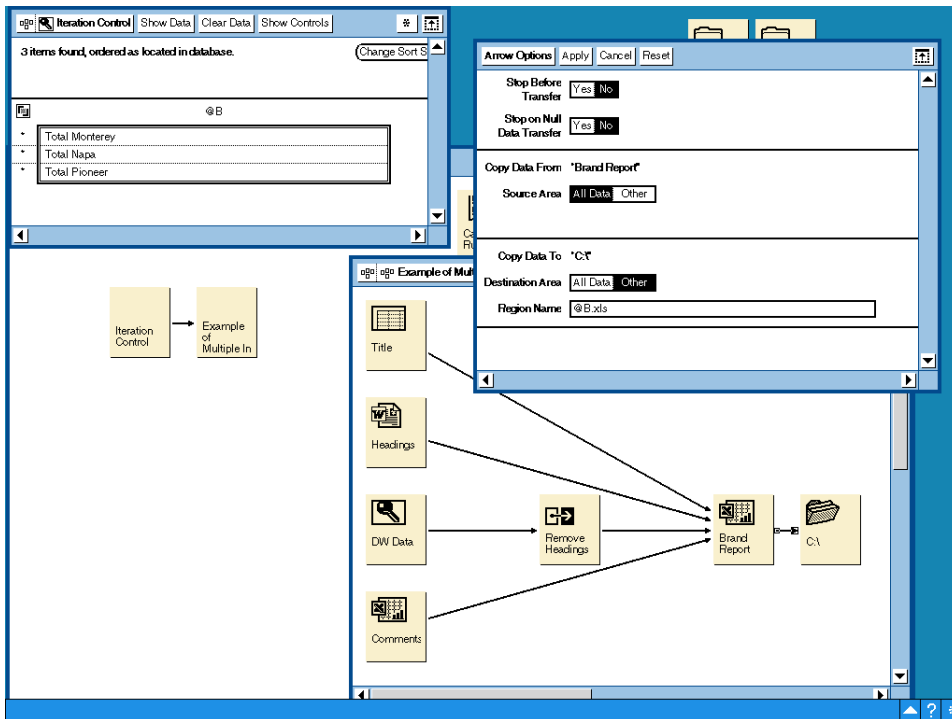


Figure 18. Iterating the Excel Spreadsheet to produce multiple reports

In Figure 18, we can see that the iteration control Query is passing the Brands Total Monterey, Total Napa and Total Pioneer in @-variable @B to the inner Capsule. This inner Capsule will be executed three times, once for each value of @B. The resultant spreadsheet is exported to the local C: drive of the PC and is given a name to reflect the brand being processed, namely "@B.xls".

On completion of the Capsule, three spreadsheets will reside on C:\ and will be named: Total Monterey.xls, Total Napa.xls and Total Pioneer.xls.

Some forward planning must be undertaken before we can create multiple sheets in a workbook. We will see later in this chapter how the iExcel Integrator can dynamically create sheets within a workbook. However the Excel Spreadsheet icon must have all of the sheets predefined in the workbook for this to work.

Once all the sheets are defined and named, we can populate the different sheets by iterating the Capsule and including the name of the sheet to be populated in the controls of the Capsule.

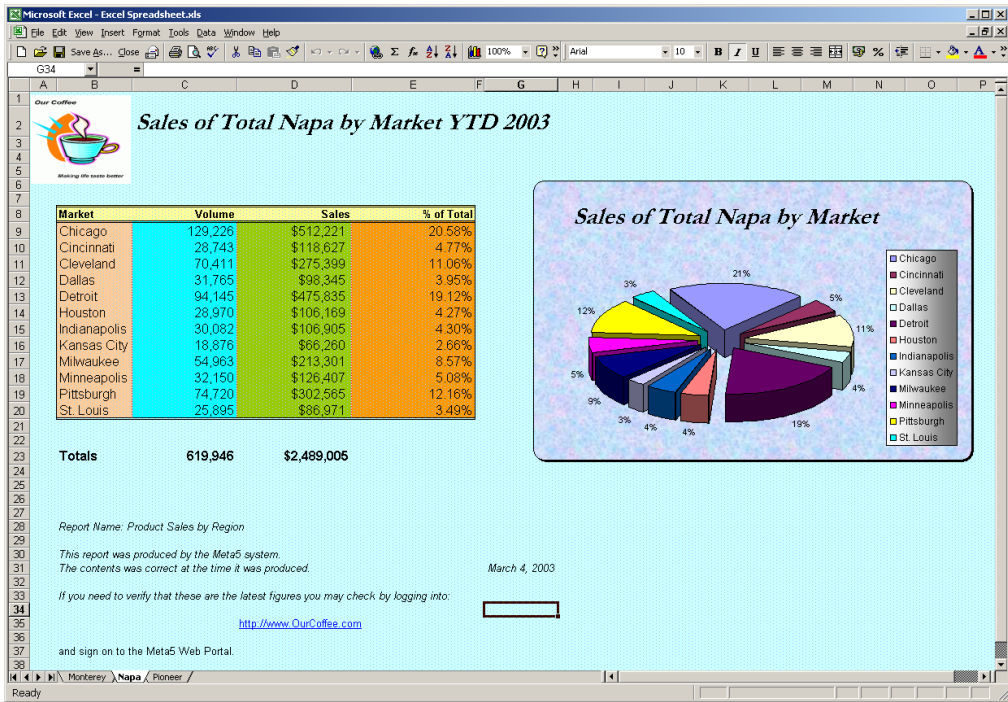


Figure 19. Pre-define the Sheet Names and any Defined Names to populate multiple sheets in a workbook

Figure 19 shows three sheets have been defined: Monterey, Napa and Pioneer. As we are directing the data from the Data Warehouse to a Defined Name, we must define three unique Names, one in each sheet, to accept the brand data. We have named them so that they share the same name as the sheet in which they are located. This means we only need to pass the name of the Brand in @B (as before) and the name of the sheet/defined name in @C (see Figure 20).

	A	B	C
1	@B	@C	
2	Total Monterey	Monterey	
3	Total Napa	Napa	
4	Total Pioneer	Pioneer	
5			
6			
7			

Figure 20. Passing the Sheet Name/Defined Name as @-variable @C

Lastly, we need to ensure that all of the inputs are directed to the appropriate sheet within the workbook. We do this by fully qualifying the source and destination region names with the name of the sheet. The name of the sheet is defined in @C. We therefore use @C when qualifying the source and destination ranges (see Figure 21).

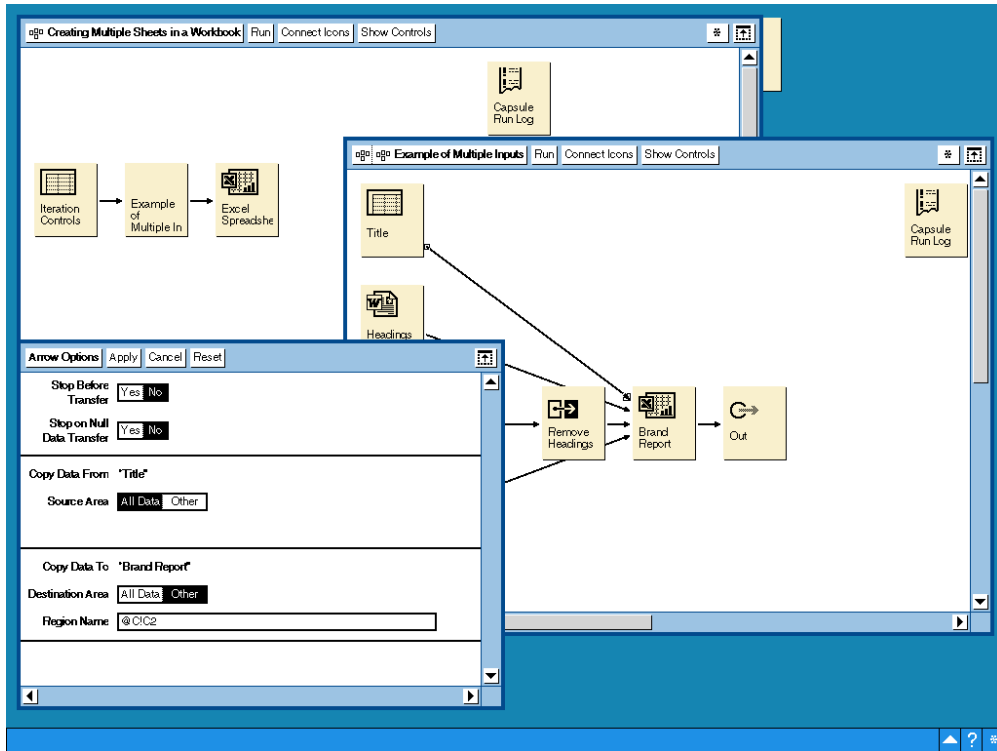


Figure 21. Use @C as the Sheet Name in the Destination Region Name

In Figure 21 we see the fully qualified destination range for the "Title" input specified as:

@C!C2

@C denotes the sheet name (Note it precedes an exclamation mark "!" to indicate that this is a sheet name) and C2 the cell reference within that sheet.

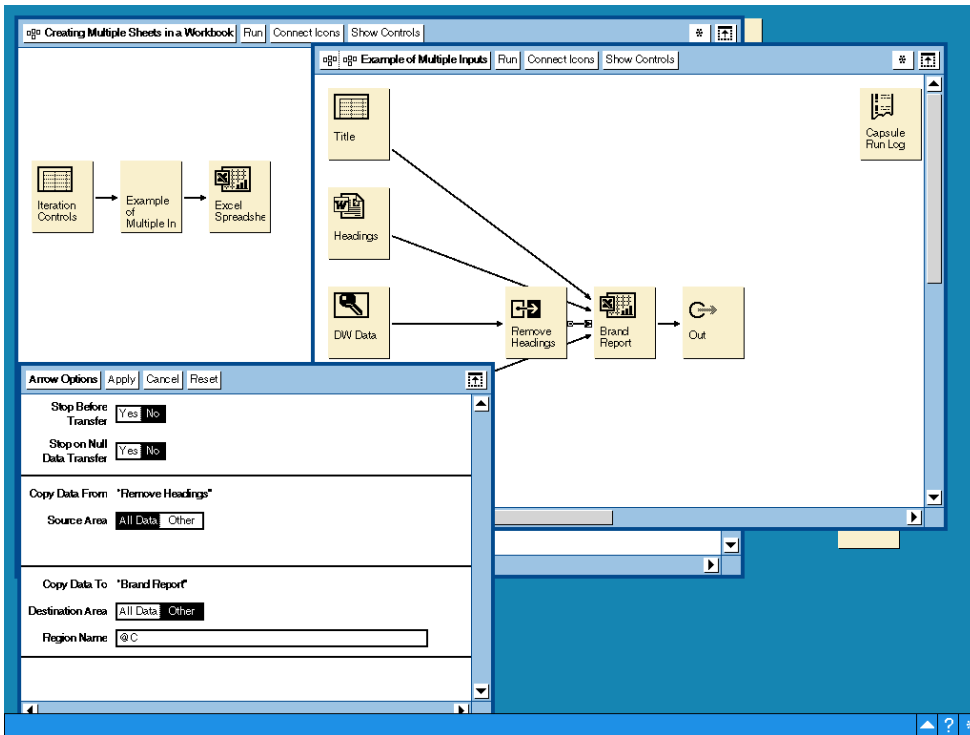


Figure 22. Use of the @-variable, @C, to point the Data Warehouse data at the appropriate Defined Name within the resultant Workbook

Note: We do not have to qualify a defined name with a sheet name because each defined name must be unique within a workbook.

Lastly, note that we have placed an Out icon in the inner Capsule and an Excel Spreadsheet after the Capsule in the outer Capsule. This allows the resultant Excel Spreadsheet to be copied out to the outer Excel Spreadsheet making it easier to view the results or pass on to the next Meta5 tool for distribution or further processing.

iExcel Integrator Icon

The iExcel Integrator icon was introduced to provide the Meta5 user with more power and control over the creation of Excel Spreadsheets. We have seen with the Meta5 Excel Spreadsheet icon that a great deal can be achieved using this tool. We will now review the capabilities of the iExcel Integrator and how they can be used to dynamically create multi-sheet workbooks catering for variable numbers of rows input to each sheet.

The Integrator family of tools has a similar look and feel to the Meta5 Transformers. The Integrator icons have input and output regions and a controls panel that allows you to configure the tool.

Examples for Use

- Creating Excel Reports with dynamic data and charts that adjust for the dynamic data
- Creating Excel Reports with a dynamic number of multiple worksheets
- Creating sophisticated Excel reports with a myriad of formats and comments
- Creating sophisticated Excel reports without relying on Visual Basic programming of complex macros in Excel.
- Saving Excel reports as other file types such as .csv

The iExcel Integrator Regions

Figure 23 shows the window displayed when you open the iExcel Integrator icon. Like the Meta5 Transformer, this window contains the input and output regions and where appropriate can display the contents of each region in the display area. Unlike the Meta5 Transformer, the Integrator icons have no Run button in the window header.

There are four input regions and one output region.

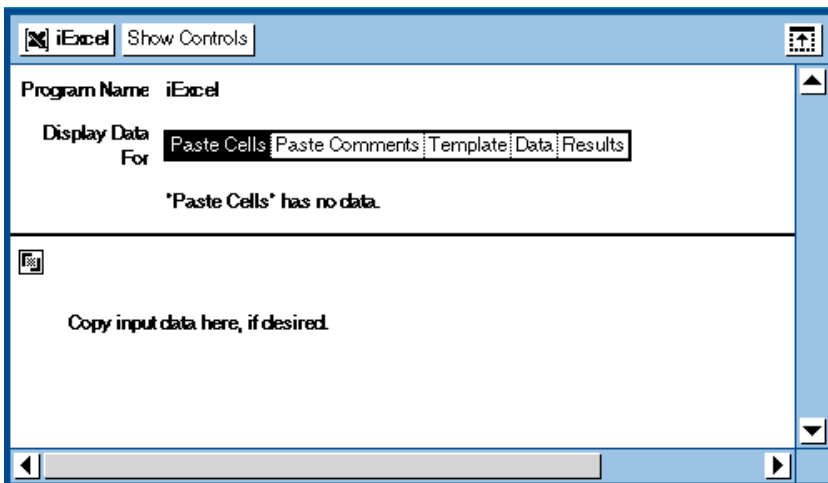


Figure 23. Input and output regions of the iExcel Integrator

These regions have the following functions:

Paste Cells	A spreadsheet containing cells populated with data that is to be copied into the result. Where data exists in the Paste Cells region, it is copied to the same location in the Results region.
Paste Comments	A spreadsheet containing data to be copied into the result as comments. Where data exists in the Paste Comments region, it is copied as a comment to the same relative location in the Results region.
Template	Excel Spreadsheet icon containing all the charts, formatting and Defined Names for the Results.
Data	The input data provided from a source such as Query, Reporter, SQL Entry, Excel Spreadsheet, Meta5 Spreadsheet, BrioQuery, Business Objects Query, Windows Application and numerous Meta5 Transformers.
Results	The output from the iExcel Integrator after all of the input regions and Program Control parameters have been processed.

The iExcel Integrator loads up each of the inputs and processes them in a work area.

It uses the Template spreadsheet as the base and loads each of the inputs over the top of it. The completed spreadsheet is then moved into the Results region.

Note that the order in which the source input regions are run into iExcel is important. The "Data" input region requires that it is run last.

Parameters of the iExcel Integrator

Click on Show Controls to review the parameters of the iExcel Integrator (see Figure 24).

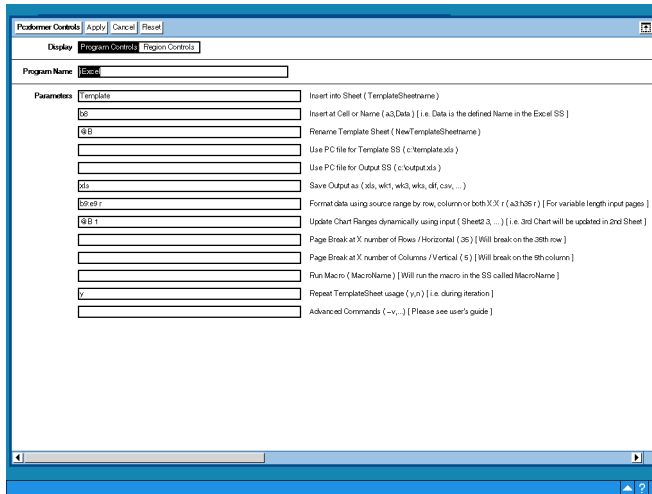


Figure 24. The Program Controls of the iExcel Integrator icon

- | | |
|-----------------------------|--|
| Program Name. | iExcel. Name of the program to run. This should not be changed. |
| Insert into Sheet | Specify the name of the Sheet in the Template file to contain the results. If left empty then the Integrator defaults to the active sheet in the Template spreadsheet. (Note: the active sheet is the one that opens up by default when you open an Excel Spreadsheet) |
| Insert at Cell or Name | Declare the start position, range of cells or Defined Name where the Data region should be loaded into the Template. This location is further qualified by the preceding parameter "Insert into Sheet". |
| Rename Template Sheet | When you provide a name in this parameter, the sheet name specified in the first parameter (or active sheet if first parameter left blank) is renamed to the contents of this parameter. |
| Use PC file for Template SS | An alternative to supplying an Excel Spreadsheet as input to the Template region. Specify the full path and file name of the Excel spreadsheet that is to be used as the Template. |

Note: The Template specified in this parameter will take priority over any Template connected to the Template region.

Use PC file for Output SS	<p>An alternative to supplying an Excel Spreadsheet as output to the Results region. Specify the full path and file name of the Excel spreadsheet that is to be used to contain the results.</p> <p><i>Note: If you use this parameter to specify a result spreadsheet, you can also use the output from the Integrator. Both will be produced.</i></p>
Save Output as	Use this parameter to create the output in a different format to the default xls type. Other file types you can specify here are: txt, wk1, wk3, wks, dif and csv.
Format data using source range	Use this parameter to specify which cell range should be used as the formatting template for all the data coming into the tool. This parameter is particularly useful when the size of the input is not known. There are three keywords that can follow the cell references.
A9:C9 Row	<p>This will replicate the formatting in cell A9 to C9 for every row of the input data in columns A through C. You may also state this parameter as</p> <p>A9:C9 r.</p>
B8:B9 Column	<p>This will replicate the formatting in cell b8 to b9 for every column of the input data in row 8 and 9.</p> <p>You may also state this parameter as</p> <p>B9:B9 c.</p>
B8:B9 Both	<p>This will replicate the formatting in cell b8 to b9 for every column and row of the input data.</p> <p>You may also state this parameter as</p> <p>B9:B9 b.</p>
B8:B9	This will use the default of Both. It will replicate the formatting in cell b8 to b9 for every column and row of the input data.
Update Chart Ranges dynamically using input	Specify which Excel charts need to be updated dynamically based on the size of the new data. (e.g. Sheet2 3 - the 3rd chart on Sheet2 will be dynamically updated)
Page break at X number of Rows / Horizontal	Insert a horizontal page break every number of rows specified. (e.g. 35 - will insert a horizontal page break every 35th row)
Page break at X number of Columns / Vertical	Insert a vertical page break every number of columns specified. (e.g. 5 - will insert a vertical page break every 5th column)

Run Macro

Runs the named Excel macro. You may run one macro in this parameter. To run more than one macro, use the Advanced Commands parameter.

Repeat Template Sheet usage

This parameter is used in conjunction with the Rename Template Sheet parameter and whether the Capsule is being iterated or not.

The purpose of this parameter is to control how the Template sheet is used and whether it remains in the result or not.

The table below shows how the Template sheet is employed.

The interaction of the Repeat Template Sheet usage and Rename Template Sheet parameters and whether it is an iterating Capsule or not

Repeat Template Sheet usage?	Rename Template Sheet Parameter used?	Capsule Iterating?	Action
N	N	N	Data is inserted into the Template Sheet, Template Sheet remains in the result.
N	Y	N	Data is inserted into the Template Sheet and is then renamed to the name specified in the Rename TemplateSheet parameter.
Y	N	N	Data is inserted into the Template Sheet, Template Sheet remains in the result.
Y	Y	N	Data is inserted into the Template Sheet, this sheet is renamed to the name specified in the Rename TemplateSheet parameter and a new Template Sheet is created.
N	N	Y	Data is inserted into the Template Sheet, Template Sheet remains in the result.
N	Y	Y	The Template Sheet is renamed to the name specified in the Rename TemplateSheet parameter.
Y	N	Y	Data is inserted into the Template Sheet, Template Sheet remains in the result.
Y	Y	Y	Data is inserted into the Template Sheet, this sheet is renamed to the name specified in the Rename TemplateSheet parameter and a new Template Sheet is created. On last iteration the Template Sheet is removed.

Advanced Commands This parameter lets the user specify additional commands for advanced features. If using multiple commands, each command must be separated by a space. It is also for backward compatibility with the PC Tool Xlaunch (Meta5 version 4.1).

A user can copy the commands from their existing PC Tool Xlaunch icon to this parameter and the tool will operate as before. No other parameters are required when this option is used.

Remember, any additional commands specified in this parameter work in conjunction with the other parameters set in this form. Therefore make sure that any additional commands used with parameters set do not conflict with each other.

A list of these Advanced Commands is included in Appendix A of this Chapter as an aid to migrating Xlaunch applications (Meta5 4.1) to this new tool.

Common advanced command examples:

-rm for running macros.

Use a space to separate multiple macro commands such as: `-rm macro5 -rm macro6`

-a for automatically adjusting column widths

-pw for printing workbook

Creating a single sheet report

We will start with a simple example to get you going. As experienced users of Meta5 will know, the power of Meta5 is such that there is usually a number of different ways to engineer a result and there is no limit to what can be produced!

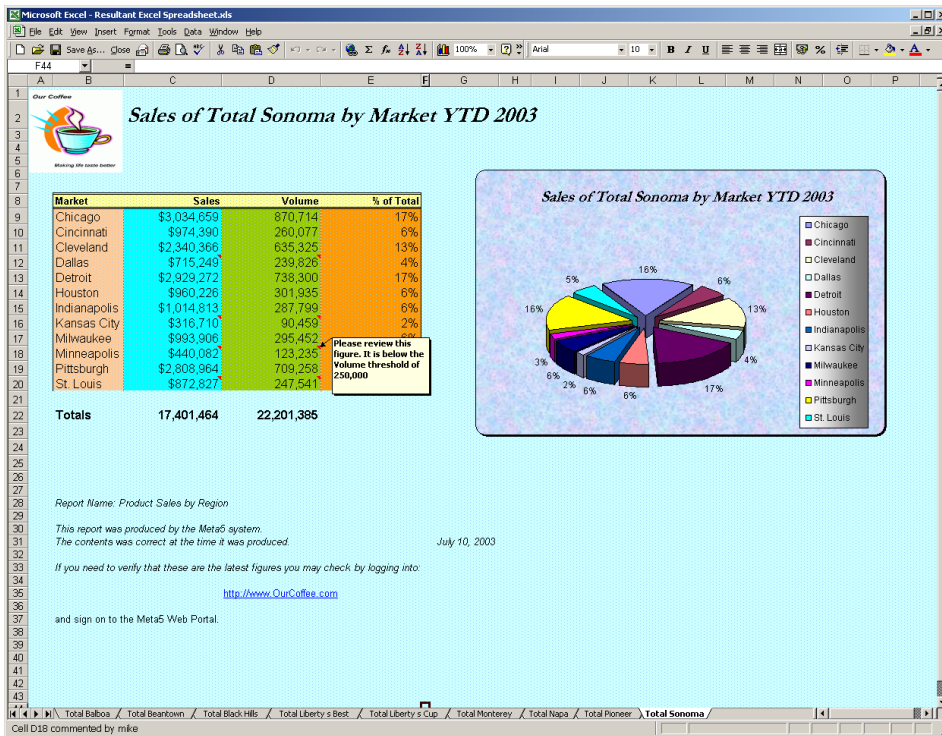


Figure 25. An example of a more complex multi-sheet workbook with variable length data, charts, calculated columns, totals, hyperlinks and comments

For an easier introduction to the iExcel Integrator we will look at a much simpler, single sheet result to get you started.

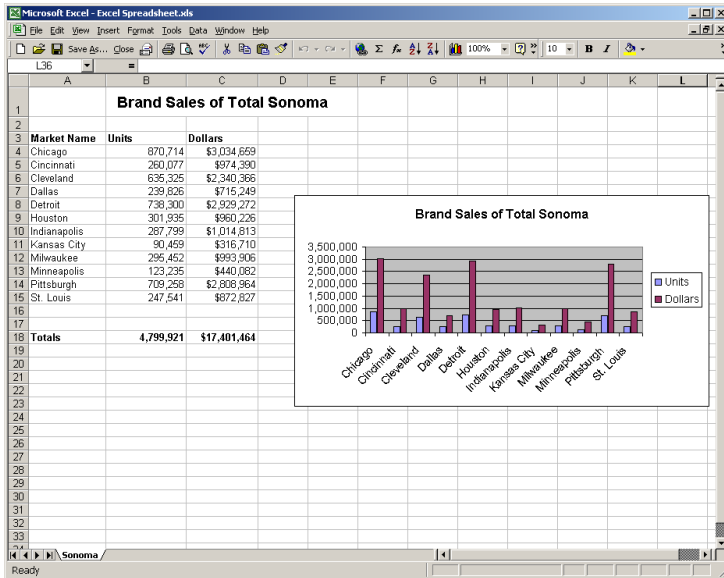


Figure 26. Excel Sheet "Sonoma" containing title, totals and a chart

Figure 26 demonstrates a simple Excel Report created using the iExcel Integrator.

To build this report we need to create a template Excel Spreadsheet to contain all of the formatting and charting required. As an aid to construction, run the data into the location you have chosen and format the cells, build the totals and generate the chart. It is important to remember that all formatting must be performed in this template. When formatting is completed, empty all of the data that will come in from other tools, such as titles and the data itself. Use the Clear Contents option instead of Delete when clearing out the input data. This will preserve the formatting and structure of your report whilst removing the data itself.

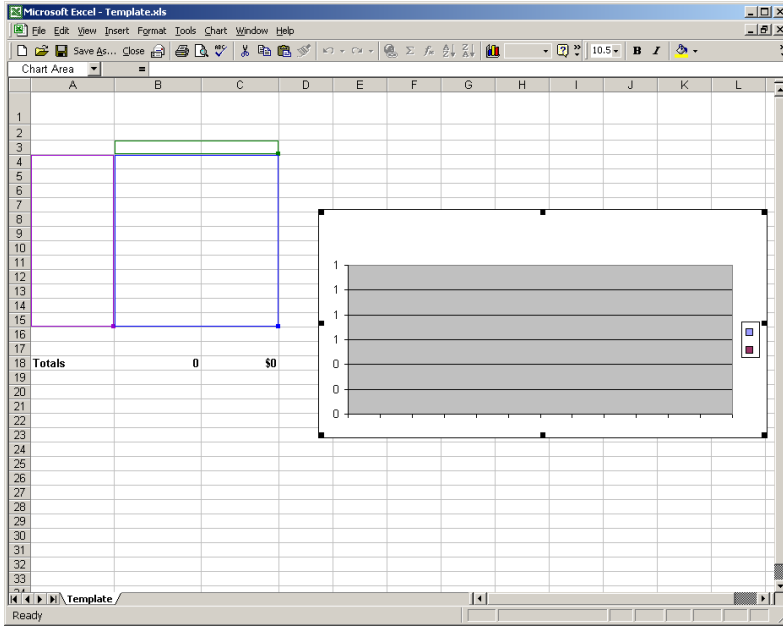


Figure 27. Contents of the Template Spreadsheet with the input data cleared

Figure 27 shows the contents of the Template Spreadsheet after the data has been cleared. Note all of the formatting is still in place but empty.

Our Capsule to produce the results in Figure 26 looks like the following.

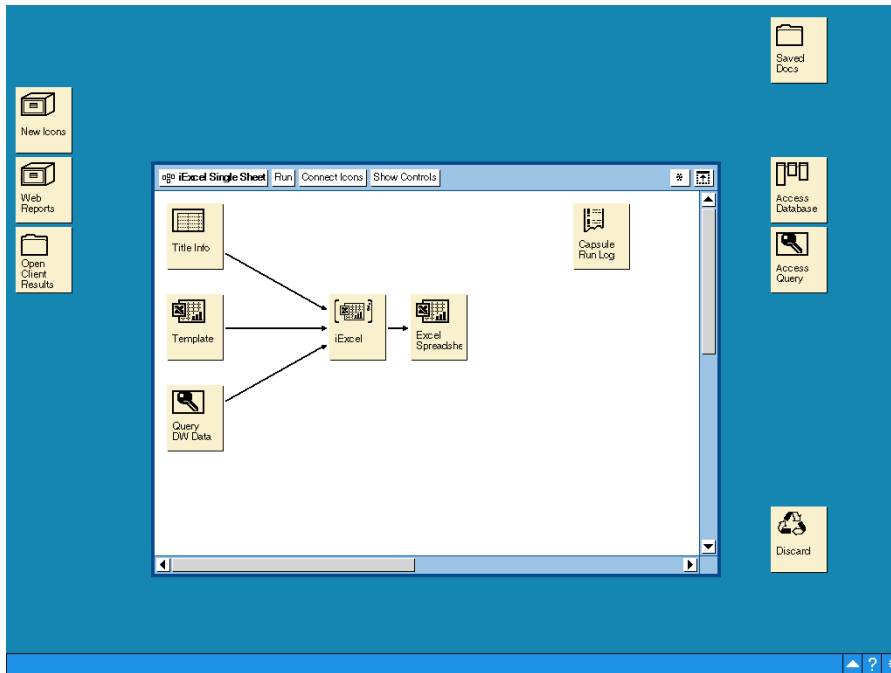


Figure 28. Sample Capsule using a Template Excel Spreadsheet

The Template Excel Spreadsheet is connected to the **Template** region of the iExcel Integrator icon.

The Title Meta5 Spreadsheet contains the title text in cell C1 and is connected to the **Paste Cells** region of the iExcel Integrator icon.

The Meta5 Query tool is configured to get data from the corporate data warehouse and passes this data to the iExcel Integrator icon through the **Data** region.

Note that it is required that the Data region is connected last to iExcel.

The iExcel Integrator is configured as follows:

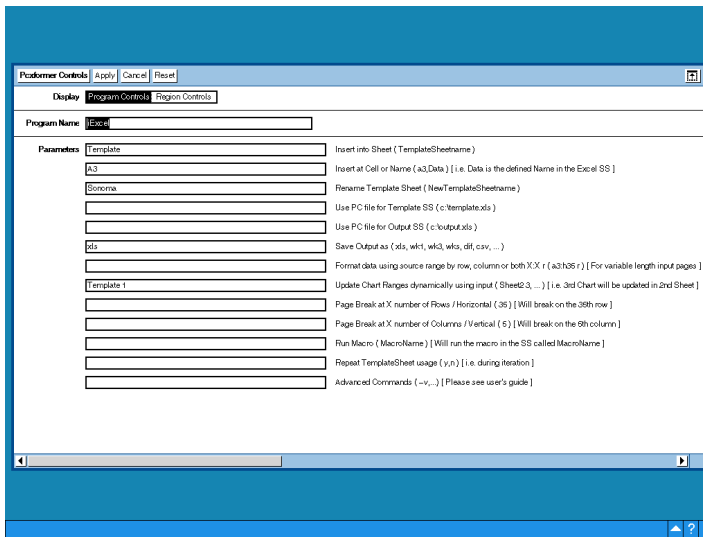


Figure 29. Parameters of the iExcel Integrator Program Controls window

Explanation of each of the parameters used in this example:

Insert into Sheet

We have renamed the default Sheet1 in the Template Excel Spreadsheet to **Template**. This sheet contains all of the formatting and charting. We therefore specify Template as the sheet to load the data into.

Insert at Cell or Name

We want the data to start in row 3, column a. We therefore specify A3 as the cell location of the start of the data. We could use an Excel Defined Name in place of this cell reference. Simply define the Name in the template sheet to cater for the size of data anticipated.

Rename Template Sheet.

The resultant sheet name is specified here to rename the Template sheet to one of our choice. In this example we are changing the sheet name to **Sonoma**.

Save Output as

We have chosen the default value of xls to create an Excel Workbook.

Update Chart Ranges dynamically using input

We need to update our chart in the event that the data is of a different size. Specify the name of the sheet followed by a space and then the number of the chart in that sheet. In our example this becomes **Template 1**.

You can have multiple charts to update. Simply separate them in a list using a comma (,).

Creating a Multiple Sheet Report

There are two ways of creating a multi-sheet report. The first is a static approach and will only produce the sheets you predefine in the Template. This approach is described under the Excel Spreadsheet section. It becomes tedious to build all of the sheets and their formatting so a way to re-use an existing template sheet was developed and put into the functionality of the iExcel Integrator tool.

The iExcel Integrator tool can dynamically create a workbook containing a variable number of sheets, each with their own name to aid in identification.

Creating a Multiple Sheet Report Dynamically

Let us take the sample displayed at the beginning of the iExcel section and repeated here.

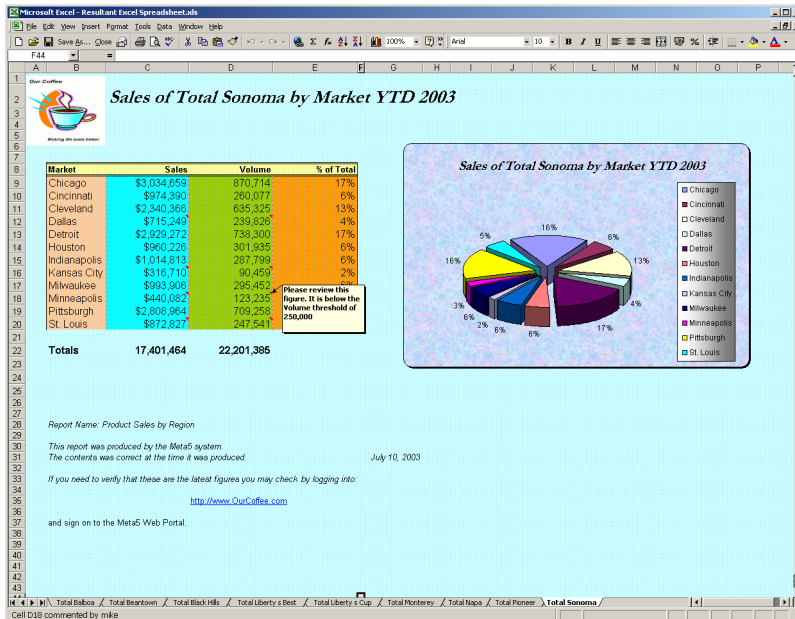


Figure 30. A multi-sheet workbook

This multi-sheet workbook contains individual sheets for each product the company sells. These sheets are named appropriately to identify which sheet has which product.

Each sheet has a table of input data, a chart related to this data, titles, calculated values and comments.

The Capsule that generated this Spreadsheet looks like this.

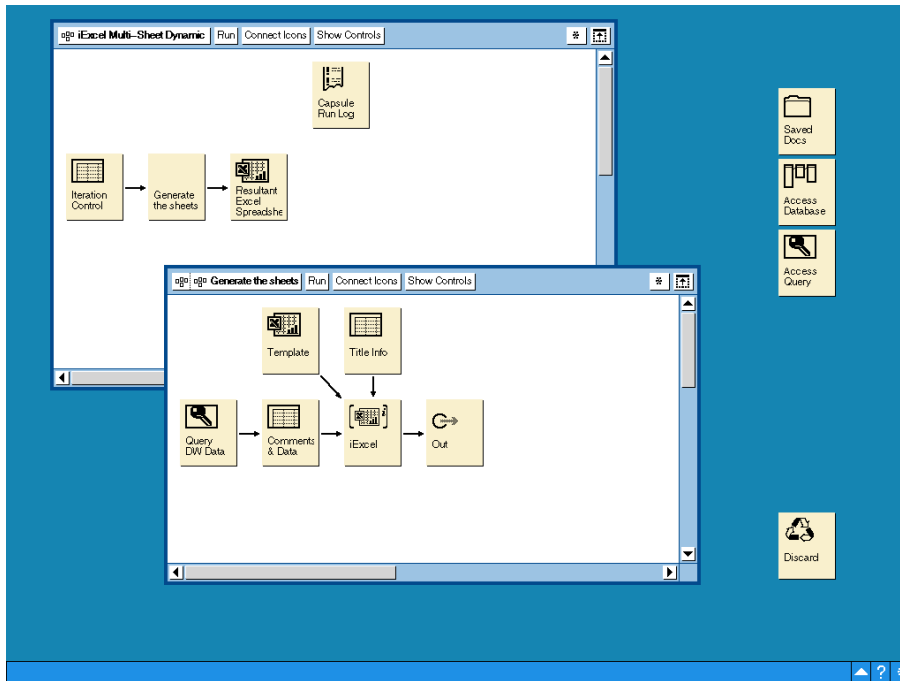


Figure 31. Iterating Capsule that produces the Multi-Sheet Workbook above

Values passed to the controls of the Iterating Capsule

The Iteration control is maintained within the spreadsheet preceding the inner Capsule. It contains the following information.

	A	B	C	D	E
1	Brand	Sheet Name	Sales Threshold	Units Threshold	
2	@A	@B	@D	@E	
3	Total Balboa	Total Balboa	300	300	
4	Total Beantown	Total Beantown	500,000	50,000	
5	Total Black Hills	Total Black Hills	50,000	20,000	
6	Total Liberty*s Best	Total Liberty s Best	1,000,000	300,000	
7	Total Liberty*s Cup	Total Liberty s Cup	100,000	30,000	
8	Total Monterey	Total Monterey	1,000,000	300,000	
9	Total Napa	Total Napa	100,000	30,000	
10	Total Pioneer	Total Pioneer	1,500,000	500,000	
11	Total Sonoma	Total Sonoma	900,000	250,000	
12					
13					

Figure 32. A region called controls containing the Brands defined to @A, the Sheet name defined to @B and threshold values to be used to generate a comment or not, passed in @-variables @D and @E

Note the use of a region in this control spreadsheet. The region is called controls in this example and has this region name specified as the source region on the arrow connecting it to the Capsule. This allows the user to specify meaningful headings above the region to describe the @-variables used without including them, and incurring an error, in the controls that are passed to the Capsule.

Template Sheets

Looking inside the inner Capsule, **Generate the Sheets**, we next choose the Template Excel Spreadsheet.

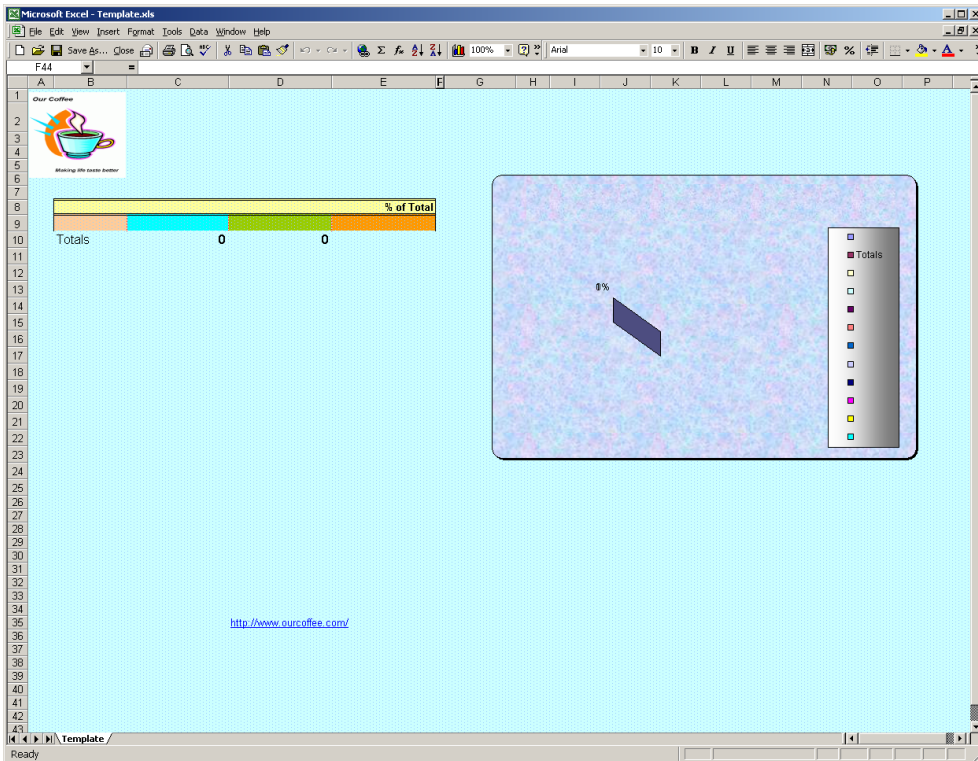


Figure 33. Template sheet with all formatting included and the data cleared

A significant amount of formatting has taken place with the use of colours and backgrounds. This is an individual choice or it will likely need to comply with your corporate standards.

The columns of input data have been formatted with different colours. We need only define a single row of colours, as we will be able to extend the formatting to include all the incoming data using the iExcel parameters.

Calculated values

The **% of Total** column is calculated from the incoming data. The normal calculation would be: $\text{this_rows_sales}/\text{sum_of_all_sales} * 100$

Since we do not know where the end of the data will be, we need to include a formula in this column from the start row (C9) through to the maximum row anticipated (C23). We do not want to calculate a value where the row is not populated by the input. Therefore, if we are catering for a maximum of 15 input rows and we want to suppress the output where no input exists then a formula in cell E10 could look like the following.

```
=IF(B10="Totals","",IF(LEN(B10)>0,IF(C10=0,"",C10/(SUM($C$9:$C$23)/2)), ""))
```

The outer IF statement is determining if this is the Totals row, in which case we do not want to display the calculation (since it will always be 100%).

The next IF statement determines whether to calculate a result or not based on the content of cell B10 (Market). If Market is empty then do not calculate.

The innermost IF statement traps a zero sales value (C10). If the Sales are 0 then ignore the calculation.

The division by 2 in the calculation is because we will be including the totals row in our summation which will have the effect of doubling the total.

We can enter this formula once in cell E10 and extend it down to cell E23.

Calculating Totals

In our example we want the Totals line to appear two lines below the last input row.

A formula of the sort

```
=IF(B9="", IF(COUNTIF ($B$8:B9,"Totals")>0, "", "Totals"), "")
```

can be entered into cell B10 and extended down to B23 as per the calculations of % of Total.

This formula looks one row behind itself to determine if the row is populated or not. If it is occupied then output nothing and continue. If the preceding row is blank we need to determine if the Totals line has already been calculated. The COUNTIF function will check the range of start of input data through to the row above this formula for the presence of the word "Totals". If it finds it then output nothing since the Totals have already been produced.

Important Note: *We must use a cell reference instead of a Defined Name since a Defined Name will expand or contract to accommodate the size of the input data. It will not overwrite occupied cells. Using a cell reference will mean that formatting is maintained but data will be overwritten by the rows of input data.*

The input data will overwrite the formulae up until the last row of data input. (See the Important Note above).

Therefore the first formula left intact will determine that the row above is occupied and skip the Totals calculation. The next row will determine that the row above is empty and check to see if Totals already exist. It doesn't so we achieve the result we needed; a single blank row between the last input row and the totals line.

Calculating the total values is achieved as follows:

```
=IF(B10="Totals",SUM($C$9:C9),"")
```


We use an absolute cell reference to locate the start of the sum and then use the cell reference above our calculation as the end of the sum. The sum is only calculated if this is the Totals row.

Simply enter this formula into cell C10 and extend down to cell C23. The absolute reference will stay fixed but the end of the sum range will increase for each successive row ensuring we include every input cell in our calculation.

Repeat the same procedure for the summing of the Volume column (D).

Hyperlinks

If you wish to include a hyperlink in your resultant spreadsheet then you need to define the link in your template sheet.

There is a limitation in Excel that restricts what Meta5 can do to this hyperlink. There are two portions to the Excel Hyperlink. The first is the text to display in the cell that contains the hyperlink. This information will be replaced by incoming data (see the next section referring to Pasting Cells). The other portion is the URL that the hyperlink will invoke on selecting this cell. This cannot be affected by incoming data.

Therefore the URL associated with the hyperlink, as defined in the Template sheet, will remain static. However the text to be displayed in that cell can be changed.

Pasting Cells

The Meta5 Spreadsheet icon called Title Info contains text that will be overlaid in the template sheet. Each cell with data in it, in the source file will be copied to the same cell in the template sheet and then placed in the result.

We are effectively pasting the contents of the Spreadsheet into the result.

The content of this Spreadsheet follows:

File	Replicate From	Recalculate	Sort Controls	Copy/Move Controls	Clear Data	Show Page Break	Print
	A	B	C	D	E	F	G
1							
2							
3							
4							
5							
6							
7							
8		Market	Sales	Volume			
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							
31							
32							
33							
34							
35							
36							
37							
38							
39							

Figure 34. Contents of the spreadsheet that will be "pasted" into the result

We connect this Meta5 Spreadsheet to the iExcel Integrator and specify the destination region name as **Paste Cells**.

Paste Comments

The source data from the corporate data warehouse is passed, in our example, into a Meta5 Spreadsheet region called Data. The purpose for this is to perform analysis of the incoming data to determine which cells should contain a comment.

The criterion for this check is whether the data in the cell is greater than the threshold value passed to the Capsule in @-variables @D and @E.

Another region has been set up in the Meta5 Spreadsheet that mirrors the positions of the cells in the template sheet. Like the Paste Cells process, each cell containing data in the source will result in that data being displayed as a comment in the corresponding cell of the resultant Excel Spreadsheet.

	A	B	C	D	E	F	G	H	I
1									
2	Column	Thresholds	@-Variable						
3	Volume	=250,000	@E						
4	Sales	=800,000	@D						
5									
6									
7									
8		Market Name	Dollars	Units					
9		Chicago	3,034,659	870,714		=	=		
10		Cincinnati	974,390	280,077		=	=		
11		Cleveland	2,340,366	636,326		=	=		
12		Dallas	718,249	239,828		=Please review this	=Please review this	figure. It is below the V	
13		Detroit	2,929,272	736,300		=	=		
14		Houston	960,226	301,935		=	=		
15		Indianapolis	1,014,813	287,799		=	=		
16		Kansas City	316,710	90,459		=Please review this	=Please review this	figure. It is below the V	
17		Milwaukee	993,906	295,452		=	=		
18		Minneapolis	440,082	123,236		=Please review this	=Please review this	figure. It is below the V	
19		Pittsburgh	2,808,964	709,256		=	=		
20		St. Louis	872,827	247,541		=Please review this	=Please review this	figure. It is below the V	
21									
22									

Figure 35. Spreadsheet containing the source data in region Data (left hand side) and calculations to produce the comments in another region called Comments (right hand side)

We can therefore pass the data into the Spreadsheet, region Data, the Comments region containing the formulae will then be calculated producing the comments where we need them. We then pass the Comments region into the iExcel Integrator, Paste Comments.

The calculations are based on the @-variables @D and @E which contain threshold values for comparison. If the cell contains a value that is less than the threshold, it displays a text message in the appropriate cell of the Comments region. For example, to check a Sales value is above its threshold (@D) we can use the following formula:

```
=if (B12="", "", if(C12<@D, concat("Please review this figure. It is below the Sales threshold of ", @D * 1), ""))
```

In brief, cell B12 (an input cell) is checked for a value; if not present then nothing is displayed. If B12 is occupied then compare the value in it to the threshold value (@D for Sales column and @E for Volume column). If it is less then display the concatenated text string. This string becomes the comment for the appropriate cell in the result.

Different products have different thresholds and therefore individual values are passed via the iteration control Spreadsheet (see Figure 31.) to this Spreadsheet.

Remember to declare as many formulae as you anticipate the maximum number of input rows to be. In our example this is determined as 12 rows.

We connect this Spreadsheet to the iExcel Integrator in order to pass both the Comments region and the Data region. We direct the Comment region to the Paste Comments region of the iExcel Integrator and the Data region to the Data region of the iExcel Integrator.

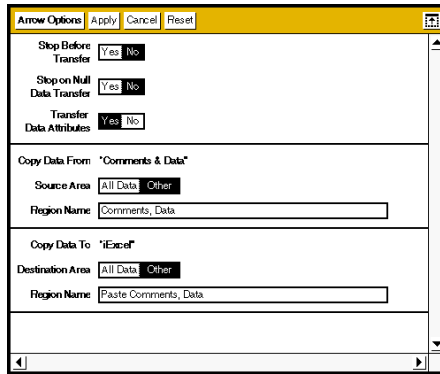


Figure 36. Example of multiple source regions to destination regions. Comments go to Paste Comments and Data goes to Data. The order is important!

iExcel Controls

Lastly the iExcel Program Controls need to be configured to achieve the result we want.

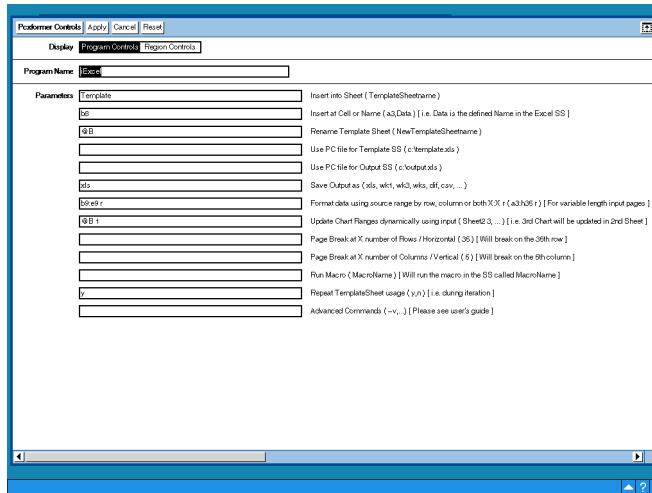


Figure 37. Program Controls of the iExcel Integrator, set to reuse the template sheet during iteration and rename the template sheet using an @-variable

Explanation of each of the parameters used in this example:

Insert into Sheet

Our template Excel Spreadsheet contains a single sheet called Template. This is the sheet that contains the template formatting, calculations and structure. The Template sheet for the iExcel Integrator to use must therefore be set to this name: **Template**.

Insert at Cell or Name

We want the data to start in row 8, column B. We therefore specify B8 as the cell location of the start of the data. Note we have not used a Defined Name here. This is because the area to be populated by the incoming data contains formula to calculate the Totals and we want these formulae to be overwritten by the incoming data leaving the remaining formulae intact to be executed. Therefore we get the Totals row appended to the data with a blank row between them.

Rename Template Sheet.

The resultant sheet name is specified here to rename the Template sheet to one of our choice. We use @B in this parameter so that we can build the new sheets using the name in @B that is passed through during each iteration.

Therefore in the first iteration the data will be loaded into the sheet called Template. It will be renamed to Total Balboa (the first value for @B in the iteration control Spreadsheet).

Since we have also specified that we want to **Repeat TemplateSheet usage** (see later in this section for more detail) the Template sheet will be copied to the new sheet name leaving the Template sheet in its original form ready to accept the data input in the next iteration.

The subsequent iterations will create the remainder of the sheets in the same way as above with one exception. Since the Capsule is **iterating**, the **Rename Template Sheet** parameter has a value and **Repeat TemplateSheet usage** is **Y**, the Template sheet is removed from the result on completion of the final iteration. (See the table in Figure 25)

Save Output as

We have chosen the default value of xls to create an Excel Workbook.

Format data using source range X:X

We want the formatting of the data columns to be extended to the end of the incoming data. The Template sheet has the first row of the input data location defined with the correct font, size, format and colour. We therefore want this formatting to be extended for every row of input. We achieve this by specifying B9:E9 r which is the same as B9:E9 ROWS. Therefore for every row of input the formatting in cells B9 to E9 will be replicated down through columns B to E.

Update Chart Ranges dynamically using input

We need to update our chart in the event that the data is of a different size. Specify the name of the sheet followed by a space and then the number of the chart in that sheet. In our example this becomes **@B 1**.

Repeat TemplateSheet usage

We specify **Y** to make the Template sheet available for second and subsequent iterations. Note the combination of the Capsule **iterating**, the **Rename Template Sheet** parameter specified and **Repeat TemplateSheet usage** set to **Y**, will cause the Template sheet to be copied to the sheet specified in the Rename Template Sheet parameter and for the Template sheet to be removed on the final iteration. (see above for an explanation of the Rename Template Sheet parameter).

Caution: Some formatting within Excel can cause the size of the Excel Spreadsheet to grow very large. An example of this is when background graphics are used, such as watermarks. It can cause the resultant file size to become very large, especially when multiple sheets are being generated. Meta5 recommends running this on a Capsule Server since the large size may cause the execution of the Capsule to take some time to complete. This is a matter outside of Meta5's control as it lies within Microsoft's Excel development.

Appendix A. Advanced Commands for iExcel

Command Line Parameters Quick Reference Chart

(note that these commands are also utilized by the Meta5 version 4.1 XLaunch tool)

Argument / Function

-a	Autoformat width and height of inserted rows and columns to fit data
-ac	Adjust comment box size to fit data
-cf	Specify comment font for all comments
-di	Disable interactive mode
-dt	Delete the template file when finished
-ff	Saves your output file in the specified spreadsheet format
-fr	Paste the format codes in the specified range throughout the output file
-hb	Insert a horizontal page break every number of rows specified
-ic	Insert user specified text into specified cell range (may be repeated)
-ir	Specify an insert range or named region in the output file
-kd	Keep the data file when finished
-ps	Print the specified worksheet when finished
-pw	Print the entire workbook when finished
-rm	Runs the specified macro after data transfer.
-sr	Specify a source range or named region in the data file
-uc	Specified sheet name contains numbered chart to be updated with inserted data
-v	Make the running Excel object visible

-vb	Insert a vertical page break every number of columns specified
--	Null argument (for visually separating arguments on the command line)

Advanced Command Line Parameters Examples

-a

Auto format

This command formats the width and height of inserted rows to fit the inserted data. To use, simply type -a on the command line. This usually will not be necessary when using a template file because the template file will contain the height and width you most probably want.

-ac

Auto format comment

This command will automatically adjust the size of comment boxes to fit their data.

-cf

Comment font specified for all comments.

For example: -cf Arial 12

-di

Disable interactive mode

Suppresses pop-up error messages when XLaunch is run on a local desktop. Has no effect when XLaunch is run on a capsule server. The capsule server only runs in disabled interactive mode.

-dt

Delete Template

This parameter deletes the template file when XLaunch finishes running. Be sure you do not wish to keep your template file before using this parameter. This feature is used when inserting data into multiple Excel Sheets and using the output from a prior run as a template for the next set of data to be inserted into the next Sheet.

-ff

File Format

Saves your output file in the specified spreadsheet format. Format choices include: CSV, DIF, WKS, WK1, WK3 or XLS.

For example: -ff WK1

-fr

Format Region

Paste the format codes in the specified range throughout the inserted data. For example, to apply the formatting characteristics of the first page of your Excel template to the entire set of data and your first page was B3 to E40, type:

-fr B3:E40

-hb

Horizontal Break

Insert a horizontal page break every number of rows specified. For example, to force Excel to print a page break after every 42nd row you would type: -hb 42. This feature is commonly used with the -fr feature described above.

-ic

Insert Text

Insert user specified text into the specified region in the output file. For example, to insert the text "Sales Report" in the cell D1, you would type: -ic D1 "Sales Report"

To insert the same text in every cell between D1 and D9, you would type: -ic D1:D9 "Sales Report" This feature is commonly used to insert @variable information in various cells in Excel. For example: -ic D1 "@A" where the value of @A is Sales Report.

Note: It is necessary to enclose inserted text within double quotes if the text contains spaces or dashes.

-ir

Insert Range

Specify where you want your inserted data to appear in the output file. To have your data start at cell B3, you would type: -ir B3.

To have your data inserted into the cell range A4:F16 you would type: -ir A4:F16

Note: If you specify a single cell, your inserted data will start at that cell and grow as large as necessary. If you specify a range of cells, your inserted data will be truncated to fit within that range. You may also insert data into an Excel Defined Name, however you must pre-create the Defined Name in your Excel template file. This works much in the same way as regions do in Meta5 except that the

region will not grow and shrink as it does in Meta5. The data will only grow to the edge of the Defined Name.

For example, if you would like to insert your data into a Defined Name called "reportbody" contained in Sheet3 of the template, you would type:

-ir Sheet3!"reportbody"

-kd

Keep Data File

Keep the data file when finished. By default, XLaunch deletes the xdata\$\$\$ dif file when it is finished. If you wish to keep this file, simply include this parameter on the command line. This feature is used mostly for debugging purposes. Also note that the \$\$\$ token will be replaced by DSK on the desktop and E00, E01... on the capsule server.

-ps

Print Worksheet

Print the specified worksheet to the default printer when finished. To use, simply type: -ps <sheet name> where <sheet name> is the name of the sheet you wish to print. Remember that this command will print the Worksheet on the default printer of where the capsule is being run. If it is run on the Capsule server then it will print on the default printer of the Capsule server, not your workstation.

-pw

Print Workbook

Print the entire workbook to the default printer when finished. To use, simply type: -pw on the command line.

-rm

Run Macro

Runs the specified macro after data transfer has completed but before printing and saving take place. You can run multiple macros by repeating the function on the command line. For example: -rm Macro1 -rm Macro2.

-sr

Source Range

Specify a source range or named region in the data file. For example, if you only want cells A3 to E16 from Meta5 to be inserted into your output file, you would type: -sr A3:E16

-uc

Update Chart

Specify a sheet name that contains a numbered chart to be updated with inserted data. This parameter can be used by itself or with one or two optional parameters.

If used by itself, XLaunch will update the first chart on the same sheet on which you just inserted data. For example, if you are inserting data on Sheet2 and wish to update graph number 1 on Sheet2 to use the newly inserted data, you would simply type: -uc

If used with one parameter, XLaunch will update the first chart on the sheet you specify in that parameter. For example, if you are inserting data on Sheet2 and wish to update graph number 1 on Sheet1 to use the newly inserted data, you would type: -uc Sheet1

If used with two parameters, XLaunch will update the chart you specify on the sheet you specify. The first parameter will be the sheet name and the second parameter will be the chart number. For example, if you wish to update graph number 3 on Sheet2 to use the newly inserted data, you would type: -uc Sheet2 3

-v

Visible

Makes the running Excel object visible. You will be able to watch XLaunch copy the data from the data file into your output spreadsheet during a capsule run.

This parameter should not be used in capsules being sent to a capsule server. To use, simply type -v on the command line.

-vb

Vertical Break

Insert a vertical page break after every number of columns specified. For example, to force Excel to print a page break after every 5th column you would type: -vb 5

--

Null Argument

This argument acts as a buffer between XLaunch features i.e. - commands and Excel files on the command line. Some of the XLaunch features have optional parameters, such as the -uc feature. If the -uc feature is the last feature on the command line before file names are specified, the filenames may be interpreted as -uc parameters. To avoid this, the null argument can be used to clearly separate features from template and output files. This will only occur when features with optional parameters are specified on the command line immediately preceding filenames. Therefore it is recommended that -uc not be used as the last parameter on the command line. However if -uc is the last feature, then use the null argument -- before specifying template and output files. For example if the command line -ir a3 -uc C:\temp.xls C:\out.xls is specified, then the two filenames will be interpreted as -uc parameters. A better syntax would be:

```
-ir a3 -uc -- C:\temp.xls C:\out.xls
```

OR

```
-uc -ir a3 C:\temp.xls C:\out.xls
```

Chapter 2. Microsoft Word Integration

Overview

Meta5 has developed an intelligent link with Microsoft's Word product. Two new icons are introduced with Meta5 version 4.2 to support Word integration: The Word Document and the iWord Integrator



Figure 38. Word Document icon



Figure 39. iWord Integrator icon

The uses and differences between these two tools will become clearer, later in this chapter.

Word Document Icon

The Meta5 Word Document icon can be regarded as a Microsoft Word Document residing on the Meta5 File Server. The internal Word data is held on the Meta5 File Server and passed to Word when the user double clicks on the Word Document icon.

This allows the user to employ the power of the Meta5 environment to automate the generation of sophisticated reports and distribution over the corporate Email system of choice. The entire process can be scheduled to execute at a frequency that best fits the business or run as required by the simple click of the mouse.

The Word Document tool is located in the * Blank Icons file drawer within New Icons.

What can the Word Document do?

You can use the Word Document tool to:

- Automate the generation of sophisticated reports.
- Include charts and graphics in the Report.
- Generate mass mailing of letters to customers.

Note: All workstations and Capsule Servers that need to run the Meta5 Word Document tool must have the software for Microsoft Word loaded on it and be fully licensed to use it. No license is granted or implied by Meta5 when using the Meta5 Word Document tool.

The Word Document icon is accessible from the Meta5 desktop and removes the limitation of the previous Meta5 version 4.1 WordTool product, of only having this document accessible from the PC's local hard drive or mapped network drive.

Examples of Use

- Accessing a Word Document
- Exporting a Word Document icon
- Producing Formatted Word Documents
- Multiple inputs to a Word Document
- Passing data from a Text tool into a Word Document
- Passing data from Meta5 Spreadsheet into Word Document
- Passing an Excel Chart into a Word Document
- Creating Multiple Word Document reports

Importing a Word Document

To make use of an existing Word Document within a Meta5 Capsule, simply drag and drop the required Microsoft Word icon from Windows Explorer or from the Windows desktop into a Meta5 Capsule. This will result in a Meta5 Word Document icon, which is a copy of the Microsoft Word Document, being placed in the Capsule. From this point onwards, the Meta5 Word Document is stored on the Meta5 File Server and is not linked in any way to the original document.

Alternatively, you may wish to automate your Capsule such that it dynamically picks up the contents of a Word Document and passes it to the next icon for further processing.

Placing a PC Directory tool in front of the Word Document icon and connecting them together can achieve this. Configure the PC Directory tool to access the directory where your external data resides, and set the source region, on the

arrow connecting them, to the name of the Word Document you want. See Figure 40.

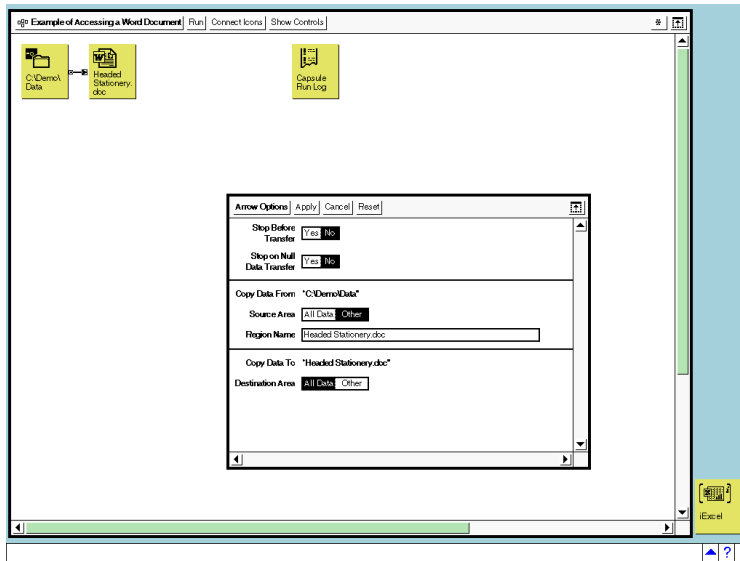


Figure 40. Accessing a Word Document via the PC Directory tool

Figure 41, below, is the result of opening the Word Document icon in Meta5. It is exactly the same as opening Word from your Windows desktop. This example is a template letter that has a bookmark defined to accept an Excel chart that will be loaded in later. (The bookmark is displayed as the vertical bar with smaller horizontal bars top and bottom and is located in the center of the page.)

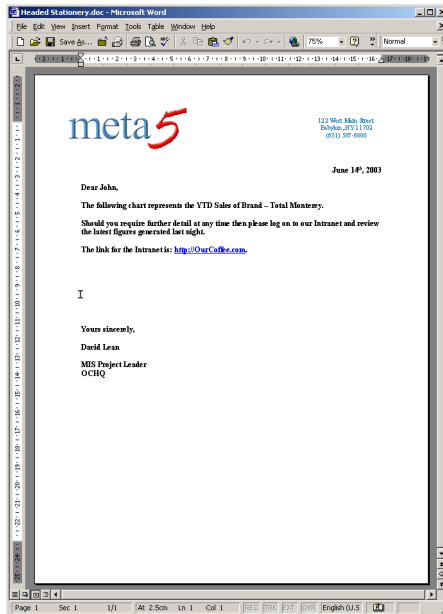


Figure 41. An open Meta5 Word Document icon

Exporting a Word Document icon

As mentioned earlier, the Meta5 Word Document icon resides on the Meta5 File Server. This icon can be distributed out to the recipients in a number of ways. It can be included as an attachment in an email and mailed out over the corporate email system, it can be made available over the Web through a standard browser or it can be exported out to the PC's local or mapped network drive.

If the document is required to be processed further outside of Meta5, then direct the Word Document through a PC Directory tool to your PC's hard drive (or networked drive).

Configure a PC Directory tool to address the directory where you wish to store the Word Document. Connect the Word Document to the PC Directory icon and configure the arrow connecting them to specify the resultant document file name in the destination region. See Figure 42.

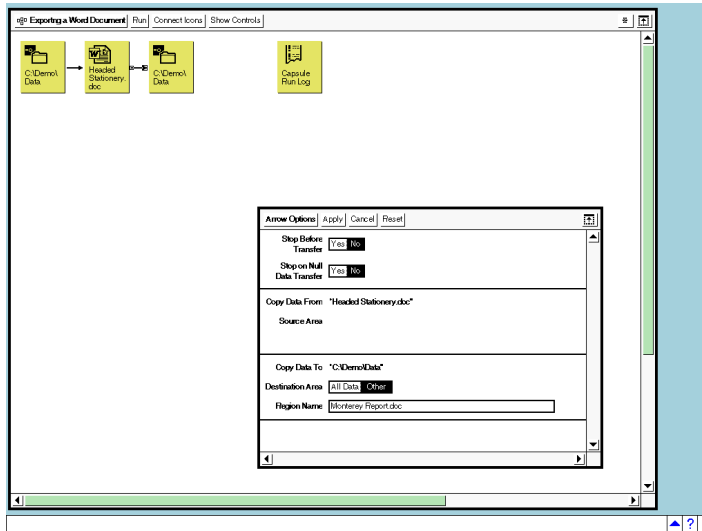


Figure 42. Placing a Meta5 Word Document on to the PC's hard drive or mapped network drive

Producing Formatted Word Documents

We have seen how easy it is to transfer data into Meta5 from Microsoft Word. The result of this transfer depends on the receiving icon. If it is another Word Document then the result is a copy of the original document. If the receiving icon is a Meta5 Text tool then the result is a raw dump of the Word data. All formatting of the Word Document that is being transferred in, is ignored.

Similarly, a Word Document that is created by Meta5 will have no formatting by default. The following section describes how to format your resultant Word Document for onward delivery to a recipient or for processing by another application.

Formatting of the Word Document is a one-time operation. Once the designer has completed the layout and formatting of a resultant Word Document it is ready to be used in a Capsule. The Capsule can be run any number of times without the need to revisit the formatting.

Regions - Bookmarks

As with the Meta5 Spreadsheet, Text and other tools, the Meta5 Word Document understands the concept of regions as well.

The equivalent concept in the Word Document is the Bookmark. A bookmark can be set up in the Word Document by first selecting the position for the bookmark and then, by going to the Insert menu, select Bookmark.

You are presented with the following window that allows you to give a name to the location you have selected. Simply type in a name such as "Body" and then click on Add. The special character denoting a bookmark is placed where you selected.



Figure 43. Creating a Bookmark in a Word Document

Note: On some implementations of Word at customer sites, the displaying of bookmarks has been turned off by default. In this case no special character will be displayed. To turn on the display of bookmarks, select Options... from the Tools menu and check the Show Bookmarks box.

Once an area is named in this way, you can address this area directly by using the name of the Bookmark in the source and/or destination regions of the arrows going in to/out of the Word Document.

By using the regions like this, Meta5 will preserve any formatting contained within the Bookmark and apply this formatting to any incoming data.

The Bookmark (Region) will expand to accommodate the data being transferred into it.

Any data, including its format, which falls outside the Bookmark, will be maintained in the result. Headings, narrative and comments may be placed in this way.

Using @-Variables with the Word Document Icon

Meta5 @-variables can be used with the Word Document tool. They only have relevance to the Arrows coming into and out of a Word Document, but it can prove to be a very useful facility.

You might consider using @-variables when choosing which location in the resultant document to populate. In other words, an @-variable may be set to different Bookmarks dependent on the logic within a Capsule.

Multiple inputs to a Word Document

The Word Document icon can accept input from a large number of Meta5 icons. These icons include:

Tool	Typically Used For:
Text	Transferring in: Headings, narrative, comments and instructions.
Spreadsheet	Transferring in: Data, headings, narrative, comments and instructions.
Excel Spreadsheet	Transferring in: Data, headings, narrative, comments, instructions and graphics.
Query	Transferring in: Data
Brio Query	Transferring in: Data, headings, narrative, comments and instructions.
Business Objects Query	Transferring in: Data, headings, narrative, comments and instructions.
Reporter	Transferring in: Data
SQL Entry	Transferring in: Data
Word Document	Transferring in: Data, headings, narrative, comments and instructions.
Transformers	Transferring in: Data, headings

We can therefore envisage the inputs to various parts of a Word Document as coming from a number of different sources. The headings may come from a Word Document whereas the data may come from the Data Warehouse. A title may be constructed from @-variable settings, for example, the Product being reported on, and therefore a Spreadsheet is the most likely source. A chart might be incorporated to give the report a pleasing and meaningful look. Lastly, instructions or comments may be included from another Text tool.

The following Capsule demonstrates this concept.

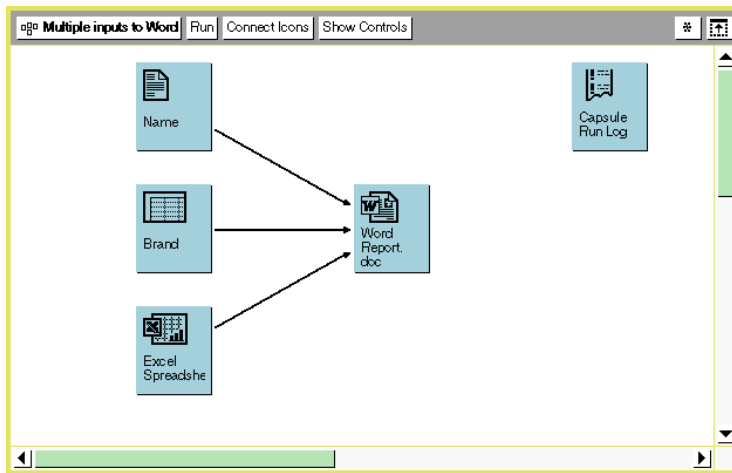


Figure 44. Example of multiple inputs to the Word Document icon

Let us first take a look at the resultant document. This document, as can be seen in Figure 45, contains the letter body with three bookmarks to locate the Name, Brand and Chart.

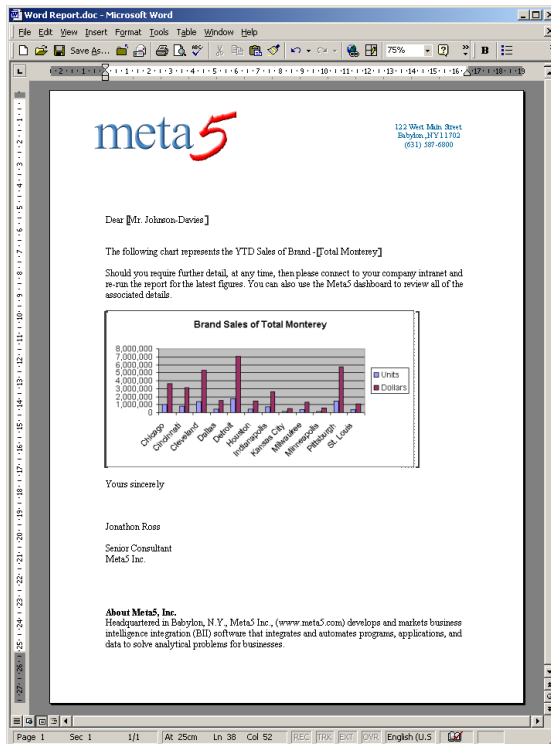


Figure 45. Resultant document from the example above

Passing data from a Text tool into a Word Document

The name for the greeting part of the letter has been derived from a Text tool, which simply contains the name.

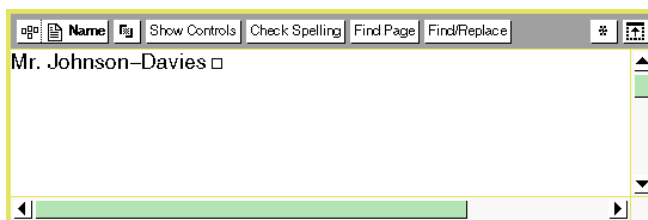


Figure 46. Text tool containing the name to use in the greeting part of the letter

By specifying the name of the bookmark on the destination area region name of the arrow connecting the Text tool to the Word Document, we transfer the contents of the Text tool (name) to the bookmark called Name in the Word Document.

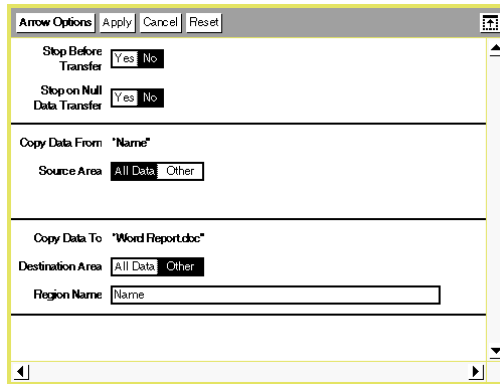


Figure 47. Specifying the location for the greeting text

Passing data from a Spreadsheet into a Word Document

Single Cell

The letter, in our example, refers to the YTD Brand Sales of Total Monterey. The brand name was derived from the Meta5 spreadsheet called Brand. Inside this spreadsheet we have declared the brand name as Total Monterey, see Figure 48.

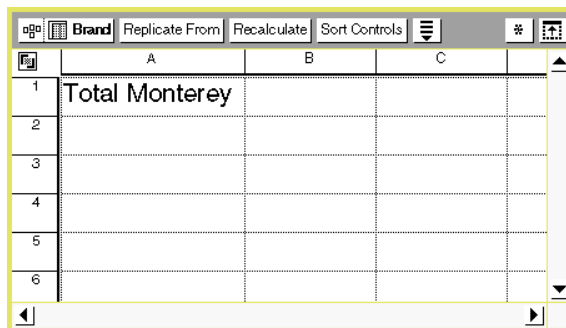


Figure 48. Brand name placed in cell A1 of the input spreadsheet

We direct this spreadsheet into a bookmark in the resultant Word Document by declaring the bookmark name in the destination area region name of the arrow connecting the Spreadsheet tool to the Word Document.

Multiple Cells

We can pass more than one cell of data into a bookmark. This is useful when building a formatted report containing rows and columns of data. Each cell in the source will occupy the next Tab position in the target Word Document. Each row

will start a new line. Therefore, by defining appropriate tab stops, we can produce a formalised report.

To produce the output displayed in Figure 49, we need to define a bookmark to contain the tabular data and format the bookmark with appropriate tabs using the Format menu and selecting Tabs.

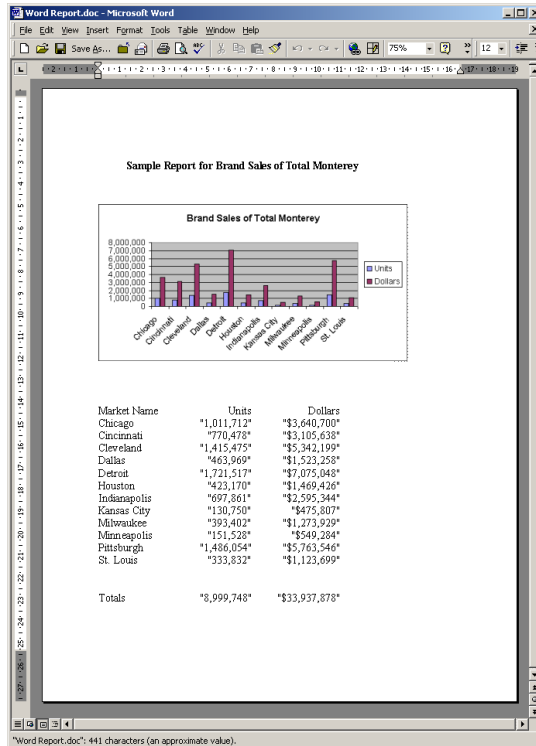


Figure 49. Sample report using Tabs to format the incoming data

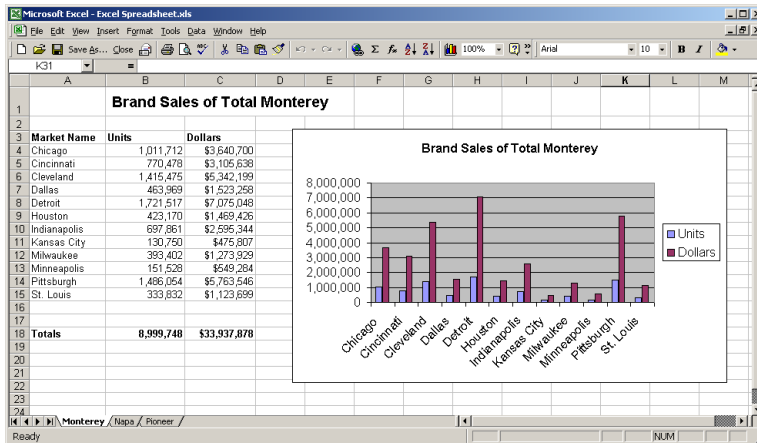


Figure 50. Excel Spreadsheet from which our sample Word report is derived

Figure 51, below, shows the Arrow Options where we are populating the Word Document with a Chart, Title, and Data from our Excel Spreadsheet.

Note the use of multiple regions on the source and destination areas. There must be an equal number of entries on the source and destination areas and each entry must be separated with a comma (,). The first source entry will go to the first destination entry, the second to the second and so forth.

Therefore, we are placing Chart 1 from the Excel Spreadsheet into the bookmark called Chart in the Word Document. We are placing the contents of cell C1 (title) of the Excel Spreadsheet into the bookmark Title in the Word Document. Lastly we are placing the data in cell range A3:C20 into bookmark Data. We have preformatted the Data bookmark with right aligned tab stops in positions 7cm and 11cm to accommodate the incoming data.

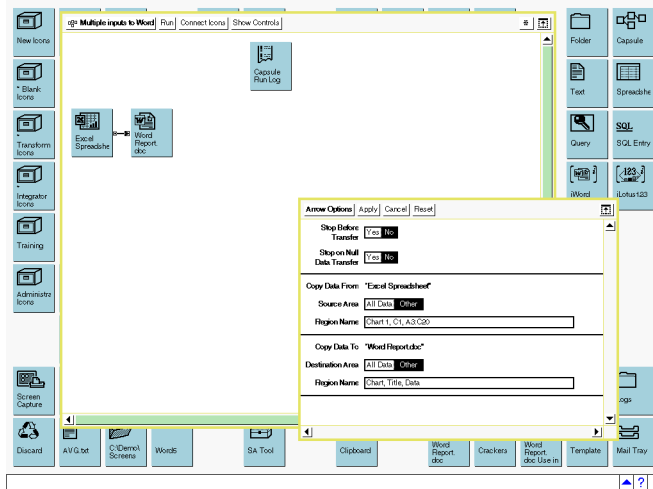


Figure 51. Arrow Options showing the transfer of a Chart, Title and Data

Passing an Excel Chart into a Word Document

We have seen, from our Multiple Cells example earlier, that an Excel Chart can be transferred into a Word Document.

The Excel Chart can be placed in a Word Document to enhance the appearance of a report.

Two things are needed to complete this action. First, we need to know the name of the chart so that we can specify it in the Source Region Name of the arrow connecting the source Excel Spreadsheet to the Word Document

Excel Chart Naming

Excel names its charts in the format "Chart x", where x is a number starting from 1 that denotes each chart in the workbook. The first chart in an Excel Workbook is named Chart 1, the second Chart 2 and so on. They are numbered sequentially as they appear in the Workbook starting from the top left cell in Sheet1 and working its way left to right and top to bottom of each sheet in turn.

Secondly, we need to define a destination in the Word Document to contain this chart. We create a bookmark inside the Word Document at the location we would like the Excel chart to appear. The bookmark can be formatted but since it is a graphic we are restricted in what we can do to it. One formatting characteristic that may prove useful is the ability to center or right justify the bookmark. Generally though, any formatting you apply will be maintained and used by Meta5.

Lastly connect the two icons together and provide the name of the Chart as the Source Region Name and the bookmark name as the Destination Region Name in the Arrow options. Figure 52 demonstrates this procedure.

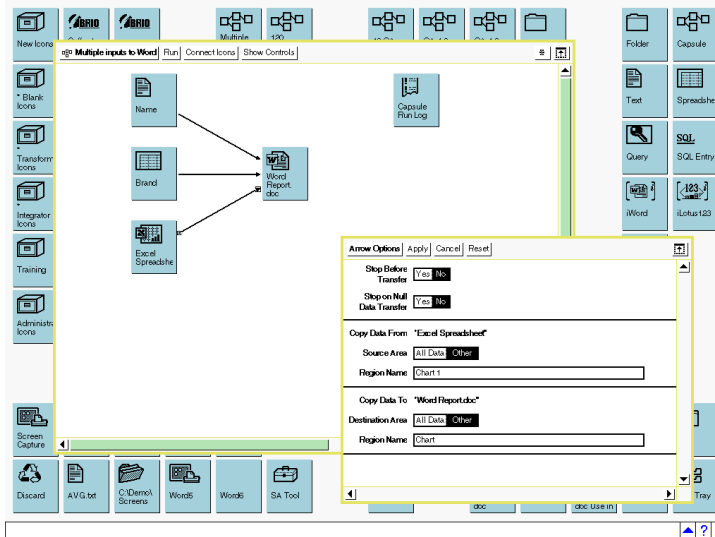


Figure 52. Directing an Excel Chart into a Word Document

Iterating the Word Document

The Meta5 Iteration capability is a very powerful mechanism to allow Capsules to run multiple times with a single click of the mouse. Iteration comprises a Capsule to be run multiple times and a controlling icon to pass the values to be used for each iteration of the Capsule. These values are passed through the Meta5 @-variables that are defined in the Capsule controls.

We can use iteration to produce multiple Word Documents, for example a mailing out.

Creating Multiple Word Document reports

Let us take the example used earlier in this chapter to demonstrate the iteration capability of the Word Document.

The first step with any iterating Capsule is to make the Capsule dependent on @-variable assignments rather than constants. For example, assign @-variables to constraints in a Query or Reporter, Headings and Titles, Naming of the resultant Word Document and recipients of Email. All of these concepts are dealt with in detail in the Meta5 Capsule Users Guide.

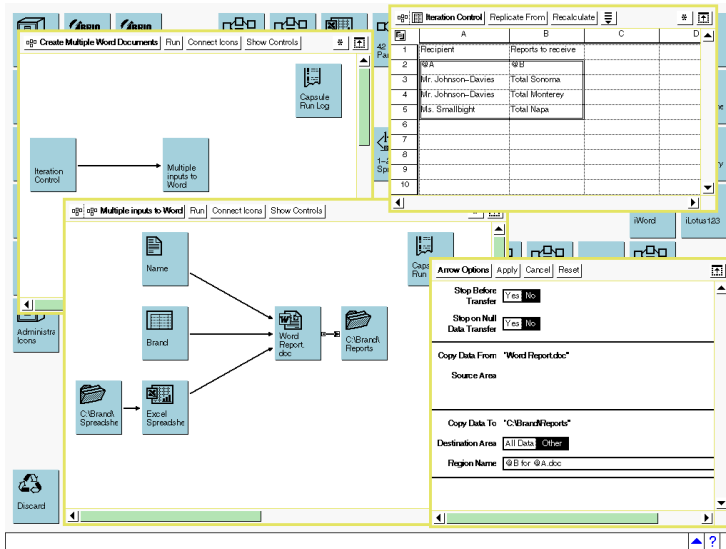


Figure 53. Iterating the Word Document to produce multiple reports

In Figure 53 we can see that the iteration control Spreadsheet is passing the brands: Total Sonoma, Total Monterey and Total Napa in @-variable @B, and the recipient of the report in @A to the inner Capsule. This inner Capsule will be executed three times, once for each pair of values in @A and @B. The resultant report is exported to the local C: drive of the PC and is given a name to reflect the report being processed, namely "@B for @A.doc".

On completion of the Capsule, three documents will reside on C:\ and will be named:

Total Sonoma for Mr. Johnson-Davies.doc,

Total Monterey for Mr. Johnson-Davies.doc and

Total Napa for Ms. Smallbright.doc.

The incoming Excel Spreadsheets, containing the charts we wish to place in our reports, reside in C:\Brand\Spreadsheets. We direct a PC Directory tool to this directory and connect it to the Excel Spreadsheet in our inner Capsule. We specify the source region as the name of the Spreadsheet we wish to access. In this example the name of the file reflects the brand name so we can use @B.xls as the source region name.

We substitute the fixed value for the Name in our Text tool with a region called @A and the Title in our Spreadsheet as a formula =@B.

Figure 48 shows the definition of the Name formula @A and figure 49 shows the definition of the Brand formula =@B.

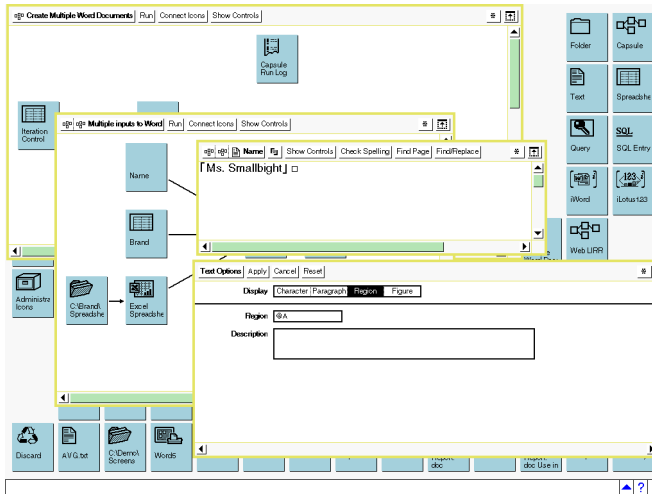


Figure 54. @A used as the Region Name in the source Text tool

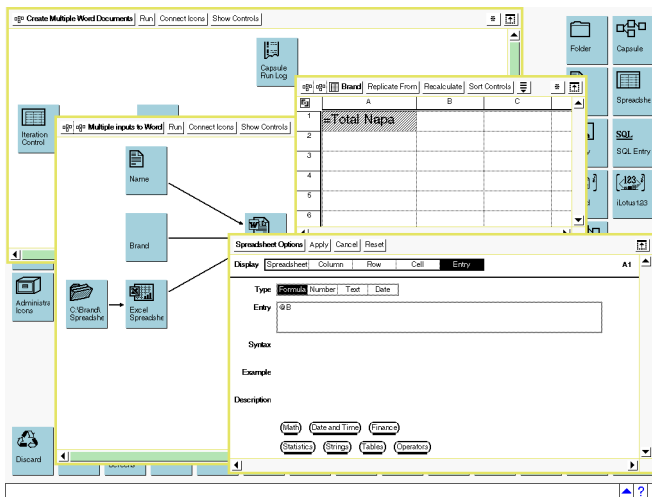


Figure 55. @B used as the Formula in the source Spreadsheet tool

iWord Integrator Icon

The iWord Integrator icon was introduced to provide the Meta5 user with more power and control over the creation of Word Documents. We have seen with the Meta5 Word Document icon that a great deal can be achieved using this tool. We will now review the capabilities of the iWord Integrator and how they can be used.

The Integrator family of tools have a similar look and feel to the Meta5 Transformers. The Integrator icons have input and output regions and a controls panel that allows you to configure the tool.

Examples of use

- Incorporate Office objects in Word documents
- Produce a variety of output formats
- Produce large volumes of data for publishing to the Web
- Automate the creation of a large number of letters

The iWord Integrator Regions

Figure 56 shows the window displayed when you open the iWord Integrator icon. Like the Meta5 Transformer, this window contains the input and output regions and where appropriate, can display the contents of each region in the display area. Unlike the Meta5 Transformer, the Integrator icons have no Run button in the window header.

There are five input regions and one output region.

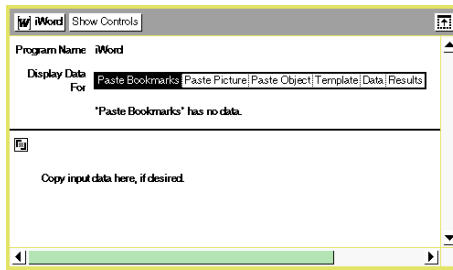


Figure 56. Input and output regions of the iWord Integrator

These regions have the following functions:

Paste Bookmarks

A control spreadsheet containing a list of bookmarks in column A and the text that is to occupy this bookmark in column B. Row 1 is reserved for comments and any column beyond B is ignored. If the value in column A is preceded with the # symbol then this row is ignored.

Paste Picture

A graphic that is to be included in the result. This could be a chart in Excel or a graphic file from your PC such as a .BMP.

Paste Object

An Excel Workbook Sheet that is to be included in the resultant Word Document

NOTE: If you accidentally paste a picture file as an object, you will receive an image that contains errors (example .bmp)

Template

A Word Document icon containing all the formatting and bookmarks to create the end result.

Data

The input data that can be sourced from a Query, Reporter, SQL Entry, Excel Spreadsheet, Meta5 Spreadsheet, Meta5 Text, Brio Query, Business Objects Query, Windows Application and numerous Meta5 Transformers.

This is a mandatory region to be filled in last. All other regions precede this in terms of the order of execution. It is only when this region is populated, will the iWord Integrator execute and produce the result.

Results

The final output from the iWord Integrator after all of the Input Regions and Program Control parameters have been processed.

The iWord Integrator loads up each of the inputs and processes them in a work area.

It uses the Template document as the base and loads each of the inputs over the top of it. The completed document is then moved into the Results region.

Parameters of the iWord Integrator

Click on Show Controls to review the parameters of the iWord Integrator (see Figure 57).

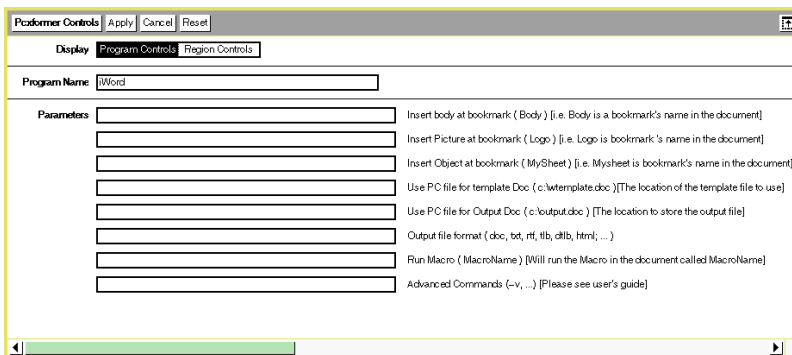


Figure 57. The parameter controls of the iWord integrator

Program Name.	iWord. This is the name of the program to run. This should not be changed.
Insert Body at bookmark	Specify the name of the Bookmark in the Templatefile to contain the Data region of iWord.
Insert Picture at bookmark	Declare the bookmark where the Paste Picture region will be located.
Insert Object at bookmark	Declare the bookmark where the Paste Object region will be located. I.e. An Excel Spreadsheet
Use PC file for template Doc	An alternative to supplying a Word Document as input to the Template region. Specify the full path and file name of the Word Document that is to be used as the Template. Note: The Template specified in this parameter will take priority over any Template connected to the Template region.
Use PC file for Output Doc	An alternative to supplying a Word Document as output to the Results region. Specify the full path and file name of the Word Document that is to be used to contain the results. Note: If you use this parameter to specify a result document, you can also use the output from the Integrator. Both will be produced.
Output file format	Use this parameter to create the output in a different format to the default doc type. Other file types you can specify here are: txt, rtf, tlb, dtlb, html.

Run Macro	Runs the named Word macro.
Advanced Commands	<p>This parameter lets the user specify additional commands. It is also for backward compatibility with the Meta5 version 4.1 PC Tool called WordTool.</p> <p>A user can copy the commands from their PC Tool WordTool icon to this parameter and the tool will operate as before. No other parameters are required when this option is used.</p> <p>Remember, any additional commands specified in this parameter work in conjunction with the other parameters set in this form. Therefore make sure that any additional commands used and parameters set, do not conflict.</p> <p>A list of these Advanced Commands is included in Appendix A of this Chapter.</p> <p>Examples of common advanced commands:</p> <p>-pd Automatically prints final Word document on the default printer for the PC or server running the Capsule.</p> <p>-rm Runs macro</p>

Examples of Use

We will take the following report and analyse each portion of it so you understand each of the parameters and how to use them.

We are producing a three-page letter containing a greeting, an explanation of the contents, a chart from Excel, a tabular report of the data that produced the chart (on the second page) and finally a copy of the Excel Spreadsheet that produced the chart (on the last page).

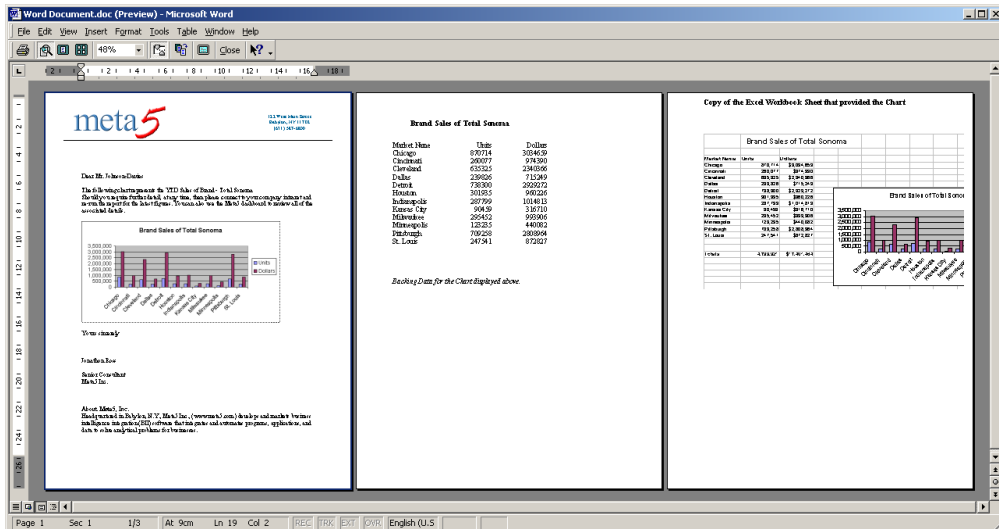


Figure 58. The three-page result to demonstrate iWord integrator capabilities

The Capsule that produced this report follows:

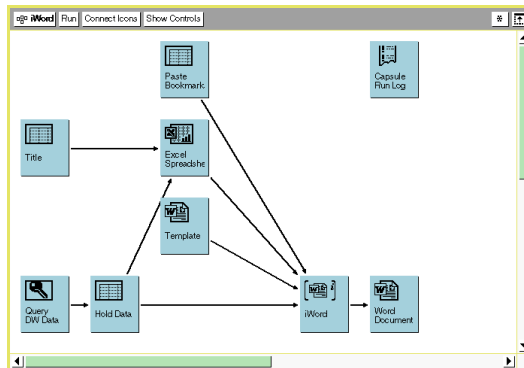


Figure 59. Capsule that utilizes the iWord integrator to produce the result above

Let's take each component in order. Firstly, the Spreadsheet labeled Paste Bookmark. This Meta5 Spreadsheet contains a table of bookmark names and the text that should occupy each bookmark.

Paste Bookmarks

	A	B	C	D	E	F
1	Bookmark name	Text				
2	Brand	Total Sonoma	Brand name used in the body of the letter			
3	Name	Mr. Johnson-Davies	Name to be used in the greeting portion of the letter			
4	Title	Brand Sales of Total Sonoma	Title to be used in the letter			
5	# End of values					
6						
7						

Figure 60. Example of a Paste Bookmark spreadsheet

iWord treats this input in a similar way to the Rules region of the Conditional Clean transformer, as follows:

The first line is ignored and can be used to identify the column contents.

- Column A contains the name of the bookmark to paste text into.
- Column B contains the text to insert into the corresponding bookmark (in column A).
- Columns beyond and including column C are ignored and can be used for comments.
- Any value entered into column A that has a # sign as the first character will cause the entire line to be ignored. Comments can be placed anywhere within the spreadsheet in this way.

Paste Picture

Next, we will take a look at the Excel Spreadsheet inside this Capsule. We are using this Spreadsheet for two purposes. The first is to generate the chart that we want to include on the first page of our report.

This Excel chart is generated by populating the data cells with the results of our Query tool that obtains the latest figures for our chosen Brand and a title that is used in the chart that comes from the Spreadsheet called Title. The chart is automatically updated to reflect this data. We can then pass the chart into the iWord integrator region, Paste Picture.

We are using a holding spreadsheet out of the Query tool so that we can pass the data to the Excel Spreadsheet and to the iWord integrator without executing the Query twice.

Including PC graphic files

An alternative to using an Excel chart would be to include a graphic file from your PC (examples: .bmp or .gif). Simply direct a Meta5 Explorer icon at the directory that contains the graphic and connect this to the iWord integrator icon. Option the

arrow and specify the name of the graphic file to include as the source region name and specify Paste Picture as the destination region name.

Paste Object

The second purpose of this Excel Spreadsheet is to provide a copy of the entire Spreadsheet as page three of our report. This Spreadsheet will be passed into the iWord integrator region, Paste Object.

The controls on our arrow that joins the Excel Spreadsheet to the iWord integrator need to pass the chart (Chart 1) into the Paste Picture region as well as passing the data (A3:C18) into the Paste Object region.

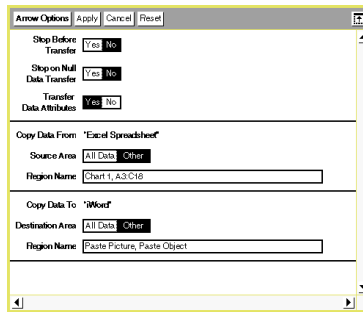


Figure 61. Multiple regions to pass a chart and entire Excel Spreadsheet into iWord

Template

The Template Word Document, as the name implies, is the template to be used in producing the final results. All bookmarks and formatting should be applied here to affect the results of iWord.

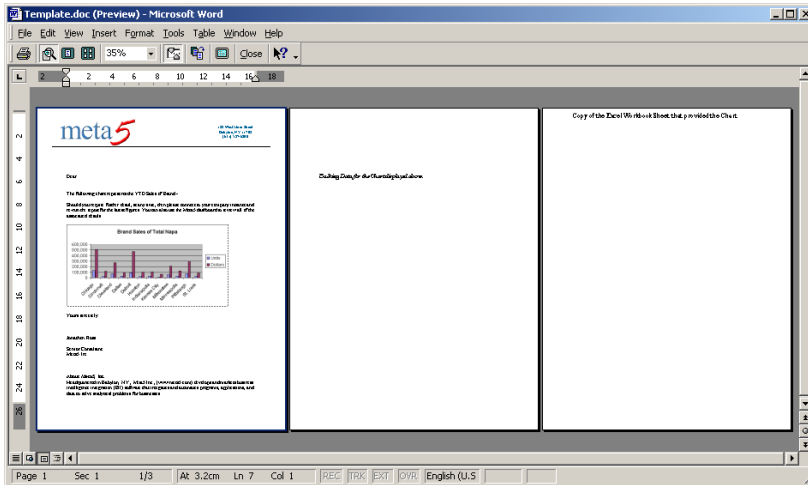


Figure 62. Example of a Word Template document. All formatting and bookmarks have been defined but the data removed

To assist in the production of the Template Document, first define the bookmarks in the approximate place you want them and run the data into them. Format the template to suit the data and presentation style.

When formatting a bookmark, ensure you have selected the bookmark as well as the data within it to ensure the formatting stays in place.

Data

The data is coming from our Query tool, via a holding spreadsheet, into the iWord region, Data. This bookmark is defined in the Template document and has been formatted with right aligned tab stops in position 7cm and 11cm.

The iWord integrator knows when to commence processing the inputs on receipt of the data into this region. It acts as the "trigger" to cause iWord to execute. Keep in mind that when you set up all of your connections to iWord that you ensure the Data input runs last.

Explanation of the iWord Integrator parameters used in this example.

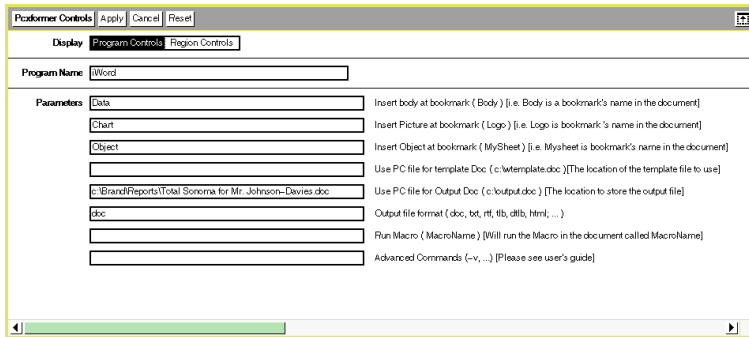


Figure 63. Parameters used in our example

Insert Body at bookmark

We specify here the bookmark name where the Data region of the iWord input region should go. In our example we have created a bookmark called Data in our template document and have formatted it with the appropriate Tab stops and font size.

Insert Picture at bookmark

We specify here the bookmark name to contain the contents of the Paste Picture input region of the iWord integrator. In our example we have created a bookmark called Chart in the template document.

Insert Object at bookmark

We specify here the bookmark name to contain the contents of the Paste Object input region of the iWord integrator. In our example we have created a bookmark called Object in the template document.

Use PC file for Output Doc

We can optionally specify here whether we want the iWord integrator to produce the results as a PC file. The full path and filename need to be specified. Note that both the Results region and this PC file will be populated with the results.

Output file format

We are using the default format of doc. This will produce an Microsoft Word Document of type .doc.

Click on the Run button to execute the Capsule and produce the results.

Alternative Output formats

One interesting alternative to producing a Word Document result is to produce a Web page in HTML format.

By specifying html as the Output file format and declaring an .html file as the entry for the "Use PC file for Output Doc" parameter, the iWord integrator will generate pages that are suitable for publishing to the Web.

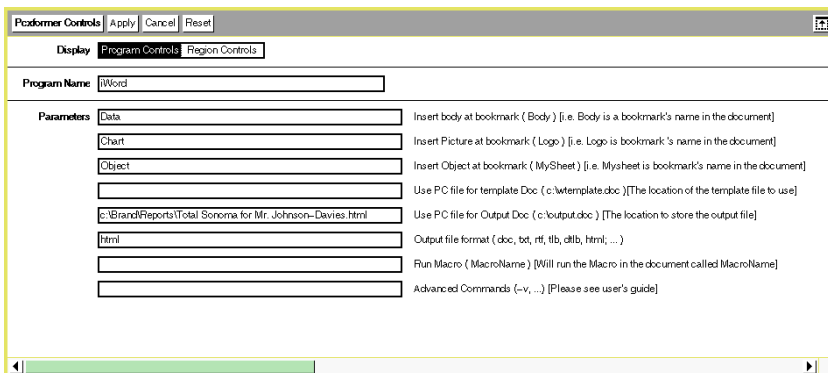


Figure 64. Generating a Web page from the iWord Integrator

Our example would produce the following:

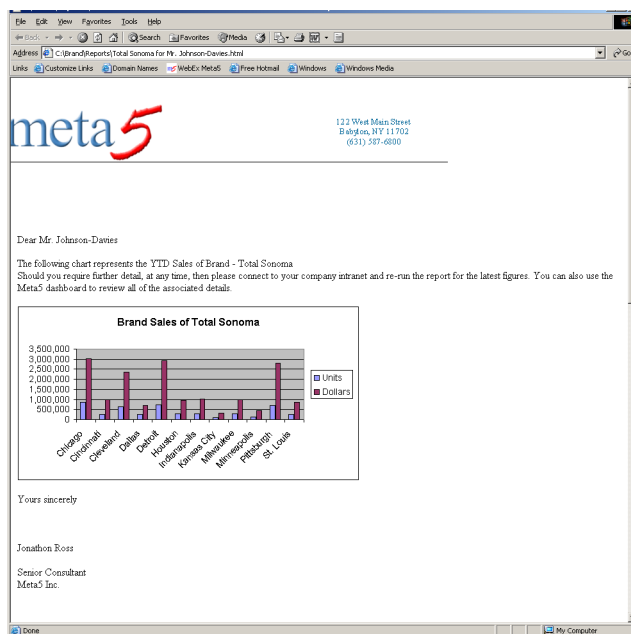


Figure 65. HTML version of the example letter

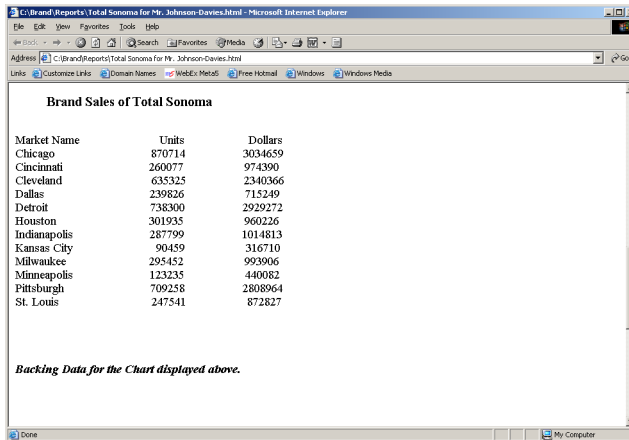


Figure 66. HTML version of the example letter

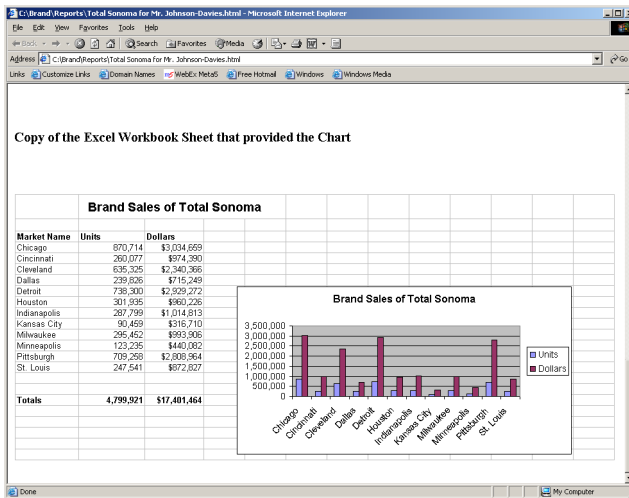


Figure 67. HTML version of the example letter

Iterating the iWord Integrator

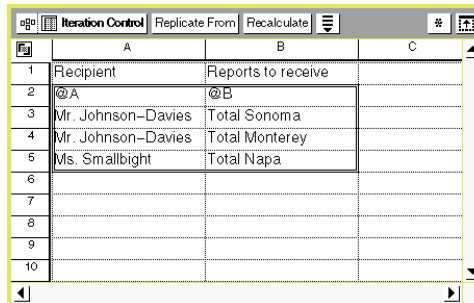
The Capsule we have been using earlier can be iterated to produce multiple reports for distribution to numerous recipients.

To do this, we need to convert the Capsule to make use of @-variables so that we can control its execution by the @-variables we pass to it.

Modifying the Capsule to use @-Variables

First, decide on the @-variables you want to employ. In this example we will use @A to hold the recipients name and @B to hold the Brand name to be reported on.

We can build the iteration control spreadsheet, place it in a new Capsule and copy our example Capsule into the new Capsule and to the right of the iteration control spreadsheet.



	A	B	C
1	Recipient	Reports to receive	
2	@A	@B	
3	Mr. Johnson-Davies	Total Sonoma	
4	Mr. Johnson-Davies	Total Monterey	
5	Ms. Smallbight	Total Napa	
6			
7			
8			
9			
10			

Figure 68. Iteration Control spreadsheet

We then need to amend the contents of the inner Capsule to make use of these @-variables.

Amend the Paste Bookmark spreadsheet so that it employs @B for the value to be passed into the bookmark called Brand. Also, amend the contents to use @A as the text to be passed into bookmark - Name. Lastly, amend the text to be placed into bookmark - Title to become a concat function that combines the text "Brand Sales of " with the @-variable @B. I.e. =concat("Brand Sales of ", @B).

Amend the Title spreadsheet so that it too uses a concat function to build the title from @B.

Amend the Query tool to constrain on the Brand @B.

Lastly, amend the iWord Integrator Controls so that it generates a PC output file employing the @-variables @A and @B in the name to aid in identification later.

I.e. C:\Brand\Reports\@B for @A.doc.

Three reports will be produced.

Connect the iteration control spreadsheet to the inner Capsule and declare the destination areas region name to be controls. Note we have defined the iteration data as residing in a region called controls. This is to allow us to specify some heading information above the @-variable assignments as an aid to understanding the application.

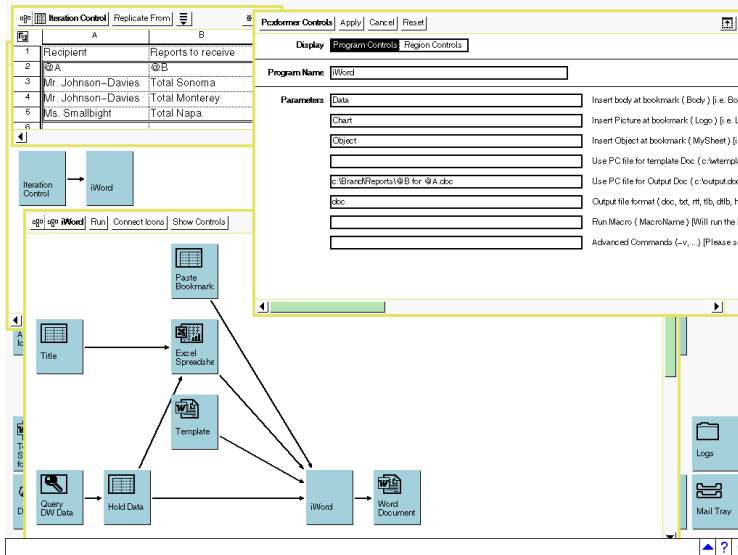


Figure 69. Iterating Capsule to produce multiple documents

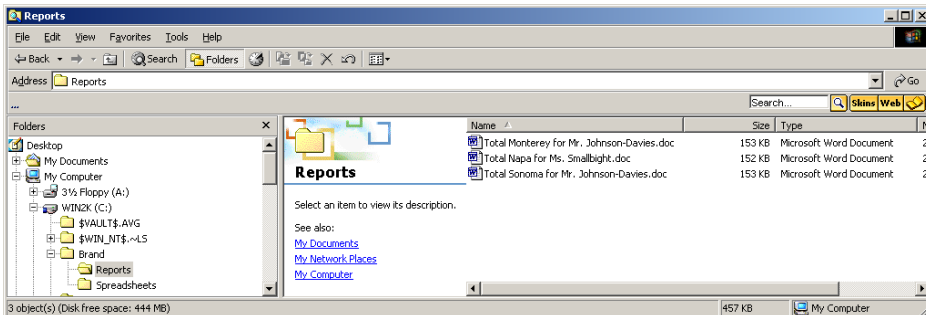


Figure 70. Windows Explorer view of the resultant three documents

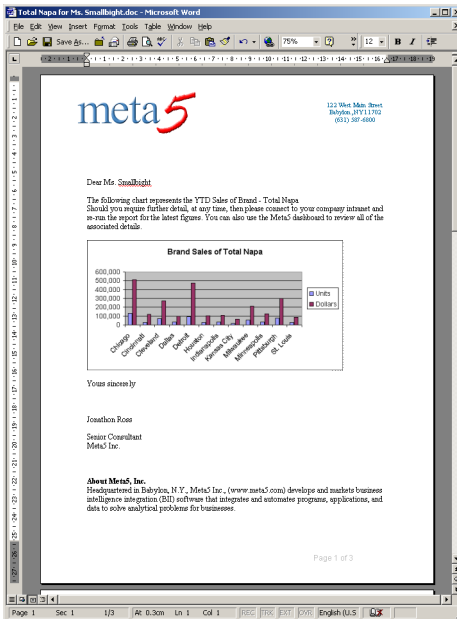


Figure 71. Page 1 of the resultant document

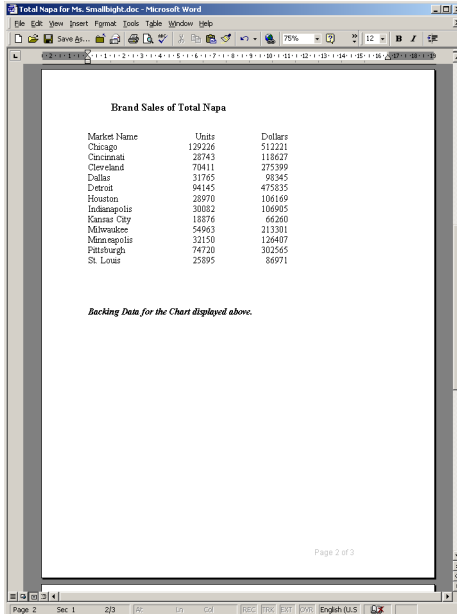


Figure 72. Page 2 of the resultant document

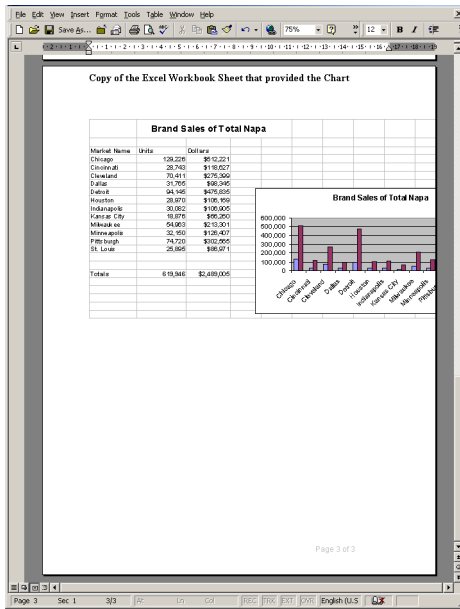


Figure 73. Page 3 of the resultant document

Appendix B. Advanced Commands for iWord

Command Line Parameters Quick Reference Chart

(Note that these are commands that are also utilized and supported by the Meta5 version 4.1 Wordtool)

Argument	Function
-di	Disable interactive mode
-dt	Delete the template file when finished
-e	Allows you to specify a file name to send error messages to
-ib	Insert the body text at the specified bookmark
-io	Insert an object file at the specified bookmark (may be repeated)
-ip	Insert a picture file at the specified bookmark (may be repeated)
-it	Insert specified text at the specified bookmark (may be repeated)
-kd	Keep the data file when finished
-pd	Print the document when finished
-rm	Runs the specified macro after data transfer
-v	Make the running Word object visible
--	Null argument (see documentation).

Command Line Parameter Examples

This section provides explanations and examples of the parameters available for Advanced Command in iWord and as supported in the earlier Meta5 version 4.1 Wordtool.

-di

Disable interactive mode.

Suppresses pop-up error messages when Wordtool is run on a local desktop. Has no effect when Wordtool is run on a capsule server. The capsule server only runs in disabled interactive mode.

-dt

Delete Template

This parameter deletes the template file when Wordtool finishes running. Be sure you do not wish to keep your template file before using this parameter. This feature is used when inserting multiple sets of data into different parts of one Word document. You would use the output from a prior run as a template for the next set of data to be inserted.

-e

Error File

This parameter allows you to specify a file name for Wordtool's error log. By default, the error log is written to C:\wlog\$\$\$\$.txt (\$\$\$ will be replaced by DSK if Wordtool is run on a workstation, or E00, E01, E02... if it is run on a capsule server). If you want to view the Wordtool error log within a capsule, you can connect Wordtool to a text icon. The error log's contents will be saved in the text icon when your capsule is run. If you specify a different error log file name than the default, you will have to edit your Wordtool icon, replacing the PC Output File entry (c:\wlog\$\$\$\$.txt) with your new error log file name.

-ib

Insert Body Text

This parameter tells Wordtool where you want the data from your source icon to be inserted in your output Word document. Wordtool will copy the data into your document at the specified bookmark. For example, to copy your Meta5 data to the bookmark called BODY, you would type: -ib BODY c:\template.doc c:\output.doc To set a bookmark in your template file, open the template file in Word and select Bookmark... from the Insert menu. Specify a name for your bookmark and click Add.

Note: If you do not use the -ib parameter in your command line, Wordtool will insert your Meta5 data at the top of your document.

-io

Insert Object File

Inserts an object file at the specified bookmark. Multiple objects can be inserted by using the -io option multiple times on the command line. For example: -io Bookmark1 c:\object1.obj -io Bookmark2 c:\object2.obj c:\template.doc c:\output.doc.

-ip

Insert Picture File

Inserts a picture file at the specified bookmark. These pictures will be displayed along with your text in your finished document. Multiple pictures can be inserted by repeating the -ip option on the command line. For example: -ip Bookmark1 c:\Photo1.bmp -ip Bookmark2 c:\Photo2.bmp c:\template.doc c:\output.doc

-it

Insert Text

Insert user specified text at the user-defined bookmark in the output file. For example, to insert the text "Sales Report" at the bookmark TITLE1, you would type: -it TITLE1 "Sales Report" To insert text at multiple bookmarks, you can repeat the -it function on the command line, i.e.:

```
-it TITLE1 "Sales Report"-it TITLE2 "Chicago Market -1998"
```

This feature is commonly used to insert @-variable information in various areas in your Word document. For example: -it TITLE1 "@A" where the value of @A is Sales Report.

Note: It is necessary to enclose inserted text within double quotes if the text contains spaces or dashes.

-kd

Keep Data File

Keep the data file when finished. By default, Wordtool deletes the wdata\$\$\$.txt file when it is finished. If you wish to keep this file, simply include this parameter on the command line. This feature is used mostly for debugging purposes. Also note that the \$\$\$ token will be replaced by DSK on the desktop and E00, E01... on the capsule server.

-pd

Print Document

Print the document on the default printer when finished. To use, simply type -pd on the command line. Remember that this command will print the document on the default printer of the machine where the capsule is being run. If it is run on a capsule server then it will print out on the default printer of the capsule server, not your workstation.

-rm

Run Macro. Runs the specified macro after data transfer has completed but before printing and saving take place. For example, to run a macro titled "Macro1" you would type:

```
-rm Macro1
```

You can run multiple macros by repeating the function on the command line.

For example:

```
-rm Macro1 -rm Macro2
```

-v

Visible

Makes the running Word object visible. You will be able to watch Wordtool copy the data from the data file into your output document during a capsule run. This parameter should not be used in capsules being sent to a capsule server. To use, simply type -v on the command line.

--Null

Argument

This argument acts as a buffer between Wordtool features (-commands) and Word files on the command line (template/output files). Other than this, Wordtool ignores the -- argument.

Chapter 3. Meta5 Email Outbox and Envelope

Overview

The combination of using the Meta5 Email Outbox with the Meta5 Envelope icon in a Capsule will enable you to send email to any valid email address on the Internet. The Email Outbox is a new icon that is located inside the *Blank Icons file drawer, which is located inside the New Icons file drawer. The Envelope icon must be attached to the Email Outbox and addressed properly. The Envelope icon is also contained inside the *Blank Icons file drawer. Note that you will find some examples for how to use this functionality in the Base Examples folder (inside the *Integrator Icons file drawer) and inside the Meta5 Sample Applications file drawer.



Figure 74. Meta5 Email Outbox

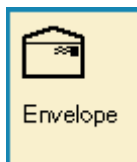


Figure 75. Meta5 Envelope

Examples for Use

- Automatically send Excel or Word output directly from Meta5 Capsules to any valid internet email address.
- Iterate sending multiple emails via the Meta5 Capsule.

Examples

Sending a reminder email to hundreds of personnel to remind them to register information on a particular web site.

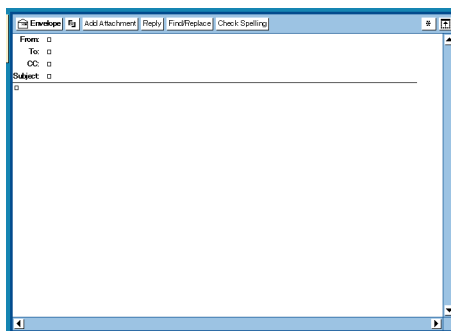
Sending a variety of reports out to specific personnel based on certain business rules or criteria.

Parameters

The Email Outbox can not be opened and contains no parameters. It only works by connecting a Meta5 Envelope to it from within a Meta5 Capsule.

The Envelope is the icon which will contain all of the necessary information to send the mail. The Envelope contains five main areas which can be filled in with text.

Note that @-variables may be used in every parameter area.



From: Type in an email address if you want it to appear INSTEAD of the actual internet address that is configured on the machine that is running the capsule. For example, if you run the capsule on a Capsule Server and the address is configured (via Meta5 Configuration Tool) to be something generic such as SalesDept@xyz.com, you may want to fill in your own personal email address.

To: Fill in valid internet email addresses here for all recipients of your email. Multiple addresses are separated by commas. Meta5 will provide an error message in the Capsule Run Log if you try to send to any invalid addresses.

The error received if sending to any invalid addresses will state:

SMTP server connection error -1006 (OR -1007)

Mail was not delivered

Run not completed. Cannot work with icon "Email Outbox". -- Check icon.

CC: Fill in valid internet email addresses here to receive copies. Multiple addresses are separated by commas.

Subject: Fill in a Subject if desired; this may contain @-variables.

Body of Envelope: Beneath the solid line, you may type any message; this may contain @-variables, Hyperlinks, HTML, or data sent into a region.

Envelope Attachments

The Envelope may contain many tools as attachments; these must precede the Envelope in the Capsule, or can be manually copied or dragged into the window that opens when you click on the window header button called [Add Attachment].

NOTE: Every time you run the capsule, the attachment(s) will automatically be replaced with the most current attachment(s); there is not need to "clean" the Envelope.

Various Meta5 icons may be valid attachments to the Meta5 Envelope:

- Excel Spreadsheet
- Word Document
- Meta5 Spreadsheet
- Text icons
- Business Objects Query
- Brio Query
- Entire Meta5 Capsules
- PC files such as DIF or RTF

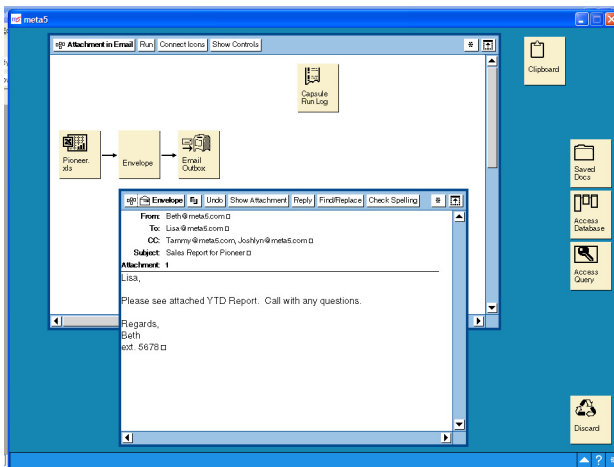
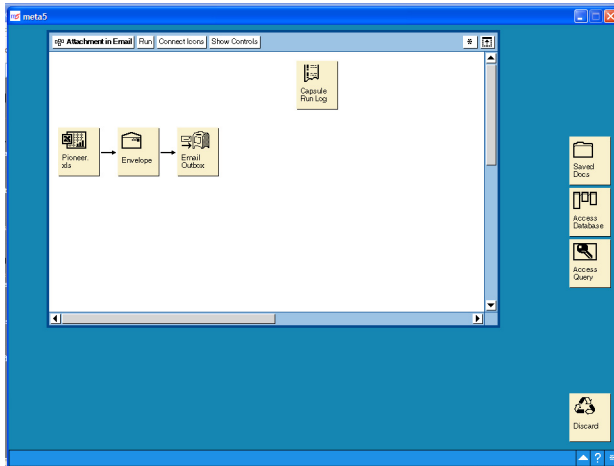
Examples for Use

Sending an attachment from Meta5

In order to send an attachment from Meta5, follow these steps:

- Copy a Capsule on to your desktop
- Copy a Meta5 Envelope and a Meta5 Email Outbox from Blank Icons into the Capsule
- Copy or Move/Drag the icon you wish to send as an attachment into the Capsule; for example, send an Excel Spreadsheet as an attachment.
- Connect the icon to be sent as an attachment (such as Excel) to the Envelope and then attach the Envelope to the Email Outbox. There are not special regions required to be set in the arrow.e
- Open the Envelope and address it properly. **NOTE:** If you enter an invalid email address, you will receive an error in the Capsule Run Log stating that the mail could not be sent.

- Run the Capsule.



Iterating Meta5 Capsule for Sending Multiple Emails

In order to "iterate" sending attachments from Meta5, follow these steps:

- Place the capsule which sends Email inside another capsule. This capsule is now the inner capsule.
- Replace all parameters that need to be dynamic inside the inner capsule with @ variables.
- Create a Spreadsheet (using a Meta5 Spreadsheet or an Excel Spreadsheet) containing all @-variables.

- Place the @-variables Spreadsheet inside the outer capsule and then connect it to the inner capsule. Set the arrow Destination Area to "controls".
- Run the Capsule.

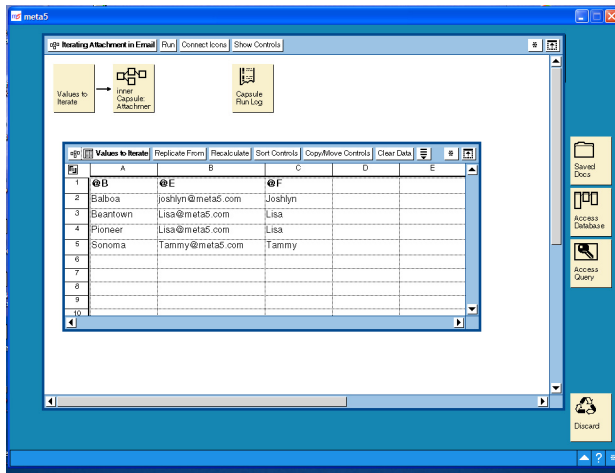
Building on the previous example, we can illustrate this.

Example @-Variables to use in the Outer Capsule:

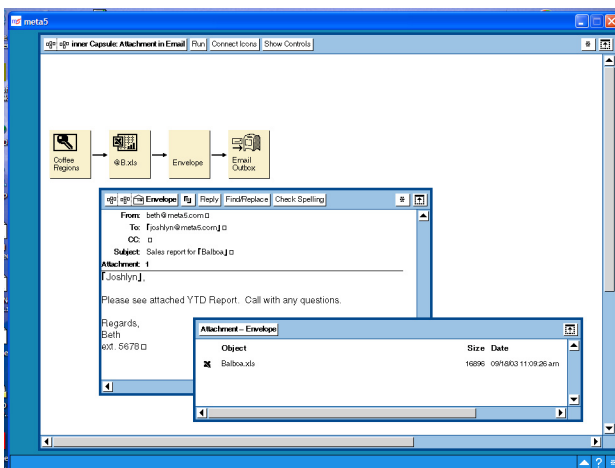
Email Address for Envelope (To:): @E

Brand for Query Constraint, Envelope Subject Area, Excel report: @B

First Name of person receiving Email for Envelope Body area: @F



The inner capsule that uses @-Variables for Query, Excel Output, and the Envelope:



Sending data into a region inside the body of the Meta5 Envelope

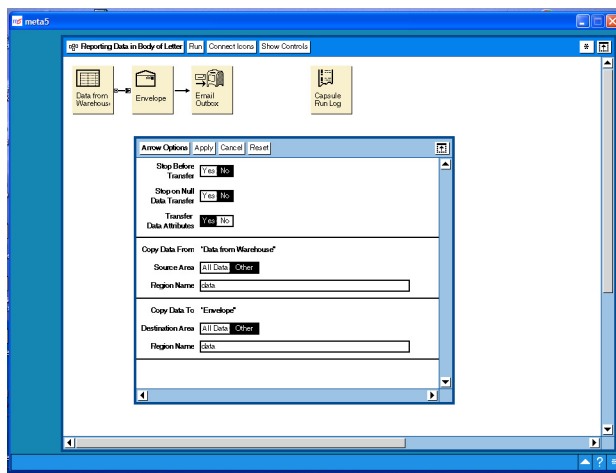
The Meta5 Envelope has functionality to contain regions in the body of the Envelope. Users can now send a table of data directly in the body of the Envelope which is an option to sending an entire attachment (such as an Excel Spreadsheet).

In order to send data into the body of the Meta5 Envelope, follow these steps:

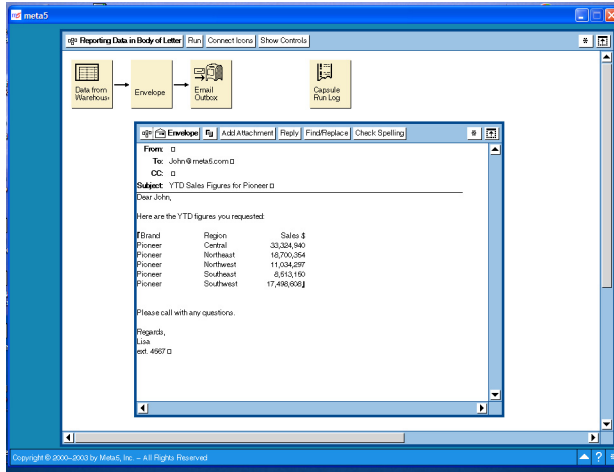
- Create a region in the Envelope's body area using the same method as you would for the Meta5 Text tool. [OPTION] the location in the body, choose [Region], and then create a name for the region, such as "data".
- Connect the source icon containing the data to the Envelope. The Meta5 Envelope will only accept text and spreadsheet data; the envelope will reject any other data format. Among the icons that can supply data to a region in the Meta5 Envelope are Meta5 Text, Word, Meta5 Spreadsheet, Excel, and Transformers.
- Set the arrow Destination Area to the name of the region created in the Envelope.

The following example illustrates this.

Data passes from a Spreadsheet into a region contained within the body of the Envelope:



Contents of the envelope:



Chapter 4. iDirList Integrator

Overview

The iDirList is a new integrator icon that allows you to list the contents of a directory or subdirectory on your computer or on any drive accessible by your computer, allowing the specification of file names or extensions. Using iDirList in a capsule will enable you to automate file processing in Meta5.

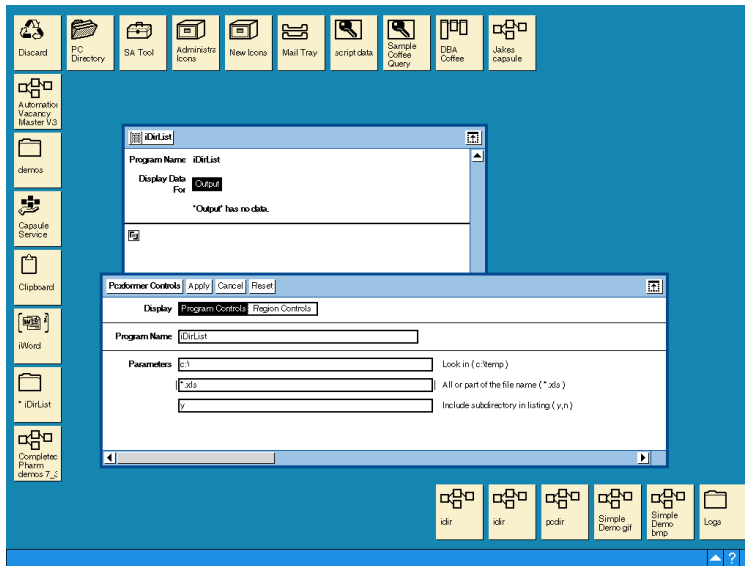
The iDirList tool is located inside the *Integrator Icons file drawer, which is located inside the New Icons file drawer. This file drawer may also be found in *Public File drawers. The iDirList tool is also located in the *Meta5 Icons file drawer within the PC Integration folder. Also, you will find some examples for how to use it in the Base Examples folder (inside the *Integrator Icons file drawer).

Examples for Use

- Find the most current spreadsheet or ASCII file within a folder. The data from the file can then be automatically extracted and loaded into a database via a Meta5 Capsule. This can be repeated at regular intervals to ensure using the most current file to load the database.
- Get a list of spreadsheets to be iterated within a capsule and then joined to other data sources.
- Show the location of specific reports.
- Create a listing of files within a subdirectory.

Parameters

When iDirList is copied to your desktop, you can open the icon and select "Show Controls" on the window header. This will display the parameters that need to be filled in.



Look in: This is the path to be searched. You can also include subdirectories such as c:\My Documents\My spreadsheets.

All or Part of the file Name: Type the file name such as September Sales.xls or part of the file name such as Sept*.xls or *.xls or *.*. All extensions are supported.

If left blank, this parameter will default to *.*.

*Note: You may use the * to look up files just as you would in a command prompt window.*

Include Subdirectory in Listing: y (default) or n. Specifying the default, y, will include any files in subdirectories within the Look in parameter. For example, if the first parameter was c:\ then all files within the subdirectories of the c drive will be included. By specifying n, only the files on the c:\ will be included.

Note that @-variables can be used for any of the parameters.

Output

The following is included as the Output of iDirList:

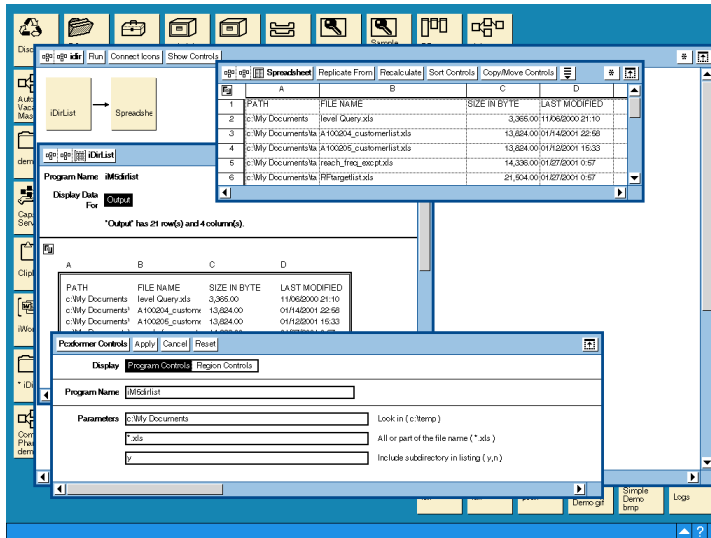
PATH: Full path to file, ie. C:\My Documents

FILENAME: Full name of file, ie. MyReport.doc

SIZEINBYTE: Size of file in bytes, ie. 14,336

LASTMODIFIED: Date file last changed, i.e 09/15/2003 21:15

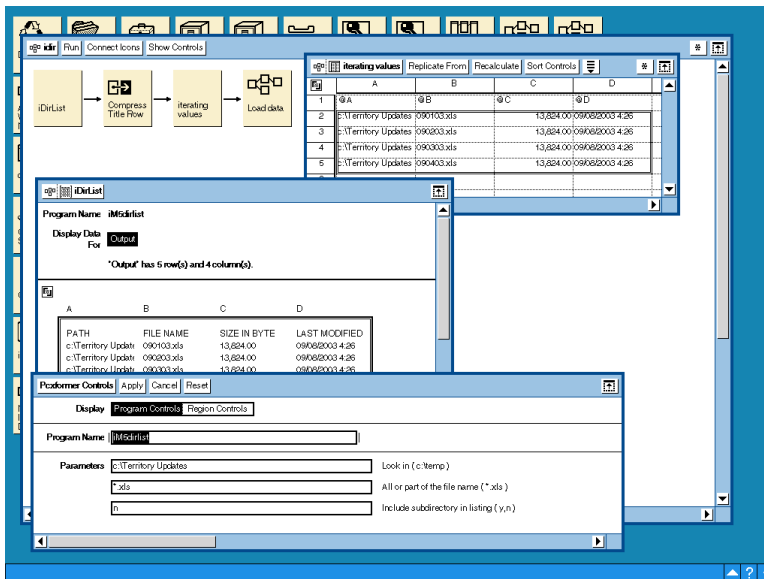
The iDirList tool can be connected to a Meta5 Spreadsheet, Text, Excel, Word, or Transformer icon to get a directory listing showing the PATH, File Name, Size, and Last Modified Date of each file. Data from the last run will appear when the tool is opened and can be copied using the "Select All" button.



Examples for Use

A folder on the C drive called Territory Updates contains daily Excel Spreadsheets with new sales territory numbers that need to be loaded into a territory table.

A completed capsule shows the parameters of the iDirList Tool where the filename can be iterated into an inner capsule and data extracted using PC directory.



Using a Sort Transformer, the files can be sorted by date modified or file name as needed.

Examples of retrieving file names using wildcards

Assume the following files exist on the C drive of your computer:

2002Budget.xls

2003Budget.xls

Butter Sales.xls

2002 Report.doc

2003 Report.doc

iDirList will return the following File Names when looking in the C drive:

**File Name
Parameter**

Files Listed with iDirList

- *.* All 5 Files will be listed
- 2003*.* 2003Budget.xls
- 2003 Report.doc

File Name Parameter	Files Listed with iDirList
*.doc	2002 Report.doc
	2003 Report.doc
Bu.xls	2002Budget.xls
	2003Budget.xls
	Butter Sales.xls

Chapter 5. iWeb Integrator

The Web Page Scraper

Meta5 has developed an intelligent link with Microsoft's Internet Explorer product. The iWeb Integrator has been developed to pull back data from a URL and pass it onto the Meta5 Excel Spreadsheet tool.

Latest stock prices, interest rates or currency conversion rates can be downloaded from the Web straight into a Meta5 Capsule.



Figure 76. iWeb Integrator icon

iWeb Integrator Icon

The iWeb Integrator icon was introduced to provide the Meta5 user with access to the Internet (or Intranet) for the purposes of pulling down information to be onward processed by Meta5. We will now review the capabilities of the iWeb Integrator and how it can be used.

The Integrator family of tools have a similar look and feel to the Meta5 Transformers. The Integrator icons have input and output regions and a controls panel that allows you to configure the tool.

What can iWeb Integrator do?

- Access data on the Web and make it available to Meta5 for onward processing.
- Treat the Web as another source of data.

The iWeb Integrator Regions

Figure 2 shows the window displayed when you open the iWeb Integrator icon.

There is a single output region called Output.

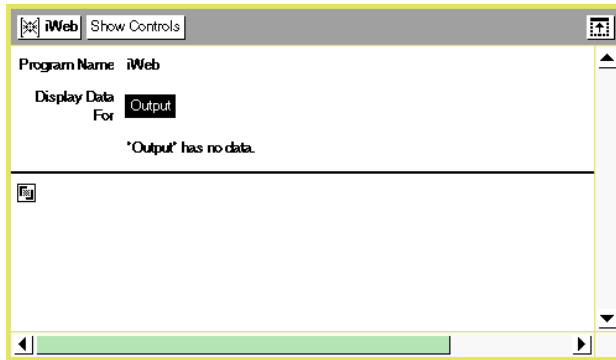


Figure 77. Output region of the iWeb Integrator

This region has the following function:

Output: The output from the iWeb Integrator after the Web has been accessed and the chosen Web page located.

The iWeb Integrator accesses the Web from the machine running the icon. For this to work, the Workstation or Capsule Server must be able to access the Internet.

Parameters of the iWeb Integrator

Click on Show Controls to review the parameters of the iWeb Integrator (see figure 3).

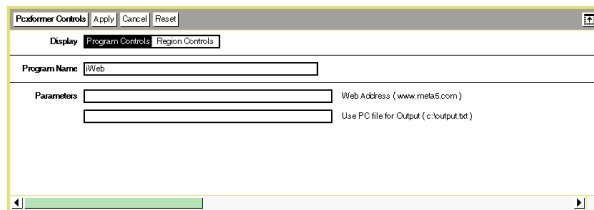


Figure 78. The parameter controls of the iWeb integrator

Program Name.	iWeb. This is the name of the program to run. This should not be changed.
---------------	---

Web Address	Specify the URL that you want Meta5 to access and place in the Output region.
Use PC file for Output	You can specify the full path and file name of where you wish to place the contents of the Web page that is pointed to by the previous parameter.

Example of use

The Web contains a lot of information that could have relevance to your business. Financial organizations may find it useful to acquire details from the various business indices and rates that are freely available on the net.

The example, chosen here, interrogates a bank's web site for their published interest rates on mortgages. The idea behind this Capsule is to be alerted the moment the interest rate drops significantly.

The Capsule will E-mail a copy of these figures to the recipient if the interest rate of the chosen product (15 Year Fixed Rate) falls below 8%.

This example can be found in the Base Examples folder within the Integrators file drawer in New Icons.

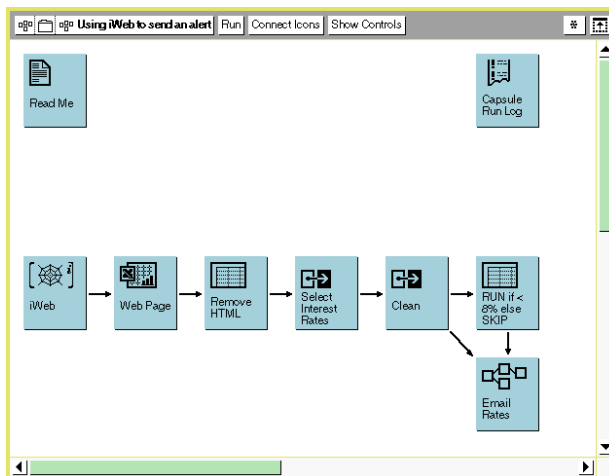


Figure 79. Using iWeb to send an alert

iWeb pulls back the entire contents of the Web page that is specified in the parameter of the iWeb integrator. By passing the data into Excel we can address the various constituents of the Web page by cell reference. To make life easier, we move the contents of the Excel Spreadsheet into a Meta5 Spreadsheet so that we can view just the texts and numerics without all the graphics and icons.

From there we clean the data so that we just pick up the table of information we need and pass that to a Capsule for emailing out. The Run Skip control is employed to decide whether the results should be emailed. By checking the interest rate for a value of less than 0.08 (8%) we can choose to run the Email Capsule or not.

Technical Note: The iWeb integrator does not currently support the use of a web proxy server. In order to use the iWeb integrator, you must be able to connect directly to the address specified in the icon controls on port 80.

Chapter 6. Brio Icon

Overview

The Brio icon allows you to run BrioQuery reports in Meta5 capsules and pass the results to other Meta5 icons. It also lets you leverage the power of capsule applications to enhance the capabilities of your Brio files. Using @variables, one can run a Brio report many times with different sets of constraints. Advanced capsule design allows automatic responses to be generated when values exceed certain ranges. The Brio icon is located in the * Blank Icons file drawer.



Figure 80. The Brio Icon

Some examples of the uses of the Brio icon are available in the Meta5 Sample Applications file drawer (inside New Icons).

The Meta5 Brio icon is a Brio .bqy file residing on the Meta5 file server. The data in the file is passed to Brio when the icon is opened.

Note: All workstations and Capsule Servers that need to use the Meta5 Brio tool must have BrioQuery installed and must be fully licensed to use it. No license is granted or implied by Meta5 when using the Meta5 Brio tool.

To export the Brio icon to disk, simply drag the icon from the Meta5 desktop onto the Windows desktop or into a Windows Explorer window. To import an existing Brio report in Meta5, simply drag the .bqy file onto the Meta5 desktop.

Brio Icon Options

There are a couple of options that can be set in the Brio icon. Select the icon and press F4 to bring up the Icon Options window. If you need to specify a user name and password to be used by Brio to connect to a data source, enter them here. Please note that only one login/password pair can be entered. This login and password will be used for all queries in the Brio icon. You can also control the format of the icon's output by setting the Transfer Type field.

The JPG format is used to output a picture of a Brio chart. This picture can be inserted directly into a Word document.

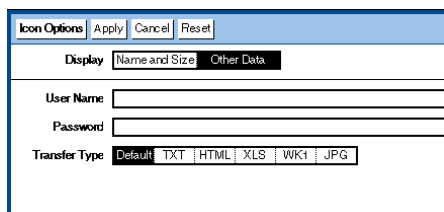


Figure 81. The Brio Icon options sheet

Using @variables in Limits in a Brio Report

When using a Brio icon in a Meta5 capsule, you can take advantage of the capsule's @variables in query limits in the Brio report. To use an @variable in a Brio query limit, double click the query limit and add the @variable to the beginning of the limit's Name field.

Note: Meta5 looks at the next two characters after the @ symbol to determine which @variable to use. To be safe, always include a space after the @variable name.

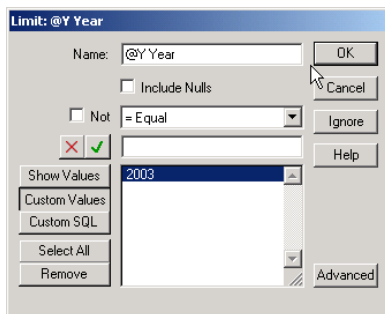


Figure 82. Using an @variable in a Brio query limit

When the capsule is run, all the @variables in the query sections are updated with the values from the capsule before the queries are processed.

Specifying an Output Section

When you run a capsule containing a Brio icon connected to another icon, Meta5 will transfer the data from the active section in the Brio report. If the active section is a query section, the query's results will be transferred. To choose a different section for output, open the arrow options and set Source Area to Other. Then, enter the name of the section in the Region Name.

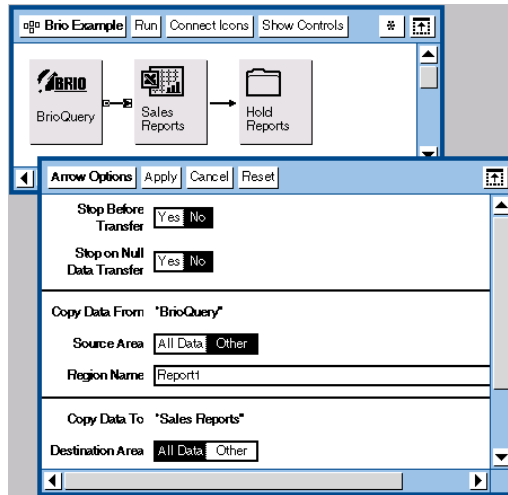


Figure 83. Specifying an Output Section

Regardless of what section is used as output when the capsule runs, all Brio query sections are processed and all sections are updated.

Using the Brio Icon in a Capsule

To use the Brio icon in a capsule, just connect it to a spreadsheet, transformer, text, Excel or Word document. Specify the section you want to transfer in the arrow options, and the username and password in the Brio icon options.

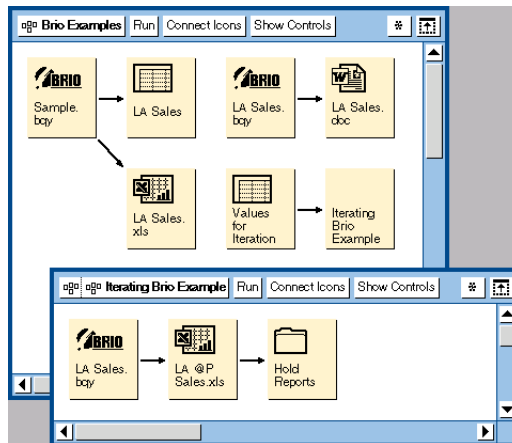


Figure 84. Using the Brio icon in a Capsule

Chapter 7. BusinessObjects Icon

Overview

The BusinessObjects icon allows you to run BusinessObjects reports in Meta5 capsules and pass the results to other Meta5 icons. It also lets you leverage the power of capsule applications to enhance the capabilities of your BusinessObjects files. Using @-variables, one can run a BusinessObjects report many times over with different sets of constraints. Advanced capsule design allows automatic responses to be generated when values exceed certain ranges. The BusinessObjects icon is located in the * Blank Icons file drawer.



Figure 85. The BusinessObjects Icon

Some examples of the uses of the BusinessObjects icon are available in the Meta5 Sample Applications file drawer (inside New Icons).

The Meta5 BusinessObjects icon is a BusinessObjects .rep file residing on the Meta5 file server. The data in the file is passed to BusinessObjects when the icon is opened.

Note: All workstations and Capsule Servers that need to use the Meta5 BusinessObjects tool must have BusinessObjects installed and must be fully licensed to use it. No license is granted or implied by Meta5 when using the Meta5 BusinessObjects tool.

To export the BusinessObjects icon to disk, simply drag the icon from the Meta5 desktop onto the Windows desktop or into a Windows Explorer window. To import an existing BusinessObjects report in Meta5, simply drag the .rep file onto the Meta5 desktop.

BusinessObjects Icon Options

There are a couple of options that can be set in the BusinessObjects icon. Select the icon and press F4 to bring up the Icon Options window. If you need to specify a user name and password to be used by BusinessObjects to connect to a data source, enter them here. Please note that only one login/password pair can be

entered. This login and password will be used for all queries in the BusinessObjects icon. You can also control the output format of the icon by setting the Transfer Type field.

Figure 86. The BusinessObjects icon options sheet

Using @-variables in conditions in a BusinessObjects Report

When using a BusinessObjects icon in a Meta5 capsule, you can take advantage of the Meta5 capsule's @-variables by setting them as conditions in the BusinessObjects report. To use an @-variable in a BusinessObjects condition, set the condition's operator to "In list" and its operand to "Type a new prompt." Type in the @-variable you wish to use as your prompt.

Figure 87. Using @variables in BusinessObjects conditions

When the capsule is run, all the conditions are updated with the values from the capsule @-variables before the queries are processed.

Specifying an Output Section

When you run a capsule containing a BusinessObjects icon connected to another icon, Meta5 will transfer the data from the active report tab in the BusinessObjects file. To choose a different report tab for output, open the arrow options and set Source Area to Other. Then, enter the name of the report tab in the Region Name.

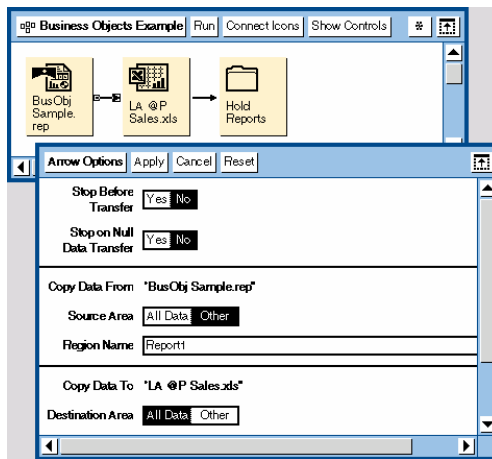


Figure 88. Specifying an Output Section

Regardless of what section is used as output when the capsule runs, all BusinessObjects sections are refreshed.

Using the BusinessObjects Icon in a Capsule

In order to run the BusinessObjects icon in a capsule and transfer the results from it to other tools, you need to connect it to Text, Excel Spreadsheet or Word Document icons. If you wish to transfer the BusinessObjects results to a Meta5 Spreadsheet icon or a Meta5 Transformer, you must first connect the BusinessObjects icon to a "Text to Spreadsheet" Transformer. Remember to specify the report tab you want to transfer in the arrow options, and the username and password in the icon options.

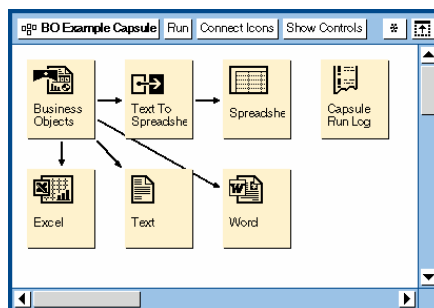


Figure 89. Using the BusinessObjects icon in a Capsule

Meta5 Publications

This section lists Meta5 publications. To order copies of the books listed here, or to get more information about a book, see your Meta5, Inc. representative.

- **Meta5 Volume 1**
 - Getting Started with the Meta5 Developer's Desktop*
 - PC Integration Tools*
- **Meta5 Volume 2**
 - Spreadsheet User's Guide*
 - Plot User's Guide*
- **Meta5 Volume 3**
 - Text User's Guide*
 - Layout User's Guide*
- **Meta5 Volume 4**
 - Capsule User's Guide*
 - BASIC Tool User's Guide*
- **Meta5 Volume 5**
 - Data Access Tools User's Guide*
- **Meta5 Volume 6**
 - Forms User's Guide*
- **Meta5 Volume 7**
 - Transformers Guide*
- **Meta5 Volume 8**
 - Error Messages and Codes*
- **Meta5 Volume 9**
 - Installing Meta5 LAN Components*
 - System Administration Guide and Reference*
- **Meta5 Volume 10**
 - Database Gateway Services Guide*
 - Administering Databases with Meta5*
- **Meta5 Volume 11**
 - Installing and Configuring Meta5 Open Clients*
 - Developing Applications with Open Data Access Service*

- **Meta5 Volume 12**

Administering Host Services for MVS:DB2 and Cooperative Application Services

Planning and Installing Host-to-LAN Communications for MVS

Glossary

This glossary defines terms that are used in this book and throughout the Meta5 library. If you do not find the term you are looking for, see the index of this book.

@-variable. A control variable used to represent a value in any tool that can be run inside a capsule application. These values are frequently used as constraints and variable titles in the Query, Plot, Spreadsheet, Layout, and other Meta5 tools.

axis. One of the horizontal or vertical reference lines relative to which data values are charted on an axes graph.

baseline. The primary reference line for an axes graph. The baseline usually represents a data value of zero.

Capsule. A tool in which you build an automated processing sequence using other icons.

capsule application. An automated processing sequence built within a Capsule tool. Typical uses for capsule applications include running standard reports or processing changing data in standard ways.

color palette. A window in which colors can be selected for a plot, slice, or desktop object.

column. (1) A vertical field in a table. Columns may be identified by letters or text strings at the top of the display area. (2) A vertical field in a spreadsheet. Columns are identified by letters at the top of the display area.

data column. A column of data from a source icon used by the Plot tool to determine values in a graph.

data row. A row of data from a source icon used by the Plot tool to determine values in a graph.

depiction area. In the Single Plots or Single Slices page of the Plot Controls window, the middle third of the window, which displays a picture of each plot or slice in the graph that has been constructed.

display area. The area below the Plot window header into which the user can copy data to be charted and where the final graph is displayed.

division. An increment between each step.

explode. To separate a slice from the other slices in a pie graph to make the slice more visible or to place more visual emphasis on the slice.

field. In a window, an identifiable area where the user can enter information, select a choice, or view information.

Gantt Chart. A time-based graph designed for comparing rates, planning schedules, and tracking milestones for a project or task; for example, planned production versus actual production.

graph. A graphic representation of quantitative information comparing one or more sets of variable data. The Plot tool can be used to create pie (circular) and axes graphs.

legend. An explanatory caption included with a graph to identify what each plot represents.

message bar. The rectangular area along the bottom of the desktop, where advisory or error messages are displayed.

page. (1) In a window, an alternate view that provides a different set of options for setting controls and defining the behavior of an icon. (2) In a window, the amount of information that can be displayed at one time.

plot. A graphical representation of one row or column of numerical values.

row. (1) The horizontal component of a table. A row contains exactly one value for each column of the table. (2) A record, or single instance, in a database table. (3) A horizontal field in a spreadsheet, identified by a row number.

scale. The range of values assigned to an axis and used for charting data on a graph. The Plot tool automatically scales each numeric axis of a graph to best display the data charted on it.

slice. A section of a pie graph that represents an individual data value.

source icon. An icon used to pass information to other icons.

Spreadsheet. A tool used to perform calculations and analyses, schedule projects, and perform iteration in embedded capsules.

step. A major increment on an axis.

tick marks. Short lines that intersect an axis to mark steps and divisions.

title frame. A transparent rectangle containing text that the user types directly into the Plot window.

x-axis. The horizontal line at the bottom of a graph, used to chart values.

y-axis. The vertical line at the left of a graph, used to chart values.

Index

B

- Brio 105
 - @variables 106
 - examples 105
 - export 105
 - icon options 105
 - in a capsule 107
 - jpg 106
 - license 105
 - output 106
 - overview 105
 - password 105
 - user name 105
- BusinessObjects 109
 - @variables 110
 - examples 109
 - export 109
 - icon options 109
 - in a capsule 111
 - license 109
 - output 110
 - overview 109
 - password 109
 - user name 109

E

- Email 87
 - attachments 89
 - envelope 87
 - examples 87, 89, 92
 - multiple addresses 90
 - outbox 87
 - overview 87
 - parameters 88
 - regions 92
 - variables 88
- Envelope 89
- Excel 1
 - @variables 7
 - active sheet 5
 - calculations 13
 - cell referencing 5
 - chart exporting 16
 - chart naming 16
 - charts 15
 - creating multiple sheets 18
 - defined name 6
 - defining region names 6
 - exporting 16
 - formatted spreadsheets 4
 - IF statement 14
 - inputs 8
 - integrator icon 1
 - iteration 18
 - license 1
 - passing data 10, 12, 13, 14

Xlaunch 2

I

- iDirList 95
 - examples 95, 97
 - output 96
 - overview 95
 - parameters 96
 - wildcards 98
- iExcel 22
 - advanced commands 28, 46
 - auto format 46
 - calculating totals 38
 - calculations 37
 - capsule server 46
 - comment font 45
 - comment sizing 45
 - defined name 47
 - delete template 46
 - disable interactive mode 45
 - file format 46
 - format data using source range 26
 - format region 47
 - hyperlinks 39
 - insert at cell or name 25
 - insert into sheet 25
 - insert range 45
 - insert text 47
 - iteration 35
 - keep data file 48
 - multiple sheets 34
 - multiple source region 42
 - null argument 49
 - page break 26
 - parameters 24
 - paste cells 39
 - paste comments 40
 - print workbook 48
 - print worksheet 48
 - regions 23
 - Rename Template Sheet 25
 - repeat template sheet usage 27
 - run macro 27
 - Save Output as 26
 - source range 48
 - template 36
 - update chart 26
 - Use PC file for Output SS 26
 - Use PC file for Template SS 25
 - Xlaunch 28
- Internet Explorer 101
- iWeb
 - alert 103
 - run skip control 104
- iWeb Integrator 101
 - examples 103

- output 102
- parameters 102
- program name 102
- proxy server 104
- regions 101
- use pc file for output 103
- web address 103

iWord 83

- advanced commands 70, 83
- Alternative Output formats 75
- data 74
- examples 70
- HTML 75
- Insert Body at bookmark 75
- Insert Object at bookmark 75
- Insert Picture at bookmark 75
- iterating 64
- parameters 68
- paste bookmark 72
- paste object 73

- paste picture 72
- regions 67
- template 73
- Use PC file for Output Doc 69, 75
- Use PC file for template Doc 69
- Wordtool 83

W

- Web Page Scraper 101

- Word 51

- @-variables 56
- bookmarks 55
- examples 52
- excel chart 63
- exporting 54
- formatting 55
- importing 52
- overview 51
- passing data 59, 60, 63