

GRANTA MI™ version 11

MI:Toolbox Bulk Data Importer (BDI) Guide

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

www.grantadesign.com

© Granta Design 2017 All rights reserved

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

Microsoft®, Excel®, PowerPoint®, Internet Explorer®, SQL Server®, Windows®, and Windows Server® are registered trademarks of Microsoft Corporation or its subsidiaries in the United States or other countries.

Granta Design Ltd. makes reasonable efforts to explicitly acknowledge all trademarks cited in our literature or on our website. If you would like us to add or alter an acknowledgement, please [contact us](#).

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

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

Document version: MI11/01
Published: December 2017

Table of Contents

1	<i>About the Bulk Data Importer.....</i>	<i>4</i>
2	<i>Data format for import</i>	<i>5</i>
2.1	Header Rows	5
2.2	Data Cells	6
2.3	Trailing zeroes	7
3	<i>Attribute name formats that use multiple columns</i>	<i>8</i>
3.1	Range	8
3.2	Multi Value Discrete	8
3.3	Hyperlink	9
3.4	File	10
3.5	Equations and Logic.....	11
4	<i>Special column names.....</i>	<i>13</i>
5	<i>Record hierarchy</i>	<i>15</i>
5.1	Special columns	16
5.2	Data values	16
6	<i>Attribute data values</i>	<i>18</i>
7	<i>Using the Bulk Data Importer CLI.....</i>	<i>20</i>
7.1	Command-line arguments	20
7.2	CLI usage examples.....	24
8	<i>Troubleshooting</i>	<i>25</i>
8.1	MI:Toolbox general errors.....	25
8.2	Bulk Data Importer errors	26

1 *About the Bulk Data Importer*

The Bulk Data Importer (BDI) is a tool for bulk import of data from Microsoft Excel files or from CSV format text files into a GRANTA MI database, and creates new records in an existing table in the selected GRANTA MI database. It can be used to create hundreds of records without the need for any user input.

- It can import data for all data types except float functional, discrete functional, tabular, and multi-value point data.
- It cannot automatically create record links.
- It cannot create new database schema objects in a GRANTA MI database. Therefore, all attributes, units, discrete types etc. must exist in the database before the import. If necessary, ask the administrator of the GRANTA MI database to edit the database with MI:Admin to prepare it for import.

The Bulk Data Importer can be run

- From within the MI:Toolbox application; see the MI:Toolbox Help for information on how to use the Bulk Data Importer plugin for MI:Toolbox.
- From a command window; see Section [7, Using the Bulk Data Importer CLI](#).

The Bulk Data Importer uses the same configuration file as the MI:Toolbox Console: MIToolbox.Console.exe.config. In a default installation, this is:

C:\Program Files\Granta\Granta MI\Toolbox\bin\MIToolbox.Console.exe.config

2 Data format for import

The Bulk Data Importer imports data from a source Microsoft Excel worksheet (file extension .xls, .xlsx or .xlsm) or a comma separated variable (csv) text file (file extensions .txt or .csv).

The data for import is treated as a table. Each row is treated as one record. A new record begins on a new line. Each column (field) is treated as an attribute's data values.

The first four rows in the source file are reserved for format information for the plug-in. The information in these header rows must match the destination database.

In this example worksheet, each row is one record, each column is an attribute:

	A	B	C	D	E	F	G
1	MI_NAME	MI_LONGNAME	Density	Young's Modulus_min	Young's Modulus_max	Standard Deviation	Price
2				MPa	MPa		currency
3						Young's Modulus	
4							
5	Normalized	AISI 4130, Normalized	0.28	29.1	30.6		0.3
6	QT	AISI 4130, Quenched & Tempered	0.28	29	30.5		0.3
7							

Figure 1. Example worksheet for import, each row is one record, each column is an attribute

The Bulk Data Importer cannot create new database schema objects in a GRANTA MI database. Therefore, all attributes, units, discrete types etc. must exist in the database before the import. If necessary, ask the administrator of the GRANTA MI database to edit the database with MI:Admin to prepare it for import.

2.1 Header Rows

Row number	Description
1	Attribute name
2	Units
3	Metadata
4	Hierarchy

2.1.1 Row 1: Attribute name

The first row contains the attributes names. For record properties, use one of the special column names.

For range attributes, use two columns, one for the minimum and one for the maximum values. If the attribute name is suffixed with *_min* or *_max*; or *_1* or *_2*, then the Bulk Data Importer will automatically recognize them as one range attribute. See cells D1 and E1 in Figure 1.

For metadata attributes, the metadata attribute name should be used. See cell F1 in Figure 1.

Hyperlink, File, Multi Value Discrete, and Equations and Logic attributes may use more than two columns to contain their data. For further information, see Attribute name formats that use multiple columns.

More than the required number of columns may be assigned to an attribute. If this is the case, then the columns will be read in the order they are found in the source file, and the first value found will be used to populate the attribute. This also applies when there are multiple columns for an attribute, e.g. for a range attribute, there may be multiple '_min' columns. It does not apply to multi value discrete attributes where all columns are used.

2.1.2 Row 2: Units

The second row will contain the units, i.e. the symbol. If no units are provided the data will be imported in the database units (for attributes with temperature attributes, this will be the absolute unit). See cell D2 in in Figure 1.

For currency data, use 'currency'. The data is assumed to be in the currency for the database, see cell G2 in Figure 1. The database currency can be overridden by adding the three letter ISO 4217 currency code in angled brackets, for example 'currency<AUD>'. The exchange rate from the configuration database is used to convert the imported value into the database currency. The currency unit can be used in combination with other units.

For discrete attributes (including multi value discrete), note that the discrete type is not needed.

2.1.3 Row 3: Metadata

The third row is used to identify metadata attributes. If the attribute is a metadata attribute, then this row contains its parent attribute name. See cell F3 in in Figure 1.

Data for a metadata attribute can only be imported if its parent attribute is also being imported.

2.1.4 Row 4: Hierarchy

The fourth row is used to indicate if the data in the column is to be used to build the hierarchy. It contains an integer that specifies what level of the hierarchy the data in the column is used to create. For example, a '1' means that the column is used to create the first level of the hierarchy, '2' is the second level of the hierarchy, and so on.

More information and alternative methods for creating a record hierarchy are available.

2.2 Data Cells

A record name (MI_NAME) should be unique for that branch of the tree.

To set a data point to 'Not Applicable', the cell should be set to '!NA'. If the cell is blank, then no value will be set.

To set the estimated flag, the numerical value should be enclosed in square brackets, e.g. [1.2].

For range attributes, open ended ranges can be imported by leaving either the minimum or the maximum cell blank.

2.3 *Trailing zeroes*

The Bulk Data Importer can import trailing zeroes for point and range data. There is no special setting in the plug-in to accomplish this. Instead, the number is imported as it is displayed in Excel, which is set by the cell formatting.

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 Bulk Data Importer reads '2.070' for import.

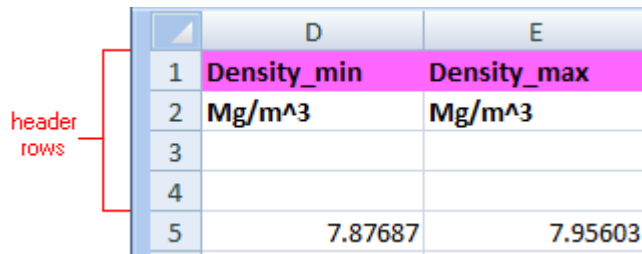
For Number formatting, the cell is displayed with the set number of decimal places. For example, if you type '2.07' and set the number of decimal places to 3, then the cell displays '2.070'. Also, if you type '2.07000' and set the number of decimal places to 3, then the cell displays '2.070'. In both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

The decimal separator is assumed to be a full stop '.' (period).

3 Attribute name formats that use multiple columns

3.1 Range

For range attributes, use two columns, one for the minimum and one for the maximum values. If the attribute name is suffixed with `_min` or `_max`; or `_1` or `_2`, then the Bulk Data Importer will automatically recognize them as one range attribute.



A screenshot of an Excel spreadsheet showing columns D and E. Row 1 has headers 'Density_min' and 'Density_max' in pink. Row 2 has units 'Mg/m^3' and 'Mg/m^3'. Row 5 has values '7.87687' and '7.95603'. A red bracket on the left labels rows 1-2 as 'header rows'.

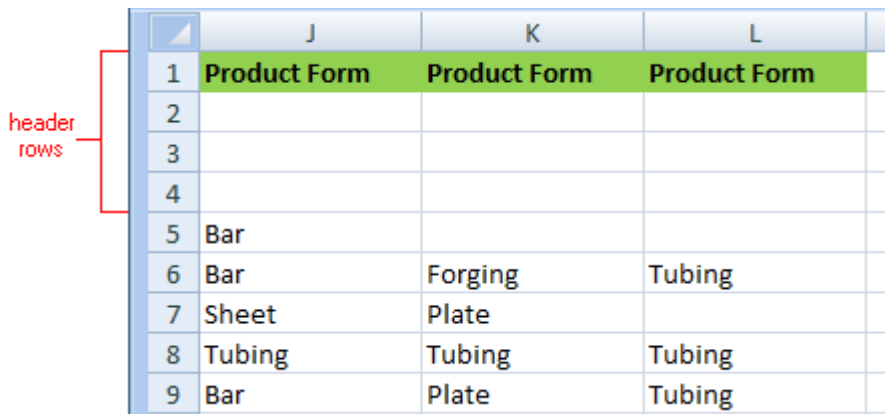
	D	E
1	Density_min	Density_max
2	Mg/m^3	Mg/m^3
3		
4		
5	7.87687	7.95603

Figure 2. Part of an example worksheet for import, showing two columns for a range attribute named 'Density':

3.2 Multi Value Discrete

For multi value discrete attributes, use as many columns as there are values. No attribute name suffix is used, repeat the attribute name for each column.

For example, if Record 1 has one value, Record 2 has two values and Record 3 has three values, you should use three columns for the attribute values.



A screenshot of an Excel spreadsheet showing columns J, K, and L. Row 1 has headers 'Product Form', 'Product Form', and 'Product Form' in green. Rows 5-9 contain data for 'Product Form' across the three columns. Row 5 has 'Bar' in all three. Row 6 has 'Bar', 'Forging', and 'Tubing'. Row 7 has 'Sheet', 'Plate', and an empty cell. Row 8 has 'Tubing', 'Tubing', and 'Tubing'. Row 9 has 'Bar', 'Plate', and 'Tubing'. A red bracket on the left labels rows 1-2 as 'header rows'.

	J	K	L
1	Product Form	Product Form	Product Form
2			
3			
4			
5	Bar		
6	Bar	Forging	Tubing
7	Sheet	Plate	
8	Tubing	Tubing	Tubing
9	Bar	Plate	Tubing

Figure 3. Part of an example worksheet for import, showing three columns for a multi value discrete attribute named 'Product Form'

3.3 Hyperlink

For hyperlink attributes, three columns may be used, containing the address, description, and target values. If the column name has no suffix, it is assumed to contain the address.



	M	N	O
1	Website_link	Website_name	Website_target
2			
3			
4			
5	http://grantadesign.com	Granta Design Ltd	top

Figure 4. Part of an example worksheet for import, showing three columns for a hyperlink attribute named 'Website'

Table 1. Attribute name suffix for hyperlink attributes

Attribute name suffix	Description	Required / Optional
_link	Address (URL) for the attribute.	required
_name	Text description for the address. If absent, the address is used.	optional
_target	Target pane for the file. Valid values (excluding quotes) are: 'blank', '_blank', 'top', '_top', 'parent', '_parent', 'self', '_self', 'newwindow', 'new_window' 'tree', 'content' 'currentframe', 'current_pane'. If absent, 'newwindow' is used.	optional

3.4 File

For file attributes, five columns may be used, containing the file name, description, and target values, and whether the file is indexed and displayed in place of the datasheet.

	H	I	J	K	L
1	Document_filename	Document_description	Document_target	Document_indexed	Document_record
2					
3					
4					
5	C:\DemoFiles\4391 ti data.txt	test data	new_window	TRUE	FALSE

Figure 5. Five columns for a file attribute named 'Document'

Table 2. Attribute name suffix for file attributes

Column	Description	Required / Optional
<attribute>_filename	File name for the attribute, including the path. The path may be an absolute path, or relative to the source file. Use '!FILENAME' excluding quotes to import the current file itself.	required
<attribute>_description	Text description of the file.	optional
<attribute>_target	Target pane for the file. Valid values are as hyperlink (see Table 1). If absent, 'newwindow' is used.	optional
<attribute>_indexed	Indicates whether the contents of the file are indexed for searching. Valid values are TRUE or FALSE (case insensitive). <ul style="list-style-type: none"> If the specified value is a valid Boolean ("true" or "false"), that value will be used for the indexability of the file. If the specified value is not a valid Boolean ("yes", "no", "bananas", etc), the attribute is not imported. If no value is specified (the column is empty), the indexability of the file will be inherited from the attribute. 	optional
<attribute>_record	Indicates whether the file is displayed instead of the datasheet when MI:Viewer is in read mode. Valid options are TRUE or FALSE. If absent, FALSE is used.	optional

3.5 Equations and Logic

	A	B	C	D	E	F	G	H
1	MI_NAME	Fatigue Model	Number of Cycles_min	Number of Cycles_max	Stress Ratio_min	Stress Ratio_max	Temperature_min	Temperature_max
2							°C	°C
3		Fatigue Model	Fatigue Model	Fatigue Model	Fatigue Model	Fatigue Model	Fatigue Model	Fatigue Model
4								
5	Example MFD record	Fatigue model	100	10000000	-1	0.9	23	300

Figure 6. Columns for an equations and logic attribute named 'Fatigue Model': expression name and parameter extents

	I	J	K	L	M	N	O	P
1	Number of Cycles_curve1	Stress Ratio_curve1	Temperature_curve1	Fatigue Model_curve1label	Number of Cycles_curve2	Stress Ratio_curve2	Temperature_curve2	Fatigue Model_curve2label
2			°C				°C	
3	Fatigue Model	Fatigue Model	Fatigue Model	Fatigue Model	Fatigue Model	Fatigue Model	Fatigue Model	Fatigue Model
4								
5	10000000	-1	23	Fully reversible	10000000	0	23	Tension

Figure 7. Columns for an equations and logic attribute named 'Fatigue Model': default curve (curve 1) and a second curve

	Q	R	S
1	Fatigue Model_scale	Fatigue Model_invert	Fatigue Model_xaxis
2			
3			
4			
5	lin	FALSE	Number of Cycles

Figure 8. Columns for an equations and logic attribute named 'Fatigue Model': chart properties

Table 3. Attribute name suffix for equations and logic attributes

Attribute name suffix	Description	Required / Optional
none	The column name without a suffix contains the name of the expression. This must be an exact match to the name of an existing expression in the database table. For example, in Figure 5 the name of the equations and logic attribute is 'Fatigue Model' (cell B1) and the name of the expression is 'Fatigue model' (cell B5).	required
<parameter name>_min	For each parameter assigned to the equations and logic attribute there should be two columns that are used to store the minimum and maximum parameter extents. The _min column contains the minimum parameter extent. For example, cell C1 in Figure 6 contains the minimum extent for the parameter 'Number of Cycles'. The second header row contains the units for the parameter. For example, cell G2 in Figure 6 contains the units for the parameter 'Temperature'. The third header row in a parameter column must contain the name of the equations and logic attribute, for example, cell C3 in Figure 6.	optional

Attribute name suffix	Description	Required / Optional
<parameter name>_max	<p>For each parameter assigned to the equations and logic attribute there should be two columns that are used to store the minimum and maximum parameter extents. The _max column contains the maximum parameter extent. For example, cell F1 in Figure 6 contains the maximum extent for the parameter 'Stress Ratio'.</p> <p>The second header row contains the units for the parameter.</p> <p>The third header row in a parameter column must contain the name of the equations and logic attribute, for example, cell F3 in Figure 6.</p>	optional
<parameter name>_curve1	<p>This column contains the default parameter values. For example, column I contains the default value for the 'Number of Cycles' parameter.</p> <p>The second header row contains the units for the parameter.</p> <p>The third header row in a parameter column must contain the name of the equations and logic attribute, for example, cell I3 in Figure 7.</p>	optional
_curve1label	The label for the default curve. See column L in Figure 7.	optional
<parameter name>_curve2, <parameter name>_curve3, etc.	Other curve parameter values, in the same format as curve 1.	optional
_curve2label, _curve3label, etc.	The labels for the other curves, in the same format as the label for curve 1.	optional
_scale	<p>Indicates if the attribute axis is set to a logarithmic or linear scale, see cell. Valid values (excluding quotes) are:</p> <p>'log'</p> <p>'lin'</p> <p>If the cell is blank, the default value is 'lin'. For example, see column Q in Figure 8.</p>	required
_invert	Indicates if the axes on the graph are inverted by default. Valid values are TRUE or FALSE. If the cell is blank, the default value is FALSE. For example, see column R in Figure 8.	required
_xaxis	Set the default x-axis parameter. It must be the name of a valid parameter. The cell may be blank. For example, see column S in Figure 8.	required

4 Special column names

A GRANTA MI database distinguishes between record 'properties' and 'attributes'. Properties identify records within the database, which allows the records to be displayed on the screen (usually in a hierarchical format on the table tree) — all records in a GRANTA MI database contain properties. Most records also contain attributes — these are used to store data and for selection in MI:Viewer. Since ordinary databases do not have this distinction, the Bulk Data Importer relies on specially named columns in the source data file to distinguish between these two types of data (properties and attributes). The special column names are as follows:

- MI_IDENTITY
- MI_PARENT
- MI_HASDATA
- MI_ISFOLDER
- MI_NAME
- MI_CODE
- MI_COLOR
- MI_LONGNAME
- MI_GRUID
- MI_GUID

Use of the special columns is optional, although Granta recommends that MI_NAME is present.

If a special column is present, but does not contain any data for an individual row, that row will not be imported as a record, and an error will be reported.

Table 4. Special column names

Name	Description	Type
MI_IDENTITY	Used to uniquely identify each record, use this column together with the PARENT column to allow a hierarchical structure to be imported. For example, when importing a database of parts the 'PART NO.' column could be renamed to 'IDENTITY' to enable the part number to uniquely identify the part record during importing.	Number
MI_PARENT	Used when importing a hierarchical structure, this column refers to the identity (IDENTITY) of the record that is the parent of this record in the hierarchical structure. For example if you were to import a table of animals, then the apes record might have a reference to the mammals record. This column should be the same type as the IDENTITY column. If this column is empty then the Bulk Data Importer places the record at the root level.	Number
MI_HASDATA	Used to distinguish between records with attribute data and records without attribute data. Records with attribute data should set this value to TRUE or 1. If this value is not set, and the Bulk Data Importer detects that the record contains data, it sets the default value for MI_HASDATA to true and the default value for MI_ISFOLDER to false.	Number (0/1) or Boolean (TRUE/FALSE)

Name	Description	Type																		
MI_ISFOLDER	Used when importing a hierarchical structure. Records used as folders should set this value to TRUE or 1. If this value is not set, and the Bulk Data Importer detects that the record does not contain data, it sets the default value for MI_HASDATA to false and the default value for MI_ISFOLDER to true.	Number (0/1) or Boolean (TRUE/FALSE)																		
MI_NAME	The name of the record as shown in the tree. (Also referred to as the 'short name' or 'tree name'.)	Text																		
MI_CODE	A short code for the record.	Text																		
MI_COLOR	<p>The color of a record is used to shade record icons in the tree and graphs. This allows the user to quickly identify record groupings. This integer field should be one of the following values.</p> <table> <tr> <td>0: Red</td><td>6: Maroon</td><td>11: Teal</td></tr> <tr> <td>1: Lime</td><td>7: Green</td><td>12: Gray</td></tr> <tr> <td>2: Blue</td><td>8: Navy</td><td>13: Black</td></tr> <tr> <td>3: Yellow</td><td>9: Olive</td><td>14: White</td></tr> <tr> <td>4: Fuchsia</td><td>10: Purple</td><td>15: Silver</td></tr> <tr> <td>5: Aqua</td><td></td><td></td></tr> </table>	0: Red	6: Maroon	11: Teal	1: Lime	7: Green	12: Gray	2: Blue	8: Navy	13: Black	3: Yellow	9: Olive	14: White	4: Fuchsia	10: Purple	15: Silver	5: Aqua			Number (0–15)
0: Red	6: Maroon	11: Teal																		
1: Lime	7: Green	12: Gray																		
2: Blue	8: Navy	13: Black																		
3: Yellow	9: Olive	14: White																		
4: Fuchsia	10: Purple	15: Silver																		
5: Aqua																				
MI_LONGNAME	The full name of the record.	Text																		
MI_GRUID	An identifier that must be unique within the table. This should only be used in consultation with Granta.	Text																		
MI_GUID	An identifier that must be unique within the database. This should only be used in consultation with Granta.	Valid GUID																		

5 Record hierarchy

Records can be imported with a hierarchy in two ways:

1. Using the MI_IDENTITY and MI_PARENT special columns.
2. Using data values for each record.

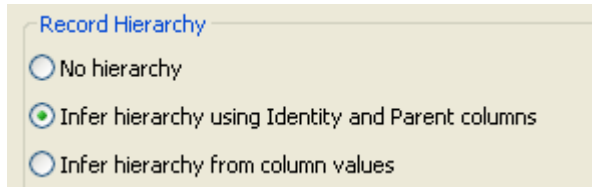


Figure 9. The record hierarchy options in the Bulk Data Importer plug-in in MI:Toolbox

The examples below shows how the two methods would create the hierarchy shown in Figure 10.

A generic record is created if the record is a folder and contains data. See the 'Bar' record in the examples below.

If a record already exists with the same tree name in the same position in the hierarchy, a [Duplicate Record Name](#) dialog will give options for proceeding. This will be logged to the log file.

The user also has the option to import the data with no hierarchy.

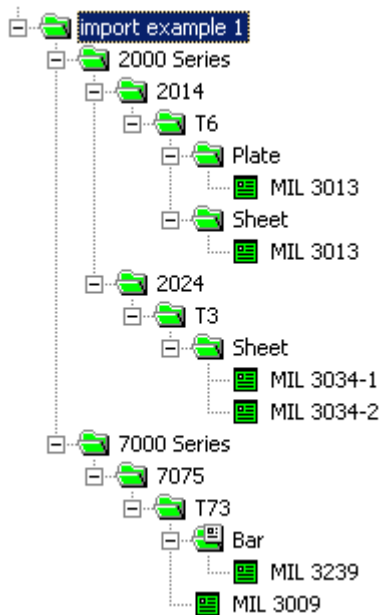


Figure 10. The record hierarchy, as shown in MI:Toolbox. The data in the source file is imported into the 'import example 1' folder. Both of the examples below result in the same hierarchy.

5.1 Special columns

This option will only be available if the MI_IDENTITY and MI_PARENT special columns are in the data file.

Every folder and generic record used to create the hierarchy must be present as a row in the file for this method to work. For anything more than a simple hierarchy, this can be very difficult for the user to construct.

If a parent value is unspecified, the record will be placed at the root level of the import, that is, the level below the 'Import Location'.

In example 1, the hierarchy is constructed with the MI_IDENTITY and MI_PARENT special columns (highlighted columns in Figure 11).

	A	B	C	D	E	
1	MI_NAME	MI_IDENTITY	MI_PARENT	MI_ISFOLDER	Density	
2					lb/in^3	
3						
4						
5	2000 Series	1		TRUE		
6	2014	2	1	TRUE		
7	T6	3	2	TRUE		
8	Plate	4	3	TRUE		
9	Sheet	5	3	TRUE		
10	2024	6	1	TRUE		
11	T3	7	6	TRUE		
12	Sheet	8	7	TRUE		
13	7000 Series	9		TRUE		
14	7075	10	9	TRUE		
15	T73	11	10	TRUE		
16	Bar	12	11	TRUE	0.1	
17	MIL 3013	13	4	FALSE	0.1	
18	MIL 3013	14	5	FALSE	0.1	
19	MIL 3034-1	15	8	FALSE	0.1	
20	MIL 3034-2	16	8	FALSE	0.1	
21	MIL 3239	17	12	FALSE	0.1	
22	MIL 3009	18	11	FALSE	0.1	
23						

Figure 11. Example 1 - worksheet for import with special columns (highlighted)

5.2 Data values

For this option to be used, the data must be in each row along with the attribute values.

The columns to be used can be pre-defined in the header rows of the source file, or set at the time of import in the plug-in. The fourth row of the Excel sheet is reserved for defining which level of the hierarchy (if any) the data in the column will be used for.

In example 2, the hierarchy is constructed using the information in the fourth header row (highlighted in Figure 12) and the data values in the specified columns (B, C, D and E). The fourth row

shows that 'Alloy Class' is 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.

If a record has no data for a column being used to create the hierarchy, the record is placed in the preceding level of the hierarchy. In the example, the 'MIL 3009' record will be imported into the 'T73' folder.

	A	B	C	D	E	F	
1	MI_NAME	Alloy Class	Alloy Name	Heat Treatment	Form	Density	
2						lb/in^3	
3							
4		1	2	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							

Figure 12. Example 2 - worksheet for import, record hierarchy inferred from column values using the fourth header row (highlighted)

Example 3 contains the same data as example 2, but there is no information in the header rows (Figure 12) and the hierarchy is created by selecting the columns in the plug-in (Figure 13).

	A	B	C	D	E	F	
1	MI_NAME	Alloy Class	Alloy Name	Heat Treatment	Form	Density	
2						lb/in^3	
3							
4							
5	MIL 3013	2000 Series	2014	T6	Plate	0.1	

Figure 13. Example 3 - worksheet for import, without using a fourth header row

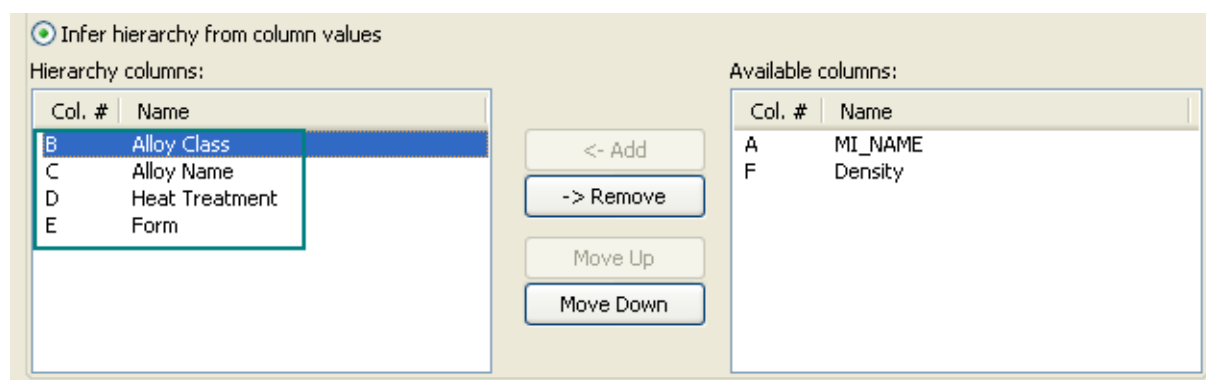


Figure 14. Example 3 - Record hierarchy inferred from column values by selecting the columns in the Bulk Data Importer plug-in

6 Attribute data values

Data type	Format
Numerical Data for Range and Point attributes	<p>Data values are a number, with a maximum of seven significant figures. Point data occupies one field; Range data occupies two fields. Range and Point data can have units.</p> <p>Data values can be entered as an integer e.g. 456, a decimal e.g. 123.456, or formatted with an exponent e.g. 456e+010 or 123.456e+009.</p> <p>Data values are stored as single precision floating point numbers and should range in value from:</p> <p>-1e+038 to -1e-039 for negative values</p> <p>1e-039 to 1e+038 for positive values.</p> <p>Trailing zeroes can be imported for point data and range data. For more information, see Section 2.3.</p>
Numerical Data for Integer attributes	<p>Data values are a number. Integer data occupies one field. Integer data cannot be assigned a unit.</p> <p>Data values are entered as an integer e.g. 456.</p> <p>Data values are stored as long integer numbers and should range in value from:</p> <p>-2,147,483,648 to 2,147,483,647</p>
Numerical Data for Float Functional attributes	<p>It is not possible to import Float functional data using the Bulk Data Importer plug-in.</p> <p>Use the Text Importer or Excel Importer plug-ins.</p>
Data for Discrete Functional attributes	<p>It is not possible to import Discrete functional data using the Bulk Data Importer plug-in.</p> <p>Use the Text Importer or Excel Importer plug-ins.</p>
Data for Equations and Logic attributes	<p>The expression name is text string. The numerical data values for parameters are as range and point attributes. The parameters must already exist in the database. Other information about the chart can also be set.</p>
Data for Tabular attributes	<p>It is not possible to import tabular data using the Bulk Data Importer plug-in. Use the Excel Importer plug-in instead.</p>
Discrete Data	<p>Discrete data values are case sensitive text. The discrete type and value must already exist in the database.</p>

Data type	Format
Logical Data	Data values can be a number (0 or 1) or a Boolean (TRUE or FALSE). No units can be assigned to logical data.
Date Data	In Excel, data stored in the DateTime format will be imported correctly. The success of importing data in any other format is culture dependent.
Short Text	A maximum of 255 characters is allowed. No units can be assigned to short text data.
Long Text	A maximum of 2 GB per attribute per record is allowed. No units can be assigned to long text data. In the case of a text attribute in text file, if the text string contains the text separator e.g. a comma ",", quote marks (" ") should be put around the text string.
Link Data	It is not possible to import Link data using the Bulk Data Importer.
Picture Data	The 'value' for Picture data is the file name, including the path. The path may be absolute, or relative to the source file.
Hyperlink Data	The 'value' for Hyperlink data is a valid URL, consisting of a protocol and location e.g. http://www.grantadesign.com as Long Text. Other information about the hyperlink, the description and the target, can also be set.
File Data	The 'value' for File data is the file name, including the path. The path may be absolute, or relative to the source file. Use '!FILENAME' excluding quotes to import the current file itself. Other information about the file can also be set.
No Data	An Attribute data value can also be left blank and it will be assumed to have no value.

7 Using the Bulk Data Importer CLI

This section describes how to run the Bulk Data Importer plug-in from the command line, allowing bulk imports to be run as a scheduled task. Note that it is not possible to select individual attributes for import when running the BDI via the command line—all attributes and record properties in the source file will be imported.

The Bulk Data Importer executable is located in the *bin* folder in the MI:Toolbox installation folder, typically:

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

To run the Bulk Data Importer from the command line, at the command prompt, type the following:

```
MIToolbox.Console.exe -plugin Granta.BDI
```

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

Markdown in long text attributes

Note that when importing Long Text attributes, if the value of an attribute to be imported starts with the text `#markdown` followed by a newline, the data will be interpreted as Markdown-formatted, and displayed as such in MI:Viewer. (Note that the newline after `#markdown` is important, otherwise you'll just see the text `"#markdown"` at the beginning of your attribute value).

7.1 Command-line arguments

The Bulk Data Importer requires a number of command-line arguments. Arguments containing spaces, colons, or equals should be double quoted e.g.

```
-username "Peter Parker"
```

```
-inputfile "C:\Users\ali.kim\Documents\data imports\SS import.xls"
```

Argument	Required?	Description
-plugin Granta.BDI	Yes	Specifies the name of the plug-in.
-server <servername>	Yes	Specifies the hostname of the server where the GRANTA MI Service is hosted.
-dbkey <key>	Yes	The database key of the database into which the data will be imported.
-table <tablename>	Yes	The name of the table into which the data will be imported.
-inputfile <filepath>	Yes	<p>The name and location of the file containing the data to be imported. For example:</p> <p>-inputfile "C:\Users\ali.kim\Documents\SS import.xls" -inputfile "D:\Data\Tensile Test 1.txt"</p> <p>For Excel workbooks that contain more than one worksheet, the worksheet must also be specified using -worksheet.</p> <p>For text files, you can also specify the column and text delimiter characters, if required; see -delimiter, -textdelimiter.</p>
-inputencoding <encoding>	Yes	<p>(Required for text input files) The character encoding of the file to be imported when importing from a text file. Case-sensitive. Examples:</p> <p>-inputencoding utf-8 -inputencoding us-ascii -inputencoding iso-8859-1</p>
-conflictresolution <value>	No	<p>Specifies how record naming conflicts should be dealt with. The values are case sensitive. Permitted values:</p> <ul style="list-style-type: none"> • Replace Replace the existing record with the new record. All links associated with the record are also deleted. • ReplaceData The existing record is kept and the incoming data is added to the record. Where there is existing data for an attribute, this will be overwritten if the Bulk Data Importer is importing new data. If there is no new data for an attribute, existing data remains unchanged. • Rename The imported record will be renamed. (Default) • DoNotImport The record is not imported and the Bulk Data 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 this argument is omitted, the default action is Rename.</p>

Argument	Required?	Description
-delimiter <char>	No	Specifies the character used to delimit columns of data when importing from a text file. For example, to use a period (.) as the column delimiter: -delimiter "." If this argument is omitted, the default delimiter character is a comma (,).
-help	No	Displays usage information for the application. If the plug-in is also specified, it will include the usage information of the specified plug-in
-hierarchy <type>	No	Specifies how the data hierarchy will be created on import. Permitted values: <ul style="list-style-type: none"> • Flat No hierarchy should be inferred. • Ordered The hierarchy should be inferred using the values in named columns. • IdentityAndParent The hierarchy should be inferred using values in the MI_IDENTITY and MI_PARENT columns. If this argument is omitted, the Bulk Data Importer will try to interpret the hierarchy from the available data in the order: IdentityAndParent, Ordered, Flat.
-logfile <filepath>	No	The name and location of the log file. If omitted, no log file will be generated.
-loglevel <level>	No	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.
-mergedelimiters	No	Specifies that adjacent column delimiters should be merged into a single delimiter when importing from a text file.
-nologo	No	Prevents the display of application information on startup.

Argument	Required?	Description
-record <name>	No	<p>Identifies one or more locations in table tree hierarchy, under which the new records will be placed. Values may be the folder path of a record in the tree, or a record GUID:</p> <ul style="list-style-type: none"> The location in the tree folder structure may be specified with the full folder path starting with the name of the top-level folder and ending with the name of the chosen record. For example: -record Metals Non-ferrous Aluminum Wrought "7000 Series" The recordGUID may be specified as 32 continuous digits, or as groups of 8, 4, 4, 4, and 12 digits separated by hyphens. For example: -record 00000026-000e-4fff-8fff-dd92ffff0000 <p>If this argument is omitted from the command line, then new records will be placed at root record level.</p>
-subset	No	A list of the subsets to which the imported records should be added.
-textdelimiter <char>	No	Specifies the character used to delimit text data when importing from a text file. If omitted, the default value is a double quote character (").
-username <username> -domain <domainname> -password <password>	No*	<p>(*Optional only when using Windows authentication)</p> <p>The authentication details used to connect to the GRANTA MI Service. A user account with at least write privileges to the database is required.</p> <p>These may be omitted from the command line if Windows authentication is being used; in this case, the import is performed with the user account used to log on to the MI:Toolbox host. Alternatively, you can perform the import under a different Windows user account by specifying the credentials of that account.</p>
-worksheet <name>	No	<p>When importing data from an Excel workbook that contains more than one worksheet, this specifies the name of the worksheet to import. May be omitted if the specified input workbook only contains one worksheet.</p> <p>Example: -worksheet "Tensile Data"</p>

7.2 CLI usage examples

```
MIToolbox.Console.exe -plugin Granta.BDI -nologo -server localhost -username "kim.smith" -password Password123 -domain ACMECORP -dbkey MI_Training -table MaterialUniverse -record Import1 -subset "All Bulk Materials" Metals -inputfile "C:\Users\kim.smith\Documents\data\SS import.xls" -worksheet "Stainless Steels 1" -logfile "C:\Users\kim.smith\Documents\BDIlog1.log" -hierarchy Flat
```

```
MIToolbox.Console.exe -plugin Granta.BDI -logfile "C:\Users\djones\Documents\Console.log" -nologo -server localhost -dbkey MI_Starter -table "Tensile Test Data" -subset "Tensile Test Data" -record Import2 -inputfile "C:\Users\djones\Data\Tensile.xls" -worksheet "Stainless Steels 2" -hierarchy IdentityAndParent
```

```
MIToolbox.Console.exe -plugin Granta.BDI -logfile "C:\Users\alex\logs\log1.log" -nologo -server localhost -dbkey MI_Training -table MaterialUniverse -inputfile "C:\Users\alex\files\SS import.xls" -worksheet "Stainless Steels 3" -hierarchy Ordered
```

```
MIToolbox.Console.exe -plugin Granta.BDI -nologo -logfile "C:\Users\pat.dale\logs\importlog1.log" -server localhost -username "pat.dale" -password pwd123 -domain ACMECO -dbkey MI_Training -table MaterialUniverse -record Import4 -subset "All Bulk Materials" Metal -inputfile "C:\Users\pat.dale\Files\ssimport.txt" -inputencoding utf-8 -delimiter | -mergedelimiters -hierarchy IdentityAndParent
```


8 Troubleshooting

The following exit codes may help you to troubleshoot problems that may arise running the Bulk Data Importer from the command line.

8.1 MI:Toolbox general errors

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

8.2 Bulk Data Importer errors

Code	Description
2000	Ignored columns in the input file - Indicates that there are one or more columns in the input file that are unused.
2001	Failed to create input filter - Indicates that an input filter for the specified input file could not be created. This may be because the file type of the input file was not recognized (identification of file type is based on file extension, e.g. .txt, .csv, .xls).
2002	Input filter configuration failed - Indicates that the configuration of the input filter failed (e.g. setting the worksheet in a spreadsheet, setting unsupported delimiter characters for CSV text files).
2003	Failed to open import log file - Indicates an error when trying to open the log file for the import process.
2004	Cannot set record hierarchy inference mode - Unable to set the hierarchy inference mode. For example, you cannot force the ParentAndIdentity inference mode if the input file does not have identity and parent columns.
2005	Data value not imported - One or more data values were not imported.
2006	Data value possibly imported incorrectly - One or more data values were possibly incorrectly imported. For example, a text value may have been truncated.
2007	Commit failed - A commit failed, indicating that one or more records were not correctly imported into the database.
2008	Import abandoned - The import process was prematurely aborted.
2009	Record not imported - One or more records were not imported.
2010	Record property possibly imported incorrectly - One or more record properties were possibly incorrectly imported. For example, an invalid record color may have been specified, so the actual color of the record created was inherited from its parent folder.