

Ansys GRANTA MI 2021 R1

GRANTA MI Excel Importer Reference Guide

Copyright and Trademark Information

© 2021 ANSYS, Inc. Unauthorized use, distribution or duplication is prohibited.

ANSYS, ANSYS Workbench, AUTODYN, CFX, FLUENT and any and all ANSYS, Inc. brand, product, service and feature names, logos and slogans are registered trademarks or trademarks of ANSYS, Inc. or its subsidiaries located in the United States or other countries. ICEM CFD is a trademark used by ANSYS, Inc. under license. CFX is a trademark of Sony Corporation in Japan. All other brand, product, service and feature names or trademarks are the property of their respective owners. FLEXlm and FLEXnet are trademarks of Flexera Software LLC.

Disclaimer Notice

THIS ANSYS SOFTWARE PRODUCT AND PROGRAM DOCUMENTATION INCLUDE TRADE SECRETS AND ARE CONFIDENTIAL AND PROPRIETARY PRODUCTS OF ANSYS, INC., ITS SUBSIDIARIES, OR LICENSORS.

The software products and documentation are furnished by ANSYS, Inc., its subsidiaries, or affiliates under a software license agreement that contains provisions concerning non-disclosure, copying, length and nature of use, compliance with exporting laws, warranties, disclaimers, limitations of liability, and remedies, and other provisions. The software products and documentation may be used, disclosed, transferred, or copied only in accordance with the terms and conditions of that software license agreement.

ANSYS, Inc. and ANSYS Europe, Ltd. are UL registered ISO 9001: 2015 companies.

U.S. Government Rights

For U.S. Government users, except as specifically granted by the ANSYS, Inc. software license agreement, the use, duplication, or disclosure by the United States Government is subject to restrictions stated in the ANSYS, Inc. software license agreement and FAR 12.212 (for non-DOD licenses).

Third-Party Software

See the legal information in the product help files for the complete Legal Notice for ANSYS proprietary software and third-party software. If you are unable to access the Legal Notice, contact ANSYS, Inc.

Published in the U.S.A.

Contents

1	<i>About this document</i>	5
1.1	Features of the Excel Importer.....	5
2	<i>About Excel import templates</i>	6
2.1	Template structure.....	6
2.2	Template workbooks.....	6
2.3	Creating Excel templates.....	7
2.4	Default template locations.....	7
3	<i>Attribute Lookup worksheet</i>	8
3.1	Attribute Name.....	9
3.2	Data Type	9
3.3	Units	11
3.4	Parent Attribute	12
3.5	Worksheet	12
3.6	Specimen	12
3.7	Data Range	12
3.8	Quality Range	13
3.9	Read Trailing Zeroes	13
3.10	Import.....	13
3.11	Line Type.....	14
3.12	Log	15
3.13	Interpolate.....	15
3.14	Invert	16
3.15	Estimate.....	16
3.16	Show As Table.....	17
3.17	Parameter n Range.....	18
3.18	Parameter n Default Value	18
3.19	Parameter n Name	19
4	<i>Parameter Lookup worksheet</i>	20
4.1	Parameter column.....	20
4.2	Data Range column	20
4.3	Units column	20

5	<i>Data worksheets</i>	21
5.1	Named Ranges.....	21
5.2	Setting 'Not applicable' values	24
5.3	Discrete data	24
5.4	File data	24
5.5	Float Functional data.....	25
5.6	Discrete functional data	30
5.7	Hyperlink data	30
5.8	Equations and Logic data.....	31
5.9	Tabular data.....	33
5.10	Multi-value data	36
5.11	Quality ratings	37
5.12	Record color	37
5.13	Subset membership.....	38
6	<i>Import Options worksheet</i>	39
6.1	Auto-placement.....	39
6.2	Conflict resolution and merging options.....	45
6.3	Placement of records into a specific folder hierarchy (Tree).....	50
6.4	Auto-Links.....	52
6.5	Access Control	55
6.6	Valid tables	57
6.7	Exported Worksheet Naming	57
6.8	MI Locale	57
7	<i>Advanced features</i>	58
7.1	Creating multiple records.....	58
7.2	Merging and appending	61
7.3	Deleting existing data	67
	<i>Appendix A. Setup for multi-step imports</i>	68
A.1	The Import Sequence worksheet	68
	<i>Appendix B. Excel Importer CLI</i>	69
B.1	Command-line arguments.....	69
B.2	Usage example	72
	<i>Appendix C. Troubleshooting</i>	73
C.1	Exit 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: www.spreadsheetgear.com/products/spreadsheetgear.net.aspx

2 About Excel import templates

2.1 Template structure

Excel import templates include a number of specialized worksheets that specify information required by the Excel Importer when importing data into GRANTA MI:

Worksheet name	Specifies	Required or Optional
Data	The data to be imported.	Required
Attribute Lookup	Attribute and record property information.	Required
Parameter Lookup	Information for Functional, Tabular, and multi-value Point data imports.	Optional
Import Options	Record placement, auto-linking, data merge/append, and access control options.	Required
Import Sequence	Options for importing data into more than one table or database.	Optional

Once set up, most of these worksheets can be hidden in the Excel template workbook to prevent unwanted changes, leaving just the *Data* worksheets visible to users.

2.2 Template workbooks

It is possible for the data to be imported to be specified in a completely separate Excel workbook: in this case, two files must be loaded into the Excel Importer:

- One or more data workbooks
- A template workbook containing Attribute Lookup, Import Options, Parameter Lookup, and any other required worksheets.

Having a separate Template workbook in this way allows multiple data workbooks to be imported using a single template. For example, you could 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 an import workbook that includes data, 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 standard import workbook, 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.

2.3 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.

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

2.4 Default template locations

By default, the Excel Importer looks for templates with a .xls, .xlsx, or .xlsm file extension in a Templates folder located with the plug-in. For example:

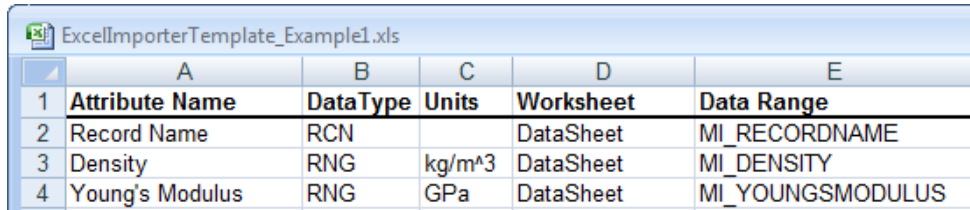
C:\Program Files\Granta\GRANTA MI\Toolbox\plugins\Importers\Excel\Templates

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.

3 Attribute Lookup worksheet

The *Attribute Lookup* worksheet contains a list of all the GRANTA MI attributes to be imported. For example:



	A	B	C	D	E
1	Attribute Name	Data Type	Units	Worksheet	Data Range
2	Record Name	RCN		DataSheet	MI_RECORDNAME
3	Density	RNG	kg/m ³	DataSheet	MI_DENSITY
4	Young's Modulus	RNG	GPa	DataSheet	MI_YOUNGSMODULUS

Each row specifies a different attribute, and each column specifies some required or optional property of the data to be imported. Some of the columns in the Attribute Lookup worksheet store values directly, and some use named ranges to look up data from another worksheet. The Attribute Name column must be the first column. The remaining columns may be in any order.

Table 1. Attribute Lookup Worksheet columns

Column Name	Required/optional...	...for these data types
Attribute Name	Required	All
Data Type	Required	All
Worksheet	Required	All
Data Range	Required	All
Units	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
Invert	Optional	Equations and Logic
Estimate	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 and Data Range columns) cannot be found in the current workbook. See Section 5.1 for more information on named ranges.

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

3.1 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.

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.

	A	B	C	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

3.2 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 2. *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)
FDD	Discrete Functional	Discrete (table)

DataType code	GRANTA MI data type	Comment
FIL or FILE	File	
GFA	Float Functional	Grid (graph)
HYP	Hyperlink	
INT	Integer	
LOG	Logical	
LTXT	Long Text	
MAFN	Equations and logic	
PIC	Picture	
PNT	Point	Includes multi-value point
RCLR	Record Color	See color codes on page 37
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 1.1
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.

3.3 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.

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 1](#)).

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 worksheet into the database value.

For example, suppose the currency for a database is US Dollars (USD). If you use units of 'currency' (see [Figure 1](#)), the value in the data import worksheet will be assumed to be in US Dollars. If you use units of 'AUD' (see [Figure 2](#)), the value in the data import worksheet will be treated as Australian Dollars.

	A	B	C	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 1. Importing using the database currency in the Attribute Lookup worksheet Units column

	A	B	C	D	E
1	Attribute Name	DataType	Units	Worksheet	Data Range
2	Capital cost	RNG	AUD	DataSheet	MI_CAPITALCOST
3	Price	RNG	AUD/kg	DataSheet	MI_PRICE

Figure 2. Importing using a chosen currency in the Attribute Lookup worksheet Units column

3.4 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 3). The meta-attribute must exist in the database, and the attribute name in the template must match the attribute name in the database exactly.

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

	A	B	C	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

Figure 3. Required columns for meta-attributes in the Attribute Lookup worksheet

3.5 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.

3.6 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.

3.7 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.)

3.8 Quality Range

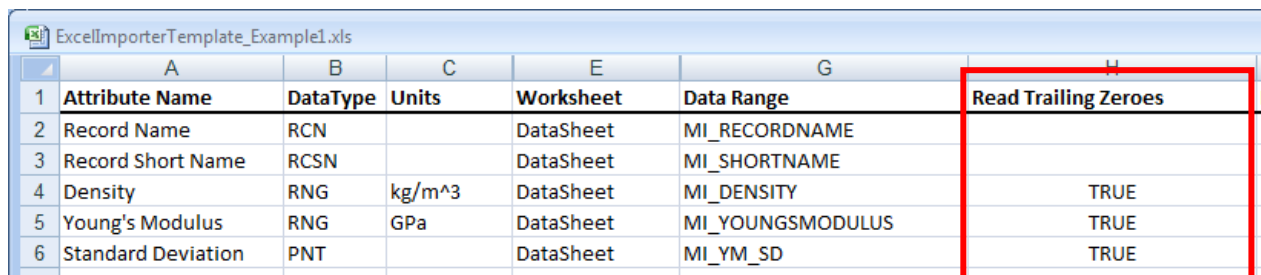
This column contains the name of the range where the quality rating for the data is stored. (See Section 5.1 for more information on named ranges.)

3.9 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.

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.



	A	B	C	E	G	H
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 4. 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.

3.10 Import

This column takes a Boolean value (TRUE/FALSE) that determines whether or not the attribute data will be imported. 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.

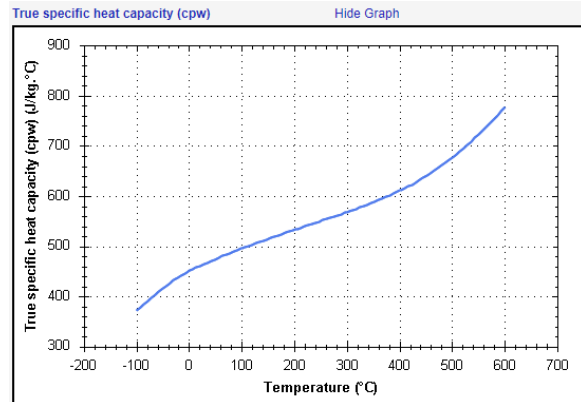
3.11 Line Type

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.

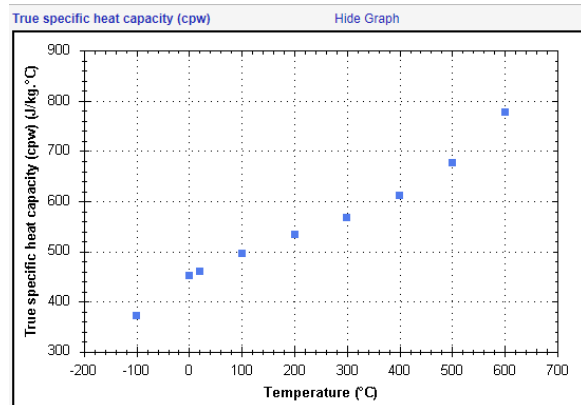
True specific heat capacity (cpw)		
Series 1		
Temperature (K)	Value (J/kg.°C)	Linetype:
173.1499939	371	371 LinesOnly
273.1499939	451	451
293.1499939	461	461
373.1499939	496	496

True specific heat capacity (cpw)		
Series 1		
Temperature (K)	Value (J/kg.°C)	Line type:
173.1499939	371	371 lines
273.1499939	451	451
293.1499939	461	461



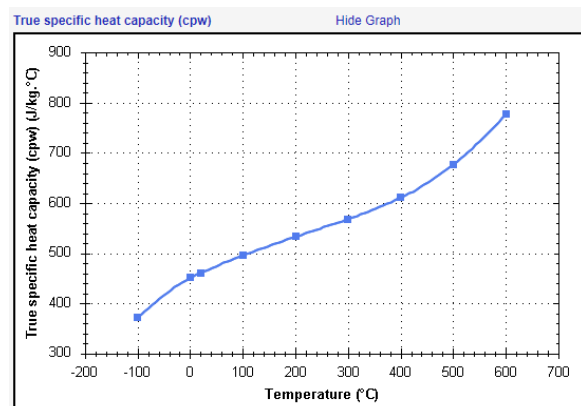
True specific heat capacity (cpw)		
Series 1		
Temperature (K)	Value (J/kg.°C)	Linetype:
173.1499939	371	371 MarkersOnly
273.1499939	451	451
293.1499939	461	461
373.1499939	496	496

True specific heat capacity (cpw)		
Series 1		
Temperature (K)	Value (J/kg.°C)	Line type:
173.1499939	371	371 markers
273.1499939	451	451
293.1499939	461	461



True specific heat capacity (cpw)		
Series 1		
Temperature (K)	Value (J/kg.°C)	Linetype:
173.1499939	371	371 Both
273.1499939	451	451

True specific heat capacity (cpw)		
Series 1		
Temperature (K)	Value (J/kg.°C)	Line type:
173.1499939	371	371 both
273.1499939	451	451
293.1499939	461	461
373.1499939	496	496



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 5.5.1 for more information.

3.12 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.

The setting applies to the entire functional data graph.

Defaults to FALSE if unspecified.

3.13 Interpolate

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

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 5.5.2); 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'.

3.14 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.

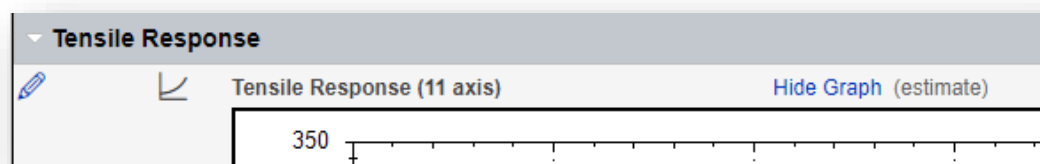
Value	Result
True	Plot the parameter value on the y axis and the attribute value on the x axis
False	The axes are not transposed. Defaults to FALSE if unspecified.

3.15 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.

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:



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.

3.16 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.

Value

Result

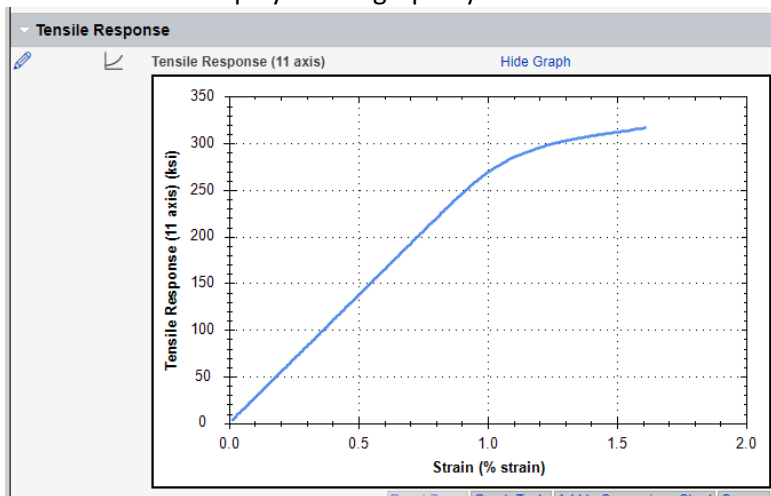
True

The data will be displayed as a table by default:

Strain (% strain)	Tensile Response (11 axis) (ksi)
0.01041	2.56
0.01388	3.84
0.01735	4.48
0.02082	5.77
0.02428	6.41
0.02775	7.69
0.03122	8.33
0.03469	9.61
0.03815	10.2
0.04163	11.5

False

The data will be displayed as a graph by default:



Defaults to FALSE if unspecified.

3.17 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 5](#)).

'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 [5.5.1](#).

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.

	A	B	C	E	F	G	J	K
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 5. Parameter Range and Default Value for a float functional attribute 'Series Daily Sunshine' containing series functional data

	A	B	C	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 6. Parameter Range for a multi-value point attribute 'Tension strength'

3.18 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 5](#)).

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.

3.19 *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 [5.5.1](#) 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 5.5.1.

The order of these columns must be as listed below; all columns are required.

1. Parameter
2. Data Range
3. Units

4.1 Parameter column

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 column

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 column

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 3.3, *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.

Table 3. Notes on Named Ranges for all data types

Data Type 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 5.10.1)
FDA	Float Functional (Series)	<p>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.</p> <p>The named range for parameter values for Parameter 1 must also point to adjacent cells in a single column for each series.</p> <p>The named range for parameter values for Parameter 2 and greater may refer to a single cell for each series.</p> <p>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 5.5.1)</p>
FDD	Discrete Functional	<p>The named range for data values must refer to adjacent cells arranged in a column for each series.</p> <p>The named range for parameter values for Parameter 1 must also point to adjacent cells in a single column for each series.</p> <p>The named range for parameter values for Parameter 2 and greater may refer to a single cell for each series. (See Section 5.6)</p>

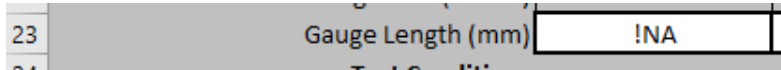
Data Type code	GRANTA MI data type	Notes
FIL or FILE	File	The named range for data values must refer to up to five adjacent cells where the data is located. (See section 5.3)
GFA	Float Functional (Gridded)	<p>The named range for data values must refer to adjacent cells arranged in a column (or two adjacent columns for range data).</p> <p>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.</p> <p>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 5.5.4)</p> <p>Grid functional data is always interpolated.</p>
HYP	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 5.7)
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 5.8)
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.

Data Type code	GRANTA MI data type	Notes
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 5.10.2)
RCLR	Record Color	The named range must refer to a single cell where the data is located. (See Section 5.12.)
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 1.1.
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 5.9)

5.2 Setting 'Not applicable' values

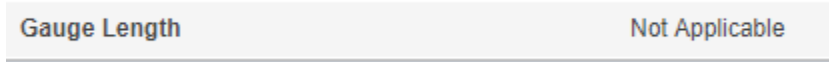
Data values can be set to 'Not applicable' by specifying !NA as the value for the Named Range. For example:

Import worksheet:



23	Gauge Length (mm)	!NA
----	-------------------	-----

Imported data viewed in MI:Viewer:



Gauge Length	Not Applicable
--------------	----------------

Note: !NA cannot be used for tabular (TABL) data types.

5.3 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 [5.10.1](#) for information on multi-value discretets.

5.4 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 the following table.

Table 4. 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 5.4.1). 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.

No column heading is required, but one can be used to make the worksheet easier to interpret, as in the example below. The column heading should not be included in the named range.

In this example, each row is a record:

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

5.4.1 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.
- If the attribute is not searchable, the file will not be indexed.

5.5 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
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

Parameter 1 Range

Parameter 1 Default Value

Parameter default values

The named range refers to a single cell that contains the default value (see example above). 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.

5.5.1 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 consist 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.

5.5.2 Interpolation

The named range <Parameter>_INTERP identifies a cell that contains the interpolation type for the parameter. The setting is stored in the cell itself; valid values are shown in the following table.

Table 5. Parameter interpolation values

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.

5.5.3 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 6 .

Table 6. 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

Range	...is set to
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

Table 7. Defining Named Ranges for series functional data

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 low 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.

Type of Functional Data	Notes
Multiple series of range data	<p>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.</p> <p>The named range(s) for parameter values must refer to adjacent cells in a single column.</p> <p>These ranges must be named using the suffix convention noted above.</p>

5.5.4 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 7 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 5.5.2 and 5.5.3 for valid values.

Table 8. 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	<p>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.</p> <p>The named range(s) for parameter values must refer to adjacent cells in a single column.</p>

	A	B	C	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	Type B
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 7. Example of grid functional data

5.6 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 5.5.1), with the exceptions that Line Type does not apply, and the data should be treated as series of point data.

5.7 Hyperlink data

Data for a hyperlink attribute can be specified as a Uniform Resource Locator (URL) or as an HTML hyperlink.

5.7.1 Uniform Resource Locator (URL)

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

Tokens can be included in the URL to allow replacement strings or short text attribute names to be inserted into the resolved address; see 5.7.3 for details.

5.7.2 HTML hyperlink

An HTML format hyperlink 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. Tokens can be included in the URL to allow replacement strings or short text attribute names to be inserted into the resolved address; see 5.7.3 for details.

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

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

The value for `target` can be one of the following:

Value	Behavior
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.

Value	Behavior
_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.

Example:

```
<a href="http://www.grantadesign.com" target="_blank">Granta website</a>
```

5.7.3 Including tokens in URLs

A URL may include tokens that are replaced with values defined in the database when the URL is resolved.

Two different types of token can be included:

- **Replacement string.** This allows a replacement string defined in the database schema to be inserted into the URL.
- **Short text attribute.** This allows the value of a short text attribute in the record to be inserted into a URL within the same record. For example, a project code, a testing series ID, or a batch number.

Tokens have the following syntax:

```
{s:name-of-replacement-string}
```

```
{a:name-of-attribute}
```

For example:

```
{s:webURL}/resources/note1.html
```

```
https://acme.com/{a:Project Code}/notes.htm
```

```
{s:webURL}/{a:Material Type}/{a:Batch Number}.html
```

See the MI:Admin help for more information about Replacement Strings in GRANTA MI.

5.8 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](#).

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 8):

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

Values	...are stored in range named
Parameter 3 value for curve 02	FatMod_Param3_Curve02
Label for curve 02	FatMod_Label_Curve02
...and so on.	...

	A	B	C	D	E	F
1						
2		Record Name	Excel Importer record			
3						
4						
5		Model type	Fatigue Model			
6						
7		Cycles	1	10000000		
8		Stress Ratio	2	32		
9		Temperature	23	300		
10						
11						
12			Label	Cycles	Stress Ratio	Temperature
13		Curve 1	Fully reversible	1000	4	317.15
14		Curve 2	Tension	2000	5	328.15
15		Curve 3				

Figure 8. 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.

5.9 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](#). 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.

5.9.1 Row order

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

5.9.2 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.

5.9.3 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\Plugin-ins\Excel\Test.txt
```

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

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

For example:

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

See Section 5.7.2 for alternative values for *target*.

5.9.4 Example

This example describes how to set up the importer for the tabular attribute *Substances found in this material* which is linked to another table in the database. 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:

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

For example:

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

On the *Data* worksheet:

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

For example:

	A	B	C	D
1				
2	Record Name	Epoxy resin, high heat		
3				
4	Attribute	Substances found in this material		
5				
6	Row Order	CAS Number	Amount (g)	Substance Name
7	1	101-77	2.5	4-4' -Methylenedianiline
8	2	62-53	0.7	Aniline
9	3	50-01	5.3	Formaldehyde
10				

5.10 Multi-value data

5.10.1 Multi-value discrete

The named range for the Data Range must consist of adjacent cells. You can use cells in multiple columns (as in the example shown below) 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.

	J	K	L
1			
2	Product Form		
3	Bar	Forging	Tubing

Data Range

5.10.2 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.

The example below 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 6](#).

The Data Range and Parameter Range for a multi-value point attribute 'Tension strength':

	C	D	E
5	Mechanical properties		
6	Tension strength		
7	Basis	Values (MPa)	Quality rating
8	Mean	456.4	Good
9	A-basis	381.3	
10	B-basis	411.6	

Quality Range
MI_TENSION_QUALITY

Parameter Range
MI_TENSIONSTRENGTH_BASIS

Data Range
MI_TENSIONSTRENGTH

5.11 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.

5.11.1 Continuous quality ratings systems

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

5.11.2 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.

5.11.3 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.

5.11.4 Multi-value attributes

A multi-value attribute only has one quality rating for its data, even though the data has multiple values.

5.12 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.

Table 9. Permitted color values

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

5.13 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.

	A	B	C	D	E	F
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						

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.

Table 10. Auto-Placement options

To...	Use Named Range...	See
Identify the database and table	CES_EXPORT_PLACEMENTDBKEY CES_EXPORT_PLACEMENTTABLE	Section 6.1.1
Place the records in the target table	CES_EXPORT_PLACEMENTFORCETOROOT CES_EXPORT_PLACEMENTSEARCHVALUE or CES_EXPORT_PLACEMENTSEARCHATTRIBUTE CES_EXPORT_PLACEMENTMATCHPROPERTY or CES_EXPORT_PLACEMENTATTRIBUTE CES_EXPORT_PLACEMENTISCHILD CES_EXPORT_PLACEMENTCREATEIFNOTFOUND	Section 6.1.2
Place the incoming records in a specific subset	CES_EXPORT_FILTER	Section 1.1

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 11. 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.

Table 12. Record placement options

Placement Option	Notes
CES_EXPORT_PLACEMENTFORCETOROOT	<p>Specifies whether to place the records being imported into the root of the table.</p> <p>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).</p> <p>If FALSE, then the option is ignored.</p> <p>The default setting is FALSE if the cell is blank.</p> <p>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.</p>
CES_EXPORT_PLACEMENTSEARCHVALUE or CES_EXPORT_PLACEMENTSEARCHATTRIBUTE	<p>These named ranges specify the search term to be used when searching and matching on an existing record.</p> <p>CES_EXPORT_PLACEMENTSEARCHVALUE specifies a value that will be used as the search term.</p> <p>CES_EXPORT_PLACEMENTSEARCHATTRIBUTE refers to a single cell containing an attribute name, as specified in the Attribute Lookup worksheet. The value of the corresponding attribute in the Data worksheet will be used as the search term.</p> <p>SEARCHVALUE and SEARCHATTRIBUTE are alternatives; both named ranges may be defined, but one must be blank. If both have values, then SEARCHVALUE takes precedence.</p>
CES_EXPORT_PLACEMENTMATCHPROPERTY or CES_EXPORT_PLACEMENTATTRIBUTE	<p>These named ranges specify the keyword to be used when searching and matching on an existing record.</p> <p>CES_EXPORT_PLACEMENTMATCHPROPERTY specifies a record property, one of FULLNAME, TREENAME, RECORDID or GUID.</p> <p>CES_EXPORT_PLACEMENTATTRIBUTE specifies the name of an attribute.</p> <p>MATCHPROPERTY and ATTRIBUTE are alternatives; both named ranges may be defined, but one must be blank.</p>

Placement Option	Notes
CES_EXPORT_PLACEMENTISCHILD	<p>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).</p> <p>If TRUE, then the data will be imported into a new record below the folder or record identified.</p> <p>If FALSE, then the data will be imported directly into the matched record or folder.</p> <p>The default setting is FALSE if the cell is blank.</p>
CES_EXPORT_PLACEMENTCREATEIFNOTFOUND	<p>Specifies what should be done with the incoming data if there is no existing record with a matching name:</p> <p>If FALSE, the data is not imported.</p> <p>If TRUE, the data is placed in a new record, created in the folder specified in MI_TREE.</p>

Table 13. CES_EXPORT_PLACEMENTISCHILD behavior

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 example 1: import records at the top level of the MaterialUniverse tree

Auto-Placement	
DB Key	MJ_Training
Table	MaterialUniverse
Force To Root	TRUE
Search Value	
Search Attribute	
Match Property	
Match Attribute	
Is Child Record	
Create If Not Found	
Subset	

CES_EXPORT_PLACEMENTFORCETOROOT
named range

Auto-placement example 2: search for a named record

“Search the MaterialUniverse table for a record named 7075”

Auto-Placement	
DB Key	MJ_Training
Table	MaterialUniverse
Force To Root	
Search Value	7075
Search Attribute	
Match Property	TREENAME
Match Attribute	
Is Child Record	TRUE
Create If Not Found	
Subset	All bulk materials

Search Value is specified, so Search Attribute is blank.

Match Property is specified, so Match Attribute is blank.

Auto-placement example 3: search for a record with a specified attribute value

“Search the MaterialUniverse table for a record whose value for Density is 2770 kg/m³”

Auto-Placement	
DB Key	MJ_Training
Table	MaterialUniverse
Force To Root	
Search Value	2270
Search Attribute	
Match Property	
Match Attribute	Density
Is Child Record	
Create If Not Found	
Subset	All bulk materials

Search Value is specified, so Search Attribute is blank.

Match Attribute is specified, so Match Property is blank.

Auto-placement example 4: search for a record with a matching attribute value

“Search the Young’s Modulus attribute in the MaterialUniverse records using the value for Young’s Modulus given in the Data worksheet”

Auto-Placement	
DB Key	MI_Training
Table	MaterialUniverse
Force To Root	
Search Value	
Search Attribute	Young's Modulus
Match Property	
Match Attribute	Young's Modulus
Is Child Record	
Create If Not Found	
Subset	All bulk materials

Search Attribute is specified, so Search Value is blank.

Match Attribute is specified, so Match Property is blank

Data worksheet:

Mechanical Properties			
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

Note that the search attribute (*Young's Modulus*) in the template in this example must be set to not import, i.e. **Import=False** in the Attribute Lookup worksheet:

	A	B	C	D	E	F	G	H	I	J
	Attribute Name	Data Type	Units	Parent Worksheet	Specimen	Data Range	Quality Range	Read Trailing	Import	
0	Young's modulus	RNG	GPa	Data		MI_YOUNGSMODULUS	MI_YOUNGSMODULUS_QUALITY	FALSE	FALSE	

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.

Auto-placement options set to import records into the 'All bulk materials' subset:

Auto-Placement	
DB Key	MI_Training
Table	MaterialUniverse
Force To Root	TRUE
Search Value	
Search Attribute	
Match Property	
Match Attribute	
Is Child Record	
Create If Not Found	
Subset	All bulk materials

CES_EXPORT_FILTER
named range

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.

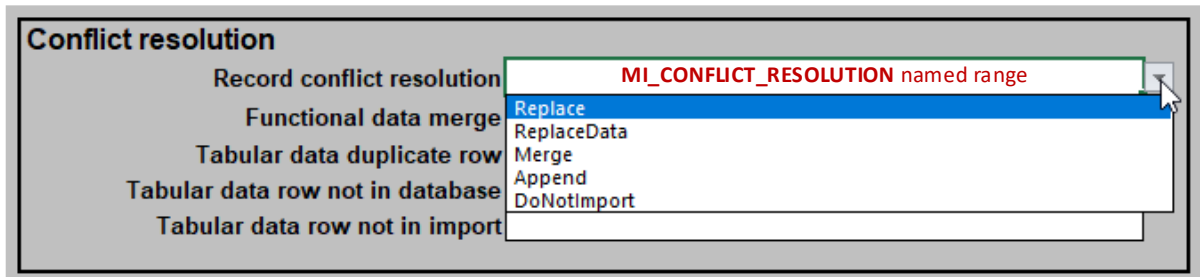
Conflict resolution options, specified in the *Import Options* worksheet, define the strategy for dealing with incoming data in this situation.

The fields and named ranges are shown here:

Conflict resolution	
Record conflict resolution	MI_CONFLICT_RESOLUTION
Functional data merge	MI_FUNCTIONALMERGEOPTION
Tabular data duplicate row	MI_TABULAR_DUPLICATE_ROW_MERGEOPTION
Tabular data row not in database	MI_TABULAR_ROW_NOT_IN_DATABASE_MERGEOPTION
Tabular data row not in import	MI_TABULAR_ROW_NOT_IN_IMPORT_MERGEOPTION

See also [Section 7.2, Merging and appending](#), for examples of appending and merging different types of data.

6.2.1 Record conflict resolution options



The MI_CONFLICTRESOLUTION named range provides options that define how conflicts between incoming data and an existing record in the database will be resolved. The values are not case sensitive.

Table 14. Record conflict resolution options

Value	Behavior
Replace	<p>The existing record and all the links associated with the record are deleted. The new record is then created.</p> <p>No information from the existing record is preserved.</p> <p>This option is not available for generic records.</p>
ReplaceData	<p>The existing record is kept and the incoming data is added to the record.</p> <ul style="list-style-type: none"> Where there is existing data, this will be replaced by the incoming data. Where there is no existing data, the incoming data will be added. If there is no incoming data for an attribute, existing data remains unchanged. <p>There is no control over which data is replaced.</p> <p>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.</p>
Merge	<p>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.</p> <ul style="list-style-type: none"> Where there is existing multi-value data, the incoming data is merged. Where there is existing functional data, the Functional data merge option specifies how that data should be merged. Where there is existing tabular data, the Tabular data row merge options specify how that data should be merged. <p>See also Section 7.2, Merging and appending, for examples of appending and merging different types of data</p>

Value	Behavior
Append	<p>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.</p> <ul style="list-style-type: none"> Where there is existing multi-value data, the incoming data is appended. Where there is existing functional data, the Functional data merge option specifies how that data should be appended. Where there is existing tabular data, the incoming data is appended. <p>See also Section 7.2, Merging and appending, for examples of appending and merging different types of data</p>
DoNotImport	<p>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.</p>

If no option is specified (the **Record conflict resolution** field is blank) in the template, 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 options

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 *Functional data merge* field:

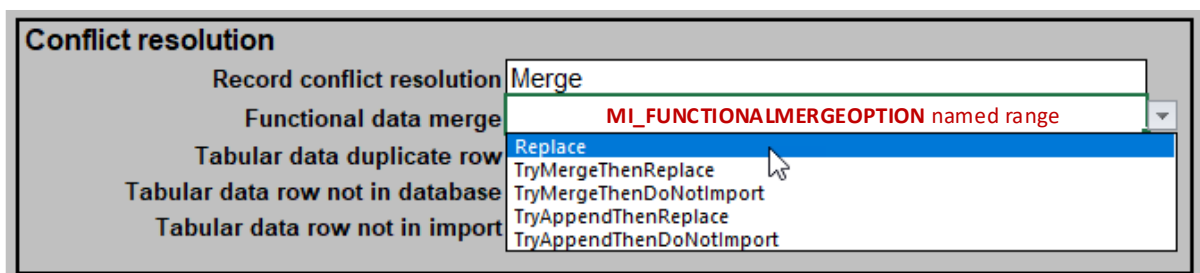


Table 15. Functional data merge options

Value	Behavior
Replace	<p>The existing functional data is deleted and the incoming functional data is added.</p> <p>No attempt is made to merge or append the incoming functional data with the existing data, even if the data is compatible.</p>
TryMergeThenReplace	<p>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.</p> <p>If the incoming functional data is not compatible with the existing functional data, the existing data is deleted and the incoming data is added.</p>
TryMergeThenDoNotImport	<p>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.</p> <p>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.</p>
TryAppendThenReplace	<p>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.</p> <p>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.</p>
TryAppendThenDoNotImport	<p>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.</p> <p>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.</p>

Note that the **Functional data merge** option overrides the **Record conflict resolution** setting. For example, if **Record conflict resolution** = Merge:

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

If no option is specified (the **Functional data merge** field is blank) in the template, 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

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.

If a duplicate record name strategy of Merge is chosen, the Excel Importer handles the incoming and existing tabular data rows according to these tabular data merge options:

Conflict resolution	
Record conflict resolution	Merge
Functional data merge	
Tabular data duplicate row	MI_TABULAR_DUPLICATE_ROW_MERGEOPTION
Tabular data row not in database	MI_TABULAR_ROW_NOT_IN_DATABASE_MERGEOPTION
Tabular data row not in import	MI_TABULAR_ROW_NOT_IN_IMPORT_MERGEOPTION

Note that if **any** of the tabular row merge 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 16* will be used.

Table 16. Tabular data merge options

Named range	Description
MI_TABULAR_DUPLICATE_ROW_MERGEOPTION	<p>Defines what happens to duplicate tabular rows. Options are:</p> <ul style="list-style-type: none"> • 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. <p>The default behavior if no value is specified is Replace.</p>

Named range	Description
MI_TABULAR_ROW_NOT_IN_DATABASE_MERGEOPTION	<p>Defines what happens when a tabular row exists in the import template but not in the database. Options are:</p> <p>Append – the incoming tabular row is appended.</p> <p>DoNotImport – the new tabular row is not imported and the existing data is unchanged.</p> <p>The default behavior if no value is specified is Append.</p>
MI_TABULAR_ROW_NOT_IN_IMPORT_MERGEOPTION	<p>Defines what happens when a tabular row exists in the database but not in the import template. Options are:</p> <p>Delete – the existing data is deleted.</p> <p>DoNotDelete – the existing data is unchanged.</p> <p>The default behavior if no value is specified is Delete.</p>

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	<p>The range must be n cells by 1 cell, or 1 cell by n cells only.</p> <p>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 Attribute Lookup worksheet. The data for the attribute on the Data worksheet(s) is used as the folder name in the tree.</p>

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 3.6). 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 12](#). 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 9) and the data values for those attributes in the *Data* worksheet (columns B, C, D and E in Figure 11). 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.

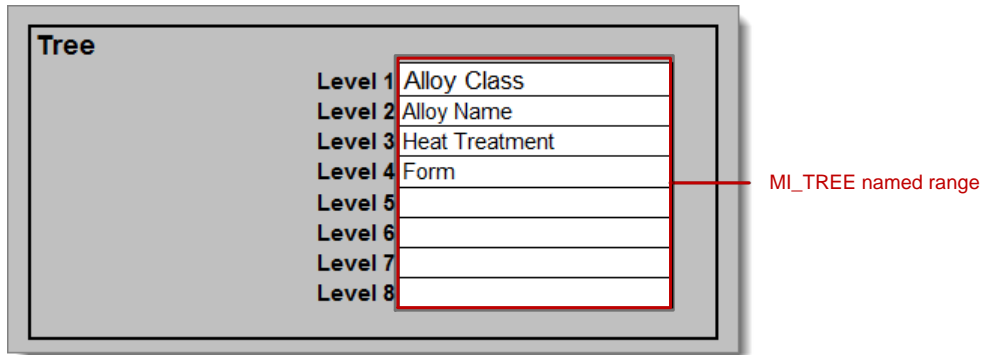


Figure 9. Named range on the *Import Options* worksheet

In Figure 9, 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	B	C	D	E	F	G	H	I	J
1	Attribute Name	Data Type	Units	Pare Worksheet	Specimen	Data Range		Qual Read	Import	
2	Record Name	RCN			DataSheet	ROW	MI_RECORDNAME			
3	Density	PNT	lb/in^3		DataSheet	ROW	MI_DENSITY			
4	Alloy Class	STXT			DataSheet	ROW	MI_LEVEL1			FALSE
5	Alloy Name	STXT			DataSheet	ROW	MI_LEVEL2			FALSE
6	Heat Treatment	STXT			DataSheet	ROW	MI_LEVEL3			FALSE
7	Form	STXT			DataSheet	ROW	MI_LEVEL4			FALSE

Figure 10. *Attribute Lookup* worksheet

Figure 10 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	B	C	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	T6	Plate	0.1
6	MIL 3013	2000 Series	2014	T6	Sheet	0.1
7	MIL 3034-1	2000 Series	2024	T3	Sheet	0.1
8	MIL 3034-2	2000 Series	2024	T3	Sheet	0.1
9	MIL 3239	7000 Series	7075	T73	Bar	0.1
10	MIL 3009	7000 Series	7075	T73		0.1
11	Bar	7000 Series	7075	T73		0.1
12						

MI_LEVEL1
MI_LEVEL2
MI_LEVEL3
MI_LEVEL4

Figure 11. *Data* worksheet

Figure 11 shows the *Data* worksheet with the data for the attributes named in the *Import Options* worksheet (Figure 9). 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.

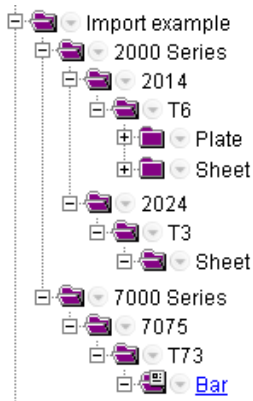


Figure 12. Tree hierarchy in MI:Viewer

Figure 12 shows the tree hierarchy in MI:Viewer below the 'Import example' folder (excluding data records) resulting from the example data in [Figure 9](#) and [Figure 11](#).

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 17](#).

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

Table 17. 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.

Named ranges	Notes
CES_EXPORT_SOURCEVALUES	<p>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.</p> <p><i>SOURCEVALUES and SOURCEATTRIBUTE are alternatives. Both named ranges may be defined, but one must be blank.</i></p>
CES_EXPORT_SOURCEATTRIBUTE	<p>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.</p> <p><i>SOURCEVALUES and SOURCEATTRIBUTE are alternatives. Both named ranges may be defined, but one must be blank.</i></p>
CES_EXPORT_DESTINATIONPROPERTY	<p>The destination table is the table that is to be searched for matching values and contains the records to be linked to.</p> <p>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.</p> <p>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.</p> <p><i>DESTINATIONPROPERTY and DESTINATIONATTRIBUTE are alternatives. Both named ranges may be defined, but one must be blank.</i></p>
CES_EXPORT_DESTINATIONATTRIBUTE	<p>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.</p> <p>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.</p> <p><i>DESTINATIONPROPERTY and DESTINATIONATTRIBUTE are alternatives. Both named ranges may be defined, but one must be blank.</i></p>

Named ranges	Notes
CES_EXPORT_LINKSCRITERIA (data link groups only)	<p>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:</p> <p>NoChecks. No checks on whether the target attributes contain data</p> <p>AtLeastOneDataRequired. At least one of the target attributes must contain data</p> <p>AllDataRequired. Each target attribute must contain data</p>
CES_EXPORT_LINKCLUSTERATTRIBUTE (data link groups only)	<p>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.</p> <p>The name of the primary attribute and the name of the data link group together uniquely identify the data link group.</p>
CES_EXPORT_LINKSOURCETABLE (data link groups only)	<p>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.</p> <p>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.</p>

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

The first set of links (Link 1 in [Figure 13](#)) 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 13](#)) 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 14](#)).

Table 18. Named ranges for access control permissions

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 Attribute Lookup worksheet, and the data for that attribute is used to set the permission. (The attribute itself can be set to not be

Named range**Notes**

imported, see Section 3.6). 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.

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.

Access Control	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			
Permission 10			
Permission 11			
Permission 12			

MI_AC_NAME
MI_AC_READ
MI_AC_WRITE

Figure 14. 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.

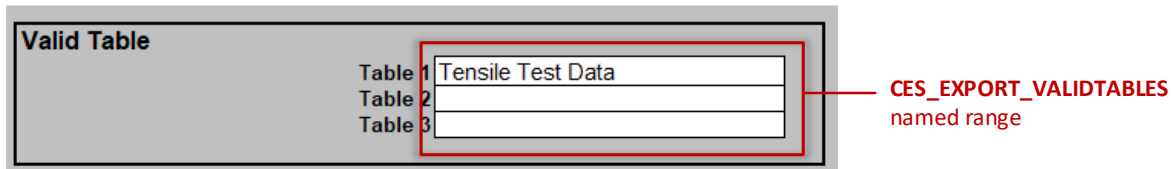


Figure 15. Valid table options in the *Import Options* worksheet

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.

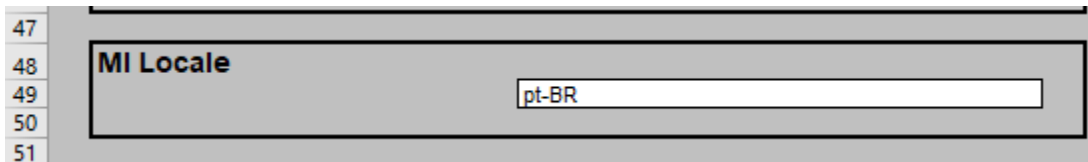
6.7 Exported Worksheet Naming

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.

6.8 MI Locale

The locale of the files to be imported with the template can be specified by including a language and culture code in the MI Locale field. This is used to determine whether a comma or a period is recognized as representing the decimal separator.

For example, to specify Brazilian Portuguese:



When a value is specified in this field, the data workbook is opened using the culture specified. If the field is empty or there's no valid value specified, the workbook is opened using the current culture.

A useful reference for information about language codes and language support in Windows can be found on this on this Microsoft Go Global Developer Center web page: [National Language Support \(NLS\) API Reference](#).

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 3.6).

- **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 16).

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.

	A	B	C	D	E	F	G	H
	Attribute Name	Parent Attribute	Data Type	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			Functional Data	CES_FUNCDATA_VALUE	CES_FUNCDATA_PAI
13	Functional Point 2		SDP			Functional Data	CES_FUNCDATA_VALUE	CES_FUNCDATA_PAF
14	Functional Point 3		SDP			Functional Data	CES_FUNCDATA_VALUE	CES_FUNCDATA_PAF
15	Functional Range		SDR			Functional Data	CES_RANGEFUNCDATA_VALUE	CES_FUNCDATA_PAI
16	Functional Point 4		SDR			Functional Data	CES_RANGEFUNCDATA_VALUE	CES_FUNCDATA_PAF
17								

Figure 16. 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.

	A	B	C	D	E	F
1						
2						
3		Specimen Code	Temperature deg C	Diameter mm	Youngs Modulus GPa	elongation %
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-Y2	400	10	125	14
15						
16		A2-Z2	500	10	110	18
17						
18						

Figure 17. Example of data in rows. Each row will form a new database record.

In Figure 17, 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

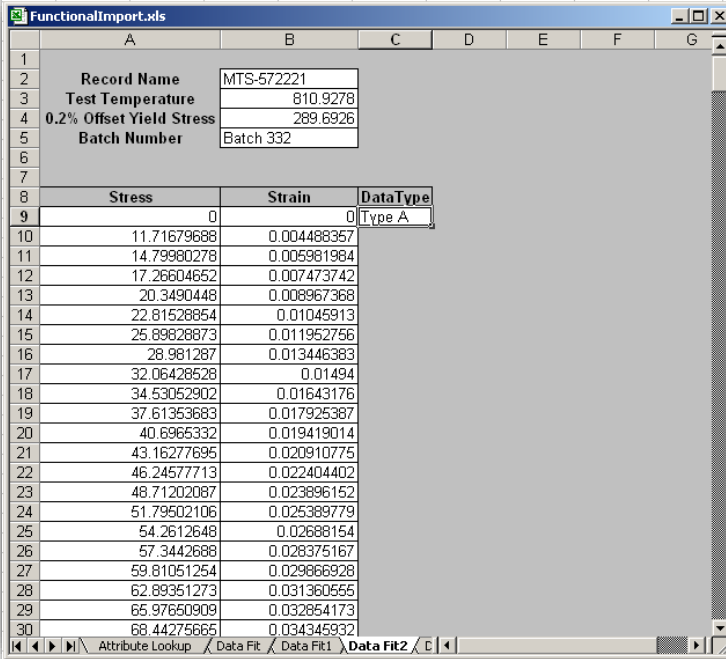
A new record will be created for each column in the *Data* worksheet. The row containing the record name in the *Data* worksheet is determined by the information in the 'Record Name' row in the *Attribute Lookup* worksheet. This row will also be used to determine the number of records to create in the 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 18, 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.



	A	B	C	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							
7							
8		Stress	Strain	DataType			
9		0	0	Type A			
10		11.71679688	0.004488357				
11		14.79980278	0.006981984				
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	0.028375167				
27		59.81051254	0.029866928				
28		62.89351273	0.031360555				
29		65.97650909	0.032854173				
30		68.44275665	0.034345932				

Figure 18. 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 19).

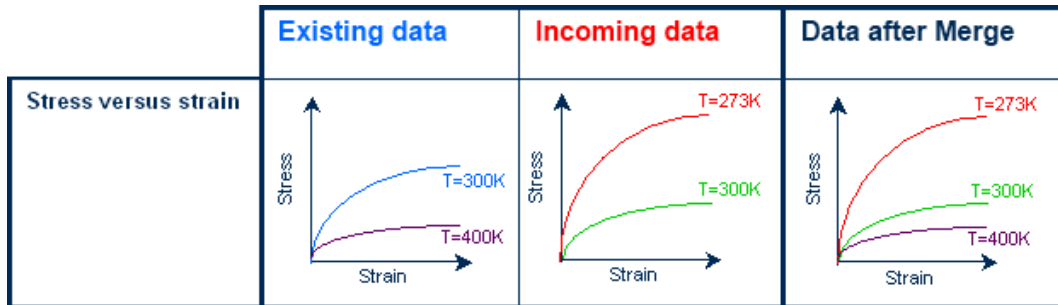


Figure 19. Example of merging series functional data

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 20).

	Existing data	Incoming data	Data after Merge																																																																						
F (X, Y)	<table border="1"> <thead> <tr> <th></th> <th colspan="3">X</th> </tr> <tr> <th></th> <th>1</th> <th>2</th> <th>3</th> </tr> </thead> <tbody> <tr> <th>Y 55</th> <td>111</td> <td>112</td> <td>113</td> </tr> <tr> <th>Y 57</th> <td>115</td> <td>116</td> <td>117</td> </tr> <tr> <th>Y 59</th> <td>118</td> <td>119</td> <td>120</td> </tr> </tbody> </table>		X				1	2	3	Y 55	111	112	113	Y 57	115	116	117	Y 59	118	119	120	<table border="1"> <thead> <tr> <th></th> <th colspan="4">X</th> </tr> <tr> <th></th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> </tr> </thead> <tbody> <tr> <th>Y 55</th> <td></td> <td></td> <td>116</td> <td>118</td> </tr> <tr> <th>Y 57</th> <td></td> <td>118</td> <td></td> <td>123</td> </tr> <tr> <th>Y 59</th> <td></td> <td></td> <td></td> <td>126</td> </tr> </tbody> </table>		X					1	2	3	4	Y 55			116	118	Y 57		118		123	Y 59				126	<table border="1"> <thead> <tr> <th></th> <th colspan="4">X</th> </tr> <tr> <th></th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> </tr> </thead> <tbody> <tr> <th>Y 55</th> <td>111</td> <td>112</td> <td>116</td> <td>118</td> </tr> <tr> <th>Y 57</th> <td>115</td> <td>118</td> <td>117</td> <td>123</td> </tr> <tr> <th>Y 59</th> <td>118</td> <td>119</td> <td>120</td> <td>126</td> </tr> </tbody> </table>		X					1	2	3	4	Y 55	111	112	116	118	Y 57	115	118	117	123	Y 59	118	119	120	126
	X																																																																								
	1	2	3																																																																						
Y 55	111	112	113																																																																						
Y 57	115	116	117																																																																						
Y 59	118	119	120																																																																						
	X																																																																								
	1	2	3	4																																																																					
Y 55			116	118																																																																					
Y 57		118		123																																																																					
Y 59				126																																																																					
	X																																																																								
	1	2	3	4																																																																					
Y 55	111	112	116	118																																																																					
Y 57	115	118	117	123																																																																					
Y 59	118	119	120	126																																																																					

Figure 20. 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 non-varying 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 21).

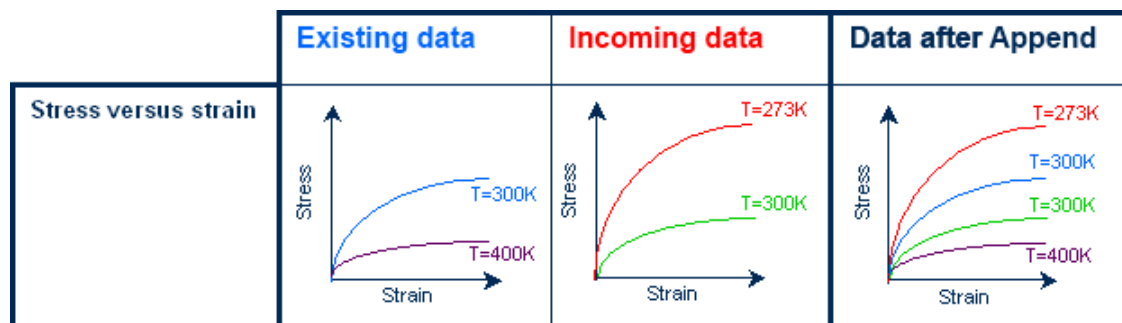


Figure 21. Example of appending series functional data

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 **Functional data merge** (MI_FUNCTIONALMERGEOPTION) value specified 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

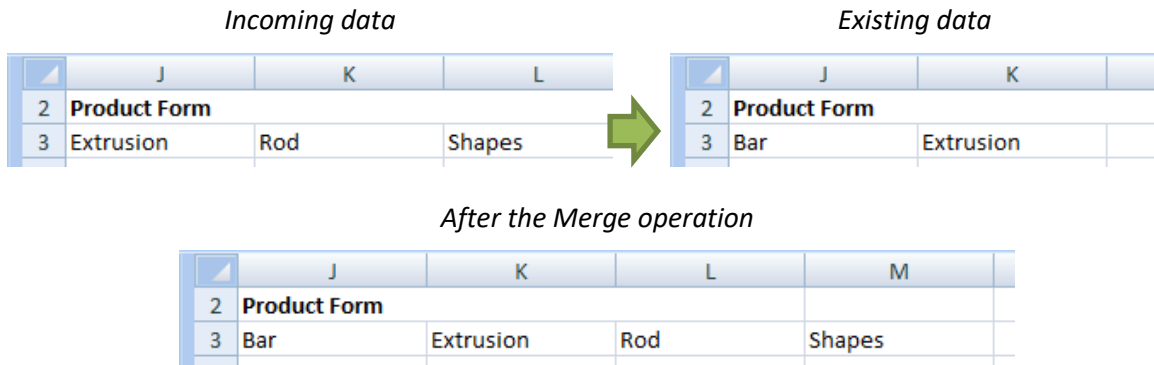
All tabular rows in the importer will be added to the existing tabular rows in the database. This may result in duplicate tabular rows (i.e. more than one row with the same linking value in the data for a tabular attribute).

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:



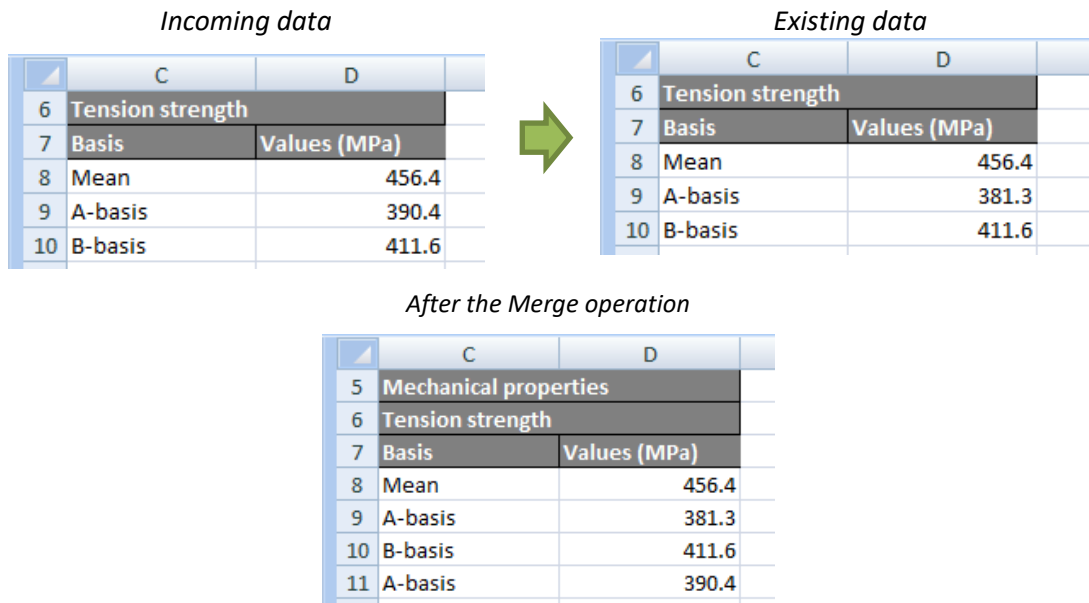
Result as displayed in MI:Viewer datasheet

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



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:

<i>Incoming data</i>			<i>Existing data</i>			
	J	K	L		J	K
2	Product Form			2	Product Form	
3	Extrusion	Rod	Shapes	3	Bar	Extrusion

After the Append operation

	J	K	L	M	N
2	Product Form				
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:

<i>Incoming data</i>			<i>Existing data</i>		
	C	D		C	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

	C	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	Mean	456.4
12	A-basis	390.4
13	B-basis	411.6

Result as displayed in MI:Viewer datasheet

Tension strength	Value	Unit
	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.

Tabular rows may be deleted using the MI_TABULAR_ROW_NOT_IN_IMPORT_MERGEPTION setting; see Section 6.2.3.

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.

Note that deleting the data value is not the same as setting the data value to 'Not Applicable'; see [Setting 'Not applicable' values](#).

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.

	A	B	C	D	E	F	G	H	I	J
1	Step	Attribute Lookup	Import Options	Parameter Lookup						
2	Materials	Materials Attributes	Materials Import Options	Materials Parameters						
3	Samples	Samples Attributes	Samples Import Options							
4										
5										
6										

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.

Because multi-step imports must use auto-placement, an Import Options worksheet is required for each step.

Note: The column names must be **Attribute Lookup**, **Import Options**, and **Parameter Lookup**. If the column names on this worksheet are not correct (for example, “*Import Option*” instead of “*Import Options*”) then the import will not work as expected and you won't get a error message explaining why.

Appendix B. Excel Importer CLI

The Excel Importer can be run as a scheduled task using a command line interface (CLI). The Excel Importer 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.ExcellImporter

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.ExcellImporter
- 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.

```
-logfile "C:\my logs\importprocess.log"
```

```
-plugin Granta.ExcellImporter
```

(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*.

-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 separated 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.ExcellImporter -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

(*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 Importer 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

Code	Description
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.
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).
