GRANTA MI[™] version 11

GRANTA MI Excel® Importer Reference Guide



GRANTA MI[™] is the leading system for materials information management in engineering organizations. It enables you to control, analyze, and securely share critical corporate data on materials and processes, managing the materials information lifecycle.

© Granta Design 2018 All rights reserved

CES Selector and GRANTA MI are trademarks of Granta Design Ltd. For other Granta product trademarks, see www.grantadesign.com/smallprint.htm

Adobe[®], Adobe[®] PDF, and Acrobat[®] are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Microsoft[®], Excel[®], PowerPoint[®], Internet Explorer[®], SQL Server[®], Windows[®], and Windows Server[®] are registered trademarks of Microsoft Corporation or its subsidiaries in the United States or other countries. SpreadsheetGear[®] is a registered trademark of SpreadsheetGear LLC.

Granta Design Ltd. makes reasonable efforts to explicitly acknowledge all trademarks cited in our literature or on our website. If you would like us to add or alter an acknowledgement, please <u>contact us</u>.

Release notes, documentation, and Knowledge Articles for the current and all previous GRANTA MI releases are all available on the Granta Support website. Go to <u>www.grantadesign.com</u> and click SIGN IN to log into your My Granta page, then click on **Documentation**.

We welcome feedback on this documentation. Please let us know if anything is unclear, if you spot an error, or have an idea for new content, by emailing <u>docs@grantadesign.com</u>.

Document version: MI11/03 Published: July 2018

Table of Contents

 1.1 Features of the Excel Importer	5 6 7 8 9 9 9 9
 Creating Excel templates	6 7 8 9 9 9 11
 MI:Toolbox documentation <i>Template overview</i>	6 7 9 9 9 11
 2 Template overview	7 9 9 9
 <i>Attribute Lookup worksheet</i> 3.2 Attribute Name 	8 9 9 11
3.2 Attribute Name	9 9 11
	9 11
3.3 DataType	11
3.4 Units	
3.5 Parent Attribute	12
3.6 Worksheet	12
3.7 Specimen	12
3.8 Data Range	12
3.9 Quality Range	13
3.10 Read Trailing Zeroes	13
3.11 Import	13
3.12 Line Type	14
3.13 Log	15
3.14 Interpolate	15
3.15 Invert	15
3.16 Estimate	16
3.17 Show As Table	17
3.18 Parameter n Range	18
3.19 Parameter n Default Value	18
3.20 Parameter n Name	19
4 Parameter Lookup worksheet	.20
4.1 Parameter	20
4.2 Data Range	20
4.3 Units	21
5 Data worksheets	. 22
5.1 Named Ranges	22
5.2 Discrete data	24
5.3 File data	24
5.4 Float Functional data	25
5.5 Discrete functional data	29
5.6 Hyperlink data	29

	5.7	Equations and Logic data	30
	5.8	Tabular data	32
	5.9	Multi-value data	34
	5.10	Quality ratings	35
	5.11	Record color	36
	5.12	Subset membership	36
6	Ітроі	rt Options worksheet	38
	6.1	Auto-placement	38
	6.2	Conflict resolution and merging options	44
	6.3	Placement of records into a specific folder hierarchy (Tree)	48
	6.4	Auto-Links	50
	6.5	Access Control	53
	6.6	Valid tables	54
	6.7	Exported Worksheet Naming	54
7	Advai	nced features	55
	7.1	Creating multiple records	55
	7.2	Merging and appending	58
	7.3	Deleting existing data	65
	7.4	Template workbooks	66
8	Тетр	late locations	67
Арр	oendix .	A. Setup for multi-step imports	68
	A.1 Th	e Import Sequence worksheet	68
Арр	pendix	B. Excel Importer CLI	69
	B.1 Co	mmand-line arguments	69
	B.2 Us	age example	72
Арр	pendix	C. Troubleshooting	73
	C.1 Exi	t codes	73

1 About this document

This document describes how to format a Microsoft Excel[®] workbook for use with the GRANTA MI Excel Importer.

1.1 Features of the Excel Importer

The Excel Importer can be run from within the MI:Toolbox application or it can be used via the Remote Import web browser interface; see the Help for these applications for details. The Excel Importer can also be run via a command-line interface (CLI); see *Appendix B*.

- All data types can be imported. This includes functional data (grid, series, discrete) and metadata.
- Multiple records can be imported at once. These can be imported from multiple workbooks, from individual worksheets within a workbook, or as individual rows or columns within a worksheet.
- New data can be imported into existing records or as new records. New functional and tabular data can be merged with or appended to existing data.
- Trailing zeroes can be imported for point data and range data, and point and range columns in tabular data.
- Record properties such as the tree name (short name), record GUID, color, and subset membership can all be imported.
- Static record links can be automatically generated using data in the Excel workbook to search for related records.
- Data links can be automatically generated using data in the Excel workbook to search for related records.
- Unit conversion is carried out during the import using the unit conversion information in the database.
- The user can select a location for imported records (if using MI:Toolbox), or records can be automatically placed in the database using criteria given in the Excel workbook.
- Settings for access control permissions can be set by users with GRANTA MI database Grant, Power User, or Admin privileges.
- Data can be imported into multiple tables across one or more databases by using a template that defines a sequence of import steps, with each step using different import options. See *Appendix A, Setup for multi-step imports*.
- Excel *.xls, and Excel 2007 *.xlsx and *.xlsm formats are supported. Note, however, that the Excel Importer cannot reference or access data in Excel Tables.
- The Excel Importer uses integrated SpreadsheetGear[®] functionality to allow Excel-compatible functions to be included in importer templates. For example, record names can be autogenerated using the CONCATENATE function to join several text strings into one text string. For information about Excel functions supported by SpreadsheetGear, please refer to this page: <u>www.spreadsheetgear.com/products/spreadsheetgear.net.aspx</u>

1.2 Creating Excel templates

Excel templates for exporting and importing data can be created in a number of different ways:

- In the MI:Toolbox Excel Exporter plug-in. 'Automatically-generated templates' can be created, based on a set of selected records. By default, automatically generated templates are saved locally, and can subsequently be edited, as required, in Microsoft Excel.
- In MI:Viewer. Any MI:Viewer user can download an Excel template for importing (or exporting) data to/from a selected table or folder, using the attribute information, formatting, and other options specified in a preconfigured Excel Template Definition (set up by the MI administrator in MI:Admin). Templates generated in this way can subsequently be edited, as required, in Microsoft Excel.
- In Microsoft Excel. Template workbooks can be defined manually in Microsoft Excel by creating correctly-formatted worksheets specifying the required data entry, lookup, and import options. This is a specialist task, requiring familiarity both with Microsoft Excel and with the structure of the database from which the data is exported/into which the data will be imported.

1.3 MI:Toolbox documentation

Reference documentation for the MI:Toolbox exporters and importers, including this document, can be found in any of the following locations:

- On your MI:Toolbox host, in the Excel Importer plugin installation folder (typically, C:\Program Files\Granta\GRANTA MI\Toolbox\plugins\Importers\Excel\Documentation).
- In MI:Viewer: click on Help > Reference Documentation.
- Sign in to your <u>My Granta Account</u> to access all of the reference documentation for the current and previous GRANTA MI releases.

2 Template overview

An Excel template for the Excel Importer contains worksheets with data to be imported and a number of template worksheets used by the importer:

- *Attribute Lookup* worksheet: This worksheet is **always** required and contains information on record properties and attributes.
- *Parameter Lookup* worksheet: This is only required for functional data, tabular data and multi-value point data, and contains information on parameters and tabular columns.
- *Import Options* worksheet: This is used to implement more advanced auto-placement and auto-linking features.

Once set up, the template worksheets can be hidden to prevent unwanted changes.

Blank example workbooks are provided in the *Documentation* folder for the Excel Importer plug-in, see Section **8**.

It is also possible for the Excel Importer to import data into more than one table, or database, by including an additional *Import Sequence* worksheet that defines a sequence of import steps, each with its own Attribute Lookup, Import Options, and Parameter Lookup worksheets. Data will then be imported in the order specified, allowing, for example, links to be created between the records being imported. See *Appendix A, Setup for multi-step imports* for details.

3 Attribute Lookup worksheet

The *Attribute Lookup* worksheet contains a list of all the GRANTA MI attributes to be imported, their data type and units, and the location of the data to be imported for each attribute in the Excel workbook. Each row represents an attribute.

Each cell or range of cells where data is stored is identified by defining a *named range*. These named ranges are used in the *Attribute Lookup* worksheet to locate the data on other worksheets within the workbook.

Some columns in the Attribute Lookup worksheet take a value directly, and some use a named range to look up data from elsewhere.

1	ExcelImporterTemplate_Example1.xls									
		А	В	С	D	E				
1	1	Attribute Name	DataType	Units	Worksheet	Data Range				
2	2	Record Name	RCN		DataSheet	MI_RECORDNAME				
3 D		Density	RNG	kg/m^3	DataSheet	MI_DENSITY				
4	ł	Young's Modulus	RNG	GPa	DataSheet	MI_YOUNGSMODULUS				

Figure 1. Example showing required columns in the Attribute Lookup worksheet

The first row of the *Attribute Lookup* worksheet must contain the column titles listed below. 'Attribute Name' must be the title of the first column. The remaining columns may be in any order.

Column Title	Required/optional	for these data types
Attribute Name	Required	All
<u>DataType</u>	Required	All
<u>Worksheet</u>	Required	All
<u>Data Range</u>	Required	All
<u>Units</u>	Required	Functional, Point, Range, Equations and Logic
Parameter n Range	Required	Functional, multi-value Point with parameters, Tabular, Equations and Logic
Import	Optional	All
Quality Range	Optional	All
Specimen	Optional	All
Read Trailing Zeroes	Optional	Point, Range, Tabular
<u>Invert</u>	Optional	Equations and Logic
<u>Estimate</u>	Optional	Float functional, Point, Range
Interpolate	Optional	Functional
Line Type	Optional	Functional
Log	Optional	Functional
Parameter n Default Value	Optional	Functional
Parameter n Name	Optional	Functional
Show As Table	Optional	Functional
Parent Attribute	Optional	Metadata

All populated rows in the worksheet are processed, blank rows are ignored. Errors will be reported if the worksheet is incorrectly formatted, or the location of the data to be imported (specified by the Worksheet andData Range columns) cannot be found in the current workbook. See Section 5.1 for more information on named ranges.

3.1.1 Valid Boolean values

The following are all valid Boolean values:

- True/False
- Yes/No
- 0, not 0 (for cells that look like integers)
- vrai/faux (attempts to use localized versions of True and False)
- Pass/Fail

3.2 Attribute Name

This column specifies the name of the attribute in the GRANTA MI database that the data is to be imported into. It can also include properties of records that are not attributes, such as subset membership and record color. The attribute name in the template must match the attribute name in the database exactly.

The Record Name attribute must be the first row in the list after the column title, and specifies the name of the new record that will be created during import (see Figure 2).

A row for the Record Short Name is optional. If the row is not included, the value for Record Name is used. The Short Name is also referred to as the *"Tree Name"*.

If the attribute is a meta-attribute, the meta-attribute name is to be entered into this column. For example, for an attribute named 'Young's Modulus' with a meta-attribute named 'Standard Deviation', 'Standard Deviation' should be entered into the Attribute Name column.

ExcelImporterTemplate_Example1.xls									
	А	В	С	D	E	F	G		
1	Attribute Name	DataType	Units	Parent Attribute	Worksheet	Specimen	Data Range		
2	Record Name	RCN			DataSheet		MI_RECORDNAME		
3	Record Short Name	RCSN			DataSheet		MI_SHORTNAME		

Figure 2. The first row in the Attribute Lookup must contain information on the record name

3.3 DataType

This column specifies the data type of the attribute using a three or four-letter code. The incoming data type must match the data type of the attribute in the database.

Table 1. DataType codes for importing into a GRANTA MI database

DataType code	GRANTA MI data type	Comment
DAT	Date	
DCT	Discrete	includes multi-value discrete
FDA	Float Functional	series (graph)

DataType code	GRANTA MI data type	Comment
FDD	Discrete Functional	discrete (table)
FIL or FILE	File	
GFA	Float Functional	grid (graph)
НҮР	Hyperlink	
INT	Integer	
LOG	Logical	
LTXT	Long Text	
MAFN	Equations and logic	
PIC	Picture	
PNT	Point	includes multi-value point
RCLR	Record Color	See Section 5.11 on page 36 for color codes
RCN	Record Name	A line will be recognized as a record name if, and only if, its datatype is RCN. The name in the Attribute Name column is for information only.
RCSN	Record Short Name	A line will be recognized as a record short name if, and only if, its datatype is RCSN. The name in the Attribute Name column is for information only.
RNG	Range	
STXT	Short Text	
SUBSET	Subset	See Section 5.12
TABL	Tabular	

Notes:

- Only one line of a given property type should exist on the *Attribute Lookup* worksheet, or errors will occur when importing the data.
- For meta-attributes, the DataType refers to the meta-attribute rather than the parent attribute. A meta-attribute may have a different data type to that of its parent.
- For information on the different formats of functional data (for example, series, grid discrete), see Section 5.

3.4 Units

This column contains the units of the data. The unit in the template must either match or be convertible into the unit of the attribute in the GRANTA MI database that the data is to be imported into. If unit conversion is required, the conversion information must be specified in the database.

For discrete attributes, you do not need to specify the name of the discrete type.

For meta attributes, the Units column refers to the meta attribute rather than the parent attribute. A meta attribute may have a different unit to that of its parent.

3.4.1 Currency

Currency is a unit, and should be treated as a unit, although it may be considered a 'special' unit.

The currency unit can be used in combination with other units, where an attribute's unit is defined in the same way (see *Figure 3*).

The currency for a database is set as a property of the database in the MI:Admin Schema tool. To import data using the currency for the database, use 'currency' in the Units column (excluding quotes). To import data in a chosen currency, use the three letter ISO 4217 currency code in the Units column. The exchange rate from the GRANTA MI configuration database is used to convert the value in the data import worksheetinto the database value.

For example, suppose the currency for a database is US Dollars (USD). If you use units of 'currency' (see *Figure 3*), the value in the data import worksheet will be assumed to be in US Dollars. If you use units of 'AUD' (see *Figure 4*), the value in the data import worksheet will be treated as Australian Dollars.

	Α	В	С	D	E
1	Attribute Name	DataType	Units	Worksheet	Data Range
2	Capital cost	RNG	currency	DataSheet	MI_CAPITALCOST
3	Price	RNG	currency/kg	DataSheet	MI_PRICE

Figure 3. Importing using the database currency in the Attribute Lookup worksheet Units column

	A	В	C	D	E
1 Attr	ibute Name	DataType	Units	Worksheet	Data Range
2 Cap	ital cost	RNG	AUD	DataSheet	MI_CAPITALCOST
3 Pric	e	RNG	AUD/kg	DataSheet	MI_PRICE

Figure 4. Importing using a chosen currency in the Attribute Lookup worksheet Units column

3.5 Parent Attribute

If the attribute is a meta-attribute, this column contains the name of the GRANTA MI database attribute that the meta-attribute belongs to. For example, for an attribute named 'Young's Modulus' with a meta-attribute named 'Standard Deviation', 'Young's Modulus' should be entered into the Parent Attribute column (see Figure 5). The meta-attribute must exist in the database, and the attribute name in the template must match the attribute name in the database exactly.

9	ExcelImporterTemplate_Example1.xls										
		A	В	С	D	E	G				
	1	Attribute Name	DataType	Units	Parent Attribute	Worksheet	Data Range				
	2	Record Name	RCN			DataSheet	MI_RECORDNAME				
	3	Record Short Name	RCSN			DataSheet	MI_SHORTNAME				
	4	Density	RNG	kg/m^3		DataSheet	MI_DENSITY				
	5	Young's Modulus	RNG	GPa		DataSheet	MI_YOUNGSMODULUS				
	6	Standard Deviation	PNT		Young's Modulus	DataSheet	MI_YM_SD				

For non-meta-attributes, the value in this column should be left blank.

Figure 5. Required columns for meta-attributes in the Attribute Lookup worksheet

3.6 Worksheet

This column contains the name or index of the worksheet where the data for the attribute is located.

When the index is used, enter the numeric value into the Worksheet column. For example, if you want the data to be read from the first worksheet, enter '1' in the Worksheet column. An index cannot be used when importing multiple records per workbook, and the template worksheets are in the same workbook as the data.

3.7 Specimen

This column determines whether multiple records are to be created from a single workbook. It contains one of the valid options in the cell itself. The valid options are:

Row	Generate records from individual rows within a worksheet.
Column	Generate records from individual columns within a worksheet.
Worksheet	Generate records from individual worksheets.

If no value is specified, then a single record is created from a single workbook.

See also Section 7.1 to see how setting a 'Specimen' value for the 'Record Name' attribute can be used to create several records from a single workbook.

3.8 Data Range

This column contains the name of the Excel range containing the cells where the data to be imported into the attribute is stored.

(See Section 5.1 for more information on named ranges.)

3.9 Quality Range

This column contains the name of the range where the quality rating for the data is stored. (See Section 5.1 for more information on named ranges.)

3.10 Read Trailing Zeroes

For use with point data (including multi-value point), range data, and point and range columns in tabular data, this column takes a Boolean value that determines whether or not numeric data will be imported with trailing zeroes. See also *Valid Boolean values* on page 9.

If the value is set to TRUE, the number is imported as it is displayed in Excel. If the value is set to FALSE, the Excel Importer will *not* pay attention to the trailing zeroes.

Defaults to FALSE if unspecified.

	📓 ExcelImporterTemplate_Example1.xls										
	А	В	С	E	G	Н					
1	Attribute Name	DataType	Units	Worksheet	Data Range	Read Trailing Zeroes					
2	Record Name	RCN		DataSheet	MI_RECORDNAME						
3	Record Short Name	RCSN		DataSheet	MI_SHORTNAME						
4	Density	RNG	kg/m^3	DataSheet	MI_DENSITY	TRUE					
5	Young's Modulus	RNG	GPa	DataSheet	MI_YOUNGSMODULUS	TRUE					
6	Standard Deviation	PNT		DataSheet	MI_YM_SD	TRUE					

Figure 6. Read Trailing Zeroes column in the Attribute Lookup worksheet

The decimal separator for trailing values is assumed to be a period.

For multi-value point data, trailing zeroes can be imported for the attribute value, but not for the parameter value.

In Excel, set the cell formatting to one that can display the required number of decimal places, for example, Text, Number, or Custom.

For Text formatting, the cell is displayed exactly as entered. For example, if you type '2.070', then the cell displays '2.070' and the Excel Importer reads '2.070' for import if Read Trailing Zeroes is TRUE or '2.07' otherwise.

For Number formatting, the cell is displayed with the set number of decimal places. For example, if you type '2.07' and set the number of decimal places to 3, then the cell displays '2.070'. Also, if you type '2.07000' and set the number of decimal places to 3, then the cell displays '2.070'. In both cases, the Excel Importer reads '2.070' for import if Read Trailing Zeroes is TRUE or '2.07' otherwise.

3.11 Import

This column takes a Boolean value (TRUE/FALSE) that determines whether or not the attribute data will be imported. See also *Valid Boolean values* on page 9. Defaults to TRUE if unspecified.

If the data is not imported, it can still be used for other purposes e.g. auto-placement. See Section 6.1 for more information.

3.12 Line Type

The **Line type** column is used to specify, for float functional attributes, how series data should be displayed on the graph: using a line, using markers, or with both a line and markers. Enter the required value for the series in the top cell of the column:

- Enter *lines, markers,* or *both,* or the name of the range in the worksheet where the option is stored.
- In auto-generated templates, where data validation is used to constrain the possible values for each series, select **LinesOnly**, **MarkersOnly**, or **Both** from the dropdown menu.



Defaults to lines. if unspecified.

If multiple series (and different line types for each series) are to be imported for an attribute, the stem of the named ranges should be entered into the **Line Type** column. See Section **5.4.1** for more information.

3.13 Log

For use with float functional data, and equations and logic data, this column takes a Boolean value (TRUE/FALSE) that determines whether or not the data on the y-axis of the graph is displayed and interpolated logarithmically. See also *Valid Boolean values* on page 9.

The setting applies to the entire functional data graph.

Defaults to FALSE if unspecified.

3.14 Interpolate

For use with float functional data, this column determines the interpolation type for parameters. Possible values:f

Value	Result
None	No interpolation. This overrides the interpolation method specified at the functional attribute level in the database schema i.e. in MI:Admin.
Automatic	The interpolation method specified at the functional attribute level in the database schema should be adopted.
Linear	Linear interpolation. This overrides the interpolation method specified at the functional attribute level in the database schema.
CubicSpline	CubicSpline interpolation. This overrides the interpolation method specified at the functional attribute level in the database schema.

Defaults to None if unspecified.

For new templates, to avoid confusion, it is recommended that this setting is set to **Automatic**, and the interpolation type is set on a per parameter basis in the *Data* worksheet (see Section 5.4.2); the settings in the *Data* worksheet will override the setting in the *Attribute Lookup* worksheet.

To maintain backwards compatibility with older templates:

- FALSE is an alias for 'None'.
- TRUE is an alias for 'Automatic'.

3.15 Invert

For use with Equations and Logic data, this column takes a Boolean value (TRUE/FALSE) that determines whether or not the curve data graph axes are transposed. See also *Valid Boolean values* on page 9.

Value	Result
True	Plot the parameter value on the y axis and the attribute value on the x axis
False	The axes are not transposed. Defaults to FALSE if unspecified.

3.16 Estimate

For use with range, point, or float functional data, this column takes a Boolean value (TRUE/FALSE) that determines whether or not the data is flagged as an estimate. See also *Valid Boolean values* on page 9.

Fatigue strength at 10^7 cycles	43.4 to 50.3	ksi (estimate)
Fracture toughness	65.5 to 104	ksi.in^0.5 (estimate)
Mechanical loss coefficient (tan delta)	0.0007 to 0.00087	(estimate)

For functional data, it applies to the 'whole' functional data graph or table, not individual data points:

- Tensile Response								
Ø	L	Fensile Response (11 axis) Hide Graph (estimate)						
		350 + + + + + + + + + + + + + + + + + + +						

For multi-value point data, it applies to all values, not individual values.

Value	Result
True	Flag the data as estimated
False	Do not flag the data as estimated.

Defaults to FALSE if unspecified.

Note that named ranges for the Estimate flag for multi-value point and float functional data will not be included in automatically generated templates created by MI:Toolbox or downloaded from MI:Viewer. If you need a template that can import estimate flags for these data types, it will be necessary to edit the auto generated template in Microsoft Excel to add the required ranges to the *Data* and *Lookup* worksheets.

3.17 Show As Table

For use with float functional data, this column takes a Boolean value (TRUE/FALSE) that specifies how the data will be displayed in MI:Viewer datasheets. See also *Valid Boolean values* on page 9.



Defaults to FALSE if unspecified.

3.18 Parameter n Range

where *n* is an integer, from 1 to 30, e.g. Parameter 1 Range, Parameter 2 Range, etc.

For use with functional data, equations and logic data, and multi-value point data, this column contains the name of the range in the worksheet where the parameter values are stored (see *Figure* **7**.

'Parameter 1 Range' is used as the **default X-axis parameter** when displaying the data as a graph. This is important to note when importing **series** data. The X axis parameter cannot be changed on series graphs when viewing the data in MI:Viewer.

If multiple series are to be imported for an attribute, the stem of the named ranges should be entered into the Parameter Range column; see Section 5.4.1.

When used with tabular attributes, this column contains the name of the range in the worksheet where the data values for a tabular column are stored. One parameter range is used for each column.

	ExcellmporterTemplate_Example1.xls									
A B C E F G -							12			
	1	Attribute Name	DataType	Units	Worksheet	Specimen	Data Range	Parameter 1 Range	Parameter 1 Default Value	
	2	Record Name	RCN		DataSheet	ROW	MI_RECORDNAME			
	3	Record Short Name	RCSN		DataSheet	ROW	MI_SHORTNAME			
	4	Series Daily Sunshine	FDA	hr	Functional Data	WORKSHEET	MI_SERIESDAILYSUNSHINE	MI_SERIESDAILYSUNSHINE_TIME	MI_SERIESDAILYSUNSHINE_TIME_DEFAULT	

Figure 7. Parameter Range and Default Value for a float functional attribute 'Series Daily Sunshine' containing series functional data

ſ	ExcelImporterTemplate_Example1.xls									
		А	В	С	D	E	F	G	Q	
	1	Attribute Name	DataType	Units	Parent Attribute	Worksheet	Specimen	Data Range	Parameter 1 Range	
	2	Record Name	RCN			Data	ROW	MI_RECORDNAME		
	3	Short Name	RCSN			Data	ROW	MI_SHORTNAME		
	5	Tension strength	PNT	MPa		MultiValueData	WORKSHEET	MI_TENSIONSTRENGTH	MI_TENSIONSTRENGTH_BASIS	

Figure 8. Parameter Range for a multi-value point attribute 'Tension strength'

3.19 Parameter n Default Value

where *n* is an integer, from 1 to 30, e.g. Parameter 1 Default Value, Parameter 2 Default Value, etc.

For use with functional data, this column contains the name of the range in the worksheet where the default parameter value is stored (see *Figure 7*).

The expected suffix for the named range is '_DEFAULT', excluding quotes.

The stem of the named range is the parameter's named range. For example, if the named range for the Parameter 1 Range is MI_TEMPERATURE, the named range for the Parameter 1 Default Value should be set to MI_TEMPERATURE_DEFAULT.

If it is not set, the data will inherit the default parameter value set for the attribute in MI:Admin.

3.20 Parameter n Name

where *n* is an integer, from 2 to 10, e.g. Parameter 2 Name, Parameter 3 Name etc.

For use with functional data, this column contains the name of the range in the worksheet where the name for parameter range is stored. For example, a tensile test carried out at temperature of 23°C can be given the name 'Room Temperature'. 'Parameter 2' is Temperature, 'Parameter 2 Range' contains the named range that refers to the cell where the value of 23 is stored, and 'Parameter 2 Name' contains the named range that refers to the cell where the value of 'Room Temperature' is stored.

The Parameter Name can be thought of as a label for that parameter in that series, and only applies to the functional data for that attribute in the template. It does not need to correspond to a discrete parameter name in the database.

A 'Parameter Name' cannot be assigned to Parameter 1 as it is the default X-axis.

If multiple series are to be imported for an attribute, the stem of the named ranges should be entered into the Parameter Range column. See Section 5.4.1 for more information.

4 Parameter Lookup worksheet

The *Parameter Lookup* worksheet contains a list of all the parameter and tabular column ranges in the *Attribute Lookup* worksheet and how they are mapped onto the GRANTA MI database parameters or tabular columns. This worksheet is only required for functional data, tabular data and multi-value point data. Each row represents the parameter range or tabular column range for an attribute.

Functional data with multiple series also only requires one row per parameter range for an attribute; see Section 5.4.1.

The order of these columns must be as listed below, but the column titles in the template need not match with those listed. All columns are required.

- 1. Parameter
- 2. Data Range
- 3. Units

4.1 Parameter

This column contains the name of the parameter in the GRANTA MI database that the data is to be imported into. The parameter name in the template must match the parameter name in the database exactly.

As each row represents the parameter range for an attribute, it is possible for the same parameter name to be repeated in the column, each instance for a different attribute. For example, the attributes *Young's Modulus with Temperature* and *Compressive Strength with Temperature* would require two *Temperature* rows in the worksheet, one for each attribute.

When used for tabular data, this column contains the name of the tabular column that the data is to be imported into. For a **Local Data** column, the column name in the template must match the column name in the database exactly. This column is also used to reference the named range for the 'linking value' for tabular rows, see Section **5.8**.

4.2 Data Range

This column contains each unique **Parameter Range** from the *Attribute Lookup* worksheet, that is, the named range in the worksheet where the parameter/tabular column data is stored. (See Section **5.1** for more information on named ranges.)

If the **Data Range** for a **Parameter Range** is not found, then the Excel Importer will look for a parameter in the database with the name given for 'Parameter Range'.

4.3 Units

This column contains the units of the parameter/tabular column data. The units in the template must match the unit symbol of the parameter/tabular column in the GRANTA MI database that the data is to be imported into. Alternatively, the units in the template must be convertible into the parameter/tabular column units using conversion information in the database.

For discrete parameters, you do not need to set a Unit.

The currency unit can be used for numeric parameters. For more information on using the currency units, see Section *3.4, Units*.

5 Data worksheets

The data to be imported into the GRANTA MI database can be stored in any number of worksheets in the workbook.

5.1 Named Ranges

Each cell or range of cells where data is stored must be identified by defining a 'named range'. It is these named ranges that are used in the template *Attribute Lookup* worksheet.

DataType code	GRANTA MI data type	Notes
DAT	Date	The named range must refer to a single cell where the data is located. The value in the destination cell must be in a valid Excel date format.
DCT	Discrete	The named range must refer to a single cell where the data is located.
DCT	Multi-value Discrete	The named range must refer to adjacent cells where the data is located. (See Section $5.9.1$)
FDA	Float Functional (Series)	 The named range for data values must refer to adjacent cells arranged in a column (or two adjacent columns for range data) for each series. The named range for parameter values for Parameter 1 must also point to adjacent cells in a single column for each series. The named range for parameter values for Parameter 2 and greater may refer to a single cell for each series. All types of series functional data to be imported are described by the same code (FDA), the different types of functional data (e.g. single series or multiple series, point or range) are deduced from the format of the data and whether it is to be interpolated or not, as specified in the 'Interpolate' column. (See Section 5.4.1)
FDD	Discrete Functional	 The named range for data values must refer to adjacent cells arranged in a column for each series. The named range for parameter values for Parameter 1 must also point to adjacent cells in a single column for each series. The named range for parameter values for Parameter 2 and greater may refer to a single cell for each series. (See Section 5.5)
FIL or FILE	File	The named range for data values must refer to up to five adjacent cells where the data is located. (See section 5.3)
GFA	Float Functional (Gridded)	 The named range for data values must refer to adjacent cells arranged in a column (or two adjacent columns for range data). The named range for parameter values for Parameter 1 must also point to adjacent cells in a single column for each series. The named range for parameter values for Parameter 2 and greater must point to adjacent cells in a single column and contain the same number of rows of data as for Parameter 1 and data values ranges.

Table 2. Notes on Named Ranges for all data types

DataType code	GRANTA MI data type	Notes
		All types of grid functional data to be imported are described by the same code (GFA), the different types of functional data (i.e. point or range) are deduced from the format of the data. (See Section 5.4.4) Grid functional data is always interpolated.
НҮР	Hyperlink	The named range must refer to a single cell where the data is located. The value in the destination cell must be formatted as a Uniform Resource Locator (URL) or a HTML hyperlink. (See Section 5.6)
INT	Integer	The named range must refer to a single cell where the data is located.
LOG	Logical	The named range must refer to a single cell where the data is located.
LTXT	Long Text	The named range can refer to multiple adjacent cells (over multiple rows and/or multiple columns) where the data is located.
MAFN	Equations and logic	The named range must refer to a single cell where the data is located. The value in the cell is the name of an Expression in the table. (See Section 5.7)
PIC	Picture	The named range must refer to a single cell where the data is located. The value in the cell is the file path of the image. The path may be an absolute path, or relative to the template.
PNT	Point	The named range must refer to a single cell where the data is located.
PNT	Multi-value Point	The named range for the Data Range must refer to adjacent cells where the data is located. The named range for each Parameter must also point to adjacent cells. (See Section 5.9.2)
RCLR	Record Color	The named range must refer to a single cell where the data is located. (See Section 5.11 .)
RCN	Record Name	The named range must refer to a single cell where the data is located.
RCSN	Record Short Name	The named range must refer to a single cell where the data is located.
RNG Range		The named range must refer to two adjacent cells in the same row or column. The first cell must contain the lower value (minimum), the second the higher value (maximum). If either cell is blank, an open-ended range is imported (minimum or maximum value only, as appropriate). If point data (i.e. a single value) is to be imported into range attribute the named range must refer to a single cell.
STXT	Short Text	The named range must refer to a single cell where the data is located.
SUBSET	Subset	The named range must refer to a single cell where the data is located. The value in the cell is a Boolean value indicating subset membership. See Section 5.12 .
TABL	Tabular	The named range for the 'Data range' must refer to multiple adjacent cells in the same column where the index number for each row of tabular data is located. The named ranges for tabular columns must also point to adjacent cells in
		a single Excel column for each tabular column. (See Section 5.8)

5.2 Discrete data

In order to import discrete type values, a discrete type must already exist in the database. This can be checked in the MI:Admin application. The discrete type values in the *Data* worksheet must match those in the database.

See Section 5.9.1 for information on multi-value discretes.

5.3 File data

For file attributes, the named range consists of up to five adjacent cells (in a row or column). The cells must always be in the same order, but may be blank.

These cells define the five properties of GRANTA MI file data, as described in Table 4.

Table 3. Defining File attributes

Cell	Cell content
Cell 1	The file name, including the path. The path may be an absolute path, or relative to the template. Use !FILENAME to import the current workbook.
Cell 2	Text description of the file
Cell 3	Where the file content should be displayed in MI:Viewer. Valid options are as hyperlink.
Cell 4	Indicates whether the contents of the file should be indexed on import (see 5.3.1). Valid values are TRUE/FALSE, YES/NO (case insensitive). See above.
Cell 5	Indicates whether the file is displayed instead of the datasheet when MI:Viewer is in read mode. Valid options are TRUE or FALSE. The default setting is FALSE if unspecified.

	Document	Doc Description	Target	Indexed	Display as record
Record 1	C:\DemoFiles\4391 Ti Data.txt	test data		TRUE	TRUE
Record 2	IFILENAME	import source	_blank	TRUE	FALSE

Figure 9. Named ranges for importing a file attribute (one highlighted).

In the example above, each row is a record. No column heading is required, but one can be used to make the worksheet easier to interpret. The column heading should not be included in the named range.

5.3.1 File indexing

Attributes of type *File* in GRANTA MI are identified as searchable or not searchable. If a file attribute is marked as not searchable, it will not be searched during any queries performed by MI users. If the attribute is searchable, then the file contents will be included in searches.

A file must be *indexed* in order to be searchable. When importing data using the Excel Importer, file attributes flagged as "indexable" have their contents indexed on import, making it possible for them

to be searched. File attributes flagged as *not indexable* in the Excel Import worksheet will not be indexed on import, and their content will not be searchable, *even if* the attribute is searchable.

If no "indexable" value is specified for a file attribute in the Excel import file, when that file is imported:

- If the attribute is searchable, the file will be indexed, and therefore its contents may be searched.
- If the attribute is not searchable, the file will not be indexed, and its contents cannot be searched.

5.4 Float Functional data

Functional attributes can have numeric and/or discrete parameters. For example, the parameter 'Test Type' can consist of the named values: 'Test Type A', 'Test Type B', etc.

In order to import named parameters values, a 'discrete' parameter must already exist in the database. This can be checked in the MI:Admin application. The named parameter values in the *Data* worksheet must match those in the database.

Parameter 1 values

On the *Data* worksheet, the float functional data should be sorted so that the values for Parameter 1 (the default x-axis) are in ascending or descending order.

Series 01	Line Type:	LINES		
Value (MPa)	Strain (%)	Temperature (°C)	Test Type	
0	0	23	Test A	
11	0.5	24	Test A	
20.5	1	23	Test A	
28	1.5	23.4	Test A	
34	2	23	Test A	Parameter 1 Default Value
38	2.5	21	Test A	
42	3	25	Test A	
43	3.5	23	Test A	
43.5	4	23	Test A	
44	4.5	23	Test A	
				1

Parameter 1 Range

Figure 10. Example of float functional data on the Data worksheet.

Parameter default values

The named range refers to a single cell that contains the default value (see Figure 10). For a numeric parameter, the value must be a number. For a discrete parameter, the value must be a text string.

If your numeric parameter uses named values, you must still refer to the default value by its numeric value. For example, if you have a parameter named *Month*, where the value of 1 has been named *January*, 2 has been named *February* and so on up to 12, if you want to refer to *September*, you should set the default value to 9.

5.4.1 Series functional data

Series functional data is imported as a set of data series. On the *Data* worksheet:

- Named ranges are used to identify the data for the functional attribute in the same way as they are used for non-functional attributes. To distinguish each series in an attribute, each name is suffixed with '_XX' where XX is the series number starting from 01.
- Named ranges are used to identify the parameter interpolation type and scale in an attribute. The suffixes are '_INTERP' and '_SCALE'.

On the Attribute Lookup worksheet, the stem of the named ranges is used.

At the time of import, the Excel Importer will assume that the name of the range should be suffixed with '_XX' for each series, where XX is the series number starting from 01.

Each series number must be assigned consecutively: 01, 02, 03 etc. This suffix convention applies to Data Range, Parameter Range, and Line Type.

The named ranges defined for a single series functional attribute should not be the same as the stem of the named ranges defined for multiple series functional attributes. The suffix convention will not distinguish between different attributes and problems will occur.

The named ranges for Data Range and Parameter Range(s) in a single series must all be of equal length (i.e. contain the same number of cells) in a given functional attribute.

For functional data with multiple series, although the number of points across series can vary, it is best to define named ranges for Data Range and Parameter Ranges that are long enough to contain the maximum number of data points. Blank cells at the end of range will be ignored.

For example, if *Series 01* contains 10 points, but *Series 02* consists of 9 points, the named ranges for *Series 02* should still consists of 10 cells, with the last cell as the blank cell.

Be aware that the longer the defined range is (for example if entire columns are selected), the slower the import will be.

5.4.2 Interpolation

The named range *Parameter* identifies a cell that contains the interpolation type for the parameter. The setting is stored in the cell itself; valid values are shown in Table 4.

Value	Description
None	No interpolation. This overrides the interpolation method specified at the functional attribute or parameter level in the database schema i.e. in MI:Admin.
Automatic	The interpolation method specified at the functional attribute or parameter level in the database schema should be adopted.
Linear	Linear interpolation. This overrides the interpolation method specified at the functional attribute or parameter level in the database schema.
CubicSpline	CubicSpline interpolation. This overrides the interpolation method specified at the functional attribute or parameter level in the database schema.

Table 4. Parameter interpolation values

5.4.3 Scale

The named range *Parameter***SCALE** identifies a cell that contains the axis scale for parameters. This setting determines whether the multi-dimensional interpolator used by functional data interpolates on linear or logarithmic scales. The setting is stored in the cell itself; valid values are shown in Table 5.

Table 5. Parameter scale values

Value	Description
Linear	The values should be represented on a linear scale. This overrides the scale specified at the functional attribute or parameter level in the database schema.
Log	The values should be represented on a logarithmic scale. This overrides the scale specified at the functional attribute or parameter level in the database schema.
Default	The values should be represented on a linear or logarithmic scale as specified at the functional attribute or parameter level in the database schema.

Example

Multiple series of data are to be imported for a functional data attribute.

• Attribute Name = Tensile Ult. Strength.

The parameters for this attribute are:

- Parameter 1 = Strain (%)
- Parameter 2 = Temperature (°C)
- Parameter 3 = Test Type

On the Attribute Lookup worksheet:

Range	is set to
Data	TensUltStr_Value
Parameter 1	TensUltStr_Strain
Parameter 2	TensUltStr_Temp
Parameter 3	TensUltStr_TestType
Line Type	TensUltStr_SeriesLineType

On the *Data* worksheet:

Values	are stored in range named
Data values for series 01	TensUltStr_Value_01
Parameter 1 values for series 01	TensUltStr_Strain_01
Parameter 2 values for series 01	TensUltStr_Temp_01
Parameter 3 values for series 01	TensUltStr_TestType_01
Line type for series 01	TensUltStr_SeriesLineType_01
Data values for series 02	TensUltStr_Value_02
Parameter 1 values for series 02	TensUltStr_Strain_02
and so on.	
Interpolation type for parameter 1	TensUltStr_INTERP
Scale type for parameter 1	TensUltStr_SCALE

Type of Functional Data	Notes
Single series of point data	The named range for data values must refer to adjacent cells in a single column.
	The named range(s) for parameter values must also refer to adjacent cells in a single column.
Single series of range data	The named range for data values must refer to adjacent cells in two adjacent columns, the lowe value (minimum) being contained in the first, the high value (maximum) in the second.
	The named range(s) for parameter values must refer to adjacent cells in a single column.
Multiple series of point data	The named range for data values for each series must refer to adjacent cells in a single column.
	The named range(s) for parameter values for each series must refer to adjacent cells in a single column.
	These ranges must be named using the suffix convention noted above.
Multiple series of range data	The named range for data values for each series must point to adjacent cells in two adjacent columns, the low value being contained in the first, the high value in the second.
	The named range(s) for parameter values must refer to adjacent cells in a single column.
	These ranges must be named using the suffix convention noted above.

Table 6. Defining Named Ranges for series functional data

5.4.4 Grid functional data

Grid functional data is imported as a single block, so it is not necessary to use the _XX suffix with grid functional data. In grid functional data, all the parameter values must be specified for each data point. Figure 11 shows an example of the layout of grid functional data. The named ranges for each of the parameters and the data range must be of equal length.

The **Parameter n Range** columns in the *Attribute Lookup* worksheet contain the stem for the named range for each parameter. In the *Data* worksheet, the suffix for the parameter interpolation type and scale in an attribute are '_INTERP' and '_SCALE'. See Sections 5.4.2 and 5.4.3 for valid values.

Table 7.	Defining	Named	Ranges	for	grid	functional	data
----------	----------	-------	--------	-----	------	------------	------

Type of data	Notes
point data	The named range for data values must refer to adjacent cells in a single column. The named range(s) for parameter values must also refer to adjacent cells in a single column.
range data	The named range for data values must refer to adjacent cells in two adjacent columns, the lower value (minimum) being contained in the first, the high value (maximum) in the second. The named range(s) for parameter values must refer to adjacent cells in a single column.

	A	В	С	D	E	F
1						
2		Data Range				
3		Min	Max	Parameter 1	Parameter 2	Parameter 3
4		1	2	25	0.05	Type A
5		2	4	50	0.05	Type A
6		3	6	100	0.05	Type A
7		4	8	150	0.05	Type A
8		5	10	200	0.05	Type A
9		3	6	25	0.05	Туре В
10		5	10	50	0.05	Type B
11		7	14	100	0.05	Type B
12		9	18	150	0.05	Type B
13		11	22	200	0.05	Type B
14						

Figure 11. Example of grid functional data

5.5 Discrete functional data

The data for discrete functional attributes is discrete text values rather than numeric values. Discrete functional data is formatted in the same way as series functional data (see Section 5.4.1), with the exceptions that Line Type does not apply, and the data should be treated as series of point data.

5.6 Hyperlink data

A cell containing data for a hyperlink attribute can contain either a Uniform Resource Locator (URL) or a HTML hyperlink.

5.6.1 Uniform Resource Locator (URL)

The URL format consists of a protocol and location e.g. http://www.grantadesign.com

5.6.2 HTML hyperlink

The HTML format consists of an address, target, and description:

```
<a href="address" target="target">Description</a>
```

where

- address is the URL of the destination
- target is the target pane for the hyperlink (see below).
- description is a description of the hyperlink.

The HTML hyperlink must contain the URL at least.

If the target is omitted, the value '_self' will be used.

If the description is omitted, the address will be used.

MI_CONTENT	Opens the new page in the right pane of MI:Viewer.
MI_TREE	Opens the new page in the left (tree) pane of MI:Viewer.
_top	Opens the new page in the full body of the same browser window. This is useful for leaving the current frameset and creating a frameless page.
_parent	This is similar to "_top" but will refer to the immediate parent of a frame. In more advanced frame usage, there may be several nested frames, and this allows more control over which frames are specified.
_blank	Opens the new page in a new browser window, leaving the page with the referring link still open behind it.
_self	Opens the new page within the same frame as the referring link.

The value for target can be one of the following:

Example:

Granta website

5.7 Equations and Logic data

Equations and Logic data is imported as the name of an expression, a set of parameter extents, and a set of curves. The parameter extents are the minimum and maximum values for each parameter, for which the expression is valid. If no value is given for the minimum and maximum, the parameter is assumed to be unbounded.

A curve consists of a value for each parameter and a label for the curve. The first curve will become the default curve.

Named ranges for equations and logic data use the stem_suffix convention, similar to Float Functional data (see section 5.4).

On the Data worksheet, named ranges are used to identify the data for the equations and logic data.

On the Attribute Lookup worksheet, the stem of the named ranges is used.

- The **Data Range** column contains the stem for the set of curves, which is also the named range for the cell containing the expression name.
 - The suffix for the parameter values for each curve (on the *Data* worksheet) is • *ParamX_CurveYY*'. Each curve number must be assigned consecutively: 01, 02, 03 etc.
 - The suffix for the label for each curve (on the *Data* worksheet) is '*Label_CurveYY*'. Each curve number must be assigned consecutively: 01, 02, 03 etc.
- The **Parameter n Range** columns contain the stem for the parameter extents. The suffixes for the parameter extents (on the *Data* worksheet) are '_*MIN*' and '_*MAX*'.

Example: an equations and logic attribute.

• Attribute Name = Fatigue model

The parameters for this attribute are:

- Parameter 1 = Number of Cycles
- Parameter 2 = Stress Ratio
- Parameter 3 = Temperature

On the Attribute Lookup worksheet:

Range	is set to
Data	FatMod
Parameter 1	Cycles
Parameter 2	SR
Parameter 3	Temp

On the Data worksheet (see Figure 12):

Values	are stored in range named
Parameter 1 min value	Cycles_MIN
Parameter 1 max value	Cycles_MAX
Parameter 2 min value	SR_MIN
Parameter 2 max value	SR_MAX
Parameter 3 min value	Temp_MIN
Parameter 3 max value	Temp_MAX
Parameter 1 value for curve 01	FatMod_Param1_Curve01
Parameter 2 value for curve 01	FatMod_Param2_Curve01
Parameter 3 value for curve 01	FatMod_Param3_Curve01
Label for curve 01	FatMod_Label_Curve01
Parameter 1 value for curve 02	FatMod_Param1_Curve02
Parameter 2 value for curve 02	FatMod_Param2_Curve02
Parameter 3 value for curve 02	FatMod_Param3_Curve02
Label for curve 02	FatMod_Label_Curve02
and so on.	

	A B	С	D	E	F		
1							
2	Record Name	Excel Impo	orter record				
3							
4				-			
5	Model type	Fatigue	Model				Expressior
6				-			
7	Cycles	1	10000000				
8	Stress Ratio	2	32				Parameter
9	Temperature	23	300				extents
10							
11		Label	Cycles	Stress Ratio	Temperature		
12	Curve 1	Fully reversible	1000	4	317.15	_	0
13	Curve 2	Tension	2000	5	328.15		Curves
14	Curve 3						
15							

Figure 12. Example of equations and logic data

In this example, the attribute 'Fatigue model' has been assigned the expression 'Fatigue Model', with three parameters: *Cycles, Stress Ratio, Temperature,* and two curves.

5.8 Tabular data

Tabular data is imported as a set of rows and columns. Data can be imported for 'Local Data' columns and linking values can be imported to populate other column types, if the tabular attribute is set up to link to another table in the database. There is no need to import data for 'Linked Attribute', 'Linked Column' or 'Linked Records' columns, as all the information needed to populate these columns is defined in MI:Admin.

On the *Data* worksheet, named ranges are used to identify the row order, linking value and local data for the tabular columns.

- The named range specified for the 'Data Range' in the *Attribute Lookup* worksheet contains an index number to identify the rows in the tabular data i.e. 1, 2, 3, 4, etc.
- The named range specified for 'Parameter 1 Range' in the *Attribute Lookup* worksheet contains the linking values used to link tabular rows to other data in the database.
- The named ranges specified for 'Parameter 2 Range' upwards in the *Attribute Lookup* worksheet contain the data to be imported into Local Data columns.

The named ranges for the local data columns and the linking value must also be declared on the *Parameter Lookup* worksheet, see Figure 13. Note that the entry MI LINKED VALUE in the **Parameter** column can appear more than once, if data for more than one tabular attribute is being imported from the same workbook. MI LINKED VALUE will have a different named range in the **Data Range** column for each tabular attribute.

5.8.1 Row order

The row order number identifies rows in the tabular data. This is useful if the data spans more than one Excel row.

5.8.2 Linking value

When a tabular attribute is linked to another table, each row of tabular data can have a 'linking value' that is used to link to records in the linked table by matching to data values for the 'linking attribute' defined in the tabular attribute.

5.8.3 Local Data columns

Local Data columns may contain data of types: range, point, integer, short text, long text, discrete, picture, file, hyperlink, logical, date. The named ranges for local data columns of these data types should be set up in the same way as for attributes of these data types. The one exception to this is 'file', described below.

File Local Data column

The named range (in the *Attribute Lookup* worksheet) must refer to a single cell (in the *Data* worksheet) where the data is located. The value in the cell is a file path or a HTML hyperlink.

If the value is a **File path**, the path may be an absolute path, or relative to the template. For example:

```
..\..\TestFiles\Plug-ins\Excel\Test.txt
```

If the value is an HTML hyperlink, the format is:

Description

For example:

```
<a href="..\..\Test\Plugins\Excel\Test.txt" target="NewWindow">Test
File</a>
```

See section 5.6.2 for alternative values for *target*.

5.8.4 Example

This example describes how to set up the Excel importer for a tabular attribute linked to another table in the database.

• Attribute Name = Substances found in this material

The tabular columns for this attribute are:

- CAS Number (local data)
- Amount (local data)
- Substance Name (linked records)
- Status (linked attribute)

There is no need to import data for the 'Substance Name' or 'Status' columns.

On the Attribute Lookup worksheet:

Range	is set to
Data	MI_ROW_DATA
Parameter 1	MI_LINKED_VALUE
Parameter 2	MI_CAS_NUMBER
Parameter 3	MI_AMOUNT

On the Parameter Lookup worksheet (see Figure 13):

Parameter	Data Range
CAS Number	MI_CAS_NUMBER
Amount	MI_AMOUNT
MI LINKED VALUE	MI LINKED VALUE

On the *Data* worksheet (see Figure 14):

Data	stored in range named
Index number for the row	MI_ROW_DATA
Data for tabular column 'CAS Number'	MI_CAS_NUMBER
Data for the tabular column 'Amount'	MI_AMOUNT
Short text data for linking value	MI_LINKED_VALUE

Figure 13 shows how the local data columns and the linking value are represented on the *Parameter Lookup* worksheet.

	A	В	С
1	Parameter	Data Range	Unit
2	CAS Number	MI_CAS_NUMBER	
3	Amount	MI_AMOUNT	g
4	MI LINKED VALUE	MI_LINKED_VALUE	

Figure 13. Example of a Parameter Lookup worksheet for tabular data

Figure 14 shows how data for local tabular columns CAS Number and Amount, and the 'linking value', which links each row to the relevant Substance record (its name), is laid out on the *Data* worksheet for the tabular attribute 'Substances found in this material'. Also required is an index number for each row.



Figure 14. Example of Data worksheet for tabular data

5.9 Multi-value data

5.9.1 Multi-value discrete

The named range for the Data Range must consist of adjacent cells. You can use cells in multiple columns (as in Figure 15) or cells in multiple rows.

In order to import discrete type values, a discrete type must already exist in the database. This can be checked in the MI:Admin application. The discrete type values in the *Data* worksheet must match those in the database.



Figure 15. The Data Range for a multi-value discrete attribute 'Product Form'

5.9.2 Multi-value point

You will require a named range for the Data Range, and one for each Parameter Range. Each named range must consist of a single column of cells.

Figure 16 shows the Data Range and Parameter Range for a multi-value point attribute '*Tension* strength' with one discrete parameter '*Basis'*. The Attribute Lookup worksheet row for this attribute is shown in *Figure 8*.



Figure 16. The Data Range and Parameter Range for a multi-value point attribute 'Tension strength'

5.10 Quality ratings

For all data types except functional data, the named range must refer to a single cell where the quality rating for the data is located.

A multi-value attribute only has one quality rating for an attribute's data, even though the data has multiple values (see example in *Figure 16*).

5.10.1 Ratings for continuous quality ratings systems

For a 'continuous' quality ratings system, the cell should contain a numeric value only.

5.10.2 Ratings for discrete quality ratings systems

For a 'discrete' quality ratings system, the cell can contain the name of the quality rating. (You may also use the object identity of the rating if it is known.) The quality rating in the cell must correspond to one of the values in the quality ratings system in the database. If it does not, neither the data value nor its quality rating will be imported and this will be reported during the import. If the cell is empty, then no quality rating is imported for this data.

5.10.3 Quality ratings for functional data

Each data point for functional data can have its own unique quality rating. The Quality named range should contain an equal number of rows to the Data Range. If there are not an equal number of rows, a warning will occur while importing and the functional data will not be created.

5.11 Record color

The color of a record can be specified using the RCLR DataType code.

- If this named range is defined but empty, the value defaults to 0 (red)
- If this named range is undefined, it defaults to the 'table default' this is the color of the root record of the table.

Possible color values are:

Record Color Code	Color Name		
0	Red		
1	Lime		
2	Blue		
3	Yellow		
4	Fuchsia		
5	Aqua		
6	Maroon		
7	Green		
8	Navy		
9	Olive		
10	Purple		
11	Teal		
12	Gray		
13	Black		
14	White		
15	Silver		

5.12 Subset membership

Subset membership information is exported and imported using the SUBSET DataType. This takes the value 'TRUE' for a given subset if the record belongs to that subset, and blank if it does not belong to the subset.

Example:

- 'Alumina (88)' is a member of the subset 'All Materials';
- 'Carbon Steel, AISI 1010, annealed' is a member of the 'All Materials' and 'Metals' subsets;
- 'Nitrile rubber (NBR, unreinforced)' and 'ABS (40% aluminum flake)' are both members of the 'All Materials' and 'Polymers' subsets.

	А	В	С	D	E	F
1	1 Record Properties		Subsets			
2	Record Name	Short Name	All Materials	Polymers	Metals	
3	Alumina (88)	Alumina (88)	TRUE			
4	Carbon steel, AISI 1010, annealed	AISI 1010, annealed	TRUE		TRUE	
5	Nitrile rubber (NBR, unreinforced)	Nitrile rubber (NBR,	TRUE	TRUE		
6	ABS (40% aluminum flake)	40% aluminum flake	TRUE	TRUE		
7						
H 4 > M Data / Attribute Lookup / Export Lookup / 💭 / 🛛 🛛 🔲						
When records are exported, a value of *_blank* is exported for 'not in the subset', and *TRUE* for 'in the subset'. When the record is re-imported back into the database:

- The record is added to any subsets that have a value of TRUE
- The record is removed from any subsets with a value of FALSE. (The parent record is not removed from the subset).
- No action is taken for subsets that are blank.

6 Import Options worksheet

The *Import Options* worksheet is used to specify where data imported using the template will be placed, and how import conflicts should be resolved. Options on this worksheet cover:

- Auto-placement of data into a particular database, table, folder, record, and subset.
- Placement of records into a specific folder hierarchy within the destination database.
- How naming conflicts between incoming data and existing records in the database should be handled.
- Automatic linking, where new records are automatically linked to existing records in the database based on their record names, GUIDs, or particular attribute values, and/or where data in imported records is linked to data in existing records.
- Setting access control permissions on incoming records.
- Identifying the tables into which records may be imported when not using auto-placement.

6.1 Auto-placement

The auto-placement options in the *Import Options* worksheet allow new records to be imported into a particular database, table, and folder, or record, with no user intervention to specify the location.

The Excel Importer will search the destination database for records that match the auto-placement information. It expects a single match from the database. An error will occur if there is more or less than one match.

Auto-placement settings are optional in workbooks used in MI:Toolbox and by the command-line interface, but must be present in workbooks used by Remote Import. When using MI:Toolbox or the command-line interface, auto-placement settings in the workbook take precedence over the location set by the user at the time of import.

The auto-placement options are specified by defining the following named ranges in the *Import Options* worksheet.

То	Use Named Range	See
Identify the database and table	CES_EXPORT_PLACEMENTDBKEY	Section 6.1.1
	CES_EXPORT_PLACEMENTTABLE	
Place the records in the target table	CES_EXPORT_PLACEMENTFORCETOROOT	Section 6.1.2
	CES_EXPORT_PLACEMENTSEARCHVALUE or	
	CES_EXPORT_PLACEMENTSEARCHATTRIBUTE	
	CES_EXPORT_PLACEMENTMATCHPROPERTY or	
	CES_EXPORT_PLACEMENTATTRIBUTE	
	CES_EXPORT_PLACEMENTISCHILD	
	CES_EXPORT_PLACEMENTCREATEIFNOTFOUND	
Place the incoming records in a specific subset	CES_EXPORT_FILTER	Section 6.1.3

Table 8. Auto-Placement opti	ions
------------------------------	------

When using auto-placement, additional options can be used to resolve duplicate record name conflicts; see Section 6.2.

6.1.1 Database and table placement options

These options set the database and the table that the new records will be imported into. If you set these two options, you must also set the placement options (in Section 6.1.2).

Table 9. Database and table placement options

Option	Notes
CES_EXPORT_PLACEMENTDBKEY	The name of the database key for the GRANTA MI database where the record will be created.
	In MI:Toolbox, the option is used to check that the same database has been selected by the user.
CES_EXPORT_PLACEMENTTABLE	The name of the table in the database into which the record will be imported.

6.1.2 Record placement options

Auto-placement can be used to place incoming data in the specified table in a range of different locations.

- The data may be placed into a new record at the top of the tree, using CES_EXPORT_PLACEMENTFORCETOROOT.
- Data may be placed into a specific location by searching and matching on existing records, which is done in a similar way to that of auto-linking (see Section 6.3). The differences are that the search occurs in the same table in which the record will be created, and must only return a single match.

When matching occurs on an existing record or folder, the data may be placed:

- Into the specified record, using CES_EXPORT_PLACEMENTISCHILD=false.
- Into a record directly below the specified record/folder, using CES_EXPORT_PLACEMENTISCHILD=true.

The CES_EXPORT_PLACEMENTISCHILD setting can be used to create **generic records** during import, as shown in Table 12.

If there is no existing record with a matching name, you can specify what should be done using the CES_EXPORT_PLACEMENTCREATEIFNOTFOUND option:

- To skip the import, set CES_EXPORT_PLACEMENTCREATEIFNOTFOUND=false.
- To create a new record, set CES_EXPORT_PLACEMENTCREATEIFNOTFOUND=true. The new record will be created in the folder specified in MI_TREE (Section 6.3).

Autoplacement can also be used to place incoming data records into a specified subset; this is specified by defining the CES_EXPORT_FILTER named range in the *Import Options* worksheet; see *6.1.3*.

Placement Option	Notes
CES_EXPORT_PLACEMENTFORCETOROOT	 Specifies whether to place the records being imported into the root of the table. If TRUE, then the data will be imported into a new record directly below the table level (at the top or root of the tree). (See Figure 18). If FALSE, then the option is ignored. The default setting is FALSE if the cell is blank. This option requires the database key and table to be set (DBKEY and PLACEMENTTABLE), and optionally can take account of a subset. This option is an alternative for the search options below, and if used, the search options will be ignored.
CES_EXPORT_PLACEMENTSEARCHVALUE or CES_EXPORT_PLACEMENTSEARCHATTRIBUTE	 These named ranges specify the search term to be used when searching and matching on an existing record. CES_EXPORT_PLACEMENTSEARCHVALUE specifies a value that will be used as the search term (see <i>Figure 18</i> and <i>Figure 19</i>). CES_EXPORT_PLACEMENTSEARCHATTRIBUTE refers to a single cell containing an <i>attribute name</i>, as specified in the <i>Attribute Lookup</i> worksheet. The value of the corresponding attribute in the <i>Data</i> worksheet will be used as the search term (see <i>Figure 20</i>). SEARCHVALUE and SEARCHATTRIBUTE are alternatives; both named ranges may be defined, but one must be blank. If both have values, then SEARCHVALUE takes precedence.
CES_EXPORT_PLACEMENTMATCHPROPERTY or CES_EXPORT_PLACEMENTATTRIBUTE	 These named ranges specify the keyword to be used when searching and matching on an existing record. CES_EXPORT_PLACEMENTMATCHPROPERTY specifies a record property, one of FULLNAME, TREENAME, RECORDID or GUID (see Figure 18). CES_EXPORT_PLACEMENTATTRIBUTE specifies the name of an attribute (see Figure 19 and Figure 20). MATCHPROPERTY and ATTRIBUTE are alternatives; both named ranges may be defined, but one must be blank.
CES_EXPORT_PLACEMENTISCHILD	 Specifies whether a new record will be imported as a child of the located record or the incoming data will replace that in the located record (see Table 12). If TRUE, then the data will be imported into a new record below the folder or record identified. If FALSE, then the data will be imported directly into the matched record or folder. The default setting is FALSE if the cell is blank.

Table 10. Record placement options

Placement Option	Notes
CES_EXPORT_PLACEMENTCREATEIFNOTFOUND	 Specifies what should be done with the incoming data if there is no existing record with a matching name: If FALSE, the data is not imported. If TRUE, the data is placed in a new record, created in the folder specified in MI_TREE.

Table 11.	CES	EXPORT	PLACEMENTISCHILD behaviour
	_		

Matched record type	CES_EXPORT_PLACEMENTISCHILD value	Outcome
Folder	FALSE	Matched folder becomes a generic record containing the incoming data.
Folder	TRUE	A new record containing the incoming data is created below the existing folder.
Generic record	FALSE	The matched generic record is updated with the incoming data.
Generic record	TRUE	A new record containing the incoming data is created below the existing generic record.
Record	FALSE	The matched record is updated with the incoming data.
Record	TRUE	Matched record becomes a generic record and a new record containing the incoming data is created below it.

Auto-placement examples:



Figure 17. Auto-placement options set to import records at the top level of the MaterialUniverse tree



Figure 18. Auto-placement options to search for a named record



7075. Search Value has been used, so Search

MaterialUniverse tree for a record named

Auto-placement options to search the

Attribute is blank.

Match Property has been used, so Match Attribute is blank.

Auto-placement options to search the MaterialUniverse records in the database for a record whose value for Density is 2770 kg/m^3.

Search Value has been used, so Search Attribute is blank.

Match Attribute has been used, so Match Property is blank.

Figure 19. Auto-placement options to search for a record with a specified density value



Data worksheet:

Young's Modulus	69	73	GPa
Bulk Modulus			GPa
Poisson's ratio	0.33	0.34	
Yield strength (elastic limit)			MPa
Tensile strength	380	520	MPa
Elongation	2	8	% strain

Auto-placement options to search the Young's Modulus attribute in the MaterialUniverse records in the database using the value for Young's Modulus given in the Data worksheet (below).

Search Attribute has been used, so Search Value is blank. Match Attribute has been used, so Match Property is blank

Figure 20. Auto-placement options to search for a record with a given attribute value

6.1.3 Subset placement

Use the CES_EXPORT_FILTER named range in the *Import Options* worksheet to place imported records into a specific subset.

- If the parent record is not in the named subset, the parent (and all its ancestors) will be added to that subset.
- The imported record will inherit subset membership from its parent in addition to the subset specified here.
- If the cell is blank, then the imported record will only be added to the same subsets as the parent record.
- If the named subset does not exist, the Excel Importer will give a warning message and the record/data will not be imported.



Figure 21. Auto-placement options set to import records into the 'All bulk materials' subset

6.2 Conflict resolution and merging options

When using auto-placement, if the Excel Importer attempts to create a new record with the same name as an existing record in the same folder, a record name conflict will occur.

Conflict resolution and merge options, specified in the *Import Options* worksheet, define the strategy for dealing with incoming data in this situation.

For tabular data, additional options in the template allow you to specify exactly what you want the Importer to do with the incoming data when merging.



See also *Section 7.2, Merging and appending*, for examples of appending and merging different types of data.

6.2.1 Conflict resolution strategy – MI_CONFLICTRESOLUTION

The MI_CONFLICTRESOLUTION option sets the strategy for dealing with the record and its data when there is a naming conflict between incoming data and an existing record in the database.

The named range must refer to a single cell containing one of the following values. The values are not case sensitive.

Table 12.	Valid values for MI	CONFLICTRESOLUTION
10010 121		

Value for MI_CONFLICTRESOLUTION	Description
DoNotImport	The record is not imported and the Excel Importer will move on to the next record in the list, or finish the import if this was the final or only record being imported.
ReplaceData	 The existing record is kept and the incoming data is added to the record. Where there is existing data, this will be replaced by the incoming data. Where there is no existing data, incoming data will be added. If there is no incoming data for an attribute, existing data remains unchanged.
	 There is no control over which data is replaced. If incoming links are included, then they will be added to the record. The existing links remain unchanged and duplicate links will not be created.

Value for MI_CONFLICTRESOLUTION	Description
Replace	 The existing record and all the links associated with the record are deleted. The new record is then created. No information from the existing record is preserved. This option is not available for generic records.
Merge	 The existing record is kept and the incoming data is added to the record. The behavior is the same as for the ReplaceData option, except for functional, tabular, and multi-value data types. Where there is existing multi-value data, the incoming data is merged. Where there is existing functional data, the <u>MI_FUNCTIONALMERGEOPTION</u> setting can be used to specify how that data will be merged. Where there is existing tabular data, additional options can be used to specify how tabular data rows should be handled; see 6.2.3. See also Section 7.2, Merging and appending, for examples of appending and merging different types of data
Append	 The existing record is kept and the incoming data is added to the record. The behavior is the same as for the ReplaceData option, except for functional, tabular, and multi-value data types. Where there is existing multi-value data, the incoming data is appended. Where there is existing functional data, the <u>MI_FUNCTIONALMERGEOPTION</u> setting can be used to specify how that data will be appended. Where there is existing tabular data, the incoming data is appended. Where there is existing tabular data, the incoming data is appended. See also Section 7.2, Merging and appending, for examples of appending and merging different types of data

If the MI_CONFLICTRESOLUTION cell is blank, duplicate record names will be treated as follows:

- In MI:Toolbox, the user will be offered a choice in the Duplicate Record Name dialog.
- When using the Excel Importer command-line interface, the setting from the command line is used. If there is no setting on the command line, the default behavior is **Merge**.
- When using Remote Import the default behavior is **Merge**.

6.2.2 Functional data merge/append options – MI_FUNCTIONALMERGEOPTION

If a duplicate record name strategy of Merge or Append is chosen, the Excel Importer will keep the existing record and attempt to combine the incoming functional data with the existing functional data. You can set the functional data merge or append strategy separately in the **Merging** section of the template:



Table 13. Valid values for MI_FUNCTIONALMERGEOPTION

Note that the functional data Merging option overrides the Conflict Resolution setting. For example, if Conflict Resolution = Merge:

- when Functional Data = TryAppendThenReplace, then the data will be appended
- when Functional Data = Replace, the data will be replaced.

Value for MI_FUNCTIONAL MERGEOPTION	Description
Replace	The existing functional data is deleted and the incoming functional data is added.
	No attempt is made to merge or append the incoming functional data with the existing data, even if the data is compatible.
TryMergeThenReplace	 If the incoming functional data is compatible with the existing functional data, it will be merged with the existing data, see Section 7.2.1. If the incoming functional data is not compatible with the existing functional data, the existing data is deleted and the incoming data is added.
TryMergeThenDoNotImport	 If the incoming functional data is compatible with the existing functional data, it will be merged with the existing data, see Section 7.2.1. If the incoming functional data is not compatible with the existing functional data, the incoming data is not added and the existing data is unchanged.
TryAppendThenReplace	 If the incoming functional data is compatible with the existing functional data, it will be appended to the existing data, see Section 7.2.2. If the incoming functional data is not compatible with the existing functional data, the existing data is deleted and the incoming functional data is added.

Value for MI_FUNCTIONAL MERGEOPTION	Description
TryAppendThenDoNotImport	 If the incoming functional data is compatible with the existing functional data, it will be appended to the existing data, see Section 7.2.2.
	 If the incoming functional data is not compatible with the existing functional data, the incoming data is not added and the existing data is unchanged.

If the cell is blank, the behavior of the Excel Importer will be as follows:

- In MI:Toolbox:
 - If the record name conflict strategy is **Merge** and the incoming functional data is compatible with the existing functional data, the default behavior is **Merge**.
 - If the record name conflict strategy is **Append** and the incoming functional data is compatible with the existing functional data, the default behavior is **Append**.
 - If the incoming functional data is not compatible with the existing functional data, the user will be offered a choice in the **Cannot Merge Data** dialog box.
- In the command-line interface
 - The setting from the command line is used.
 - If there was no setting on the command line, the default behavior is **TryMergeThenReplace**.
- In Remote Import the default behavior is **TryMergeThenReplace**.

6.2.3 Tabular data row merge options

If a duplicate record name strategy of Merge is chosen, the Excel Importer handles the incoming and existing tabular data rows according to the settings in the **Merging** section of the template:



Note that if **any** of the tabular row Merging options are set, **all** must be set, otherwise the import will terminate with an error, and no data will be imported. All options can be left blank, in which case the defaults as shown in *Table 14* will be used.

A tabular row is identified by its **linking value**. Therefore a 'duplicate' tabular row occurs when a row in the database and in the importer have the same linking value.

Option	Description
MI_TABULAR_DUPLICATE_ROW_MERGEOPTION	The strategy for dealing with duplicate tabular rows. May be set to:
	• Replace – the incoming tabular row will replace the existing tabular row.
	• Append – the incoming tabular row is appended.
	• DoNotImport – the new tabular row is not imported and the existing data is unchanged.
	The default behavior if no value is specified is Replace .
MI_TABULAR_ROW_NOT_IN_DATABASE_MERGEOPTION	What to do when a tabular row exists in the import template but not in the database. Value may be one of:
	• Append – the incoming tabular row is appended.
	• DoNotImport – the new tabular row is not imported and the existing data is unchanged.
	The default behavior if no value is specified is Append .
MI_TABULAR_ROW_NOT_IN_IMPORT_MERGEOPTION	What to do when a tabular row exists in the database but not in the import template. Value may be one of:
	• Delete – the existing data is deleted.
	• DoNotDelete – the existing data is unchanged.
	The default behavior if no value is specified is Delete .

Table 14. Values for tabular data row merge options

6.3 Placement of records into a specific folder hierarchy (Tree)

The Excel Importer can place the records being imported into a hierarchy of folders. The named range **MI_TREE** defines the tree hierarchy.

Named range	Notes
MI_TREE	The range must be <i>n</i> cells by 1 cell, or 1 cell by <i>n</i> cells only.
	Each cell may contain the name of an attribute or a text value. If it contains the name of an attribute, the attribute must also be listed on the <i>Attribute Lookup</i> worksheet. The data for the attribute on the <i>Data</i> worksheet(s) is used as the folder name in the tree.

Note that that attributes defined in the template to construct the tree do not have to match existing attributes in the database, provided that the 'Import' setting on the *Attribute Lookup* worksheet is set to 'FALSE' (see Section 3.7). If the attribute name is not listed on the *Attribute Lookup* worksheet, then it is treated as a text value, and the text value in the cell itself is used as the folder name in the tree.

Only one name value can be imported for folders in the tree; the folder's short name and full name are set to the same value.

Creating a record hierarchy can be combined with using auto-placement to locate the 'base' folder, for example the 'import example' folder in *Figure 25*. In order to do this, the 'CES_EXPORT_PLACEMENTISCHILD' named range must be set to TRUE (see Section *6.1.2*).

Note that if the value specified for the lowest level of the tree matches an existing record (rather than a folder), this record will become a **generic** record, with its data preserved, and the record being imported will become a child of the generic record.

Example using attribute data

The hierarchy is constructed using the attribute names in the named range on the *Import Options* worksheet (highlighted in *Figure 22*) and the data values for those attributes in the *Data* worksheet (columns B, C, D and E in *Figure 24*). The values for the attribute 'Alloy Class' are used to define Level 1, 'Alloy Name' is used to define Level 2, 'Heat Treatment' is used to define Level 3, 'Form' is used to define Level 4.

Tree			
	Level 1 Alloy Cl	ass	
	Level 2 Alloy Nar	me	
	Level 3 Heat Tre	atment	
	Level 4 Form		 ML TREE named range
	Level 5		
	Level 6		
	Level 7		
	Level 8		

Figure 22. Named range on the Import Options worksheet

In Figure 22, the named range MI_TREE on the *Import Options* worksheet contains attribute names. In the example, only four of the cells in the range are used.

	A	В	С	D	E	F	G	Н	
1	Attribute Name	DataType	Units	Parent Attribute	Worksheet	Specimen	Data Range	Quality Range	Import
2	Record Name	RCN			DataSheet	ROW	MI_NAME		
3	Density	PNT	lb/in^3		DataSheet	ROW	MI_DENSITY		
4	Alloy Class				DataSheet	ROW	MI_LEVEL1		FALSE
5	Alloy Name				DataSheet	ROW	MI_LEVEL2		FALSE
6	Heat Treatment				DataSheet	ROW	MI_LEVEL3		FALSE
7	Form				DataSheet	ROW	MI_LEVEL4		FALSE

Figure 23. Attribute Lookup worksheet

Figure 23 shows the corresponding *Attribute Lookup* worksheet, listing the attribute names to be used for constructing the tree with their **Import** setting set to **FALSE**.

	A	В	С	D	E	F		
1	MI_NAME	Level 1	Level 2	Level 3	Level 4	Density		
2		Alloy Class	Alloy Name	Heat Treatment	Form	lb/in^3		
3								
4								
5	MIL 3013	2000 Series	2014	Т6	Plate	0.1		
6	MIL 3013	2000 Series	2014	Т6	Sheet	0.1	Data used to	
7	MIL 3034-1	2000 Series	2024	T3	Sheet	0.1		
8	MIL 3034-2	2000 Series	2024	T3	Sheet	0.1		construct tre
9	MIL 3239	7000 Series	7075	173	Bar	0.1		
10	MIL 3009	7000 Series	7075	173		0.1		
11	Bar	7000 Series	7075	173		0.1		
12								
	1							

MI_LEVEL1 MI_LEVEL2 MI_LEVEL3 MI_LEVEL4

Figure 24. Data worksheet

Figure 24 shows the *Data* worksheet with the data for the attributes named in the *Import Options* worksheet (Figure 22). The named ranges used in the example are MI_LEVEL1, MI_LEVEL2, etc. The attribute names and hierarchy level are repeated on the worksheet for clarity.



Figure 25. Tree hierarchy in MI:Viewer

Figure 25 shows the tree hierarchy in MI:Viewer below the 'Import example' folder (excluding data records) resulting from the example data in *Figure 22* and *Figure 24*.

6.4 Auto-Links

The criteria given in the auto-linking options are used to search the database for matching records. A table , record property or attribute and a search value are specified in the template and the Excel Importer uses the supplied value to search the specified data in all records of the specified table. If an attribute is specified, the Excel Importer will search every value of the specified attribute in the table for a match. A property of the record can also be used, for example, the name of the record. In this case the Excel Importer will look for a match between the search value and the record name.

The auto-linking options are specified by using five columns in the *Import Options* worksheet. The data in these columns is used to identify the link group and to match record(s) in the destination

table to link to. Each row contains a set of criteria for making data links. A named range is defined in each column; see *Table 15*.

Auto-linking options for data link groups are specified by using three additional columns to define properties of the data link itself.

Table 15. Named ranges required for record and data link groups

Named ranges	Notes
CES_EXPORT_LINKCLUSTERS	The named range must refer to a single cell, or range of adjacent cells in a single column, containing the names of the data link groups in the database that the links will be created in. All added data links for an imported record do not have to belong to the same data link group.
CES_EXPORT_SOURCEVALUES	The named range must refer to a single cell, or range of adjacent cells in a single column, containing the values that will be used by the Excel Importer to search the specified table. Each cell will contain the value for a link. SOURCEVALUES and SOURCEATTRIBUTE are alternatives. Both named ranges may be defined, but one must be blank.
CES_EXPORT_SOURCEATTRIBUTE	The named range must refer to a single cell, or range of adjacent cells in a single column, containing an attribute name, as specified in the <i>Attribute Lookup</i> worksheet. The value for this attribute will be used to search the table that the data is imported into. <i>SOURCEVALUES and SOURCEATTRIBUTE are alternatives. Both</i> <i>named ranges may be defined, but one must be blank.</i>
CES_EXPORT_DESTINATIONPROPERTY	The destination table is the table that is to be searched for matching values and contains the records to be linked to. The named range must refer to a single cell, or range of adjacent cells in a single column, containing one of three values, which specifies a record property in the destination table. The possible values are: FULLNAME, TREENAME, RECORDID, or GUID. The Excel Importer will search this property of every record in the destination table and return a match if it is equal to the search value. DESTINATIONPROPERTY and DESTINATIONATTRIBUTE are alternatives. Both named ranges may be defined, but one must be blank.
CES_EXPORT_DESTINATIONATTRIBUTE	The named range must refer to a single cell, or range of adjacent cells in a single column, containing the name of one of the attributes in the destination table, that is, the database table to be searched. The Excel Importer will search through the values of this attribute for every record in the destination table and return a match where they are equal. DESTINATIONPROPERTY and DESTINATIONATTRIBUTE are alternatives. Both named ranges may be defined, but one must be blank.

Named ranges	Notes
CES_EXPORT_LINKSCRITERIA (data link groups only)	The named range must refer to a single cell, or range of adjacent cells in a single column, containing one of three values, which specifies if the target attribute of the data link group is required to contain data. The valid values are:
	 NoChecks. No checks on whether the target attributes contain data AtLeastOneDataRequired. At least one of the target attributes must contain data AllDataRequired. Each target attribute must contain data
CES_EXPORT_LINKCLUSTERATTRIBUTE (data link groups only)	The named range must refer to a single cell, or range of adjacent cells in a single column, containing the names of the primary attributes in the data link groups. The name of the primary attribute and the name of the data link group together uniquely identify the data link group.
CES_EXPORT_LINKSOURCETABLE (data link groups only)	The primary attribute of the data link group is assumed to be in the table that the data is imported into, <i>unless</i> the CES_EXPORT_LINKSOURCETABLE named range is used. The named range must refer to a single cell, or range of adjacent cells in a single column, containing the names of the tables that contain the primary attributes of the data link groups. Expressed another way, it contains the name of the table from which the data link group starts.

It is important to take care when defining these auto-linking named ranges that:

- All the defined named ranges are the same length as each other, that is, contain the same number of cells.
- All rows containing data are included in these named ranges.

If the values of these defined named ranges are blank, then auto-linking will not be activated during import.

If no match is made, then no link will be made. If there is more than one record or folder matching the linking criteria, a link will be created for each.

Auto-linking option example

Auto-Links					
	Link Group	Search Value	Search Attribute	Match Attribute	Match Property
Link 1	Materials Pedigree	Maraging Steel			TREENAME
Link 2	Components	5370045		Batch Number	
Link 3	Components		Component Code	Batch Number	
Link 4	Components	5370045			TREENAME

Figure 26. Example of Auto-linking options in the Import Options worksheet

The first set of links (Link 1 in *Figure 26*) will be created in the 'Materials Pedigree' record link group. The record is being imported into the source table of the 'Materials Pedigree' record link group. The Excel Importer searches the short name (the name that appears in the tree) of each record in the destination table of the 'Materials Pedigree' record link group, for the value 'Maraging Steel'. If it finds a match, a link is made between the imported record and the matched record. If it finds more than one match, a link is created for each match, so multiple links may be created.

The third set of links (Link 3 in *Figure 26*) will be created in the 'Components' record link group. The record is being imported into the source table of the 'Components' record link group. (This table is also the source table of the 'Materials Pedigree' record link group.) The Excel Importer takes the value for the 'Component Code' attribute in the imported record and searches the 'Batch Number' attribute of each record in the destination table of the 'Components' record link group for the same value. If it finds a match, a link is made between the imported record and the matched record. If it finds more than one match, a link is created for each match, so multiple links may be created.

6.5 Access Control

The Excel Importer can set read/write permissions at the record level during import, for databases with an existing permission-based access control schema. Note that this is only available if the user importing the data has grant or administrative privileges for the database.

Access control permissions can be specified by defining the following named ranges in the *Import Options* worksheet. All three ranges must be of the same length (see *Figure 27*).

Named range	Notes
MI_AC_NAME	The named range must refer to a single column.
	Each cell contains the name of a permission in the access control schema.
MI_AC_READ	Each named range must refer to a single column.
MI_AC_WRITE	Each cell can contain the value TRUE or FALSE to set the permission directly. Alternatively, each cell can contain the name of an 'attribute' in the <i>Attribute Lookup</i> worksheet, and the data for that attribute is used to set the permission. (The attribute itself can be set to not be imported, see Section 3.7). This allows different access control settings on different records if importing multiple records from the same file.
	If an access control permission is not defined in the list, or either the READ or WRITE cells are blank, then the access control permission is set according to the access control inheritance rules.

Table 16. Named ranges for access control permissions

Access control is set at the record level, not the data level. If access control permissions are not set explicitly, they are inherited from the permissions of the 'parent' object at the time of import.

Overwriting existing records/data

If a record is being created or replaced, or data is being created, then the access control settings of the incoming data (as set in the template), are used. If data is being replaced/merged/appended, then the access control settings of the existing data are used.

	Name	Read	Write
Permission 1	US	US only?	FALSE
Permission 2	Non-US	Non-US?	FALSE
Permission 3	M&P	Test Data?	TRUE
Permission 4	Designers	Design Data?	FALSE
Permission 5			
Permission 6			
Permission 7			
Permission 8			
Permission 9			
ermission 10			

MI AC NAME MI_AC_READ MI_AC_WRITE

Figure 27. Access control permission settings on the Import Options worksheet

In this example, the **Name** settings are the names of access control permissions in the database access control schema. The **Read** settings are the names of entries in the Attribute Lookup worksheet in this case, whereas the **Write** settings are TRUE or FALSE.

6.6 Valid tables

When auto-placement options are not being used, the CES_EXPORT_VALIDTABLES named range in the *Import Options* worksheet is used identify the tables into which records may be imported.

The named range must refer to a single cell, or range of adjacent cells in a single column, containing the names of the tables that the data can be imported to.

Valid Table Table Table Table	CES_EXPORT_VALIDTABLE
--	-----------------------

Figure 28. Valid table options in the Import Options worksheet

The option is also used in MI:Toolbox when loading valid templates in the user interface, if the template workbook is separate to the data workbook.

6.7 Exported Worksheet Naming

MI_SHEETNAME_FORMAT controls the format of worksheets created when exporting data using the template. See the *GRANTA MI Excel Exporter Reference Guide* for more information.

7 Advanced features

7.1 Creating multiple records

The Excel Importer has the ability to create more than one record from a single workbook. The individual records can be read either as a single record per row, a single record per column or a single record per worksheet. This is specified in the *Attribute Lookup* worksheet in the **Specimen** column (see Section 3.7).

- **Row** creates a single record from each row in a worksheet.
- Column creates a single record from each column in a worksheet.
- Worksheet creates a single record from each worksheet in the workbook.

To use this feature, **Specimen** must be set for the *Record Name* attribute. If the *Record Name* does not have a setting, a setting cannot be used by any other attribute (see Figure 29).

If the 'Row' and 'Worksheet' settings are used in combination, then the number of worksheets must equal the number of rows and vice versa.

Similarly, if the 'Column' and 'Worksheet' settings are used in combination, then the number of worksheets must equal the number of columns.

The Record Name data and Specimen type are used to determine the number of records being imported. The Excel Importer will then expect to find data in that many rows, columns, or worksheets, depending on which is used.

P	GrantaMLExport Test2.	.xls						
	A	В	С	D	E	F	G	Н
1	Attribute Name	Parent Attribute	DataType	Units	Specimen	Worksheet	Data Range	Parameter 1 Range
2	Record Name		RCN		Row	Data	CES_RECORDNAME	
3	Range Attribute		RNG			Data	CES_RANGEDATA	
4	Notes	Range Attribute	DCT			Data	CES_RANGEMETA	
5	Point Attribute		PNT			Data	CES_POINTDATA	
6	Integer Attribute		INT			Data	CES_INTEGERDATA	
7	Short Text Attribute		TXT			Data	CES_SHORTTEXTDATA	
8	Long Text Attribute		TXT		Row	Data	CES_LONGTEXTDATA	
9	Discrete Attribute		DCT		Row	Data	CES_DISCRETEDATA	
10	Logical Attribute		LOG			Data	CES_LOGICALDATA	
11	Date Attribute		DAT			Data	CES_DATEDATA	
12	Functional Point		FDP			unctional Data	CES_FUNCDATA_VALUE	CES_FUNCDATA_PA
13	Functional Point 2		SDP			unctional Data	CES_FUNCDATA_VALUE	CES_FUNCDATA_PAR
14	Functional Point 3		SDP			unctional Data	CES_FUNCDATA_VALUE	CES_FUNCDATA_PAR
15	Functional Range		SDR			unctional Data	CES_RANGEFUNCDATA_VALUE	CES_FUNCDATA_PA
16	Functional Point 4		SDR			unctional Data	CES_RANGEFUNCDATA_VALUE	CES_FUNCDATA_PAR
17								
	🕩 🕨 Data 🖉 Functio	nal Data Attribute Lo	okup / Parame	eter Lookup	7			

Figure 29. The Specimen column in the Attribute Lookup worksheet

Note that if the Specimen column has no value for the Record Name property, the named range to which the Record Name is mapped must be populated, otherwise Toolbox will report an error, reporting that it can't read the record name.

In a multi-table import, where some sequence steps are optional, without a Specimen setting, record names would be expected for those sequence steps. In this case, if you want a particular import step to be optional, set Specimen to 'Row' for Record Name. This will then be skipped over if no value for the record name is found

7.1.1 Row data

A new record will be created for each row in the *Data* worksheet. The column containing the record name in the *Data* worksheet is determined by the information in the 'Record Name' row in the *Attribute Lookup* worksheet. This column will also be used to determine the number of records to create in the database. The Excel Importer will process the data until it encounters a blank cell in the record name column.

For multiple row data, only the first cell in each column needs to be defined as the named range, thus allowing the number of rows to vary between worksheets. This provides the Excel Importer with the column and first row for the attribute data. Attribute data cells can be blank.

	Α	B	С	D	E	F	
1							
2		Specimen Code	Temperature	Diameter	Youngs Modulus	elongation	
3		Specimen code	deg C	mm	GPa	%	
4		A1-X1	25	10	120	5	
5		A1-Y1	200	10	125	8	
6		A1-Z1	300	10	130	12	
7		A1-X2	400	10	125	15	
8		A1-Y2	500		110	20	
9		A1-Z2	600	10	100	22	
10		A2-X1	700	10	90	25	
11		A2-Y1	25	10	120	6	
12		A2-Z1	200			8	
13		A2-X2	300	10	120	12	
14		A2-¥2	400	10	125	14	
15							
16		A2-Z2	500	10	110	18	
17							
18							

Figure 30. Example of data in rows. Each row will form a new database record.

In Figure 30, the Specimen Code is used to name the record (column B).

- Rows 4 to 14 will be imported.
- Row 16 will not be imported due to the empty cell B15.
- Record 'A2-Z1' on row 12 will be imported but will not have any data for the 'Diameter' and 'Young's Modulus' attributes.
- In the columns containing data (C F), the named range should be set to the cell in row 4 only. It is not necessary to define the range as, for example, C4–C14.

To import tabular data with "Specimen = Row"

On the Attribute Lookup worksheet

- 1. Add a 'Specimen Number' column.
- 2. In the Specimen Number cell for the tabular attribute, enter the name of a single-cell named range, defined in the *Data* worksheet.

On the Data worksheet

- 3. Add a 'Specimen Number' column.
- 4. Define the named range for Specimen Number as a single cell in the first row of data in the 'Specimen Number' column.
- 5. As the Excel Importer reads the tabular data, it will read the number of the specimen (imported record) from the 'Specimen Number' column and use this to determine which Excel rows contain data for the same record.

6. Other (non-tabular) attributes must be spaced vertically according to the number of Excel rows used for the tabular data. The data ranges for these should ideally be horizontally aligned with the data and parameter ranges for the tabular attribute, so that there is a rectangular area for each specimen (record) imported.

7.1.2 Column data

A new record will be created for each column in the *Data* worksheet. The row containing the record name in the *Data* worksheet is determined by the information in the 'Record Name' row in the *Attribute Lookup* worksheet. This row will also be used to determine the number of records to create in the database. The Excel Importer will process the data until it encounters a blank cell in the record name row.

For multiple column data, only the first cell in each row needs to be defined as the named range, thus allowing the number of columns to vary between worksheets. This provides the Excel Importer with the row and first column for the attribute data. Attribute data cells can be blank.

7.1.3 Worksheet data

A new record will be created for each *Data* worksheet in the workbook. This works in a similar way as the normal Excel templates. There will be a master *Data* worksheet that should have exactly the same name as the name in the 'Worksheet' column in the *Attribute Lookup* worksheet. The ranges in this master *Data* worksheet are then mapped onto the other *Data* worksheets that are to be imported. The Excel Importer uses the worksheet name to identify which data to import. Any worksheet that contains the master worksheet name as part of its name is treated as a *Data* worksheet.

In Figure 31, the master *Data* worksheet containing all the named range information is named 'Data Fit'. Any worksheet name containing 'Data Fit' will be imported. Therefore, the Excel Importer will create three new records in the database for the example shown.

e) F	unctionalImport.xls						
	A	В	С	D	E	F	G
1							
2	Record Name	MTS-572221					
3	Test Temperature	810.9278					
4	0.2% Offset Yield Stress	289.6926					
5	Batch Number	Batch 332					
6							
/	Strace	Strain	DataTyno				
9		Suam	Tyne A				
10	11 71679688	0.004488357	T DO TO D				
11	14.79980278	0.005981984					
12	17.26604652	0.007473742					
13	20.3490448	0.008967368					
14	22.81528854	0.01045913					
15	25.89828873	0.011952756					
16	28.981287	0.013446383					
17	32.06428528	0.01494					
18	34.53052902	0.01643176					
19	37.61353683	0.017925387					
20	40.6965332	0.019419014					
21	43.16277695	0.020910775					
22	46.24577713	0.022404402					
23	48.71202087	0.023896152					
24	51.79502106	0.025389779					
25	54.2612648	0.02688154					
26	57.3442688	0.0283/516/					
27	59.81051254	0.029866928					
20	62.09351273	0.031360555					
29	68 44275555	0.032004173					
	00.44275005	Data Fit 🖌 Data Fit 1 🕽 🖸	ata Fit2 🖉 🗍	•			

Figure 31. Example of data in multiple worksheets. Each worksheet will become a new database record.

7.2 Merging and appending

The Excel Importer can merge or append functional data and multi-value data, and append, replace or delete tabular data. Information on how to set this behavior is given in Section 6.2. You can also use the merge/append capability to delete existing data values (see Section 7.3).

7.2.1 Merging functional data

Incoming functional data must be compatible with existing functional in order to be merged. See Section **7.2.3** for details of importer behavior when incoming and existing functional data are incompatible.

Merging series functional data and discrete functional data

- Incoming series with different parameter values (for the non-varying parameter) to the existing data are added.
- Existing series with the same parameter values (for the non-varying parameter) as the incoming data are replaced.
- Existing series with different parameter values (for the non-varying parameter) to the incoming data are kept.

Series example:

If an existing chart has two stress-strain curves at 300K and 400K, in this case the non-varying parameter is temperature. When a chart with stress-strain curves at 273K and 300K is imported, then the old 300K curve will be replaced with the incoming 300K curve and the 273K curve will be added to the existing curves. The existing 400K curve is kept. The result of the merge is a chart with three stress-strain curves, at 273K, 300K and 400K (see Figure 32).



Figure 32. Example of merging series functional data

When merging series data, the incoming data must have the same parameters as the existing data. In the above example where the existing chart has two stress-strain curves at 300K and 400K, it is not possible to add a series without a temperature value, or add a series with temperature and test-type values.

Merging grid functional data

- Incoming grid values with different parameter values to the existing data are added.
- Existing grid values with the same parameter values as the incoming data are replaced.
- Existing grid points with different parameter values to the incoming data are kept.

Grid example:

- The example two dimensional grid has grid values for parameter X=1, X=2, X=3 and parameter Y=55, Y=57, Y=59.
- Grid values for parameter X=4 are added.
- Grid values for (X=3, Y=55) and (X=2, Y=57) are replaced.
- All other grid values are kept (see Figure 33).

	Existi	ng da	ata		In	Incoming data				Data after Merge						
F (X, Y)	55 Y 57 55	1 5 111 7 115 9 118	X 112 116 119	3 113 117 120	Y	55 _ 57 _ 59	1	X 2 118	3 116	4 118 123 126	Y	55 57 59	1 111 115 118	X 112 118 119	3 116 117 120	4 118 123 126

Figure 33. Example of merging grid functional data

When merging grid data, the incoming data must have the same parameters as the existing data. In the above example, it is not possible to add a grid value without a X parameter value, or add a grid value with X, Y and Z parameter values.

Merging Equations and Logic data

Incoming equations and logic data replaces existing data.

7.2.2 Appending functional data

Incoming functional data must be compatible with existing functional data in order to be appended. See Section **7.2.3** for details of importer behavior when incoming and existing functional data are incompatible.

Appending series functional data and discrete functional data

- Incoming series with different parameter values (for the non-varying parameter) are added.
- Incoming series with the same parameter values (for the non-varying parameter) are added.
- Existing series with the same parameter values (for the non-varying parameter) as the incoming data are kept.
- Existing series with different parameter values (for the non-varying parameter) to the incoming data are kept.

For example, if an existing chart has two stress-strain curves at 300K and 400K, in this case the nonvarying parameter is temperature. When a chart with stress-strain curves at 273K and 300K is imported, the old 300K curve is kept and the incoming 300K curve is added. The incoming 273K curve is added to the existing curves. The existing 400K curve is kept.

The result of the merge is a chart with four stress-strain curves, one at 273K, two at 300K and one at 400K (see Figure 34).



Figure 34. Example of appending series functional data

When appending series data, the incoming data must have the same parameters as the existing data. In the above example where the existing chart has two stress-strain curves at 300K and 400K, it is not possible to add a series without a temperature value, or add a series with a temperature and test-type parameter.

Appending grid functional data

Appending grid functional data has the same result as a merge.

- Incoming grid values with different parameter values to the existing data are added.
- Existing grid values with the same parameter values as the incoming data are replaced.
- Existing grid points with different parameter values to the incoming data are kept.

See Section 7.2.1 on Grid functional data for an example.

Appending Equations and Logic data

Appending equations and logic functional data has the same result as a merge, that is, incoming equations and logic data replaces existing data.

7.2.3 Incompatible functional data

If the incoming functional data is **not** compatible with the existing functional data then there are two possible actions.

- Replace The existing data is deleted. The incoming data is then imported.
- Do Not Import The incoming data is not imported and the Excel Importer will move on to the next data item to be imported.

Which action is taken depends on the MI_FUNCTIONALMERGEOPTION option that was set in the template (see Section 6.2.3).

7.2.4 Merging tabular data

Tabular data cannot be merged as such, its rows can only be appended, replaced or deleted. See Section 6.2.3 for details of the options for handling tabular rows when the record name conflict resolution of Merge is in use.

7.2.5 Appending tabular data

All tabular rows in the importer will be added to the existing tabular rows in the database. This may result in duplicate tabular rows (i.e. more than one row with the same linking value in the data for a tabular attribute).

7.2.6 Merging multi-value data

Merging multi-value discrete data

- Existing values are kept.
- Incoming values that are different from the existing values are added.
- Duplicate values will not result from a merge.

These examples show the results of merging multi-value discrete data for the 'Product Form' attribute:

		L	ncoming data	Existing data							
	🖌 J K L							J		К	
2	2 Product Form					2	Produ	ct Form			
3	Extrusion		Rod	Shapes		3	Bar		Extrusio	on	
	After the Merg						on			_	
	J K					L		M			
	2 Product Form										
	3 Bar Extrusion		Rod			Shapes					

Result as displayed in MI:Viewer datasheet

Product Form

Bar, Extrusion, Rod, Shapes

Merging multi-value point data:

- Existing values are kept.
- Incoming values that are different from the existing values are added. The combination of attribute value plus parameter values is considered as 'one' multi-value.
- Duplicate values will not result from a merge.

These examples show the results of merging multi-value data for the 'Tension strength' attribute

	Incomin	g data	_	Existing data					
	C	D			С	D			
6	C Tonsion strongth	U		6	Tension strength				
-	rension suengui	1 (1 (1 (1 (1 (1 (1 (1 (1 (1 (1 (1 (1 (1		7	Basis	Values (MPa)			
-	Basis	Values (MPa)		8	Mean	456.4			
8	Mean	456.4		9	A-basis	201.2			
9	A-basis	390.4		- 10	A-Dasis	301.3			
10	B-basis	411.6		10	B-basis	411.6			

After the Merge operation

	С	D						
5	Mechanical properties							
6	Tension strength							
7	Basis	Values (MPa)						
8	Mean	456.4						
9	A-basis	381.3						
10	B-basis	411.6						
11	A-basis	390.4						

Result as displayed in MI:Viewer datasheet:

Tension strength	456.4	MPa (Mean)
	381.3	MPa (A-basis)
	411.6	MPa (B-basis)
	390.4	MPa (A-basis)

7.2.7 Appending multi-value data

Appending multi-value discrete data

- Existing values are kept.
- Incoming values are added.
- Duplicate values can result from an append.

These examples show the results of appending multi-value data for the 'Product form' attribute:

	Inc	coming data			Existin	g data		
	🖌 J K L					J	К	
2	Product Form				2	Product Form		
3	Extrusion	Rod	Shapes		3	Bar	Extrusion	

After the Append operation

2 Product Form Extrusion Rod Shapes 3 Bar Extrusion Rod Shapes			J	К	L	М	N
3 Bar Extrusion Extrusion Rod Shapes	2	2	Product Form				
	3	3	Bar	Extrusion	Extrusion	Rod	Shapes

Result as displayed in MI:Viewer datasheet

Product Form	Bar, Extrusion, Extrusion, Rod, Shapes

Appending multi-value point data

- Existing values are kept.
- Incoming values are added. The combination of attribute value plus parameter values is considered as 'one' multi-value.
- Duplicate values can result from an append.

These examples show the results of appending multi-value point data for the 'Tension strength' attribute:

	Incomin	g data		Existin	ig data	
	С	D		С	D	
6	Tension strength	6	Tension strength			
7	Basis	Values (MPa)	7	Basis	Values (MPa)	
8	Mean	456.4	8	Mean	456.4	
9	A-basis	390.4	9	A-basis	381.3	
10	B-basis	411.6	10	B-basis	411.6	

After the Append operation

С	D			
Mechanical properties				
Tension strength				
Basis	Values (MPa)			
Mean	456.4			
A-basis	381.3			
B-basis	411.6			
Mean	456.4			
A-basis	390.4			
B-basis	411.6			
	C Mechanical prope Tension strength Basis Mean A-basis B-basis Mean A-basis B-basis	CDMechanical propertiesTension strengthBasisValues (MPa)Mean456.4A-basis381.3B-basis411.6A-basis390.4B-basis411.6		



Tension strength	456.4	MPa (Mean)
	381.3	MPa (A-basis)
	411.6	MPa (B-basis)
	456.4	MPa (Mean)
	390.4	MPa (A-basis)
	411.6	MPa (B-basis)

7.3 Deleting existing data

When using the Merge, Append, or ReplaceData conflict resolution strategies, existing data can be deleted by setting the data value being imported to [-DELETE-].

For complex data types, this means setting the first (top left) cell of the Data Range to [-DELETE-].

For range values, if the maximum or minimum value is set to [-DELETE-], then both values are deleted. For multi-value attributes, if one [-DELETE-] value is imported, then all values are deleted.

Note that deleting the data value is not the same as setting the data value to 'Not Applicable'.

7.3.1 Tabular data example

Suppose the following data is imported:

	A	В	С	
1	RCSN	Density	RefCode	
2	Record 1	1500	10	
3	Record 2	1200	15	
4	Record 3	2000	25	
5				

A mistake is discovered and the following data is re-imported:

	A	В	С	
1	RCSN	Density	RefCode	
2	Record 1			
3	Record 2		20	
4	Record 3		[-DELETE-]	
~				

When the 'Merge' option is selected for the duplicate record name conflict resolution strategy, the resultant data in the records is:

	A	В	С	
1	RCSN	Density	RefCode	
2	Record 1	1500	10	
3	Record 2	1200	20	
4	Record 3	2000		
E				

Tabular rows may deleted using the MI_TABULAR_ROW_NOT_IN_IMPORT_MERGEOPTION setting; see Section *6.2.3*.

7.3.2 Metadata

Metadata is treated as follows:

- Any existing metadata associated with a datum which is imported with the [-DELETE-] option will also be deleted.
- If metadata exists in the import template, it will be imported.
- Order matters. If metadata appears in the Attribute Lookup sheet before its parent (which is getting deleted), then the metadata will not be imported. This is because it gets imported before its parent gets deleted, and in the following step when its parent gets deleted, the metadata also gets deleted.

7.4 Template workbooks

It is possible to use the Excel Importer to import data from an Excel workbook that does not contain any of the required template information. The data workbook is imported in conjunction with a separate, template workbook. This allows multiple data workbooks to be imported using the same template. You might use this feature to import legacy data that was stored in a consistent format.

For a workbook to be imported in this way, the data workbooks must match the template workbook. The data for an attribute must always appear in the same location (row and column) for each workbook that is to be imported. The Excel Importer will read the named ranges from the template workbook and apply them to the data workbook.

The template workbook is created in exactly the same way as a normal workbook for import, and can take advantage of the same features, that is, with *Attribute Lookup*, *Parameter Lookup*, *Import Options* worksheets and the ability to create multiple records. The *Data* worksheet within the template workbook should be constructed in the same way as a normal workbook for import, that is, with named ranges. However, the data from this worksheet will not be read, it will be used as template for where to find data in the separate 'data only' workbook.

A template can be assigned to a specific table or set of tables. If the CES_EXPORT_VALIDTABLES named range is set in the *Import Options* worksheet (see Section 6.6), then the template can only be used to import records into a database table whose name corresponds to one of the table names in the template.

MI:Toolbox requires the folder locations of template workbooks to be specified before templates are visible to the user to select. See Section 8 for further details.

The command-line interface requires a command-line argument to specify the location of the template workbook to be used, see *Appendix B, Excel Importer CLI*.

Template workbooks can be uploaded from any location to Remote Import at the time of import, and can also be stored centrally by administrators for users to select from.

8 Template locations

By default, the Excel Importer looks for templates in a Templates folder located with the plug-in. For example:

C:\Program Files\Granta\GRANTA MI\Toolbox\plugins\Importers\Excel\Templates

This location is shared with all users of the computer on which MI:Toolbox is installed.

Each user can also specify additional, local template folders in Step 3 on the Excel Importer page in MI:Toolbox plug-in.

Excel Importer templates must have a .xls, .xlsx, or .xlsm file extension.

Appendix A. Setup for multi-step imports

Sections 2 through 6 describe how to design templates that can be used to import data into a single table, by defining data worksheet(s) plus Attribute Lookup, Import Options, and Parameter Lookup worksheets.

It is also possible for the Excel Importer to import data from one workbook into more than one table and into tables in different databases. This is done by defining a number of import steps in an **Import Sequence** worksheet.

For each step defined in the import sequence, separate Attribute Lookup, Import Options, and Parameter Lookup worksheets should be defined. Each of these worksheets should be named to reflect its content, for example, *Materials Attributes, Materials Parameters, Materials Import Options, Samples Attributes, Samples Import Options*. The worksheet is then associated with a specific import step by referencing it on the Import Sequence worksheet.

Each Import Options worksheet must define its own auto-placement settings. The table and database to import data into is defined by these auto-placement settings, therefore each Import Options worksheet within the workbook can specify a different table and/or database.

A.1 The Import Sequence worksheet

A workbook that defines a multi-step import must include a worksheet called **Import Sequence** that defines the import steps, and identifies the Attribute Lookup, Import Options, and Parameter Lookup worksheets for each step.

MultiStepImportExample.xlsm									e 23		
	A		В	С	D	E	F	G	Н	1	J
1	Step		Attribute Lookup	Import Options	Parameter Lookup						
2	Materials		Materials Attributes	Materials Import Options	Materials Parameters	Column	A defines	the step	os;		=
3	Samples		Samples Attributes	Samples Import Options		Columns	B-D ide	ntify the	Attribute	Looku),
4						Import O	ptions, a	nd Parar	neter Loo	okup	
5		The e	order of rows on this w	vorksheet		workshe	ets asso	ciated wi	th each s	step	
6		defin	e the import sequence	e steps							-
H.	🕩 H 🛛 Impo	ort Sequer	ice Materials Attributes	Materials Import Options	Materials Parameters	Samples Attr	butes 🦯	Samples Im	oort Options		

- The Step column defines an information name for each step. The import steps are performed in the order in which they are listed here, so in this example, it is possible for links from Samples records to the Materials records to be created during import, as the Materials data will be imported first.
- 2. Because multi-step imports must use auto-placement, an Import Options worksheet is required for each step.

Appendix B. Excel Importer CLI

The Excel Importer can be run as a scheduled task using a command line interface (CLI). The Excel Importer is executable is located in the bin folder of the MI:Toolbox installation folder, typically:

C:\Program Files\Granta\Granta MI\Toolbox\bin\

To run it from the command line, at the command prompt, type the following:

MIToolbox.Console.exe -plugin Granta.ExcelImporter

The Excel Importer requires several command line arguments to provide the information that is supplied by the user when using the plug-in inside MI:Toolbox. These include:

- The plug-in name, Granta.ExcelImporter
- The name of the server where the GRANTA MI service is running.
- The authentication details used to connect to the GRANTA MI Service (if System authentication is not used). The user will require at least write privileges to the database.
- The location of the data source file(s). Excel *.xls, *.xlsx and *.xlsm formats are supported.
- If auto-placement is not being used to specify where the incoming data will be placed, you will need to specify the destination database and table. If using auto-placement, it is not necessary to specify a database or table on the command-line.

Optional command-line arguments may also be specified:

- The path to an Excel template that will be used to import the data files into the database; this template may specify auto-placement options.
- The location in the tree hierarchy where the new records will be placed.
- Options specifying how to deal with naming conflicts between records.

The MI:Toolbox application should be closed before running a plug-in from the command line.

B.1 Command-line arguments

Arguments containing spaces, colons, or equals characters should be double quoted e.g.

-importlogfile "C:\my logs\importprocess.log"

-plugin Granta.Excellmporter

(Required) The name of the plug-in.

-server <servername>

(Required) The name of the server where the GRANTA MI Service is hosted.

-username < username >

The user name required to access the GRANTA MI Service. May be omitted if Windows authentication is being used for GRANTA MI; in this case, the import is performed with the user account used to log on to the computer.

If used, you must also supply the corresponding password/domain.

-password <password>

The password required to access the GRANTA MI Service. This must be included if Windows authentication is not used.

-domain <domainname>

The Windows domain for the user name and password.

-dbkey <dbkey>

(*Required when auto-placement is not used*) The database key of the database into which the data will be imported.

If the Excel import template includes autoplacement options, the database key does not need to be specified on the command line; if it is specified, it will be ignored.

-table <tablename>

(*Required when auto-placement is not used*) The name of the table into which the data will be imported.

If the Excel import template includes autoplacement options, the table does not need to be specified on the command line; if it is specified, it will be ignored.

-sourcedata <folder>

(*Required*) When importing data from multiple files, this specifies the name and location of a folder containing the source data files. The Excel *.xls, *.xlsx and *.xlsm formats are supported. When importing data from a single file, you can use the **-sourcefile** argument instead;

see -sourcefile.

-sourcefile <filepath>

(*Required*) When importing data from a single file, this specifies the file containing the source data to be imported. The Excel *.xls, *.xlsx and *.xlsm formats are supported.

When importing data from multiple files, use **-sourcedata** instead to specify the folder containing the source files – see *-sourcedata*.-sourcedata

-template <template>

(*Optional*) The path to an Excel template that will be used to import the data into the database. For example:

-template C:\Users\Public\Templates\test1.xlsx

-record < recordlocation | recordGUID >

(*Optional*) Identifies a record location in table's tree hierarchy, under which the new records will be placed. If this option is not specified, then new records will be placed at root record level.

• *recordlocation* is record at the end of a series of continuous records in the table's tree hierarchy, starting with the name of a top-level record and ending in the name of the chosen record. For example:

-record Metals Non-ferrous Aluminum Wrought "7000 Series"

recordGUID may be specified as 32 continuous digits, or as groups of 8, 4, 4, 4, and 12 digits seprated by hyphens. For example:

-records 00000026-000e-4fff-8fff-dd92ffff0000

-subset <subset1> [<subset2>...<subsetN>]

(Optional) A list of the subsets to which the imported records should be added.

-conflictresolution < Replace | ReplaceData | Merge | Append | DoNotImport >

(*Optional*) Specifies how record name conflicts on the tree should be dealt with. The values are not case sensitive. The default behavior is to Merge.

See Section 6.2, Conflict resolution and merging options for details of these options.

-functionaldatamerge < Replace | TryMergeThenReplace | TryMergeThenDoNotImport | TryAppendThenReplace | TryAppendThenDoNotImport | AppendDoNotImport >

(*Optional*) Specifies how functional data is handled when there is existing data. This is only applicable if the conflict resolution value is **Merge** or **Append**. The values are not case sensitive. The default behavior is **TryMergeThenReplace**. See Section **7**.2, Merging and appending, for details of these options.

-help

(*Optional*) Displays usage information. If the plug-in name is also specified, it will include the usage information of the plug-in. For example:

MIToolbox.Console.exe -plugin Granta.ExcelImporter -help

-logfile <path>

(Optional) Log file name and location. If omitted, no log file will be generated.

-loglevel < ERROR | WARN | INFO | DEBUG | TRACE | ALL >

(*Optional*) Sets the verbosity of logging sent to the command window console. Note that this setting has no effect on the contents of the log file. Permitted values (in order of increasing verbosity) are: ERROR, WARN, INFO, DEBUG, TRACE, ALL (case-insensitive). If omitted, the default value is INFO.

-nologo

© Granta Design 2018

(Optional) Prevents the display of application information on startup.

B.2 Usage example

```
C:\Program Files\Granta\Granta MI\Toolbox\bin\MIToolbox.Console.exe
-plugin Granta.ExcelImporter -logfile "C:\Users\kim.smith\Documents\Console.log"
-nologo -server localhost -username "kim.smith"
-password MyPassword -domain ACMECO -dbkey testdb
-table "Tensile Test Data" -conflictresolution Replace
-template "C:\Program Files\Granta\GRANTA
MI\Toolbox\plugins\Importers\Excel\Templates\Import Single Tensile Test
(Metric).xls" -sourcedata "C:\mydata\Excel"
```
Appendix C. Troubleshooting

If a problem occurs with the data being imported, The Excel Importer will continue to the next item, except in the case of problems with access control settings.

Problem with	Action	What happens to problem item
Creating the record	Continues to next record	Current record is not created
Applying access control	Stops import process	Current record is not created
Creating attribute data	Continues to next attribute data	Problem data is not imported
Creating links	Continues to next record	Current record is not created

If the import process stops, the records imported up to that point are kept.

C.1 Exit codes

The following MI:Toolbox exit codes may help you to troubleshoot problems that may arise when running the Excel Impoter from the command line.

Code	Description
0	Success - Indicates successful execution of the MI:Toolbox console application.
1	Unspecified error - Indicates that an unspecified error was encountered.
2	Unhandled exception - Indicates that an unhandled exception was caught.
3	Invalid command-line option value - Indicates a problem with one of the specified command-line options (e.g. an option specifies a table or record that does not exist).
4	Syntax error in command-line option - Indicates a basic syntax error with one of the specified command-line options.
5	Could not load plugin - Indicates that the named plugin could not be loaded.
6	Plugin initialization error - Indicates an error during the initialization phase of the plugin, typically caused when trying to validate an option against data in a database (e.g. a database specified by a particular dbkey may not exist).
7	Cannot open the log file - Indicates that the log file could not be opened.
8	Plugin found no work to do - Indicates that the plugin initialization found nothing to import/export
1100	Failed to establish a connection to an MI:Server - Indicates that a connection to an MI:Server instance could not be established.
1110	No default database available - Indicates that the server does not have a default database, or the default database is unavailable.
1111	No such database - Indicates that the specified database does not exist on the server, or is otherwise unavailable (e.g. due to access control).
1112	Failed to retrieve database - Indicates that the specified database could not be retrieved from the server.

Code	Description
1120	No such record - Indicates that the record does not exist on the server, or is otherwise unavailable (e.g. due to access control).
1121	Failed to retrieve record - Indicates that the specified record could not be obtained from the server.
1122	Multiple records matched - Indicates that multiple matching records have been found that satisfy criteria intended to identify a single record.
1130	No such subset - Indicates that the one or more specified subsets do not exist or are otherwise unavailable on the server (e.g. due to access control).
1140	No default table - Indicates that the database does not contain a default table, or the default table is otherwise unavailable (e.g. due to access control).
1141	No such table - Indicates that the database does not contain a default table, or the default table is otherwise unavailable (e.g. due to access control).