Oracle® Retail Macro Space Planning

Data Importer User Guide
Release 13.3.1

E41950-01

July 2013

Oracle® Retail Macro Space Planning Data Importer User Guide, Release 13.3.1

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Primary Author: Philip Wells

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Value-Added Reseller (VAR) Language

Oracle Retail VAR Applications

The following restrictions and provisions only apply to the programs referred to in this section and licensed to you. You acknowledge that the programs may contain third party software (VAR applications) licensed to Oracle. Depending upon your product and its version number, the VAR applications may include:

(i) the **MicroStrategy** Components developed and licensed by MicroStrategy Services Corporation (MicroStrategy) of McLean, Virginia to Oracle and imbedded in the MicroStrategy for Oracle Retail Data Warehouse and MicroStrategy for Oracle Retail Planning & Optimization applications.

(ii) the **Wavelink** component developed and licensed by Wavelink Corporation (Wavelink) of Kirkland, Washington, to Oracle and imbedded in Oracle Retail Mobile Store Inventory Management.

(iii) the software component known as **Access Via**™ licensed by Access Via of Seattle, Washington, and imbedded in Oracle Retail Signs and Oracle Retail Labels and Tags.

(iv) the software component known as **Adobe Flex**™ licensed by Adobe Systems Incorporated of San Jose, California, and imbedded in Oracle Retail Promotion Planning & Optimization application.

You acknowledge and confirm that Oracle grants you use of only the object code of the VAR Applications. Oracle will not deliver source code to the VAR Applications to you. Notwithstanding any other term or condition of the agreement and this ordering document, you shall not cause or permit alteration of any VAR Applications. For purposes of this section, "alteration" refers to all alterations, translations, upgrades, enhancements, customizations or modifications of all or any portion of the VAR Applications including all reconfigurations, reassembly or reverse assembly, re-engineering or reverse engineering and recompilations or reverse compilations of the VAR Applications or any derivatives of the VAR Applications. You acknowledge that it shall be a breach of the agreement to utilize the relationship, and/or confidential information of the VAR Applications for purposes of competitive discovery.

The VAR Applications contain trade secrets of Oracle and Oracle's licensors and Customer shall not attempt, cause, or permit the alteration, decompilation, reverse engineering, disassembly or other reduction of the VAR Applications to a human perceivable form. Oracle reserves the right to replace, with functional equivalent software, any of the VAR Applications in future releases of the applicable program.

Contents

Send Us Your Comments

Oracle Retail Macro Space Planning, Data Importer User Guide, Release 13.3.1

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document.

Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

> **Note:** Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the Online Documentation available on the Oracle Technology Network Web site. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: retail-doc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at www.oracle.com.

Preface

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Retail Macro Space Planning Release 13.3.1 documentation set:

- *Oracle Retail Macro Space Management Administration User Guide*
- *Oracle Retail Macro Space Management Administration Module Online Help*
- *Oracle Retail Macro Space Management Configuration User Guide*
- *Oracle Retail Macro Space Management Configuration Module Online Help*
- *Oracle Retail Macro Space Management Fixture Studio User Guide*
- *Oracle Retail Macro Space Management Fixture Studio Online Help*
- *Oracle Retail Macro Space Management Product Studio User Guide*
- *Oracle Retail Macro Space Management Product Studio Online Help*
- *Oracle Retail Macro Space Management Store Planning User Guide*
- *Oracle Retail Macro Space Management Store Planning Online Help*
- *Oracle Retail Macro Space Management Report Designer User Guide*
- *Oracle Retail Macro Space Management Report Designer Online Help*

- *Oracle Retail Macro Space Management Release Notes*
- *Oracle Retail Macro Space Planning Installation Guide*
- *Oracle Retail Macro Space Planning License Information*
- *Oracle Retail Macro Space Planning Data Model*

For more information on In-Store Space Collaboration see the following documents in the Oracle Retail In-Store Space Collaboration Release 13.3.1 documentation set:

- *Oracle Retail In-Store Space Collaboration Release Notes*
- *Oracle Retail In-Store Space Collaboration User Guide*
- *Oracle Retail In-Store Space Collaboration Online Help*

Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:

https://support.oracle.com

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received
- Screen shots of each step you take

Review Patch Documentation

When you install the application for the first time, you install either a base release (for example, 13.3) or a later patch release (for example, 13.3.1). If you are installing the base release or additional patch releases, read the documentation for all releases that have occurred since the base release before you begin installation. Documentation for patch releases can contain critical information related to the base release, as well as information about code changes since the base release.

Oracle Retail Documentation on the Oracle Technology Network

Documentation is packaged with each Oracle Retail product release. Oracle Retail product documentation is also available on the following Web site:
http://www.oracle.com/technology/documentation/oracle_retail.html

(Data Model documents are not available through Oracle Technology Network. These documents are packaged with released code, or you can obtain them through My Oracle Support.)

Documentation should be available on this Web site within a month after a product release.

Conventions

Navigate: This is a navigate statement. It tells you how to get to the start of the procedure and ends with a screen shot of the starting point and the statement "the Window Name window opens."

```
This is a code sample
    It is used to display examples of code
```

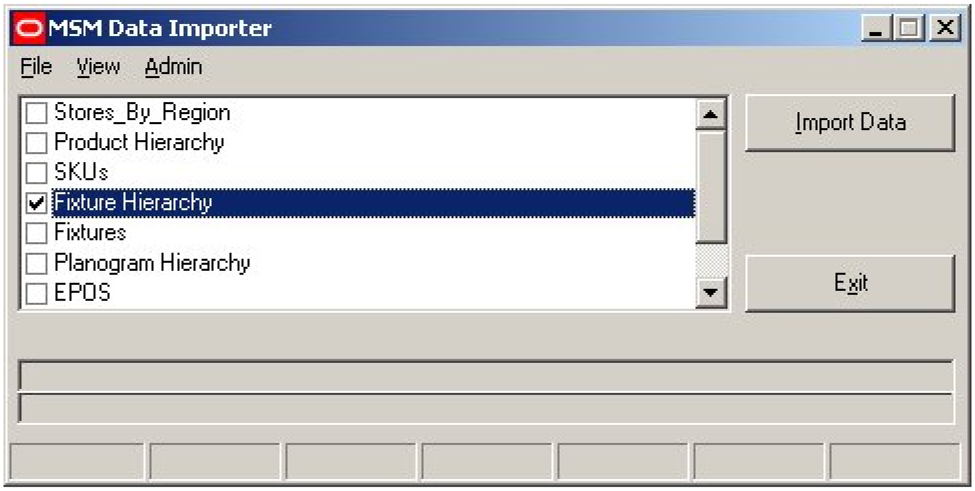
1

Overview of Data Importer

What is Data Importer?

> **Important**: Data Importer can be used at several levels. It is possible for a normal Macro Space Management user to run a pre-configured import by selecting a few options in the Data Importer GUI. It is also possible for a user with Administrator's rights to configure the basic parameters determining how a specific import of data works. Users configuring the way data imports need a understanding of the Macro Space Management database schema and a good knowledge of how to write SQL statements. If you do not have these two skills, Oracle STRONGLY recommends that you do not attempt to configure the Data Import module.

Data Importer allows users to import data into Macro Space Management.



- □ Stores_By_Region
- □ Product Hierarchy
- □ SKUs
- ☑ Fixture Hierarchy
- □ Fixtures
- □ Planogram Hierarchy
- □ EPOS

It takes data in one of the three following formats:

- CSV (Comma Separated Value)
- Spreadsheet
- XML File (Extensible Mark-up Language)

and inserts the data into the central Macro Space Management database.

The data to be imported must be in a rigidly defined format that is not changed:

Fixture Name	Fixture Length	Fixture Depth	Fixture Height	File Name
Back Panel-1	600	10	1250	Back_panel_1.dwg
Back Panel-2	600	10	1500	Back_panel_1.dwg

Fixture Name	Fixture Length	Fixture Depth	Fixture Height	File Name
Back Panel-3	800	10	1800	Back_panel_1.dwg

Data Importer will then put this into very specific places within the central Macro Space Management database. For example, this information might go into the AVTTB_BLOCK_DEFINITION table.

This imported data then becomes usable in Macro Space Management.

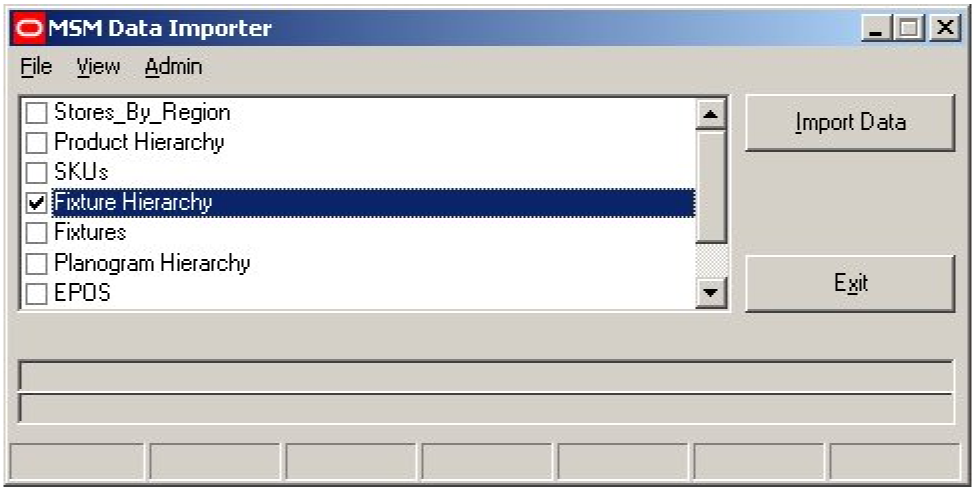
Using Data Importer

There are 3 basic levels of using Data Importer:

- Via the GUI
- Via the Command Line
- Configuring Data Importer

Via the GUI

Standard users operating the Data Importer via the Graphical User Interface (GUI) can select the data to import by means of the check boxes, click on the Import Data button and then check the error logs to confirm correct import of data.



Via the Command Line

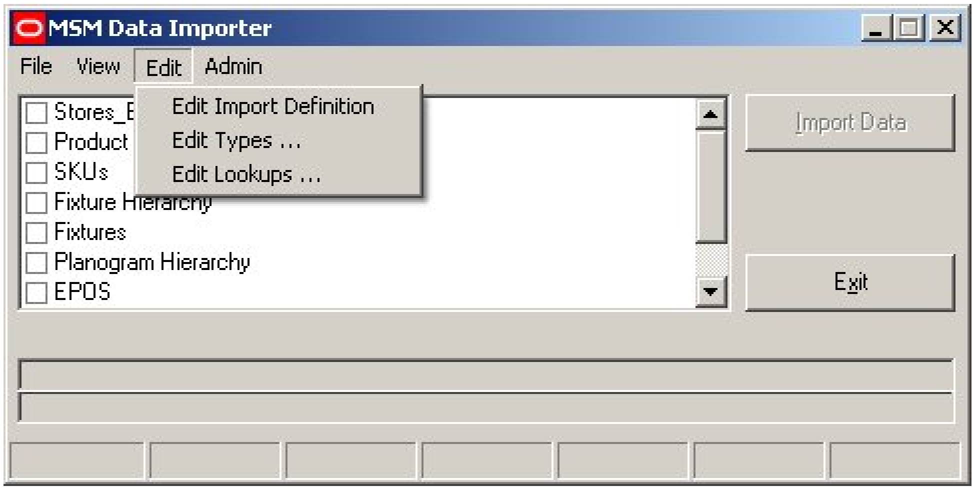
Users operating via the command line can use switches to determine which data to import.

This form of operation is usually used in conjunction with automatic scheduling so data import can be carried out automatically overnight.

Configuring Data Importer

Users can configure the exact manner data is imported using the Data Importer. To do this you must have Administrators rights.

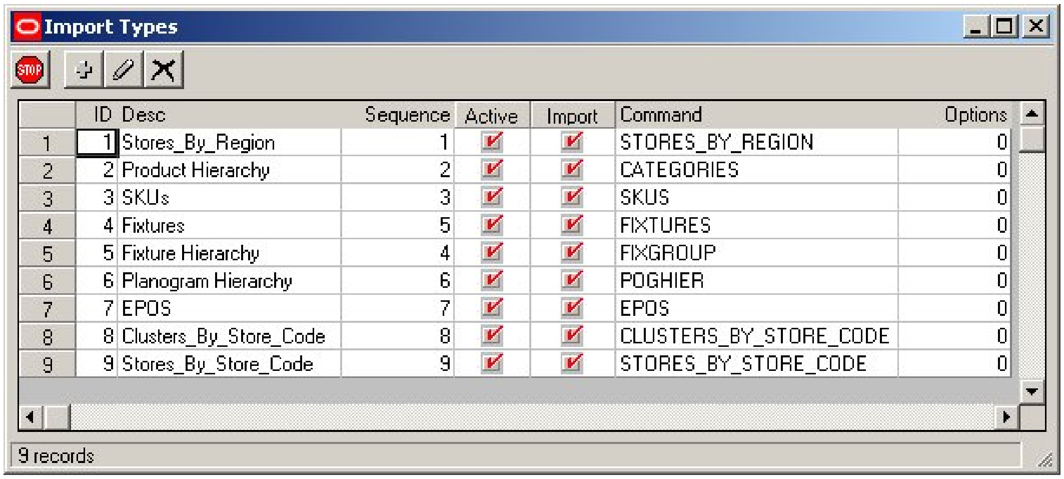
Using <Ctrl + A> will toggle an additional menu into view.



This allows users to modify Import Types, Look-up Values and Import Definitions.

The Import Types Dialogue Box

The **Import Types dialogue box** (available to Admin Users only) allows users to set up lists of basic Admin types.

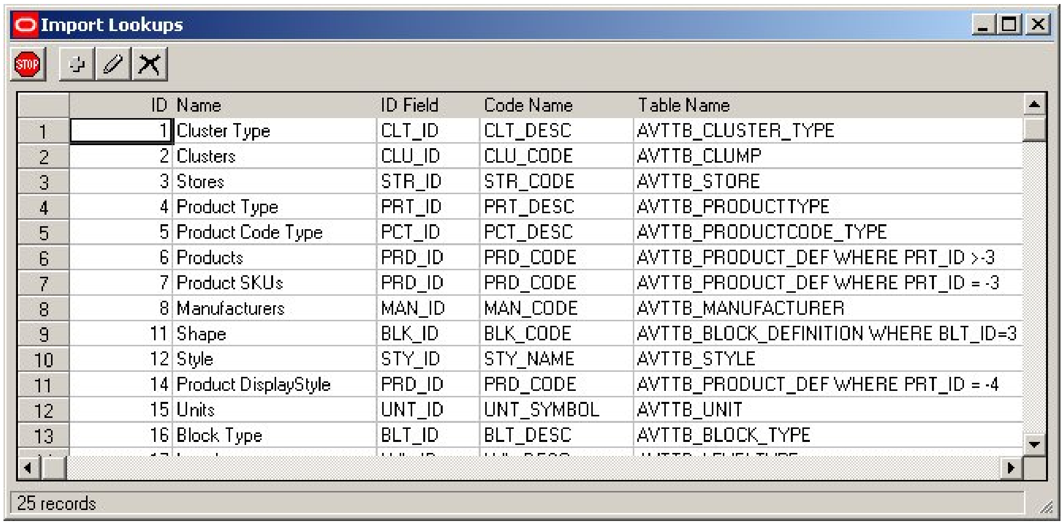


	ID	Desc	Sequence	Active	Import	Command	Options
1	1	Stores_By_Region	1	☑	☑	STORES_BY_REGION	0
2	2	Product Hierarchy	2	☑	☑	CATEGORIES	0
3	3	SKUs	3	☑	☑	SKUS	0
4	4	Fixtures	5	☑	☑	FIXTURES	0
5	5	Fixture Hierarchy	4	☑	☑	FIXGROUP	0
6	6	Planogram Hierarchy	6	☑	☑	POGHIER	0
7	7	EPOS	7	☑	☑	EPOS	0
8	8	Clusters_By_Store_Code	8	☑	☑	CLUSTERS_BY_STORE_CODE	0
9	9	Stores_By_Store_Code	9	☑	☑	STORES_BY_STORE_CODE	0

9 records

The Lookups Dialogue Box

The **Lookups dialogue box** allows users to define Look-ups



Import Lookups

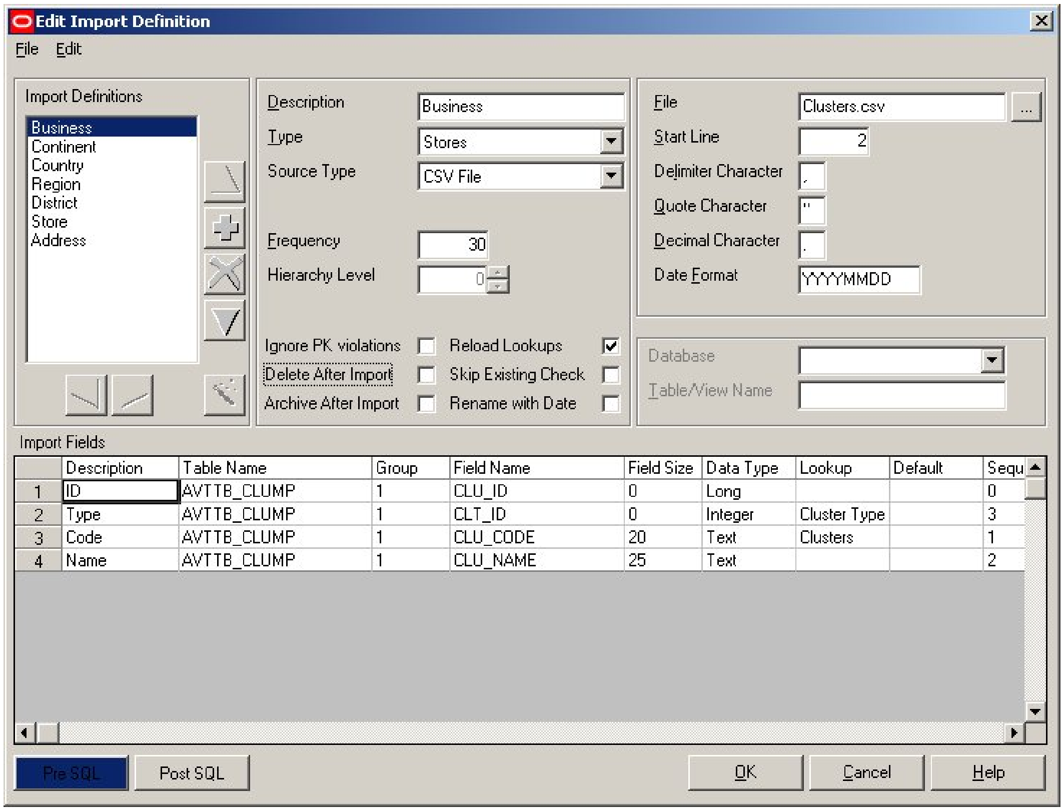
	ID	Name	ID Field	Code Name	Table Name
1	1	Cluster Type	CLT_ID	CLT_DESC	AVTTB_CLUSTER_TYPE
2	2	Clusters	CLU_ID	CLU_CODE	AVTTB_CLUMP
3	3	Stores	STR_ID	STR_CODE	AVTTB_STORE
4	4	Product Type	PRT_ID	PRT_DESC	AVTTB_PRODUCTTYPE
5	5	Product Code Type	PCT_ID	PCT_DESC	AVTTB_PRODUCTCODE_TYPE
6	6	Products	PRD_ID	PRD_CODE	AVTTB_PRODUCT_DEF WHERE PRT_ID >-3
7	7	Product SKUs	PRD_ID	PRD_CODE	AVTTB_PRODUCT_DEF WHERE PRT_ID = -3
8	8	Manufacturers	MAN_ID	MAN_CODE	AVTTB_MANUFACTURER
9	11	Shape	BLK_ID	BLK_CODE	AVTTB_BLOCK_DEFINITION WHERE BLT_ID=3
10	12	Style	STY_ID	STY_NAME	AVTTB_STYLE
11	14	Product DisplayStyle	PRD_ID	PRD_CODE	AVTTB_PRODUCT_DEF WHERE PRT_ID = -4
12	15	Units	UNT_ID	UNT_SYMBOL	AVTTB_UNIT
13	16	Block Type	BLT_ID	BLT_DESC	AVTTB_BLOCK_TYPE

25 records

Look ups allow users to convert one form of data being imported into another - for example the name of a cluster might be converted into its (numerical) primary key.

The Import Definition Dialogue Box

The **Import Definition dialogue box** allows users to configure Import Definitions.



Edit Import Definition

Import Definitions
- Business
- Continent
- Country
- Region
- District
- Store
- Address

Description	Business
Type	Stores
Source Type	CSV File
Frequency	30
Hierarchy Level	0
File	Clusters.csv
Start Line	2
Delimiter Character	,
Quote Character	"
Decimal Character	.
Date Format	YYYYMMDD
Database	
Table/View Name	

Ignore PK violations ☐ Reload Lookups ☑
Delete After Import ☐ Skip Existing Check ☐
Archive After Import ☐ Rename with Date ☐

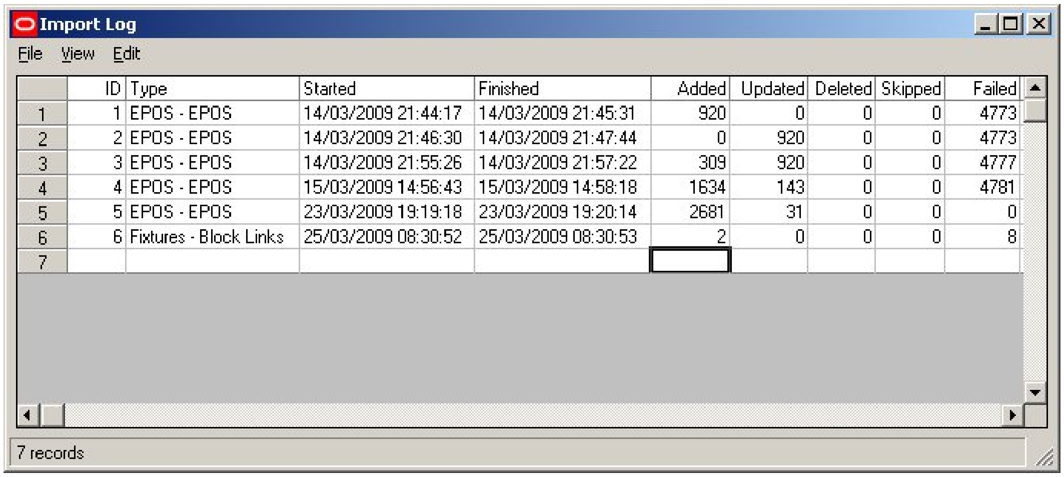
Import Fields

	Description	Table Name	Group	Field Name	Field Size	Data Type	Lookup	Default	Sequ
1	ID	AVTTB_CLUMP	1	CLU_ID	0	Long			0
2	Type	AVTTB_CLUMP	1	CLT_ID	0	Integer	Cluster Type		3
3	Code	AVTTB_CLUMP	1	CLU_CODE	20	Text	Clusters		1
4	Name	AVTTB_CLUMP	1	CLU_NAME	25	Text			2

This allows users to configure how data is extracted from the source file (in this example Fixtures.csv) and inserted into the database.

The Import Log Dialogue Box

The **Import Log dialogue box** allows users to see the results of recent data imports.

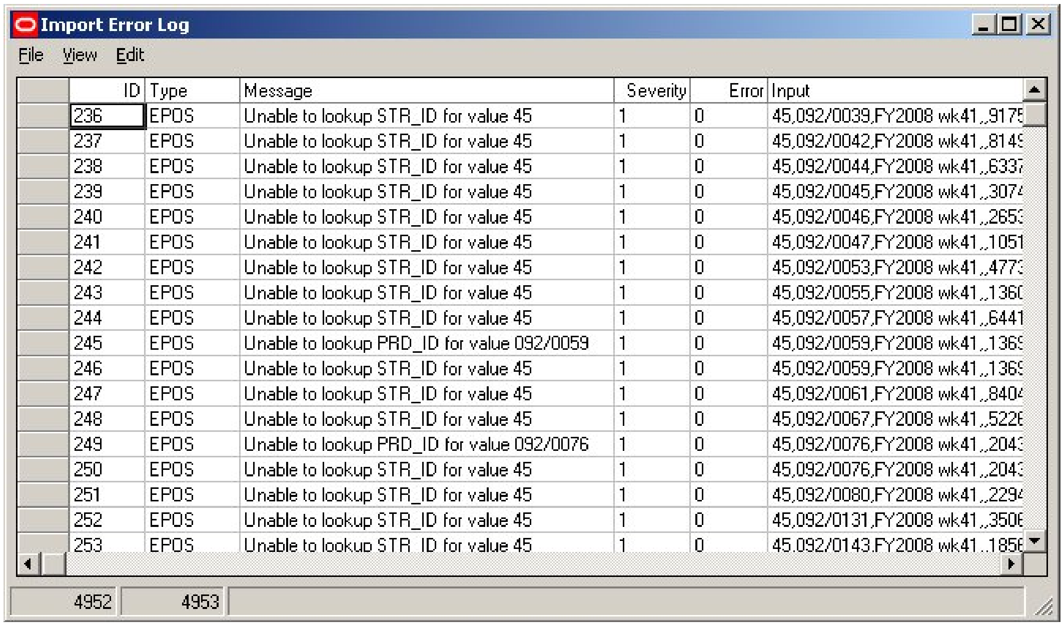


	ID	Type	Started	Finished	Added	Updated	Deleted	Skipped	Failed
1	1	EPOS - EPOS	14/03/2009 21:44:17	14/03/2009 21:45:31	920	0	0	0	4773
2	2	EPOS - EPOS	14/03/2009 21:46:30	14/03/2009 21:47:44	0	920	0	0	4773
3	3	EPOS - EPOS	14/03/2009 21:55:26	14/03/2009 21:57:22	309	920	0	0	4777
4	4	EPOS - EPOS	15/03/2009 14:56:43	15/03/2009 14:58:18	1634	143	0	0	4781
5	5	EPOS - EPOS	23/03/2009 19:19:18	23/03/2009 19:20:14	2681	31	0	0	0
6	6	Fixtures - Block Links	25/03/2009 08:30:52	25/03/2009 08:30:53	2	0	0	0	8
7									

7 records

The Import Error Log Dialogue Box

The **Import Error Log dialogue box** allows users to see details of any errors that might have resulted during recent data imports.



	ID	Type	Message	Severity	Error	Input
	236	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0039,FY2008 wk41,,9175
	237	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0042,FY2008 wk41,,8149
	238	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0044,FY2008 wk41,,6337
	239	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0045,FY2008 wk41,,3074
	240	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0046,FY2008 wk41,,2653
	241	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0047,FY2008 wk41,,1051
	242	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0053,FY2008 wk41,,4773
	243	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0055,FY2008 wk41,,1360
	244	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0057,FY2008 wk41,,6441
	245	EPOS	Unable to lookup PRD_ID for value 092/0059	1	0	45,092/0059,FY2008 wk41,,1369
	246	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0059,FY2008 wk41,,1369
	247	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0061,FY2008 wk41,,8404
	248	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0067,FY2008 wk41,,5226
	249	EPOS	Unable to lookup PRD_ID for value 092/0076	1	0	45,092/0076,FY2008 wk41,,2043
	250	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0076,FY2008 wk41,,2043
	251	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0080,FY2008 wk41,,2294
	252	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0131,FY2008 wk41,,3506
	253	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0143,FY2008 wk41,,1856

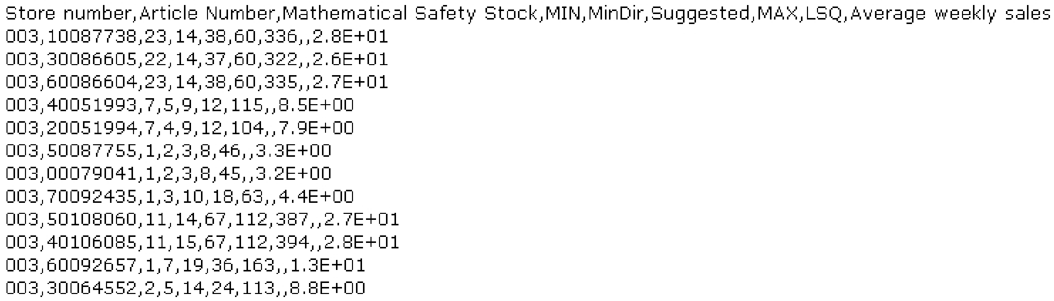
4952 4953

2

Basic Concepts

CSV File

A **CSV File (Comma Separated Value File)** is a way of moving data between one database and another. Each item of data is separated by a comma.



```
Store number,Article Number,Mathematical Safety Stock,MIN,MinDir,Suggested,MAX,LSQ,Average weekly sales
003,10087738,23,14,38,60,336,,2.8E+01
003,30086605,22,14,37,60,322,,2.6E+01
003,60086604,23,14,38,60,335,,2.7E+01
003,40051993,7,5,9,12,115,,8.5E+00
003,20051994,7,4,9,12,104,,7.9E+00
003,50087755,1,2,3,8,46,,3.3E+00
003,00079041,1,2,3,8,45,,3.2E+00
003,70092435,1,3,10,18,63,,4.4E+00
003,50108060,11,14,67,112,387,,2.7E+01
003,40106085,11,15,67,112,394,,2.8E+01
003,60092657,1,7,19,36,163,,1.3E+01
003,30064552,2,5,14,24,113,,8.8E+00
```

In the example above, the first line contains the descriptions of the items in the rest of the CSV table.

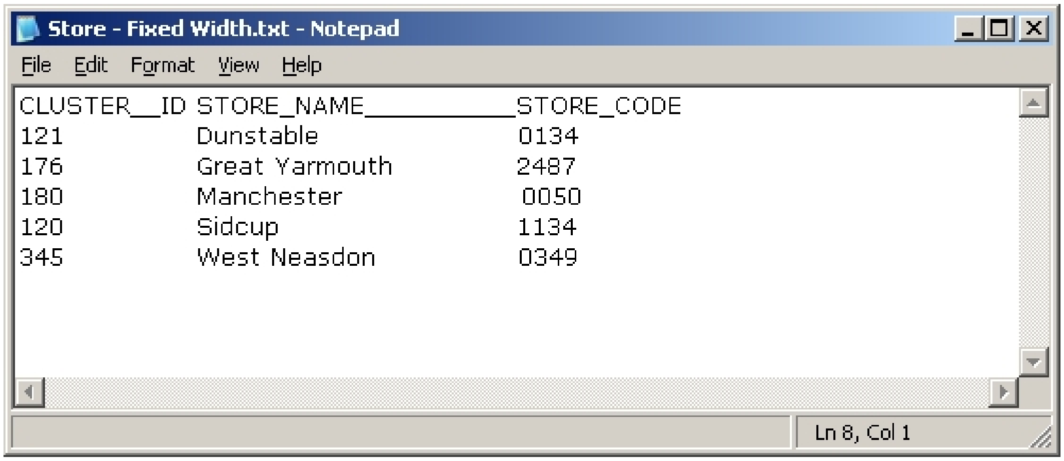
Each following line then contains one record, each item of data in the line being separated from another by a comma.

Note: The data must be precisely entered into the CSV file. For example the first position on every line must be contain the Store number, the second the article number, etc. If data gets out of sequence, then there will be errors transferring data from one database to another.

Important: It is important to save the CSV file in the format that is appropriate for the system it will be used on. If you do not save the CSV file in the format appropriate for the system you are working on, errors may result. For example, if you are working on a computer with the Windows operating System, and you want to use the file on that computer, it is important to save the file in the CSV (Windows) format. If however, you are using Windows and want to use the text file on a Macintosh computer, save the file in the CSV (Macintosh) format. If you are using a Macintosh computer and want to use the text file on a system running Windows or Windows NT, save the file in the CSV (Windows) format.

Fixed Width Value File

A **Fixed Width Value File** is a way of moving data between one database and another. Each item of data occupies a specific number of spaces along a line.



```
CLUSTER__ID STORE_NAME___________STORE_CODE
121         Dunstable             0134
176         Great Yarmouth        2487
180         Manchester            0050
120         Sidcup                1134
345         West Neasdon          0349
```

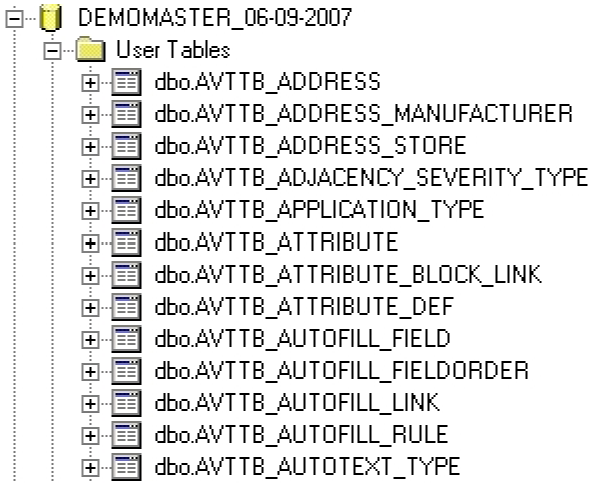
In the example above, the first line contains the descriptions of the items in the rest of the Fixed Width table.

Each following line then contains one record, each item of data in the line being a precise number of spaces. Where the actual data is not long enough to occupy the designated length, it must be padded out so that it does, for example with spaces.

Note: The data must be precisely entered into the Fixed Width file. For example the Cluster ID must always be 12 characters long. If the cluster ID is 11 or 13 characters long, there will be errors in transferring the entire sequence of data from one database to another.

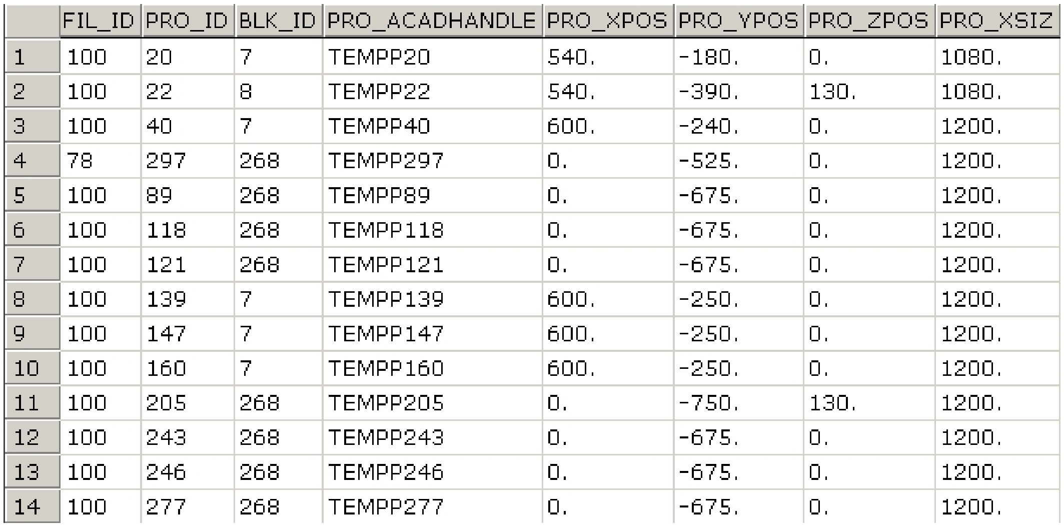
Macro Space Management Database

The **Macro Space Management database** consists of a large number of tables.



```
DEMOMASTER_06-09-2007
  User Tables
    dbo.AVTTB_ADDRESS
    dbo.AVTTB_ADDRESS_MANUFACTURER
    dbo.AVTTB_ADDRESS_STORE
    dbo.AVTTB_ADJACENCY_SEVERITY_TYPE
    dbo.AVTTB_APPLICATION_TYPE
    dbo.AVTTB_ATTRIBUTE
    dbo.AVTTB_ATTRIBUTE_BLOCK_LINK
    dbo.AVTTB_ATTRIBUTE_DEF
    dbo.AVTTB_AUTOFILL_FIELD
    dbo.AVTTB_AUTOFILL_FIELDORDER
    dbo.AVTTB_AUTOFILL_LINK
    dbo.AVTTB_AUTOFILL_RULE
    dbo.AVTTB_AUTOTEXT_TYPE
```

Each of these tables contain many items of data



	FIL_ID	PRO_ID	BLK_ID	PRO_ACADHANDLE	PRO_XPOS	PRO_YPOS	PRO_ZPOS	PRO_XSIZ
1	100	20	7	TEMPP20	540.	-180.	0.	1080.
2	100	22	8	TEMPP22	540.	-390.	130.	1080.
3	100	40	7	TEMPP40	600.	-240.	0.	1200.
4	78	297	268	TEMPP297	0.	-525.	0.	1200.
5	100	89	268	TEMPP89	0.	-675.	0.	1200.
6	100	118	268	TEMPP118	0.	-675.	0.	1200.
7	100	121	268	TEMPP121	0.	-675.	0.	1200.
8	100	139	7	TEMPP139	600.	-250.	0.	1200.
9	100	147	7	TEMPP147	600.	-250.	0.	1200.
10	100	160	7	TEMPP160	600.	-250.	0.	1200.
11	100	205	268	TEMPP205	0.	-750.	130.	1200.
12	100	243	268	TEMPP243	0.	-675.	0.	1200.
13	100	246	268	TEMPP246	0.	-675.	0.	1200.
14	100	277	268	TEMPP277	0.	-675.	0.	1200.

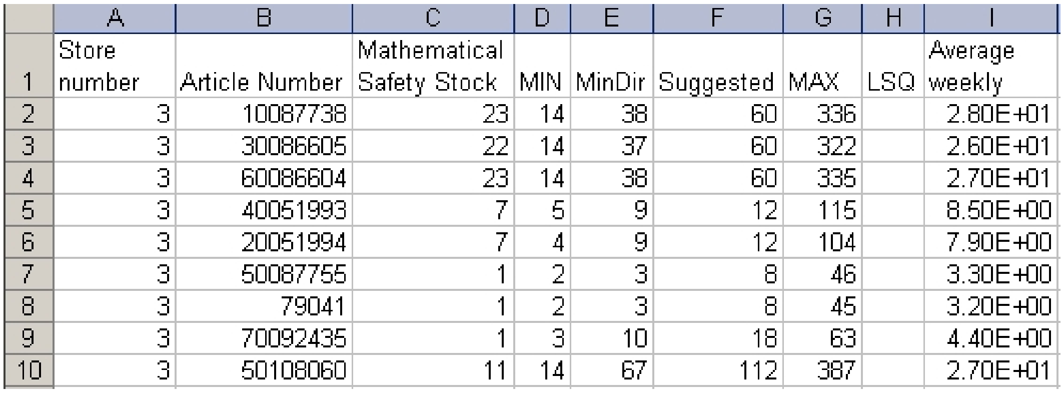
Each line of this table represents one record of an object within Macro Space Management.

Each of the columns within the line contains information on one aspect of that record.

In the first line of this example we can see this record has a File Identification Number (FIL_ID) of 100 and a Product Identification Number (PRO_ID) of 20. Each of the other columns contains yet more information.

Spreadsheet

A **spreadsheet** provides a structured way to move data between one database and another. Each item of data is separated put into a separate cell.



	A	B	C	D	E	F	G	H	I
1	Store number	Article Number	Mathematical Safety Stock	MIN	MinDir	Suggested	MAX	LSQ	Average weekly
2	3	10087738	23	14	38	60	336		2.80E+01
3	3	30086605	22	14	37	60	322		2.60E+01
4	3	60086604	23	14	38	60	335		2.70E+01
5	3	40051993	7	5	9	12	115		8.50E+00
6	3	20051994	7	4	9	12	104		7.90E+00
7	3	50087755	1	2	3	8	46		3.30E+00
8	3	79041	1	2	3	8	45		3.20E+00
9	3	70092435	1	3	10	18	63		4.40E+00
10	3	50108060	11	14	67	112	387		2.70E+01

In the example above, the first line contains the descriptions of the items in the rest of the spreadsheet.

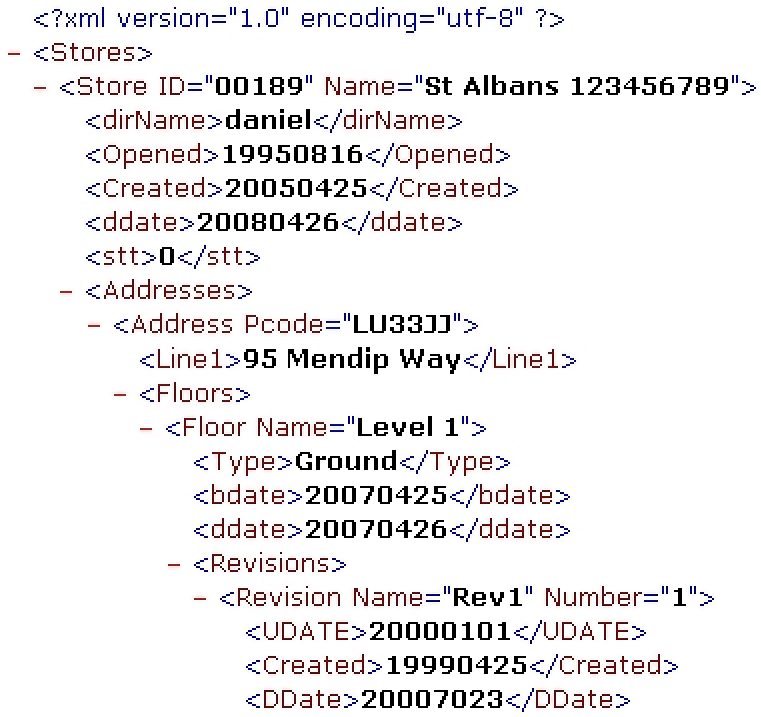
Each following line then contains one record, each item of data being placed in successive cells.

> **Note**: The data must be precisely entered into the spreadsheet. For example the first cell on every line must be contain the Store number, the second the article number, etc.

If data gets out of sequence, then there will be errors transferring data from one database to another.

XML File

A **XML File (Extended Mark-Up Language file)** provides a structured way to move data between one database and another. Each item of data is separated by tags.



```
  <?xml version="1.0" encoding="utf-8" ?>
- <Stores>
  - <Store ID="00189" Name="St Albans 123456789">
      <dirName>daniel</dirName>
      <Opened>19950816</Opened>
      <Created>20050425</Created>
      <ddate>20080426</ddate>
      <stt>0</stt>
    - <Addresses>
      - <Address Pcode="LU33JJ">
          <Line1>95 Mendip Way</Line1>
        - <Floors>
          - <Floor Name="Level 1">
              <Type>Ground</Type>
              <bdate>20070425</bdate>
              <ddate>20070426</ddate>
            - <Revisions>
              - <Revision Name="Rev1" Number="1">
                  <UDATE>20000101</UDATE>
                  <Created>19990425</Created>
                  <DDate>20007023</DDate>
```

Note: The data must be precisely entered into the XML file. If data gets out of sequence, then there will be errors transferring data from one database to another.

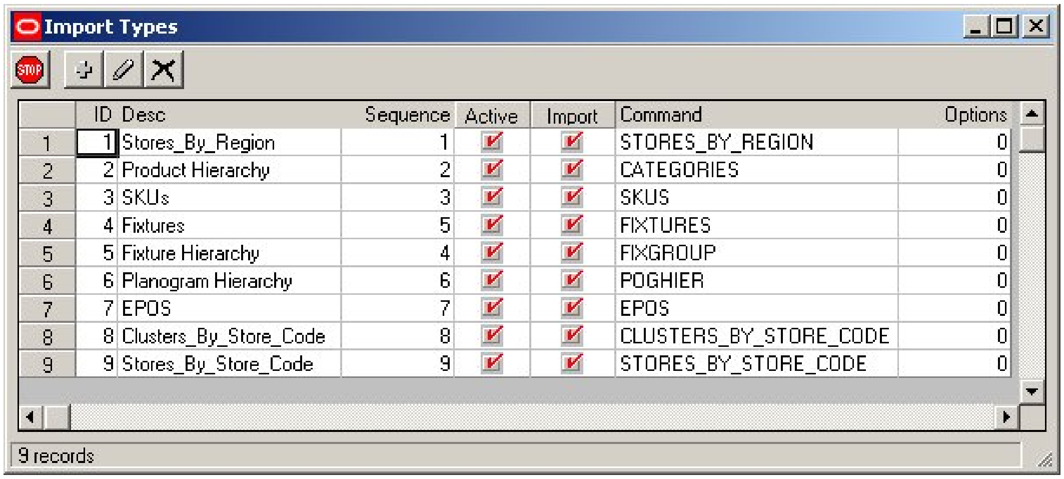
3

Data Importer Dialog Boxes

Import Type dialog box

The Import Type GUI

The **Import Type GUI** lists all Import Types.

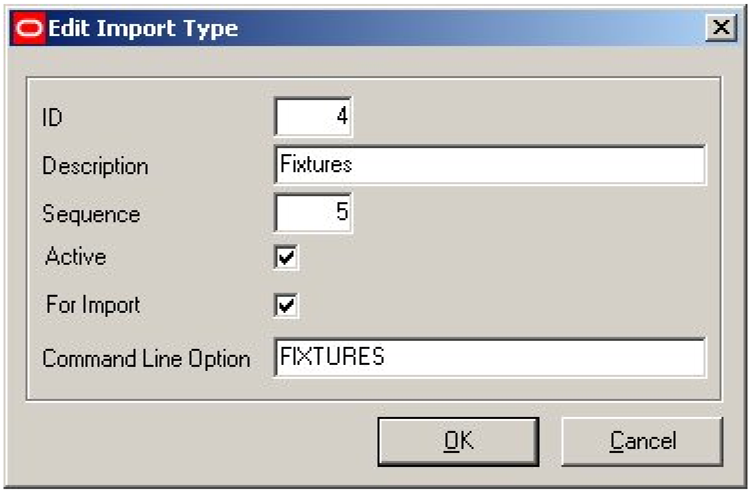


	ID	Desc	Sequence	Active	Import	Command	Options
1	1	Stores_By_Region	1	☑	☑	STORES_BY_REGION	0
2	2	Product Hierarchy	2	☑	☑	CATEGORIES	0
3	3	SKUs	3	☑	☑	SKUS	0
4	4	Fixtures	5	☑	☑	FIXTURES	0
5	5	Fixture Hierarchy	4	☑	☑	FIXGROUP	0
6	6	Planogram Hierarchy	6	☑	☑	POGHIER	0
7	7	EPOS	7	☑	☑	EPOS	0
8	8	Clusters_By_Store_Code	8	☑	☑	CLUSTERS_BY_STORE_CODE	0
9	9	Stores_By_Store_Code	9	☑	☑	STORES_BY_STORE_CODE	0

9 records

Import Types can be Added, Edit or Deleted using the appropriate Icons.

Some information for Import Types can be configured via the Add or Edit Options.



ID: 4

Description: Fixtures

Sequence: 5

Active ☑

For Import ☑

Command Line Option: FIXTURES

ID is assigned by the software and cannot be user configured.

Description is the name of the Import type.

Sequence determines the order the data import will be carried out.

The Active and For Import check boxes determine whether a Data Import Type is active and will be imported.

The Command Line Option allows users to specify the command line switch that will be used to invoke the Import Type if data Importer is run using the Command Line.

Deleting Import Types

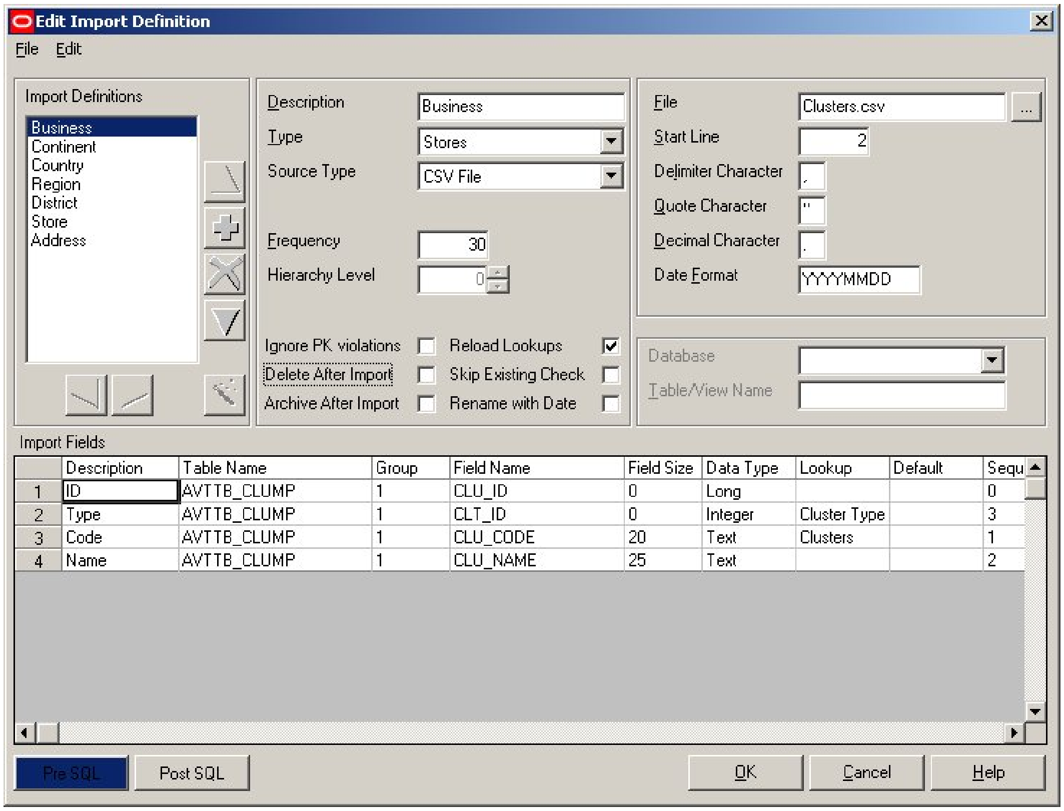
Import Types cannot be deleted until all child import definitions have been deleted.

If child import definitions exist, the Import Type will apparently deleted from the list of Import Types but will reappear the next time Data Importer is opened.

Import Definitions dialog box

Import Definition Dialogue Box

The **Import Definition dialogue box** allows users to configure Import Definitions.



Import Definitions

- Business
- Continent
- Country
- Region
- District
- Store
- Address

Description: Business
Type: Stores
Source Type: CSV File
Frequency: 30
Hierarchy Level: 0

File: Clusters.csv
Start Line: 2
Delimiter Character: ,
Quote Character: "
Decimal Character: .
Date Format: YYYYMMDD

Database:
Table/View Name:

Ignore PK violations □ Reload Lookups ☑
Delete After Import □ Skip Existing Check □
Archive After Import □ Rename with Date □

Import Fields

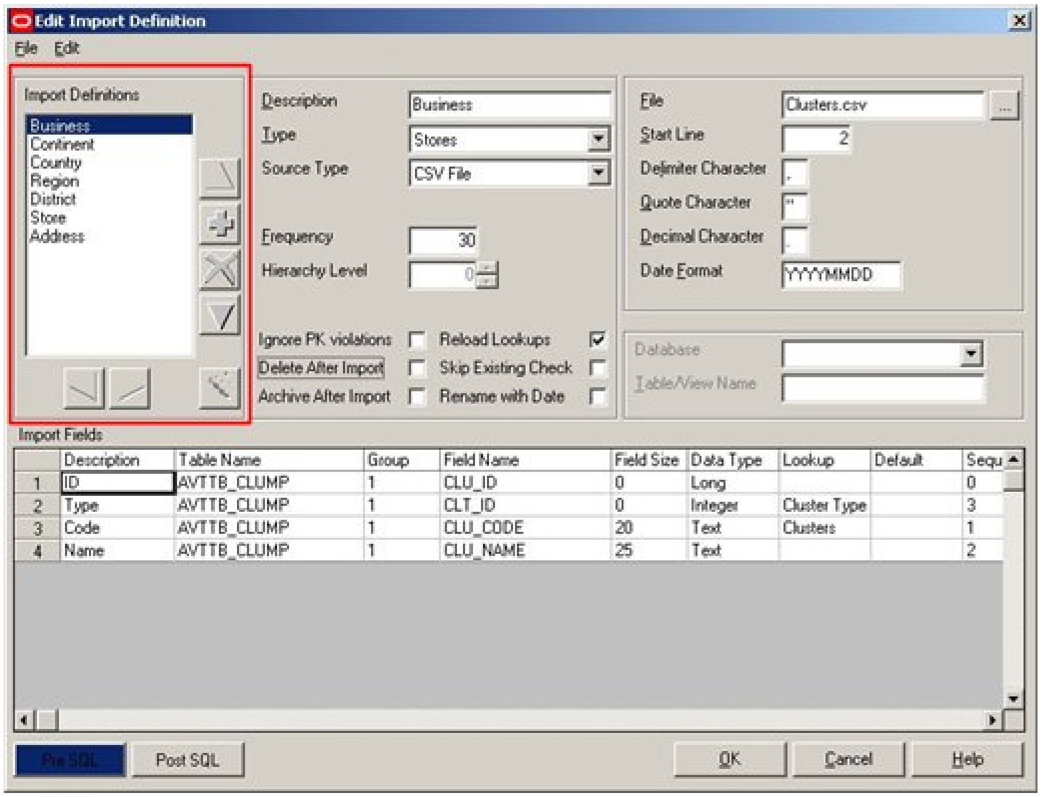
	Description	Table Name	Group	Field Name	Field Size	Data Type	Lookup	Default	Sequ
1	ID	AVTTB_CLUMP	1	CLU_ID	0	Long			0
2	Type	AVTTB_CLUMP	1	CLT_ID	0	Integer	Cluster Type		3
3	Code	AVTTB_CLUMP	1	CLU_CODE	20	Text	Clusters		1
4	Name	AVTTB_CLUMP	1	CLU_NAME	25	Text			2

This allows users to configure how data is extracted from the source file (in this example Fixtures.csv) and inserted into the database.

Import Definitions Frame

The **Import Definitions Frame** contains the list of Import Definitions.

(Each Import Definition is configured in the Import Fields Frame).



Import Definitions

- Business
- Continent
- Country
- Region
- District
- Store
- Address

Description: Business
Type: Stores
Source Type: CSV File
Frequency: 30
Hierarchy Level: 0

Ignore PK violations □ Reload Lookups ☑
Delete After Import □ Skip Existing Check □
Archive After Import □ Rename with Date □

File: Clusters.csv
Start Line: 2
Delimiter Character: ,
Quote Character: "
Decimal Character: .
Date Format: YYYYMMDD

Database:
Table/View Name:

Import Fields

	Description	Table Name	Group	Field Name	Field Size	Data Type	Lookup	Default	Sequ
1	ID	AVTTB_CLUMP	1	CLU_ID	0	Long			0
2	Type	AVTTB_CLUMP	1	CLT_ID	0	Integer	Cluster Type		3
3	Code	AVTTB_CLUMP	1	CLU_CODE	20	Text	Clusters		1
4	Name	AVTTB_CLUMP	1	CLU_NAME	25	Text			2

Import Definitions can be Added or Deleted using the appropriate icons.

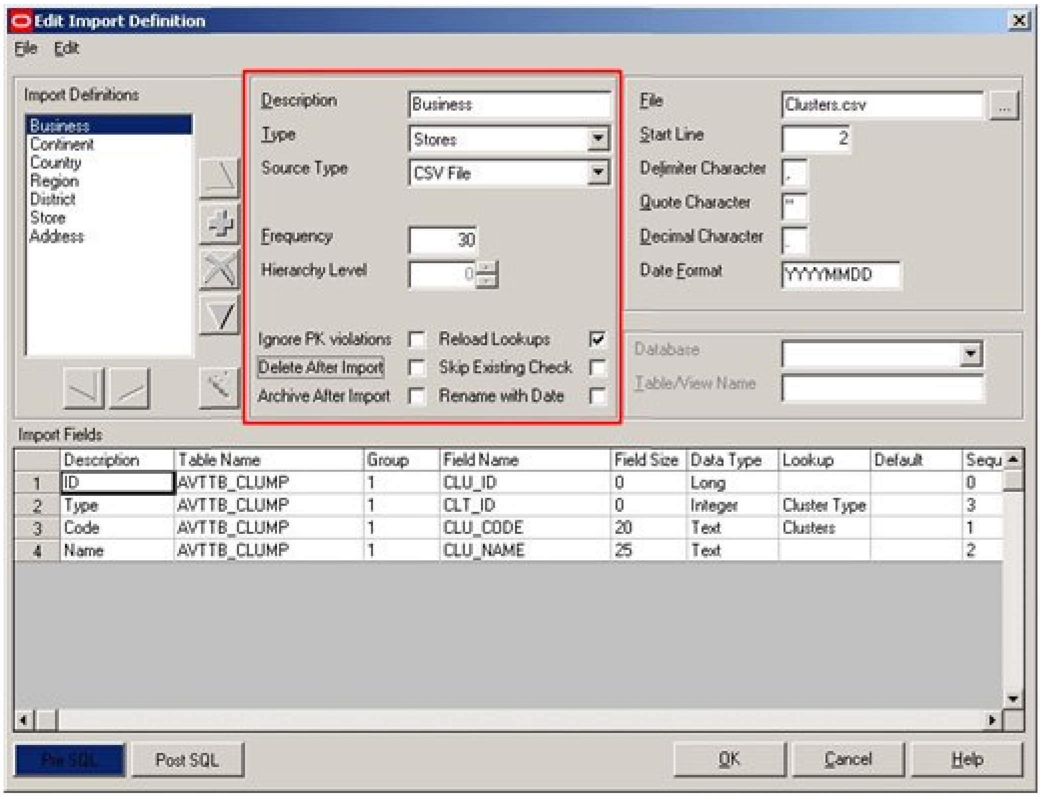
The Import Definitions will be executed in the order that they are in the list. The up and down arrows allow users to move an individual import definition up or down in the list.

Import Definitions can also be indented using the left and right arrows - this option is for XML files only and will be greyed out unless and XML file is specified in the Source type in the Description Frame.

The Wizard Option is for XML files only and is not currently functional.

Descriptions Frame

The **Descriptions Frame** contains a series of options defining how the import will work.



Edit Import Definition

Import Definitions: Business, Continent, Country, Region, District, Store, Address

Description: Business

Type: Stores

Source Type: CSV File

Frequency: 30

Hierarchy Level: 0

Ignore PK violations □ Reload Lookups ☑

Delete After Import □ Skip Existing Check □

Archive After Import □ Rename with Date □

File: Clusters.csv

Start Line: 2

Delimiter Character: ,

Quote Character: "

Decimal Character: .

Date Format: YYYYMMDD

Database:

Table/View Name:

Import Fields

	Description	Table Name	Group	Field Name	Field Size	Data Type	Lookup	Default	Sequ
1	ID	AVTTB_CLUMP	1	CLU_ID	0	Long			0
2	Type	AVTTB_CLUMP	1	CLT_ID	0	Integer	Cluster Type		3
3	Code	AVTTB_CLUMP	1	CLU_CODE	20	Text	Clusters		1
4	Name	AVTTB_CLUMP	1	CLU_NAME	25	Text			2

Description

This can be any helpful description of the Import definition currently being configured. The description will also be displayed in the list of Import Definitions to the left of the Definitions frame.

> **Note**: The description for XML files must match the property tag, e.g. <Store> --> "Store"; this is case sensitive and must not include any spaces.

Type

This is the Type of Import Definition - it is selected from a drop down list containing the Types set up using the Import Types GUI.

Source Type

This is the type of file that will be used to import the data from. It may be one of three types:

- CSV (Comma Separated Value)
- Fixed Width
- XML (eXtended Mark-up Language)

It is selected from a drop down list.

Transaction

This functionality is not yet enabled.

Frequency

This functionality is not yet enabled.

Hierarchy Level

This is used for XML files. (The label and the text box will be disabled for other source types).

The spin button will increase or decrease the level. This can be used as an alternative to the left and right arrow buttons below the definition list). The same rules will apply regarding the values, i.e. no value can be less than 1, and no value can be greater than the preceding definition +1.

A value of 0 will show when the source type is not XML.

If the source type is XML and the level >1, then the file frame will be disabled.

Application

This functionality is not yet enabled.

Ignore PK Violations

Important: Please use this function with caution - it could result in significant errors during data import if incorrectly used.

Checking this option means Data Importer will not raise an error if the Primary Key constraint is violated.

This would typically be used on link tables (e.g. Store Cluster Link) to avoid checking whether a composite key already exists; Date Importer will add the record and ignore the error raised by the database.

Delete After Import

Checking this option means Data Importer will delete the file after it had been imported.

Archive After Import

Checking this option means Data Importer will move the import file to the "*Archived Files*" directory, (as defined in the Configuration Module).

Reload Lookups

This will force Data Importer to clear the relevant look-up cache before items from the file are read.

Skip Existing Check

Important: Please use this function with caution - it could result in significant errors during data import if incorrectly used.

This option is for speed enhancement.

It is intended for when the user knows that the table is empty, so that Data Importer can skip the process of checking whether each record exists in order to determine whether to add or update an existing record.

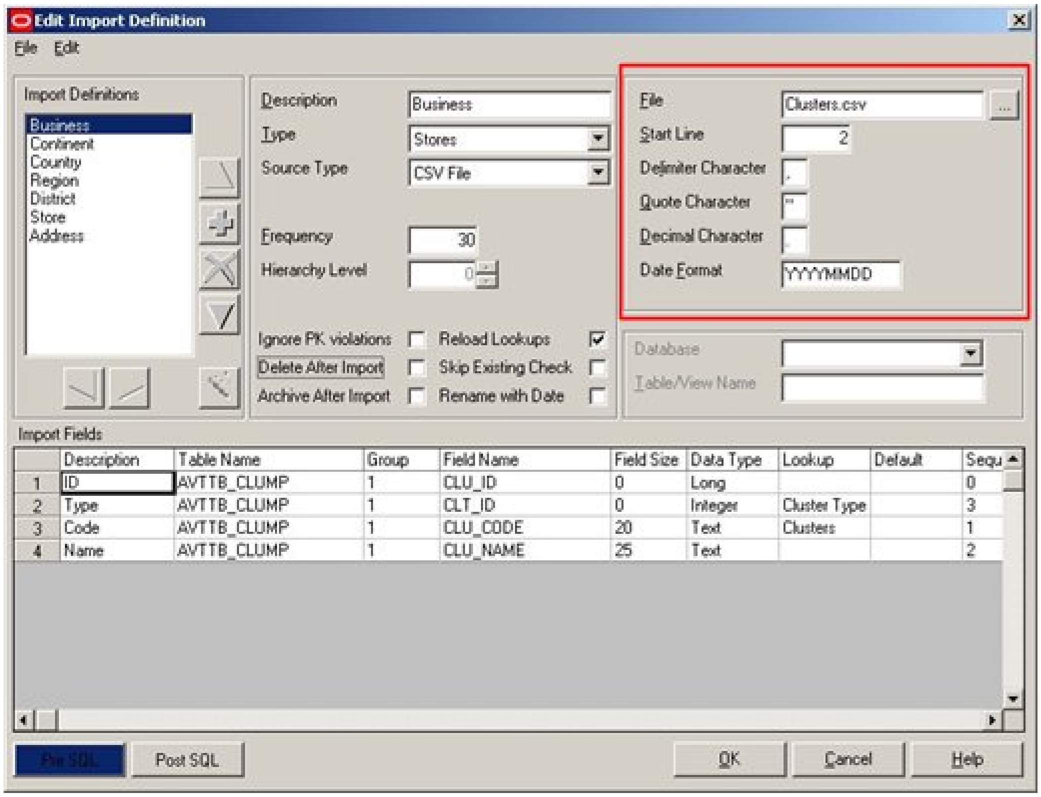
Rename with Date

If this option is checked, the file being imported will be renamed by appending the date once the import process for that file is complete.

For example Products would be renamed Products 25/12/2005. This is often done in conjunction with the Archive After Import option.

File Frame

The **File Frame** contains information on the file to be used, and how to interpret the information in it.



Import Definitions
Business
Continent
Country
Region
District
Store
Address

Description: Business
Type: Stores
Source Type: CSV File
Frequency: 30
Hierarchy Level: 0

File: Clusters.csv
Start Line: 2
Delimiter Character: ,
Quote Character: "
Decimal Character: .
Date Format: YYYYMMDD

Ignore PK violations ☐
Reload Lookups ☑
Delete After Import ☐
Skip Existing Check ☐
Archive After Import ☐
Rename with Date ☐

Database:
Table/View Name:

Import Fields

	Description	Table Name	Group	Field Name	Field Size	Data Type	Lookup	Default	Sequ
1	ID	AVTTB_CLUMP	1	CLU_ID	0	Long			0
2	Type	AVTTB_CLUMP	1	CLT_ID	0	Integer	Cluster Type		3
3	Code	AVTTB_CLUMP	1	CLU_CODE	20	Text	Clusters		1
4	Name	AVTTB_CLUMP	1	CLU_NAME	25	Text			2

File is the name of the file that will be used for that import definition.

Start Line is the line which Data Import will start reading from in the file.

For example if the first line of a .csv file contains the name of a field and the second the sequence, we would want to start reading data from line 3.

Delimiter Character is the character that separates items of data in a .csv file (for example a comma). This option will not be available for XML and Fixed Width files.

Quote Character specified which character in the import file is used to indicate text. The quote characters will be stripped off when the data is imported into the database.

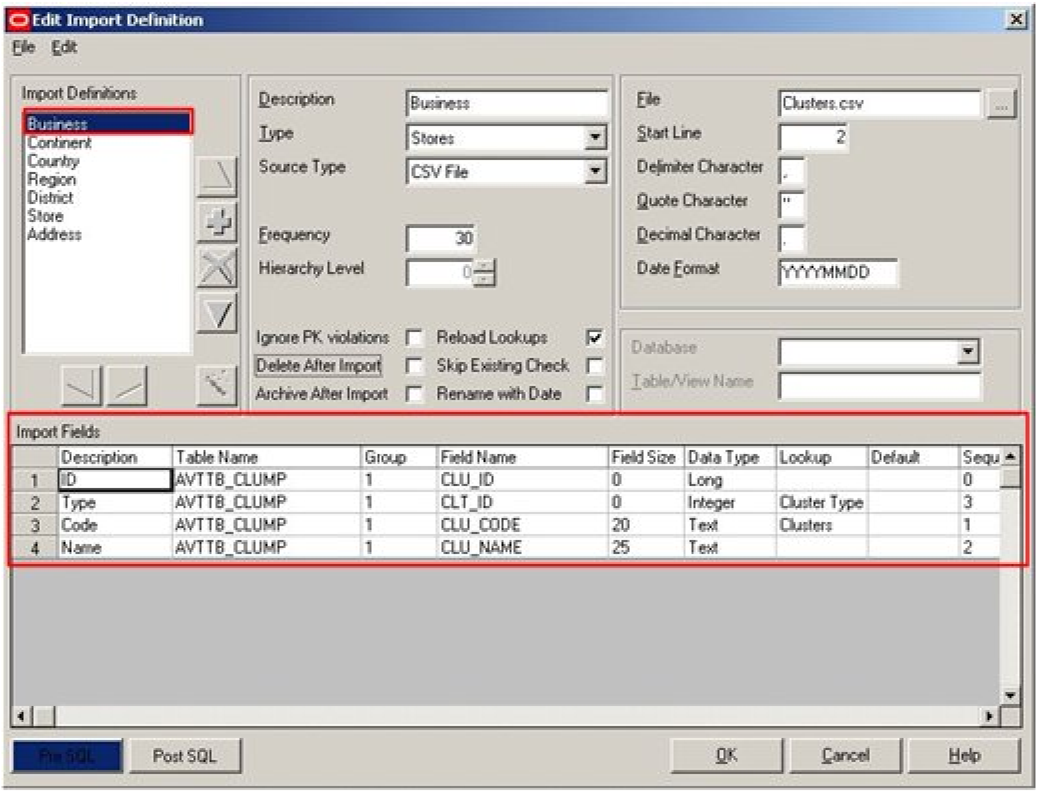
Decimal Character indicates which character is used to denote the decimal point in real numeric values.

For example, the UK might use a period (3.14259) while France might use a comma (3,14159).

Date Format denotes how to interpret dates. These are always assumed to be numeric, but can be separated by delimiters, for example DD/MM/YY or YYYYMMDD.

Import Fields Frame

The **Import Fields Frame** specifies all fields for the Import Definition highlighted in the Import Definitions Frame.



Import Definitions

- Business
- Continent
- Country
- Region
- District
- Store
- Address

Description: Business

Type: Stores

Source Type: CSV File

Frequency: 30

Hierarchy Level: 0

Ignore PK violations □ Reload Lookups ☑

Delete After Import □ Skip Existing Check □

Archive After Import □ Rename with Date □

File: Clusters.csv

Start Line: 2

Delimiter Character: ,

Quote Character: "

Decimal Character: .

Date Format: YYYYMMDD

Database:

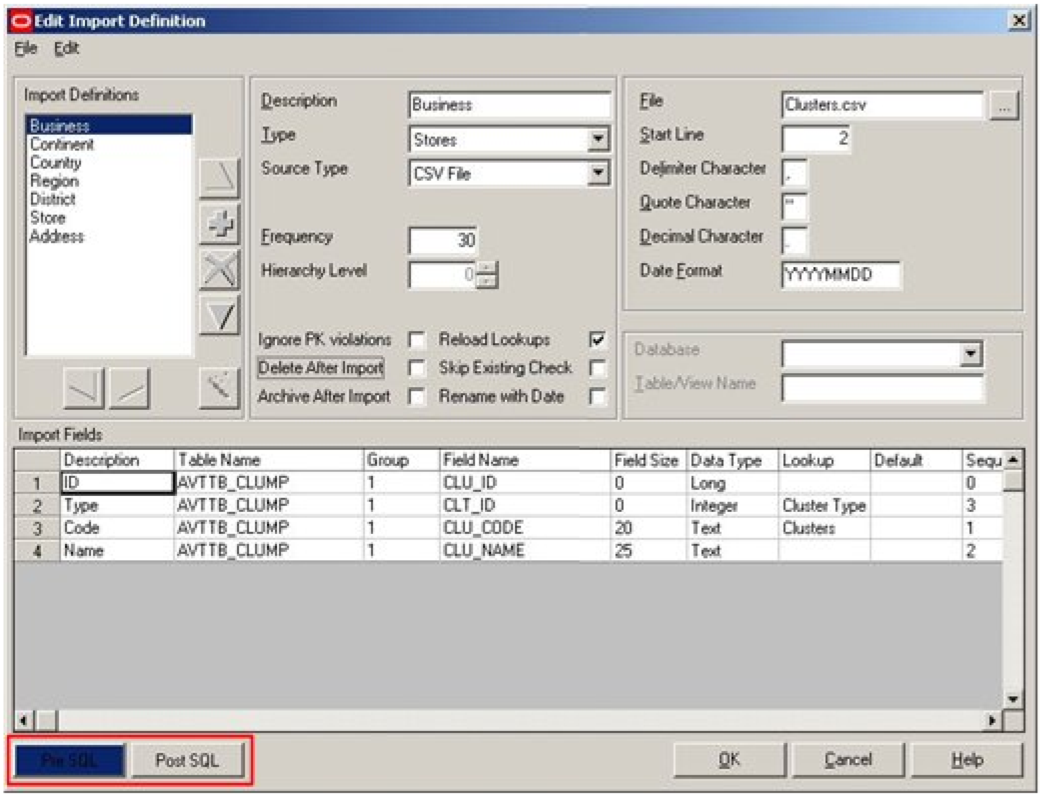
Table/View Name:

Import Fields

	Description	Table Name	Group	Field Name	Field Size	Data Type	Lookup	Default	Sequ
1	ID	AVTTB_CLUMP	1	CLU_ID	0	Long			0
2	Type	AVTTB_CLUMP	1	CLT_ID	0	Integer	Cluster Type		3
3	Code	AVTTB_CLUMP	1	CLU_CODE	20	Text	Clusters		1
4	Name	AVTTB_CLUMP	1	CLU_NAME	25	Text			2

Pre and Post SQL Options

The **Pre and Post SQL** options allow the user to add SQL statements to an import definition.



Import Definitions: Business, Continent, Country, Region, District, Store, Address

Description: Business

Type: Stores

Source Type: CSV File

Frequency: 30

Hierarchy Level: 0

File: Clusters.csv

Start Line: 2

Delimiter Character: ,

Quote Character: "

Decimal Character: .

Date Format: YYYYMMDD

Database:

Table/View Name:

Ignore PK violations □ Reload Lookups ☑

Delete After Import □ Skip Existing Check □

Archive After Import □ Rename with Date □

Import Fields

	Description	Table Name	Group	Field Name	Field Size	Data Type	Lookup	Default	Sequ
1	ID	AVTTB_CLUMP	1	CLU_ID	0	Long			0
2	Type	AVTTB_CLUMP	1	CLT_ID	0	Integer	Cluster Type		3
3	Code	AVTTB_CLUMP	1	CLU_CODE	20	Text	Clusters		1
4	Name	AVTTB_CLUMP	1	CLU_NAME	25	Text			2

About Pre and Post SQL

Pre and Post SQL allow the Data Import module to execute changes to the database tables using SQL statements. Accordingly, Pre and Post SQL functionality is disabled by default. Please refer to the Functional Security section in the Administration Module User Guide for information on how to give access to this functionality.

Using Pre and Post SQL

If either the Pre SQL or Post SQL options contains SQL to be executed, it will be shown colored blue.

Pre SQL will be executed before the import is executed, and Post SQL will be executed after the import is executed.

An example of Pre SQL might be:

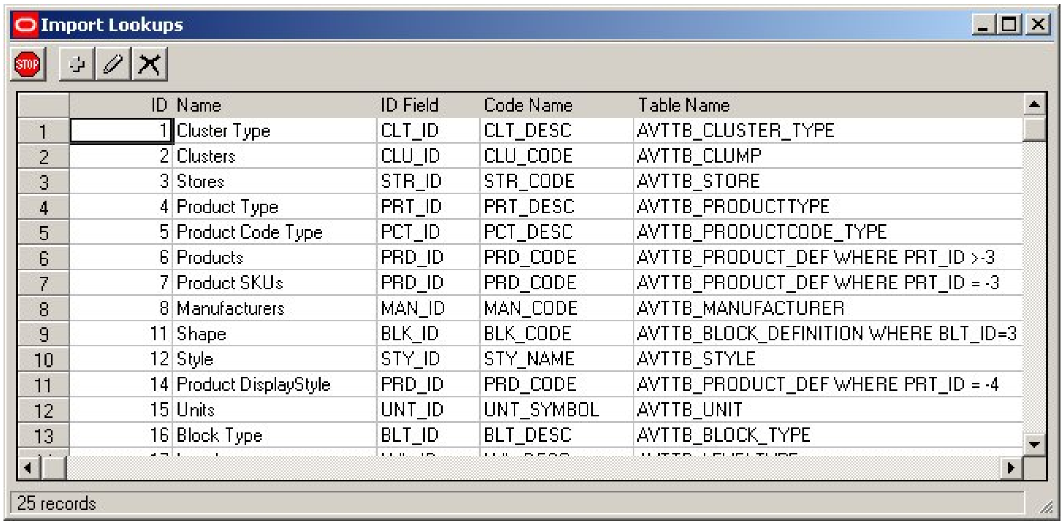
UPDATE AVTTB_NEXTID SET NID_NEXTID = (SELECT COALESCE(MAX(CLU_ID),0) + 1 FROM AVTTB_CLUMP) WHERE NID_TABLENAME='AVTTB_CLUMP'

This ensures that the AVTTB_NEXTID table is updated with the correct value to use when adding a new record to the database.

Lookups dialog box

The Lookup Dialogue Box

The **Look-up dialogue box** lists all defined lookups.

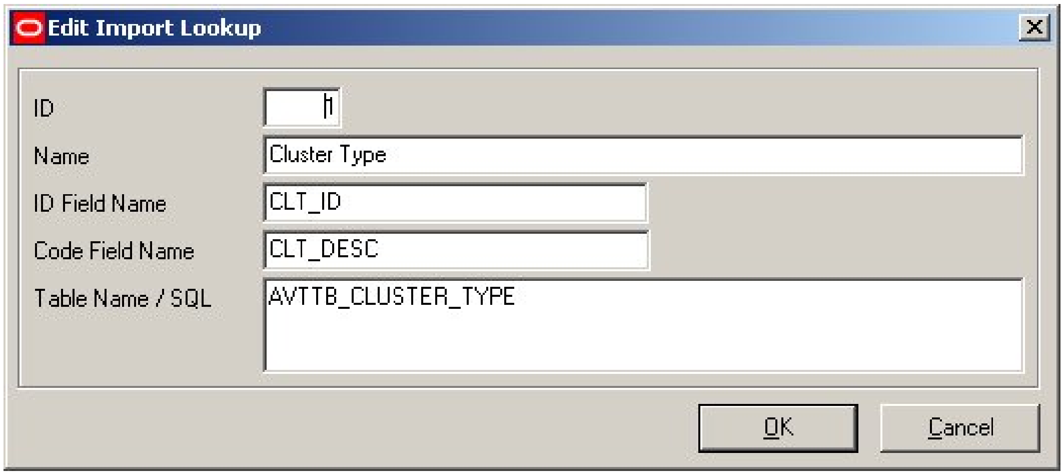


	ID	Name	ID Field	Code Name	Table Name
1	1	Cluster Type	CLT_ID	CLT_DESC	AVTTB_CLUSTER_TYPE
2	2	Clusters	CLU_ID	CLU_CODE	AVTTB_CLUMP
3	3	Stores	STR_ID	STR_CODE	AVTTB_STORE
4	4	Product Type	PRT_ID	PRT_DESC	AVTTB_PRODUCTTYPE
5	5	Product Code Type	PCT_ID	PCT_DESC	AVTTB_PRODUCTCODE_TYPE
6	6	Products	PRD_ID	PRD_CODE	AVTTB_PRODUCT_DEF WHERE PRT_ID >-3
7	7	Product SKUs	PRD_ID	PRD_CODE	AVTTB_PRODUCT_DEF WHERE PRT_ID = -3
8	8	Manufacturers	MAN_ID	MAN_CODE	AVTTB_MANUFACTURER
9	11	Shape	BLK_ID	BLK_CODE	AVTTB_BLOCK_DEFINITION WHERE BLT_ID=3
10	12	Style	STY_ID	STY_NAME	AVTTB_STYLE
11	14	Product DisplayStyle	PRD_ID	PRD_CODE	AVTTB_PRODUCT_DEF WHERE PRT_ID = -4
12	15	Units	UNT_ID	UNT_SYMBOL	AVTTB_UNIT
13	16	Block Type	BLT_ID	BLT_DESC	AVTTB_BLOCK_TYPE

25 records

The Look-ups dialog box allows changes to the database tables using SQL statements. Accordingly, access to the dialog box is disabled by default. Please refer to the Functional Security section in the Administration Module User Guide for information on how to give access to this functionality.

If the user has the appropriate permissions, Lookups can be Added, Edit or Deleted using the appropriate Icons.



ID: 1

Name: Cluster Type

ID Field Name: CLT_ID

Code Field Name: CLT_DESC

Table Name / SQL: AVTTB_CLUSTER_TYPE

ID is assigned by the software and cannot be user configured.

Name is the Name Assigned to the Lookup.

ID Field Name is the name of the field in the Macro Space Management database field table that will return the lookup value.

Code Field Name is the name of the field that will be used for the lookup.

Table Name / SQL is the name of the table the lookup will be executed in.

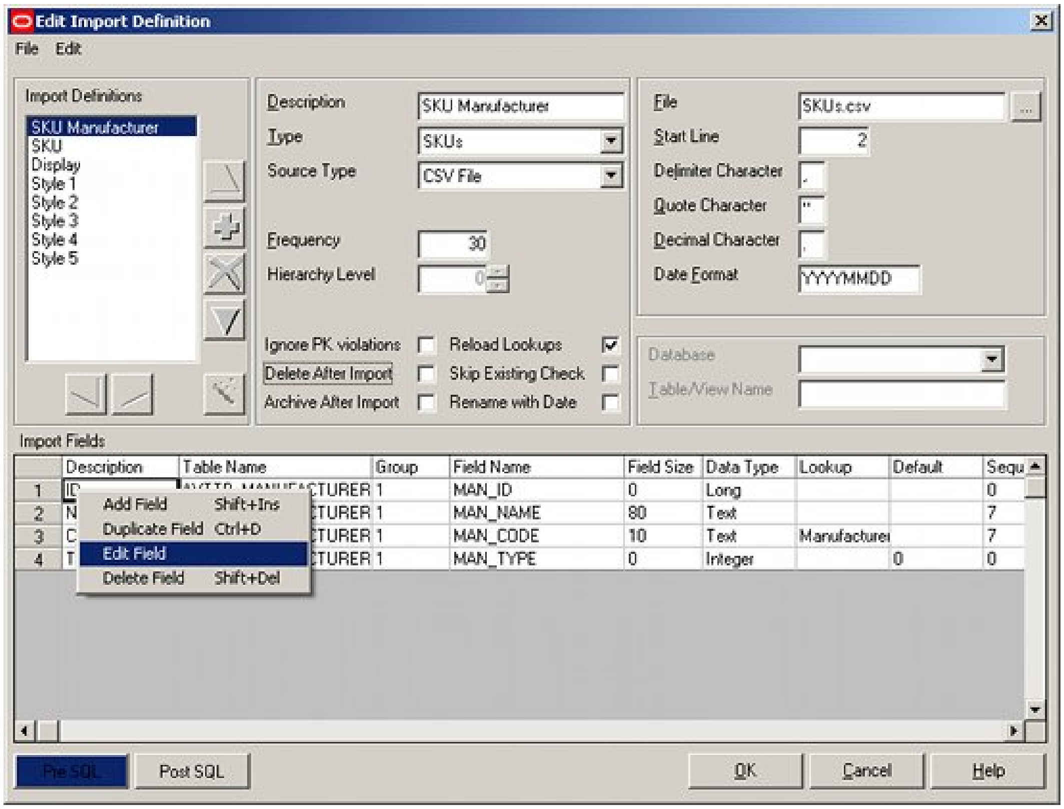
The SQL is optional, but may be used to limit the options within the table that can be used to return a lookup value.

For example, a SQL statement might restrict the cluster types that can be imported.

Import Field dialog box

Import Field Dialogue Box

The **Import Field dialogue box** is accessible from the Import Definitions dialogue box.



Import Definitions
SKU Manufacturer
SKU
Display
Style 1
Style 2
Style 3
Style 4
Style 5

Description: SKU Manufacturer
Type: SKUs
Source Type: CSV File
Frequency: 30
Hierarchy Level: 0

File: SKUs.csv
Start Line: 2
Delimiter Character: ,
Quote Character: "
Decimal Character: .
Date Format: YYYYMMDD

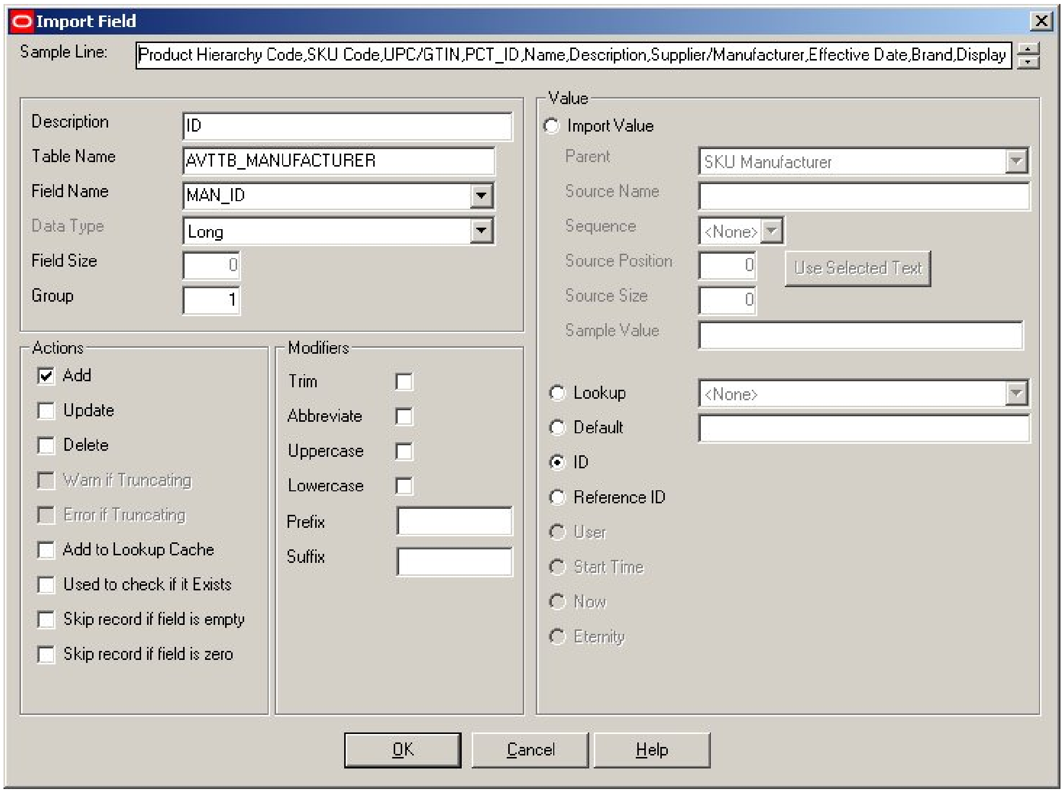
Ignore PK violations □
Delete After Import □
Archive After Import □
Reload Lookups ☑
Skip Existing Check □
Rename with Date □

Database
Table/View Name

Import Fields

	Description	Table Name	Group	Field Name	Field Size	Data Type	Lookup	Default	Sequ
1	ID	AVTTP_MANUFACTURER	1	MAN_ID	0	Long			0
2	N	CTURER	1	MAN_NAME	80	Text			7
3	C	CTURER	1	MAN_CODE	10	Text	Manufacturer		7
4	T	CTURER	1	MAN_TYPE	0	Integer		0	0

Right clicking in the Import Fields frame will bring up a pop-up menu. This can be used to bring up the Import Field dialogue box.



Import Field

Sample Line: Product Hierarchy Code,SKU Code,UPC/GTIN,PCT_ID,Name,Description,Supplier/Manufacturer,Effective Date,Brand,Display

Description: ID
Table Name: AVTTB_MANUFACTURER
Field Name: MAN_ID
Data Type: Long
Field Size: 0
Group: 1

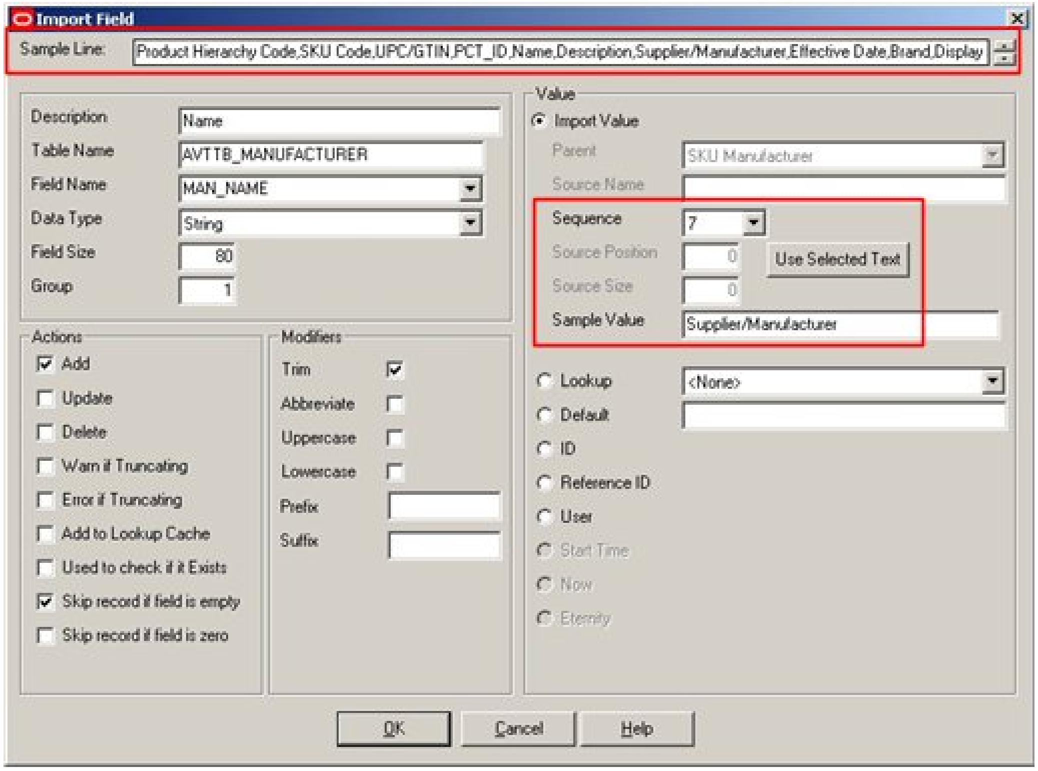
Actions
☑ Add
□ Update
□ Delete
□ Warn if Truncating
□ Error if Truncating
□ Add to Lookup Cache
□ Used to check if it Exists
□ Skip record if field is empty
□ Skip record if field is zero

Modifiers
Trim □
Abbreviate □
Uppercase □
Lowercase □
Prefix
Suffix

Value
○ Import Value
Parent: SKU Manufacturer
Source Name
Sequence: <None>
Source Position: 0
Source Size: 0
Sample Value
○ Lookup: <None>
○ Default
● ID
○ Reference ID
○ User
○ Start Time
○ Now
○ Eternity

Sample Line and Sample Value

The **Sample Line** and **Sample Value** show a sample of the data that is to be imported and the specific value referenced by the definition being configured.



Import Field

Sample Line: Product Hierarchy Code,SKU Code,UPC/GTIN,PCT_ID,Name,Description,Supplier/Manufacturer,Effective Date,Brand,Display

Description: Name
Table Name: AVTTB_MANUFACTURER
Field Name: MAN_NAME
Data Type: String
Field Size: 80
Group: 1

Actions
☑ Add
□ Update
□ Delete
□ Warn if Truncating
□ Error if Truncating
□ Add to Lookup Cache
□ Used to check if it Exists
☑ Skip record if field is empty
□ Skip record if field is zero

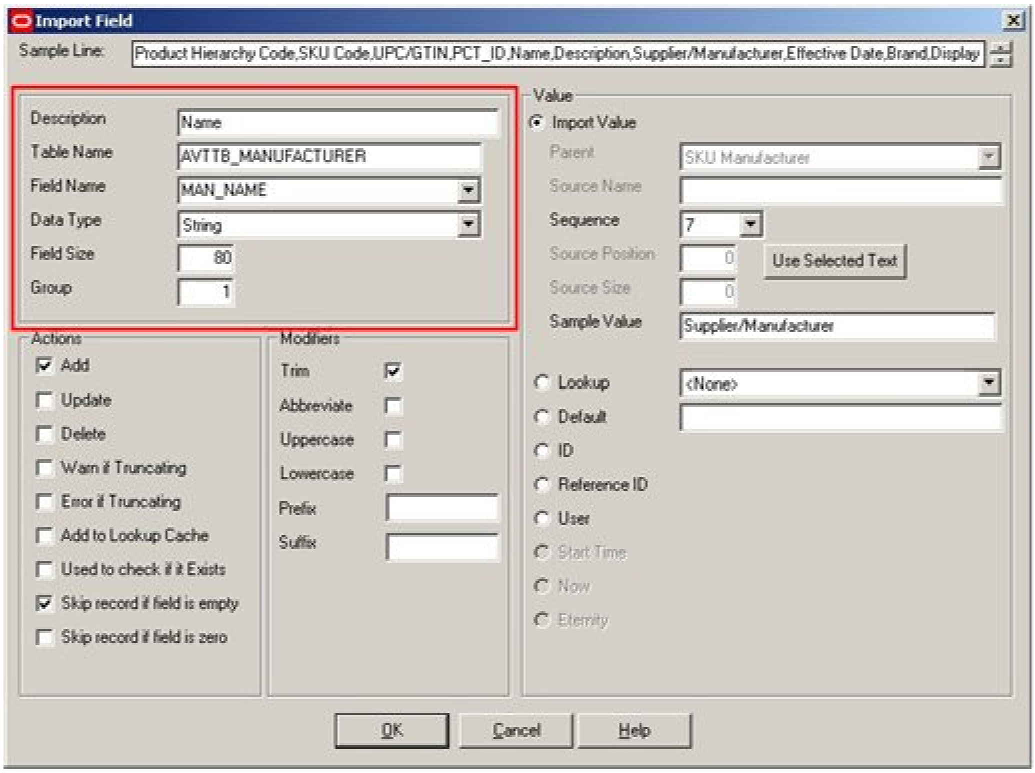
Modifiers
Trim ☑
Abbreviate □
Uppercase □
Lowercase □
Prefix
Suffix

Value
● Import Value
Parent: SKU Manufacturer
Source Name
Sequence: 7
Source Position: 0
Source Size: 0
Sample Value: Supplier/Manufacturer
○ Lookup: <None>
○ Default
○ ID
○ Reference ID
○ User
○ Start Time
○ Now
○ Eternity

- Sample Line shows a full line of data. (The line selected is determined by the designated Start Line in the Import Design GUI.)
- Sample Value shows that value defined in the Value frame by the selected option.

Description Frame

The **Description Frame** enables users to specify information pertinent to the table that the data is to be imported into.


Import Field

Sample Line: Product Hierarchy Code,SKU Code,UPC/GTIN,PCT_ID,Name,Description,Supplier/Manufacturer,Effective Date,Brand,Display

Description: Name
Table Name: AVTTB_MANUFACTURER
Field Name: MAN_NAME
Data Type: String
Field Size: 80
Group: 1

Actions
☑ Add
☐ Update
☐ Delete
☐ Warn if Truncating
☐ Error if Truncating
☐ Add to Lookup Cache
☐ Used to check if it Exists
☑ Skip record if field is empty
☐ Skip record if field is zero

Modifiers
Trim ☑
Abbreviate ☐
Uppercase ☐
Lowercase ☐
Prefix:
Suffix:

Value
● Import Value
Parent: SKU Manufacturer
Source Name:
Sequence: 7
Source Position: 0
Source Size: 0
Sample Value: Supplier/Manufacturer
○ Lookup <None>
○ Default
○ ID
○ Reference ID
○ User
○ Start Time
○ Now
○ Eternity

Description is a user input name that is used to describe the item of data bring imported. In this example, Block Name indicates that the user is bringing a block name into the database.

Table Name is the name of the table in the database into which the item of data must be written. In this example, the AVTTB_MANUFACTURER table name has been added.

Field Name is the name of the specific field within the table that the item of data is to be written to. Once the table name has been added, the field name can be selected from a drop down list.

Data Type and **Field Size** will be populated automatically after the Field Name has been selected.

Group indicates which table the data is to be added to.

Each Import Definition is made up of a number of import fields. If these import fields are to be used to put data into two or more tables, then the group number must be used to indicate which table the data is to be imported to.

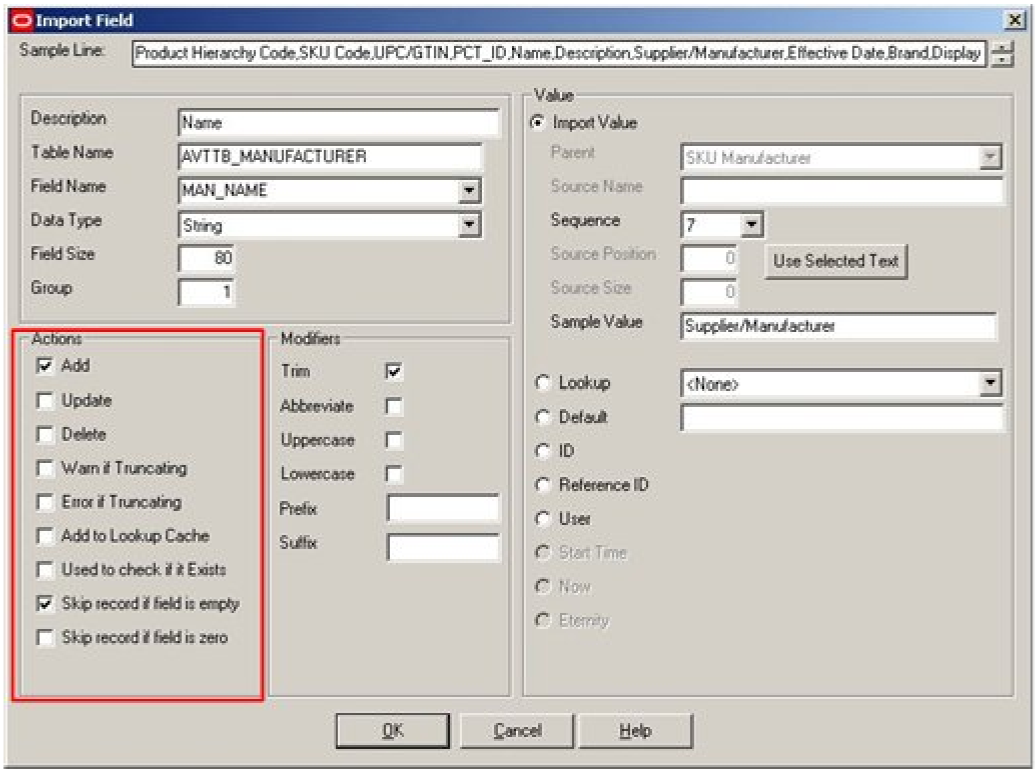
For example, the Fixture Import type has a number of Import Definitions. The Block Definition Import Definition writes information to the AVTTB_BLOCK_DEFINITION and the AVTTB_BLOCK_GROUP_LINK tables. The AVTTB_BLOCK_DEFINITION

TABLE has been defined as Group 1 and the AVTTB_BLOCK_GROUP_LINK table as Group 2.

Note: Data Importer works by producing a series of SQL Update statements for the data to be imported. Assigning each table to a unique group means that when Data Importer produces the Update statements, it assigns the right information to the statement for that table. If you do not assign tables to groups during configuring the import field, Data Importer will try and produce a single SQL statement that writes data to multiple tables.

Actions Frame

The **Actions Frame** enables users to specify how to manipulate the data being imported.



Import Field

Sample Line: Product Hierarchy Code,SKU Code,UPC/GTIN,PCT_ID,Name,Description,Supplier/Manufacturer,Effective Date,Brand,Display

Description: Name
Table Name: AVTTB_MANUFACTURER
Field Name: MAN_NAME
Data Type: String
Field Size: 80
Group: 1

Actions
☑ Add
☐ Update
☐ Delete
☐ Warn if Truncating
☐ Error if Truncating
☐ Add to Lookup Cache
☐ Used to check if it Exists
☑ Skip record if field is empty
☐ Skip record if field is zero

Modifiers
Trim ☑
Abbreviate ☐
Uppercase ☐
Lowercase ☐
Prefix:
Suffix:

Value
● Import Value
Parent: SKU Manufacturer
Source Name:
Sequence: 7
Source Position: 0
Source Size: 0
Sample Value: Supplier/Manufacturer
○ Lookup: <None>
○ Default:
○ ID
○ Reference ID
○ User
○ Start Time
○ Now
○ Eternity

Add indicates that the data is to be added.

Update indicates that the data is to be updated.

Delete indicates that existing data is to be deleted out of the specified table.

Warn if Truncating is only enabled for String data types. If selected, it adds a note to the Import Error Log that the imported value exceeded the field size.

If this option is selected the data will still be imported.

Error if Truncating is only enabled for String data types. If selected, it adds a note to the Import Error Log that the imported value exceeded the field size and does not import the data.

Left Justify is not enabled in this release.

Add to Lookup Cache adds the result of the lookup so that when a new record is added during that specific import, it is first checked against the Lookup cache. This improves performance as it means that not all imported data needs to be checked against the database.

Used to Check if it Exists indicates that this field will be used to check if the record currently exists in the database. (This option is often used in conjunction with the lookup option in the Value frame).

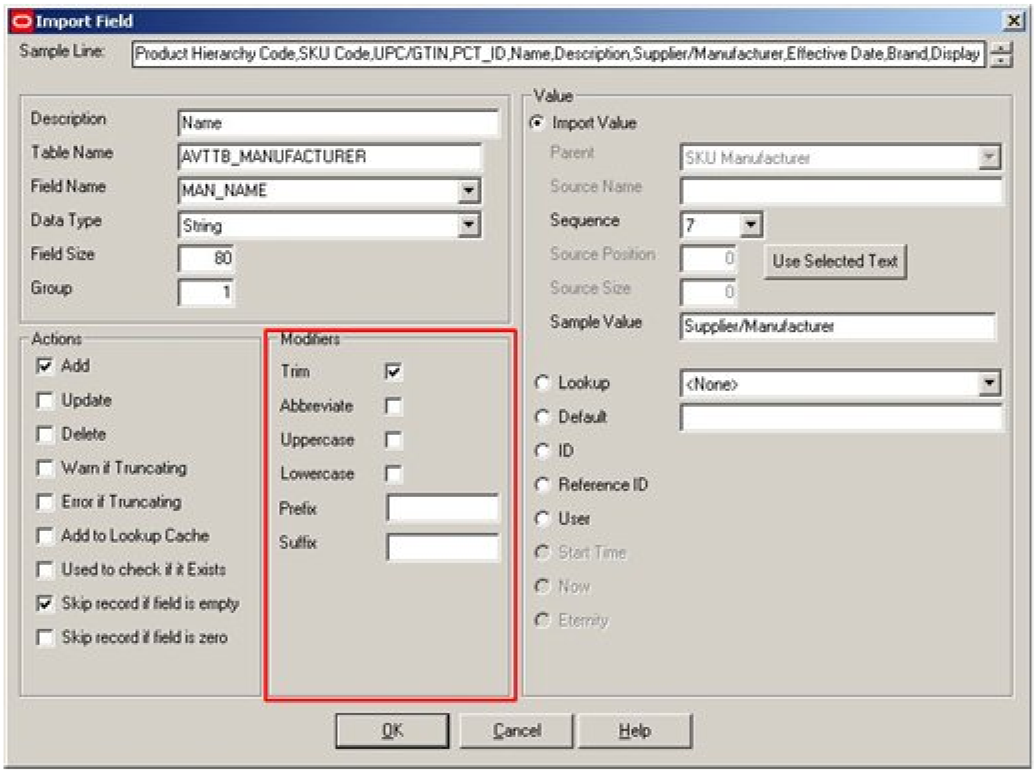
If the record exists, and the Update option is not enabled, the record will be skipped.

Skip Record if Field is Empty will result in this record being skipped during import if the field has been left empty.

Skip Record if Field is Zero will result in this record being skipped if a numeric field is zero. (An example might be if a product dimension is zero).

Modifiers Frame

The **Modifiers Frame** is used to modify the data being imported.



Import Field

Sample Line: Product Hierarchy Code,SKU Code,UPC/GTIN,PCT_ID,Name,Description,Supplier/Manufacturer,Effective Date,Brand,Display

Description: Name

Table Name: AVTTB_MANUFACTURER

Field Name: MAN_NAME

Data Type: String

Field Size: 80

Group: 1

Actions

☑ Add
☐ Update
☐ Delete
☐ Warn if Truncating
☐ Error if Truncating
☐ Add to Lookup Cache
☐ Used to check if it Exists
☑ Skip record if field is empty
☐ Skip record if field is zero

Modifiers

Trim ☑
Abbreviate ☐
Uppercase ☐
Lowercase ☐
Prefix
Suffix

Value

● Import Value
Parent: SKU Manufacturer
Source Name:
Sequence: 7
Source Position: 0
Source Size: 0
Sample Value: Supplier/Manufacturer
○ Lookup: <None>
○ Default
○ ID
○ Reference ID
○ User
○ Start Time
○ Now
○ Eternity

Trim will remove leading and trailing spaces.

Abbreviate will truncate text strings if they exceed the maximum size of that field in the database.

Uppercase will ensure that the imported data is all converted to uppercase.

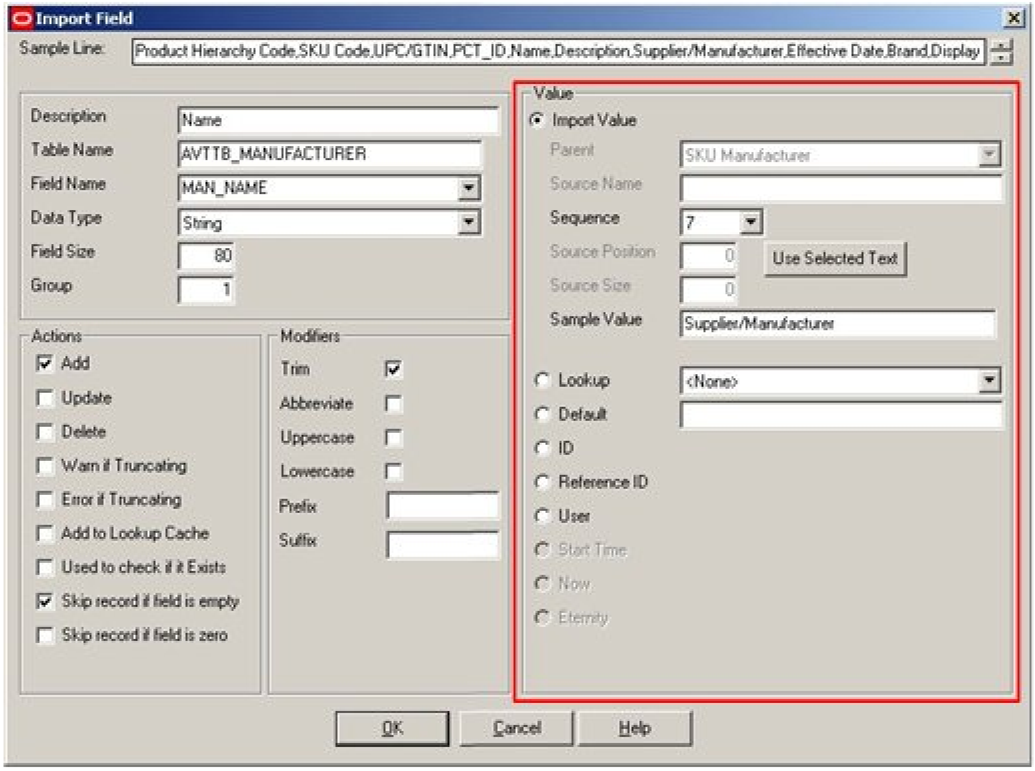
Lowercase will ensure that the imported data is all converted to lowercase.

Prefix will add the specified prefix to the imported value if the data being imported is a text string.

Suffix will add the specified suffix to the imported value if the data being imported is a text string.

Value Frame

The **Value Frame** enables the user to select the specific data to be imported.



Sample Line: Product Hierarchy Code,SKU Code,UPC/GTIN,PCT_ID,Name,Description,Supplier/Manufacturer,Effective Date,Brand,Display

Description: Name
Table Name: AVTTB_MANUFACTURER
Field Name: MAN_NAME
Data Type: String
Field Size: 80
Group: 1

Actions
- ☑ Add
- ☐ Update
- ☐ Delete
- ☐ Warn if Truncating
- ☐ Error if Truncating
- ☐ Add to Lookup Cache
- ☐ Used to check if it Exists
- ☑ Skip record if field is empty
- ☐ Skip record if field is zero

Modifiers
- Trim ☑
- Abbreviate ☐
- Uppercase ☐
- Lowercase ☐
- Prefix:
- Suffix:

Value
- ● Import Value
  - Parent: SKU Manufacturer
  - Source Name:
  - Sequence: 7
  - Source Position: 0
  - Source Size: 0
  - Sample Value: Supplier/Manufacturer
- ○ Lookup: <None>
- ○ Default:
- ○ ID
- ○ Reference ID
- ○ User
- ○ Start Time
- ○ Now
- ○ Eternity

Import Value Radio Button - when selected, this indicated that the data will be imported (as opposed to being allocated in the form of an ID, etc).

When importing a value, it is possible to include either a default value, and/or a lookup along with the item of data being imported.

Parent is the name of the file from which the data will be imported. It must be located in the Macro Space Management Import Directory.

Source Name is only enabled when importing from an XML file. The drop down list will display all properties and attributes for the definition (hierarchy level); excluding properties which represent child objects.

Note that if a parent is defined, then the properties/attributes should relate to the parent

Sequence is only enabled when importing from CSV files and defines which item in the list will be imported.

Source Position and **Source Size** are only enabled when importing from Fixed Width Files and define the starting character position and width of the field.

Use Selected Text is only enabled when importing from Fixed Width Files and allows the user to select part of the text in the sample line in order to define both the Source Position and Source Size.

Sample Value is an example of the data that will be imported.

Lookup will use the import value and convert it to an ID.

For example, the imported value might represent the Product Code, but then convert this to the Product ID, which is often done to check whether the record exists.

Note: Lookup works in two different ways, depending on whether Used to Check if Exists has been selected in the Action Frame. If Used to Check if Exists has not been selected, the lookup will convert the value being imported from one value to another. The value will then be imported in the converted form. If Used to Check if Exists has been selected, the lookup will convert the value being imported from one value to another, and use that value to check if the data being imported currently exists in the database. The data will still be imported in the original form.

Default will use the specified default value, instead of reading the Import Value.

ID is used when the application has to determine the next ID.

Reference ID is used when multiple groups are defined, and is used in the subordinate records to refer to the new ID that was used in the main record.

Note: This is a frequent source of error in defining imports. If BLK_ID has been imported earlier during the Import into the AVTTB_BLOCK_DEFINITION table, and it is to be added again to the AVTTB_BLOCK_GROUP_LINK table, then it should be added as a Reference ID and not as an ID. Attempting to adding it again as an ID will cause Data Importer to error with the message "Duplicate Primary Key".

User sets the field value to the current system user for the Modified_By, or Created_By fields.

Start Time sets a date field value to the date and time of when this particular import process was started. Unlike the *Now* option, this ensures that all records have the same date time stamp.

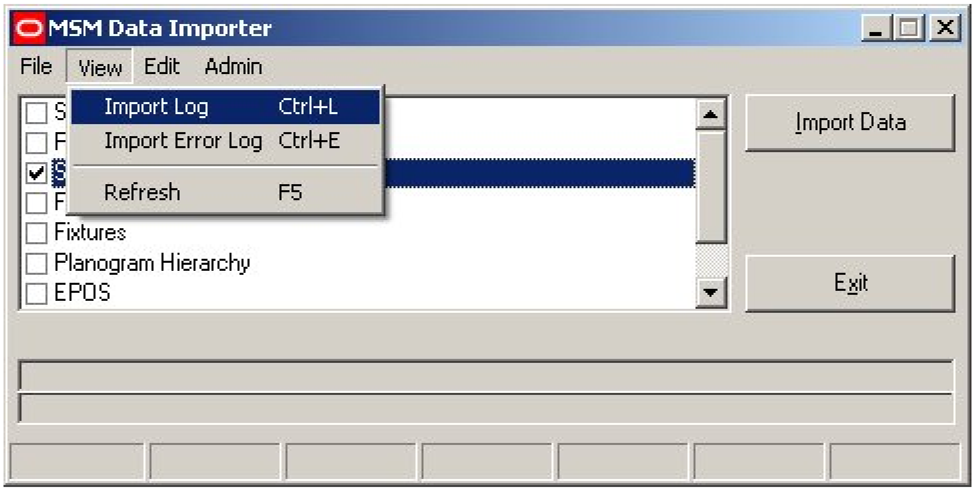
Now sets the date field value to the current date and time for the instant the item of data was imported.

Eternity sets a date field value to the indefinite future; i.e. 31st December 2999.

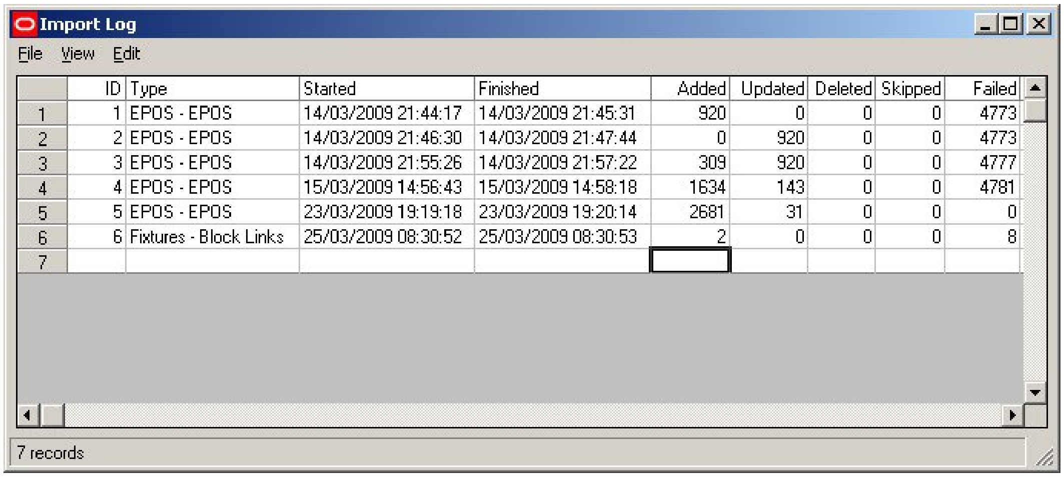
Import Log

Overview of the Import Log

The **Import Log** is activated from the View pull down menu.



This will bring up the Import Log.



	ID	Type	Started	Finished	Added	Updated	Deleted	Skipped	Failed
1	1	EPOS - EPOS	14/03/2009 21:44:17	14/03/2009 21:45:31	920	0	0	0	4773
2	2	EPOS - EPOS	14/03/2009 21:46:30	14/03/2009 21:47:44	0	920	0	0	4773
3	3	EPOS - EPOS	14/03/2009 21:55:26	14/03/2009 21:57:22	309	920	0	0	4777
4	4	EPOS - EPOS	15/03/2009 14:56:43	15/03/2009 14:58:18	1634	143	0	0	4781
5	5	EPOS - EPOS	23/03/2009 19:19:18	23/03/2009 19:20:14	2681	31	0	0	0
6	6	Fixtures - Block Links	25/03/2009 08:30:52	25/03/2009 08:30:53	2	0	0	0	8
7									

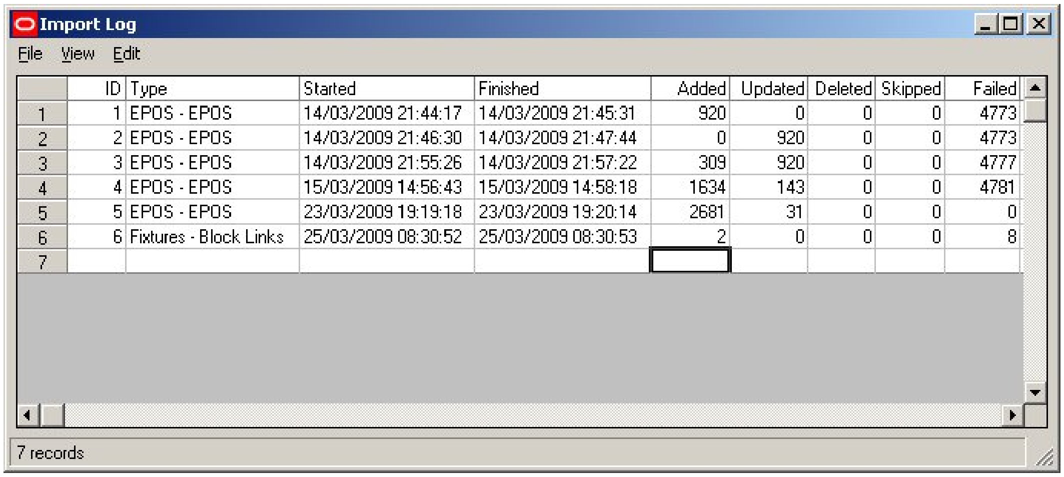
7 records

The Import Log contains information on recent imports.

In addition it can be used to access the Import Error Log for specific imports (View Menu).

The Import Log

The **Import Log** contains information on Recent Imports.



	ID	Type	Started	Finished	Added	Updated	Deleted	Skipped	Failed
1	1	EPOS - EPOS	14/03/2009 21:44:17	14/03/2009 21:45:31	920	0	0	0	4773
2	2	EPOS - EPOS	14/03/2009 21:46:30	14/03/2009 21:47:44	0	920	0	0	4773
3	3	EPOS - EPOS	14/03/2009 21:55:26	14/03/2009 21:57:22	309	920	0	0	4777
4	4	EPOS - EPOS	15/03/2009 14:56:43	15/03/2009 14:58:18	1634	143	0	0	4781
5	5	EPOS - EPOS	23/03/2009 19:19:18	23/03/2009 19:20:14	2681	31	0	0	0
6	6	Fixtures - Block Links	25/03/2009 08:30:52	25/03/2009 08:30:53	2	0	0	0	8
7									

7 records

ID is assigned by the software and is not user customizable.

Type is the name of the import type, followed by the specific import definition.

Fixtures - Fixture Information is thus the Fixture Information import definition in the Fixtures import type.

Started and **Finished** are when the import started and finished.

Added is the number of records added.

Updated is the number of records updated.

Deleted is the number of records deleted.

Skipped is the number of records omitted. (This will occur when the Skip Record if Field is Empty or Skip Record if Field is Zero options are used when importing data)

User and **Machine** indicate which user carried out the import and which computer was used.

Number of Records Imported

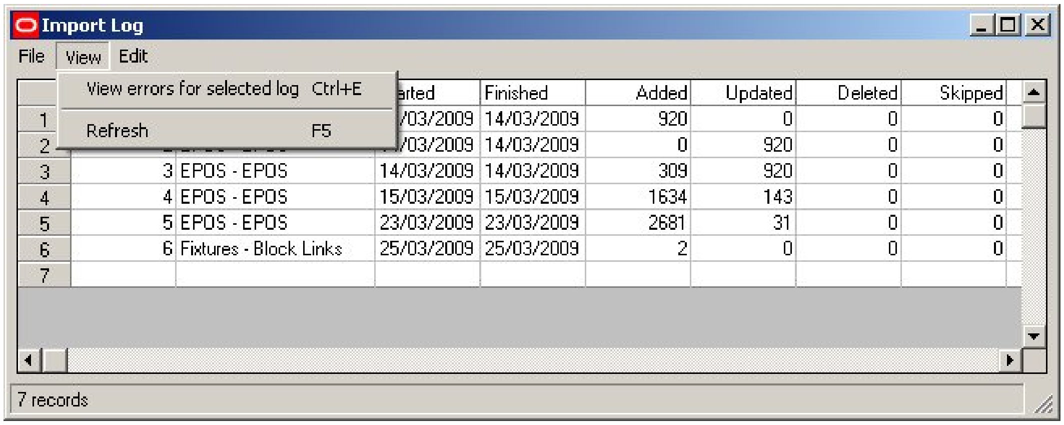
The **number of records imported** is calculated as follows for each import definition within an import type:

- The number of items of data to be input is calculated.
- This is multiplied by the number of tables the data is to be inserted into.

If there are 20 items of data to be imported for a particular definition, and the definition references 3 tables, the number of records imported will show as 60 (20 x 3).

Import Log Options

Several options are available from the Menus in the Import Log.

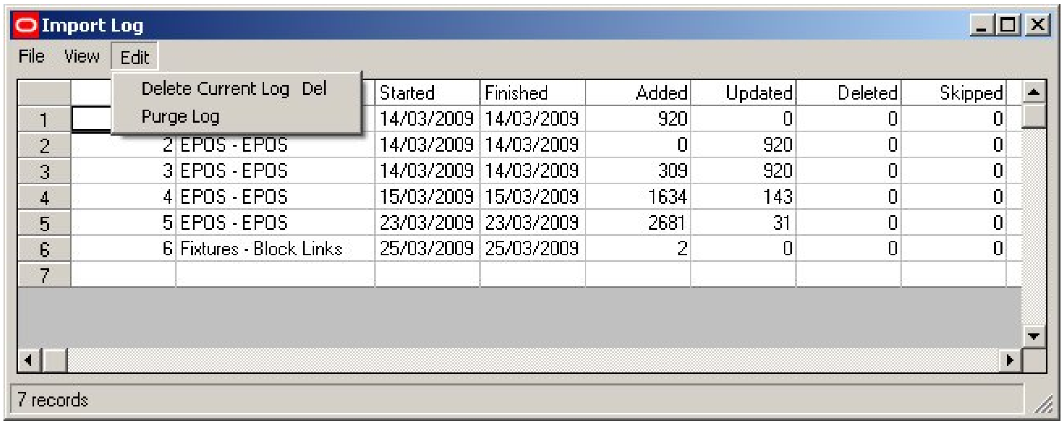


			arted	Finished	Added	Updated	Deleted	Skipped
1			/03/2009	14/03/2009	920	0	0	0
2			/03/2009	14/03/2009	0	920	0	0
3	3	EPOS - EPOS	14/03/2009	14/03/2009	309	920	0	0
4	4	EPOS - EPOS	15/03/2009	15/03/2009	1634	143	0	0
5	5	EPOS - EPOS	23/03/2009	23/03/2009	2681	31	0	0
6	6	Fixtures - Block Links	25/03/2009	25/03/2009	2	0	0	0
7								

7 records

The **View** menu allows two options:

- View errors for selected log allow users to view the errors for a selected import definition. (This is often a better option than viewing the Error Log directly, as it allows users to see specific errors, rather than all errors for all imports).
- Refresh refreshes the current view.



			Started	Finished	Added	Updated	Deleted	Skipped
1			14/03/2009	14/03/2009	920	0	0	0
2	2	EPOS - EPOS	14/03/2009	14/03/2009	0	920	0	0
3	3	EPOS - EPOS	14/03/2009	14/03/2009	309	920	0	0
4	4	EPOS - EPOS	15/03/2009	15/03/2009	1634	143	0	0
5	5	EPOS - EPOS	23/03/2009	23/03/2009	2681	31	0	0
6	6	Fixtures - Block Links	25/03/2009	25/03/2009	2	0	0	0
7								

7 records

The **Edit** menu allows two options:

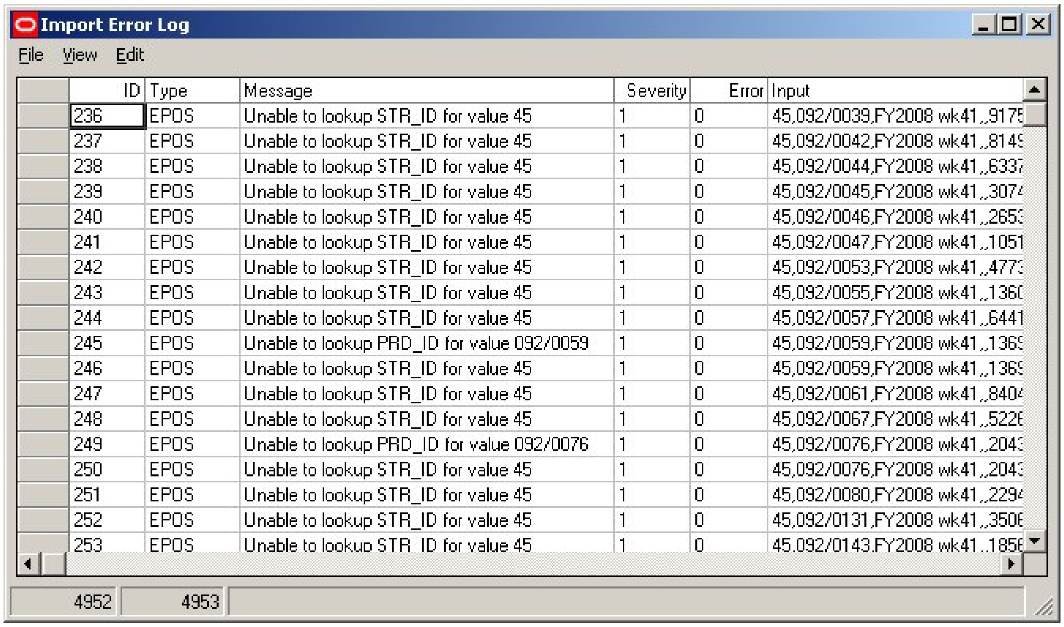
- Delete Current Log deletes the highlighted Import log.
- Purge Log deletes all import logs.

Import Error Log

Overview of the Import Error Log

The **Import Error Log** contains details of all the errors that have occurred during data imports.

It can be called directly from the main data importer dialogue box View menu, in which case it will show all errors in the import log. Alternatively, the Import Error Log may be called via the Import Log, allowing specific errors to be viewed.



ID	Type	Message	Severity	Error	Input
236	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0039,FY2008 wk41,,9175
237	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0042,FY2008 wk41,,8149
238	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0044,FY2008 wk41,,6337
239	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0045,FY2008 wk41,,3074
240	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0046,FY2008 wk41,,2653
241	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0047,FY2008 wk41,,1051
242	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0053,FY2008 wk41,,4773
243	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0055,FY2008 wk41,,1360
244	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0057,FY2008 wk41,,6441
245	EPOS	Unable to lookup PRD_ID for value 092/0059	1	0	45,092/0059,FY2008 wk41,,1369
246	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0059,FY2008 wk41,,1369
247	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0061,FY2008 wk41,,8404
248	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0067,FY2008 wk41,,5226
249	EPOS	Unable to lookup PRD_ID for value 092/0076	1	0	45,092/0076,FY2008 wk41,,2043
250	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0076,FY2008 wk41,,2043
251	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0080,FY2008 wk41,,2294
252	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0131,FY2008 wk41,,3506
253	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0143,FY2008 wk41,,1856

ID is assigned by the software and is not user customizable.

Type is the Import Definition Type

Message is a description of the error.

Severity is the severity of the error. 1 is an error, 2 is a warning.

The Import can set to warn only if (for example in an import definition) Data Importer is configured to truncate a text string being imported.

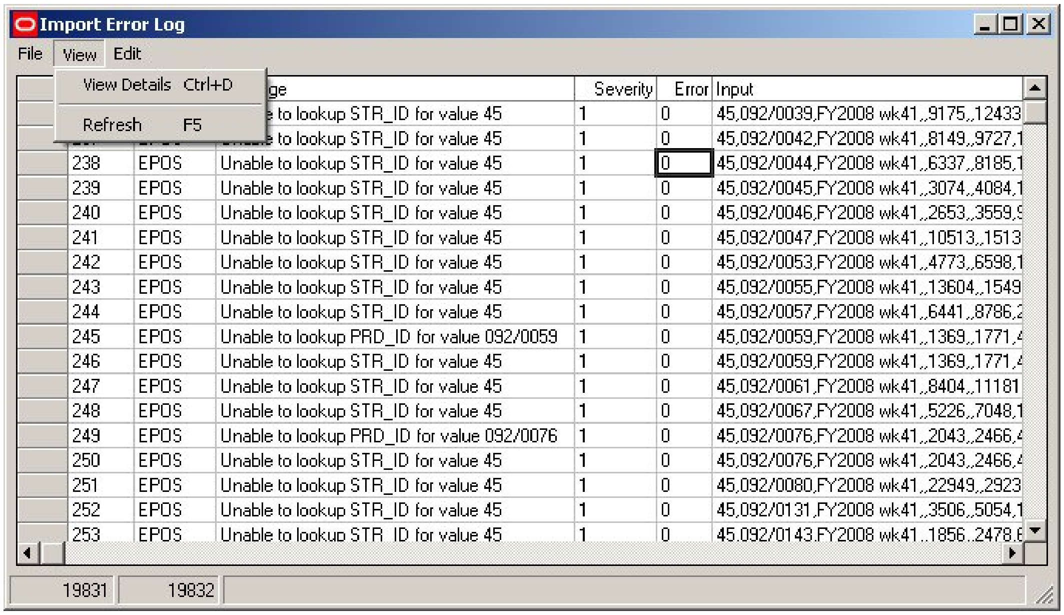
Error is the code number of the error that has occurred.

Input is the data being used by Data Importer.

Output is the SQL statement that Data Importer is attempting to compose.

Import Error Log Options

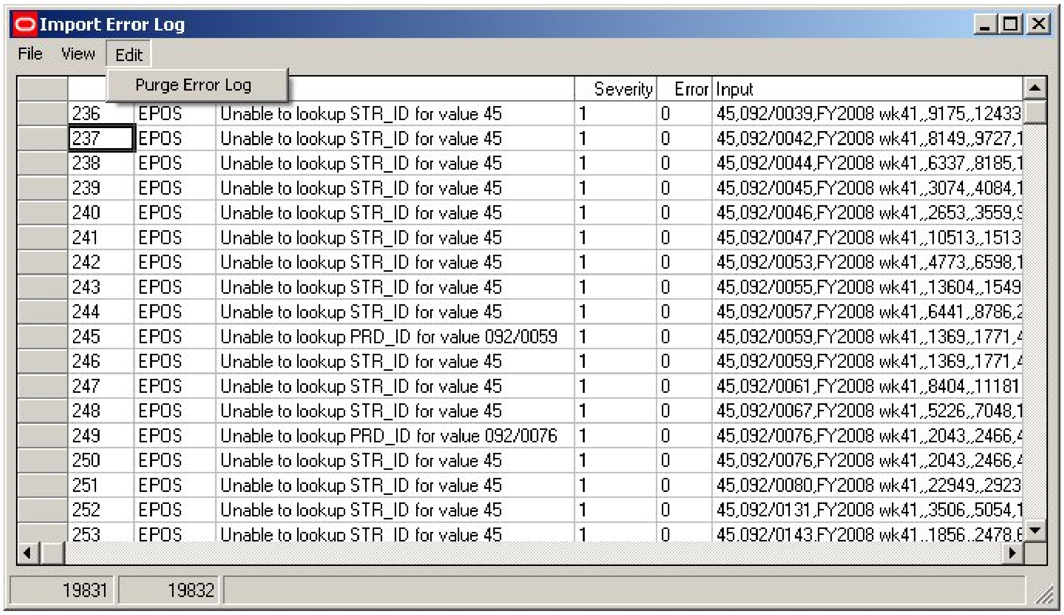
Several **options** are available from the Menus in the Import Error Log.



		ge	Severity	Error	Input
		e to lookup STR_ID for value 45	1	0	45,092/0039,FY2008 wk41,,9175,,12433
		to lookup STR_ID for value 45	1	0	45,092/0042,FY2008 wk41,,8149,,9727,1
238	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0044,FY2008 wk41,,6337,,8185,1
239	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0045,FY2008 wk41,,3074,,4084,1
240	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0046,FY2008 wk41,,2653,,3559,9
241	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0047,FY2008 wk41,,10513,,1513
242	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0053,FY2008 wk41,,4773,,6598,1
243	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0055,FY2008 wk41,,13604,,1549
244	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0057,FY2008 wk41,,6441,,8786,2
245	EPOS	Unable to lookup PRD_ID for value 092/0059	1	0	45,092/0059,FY2008 wk41,,1369,,1771,4
246	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0059,FY2008 wk41,,1369,,1771,4
247	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0061,FY2008 wk41,,8404,,11181
248	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0067,FY2008 wk41,,5226,,7048,1
249	EPOS	Unable to lookup PRD_ID for value 092/0076	1	0	45,092/0076,FY2008 wk41,,2043,,2466,4
250	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0076,FY2008 wk41,,2043,,2466,4
251	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0080,FY2008 wk41,,22949,,2923
252	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0131,FY2008 wk41,,3506,,5054,1
253	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0143,FY2008 wk41,,1856,,2478,6

The **View** menu allows two options:

- View Details allows user to view a detailed report for a selected error.
- Refresh refreshes the current view.



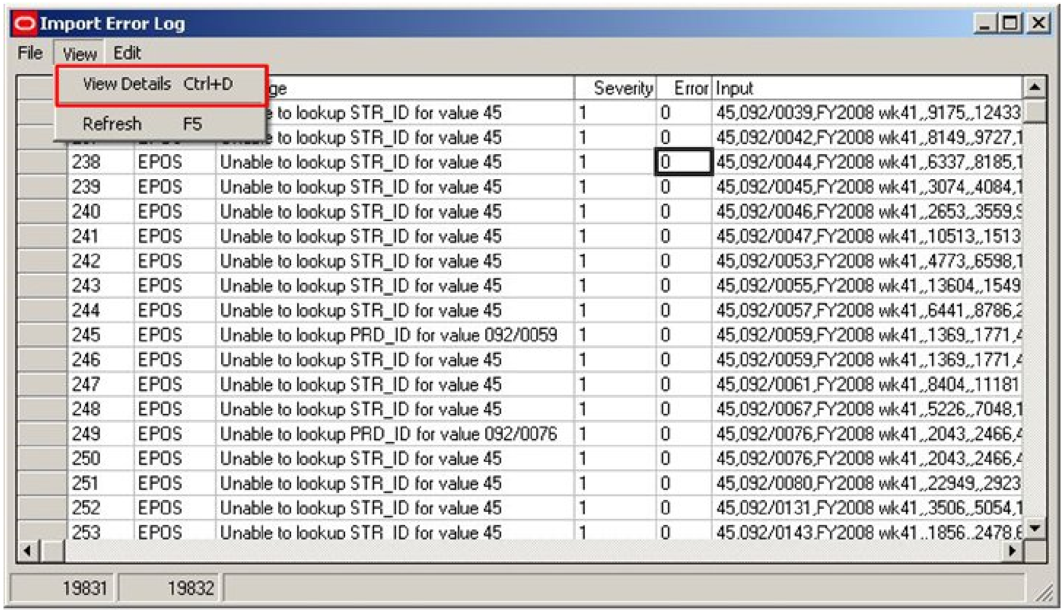
			Severity	Error	Input
236	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0039,FY2008 wk41,,9175,,12433
237	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0042,FY2008 wk41,,8149,,9727,1
238	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0044,FY2008 wk41,,6337,,8185,1
239	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0045,FY2008 wk41,,3074,,4084,1
240	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0046,FY2008 wk41,,2653,,3559,9
241	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0047,FY2008 wk41,,10513,,1513
242	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0053,FY2008 wk41,,4773,,6598,1
243	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0055,FY2008 wk41,,13604,,1549
244	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0057,FY2008 wk41,,6441,,8786,2
245	EPOS	Unable to lookup PRD_ID for value 092/0059	1	0	45,092/0059,FY2008 wk41,,1369,,1771,4
246	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0059,FY2008 wk41,,1369,,1771,4
247	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0061,FY2008 wk41,,8404,,11181
248	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0067,FY2008 wk41,,5226,,7048,1
249	EPOS	Unable to lookup PRD_ID for value 092/0076	1	0	45,092/0076,FY2008 wk41,,2043,,2466,4
250	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0076,FY2008 wk41,,2043,,2466,4
251	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0080,FY2008 wk41,,22949,,2923
252	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0131,FY2008 wk41,,3506,,5054,1
253	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0143,FY2008 wk41,,1856,,2478,6

19831 19832

The **Edit** menu allows users to delete the highlighted error log.

Import Error Details

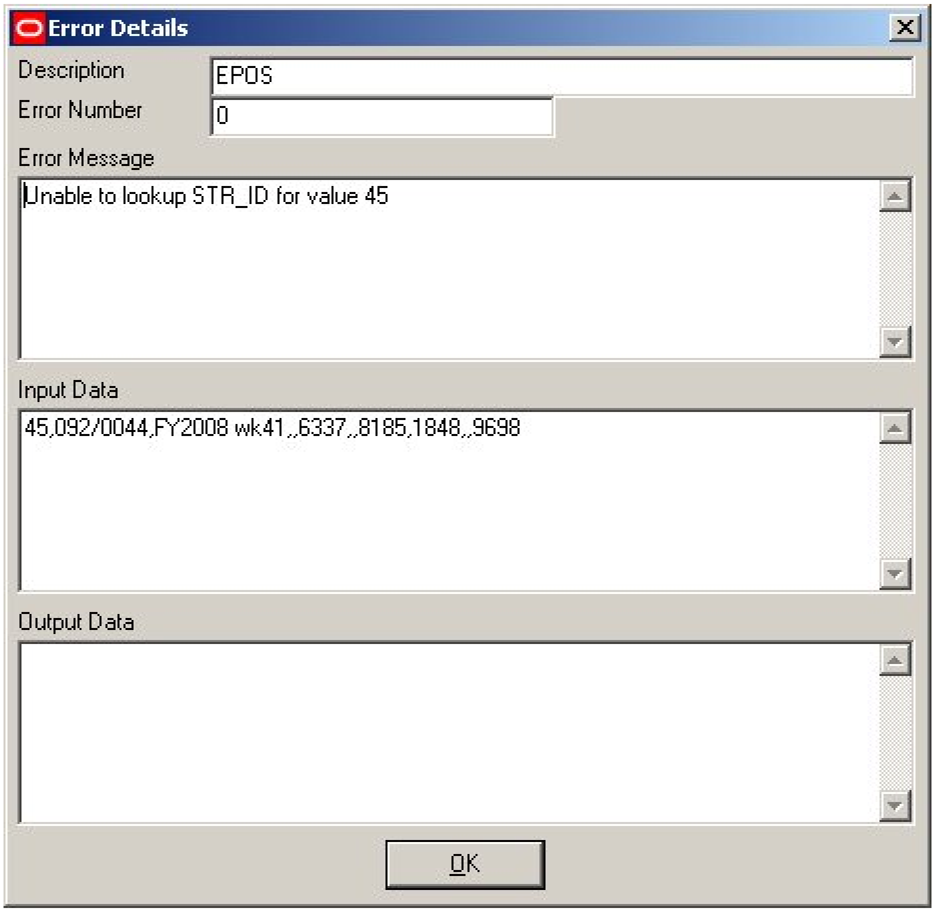
To see the error in more detail, it is possible to do so by selecting the View Details option from the View pull down menu.



		ge	Severity	Error	Input
		to lookup STR_ID for value 45	1	0	45,092/0039,FY2008 wk41,,9175,,12433
		to lookup STR_ID for value 45	1	0	45,092/0042,FY2008 wk41,,8149,,9727,1
238	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0044,FY2008 wk41,,6337,,8185,1
239	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0045,FY2008 wk41,,3074,,4084,1
240	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0046,FY2008 wk41,,2653,,3559,9
241	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0047,FY2008 wk41,,10513,,1513
242	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0053,FY2008 wk41,,4773,,6598,1
243	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0055,FY2008 wk41,,13604,,1549
244	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0057,FY2008 wk41,,6441,,8786,2
245	EPOS	Unable to lookup PRD_ID for value 092/0059	1	0	45,092/0059,FY2008 wk41,,1369,,1771,4
246	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0059,FY2008 wk41,,1369,,1771,4
247	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0061,FY2008 wk41,,8404,,11181
248	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0067,FY2008 wk41,,5226,,7048,1
249	EPOS	Unable to lookup PRD_ID for value 092/0076	1	0	45,092/0076,FY2008 wk41,,2043,,2466,4
250	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0076,FY2008 wk41,,2043,,2466,4
251	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0080,FY2008 wk41,,22949,,2923
252	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0131,FY2008 wk41,,3506,,5054,1
253	EPOS	Unable to lookup STR_ID for value 45	1	0	45,092/0143,FY2008 wk41,,1856,,2478,6

19831 19832

This will bring up the Error Details dialogue box, which allows the error to be seen more clearly.



Description: EPOS

Error Number: 0

Error Message:

Unable to lookup STR_ID for value 45

Input Data:

45,092/0044,FY2008 wk41,,6337,,8185,1848,,9698

Output Data:

This is useful if the data is extensive and difficult to see in its entirety in the Import Error Log.

4

Configuring Data Importer

Preparing to Configure Data Importer

Overview of Preparations

> **Note**: The examples given in this section of the User Guide are based on data imports set up for Macro Space Management v5.25**.** If applied to other builds (or to a schema modified by an upgrade script) the precise example may no longer work.
>
> **Important Note**: Import of data using Data Importer It is recommended that the user back up the database before import. Should there be errors during the import, this will enable the user to restore the database to its condition before the import. There should not be errors resulting from a well configured up Data Import - but as backing up a database is quick to carry out, it is a prudent precaution to take.

Before starting to configure Data Importer the following must true:

- The user must be able to access the central Macro Space Management database.
- The use must have software capable of reading the central Macro Space Management database.
- The user should be familiar with the schema for the database.

For configuring a specific import:

1. The user should list all the tables that will require data to be inserted
2. For each affected table the user should list all fields that will require data inserted
3. For all data to be imported the user should identify if a look up is required.

> **Important**: All non-nullable fields in a table must have a value inserted. This may be done in one of two ways:

4. Directly inserted from Data Importer
5. Added as a default from the database.

(Adding as a default from the database is only possible if that column has had a default value defined).

Accessing and Viewing the Central Macro Space Management Database

The central **Macro Space Management** database will be located on a specific server (or servers) within the customers network.

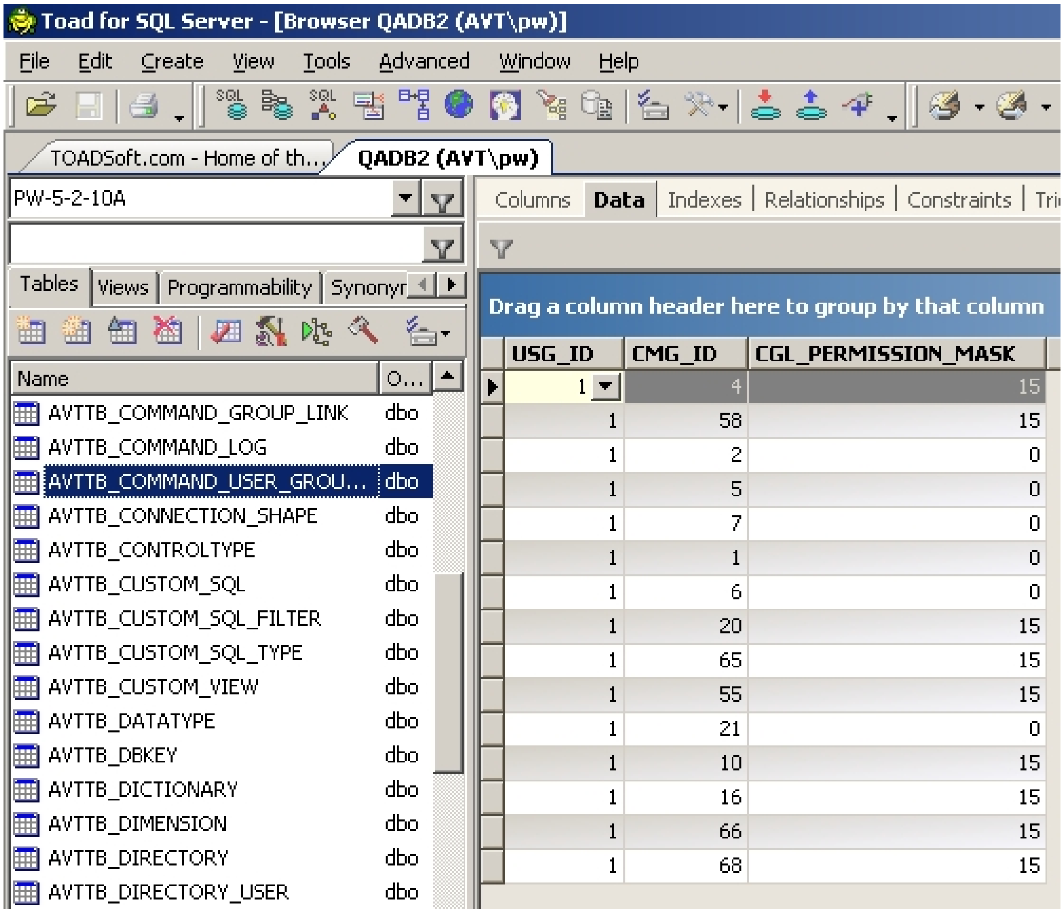
The user will need to examine data in the database during the configuration of an Import Definition. They must accordingly have access rights to those servers.

The user must also have appropriate software installed on their computer allowing them to view the contents of the database. Suitable software includes:

- Microsoft SQL Server Management Studio

- Microsoft Enterprise Manager and/or Query Analyzer
- Toad

In the example below, some of the tables in the database are listed in the left hand pane, while the contents of the tables are listed in the right hand one.


Toad for SQL Server - [Browser QADB2 (AVT\pw)]
File Edit Create View Tools Advanced Window Help
TOADSoft.com - Home of th...
QADB2 (AVT\pw)
PW-5-2-10A
Tables Views Programmability Synonyr
Columns Data Indexes Relationships Constraints Tri
Drag a column header here to group by that column

Name	O...
AVTTB_COMMAND_GROUP_LINK	dbo
AVTTB_COMMAND_LOG	dbo
AVTTB_COMMAND_USER_GROU...	dbo
AVTTB_CONNECTION_SHAPE	dbo
AVTTB_CONTROLTYPE	dbo
AVTTB_CUSTOM_SQL	dbo
AVTTB_CUSTOM_SQL_FILTER	dbo
AVTTB_CUSTOM_SQL_TYPE	dbo
AVTTB_CUSTOM_VIEW	dbo
AVTTB_DATATYPE	dbo
AVTTB_DBKEY	dbo
AVTTB_DICTIONARY	dbo
AVTTB_DIMENSION	dbo
AVTTB_DIRECTORY	dbo
AVTTB_DIRECTORY_USER	dbo

USG_ID	CMG_ID	CGL_PERMISSION_MASK
1	4	15
1	58	15
1	2	0
1	5	0
1	7	0
1	1	0
1	6	0
1	20	15
1	65	15
1	55	15
1	21	0
1	10	15
1	16	15
1	66	15
1	68	15

Planning the Import

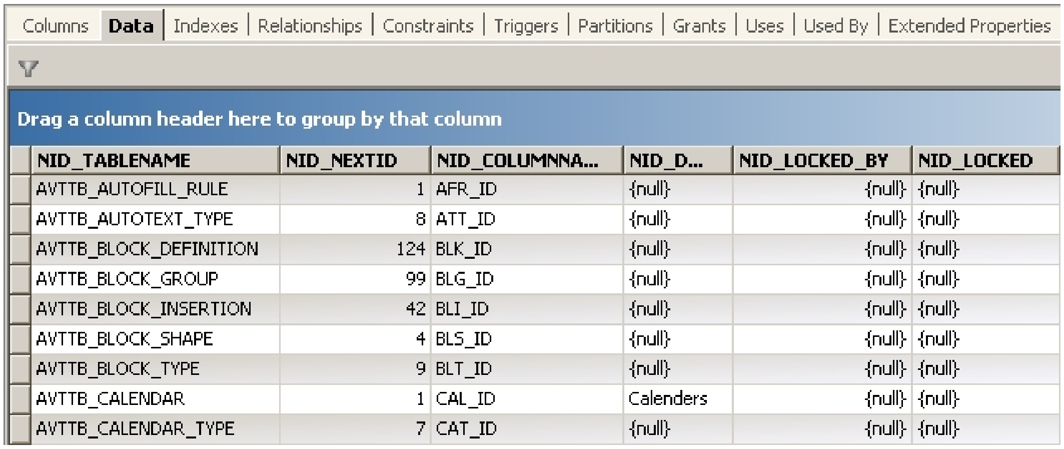
It is recommended the following sequence of actions is carried when an import is configured and executed:

- The information available for import is documented. (This can be in spreadsheet or table form)
- The Schema is consulted to find all tables related to the intended import.
- The fields in the table(s) are listed and the fields that want populating identified.
- The Import Type is given a name.
- The Import Type is broken down into Import Definitions.
- Each Import definition is broken down into Import fields.
- Any Lookups required are identified.
- A spreadsheet template should be produced for entering the data to be input.
- A copy of the populated spreadsheet containing just the first few lines of data is put in the Macro Space Management import directory.
- Date Importer is configured.
- Data Importer is tested using the initial few lines of data.

- The full import is carried out.

The AVTTB_NEXTID Table

The **AVTTB_NEXT_ID Table** contains the values to be assigned for the next instance of every primary key in the Macro Space Management database.



NID_TABLENAME	NID_NEXTID	NID_COLUMNNA...	NID_D...	NID_LOCKED_BY	NID_LOCKED
AVTTB_AUTOFILL_RULE	1	AFR_ID	{null}	{null}	{null}
AVTTB_AUTOTEXT_TYPE	8	ATT_ID	{null}	{null}	{null}
AVTTB_BLOCK_DEFINITION	124	BLK_ID	{null}	{null}	{null}
AVTTB_BLOCK_GROUP	99	BLG_ID	{null}	{null}	{null}
AVTTB_BLOCK_INSERTION	42	BLI_ID	{null}	{null}	{null}
AVTTB_BLOCK_SHAPE	4	BLS_ID	{null}	{null}	{null}
AVTTB_BLOCK_TYPE	9	BLT_ID	{null}	{null}	{null}
AVTTB_CALENDAR	1	CAL_ID	Calenders	{null}	{null}
AVTTB_CALENDAR_TYPE	7	CAT_ID	{null}	{null}	{null}

It is sometimes necessary to use PRE-SQL statements during the import process to ensure the values in the table are correct.

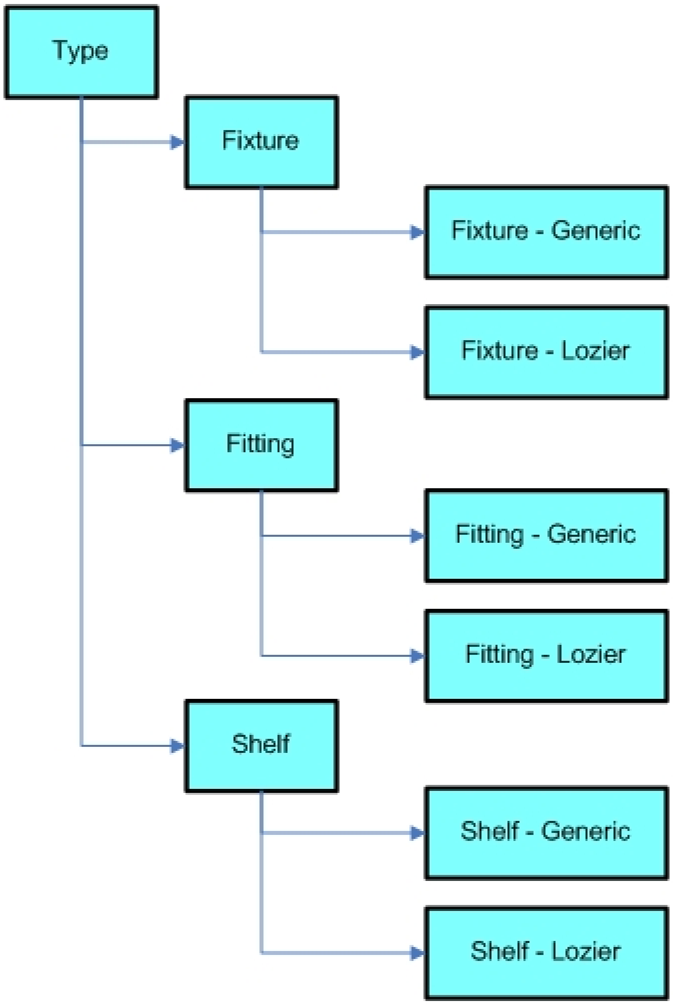
5

Worked Example - Fixture Hierarchy

Planning Adding Fixture Groups

The Required Fixture Hierarchy

The **Required Fixture Hierarchy** has been defined as follows:



The hierarchy has three levels:

- Type is the default root of the hierarchy and cannot be changed.
- Fixture, Fitting and Shelf form the three options for the next level in the hierarchy.
- The final level has two options for each type of equipment, for example Fixture - Generic and Fixture - Lozier.

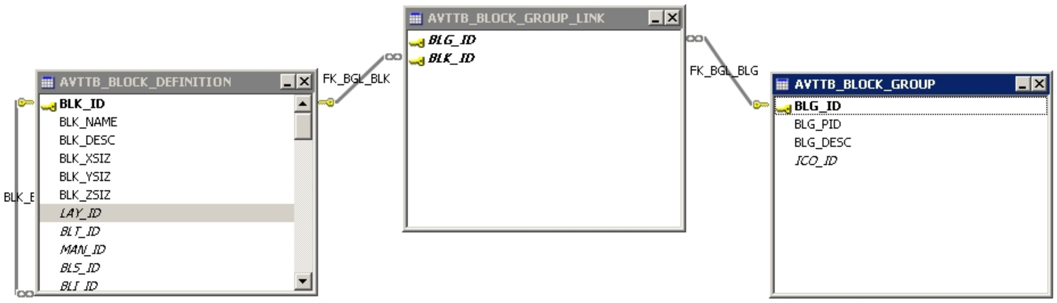
Note: If importing a hierarchy, it is strongly recommended that each node in the hierarchy has a unique name.

That is because Data Importer will use a Look-up to assign a Block Group ID to each node. (BLG_ID in the AVTTB_BLOCK_GROUP table).

If the node names are not unique (for example if we had Generic at each level rather than Fixture-Generic, Fitting-Generic and Shelf-Generic) the look up would fail because the node name is not unique.

Planning the Import from the Schema

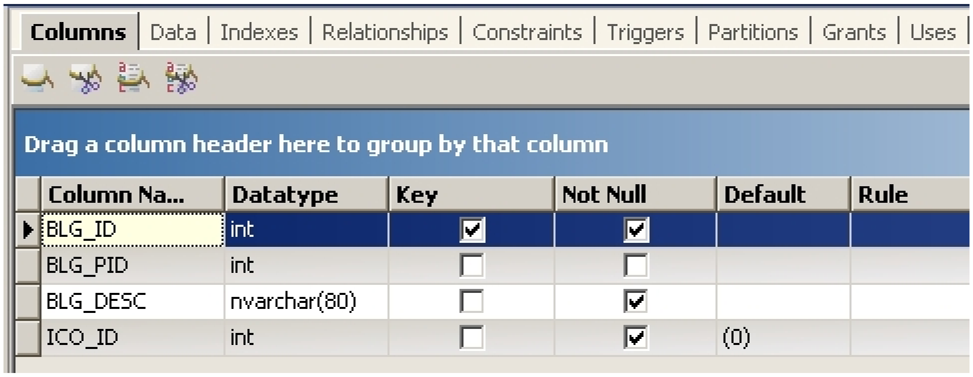
To add the **Fixture Hierarchy**, we consult the schema:



We note that we only have to populate the AVTTB_BLOCK_GROUP table.

(If we add fixtures, much of the information will go into the AVTTB_BLOCK_DEFINITION table. The AVTTB_BLOCK_GROUP_LINK table will then form the necessary line between the Block Groups and the Blocks they contain).

And the Block Group Table is as follows:



Column Na...	Datatype	Key	Not Null	Default	Rule
BLG_ID	int	☑	☑		
BLG_PID	int	☐	☐		
BLG_DESC	nvarchar(80)	☐	☑		
ICO_ID	int	☐	☑	(0)	

There are four columns in the table. We must populate three of them:

Column	Purpose	Comment
BLG_ID	Block Group ID	(Primary Key)
BLG_PID	Block Group Parent ID	Required for anything but the first level of the hierarchy (Type)
BLG_DESC	Block Group Description	The name of the Block Group

The fourth column ICO_ID (Icon ID) is non-nullable but has a default, so we will not need to import this information.

Import Type, Definition and Lookups

We will create a new import type: **Fixture Hierarchy**.

Fixture Hierarchy will have three Import Definitions in it: Hierarchy Level 1, Hierarchy Level 2 and Hierarchy Level 3.

The Import Definitions will have the following Import Fields:

Hierarchy Level 1

Data	Column	Data Type	Key	Not Null	Default
Group ID	BLG_ID	Integer	Yes	Yes	
Block Description	BLG_DESC	Nvarchar	No	Yes	

The Group ID will have to be assigned by looking up the next required value from the database.

The Block Description will be read from the Data Import spreadsheet.

The Icon (which has an available default set) can be read from the database.

> **Note**: No Group Parent will be required for this level of the hierarchy, as it is the top level.

Hierarchy Level 2

Data	Column	Data Type	Key	Not Null	Default
Group ID	BLG_ID	Integer	Yes	Yes	
Group Parent	BLG_PID	Integer	No	No	
Block Description	BLG_DESC	Nvarchar	No	Yes	

The Group ID will have to be assigned by looking up the next required value from the database.

The Group Parent will be defined as a Look-up value.

The Block Description will be read from the Data Import spreadsheet.

The Icon (which has an available default set) can be read from the database.

Hierarchy Level 3

Data	Column	Data Type	Key	Not Null	Default
Group ID	BLG_ID	Integer	Yes	Yes	
Group Parent	BLG_PID	Integer	No	No	
Block Description	BLG_DESC	Nvarchar	No	Yes	

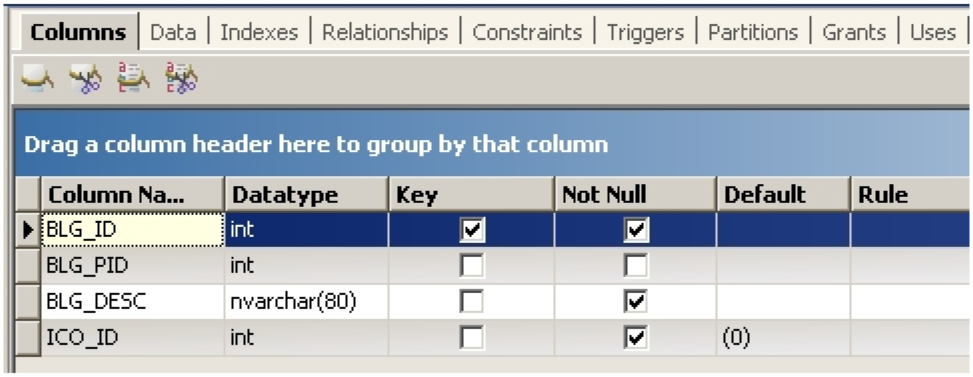
The Group ID will have to be assigned by looking up the next required value from the database.

The Group Parent will be defined as a Look-up value.

The Block Description will be read from the Data Import spreadsheet.

The Icon (which has an available default set) can be read from the database.

> **Note**: Note how these Import fields match up with the columns in the AVTTB_BLOCK_GROUP table.

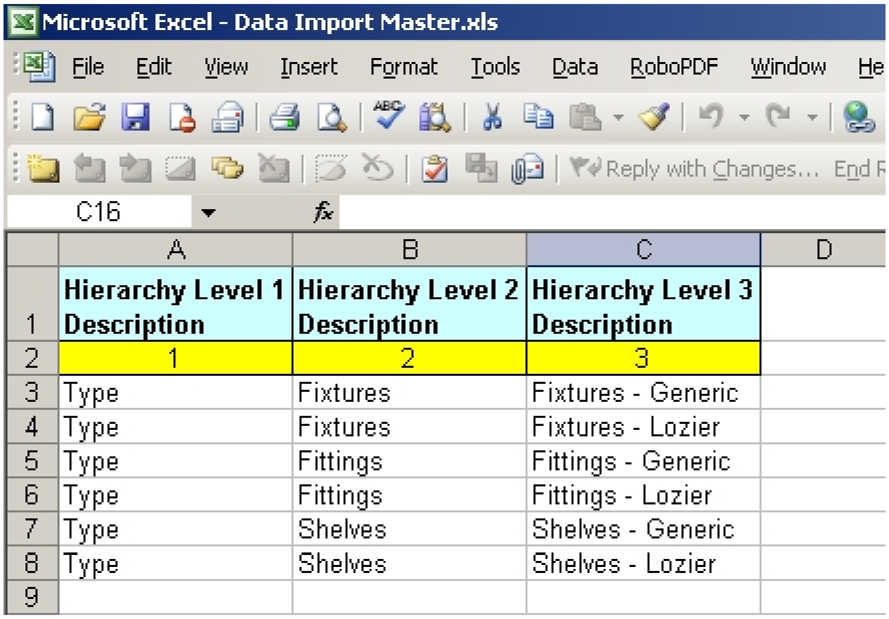


Column Na...	Datatype	Key	Not Null	Default	Rule
BLG_ID	int	☑	☑		
BLG_PID	int	□	□		
BLG_DESC	nvarchar(80)	□	☑		
ICO_ID	int	□	☑	(0)	

We will also need a look up (Block Group - PK). This will be used to convert the Block Group Description (BLG_DESC) into the Block Group ID (BLG_ID).

The Data Import Spreadsheet

Once the required data is known, the data import spreadsheet can be configured as required.



	A	B	C	D
1	Hierarchy Level 1 Description	Hierarchy Level 2 Description	Hierarchy Level 3 Description	
2	1	2	3	
3	Type	Fixtures	Fixtures - Generic	
4	Type	Fixtures	Fixtures - Lozier	
5	Type	Fittings	Fittings - Generic	
6	Type	Fittings	Fittings - Lozier	
7	Type	Shelves	Shelves - Generic	
8	Type	Shelves	Shelves - Lozier	
9				

Row 1 contains the title of the information to be held below. This will be useful when configuring the Import Definition, but will not be imported itself.

Row 2 numbers the information to be imported. This will also be useful when configuring the Import Definition, but again will not be imported.

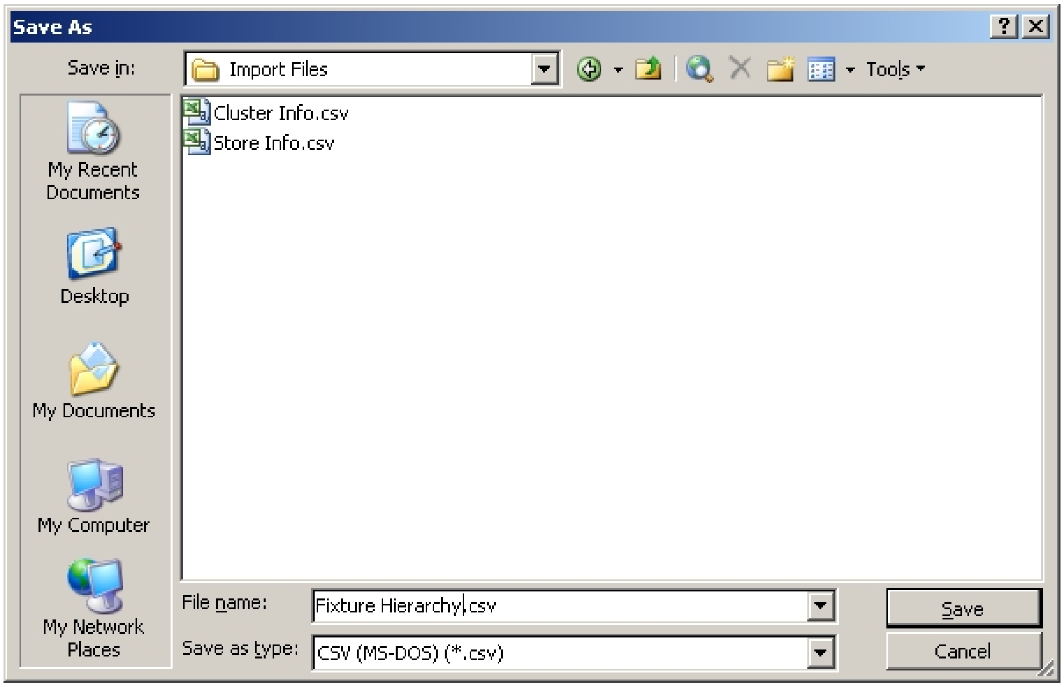
Rows 3 - 8 contain the information to be imported. Note that each row must contain a full list of every level in the hierarchy.

Output from the Spreadsheet

Once the information has been entered into the spreadsheet it must be saved into the Import Directory for Macro Space Management.

> **Note**: Data Import will only work if the import information is in this specific directory.

This is done by using the Save-As option in Excel.



Save in: Import Files

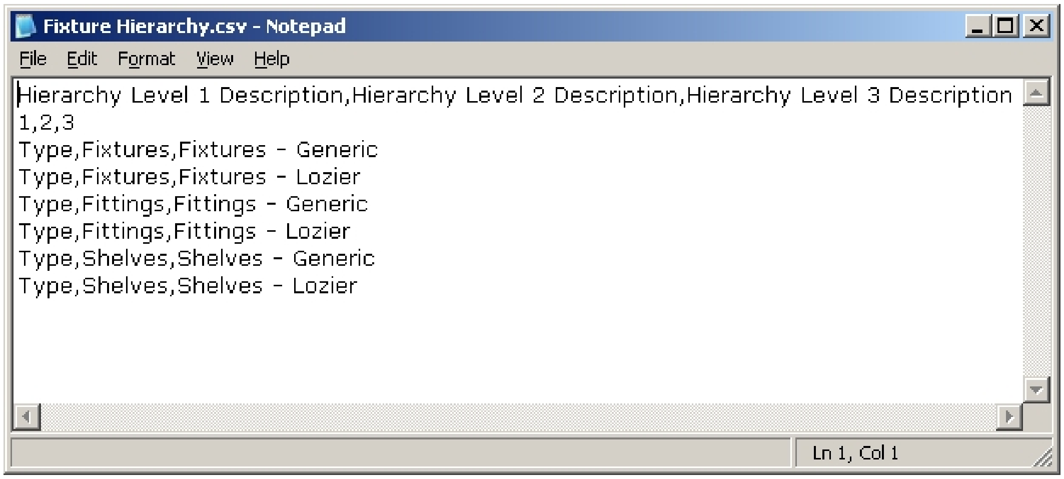
Cluster Info.csv
Store Info.csv

File name: Fixture Hierarchy.csv

Save as type: CSV (MS-DOS) (*.csv)

> **Note**: The Type must be set to CSV (or other type compatible with Data Importer)

This will result in a file with all the data for an individual Fixture Group on one line, with the data separated by commas.

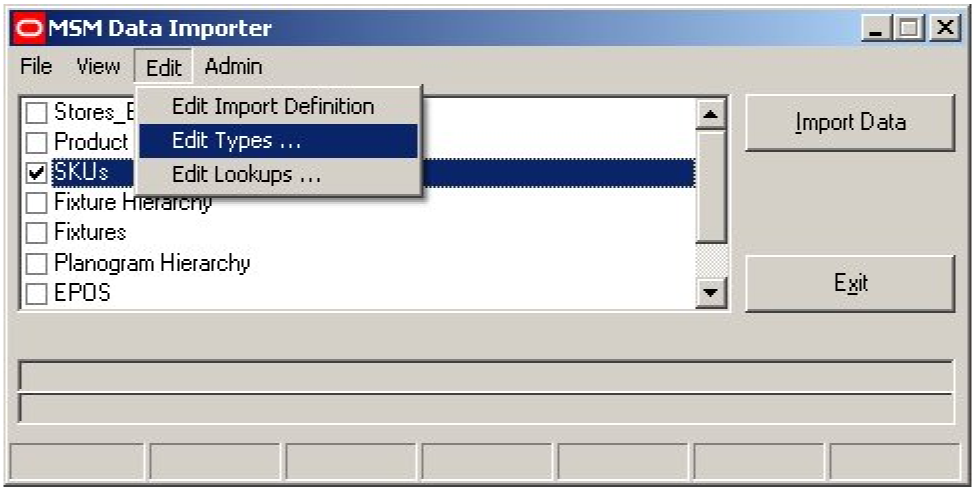


```
Hierarchy Level 1 Description,Hierarchy Level 2 Description,Hierarchy Level 3 Description
1,2,3
Type,Fixtures,Fixtures - Generic
Type,Fixtures,Fixtures - Lozier
Type,Fittings,Fittings - Generic
Type,Fittings,Fittings - Lozier
Type,Shelves,Shelves - Generic
Type,Shelves,Shelves - Lozier
```

Adding an Import Type

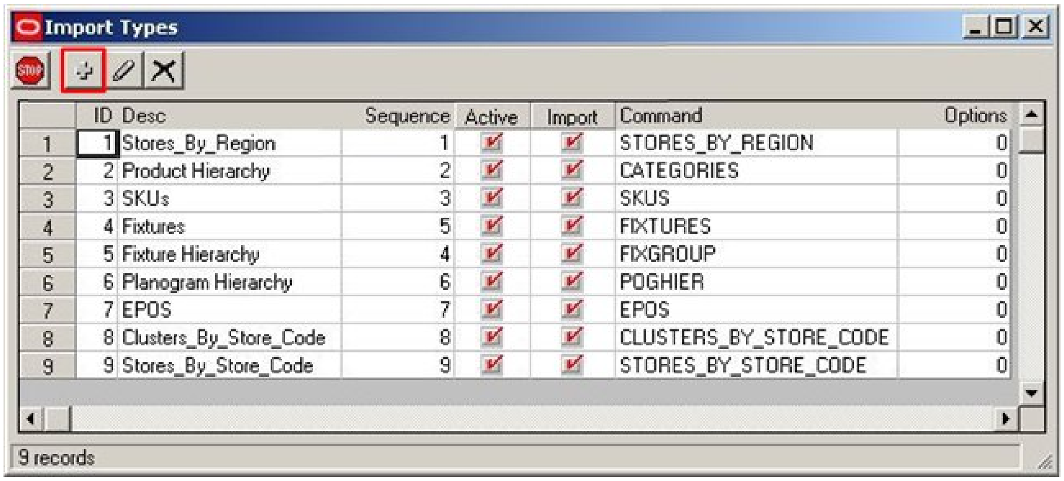
Configuring the Import Type

To **Add a new Import Type**, use <Ctrl + A> to show the Edit menu and select Edit Types



☐ Stores_B
☐ Product
☑ SKUs
☐ Fixture Hierarchy
☐ Fixtures
☐ Planogram Hierarchy
☐ EPOS

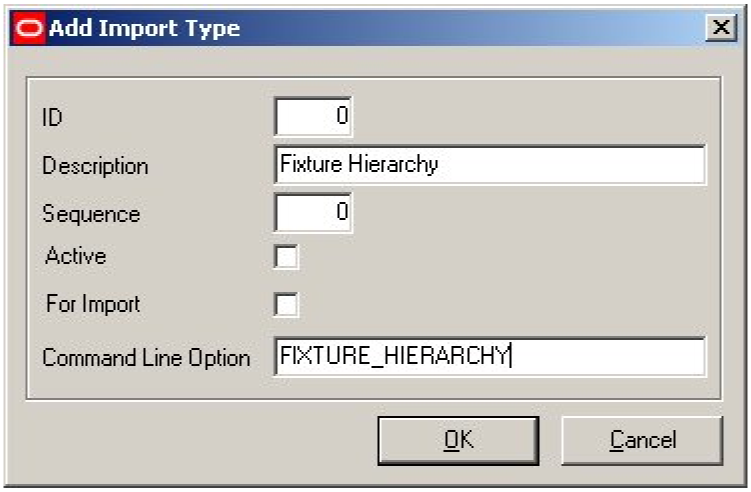
This will bring up the Import types dialogue box. Click on the **Add** icon.



	ID	Desc	Sequence	Active	Import	Command	Options
1	1	Stores_By_Region	1	✓	✓	STORES_BY_REGION	0
2	2	Product Hierarchy	2	✓	✓	CATEGORIES	0
3	3	SKUs	3	✓	✓	SKUS	0
4	4	Fixtures	5	✓	✓	FIXTURES	0
5	5	Fixture Hierarchy	4	✓	✓	FIXGROUP	0
6	6	Planogram Hierarchy	6	✓	✓	POGHIER	0
7	7	EPOS	7	✓	✓	EPOS	0
8	8	Clusters_By_Store_Code	8	✓	✓	CLUSTERS_BY_STORE_CODE	0
9	9	Stores_By_Store_Code	9	✓	✓	STORES_BY_STORE_CODE	0

9 records

This will bring up the Add Import type dialogue box.



ID: 0
Description: Fixture Hierarchy
Sequence: 0
Active: □
For Import: □
Command Line Option: FIXTURE_HIERARCHY

Fill it in as required and click on **OK**.

(The ID will be automatically changed by Macro Space Management).

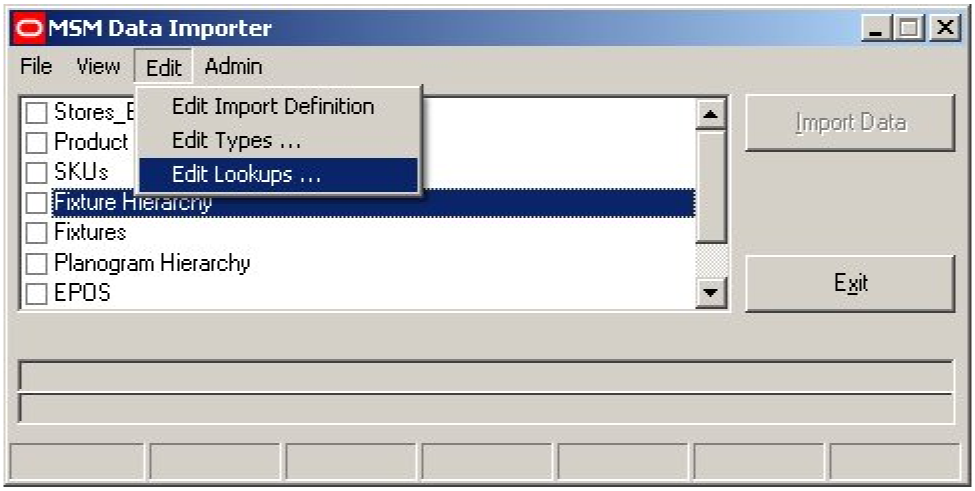
(For the Import type to be available for further editing, the Active and For Import check boxes should be ticked).

Adding a Look Up

Configuring the Look-Up

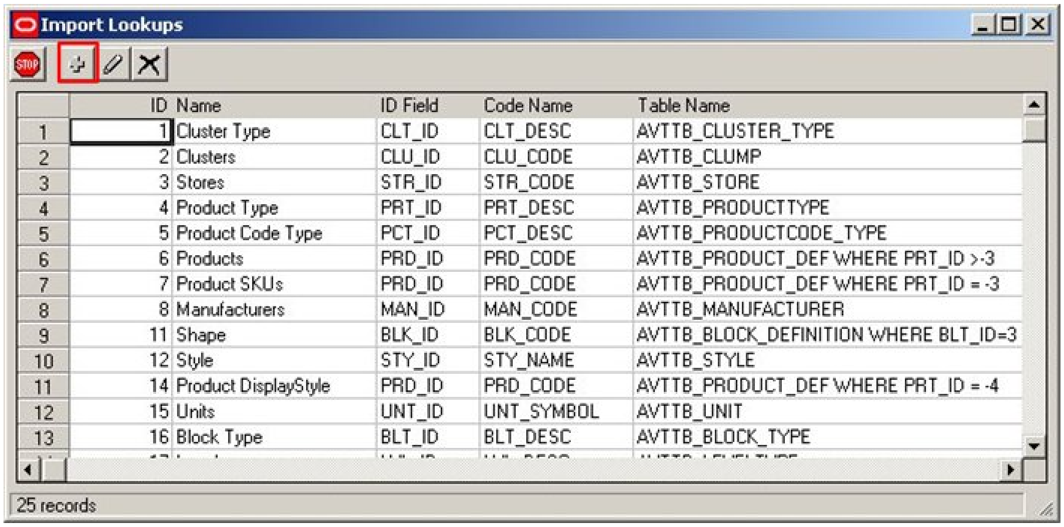
To configure a lookup, select the Edit Lookups option from the Edit pull down menu.

> **Note**: The Look-ups dialog box allows changes to the database tables using SQL statements. Accordingly, access to the dialog box is disabled by default. Please refer to the Functional Security section in the Administration Module User Guide for information on how to give access to this functionality.



□ Stores_E
□ Product
□ SKUs
□ Fixture Hierarchy
□ Fixtures
□ Planogram Hierarchy
□ EPOS

This will bring up the Import Lookups dialogue box. Click on the Add icon to add a new Lookup.

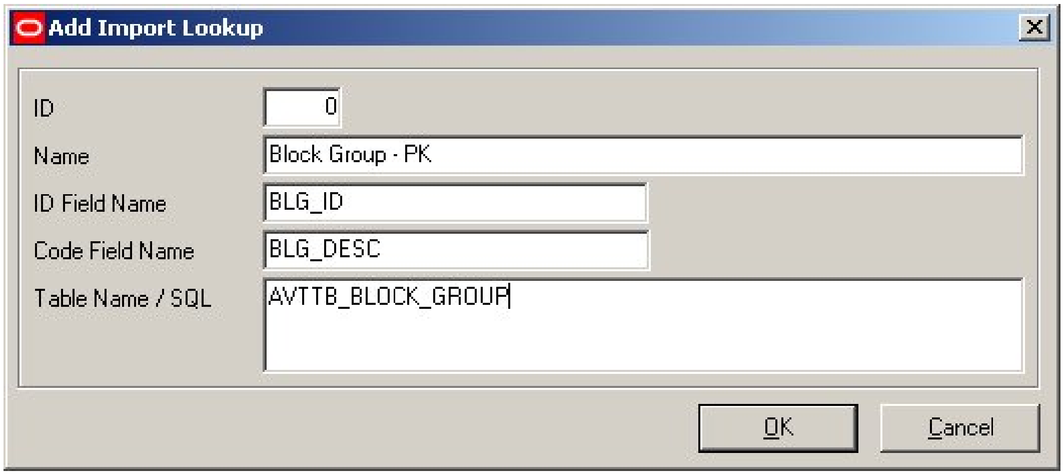


Import Lookups

	ID	Name	ID Field	Code Name	Table Name
1	1	Cluster Type	CLT_ID	CLT_DESC	AVTTB_CLUSTER_TYPE
2	2	Clusters	CLU_ID	CLU_CODE	AVTTB_CLUMP
3	3	Stores	STR_ID	STR_CODE	AVTTB_STORE
4	4	Product Type	PRT_ID	PRT_DESC	AVTTB_PRODUCTTYPE
5	5	Product Code Type	PCT_ID	PCT_DESC	AVTTB_PRODUCTCODE_TYPE
6	6	Products	PRD_ID	PRD_CODE	AVTTB_PRODUCT_DEF WHERE PRT_ID >-3
7	7	Product SKUs	PRD_ID	PRD_CODE	AVTTB_PRODUCT_DEF WHERE PRT_ID = -3
8	8	Manufacturers	MAN_ID	MAN_CODE	AVTTB_MANUFACTURER
9	11	Shape	BLK_ID	BLK_CODE	AVTTB_BLOCK_DEFINITION WHERE BLT_ID=3
10	12	Style	STY_ID	STY_NAME	AVTTB_STYLE
11	14	Product DisplayStyle	PRD_ID	PRD_CODE	AVTTB_PRODUCT_DEF WHERE PRT_ID = -4
12	15	Units	UNT_ID	UNT_SYMBOL	AVTTB_UNIT
13	16	Block Type	BLT_ID	BLT_DESC	AVTTB_BLOCK_TYPE

25 records

This will bring up the Add Import Lookup dialogue box.



Add Import Lookup

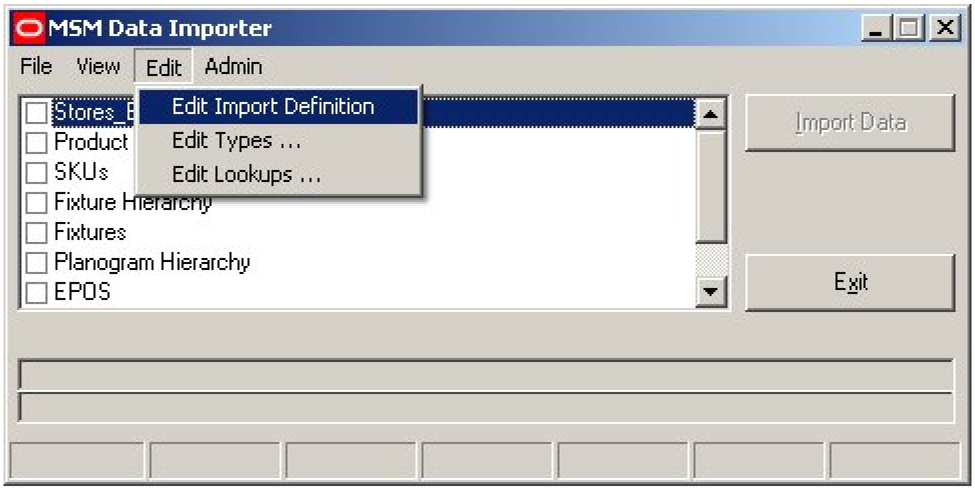
ID	0
Name	Block Group - PK
ID Field Name	BLG_ID
Code Field Name	BLG_DESC
Table Name / SQL	AVTTB_BLOCK_GROUP

This look up can be used to convert a Block Group Description (BLG_DESC) into a Block Group ID (BLG_ID).

Adding the Fixture Hierarchy Import Definitions

Opening the Dialogue

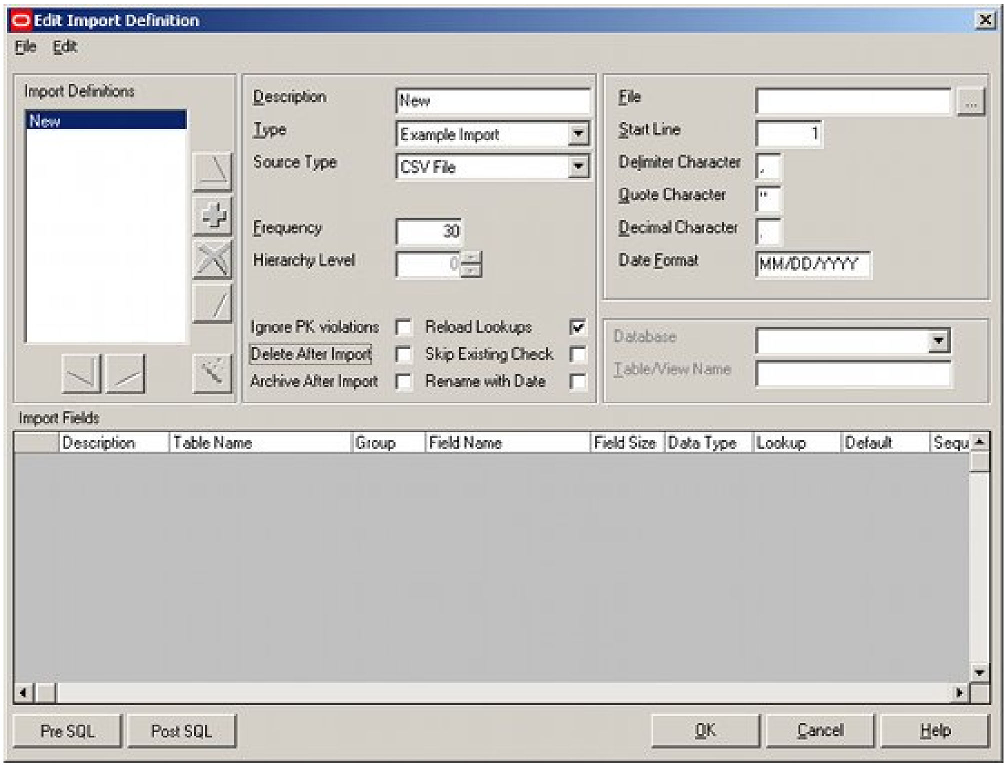
To **open the Import Dialogue option**, highlight the Import type and select Edit Import Definition from the Edit pull down menu.



□ Stores_E
□ Product
□ SKUs
□ Fixture Hierarchy
□ Fixtures
□ Planogram Hierarchy
□ EPOS

Note: The edit option is only available to Administrators. It can be toggled into view by the use of <Ctrl + a>.

This will bring up the Import Design dialogue box.



New

Description: New
Type: Example Import
Source Type: CSV File
Frequency: 30
Hierarchy Level: 0

File:
Start Line: 1
Delimiter Character: ,
Quote Character: "
Decimal Character: .
Date Format: MM/DD/YYYY

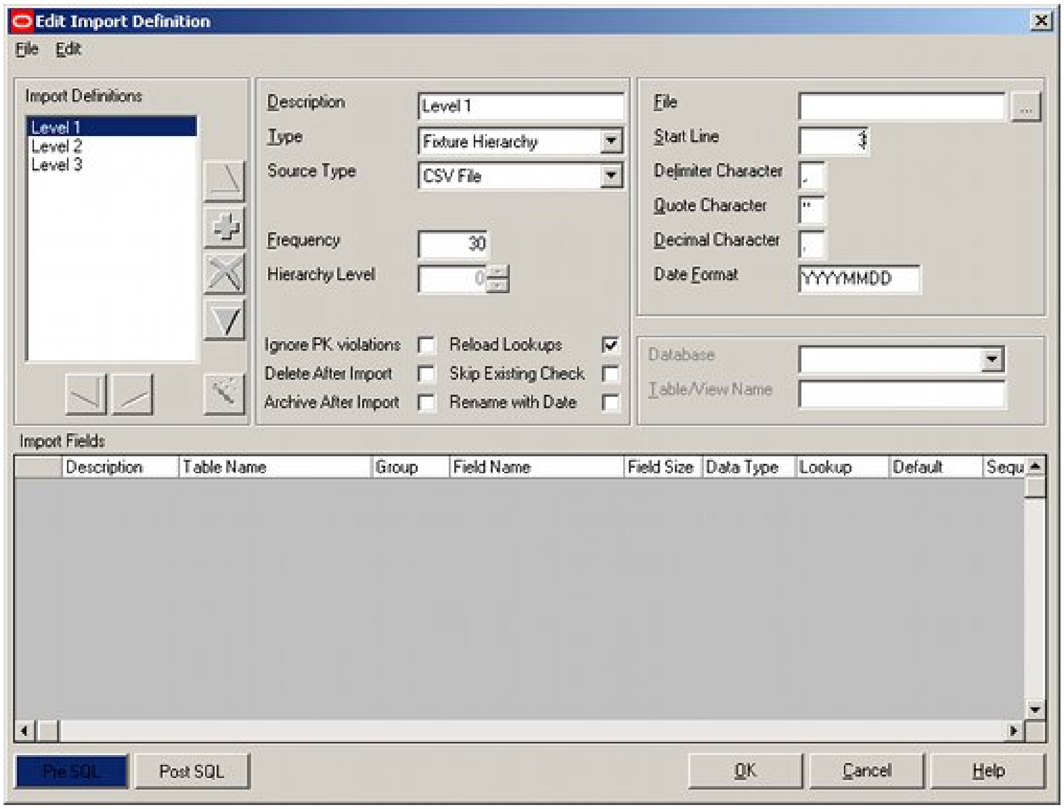
Ignore PK violations □ Reload Lookups ☑
Delete After Import □ Skip Existing Check □
Archive After Import □ Rename with Date □

Database:
Table/View Name:

Description	Table Name	Group	Field Name	Field Size	Data Type	Lookup	Default	Sequ

Setting the Levels for the Import Definitions

The **Levels for the Import Definitions** should be set first.



Import Definitions: Level 1, Level 2, Level 3

Description: Level 1
Type: Fixture Hierarchy
Source Type: CSV File
Frequency: 30
Hierarchy Level: 0

Ignore PK violations ☐ Reload Lookups ☑
Delete After Import ☐ Skip Existing Check ☐
Archive After Import ☐ Rename with Date ☐

File:
Start Line: 3
Delimiter Character: ,
Quote Character: "
Decimal Character: .
Date Format: YYYYMMDD

Database:
Table/View Name:

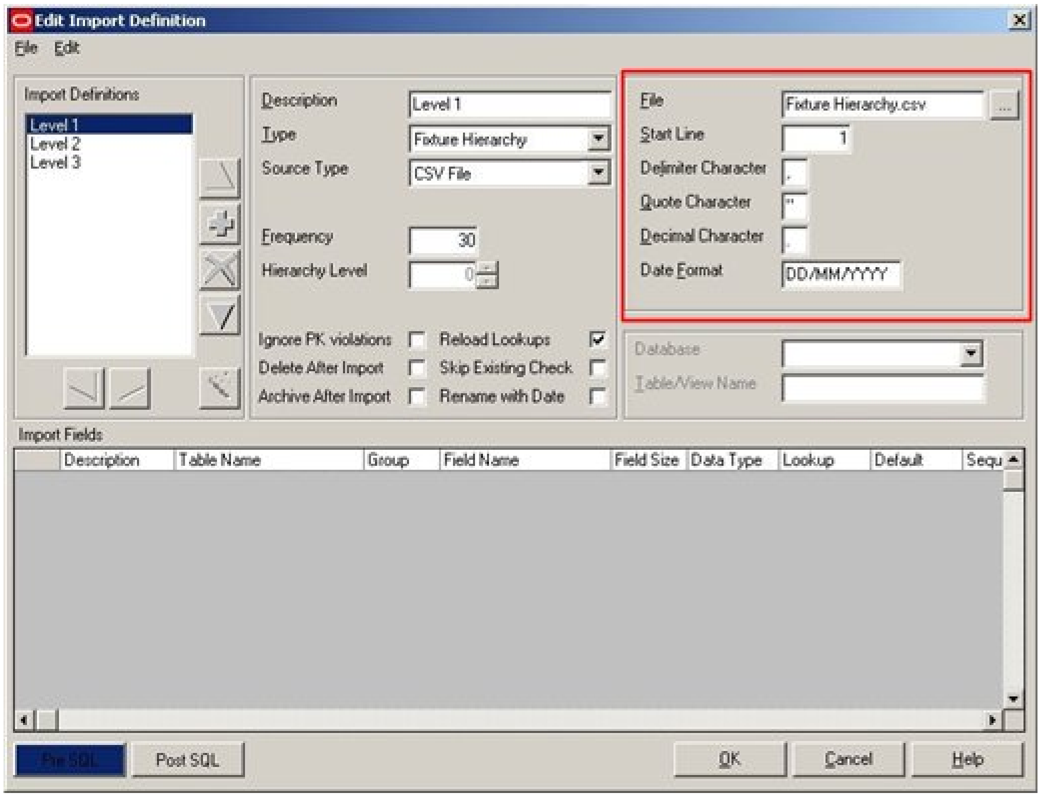
As there are three levels in the import definition, these will be called Hierarchy Level 1, Hierarchy Level 2 and Hierarchy Level 3 (as per the data to be imported).

The levels are added by clicking on the Add icon in the Import Definitions frame. The level names are changed by editing the data in the Description field in the Description Frame.

Setting the File Information

The **File information** is set in the File Information frame.

Each Import Definition will require the file to be set - this is because information could be read from several different files during the import.


Edit Import Definition
File Edit
Import Fields
Description Table Name Group Field Name Field Size Data Type Lookup Default Sequ
Pre SQL
Post SQL
OK
Cancel
Help

Import Definitions: Level 1, Level 2, Level 3

Description: Level 1
Type: Fixture Hierarchy
Source Type: CSV File
Frequency: 30
Hierarchy Level: 0

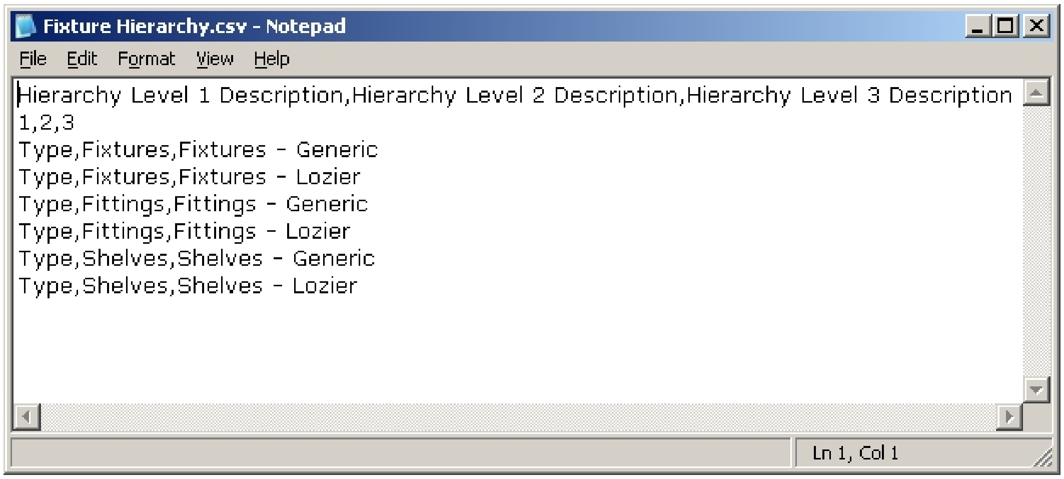
Ignore PK violations □ Reload Lookups ☑
Delete After Import □ Skip Existing Check □
Archive After Import □ Rename with Date □

File: Fixture Hierarchy.csv
Start Line: 1
Delimiter Character: ,
Quote Character: "
Decimal Character: .
Date Format: DD/MM/YYYY

Database:
Table/View Name:

File is set by clicking on the File button and selecting the Fixture Hierarchy.csv file in the Import directory.

The rest of the information has to be set pertinent to the format of the file:



```
Hierarchy Level 1 Description,Hierarchy Level 2 Description,Hierarchy Level 3 Description
1,2,3
Type,Fixtures,Fixtures - Generic
Type,Fixtures,Fixtures - Lozier
Type,Fittings,Fittings - Generic
Type,Fittings,Fittings - Lozier
Type,Shelves,Shelves - Generic
Type,Shelves,Shelves - Lozier
```

Start Line is set to 1: this will allow us to confirm the data we are selecting is correct. (This setting will be changed later to 3).

Delimiter Character is set to ',' as each item of data is separated from the next by a comma.

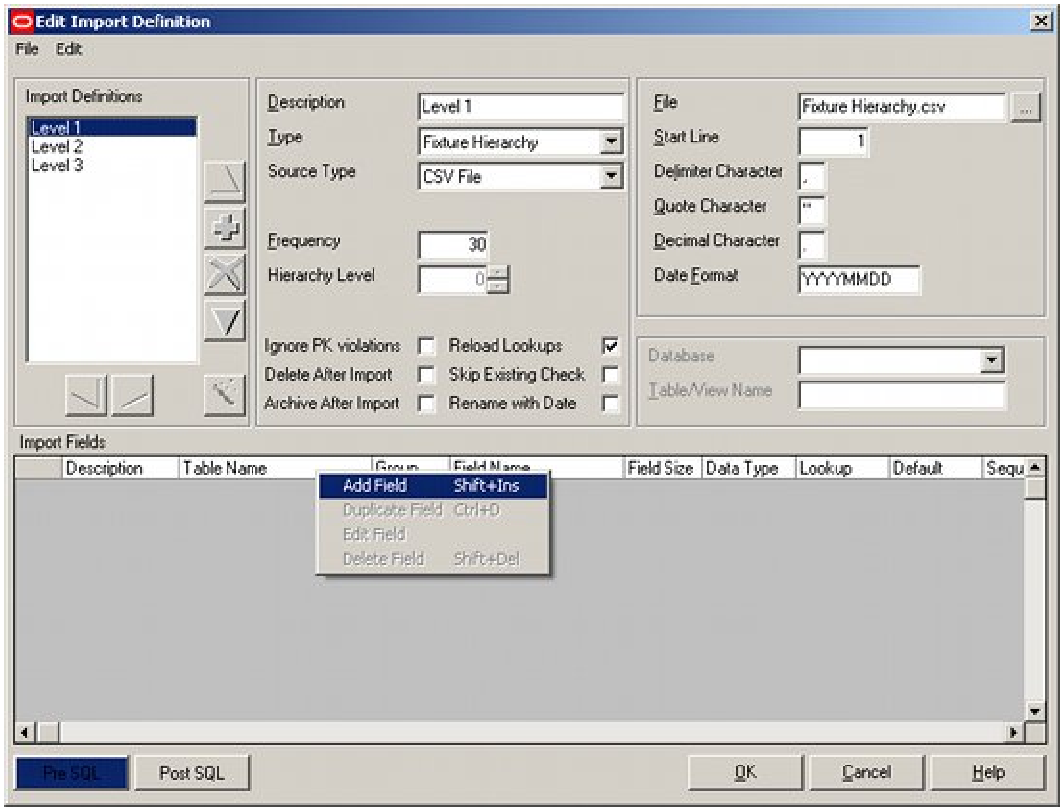
Quote Character is set to " to enable us to import text strings if required. (Not required for this definition).

Decimal Character is set to '.' as we will be using a full stop, not a comma, to indicate decimal places. (Not required for this definition).

Date Format is set to DD/MM/YYYY to indicate we will be importing dates in the form 23/12/2007. (Not required for this definition).

Bringing up the Import Field GUI

To set the **Hierarchy Level 1 import definition**, right click on the titles at the head of the Import Field frame.



Edit Import Definition

Import Definitions: Level 1, Level 2, Level 3

Description: Level 1
Type: Fixture Hierarchy
Source Type: CSV File
Frequency: 30
Hierarchy Level: 0

Ignore PK violations □ Reload Lookups ☑
Delete After Import □ Skip Existing Check □
Archive After Import □ Rename with Date □

File: Fixture Hierarchy.csv
Start Line: 1
Delimiter Character: ,
Quote Character: "
Decimal Character: .
Date Format: YYYYMMDD

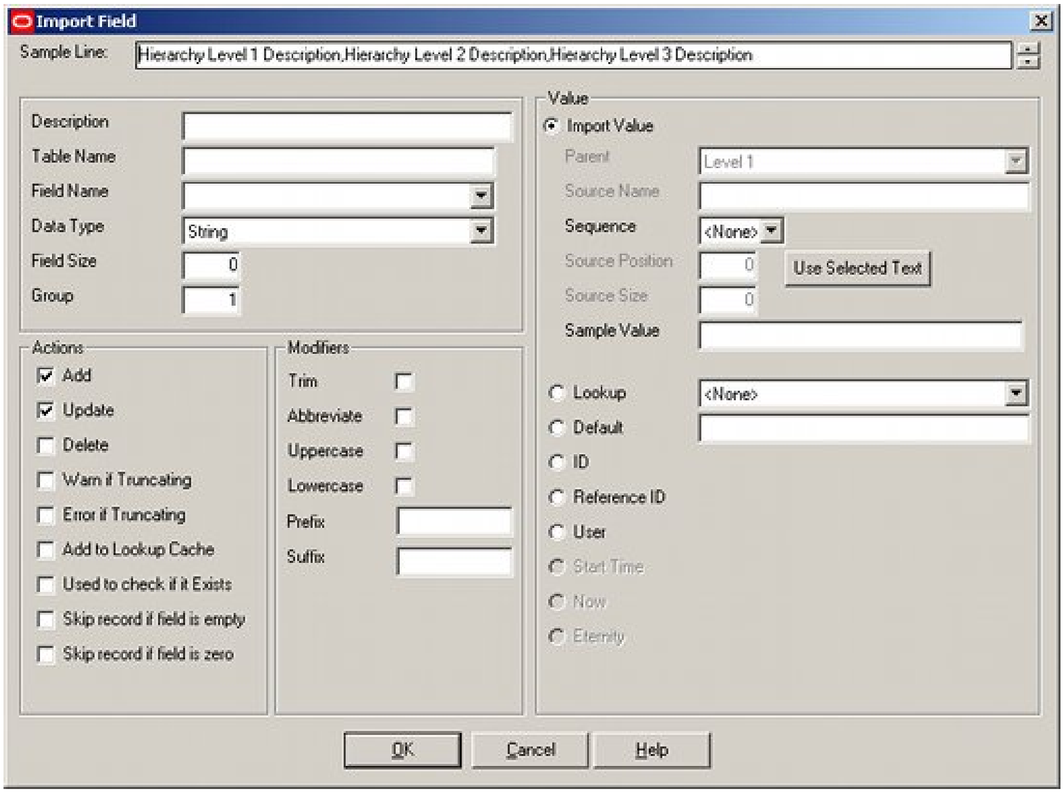
Database:
Table/View Name:

Import Fields

	Description	Table Name	Group	Field Name	Field Size	Data Type	Lookup	Default	Sequ

Right click and select Add Field.

This will bring up the Import field dialogue box which can then be configured for the required parameters.



Sample Line: Hierarchy Level 1 Description,Hierarchy Level 2 Description,Hierarchy Level 3 Description

Description:
Table Name:
Field Name:
Data Type: String
Field Size: 0
Group: 1

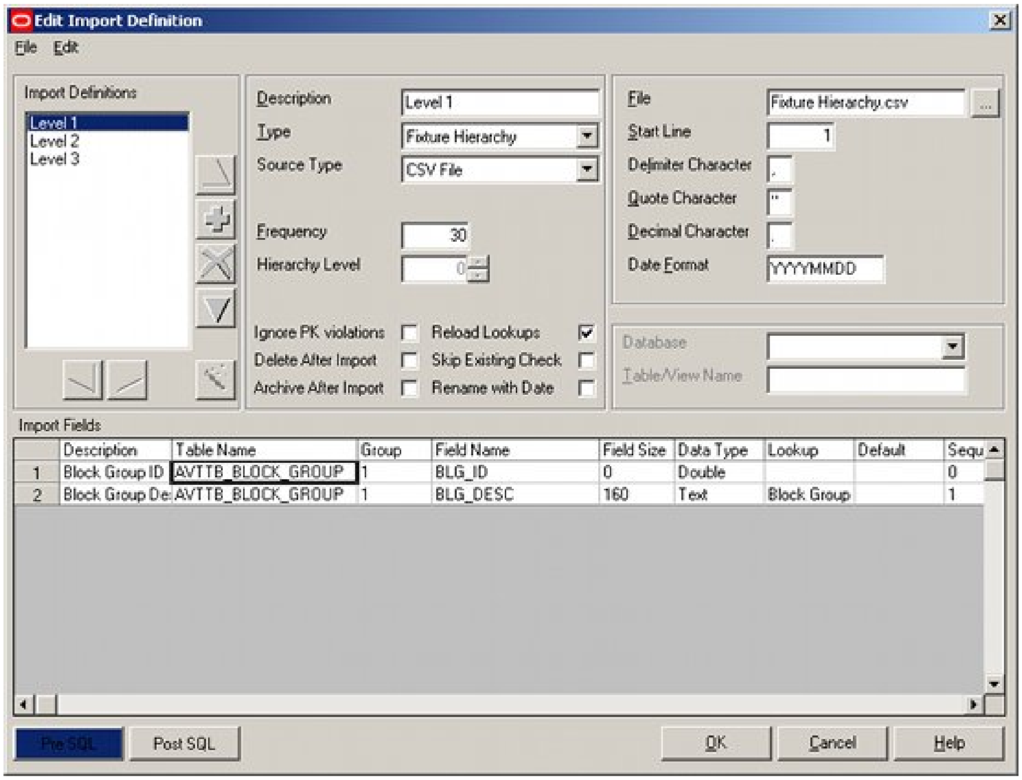
- ☑ Add
- ☑ Update
- □ Delete
- □ Warn if Truncating
- □ Error if Truncating
- □ Add to Lookup Cache
- □ Used to check if it Exists
- □ Skip record if field is empty
- □ Skip record if field is zero

Trim □
Abbreviate □
Uppercase □
Lowercase □
Prefix:
Suffix:

● Import Value
Parent: Level 1
Source Name:
Sequence: <None>
Source Position: 0
Source Size: 0
Sample Value:
○ Lookup <None>
○ Default
○ ID
○ Reference ID
○ User
○ Start Time
○ Now
○ Eternity

Hierarchy Level 1 Import Fields

There are two Import Fields to configure for Hierarchy Level 1, Block Group ID and Block Group Description.



Import Definitions: Level 1, Level 2, Level 3

Description: Level 1
Type: Fixture Hierarchy
Source Type: CSV File
Frequency: 30
Hierarchy Level: 0

Ignore PK violations □ Reload Lookups ☑
Delete After Import □ Skip Existing Check □
Archive After Import □ Rename with Date □

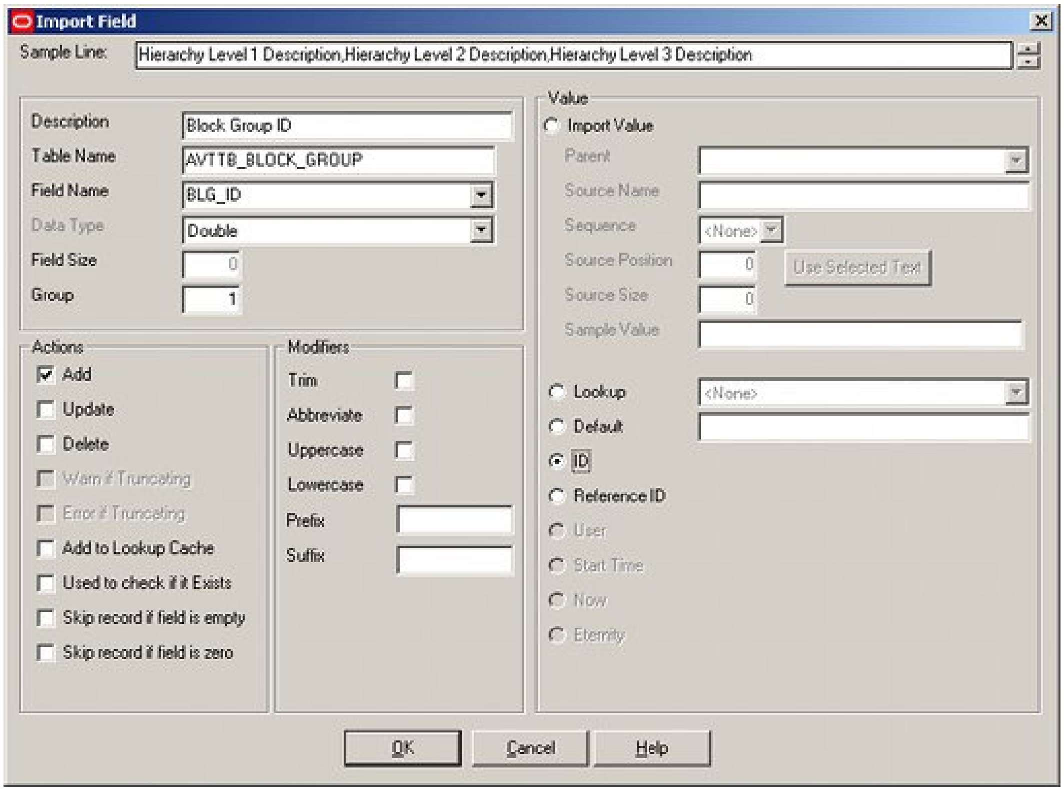
File: Fixture Hierarchy.csv
Start Line: 1
Delimiter Character: ,
Quote Character: "
Decimal Character: .
Date Format: YYYYMMDD
Database:
Table/View Name:

	Description	Table Name	Group	Field Name	Field Size	Data Type	Lookup	Default	Sequ
1	Block Group ID	AVTTB_BLOCK_GROUP	1	BLG_ID	0	Double			0
2	Block Group De	AVTTB_BLOCK_GROUP	1	BLG_DESC	160	Text	Block Group		1

This Hierarchy level will require a Pre SQL statement. This will be used to ensure the Next ID table (AVTTB_NEXTID) is up to date with the latest ID for Block Groups.

Configuring the Level 1 Information - Block Group ID

The **Level 1 Block Group** Information is configured as required:



Sample Line: Hierarchy Level 1 Description,Hierarchy Level 2 Description,Hierarchy Level 3 Description

Description	Block Group ID
Table Name	AVTTB_BLOCK_GROUP
Field Name	BLG_ID
Data Type	Double
Field Size	0
Group	1

Actions

- ☑ Add
- □ Update
- □ Delete
- □ Warn if Truncating
- □ Error if Truncating
- □ Add to Lookup Cache
- □ Used to check if it Exists
- □ Skip record if field is empty
- □ Skip record if field is zero

Modifiers

- Trim □
- Abbreviate □
- Uppercase □
- Lowercase □
- Prefix
- Suffix

Value

- ○ Import Value
  - Parent
  - Source Name
  - Sequence <None>
  - Source Position 0
  - Source Size 0
  - Sample Value
- ○ Lookup <None>
- ○ Default
- ● ID
- ○ Reference ID
- ○ User
- ○ Start Time
- ○ Now
- ○ Eternity

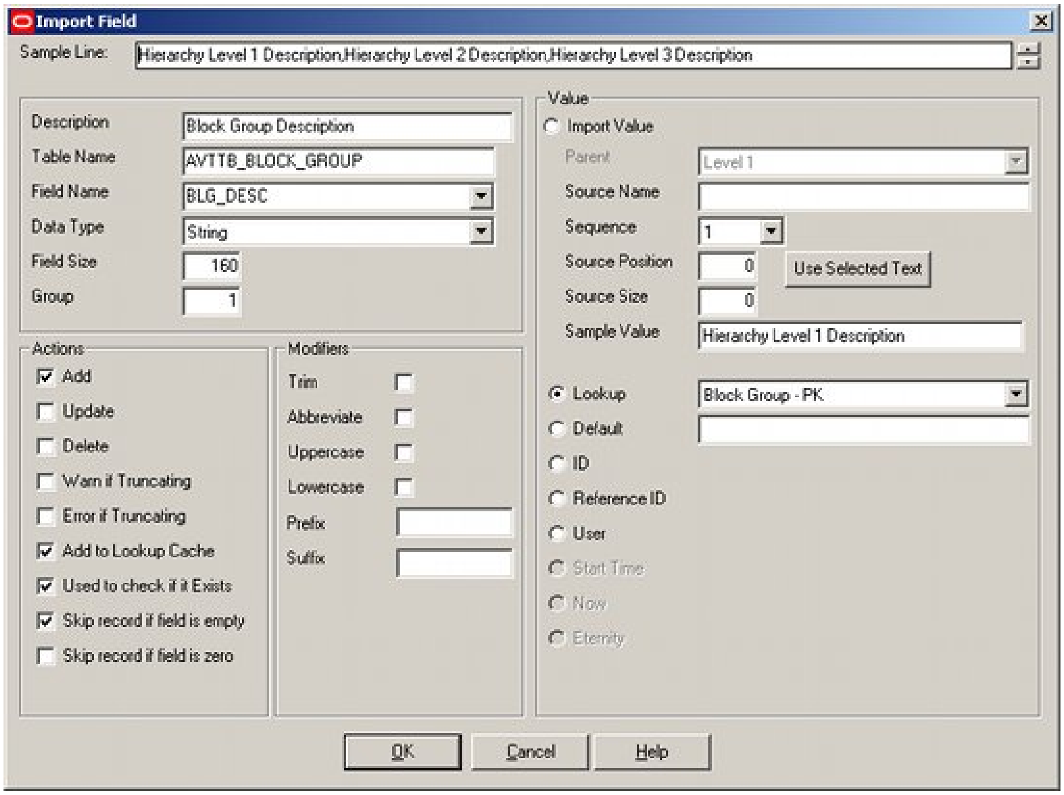
Comments

The Add option has been checked in the Actions frame as we will wish to add a Block Group ID if the Block Group does not exist.

The ID radio button in the Value Frame has been selected to indicate we need to add a numeric value from the AVTTB_NEXTID table.

Configuring the Level 1 Information - Description

The **Level 1 Description** is configured as follows:



Import Field

Sample Line: Hierarchy Level 1 Description,Hierarchy Level 2 Description,Hierarchy Level 3 Description

Description	Block Group Description
Table Name	AVTTB_BLOCK_GROUP
Field Name	BLG_DESC
Data Type	String
Field Size	160
Group	1

Actions

- ☑ Add
- ☐ Update
- ☐ Delete
- ☐ Warn if Truncating
- ☐ Error if Truncating
- ☑ Add to Lookup Cache
- ☑ Used to check if it Exists
- ☑ Skip record if field is empty
- ☐ Skip record if field is zero

Modifiers

- Trim ☐
- Abbreviate ☐
- Uppercase ☐
- Lowercase ☐
- Prefix
- Suffix

Value

○ Import Value

Parent	Level 1
Source Name	
Sequence	1
Source Position	0
Source Size	0
Sample Value	Hierarchy Level 1 Description

● Lookup: Block Group - PK

○ Default

○ ID

○ Reference ID

○ User

○ Start Time

○ Now

○ Eternity

Comments

Add has been checked because we wish to add the block description to the AVTTB_BLOCK_GROUP_LINK table.

Add to Lookup Cache has been selected because we are using a lookup and this will speed up performance.

Used to check if it Exists has been checked because we want to ensure that the description does not already exist in the database. This will allow us to avoid duplicate entries.

Skip record if field is empty has been checked to allow a record to be skipped if it contains a blank description.

Abbreviate has been chosen to so that any data being imported over 80 characters in length are truncated to that length.

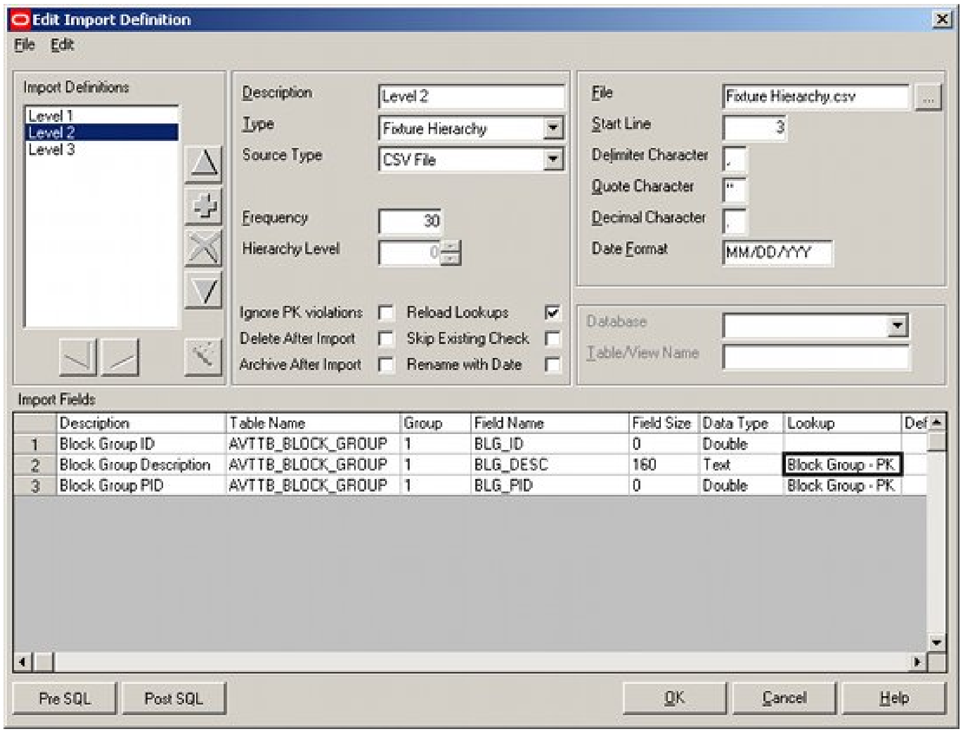
Sequence has been set to 1, to enable data to be selected from the right section of the .csv file.

We are using the Lookup 'Block Groups' to convert the Description into a Block Group ID. This will be used to check if the description already exists.

Note: As we have selected Sequence 1 from the .csv file, the Sample value is showing what is in that position in the file - in this instance Hierarchy Level 1.

Hierarchy Level 2 Import Fields

There is **three Import Fields** to configure for Hierarchy Level 2, Block Group ID, Block Group Description and Block Group Parent ID.



Import Definitions: Level 1, Level 2 (selected), Level 3

Description: Level 2
Type: Fixture Hierarchy
Source Type: CSV File
Frequency: 30
Hierarchy Level: 0

File: Fixture Hierarchy.csv
Start Line: 3
Delimiter Character: ,
Quote Character: "
Decimal Character: .
Date Format: MM/DD/YYY

Database:
Table/View Name:

☐ Ignore PK violations
☐ Delete After Import
☐ Archive After Import
☑ Reload Lookups
☐ Skip Existing Check
☐ Rename with Date

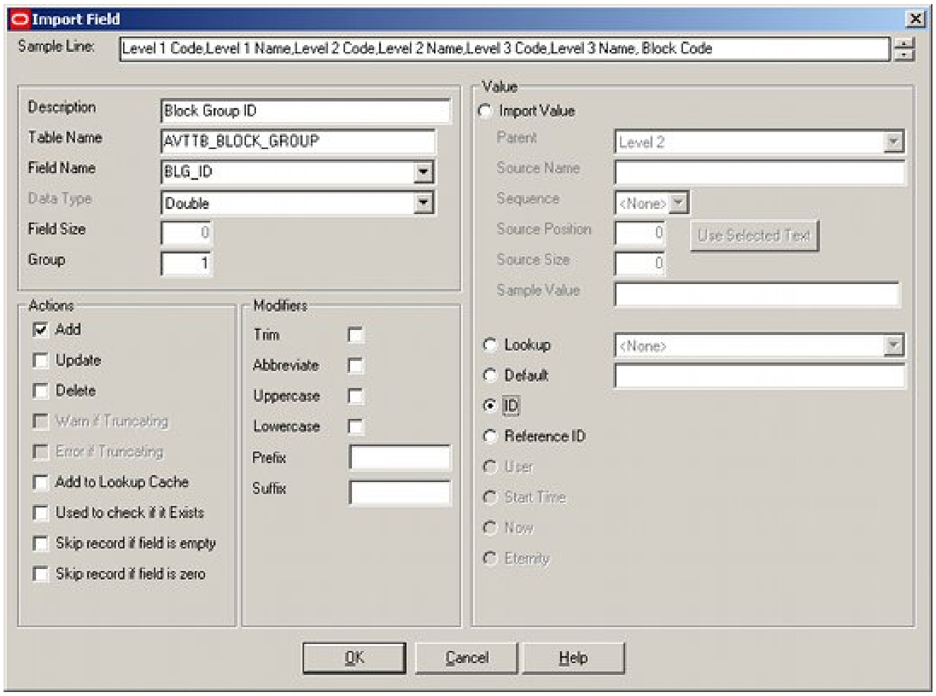
Import Fields

	Description	Table Name	Group	Field Name	Field Size	Data Type	Lookup	Def
1	Block Group ID	AVTTB_BLOCK_GROUP	1	BLG_ID	0	Double		
2	Block Group Description	AVTTB_BLOCK_GROUP	1	BLG_DESC	160	Text	Block Group - PK	
3	Block Group PID	AVTTB_BLOCK_GROUP	1	BLG_PID	0	Double	Block Group - PK	

This Hierarchy level will not require any Pre or Post SQL.

Configuring the Level 2 Information - Block Group ID

The **Level 2 Block Group Information** is configured as required:



Sample Line: Level 1 Code,Level 1 Name,Level 2 Code,Level 2 Name,Level 3 Code,Level 3 Name, Block Code

Description: Block Group ID
Table Name: AVTTB_BLOCK_GROUP
Field Name: BLG_ID
Data Type: Double
Field Size: 0
Group: 1

Actions
☑ Add
☐ Update
☐ Delete
☐ Warn if Truncating
☐ Error if Truncating
☐ Add to Lookup Cache
☐ Used to check if it Exists
☐ Skip record if field is empty
☐ Skip record if field is zero

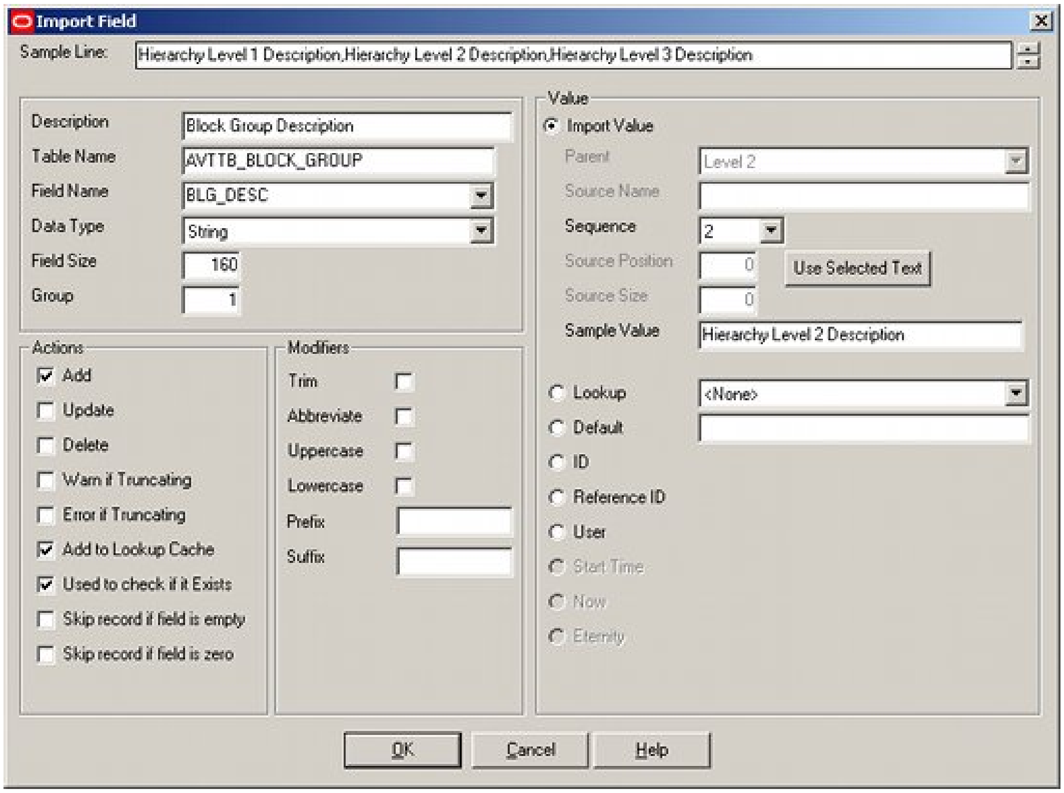
Modifiers
Trim ☐
Abbreviate ☐
Uppercase ☐
Lowercase ☐
Prefix:
Suffix:

Value
○ Import Value
Parent: Level 2
Source Name:
Sequence: <None>
Source Position: 0
Source Size: 0
Sample Value:
○ Lookup <None>
○ Default
● ID
○ Reference ID
○ User
○ Start Time
○ Now
○ Eternity

It is essentially a duplicate of the Level 1 Block Group ID import.

Configuring the Level 2 Information - Description

The **Level 2 Description** is configured as follows:



Sample Line: Hierarchy Level 1 Description,Hierarchy Level 2 Description,Hierarchy Level 3 Description

Description	Block Group Description
Table Name	AVTTB_BLOCK_GROUP
Field Name	BLG_DESC
Data Type	String
Field Size	160
Group	1

Actions

- ☑ Add
- □ Update
- □ Delete
- □ Warn if Truncating
- □ Error if Truncating
- ☑ Add to Lookup Cache
- ☑ Used to check if it Exists
- □ Skip record if field is empty
- □ Skip record if field is zero

Modifiers

- Trim □
- Abbreviate □
- Uppercase □
- Lowercase □
- Prefix
- Suffix

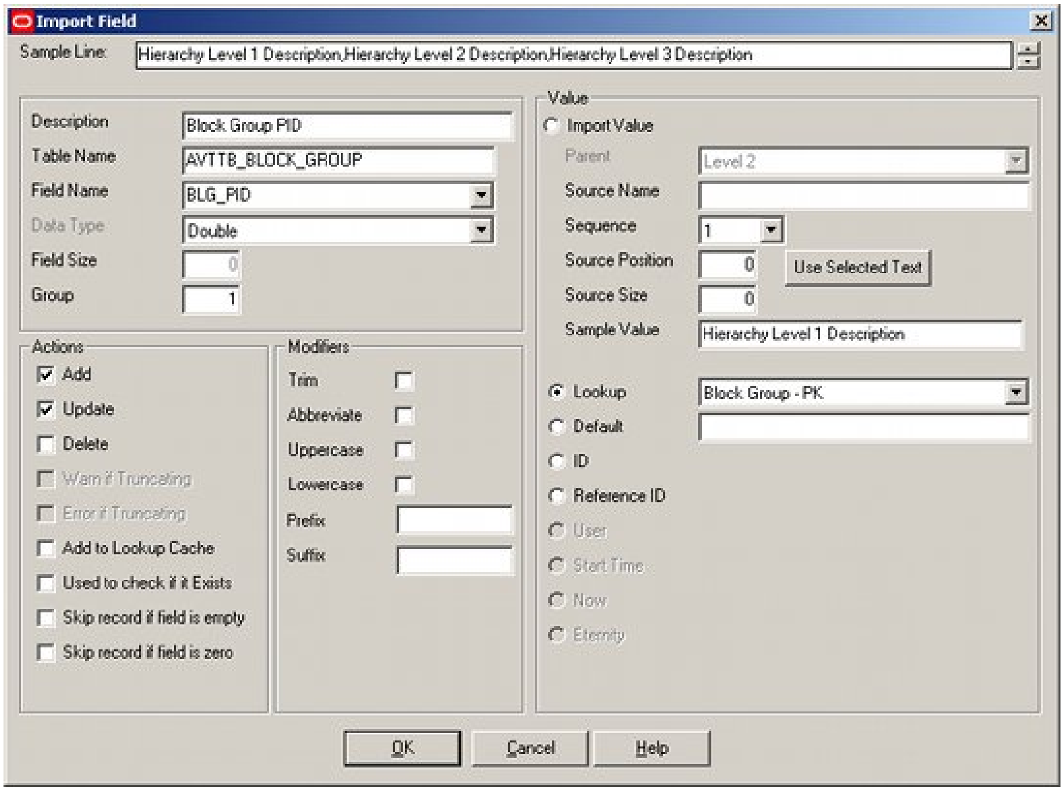
Value

- ● Import Value
  - Parent: Level 2
  - Source Name:
  - Sequence: 2
  - Source Position: 0
  - Source Size: 0
  - Use Selected Text
  - Sample Value: Hierarchy Level 2 Description
- ○ Lookup: <None>
- ○ Default
- ○ ID
- ○ Reference ID
- ○ User
- ○ Start Time
- ○ Now
- ○ Eternity

It is essentially a duplicate of the Level 1 Description, except the Sequence has been changed to 2 to read information from a different part of the .csv file.

Configuring the Level 2 Information - Parent Group

Parent Group is a new import field for Level 2 - it was not required for Level 1.



Import Field

Sample Line: Hierarchy Level 1 Description,Hierarchy Level 2 Description,Hierarchy Level 3 Description

Description	Block Group PID
Table Name	AVTTB_BLOCK_GROUP
Field Name	BLG_PID
Data Type	Double
Field Size	0
Group	1

Actions

- ☑ Add
- ☑ Update
- □ Delete
- □ Warn if Truncating
- □ Error if Truncating
- □ Add to Lookup Cache
- □ Used to check if it Exists
- □ Skip record if field is empty
- □ Skip record if field is zero

Modifiers

Trim	□
Abbreviate	□
Uppercase	□
Lowercase	□
Prefix	
Suffix	

Value

○ Import Value

Parent	Level 2
Source Name	
Sequence	1
Source Position	0
Source Size	0
Sample Value	Hierarchy Level 1 Description

● Lookup	Block Group - PK
○ Default	
○ ID	
○ Reference ID	
○ User	
○ Start Time	
○ Now	
○ Eternity	

Comments

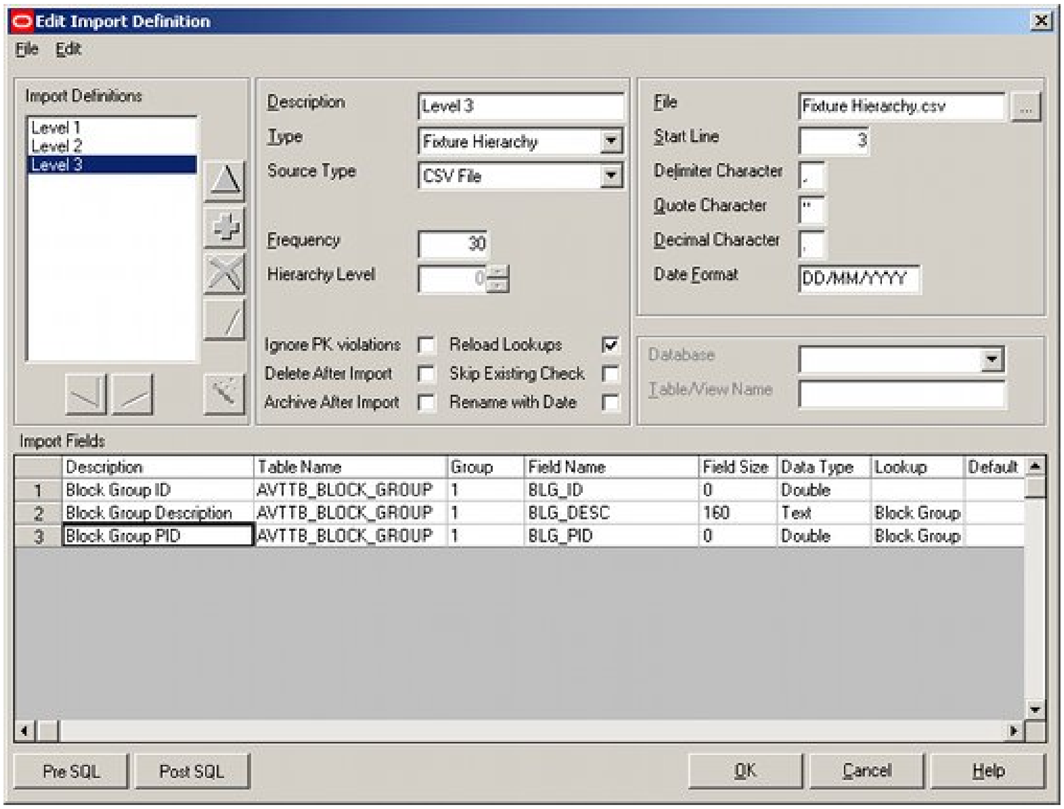
Add and Update have been checked because we wish to add the description if new, or update it if the description already exists.

Sequence has been set to 1 to pick up the information from the correct part of the CSV file.

Lookup is being used to convert the parent group name to a Block Group ID (parent ID) that will be inserted into the database.

Hierarchy Level 3 Import Fields

There is **three Import Fields** to configure for Hierarchy Level 2, Block Group ID, Block Group Description and Block Group Parent ID.



Import Definitions
- Level 1
- Level 2
- Level 3

Description	Level 3
Type	Fixture Hierarchy
Source Type	CSV File
Frequency	30
Hierarchy Level	0

Ignore PK violations	□	Reload Lookups	☑
Delete After Import	□	Skip Existing Check	□
Archive After Import	□	Rename with Date	□

File	Fixture Hierarchy.csv
Start Line	3
Delimiter Character	,
Quote Character	"
Decimal Character	.
Date Format	DD/MM/YYYY
Database	
Table/View Name	

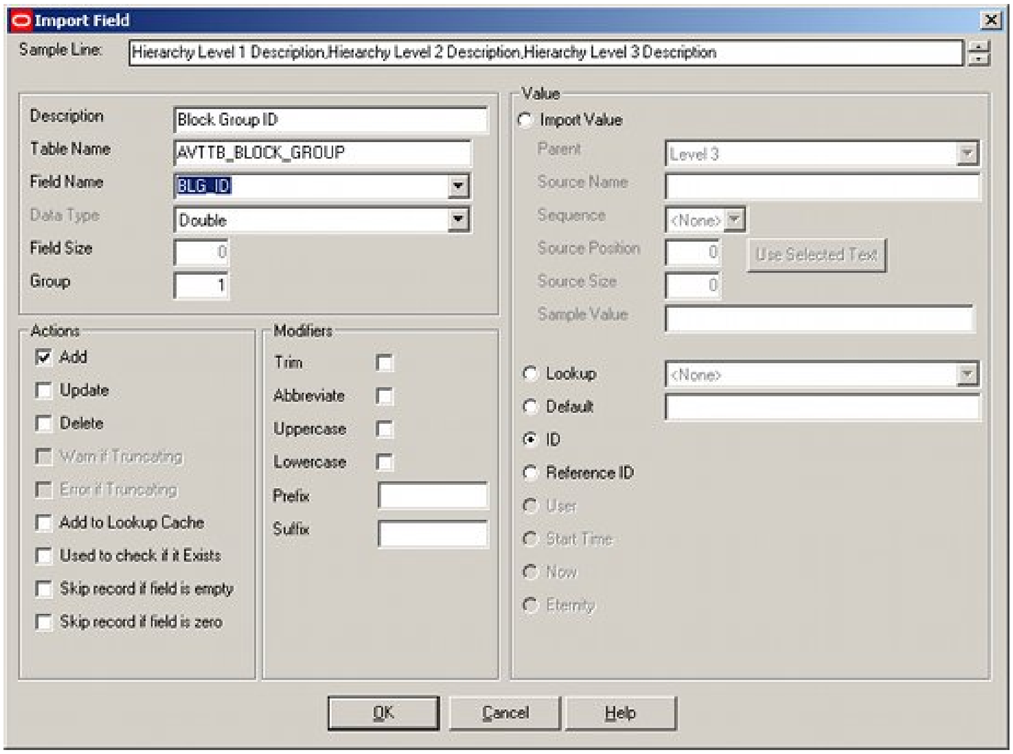
Import Fields

	Description	Table Name	Group	Field Name	Field Size	Data Type	Lookup	Default
1	Block Group ID	AVTTB_BLOCK_GROUP	1	BLG_ID	0	Double		
2	Block Group Description	AVTTB_BLOCK_GROUP	1	BLG_DESC	160	Text	Block Group	
3	Block Group PID	AVTTB_BLOCK_GROUP	1	BLG_PID	0	Double	Block Group	

This Hierarchy level will not require any Pre or Post SQL.

Configuring the Level 3 Information - Block Group ID

The **Level 3 Block Group Information** is configured as required:



Sample Line: Hierarchy Level 1 Description,Hierarchy Level 2 Description,Hierarchy Level 3 Description

Description	Block Group ID
Table Name	AVTTB_BLOCK_GROUP
Field Name	BLG_ID
Data Type	Double
Field Size	0
Group	1

Actions
- ☑ Add
- □ Update
- □ Delete
- □ Warn if Truncating
- □ Error if Truncating
- □ Add to Lookup Cache
- □ Used to check if it Exists
- □ Skip record if field is empty
- □ Skip record if field is zero

Modifiers

Trim	□
Abbreviate	□
Uppercase	□
Lowercase	□
Prefix	
Suffix	

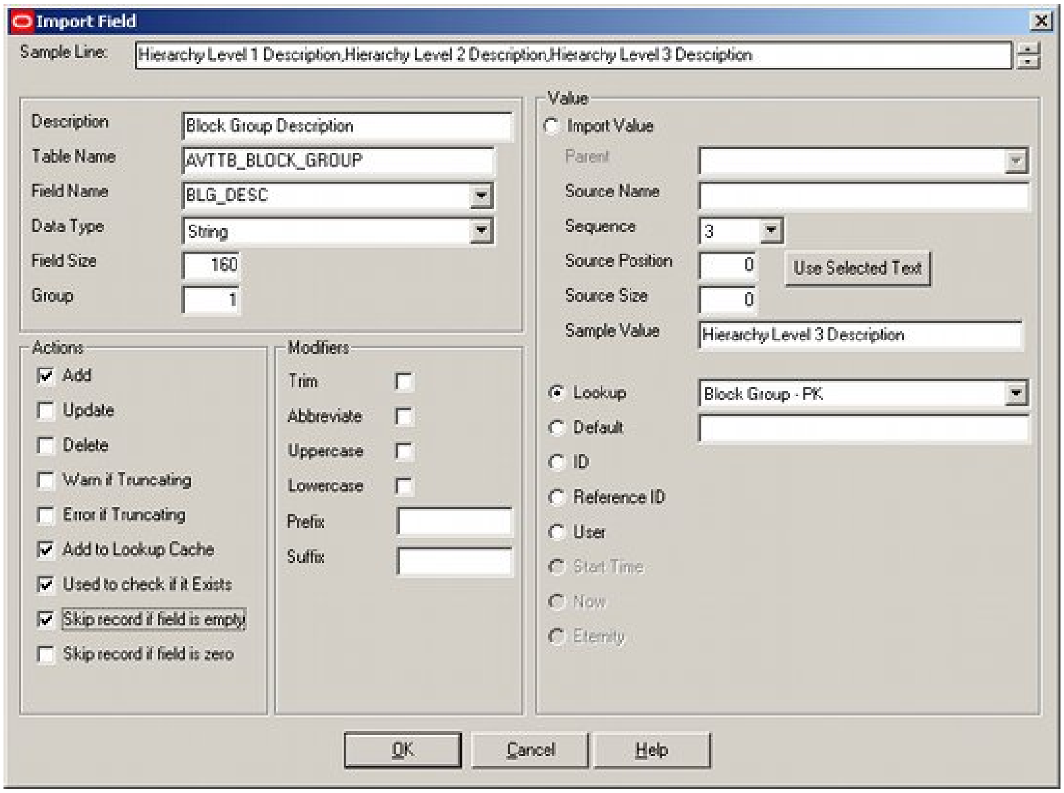
Value

○ Import Value	
Parent	Level 3
Source Name	
Sequence	<None>
Source Position	0
Source Size	0
Sample Value	
○ Lookup	<None>
○ Default	
● ID	
○ Reference ID	
○ User	
○ Start Time	
○ Now	
○ Eternity	

It is essentially a duplicate of the Level 2 Block Group ID import.

Configuring the Level 3 Information - Description

The **Level 3 Description** is configured as follows:



Import Field

Sample Line: Hierarchy Level 1 Description,Hierarchy Level 2 Description,Hierarchy Level 3 Description

Description: Block Group Description
Table Name: AVTTB_BLOCK_GROUP
Field Name: BLG_DESC
Data Type: String
Field Size: 160
Group: 1

Actions
☑ Add
□ Update
□ Delete
□ Warn if Truncating
□ Error if Truncating
☑ Add to Lookup Cache
☑ Used to check if it Exists
☑ Skip record if field is empty
□ Skip record if field is zero

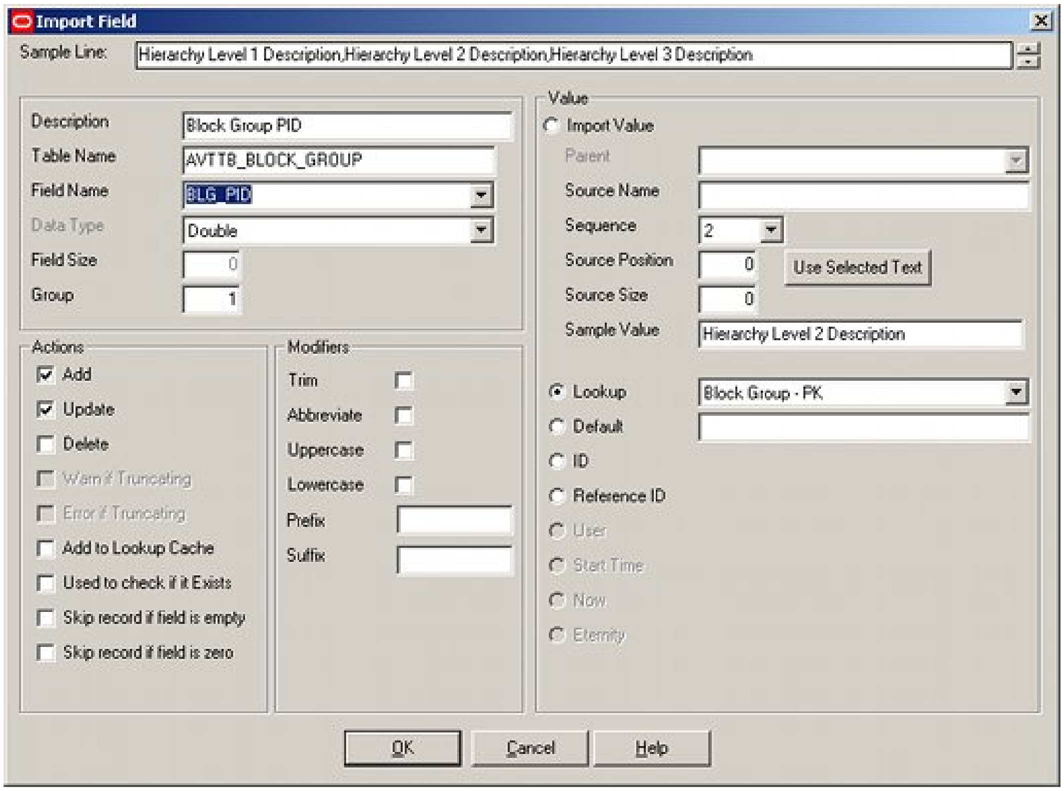
Modifiers
Trim □
Abbreviate □
Uppercase □
Lowercase □
Prefix
Suffix

Value
○ Import Value
Parent
Source Name
Sequence: 3
Source Position: 0
Source Size: 0
Sample Value: Hierarchy Level 3 Description
● Lookup: Block Group - PK
○ Default
○ ID
○ Reference ID
○ User
○ Start Time
○ Now
○ Eternity

It is essentially a duplicate of the Level 1 and 2 Descriptions, except the Sequence has been changed to 3 to read information from a different part of the .csv file.

Configuring the Level 3 Information - Parent Group

The **Level 3 Parent Group Information** is configured as required:



Sample Line: Hierarchy Level 1 Description,Hierarchy Level 2 Description,Hierarchy Level 3 Description

Description: Block Group PID
Table Name: AVTTB_BLOCK_GROUP
Field Name: BLG_PID
Data Type: Double
Field Size: 0
Group: 1

Actions
- ☑ Add
- ☑ Update
- □ Delete
- □ Warn if Truncating
- □ Error if Truncating
- □ Add to Lookup Cache
- □ Used to check if it Exists
- □ Skip record if field is empty
- □ Skip record if field is zero

Modifiers
- Trim □
- Abbreviate □
- Uppercase □
- Lowercase □
- Prefix
- Suffix

Value
- ○ Import Value
- Parent
- Source Name
- Sequence: 2
- Source Position: 0
- Source Size: 0
- Sample Value: Hierarchy Level 2 Description
- ● Lookup: Block Group - PK
- ○ Default
- ○ ID
- ○ Reference ID
- ○ User
- ○ Start Time
- ○ Now
- ○ Eternity

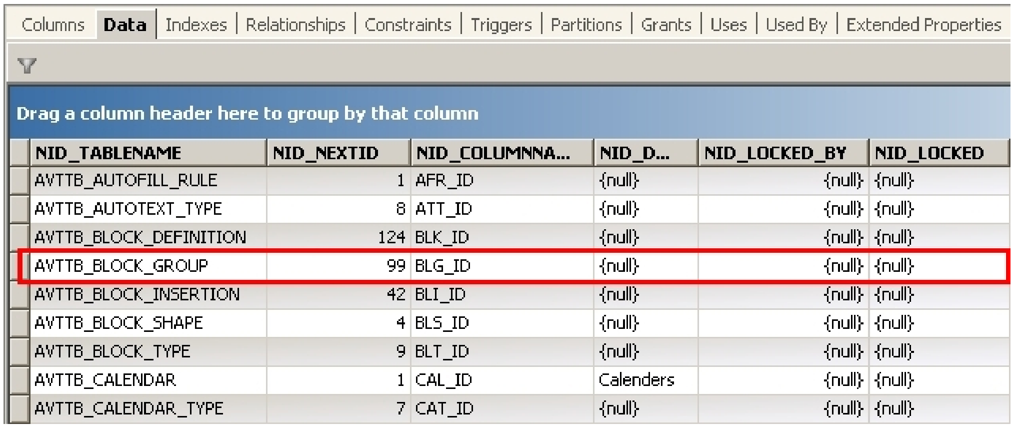
It is essentially a duplicate of the Level 2 Parent Group, except the Sequence has been changed to 2 to read information from a different part of the .csv file.

Adding the SQL Statements

Only one **SQL statement** is required for this definition.

> **Note**: Pre and Post SQL allow the Data Import module to execute changes to the database tables using SQL statements. Accordingly, Pre and Post SQL functionality is disabled by default. Please refer to the Functional Security section in the Administration Module User Guide for information on how to give access to this functionality.

It is to ensure that the value for the Block Group ID (BLK_ID in the AVTTB_BLOCK_GROUP table) is correct.

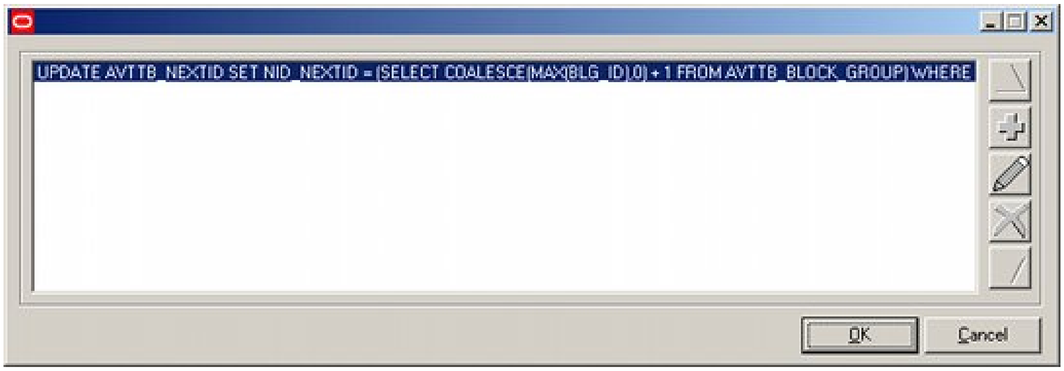


NID_TABLENAME	NID_NEXTID	NID_COLUMNNA...	NID_D...	NID_LOCKED_BY	NID_LOCKED
AVTTB_AUTOFILL_RULE	1	AFR_ID	{null}	{null}	{null}
AVTTB_AUTOTEXT_TYPE	8	ATT_ID	{null}	{null}	{null}
AVTTB_BLOCK_DEFINITION	124	BLK_ID	{null}	{null}	{null}
AVTTB_BLOCK_GROUP	99	BLG_ID	{null}	{null}	{null}
AVTTB_BLOCK_INSERTION	42	BLI_ID	{null}	{null}	{null}
AVTTB_BLOCK_SHAPE	4	BLS_ID	{null}	{null}	{null}
AVTTB_BLOCK_TYPE	9	BLT_ID	{null}	{null}	{null}
AVTTB_CALENDAR	1	CAL_ID	Calenders	{null}	{null}
AVTTB_CALENDAR_TYPE	7	CAT_ID	{null}	{null}	{null}

The following statement will be required:

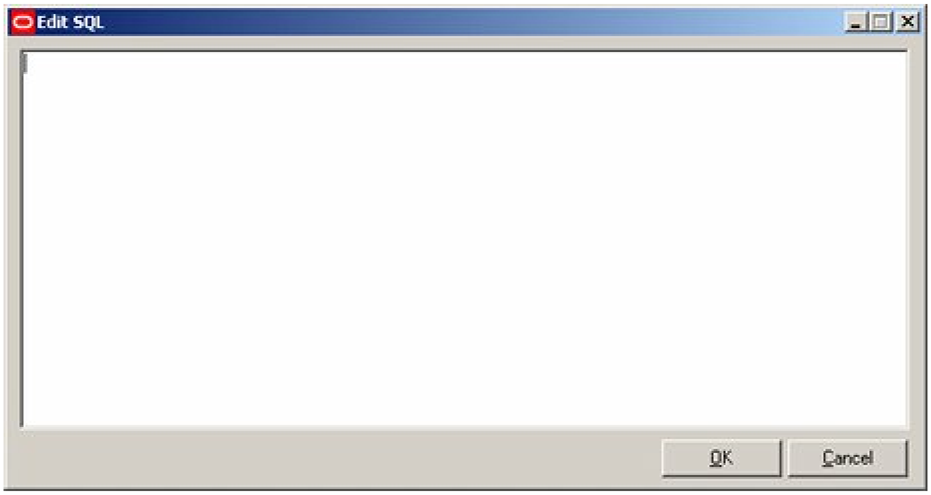
UPDATE AVTTB_NEXTID SET NID_NEXTID = (SELECT COALESCE(MAX(BLK_ID),0) + 1 FROM AVTTB_BLOCK_GROUP) WHERE NID_TABLENAME='AVTTB_BLOCK_GROUP'

This is added by clicking on the Pre SQL button at the bottom of the Import Design dialogue box for Hierarchy Level 1 to bring up the Pre-SQL dialogue box.



```
UPDATE AVTTB_NEXTID SET NID_NEXTID = (SELECT COALESCE(MAX(BLG_ID),0) + 1 FROM AVTTB_BLOCK_GROUP) WHERE
```

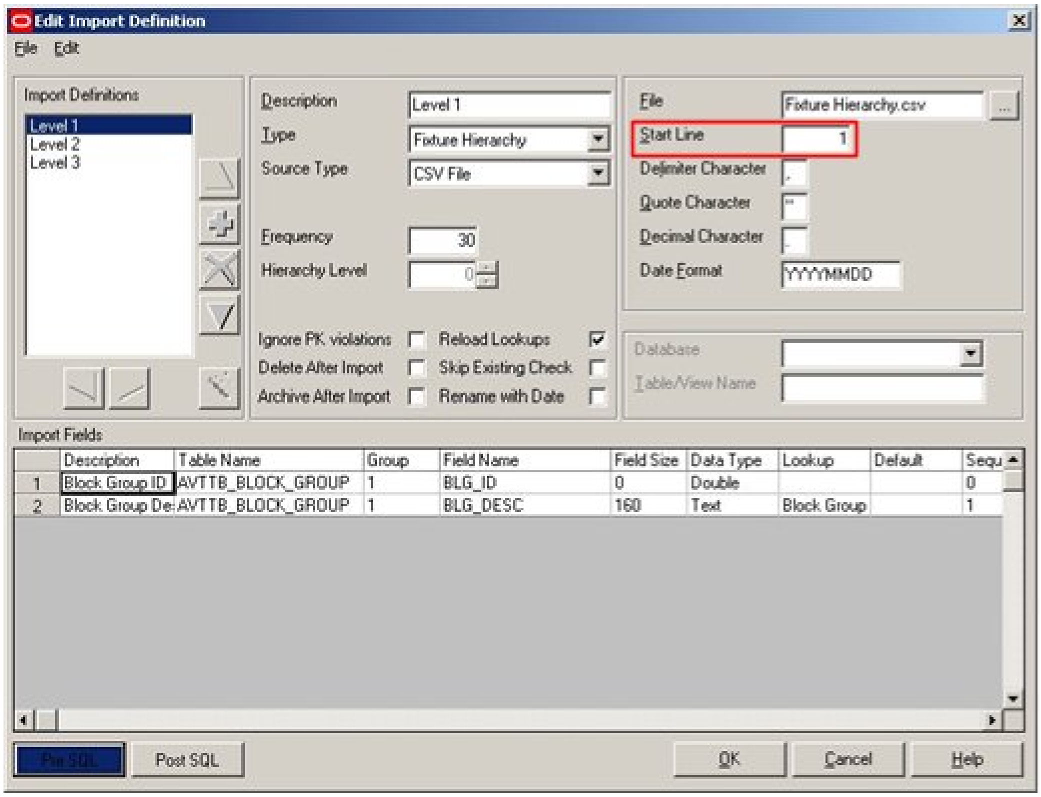
Clicking on the Add button will bring up the Add SQL dialogue box, where the SQL statement can be added.



Resetting the Start Line Number

Resetting the line number is important.

Until now we have had the line number set to 1, because this reads the first line of the .csv file - the information titles.



Edit Import Definition

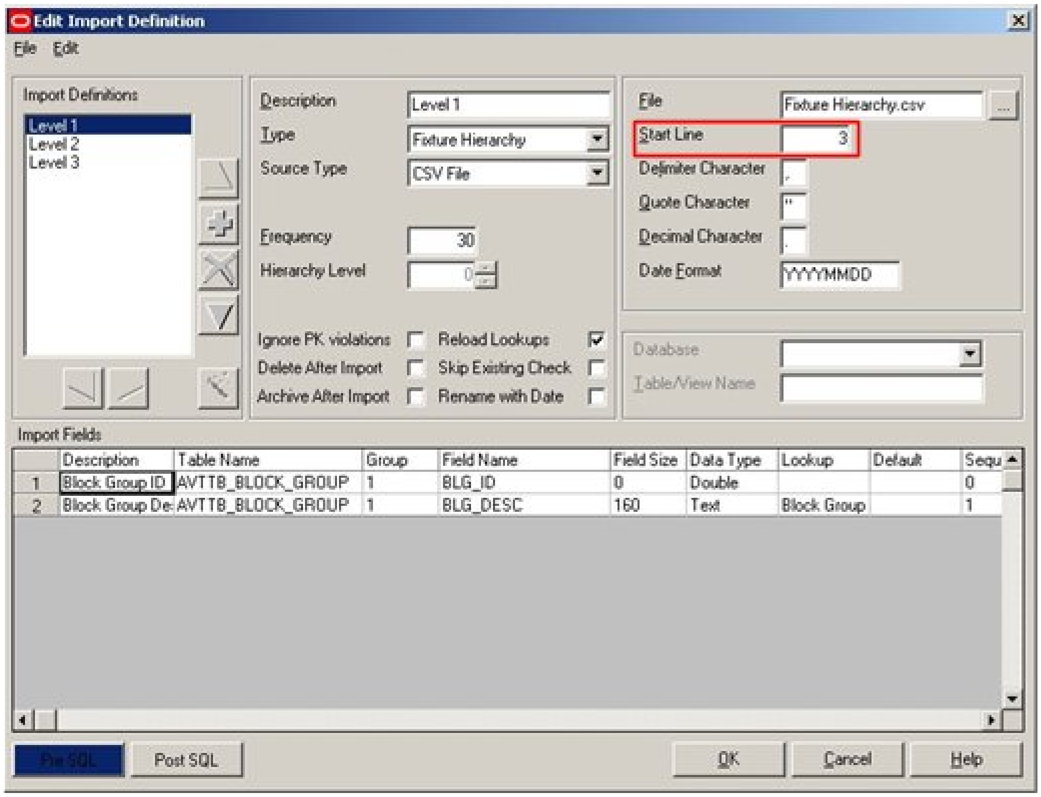
Description: Level 1
Type: Fixture Hierarchy
Source Type: CSV File
Frequency: 30
Hierarchy Level: 0
☐ Ignore PK violations ☑ Reload Lookups
☐ Delete After Import ☐ Skip Existing Check
☐ Archive After Import ☐ Rename with Date

File: Fixture Hierarchy.csv
Start Line: 1
Delimiter Character: ,
Quote Character: "
Decimal Character: .
Date Format: YYYYMMDD
Database:
Table/View Name:

Import Fields

	Description	Table Name	Group	Field Name	Field Size	Data Type	Lookup	Default	Sequ
1	Block Group ID	AVTTB_BLOCK_GROUP	1	BLG_ID	0	Double			0
2	Block Group De	AVTTB_BLOCK_GROUP	1	BLG_DESC	160	Text	Block Group		1

We must now set this to line number 3 for each of the Import definitions, so we read the first line of actual data.



Edit Import Definition

Description: Level 1
Type: Fixture Hierarchy
Source Type: CSV File
Frequency: 30
Hierarchy Level: 0
☐ Ignore PK violations ☑ Reload Lookups
☐ Delete After Import ☐ Skip Existing Check
☐ Archive After Import ☐ Rename with Date

File: Fixture Hierarchy.csv
Start Line: 3
Delimiter Character: ,
Quote Character: "
Decimal Character: .
Date Format: YYYYMMDD
Database:
Table/View Name:

Import Fields

	Description	Table Name	Group	Field Name	Field Size	Data Type	Lookup	Default	Sequ
1	Block Group ID	AVTTB_BLOCK_GROUP	1	BLG_ID	0	Double			0
2	Block Group De	AVTTB_BLOCK_GROUP	1	BLG_DESC	160	Text	Block Group		1

This means the Data Import Process will read from line 3, which is the first line of data we actually want to import.

Note: Failing to reset the line number before running Data Importer is a common source of error.

Testing the Import Definition

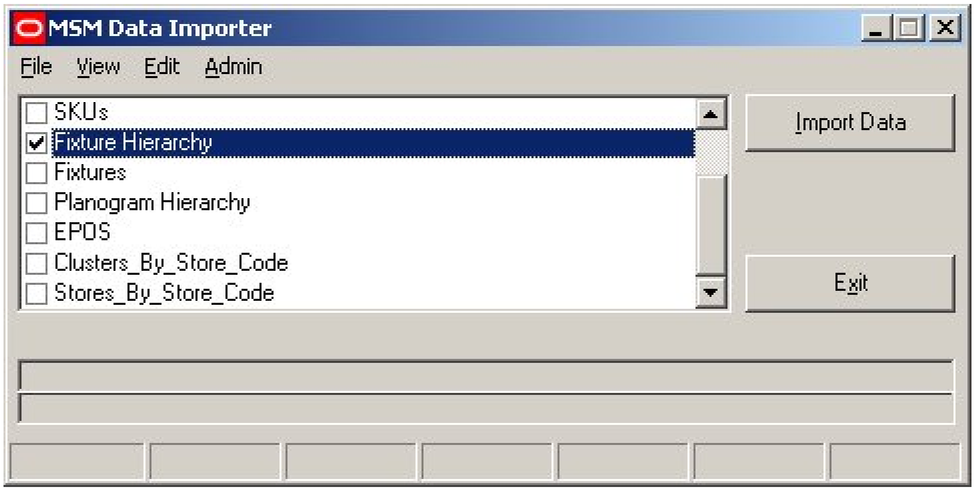
Overview of Testing the Import Definition

It is generally a good idea to test any new import definition with a few lines of data. This will allow problems to be identified and analyzed before a full import (which could take an appreciable period of time) is carried out.

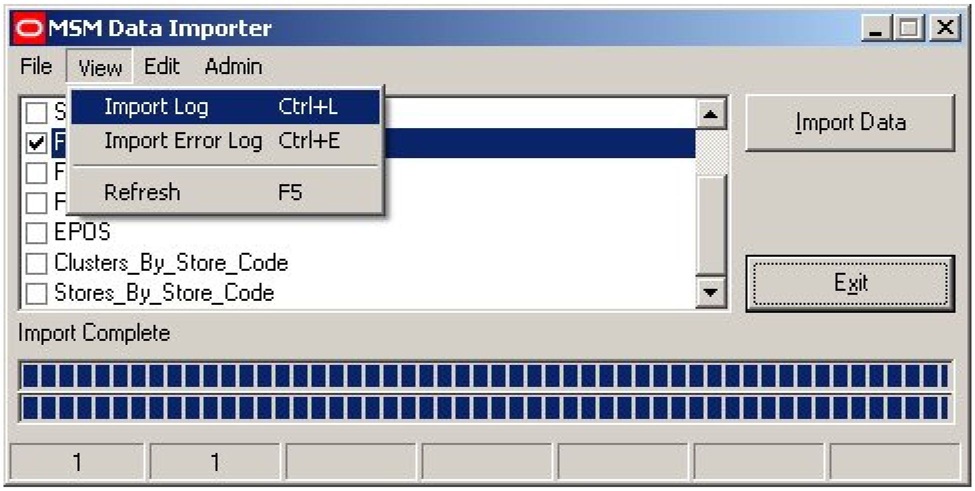
In this instance, there are only three lines of data in the Hierarchy to be imported, so it is unnecessary to carry out a small scale test.

Running Data Importer and Viewing the Error Log

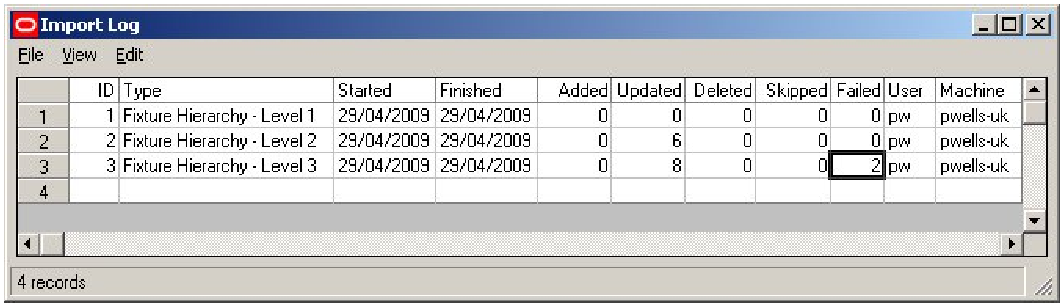
Data Importer can be run by ticking the Fixture Hierarchy checkbox and clicking on Import Data.



After the Import has run use the Import Log option on the View pull down menu to see how the Import has performed.



This will bring up the Import log.



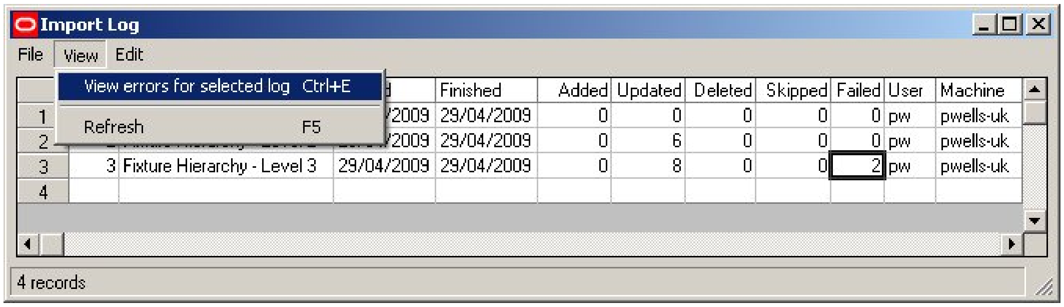
	ID	Type	Started	Finished	Added	Updated	Deleted	Skipped	Failed	User	Machine
1	1	Fixture Hierarchy - Level 1	29/04/2009	29/04/2009	0	0	0	0	0	pw	pwells-uk
2	2	Fixture Hierarchy - Level 2	29/04/2009	29/04/2009	0	6	0	0	0	pw	pwells-uk
3	3	Fixture Hierarchy - Level 3	29/04/2009	29/04/2009	0	8	0	0	2	pw	pwells-uk
4											

4 records

Examination of the record shows that there has been a failure during the import of Hierarchy Level 3.

Viewing the Error Log

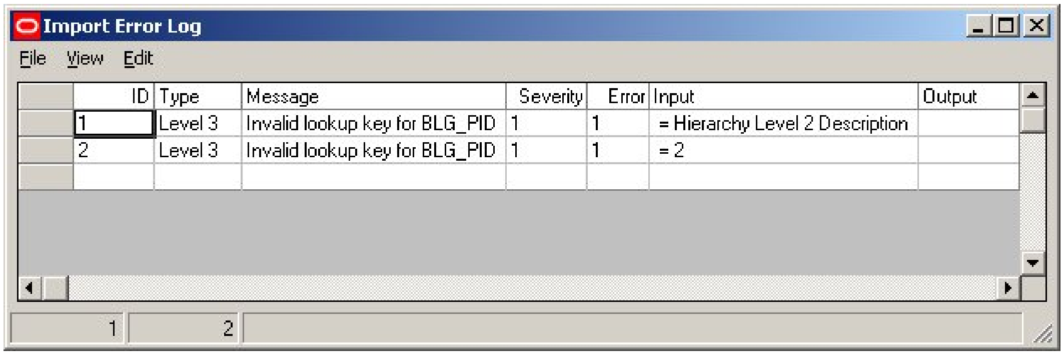
We need to view the Error log for that specific part of the import. To do this, highlight the required line in the Import Log and select the View errors for selected log option from the View pull down menu.



		d	Finished	Added	Updated	Deleted	Skipped	Failed	User	Machine
1		/2009	29/04/2009	0	0	0	0	0	pw	pwells-uk
2		/2009	29/04/2009	0	6	0	0	0	pw	pwells-uk
3	3 Fixture Hierarchy - Level 3	29/04/2009	29/04/2009	0	8	0	0	2	pw	pwells-uk
4										

4 records

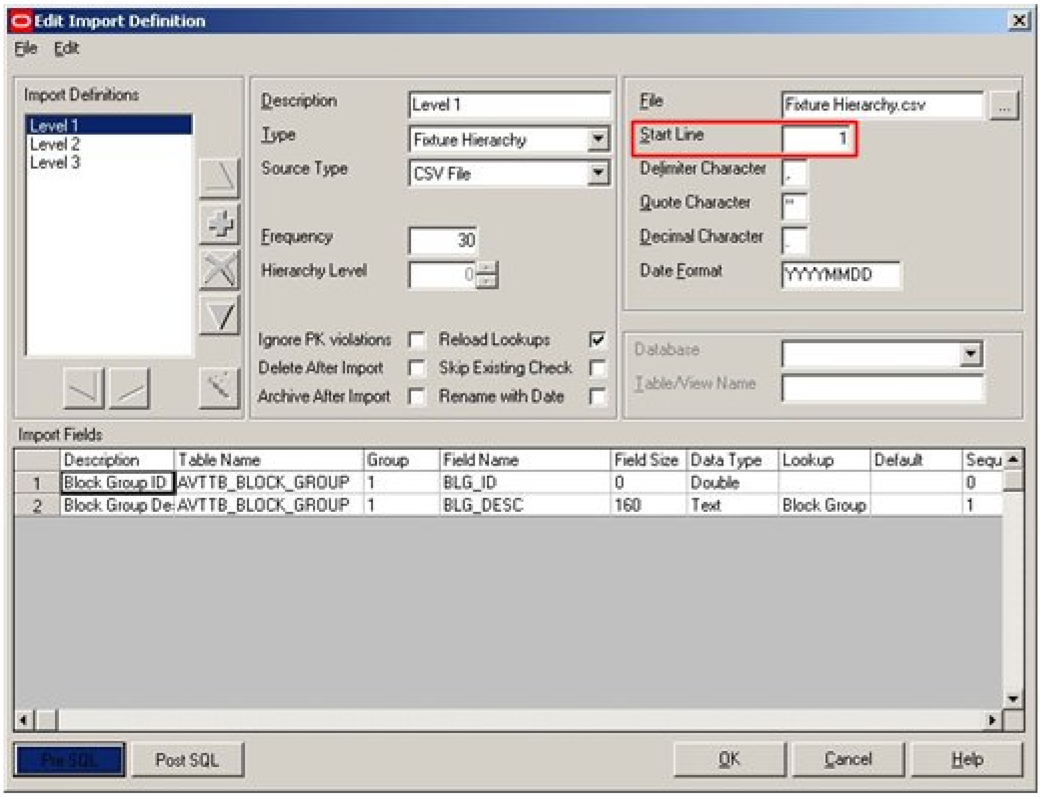
This will bring up the Import Error Log.



	ID	Type	Message	Severity	Error	Input	Output
	1	Level 3	Invalid lookup key for BLG_PID	1	1	= Hierarchy Level 2 Description	
	2	Level 3	Invalid lookup key for BLG_PID	1	1	= 2	

1 2

The Input in the error log is 'Hierarchy Level 2' and then '2'. This suggests we are importing data from lines one and two of the .csv file, not line three as we should be. Because the data is not as expected, the import is errorring.



Import Definitions: Level 1, Level 2, Level 3

Description	Level 1
Type	Fixture Hierarchy
Source Type	CSV File
Frequency	30
Hierarchy Level	0
File	Fixture Hierarchy.csv
Start Line	1
Delimiter Character	,
Quote Character	"
Decimal Character	.
Date Format	YYYYMMDD
Database	
Table/View Name	

Ignore PK violations ☐ Reload Lookups ☑
Delete After Import ☐ Skip Existing Check ☐
Archive After Import ☐ Rename with Date ☐

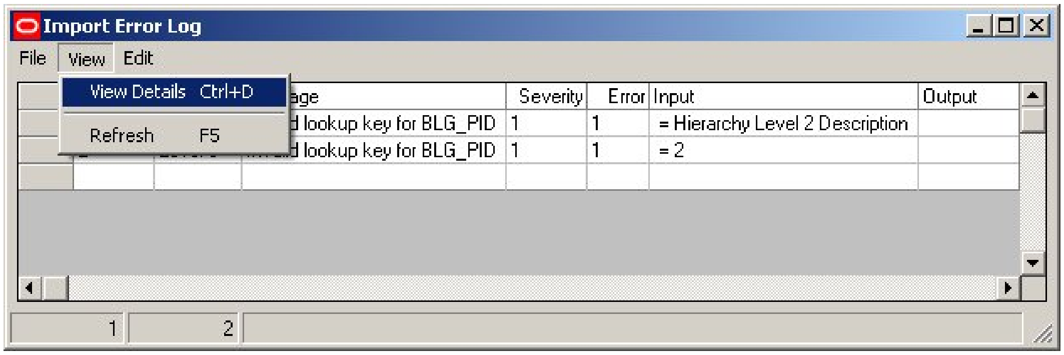
Import Fields

	Description	Table Name	Group	Field Name	Field Size	Data Type	Lookup	Default	Sequ
1	Block Group ID	AVTTB_BLOCK_GROUP	1	BLG_ID	0	Double			0
2	Block Group De:	AVTTB_BLOCK_GROUP	1	BLG_DESC	160	Text	Block Group		1

This would be readily corrected by changing the Start Line to '3', allowing the import to run correctly.

Seeing the Error in more Detail

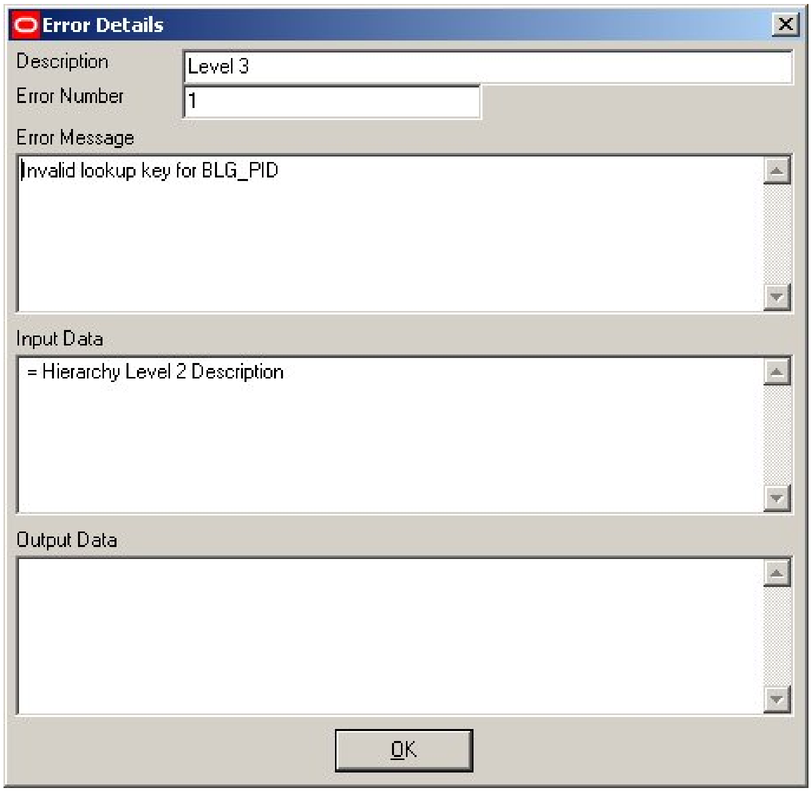
To **see the error in more detail**, it is possible to do so by selecting the View Details option from the View pull down menu.



	...age	Severity	Error	Input	Output
	...d lookup key for BLG_PID	1	1	= Hierarchy Level 2 Description	
	...d lookup key for BLG_PID	1	1	= 2	

1 2

This will bring up the Error Details dialogue box, which allows the error to be seen more clearly.



Description: Level 3

Error Number: 1

Error Message: Invalid lookup key for BLG_PID

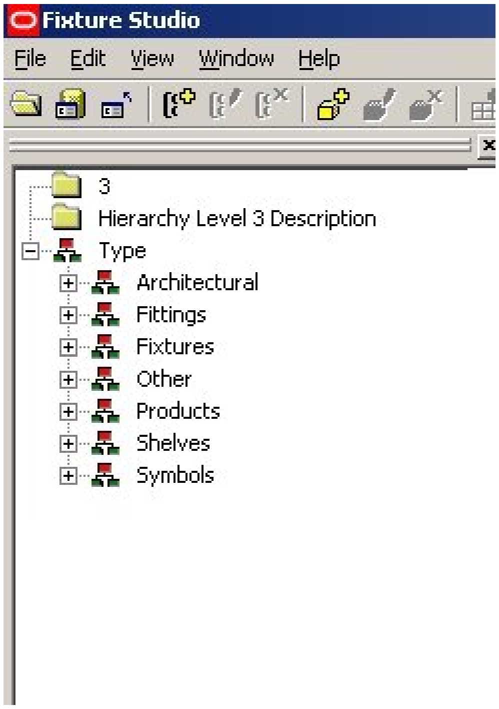
Input Data: = Hierarchy Level 2 Description

Output Data:

This is useful if the data is extensive and difficult to see in its entirety in the Import Error Log.

Consequences of the Error

The error has resulted in some unintended groups being added to the Fixture hierarchy.



This is a good illustration as to why the import should be tested on small quantities of data before being run on a large quantity.

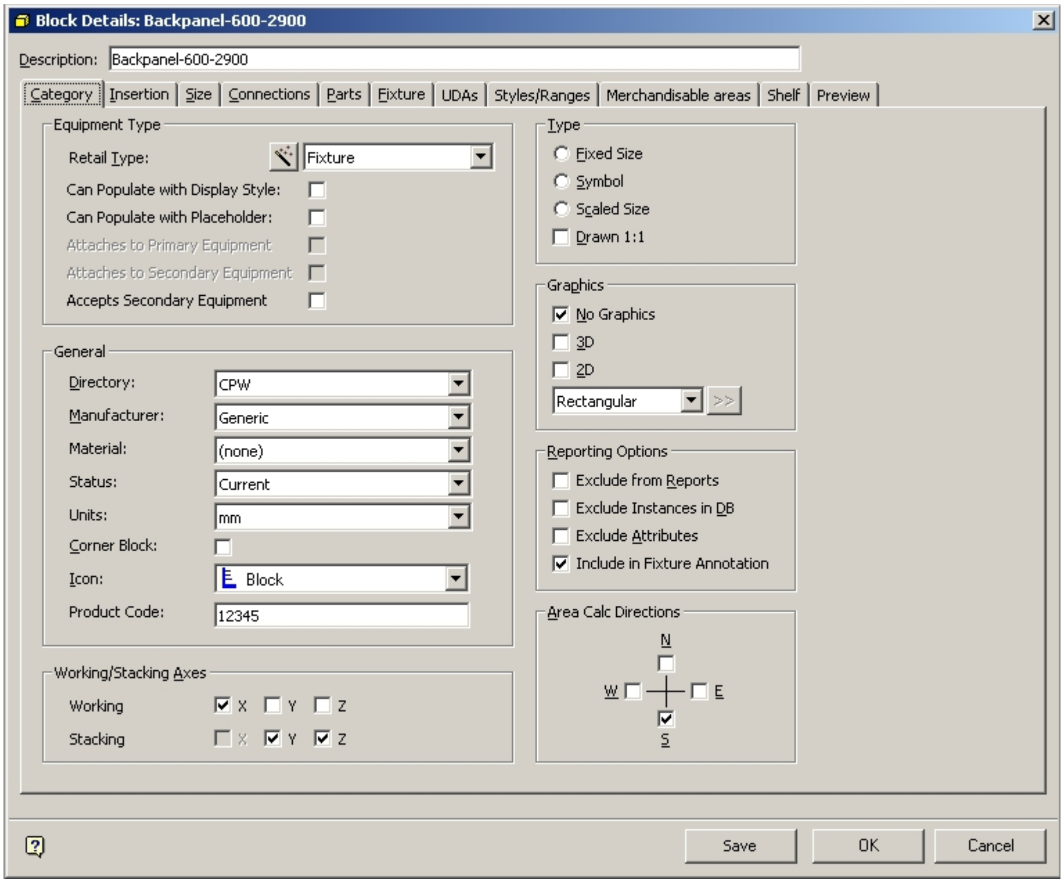
6

Worked Example - Adding Fixtures

Planning Adding Fixtures

Adding Fixtures

Adding Fixtures is more complex than adding groups to the fixture hierarchy.



Block Details: Backpanel-600-2900

Description: Backpanel-600-2900

Equipment Type

Retail Type: Fixture

Can Populate with Display Style: □

Can Populate with Placeholder: □

Attaches to Primary Equipment □

Attaches to Secondary Equipment □

Accepts Secondary Equipment □

General

Directory: CPW

Manufacturer: Generic

Material: (none)

Status: Current

Units: mm

Corner Block: □

Icon: Block

Product Code: 12345

Working/Stacking Axes

Working ☑ X □ Y □ Z

Stacking □ X ☑ Y ☑ Z

Type

○ Fixed Size

○ Symbol

○ Scaled Size

□ Drawn 1:1

Graphics

☑ No Graphics

□ 3D

□ 2D

Rectangular

Reporting Options

□ Exclude from Reports

□ Exclude Instances in DB

□ Exclude Attributes

☑ Include in Fixture Annotation

Area Calc Directions

N □

W □ □ E

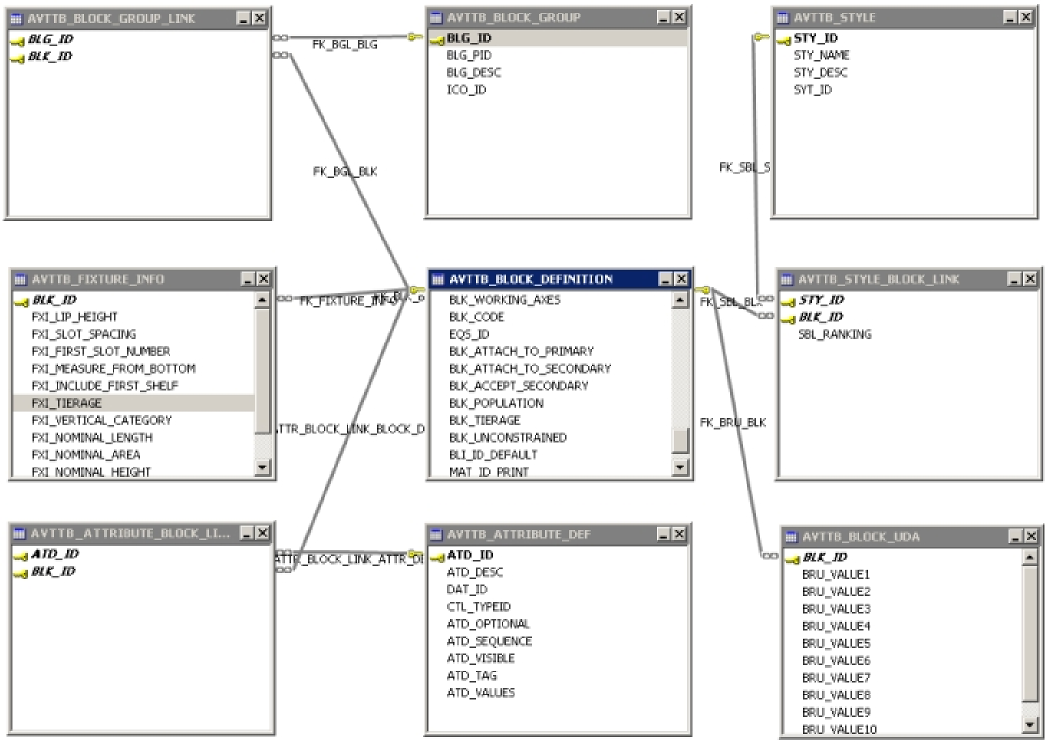
S ☑

Each Fixture has a significant amount of information associated with it - much of which can be imported via Data Importer.

This example will import some (but not all) of the data required to fully define a fixture.

Planning the Fixture Import from the Schema

The exercise begins by reviewing the Entity Relationship Diagram.



AVTTB_BLOCK_GROUP_LINK
BLG_ID
BLK_ID

AVTTB_BLOCK_GROUP
BLG_ID
BLG_PID
BLG_DESC
ICO_ID

AVTTB_STYLE
STY_ID
STY_NAME
STY_DESC
SYT_ID

AVTTB_FIXTURE_INFO
BLK_ID
FXI_LIP_HEIGHT
FXI_SLOT_SPACING
FXI_FIRST_SLOT_NUMBER
FXI_MEASURE_FROM_BOTTOM
FXI_INCLUDE_FIRST_SHELF
FXI_TIERAGE
FXI_VERTICAL_CATEGORY
FXI_NOMINAL_LENGTH
FXI_NOMINAL_AREA
FXI_NOMINAL_HEIGHT

AVTTB_BLOCK_DEFINITION
BLK_WORKING_AXES
BLK_CODE
EQS_ID
BLK_ATTACH_TO_PRIMARY
BLK_ATTACH_TO_SECONDARY
BLK_ACCEPT_SECONDARY
BLK_POPULATION
BLK_TIERAGE
BLK_UNCONSTRAINED
BLI_ID_DEFAULT
MAT_ID_PRINT

AVTTB_STYLE_BLOCK_LINK
STY_ID
BLK_ID
SBL_RANKING

AVTTB_ATTRIBUTE_BLOCK_LI...
ATD_ID
BLK_ID

AVTTB_ATTRIBUTE_DEF
ATD_ID
ATD_DESC
DAT_ID
CTL_TYPEID
ATD_OPTIONAL
ATD_SEQUENCE
ATD_VISIBLE
ATD_TAG
ATD_VALUES

AVTTB_BLOCK_UDA
BLK_ID
BRU_VALUE1
BRU_VALUE2
BRU_VALUE3
BRU_VALUE4
BRU_VALUE5
BRU_VALUE6
BRU_VALUE7
BRU_VALUE8
BRU_VALUE9
BRU_VALUE10

Note: The above diagram has been simplified from the full diagram. It shows the main tables, but further reference to the Entity Relationship Diagram may be required.

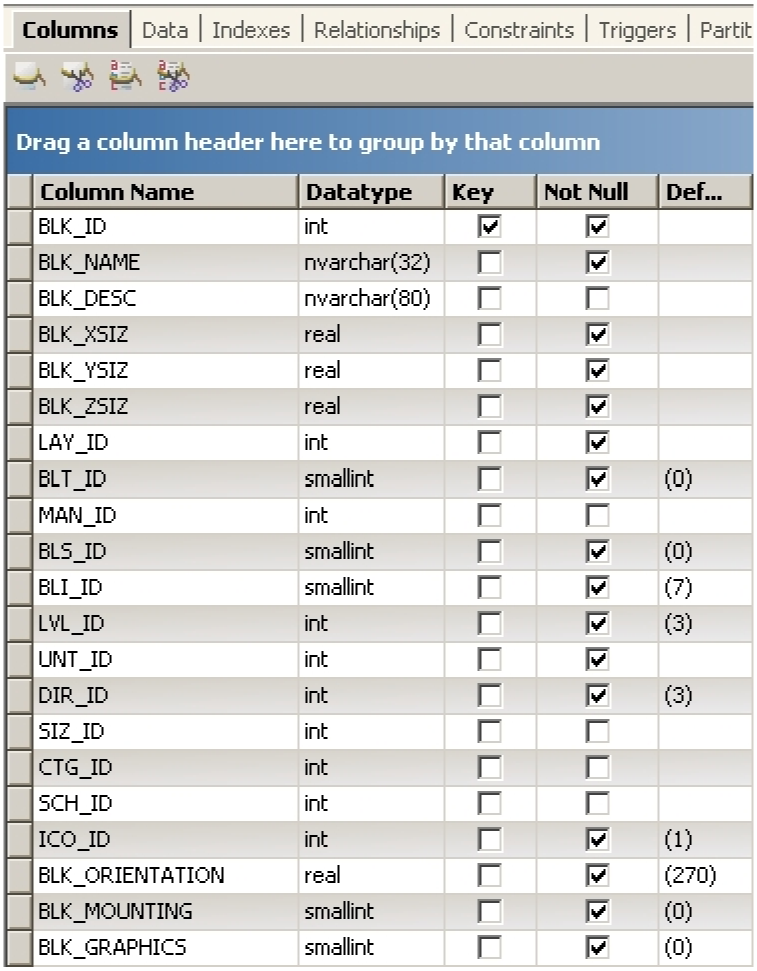
The majority of the information for a specific block is held in the Block Definitions Table (AVTTB_BLOCK_DEFINITIONS), but other tables will necessitate information being added.

Table	Purpose
AVTTB_BLOCK_GROUP_LINK	Holds the link between a Fixture and it's parent Fixture Group
AVTTB_FIXTURE_INFO	Holds information on a number of settings for the Fixture
AVTTB_ATTRIBUTE_BLOCK_LINK	Holds the link between a Fixture and it's assigned Attributes
AVTTB_ATTRIBUTE_DEF	Holds the block's (Fixture's) assigned Attributes
AVTTB_STYLE_BLOCK_LINK	Holds the link between a fixture and its assigned Style.
AVTTB_BLOCK_UDA	Holds the information assigned to the UDA

We next need to identify which information is required for each of these tables.

Reviewing the Block Definition Table

The Block Definition table (AVTTB_BLOCK_DEFINITION) holds much of the information on a specific type of block. It should be reviewed to establish which columns of data need to be populated.



Column Name	Datatype	Key	Not Null	Def...
BLK_ID	int	☑	☑	
BLK_NAME	nvarchar(32)	□	☑	
BLK_DESC	nvarchar(80)	□	□	
BLK_XSIZ	real	□	☑	
BLK_YSIZ	real	□	☑	
BLK_ZSIZ	real	□	☑	
LAY_ID	int	□	☑	
BLT_ID	smallint	□	☑	(0)
MAN_ID	int	□	□	
BLS_ID	smallint	□	☑	(0)
BLI_ID	smallint	□	☑	(7)
LVL_ID	int	□	☑	(3)
UNT_ID	int	□	☑	
DIR_ID	int	□	☑	(3)
SIZ_ID	int	□	□	
CTG_ID	int	□	□	
SCH_ID	int	□	□	
ICO_ID	int	□	☑	(1)
BLK_ORIENTATION	real	□	☑	(270)
BLK_MOUNTING	smallint	□	☑	(0)
BLK_GRAPHICS	smallint	□	☑	(0)

Note: The above screen shot only shows part of the Block Definition (AVTTB_BLOCK_DEFINITION table).

All fields marked as not-null, need to be populated (either directly or by means of a default). In additional, many nullable fields may require populating - for example the Block Description (BLK_DESC) field.

Note: Not Null fields that have assigned default values may not need to be populated as the default value will be adopted in the case of no value being directly input.

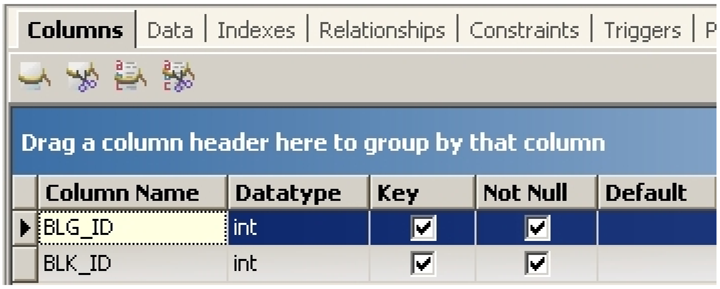
Some fields will be populated directly in this table, and some fields will be read from other tables. Accordingly, as the required input information is listed, it will be necessary to identify which parent table the information is input into.

Field	Purpose	Data Type	Not Null	Default	Parent table	Form
BLK_ID	Block ID (Primary key)	Int	Yes	None	N/A	Look up ID
BLK_NAME	Block Name	Nvarchar	Yes	None	N/A	User Input
BLK_DESC	Block Description	Nvarchar	No	None	N/A	User Input
BLK_XSIZ	Block Length	Real	Yes	None	N/A	User Input
BLK_YSIZ	Block Depth	Real	Yes	None	N/A	User Input
BLK_ZSIZ	Block Height	Real	Yes	None	N/A	User Input
LAY_ID	Parent AutoCAD layer	Int	Yes	None	AVTTB_LAYER_ALIAS	User Input /look-up
BLK_TYP_ID	Block type	Small Int	Yes	0	AVTTB_BLOCK_TYPE	User Input /look-up
BLS_ID	Block Shape	Small Int	Yes	0	AVTTB_BLK_SHAPE	User Input /look-up
BLI_ID	Block Insertion Point	Small Int	Yes	0	AVTTB_BLK_INSERTION	User Input /look-up
LVL_ID	Block Level	Int	Yes	3	AVTTB_LEVELTYPE	User Input /look-up
UNIT_ID	Units	Int	Yes	None	AVTTB_UNIT	User Input /look-up
DIR_ID	Directory	Int	Yes	3	AVTTB_DIRECTORY	User Input /look-up
SIZ_ID	Size Group	Int	No	None	AVTTB_SIZE	User Input /look-up

Field	**Purpose**	**Data Type**	**Not Null**	**Default**	**Parent table**	**Form**
STA_ID	Status	Int	Yes	12	AVTTB_STATUS	User Input /look-up
BLK_CREATED	Date block Created	Datetime	Yes	None	N/A	Date Option
BLK_UPDATED	Date block updated	Datetime	Yes	None	N/A	Date Option
BLK_CREATED_BY	User who Created Block	Nvarchar	Yes	None	N/A	User Option
BLK_UPDATED_BY	User who updated block	Nvarchar	Yes	None	N/A	User Option
BLK_CODE	Block Code	Nvarchar	No	None	N/A	User Option

Reviewing the Block-Group Link Table

The Block-Group Link table (AVTTB_BLOCK_GROUP_LINK) should be reviewed to establish which columns of data need to be populated.



Column Name	Datatype	Key	Not Null	Default
BLG_ID	int	☑	☑	
BLK_ID	int	☑	☑	

All fields marked as not-null need to be populated.

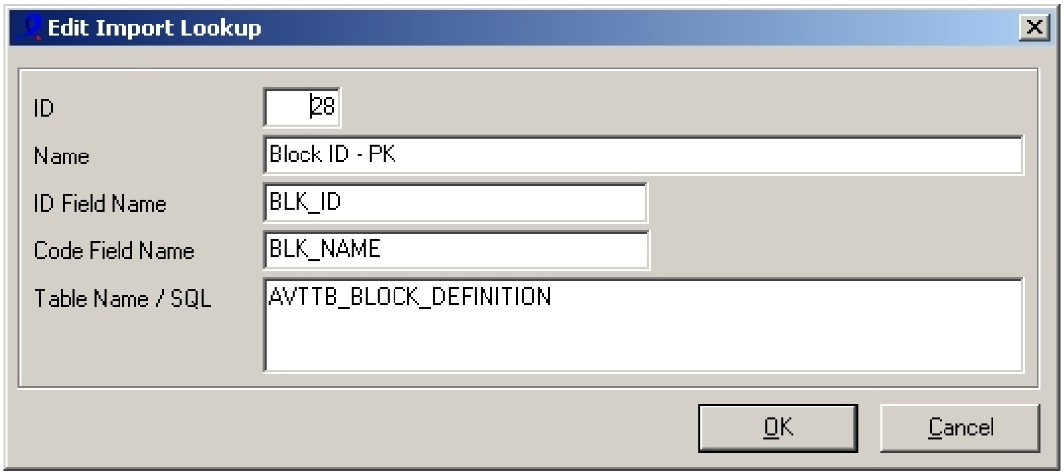
Both fields will be read from other tables. Accordingly, as the required input information is listed, it will be necessary to identify which parent table the information is input into.

Field	**Purpose**	**Data Type**	**Not Null**	**Default**	**Parent table**	**Form**
BLK_ID	Block ID	Int	Yes	No	AVTTB_BLOCK_DEFINITION	User Input /look-up
BLG_ID	Fixture Group ID	Int	Yes	No	AVTTB_BLOCK_GROUP	User Input /look-up

Populating this table will require two lookups: one in the AVTTB_BLOCK_DEFINITION table and one in the AVTTB_BLOCK_GROUP table.

BLK_ID

This look up is defined for the AVTTB_BLOCK_DESCRIPTION TABLE

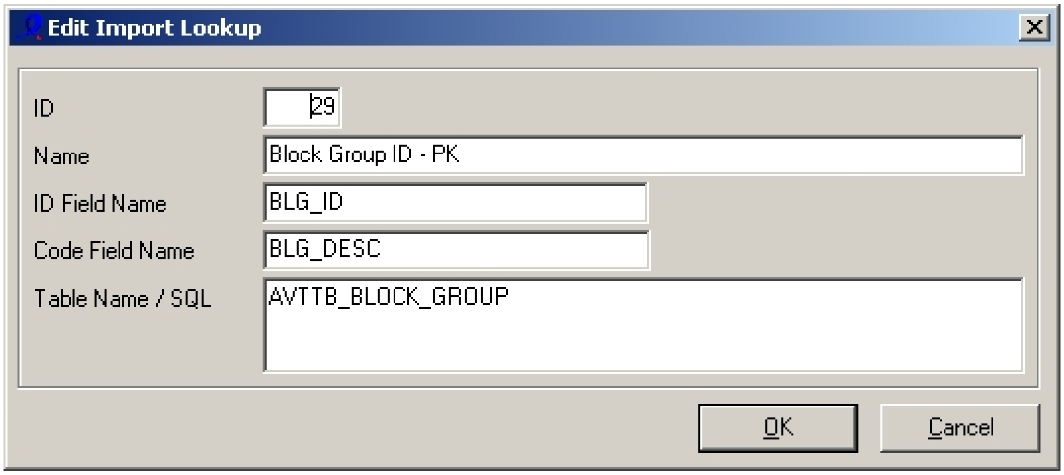


ID	28
Name	Block ID - PK
ID Field Name	BLK_ID
Code Field Name	BLK_NAME
Table Name / SQL	AVTTB_BLOCK_DEFINITION

The input from the .csv file will be the name of a Block [Fixture] (BLK_Name), and the look-up will return the Block ID (BLK_ID) assigned to that block in the database.

BLG_ID

This look up is defined for the AVTTB_BLOCK_GROUP table.

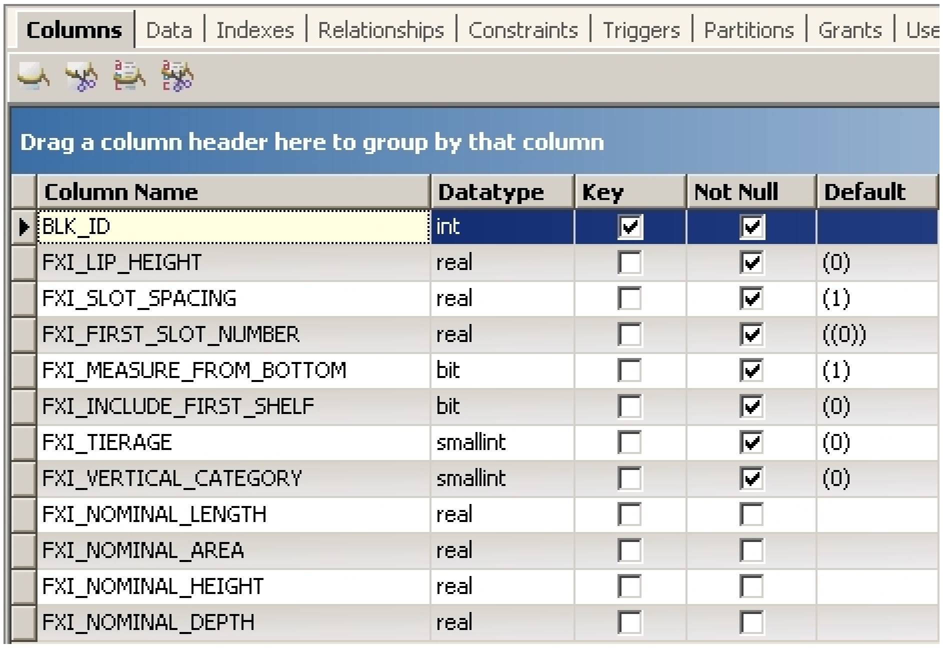


ID	29
Name	Block Group ID - PK
ID Field Name	BLG_ID
Code Field Name	BLG_DESC
Table Name / SQL	AVTTB_BLOCK_GROUP

The input from the .csv file will be the name of a Block [Fixture] Group (BLG_DESC), and the look-up will return the Block Group ID (BLG_ID) assigned to that block group in the database.

Reviewing the Fixture Info Table

The **Fixture Info Definition table (AVTTB_FIXTURE_INFO)** should be reviewed to establish which columns of data need to be populated.



Drag a column header here to group by that column

Column Name	Datatype	Key	Not Null	Default
BLK_ID	int	☑	☑	
FXI_LIP_HEIGHT	real	□	☑	(0)
FXI_SLOT_SPACING	real	□	☑	(1)
FXI_FIRST_SLOT_NUMBER	real	□	☑	((0))
FXI_MEASURE_FROM_BOTTOM	bit	□	☑	(1)
FXI_INCLUDE_FIRST_SHELF	bit	□	☑	(0)
FXI_TIERAGE	smallint	□	☑	(0)
FXI_VERTICAL_CATEGORY	smallint	□	☑	(0)
FXI_NOMINAL_LENGTH	real	□	□	
FXI_NOMINAL_AREA	real	□	□	
FXI_NOMINAL_HEIGHT	real	□	□	
FXI_NOMINAL_DEPTH	real	□	□	

All fields marked as not-null need to be populated. In additional, some nullable fields may require populating - for example the Nominal Length (FXI_NOMINAL_LENGTH) field.

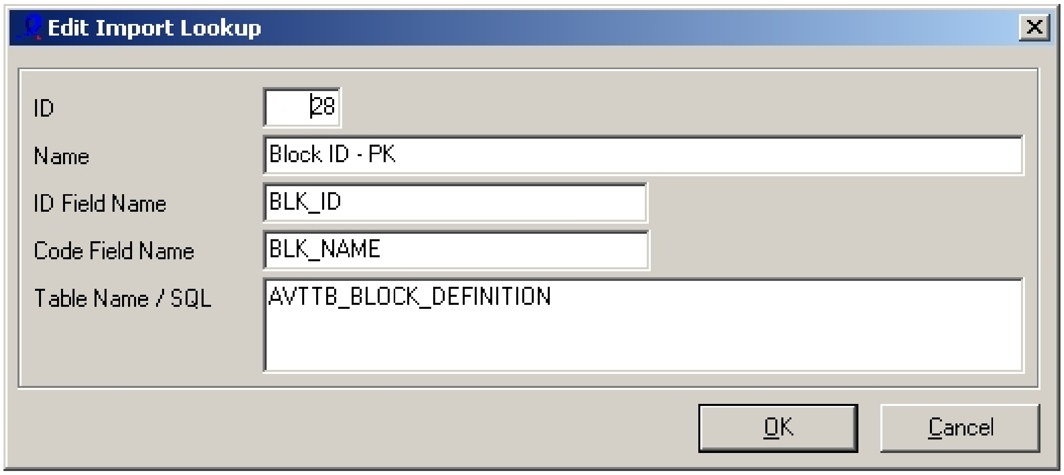
> **Note**: Not Null fields that have assigned default values may not need to be populated as the default value will be adopted in the case of no value being directly input.

Some fields will be populated directly in this table, and some fields will be read from other tables. Accordingly, as the required input information is listed, it will be necessary to identify which parent table the information is input into.

Field	**Purpose**	**Data Type**	**Not Null**	**Default**	**Parent table**	**Form**
BLK_ID	Block ID	Int	Yes	No	AVTTB_BLOCK_DEFINITION	User Input /look-up
FXI_NOMINAL_LENGTH	Nominal Length	Real	No	No	N/A	User Input
FXI_NOMINAL_HEIGHT	Nominal Height	Real	No	No	N/A	User Input
FXI_NOMINAL DEPTH	Nominal Depth	Real	No	No	N/A	User Input

BLK_ID

This look up is defined for the AVTTB_BLOCK_DESCRIPTION TABLE

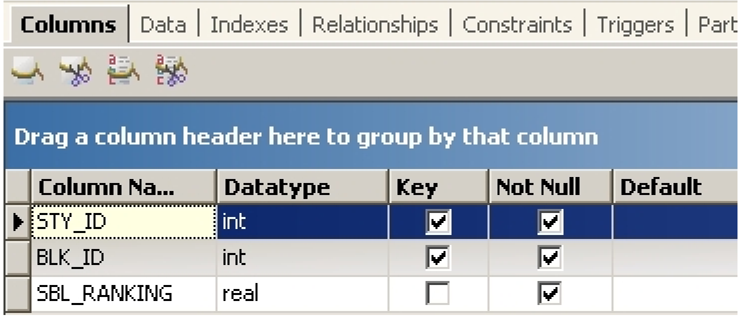


ID	28
Name	Block ID - PK
ID Field Name	BLK_ID
Code Field Name	BLK_NAME
Table Name / SQL	AVTTB_BLOCK_DEFINITION

The input from the .csv file will be the name of a Block (Fixture), and the look-up will return the Block ID (BLK_ID) assigned to that block in the database.

Reviewing the Style-Block Link Table

The **Style-Block Link table (AVTTB_STYLE_BLOCK_LINK)** should be reviewed to establish which columns of data need to be populated.



Column Na...	Datatype	Key	Not Null	Default
STY_ID	int	☑	☑	
BLK_ID	int	☑	☑	
SBL_RANKING	real	☐	☑	

All fields marked as not-null need to be populated.

Two of the three fields will be read from other tables. Accordingly, as the required input information is listed, it will be necessary to identify which parent table the information is input into.

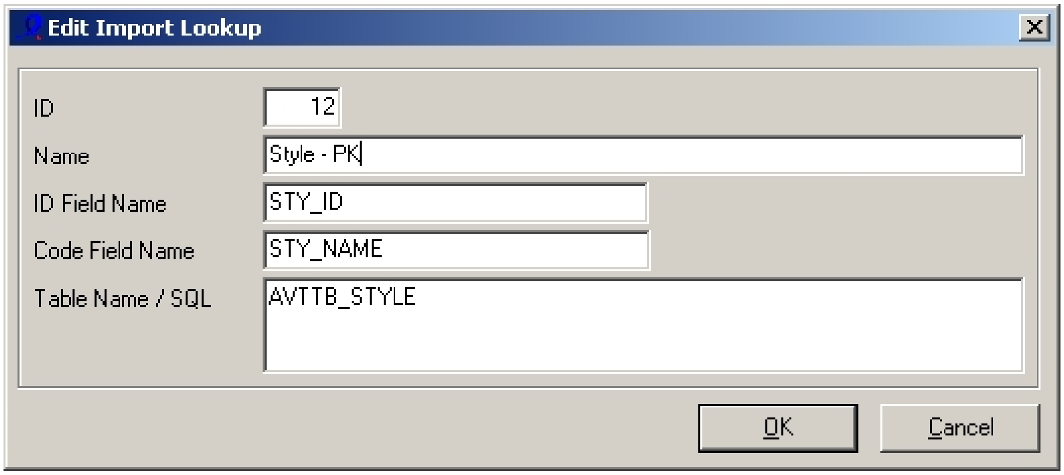
Field	Purpose	Data Type	Not Null	Default	Parent table	Form
STY_ID	Style ID	Int	Yes	No	AVTTB_STYLE	User Input /look-up
BLK_ID	Block ID	Int	Yes	No	AVTTB_BLOCK_DEFINITION	User Input /look-up
SBL_RANKING	Style Block Link Ranking	Real	Yes	No	N/A	User Input

Populating this table will require two lookups: one in the AVTTB_STYLE table and one in the AVTTB_BLOCK_DEFINITION table.

It will also require information to be input for the Style Block Link Ranking.

STY_ID

This look up is defined for the AVTTB_STYLE table.



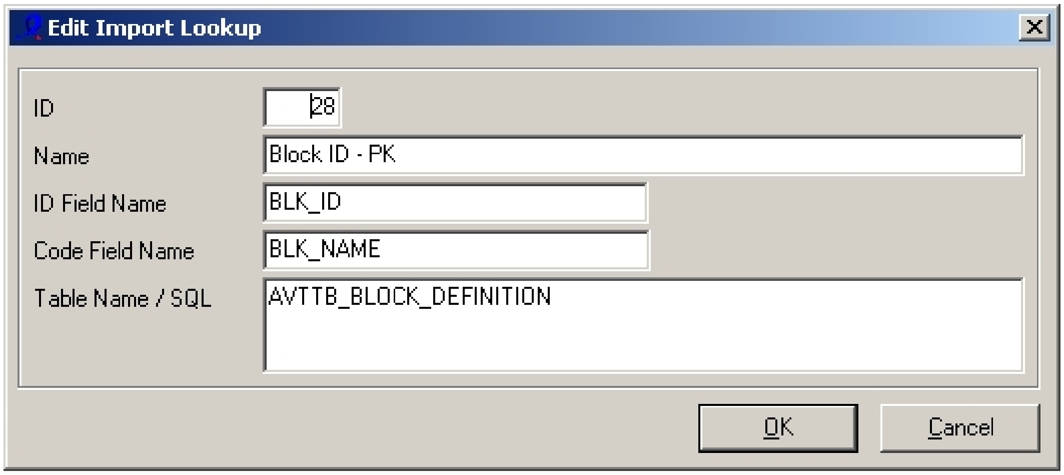
Edit Import Lookup

ID: 12
Name: Style - PK
ID Field Name: STY_ID
Code Field Name: STY_NAME
Table Name / SQL: AVTTB_STYLE

The input from the .csv file will be the name of a Style (STY_NAME), and the look-up will return the Style ID (STY_ID) assigned to that Style in the database. More Info

BLK_ID

This look up is defined for the AVTTB_BLOCK_DESCRIPTION table.



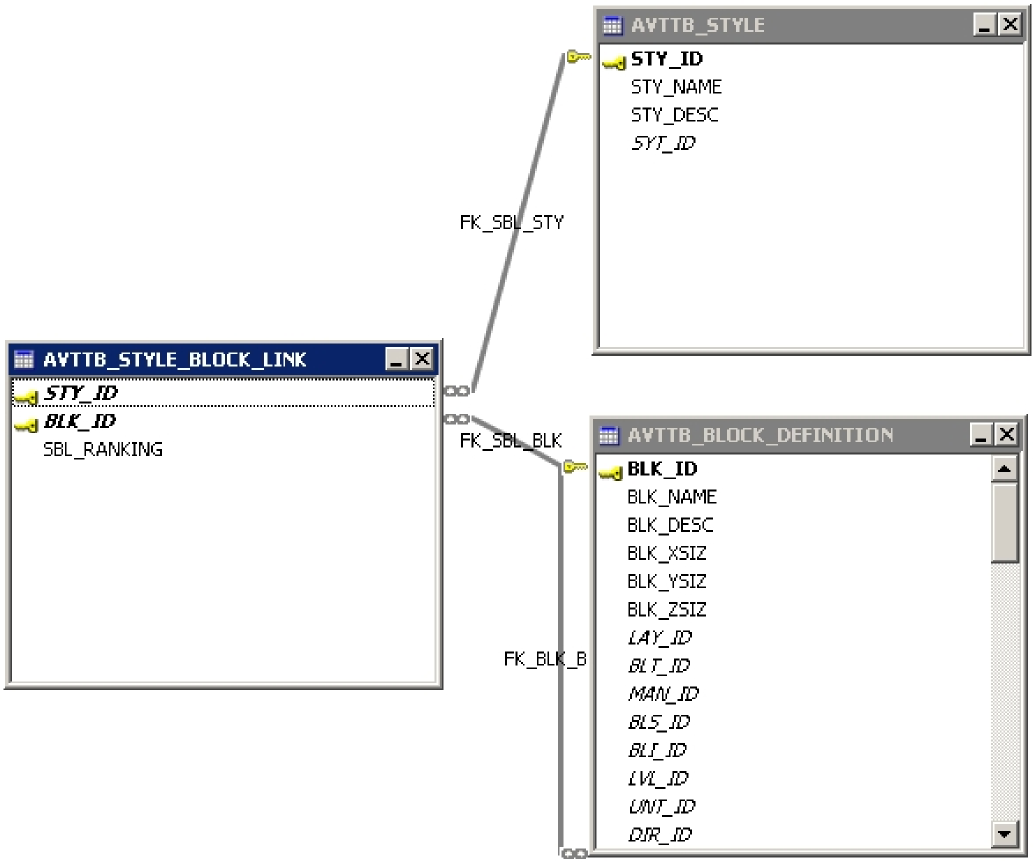
Edit Import Lookup

ID: 28
Name: Block ID - PK
ID Field Name: BLK_ID
Code Field Name: BLK_NAME
Table Name / SQL: AVTTB_BLOCK_DEFINITION

The input from the .csv file will be the name of a Block (Fixture), and the look-up will return the Block ID (BLK_ID) assigned to that block in the database.

Fixtures and Styles in the Schema

The need to **look up a Style ID** provides a good example of why the schema needs to be understood in detail.



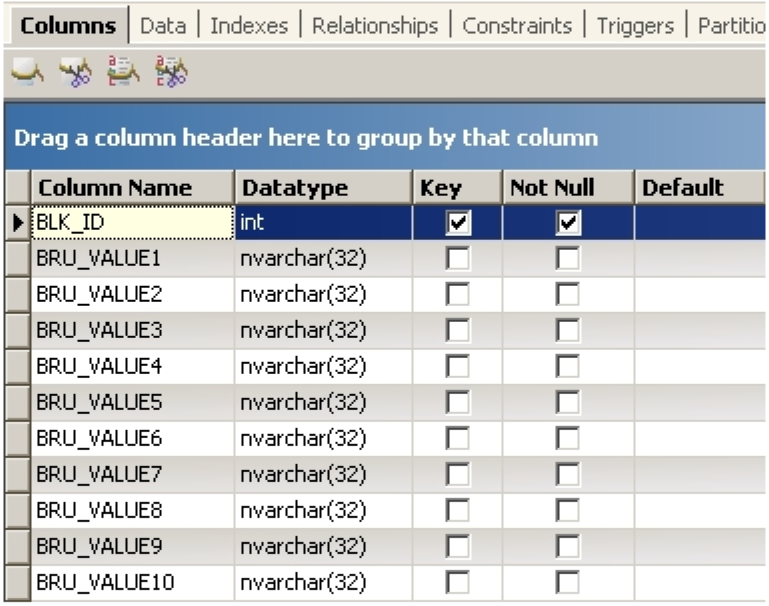
We need to add a Style ID (STY_ID) to the AVTTB_STYLE_BLOCK_LINK table. We will know the Style Name, and need to perform a Lookup on the AVTTB_STYLE table in order to return the value of the Style ID (STY_ID).

The AVTTB_STYLE database will not be changed in any way - just used as a source of information to populate the AVTTB_STYLE_BLOCK_LINK table.

Without this level of knowledge, we cannot populate the database with the correct values.

Reviewing the Block UDA Table

The **Block UDA table (AVTTB_BLOCK_UDA)** holds information on the User Defined Attributes assigned to each block. It should be reviewed to establish which columns of data need to be populated.

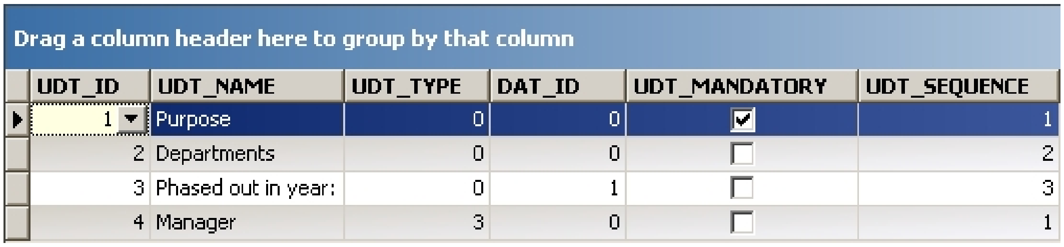


Drag a column header here to group by that column

Column Name	Datatype	Key	Not Null	Default
BLK_ID	int	☑	☑	
BRU_VALUE1	nvarchar(32)	□	□	
BRU_VALUE2	nvarchar(32)	□	□	
BRU_VALUE3	nvarchar(32)	□	□	
BRU_VALUE4	nvarchar(32)	□	□	
BRU_VALUE5	nvarchar(32)	□	□	
BRU_VALUE6	nvarchar(32)	□	□	
BRU_VALUE7	nvarchar(32)	□	□	
BRU_VALUE8	nvarchar(32)	□	□	
BRU_VALUE9	nvarchar(32)	□	□	
BRU_VALUE10	nvarchar(32)	□	□	

The BLK_ID field marked as not-null needs to be populated.

In additional, some nullable fields may require populating. This will depend on whether the specific UDA fields have been designated as mandatory when they were configured. This is quickly established by looking at the date in the AVTTB_UDA_TEMPLATE TABLE.



Drag a column header here to group by that column

UDT_ID	UDT_NAME	UDT_TYPE	DAT_ID	UDT_MANDATORY	UDT_SEQUENCE
1	Purpose	0	0	☑	1
2	Departments	0	0	□	2
3	Phased out in year:	0	1	□	3
4	Manager	3	0	□	1

UDT_TYPE type is Blocks (AVTTB_DATATYPE table). 'Purpose' is mandatory, while 'Departments' and 'Phased out in year:' are optional.

The Data Type (DAT_ID) is found in the Data Type table (AVTTB_DATATYPE): type 0 is string for example.

The UDT Sequence describes the sequence the columns will be populated in the AVTTB_BLOCKS_UDA table: 1 populating column one, etc.

The Block ID (BLK_ID) will be read from another table (AVTTB_BLOCK_DEFINITION) via a look up.

Field	Purpose	Data Type	Not Null	Default	Parent table	Form
BLK_ID	Block ID	Int	Yes	No	AVTTB_BLOCK_DEFINITION	User Input /look-up
BRU_Value1	Purpose	Text	Yes	No	N/A	User Input

Field	Purpose	Data Type	Not Null	Default	Parent table	Form
BRU_Value2	Department	Text	No	No	N/A	User Input
BRU_Value3	Phased out in Year:	Integer	No	No	N/A	User Input

The Capture Spreadsheet

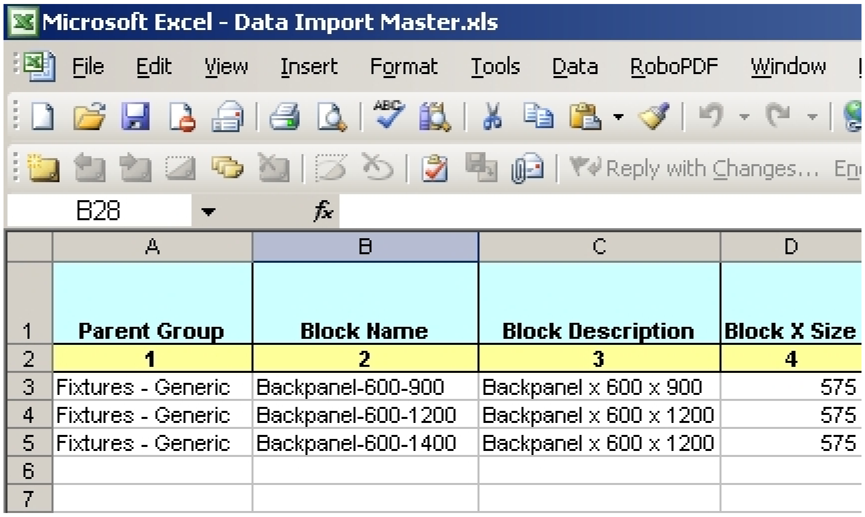
The Data Import Spreadsheet - Design

Before creating the Data Import spreadsheet, its **design** must be considered.

As errors in entering data into the spreadsheet, and errors extracting data from it, can lead to errors in importing data into the database, it is useful to design the spreadsheet in such a way that the possibility of error is minimized. There are several ways of achieving this:

- Labelling the spreadsheet columns clearly
- Numbering the spreadsheet columns
- Listing the tables and required fields
- Using lookups
- Validating Data input

Labelling the spreadsheet columns clearly

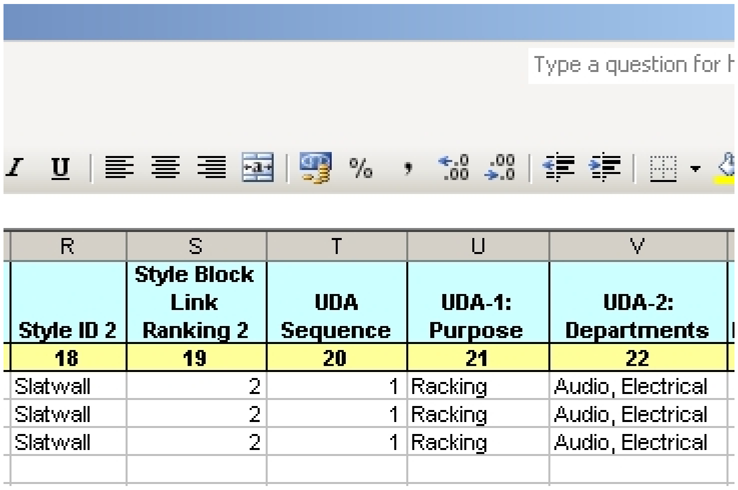


	A	B	C	D
1	Parent Group	Block Name	Block Description	Block X Size
2	1	2	3	4
3	Fixtures - Generic	Backpanel-600-900	Backpanel x 600 x 900	575
4	Fixtures - Generic	Backpanel-600-1200	Backpanel x 600 x 1200	575
5	Fixtures - Generic	Backpanel-600-1400	Backpanel x 600 x 1200	575
6				
7				

Column 1 of the spreadsheet can be read in during configuring data import. Clearly labelling the columns will make it easy to see the correct data is being selected during configuration.

Numbering the spreadsheet columns

In a large spreadsheet, it may be difficult to easily identify the correct column to reference. One way of overcoming this is to number the second row.

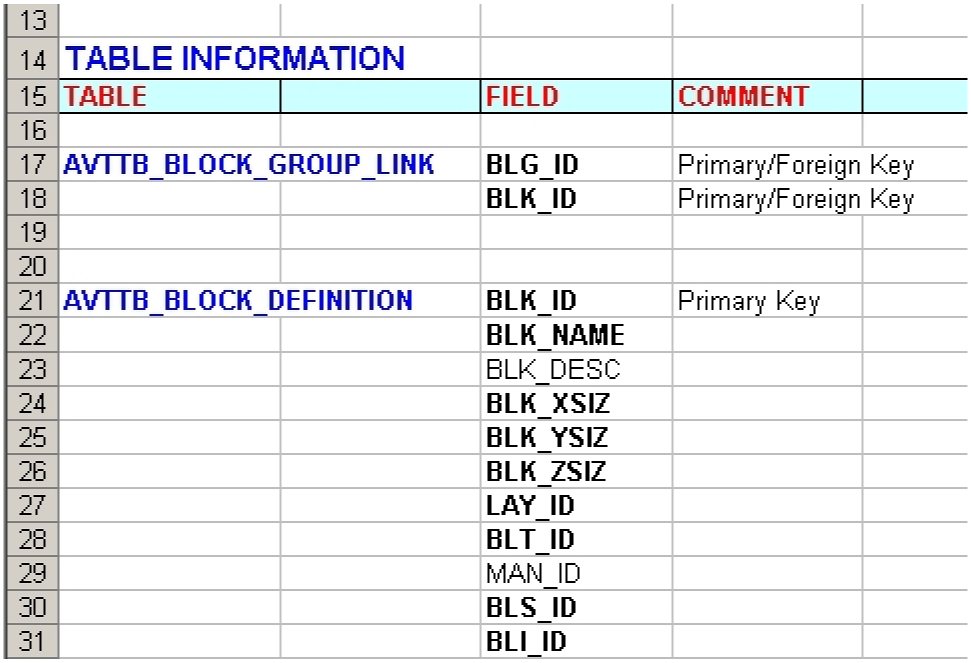


R	S	T	U	V
Style ID 2	Style Block Link Ranking 2	UDA Sequence	UDA-1: Purpose	UDA-2: Departments
18	19	20	21	22
Slatwall	2	1	Racking	Audio, Electrical
Slatwall	2	1	Racking	Audio, Electrical
Slatwall	2	1	Racking	Audio, Electrical

In the above example, we can see if we want information for UDA-2, we require column 22.

Listing the tables and required fields

It is helpful to list the required tables and fields in a worksheet within the Data Import spreadsheet.



13					
14	TABLE INFORMATION				
15	TABLE		FIELD	COMMENT	
16					
17	AVTTB_BLOCK_GROUP_LINK		**BLG_ID**	Primary/Foreign Key	
18			**BLK_ID**	Primary/Foreign Key	
19					
20					
21	AVTTB_BLOCK_DEFINITION		**BLK_ID**	Primary Key	
22			**BLK_NAME**		
23			BLK_DESC		
24			**BLK_XSIZ**		
25			**BLK_YSIZ**		
26			**BLK_ZSIZ**		
27			**LAY_ID**		
28			**BLT_ID**		
29			MAN_ID		
30			**BLS_ID**		
31			**BLI_ID**		

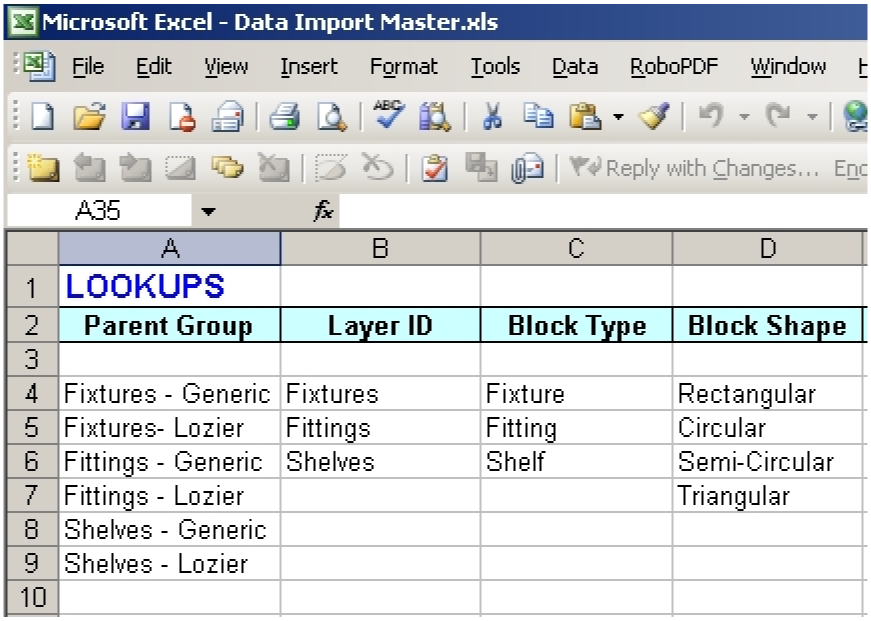
Here, the required tables and fields are identified. Fields that are primary keys are identified, while non-nullable fields are in Bold.

Listing data in this manner will also help if the configuration has to be altered at some point in the future.

Using lookups

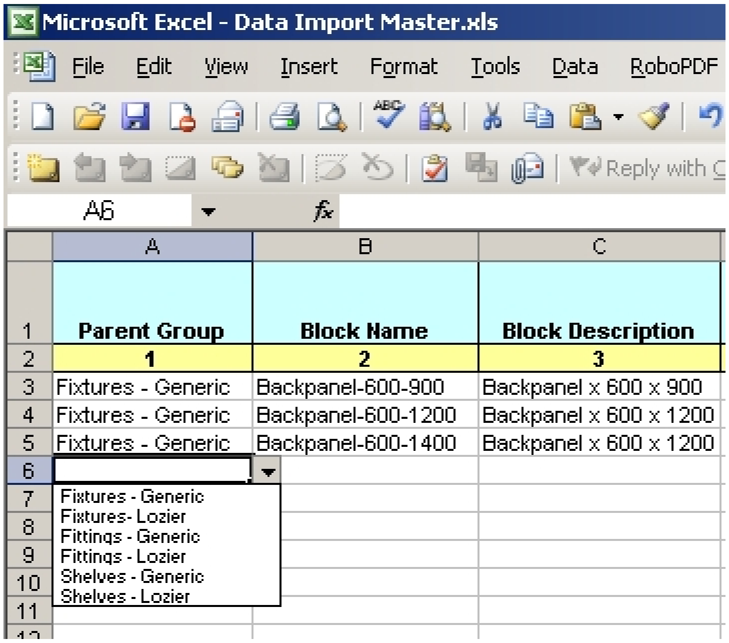
It is sometimes useful to restrict the data that a user can enter to specific values. One way of restricting such values is by using lookups.

Data can be entered in a list on one worksheet within the spreadsheet - in the example below lists of values appropriate to entering Fixture information are entered.



	A	B	C	D
1	LOOKUPS			
2	Parent Group	Layer ID	Block Type	Block Shape
3				
4	Fixtures - Generic	Fixtures	Fixture	Rectangular
5	Fixtures- Lozier	Fittings	Fitting	Circular
6	Fittings - Generic	Shelves	Shelf	Semi-Circular
7	Fittings - Lozier			Triangular
8	Shelves - Generic			
9	Shelves - Lozier			
10				

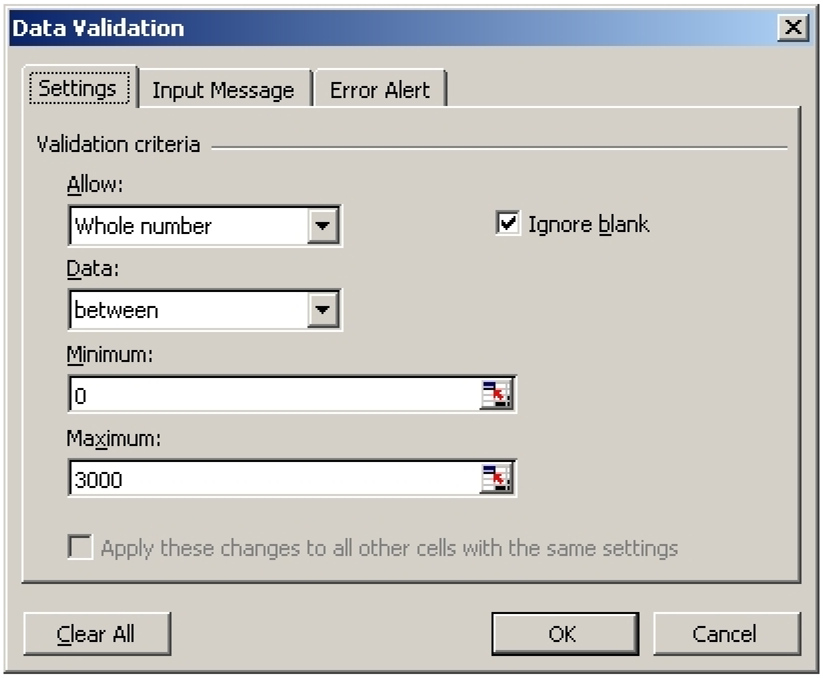
Input in the section of the spreadsheet used for importing data can then be restricted to those values defined in the look ups, only allowing the user to choose valid options.



	A	B	C
1	Parent Group	Block Name	Block Description
2	1	2	3
3	Fixtures - Generic	Backpanel-600-900	Backpanel x 600 x 900
4	Fixtures - Generic	Backpanel-600-1200	Backpanel x 600 x 1200
5	Fixtures - Generic	Backpanel-600-1400	Backpanel x 600 x 1200
6			
7			
8			
9			
10			
11			

Validating Data Input

As well as restricting data to that contained in lookups, it is also possible to restrict data in other ways using data validation.



Validation criteria

Allow: Whole number

☑ Ignore blank

Data: between

Minimum: 0

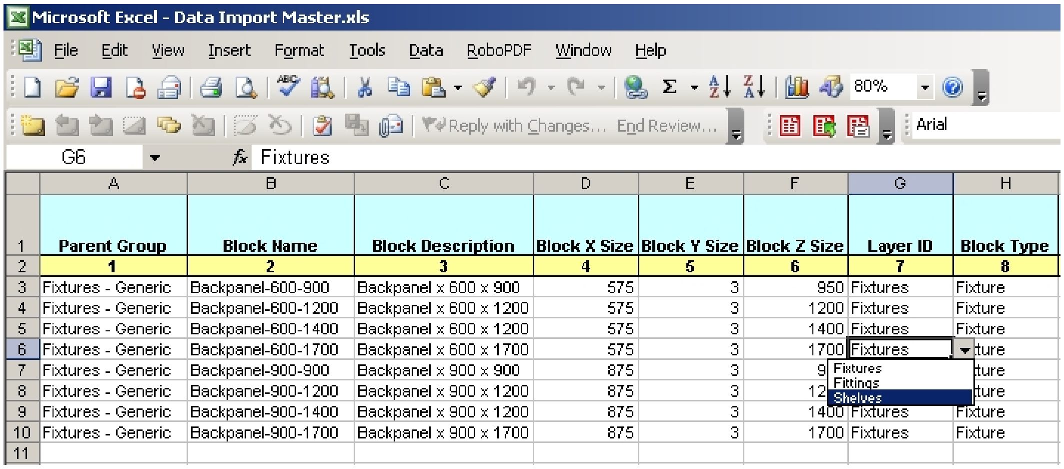
Maximum: 3000

□ Apply these changes to all other cells with the same settings

In the above example, data has been restricted to whole numbers between 0 - 3000. This would be useful in restricting (for example) dimensions to integers between specified values.

The Data Import Spreadsheet - Defining

The **Data Import spreadsheet** can then be set up.



G6 Fixtures

	A	B	C	D	E	F	G	H
1	Parent Group	Block Name	Block Description	Block X Size	Block Y Size	Block Z Size	Layer ID	Block Type
2	1	2	3	4	5	6	7	8
3	Fixtures - Generic	Backpanel-600-900	Backpanel x 600 x 900	575	3	950	Fixtures	Fixture
4	Fixtures - Generic	Backpanel-600-1200	Backpanel x 600 x 1200	575	3	1200	Fixtures	Fixture
5	Fixtures - Generic	Backpanel-600-1400	Backpanel x 600 x 1200	575	3	1400	Fixtures	Fixture
6	Fixtures - Generic	Backpanel-600-1700	Backpanel x 600 x 1700	575	3	1700	Fixtures	ture
7	Fixtures - Generic	Backpanel-900-900	Backpanel x 900 x 900	875	3	9		ture
8	Fixtures - Generic	Backpanel-900-1200	Backpanel x 900 x 1200	875	3	12		ture
9	Fixtures - Generic	Backpanel-900-1400	Backpanel x 900 x 1200	875	3	1400	Fixtures	Fixture
10	Fixtures - Generic	Backpanel-900-1700	Backpanel x 900 x 1700	875	3	1700	Fixtures	Fixture
11								

Fixtures
Fittings
Shelves

The spreadsheet above has been set up with clear column names, and these column names have been arranged in sequence. For example, the earlier columns are for the Block Definition table, with columns associated with the Fixture Attribute come later.

The columns are sequentially numbered to assist in identifying them. This will assist during configuring the import.

Look-ups have been used to restrict data entry in a number of columns - for example the layer ID column is restricted to Fixtures, Fittings and Shelves.

Data entry has also been restricted in other columns. For example the block name column has been restricted to a text length of 32 characters (as per the database), while the X, Y and Z Block Size columns have been restricted to integers between the range 0 - 3000.

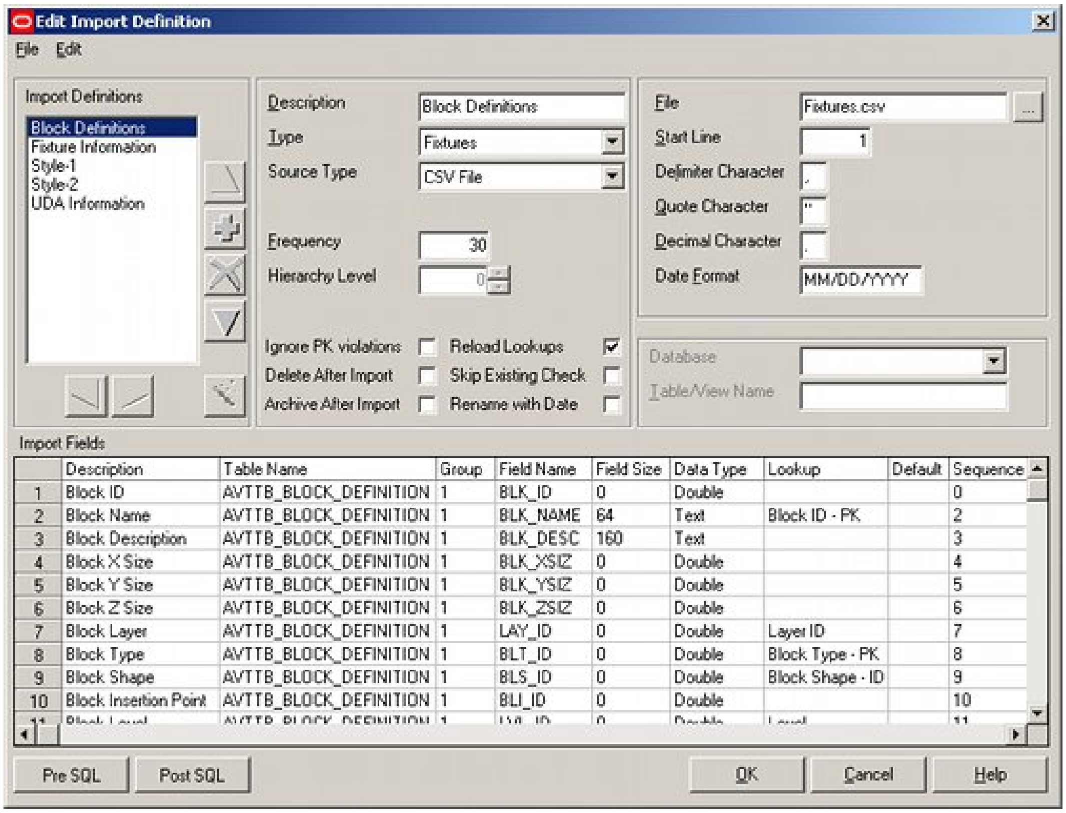
> **Note**: While these measures will not prevent errors in entering data into the spreadsheet, they can substantially cut down on the incidence.

Adding the Fixture Import Definitions

Setting up the Import Definitions

> **Note**: This import will not be covered in as much detail as the Fixture Hierarchy - only screen shots illustrating new facets of Import definitions will be discussed.

Five import definitions are required.



Import Definitions

- Block Definitions
- Fixture Information
- Style-1
- Style-2
- UDA Information

Description: Block Definitions

Type: Fixtures

Source Type: CSV File

Frequency: 30

Hierarchy Level: 0

Ignore PK violations ☐ Reload Lookups ☑

Delete After Import ☐ Skip Existing Check ☐

Archive After Import ☐ Rename with Date ☐

File: Fixtures.csv

Start Line: 1

Delimiter Character: ,

Quote Character: "

Decimal Character: .

Date Format: MM/DD/YYYY

Database:

Table/View Name:

Import Fields

	Description	Table Name	Group	Field Name	Field Size	Data Type	Lookup	Default	Sequence
1	Block ID	AVTTB_BLOCK_DEFINITION	1	BLK_ID	0	Double			0
2	Block Name	AVTTB_BLOCK_DEFINITION	1	BLK_NAME	64	Text	Block ID - PK		2
3	Block Description	AVTTB_BLOCK_DEFINITION	1	BLK_DESC	160	Text			3
4	Block X Size	AVTTB_BLOCK_DEFINITION	1	BLK_XSIZ	0	Double			4
5	Block Y Size	AVTTB_BLOCK_DEFINITION	1	BLK_YSIZ	0	Double			5
6	Block Z Size	AVTTB_BLOCK_DEFINITION	1	BLK_ZSIZ	0	Double			6
7	Block Layer	AVTTB_BLOCK_DEFINITION	1	LAY_ID	0	Double	Layer ID		7
8	Block Type	AVTTB_BLOCK_DEFINITION	1	BLT_ID	0	Double	Block Type - PK		8
9	Block Shape	AVTTB_BLOCK_DEFINITION	1	BLS_ID	0	Double	Block Shape - ID		9
10	Block Insertion Point	AVTTB_BLOCK_DEFINITION	1	BLI_ID	0	Double			10
11	Block Level	AVTTB_BLOCK_DEFINITION	1	LVL_ID	0	Double	Level		11

Block Definition will populate the Block Definitions table (AVTTB_BLOCK_DEFINITIONS) and link with the parent Fixture Group in the AVTTB_FIXTURE_BLOCK_LINK table.

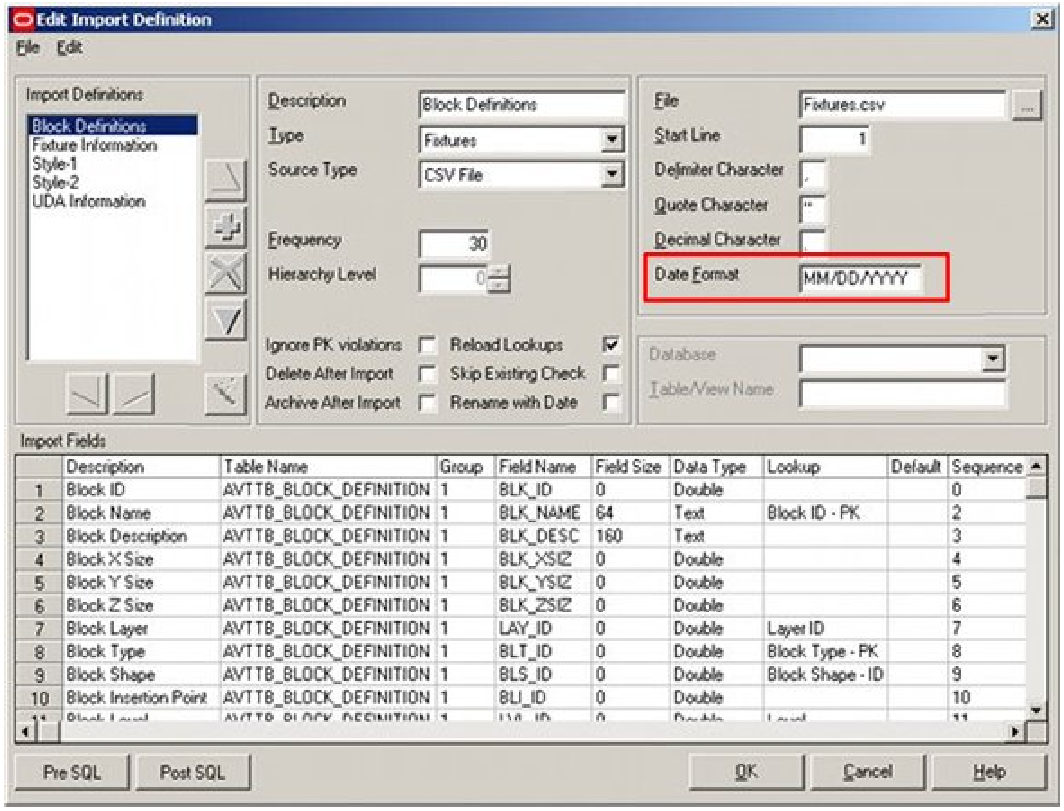
Fixture Information will populate the AVTTB_FIXTURE_INFO table.

Style-1 and Style-2 will populate the AVTTB_STYLE_BLOCK LINK TABLE to define the required styles. We are entering two styles, and using two definitions to do so.

UDA Information will populate the AVTTB_BLOCK_UDA table.

Date Format

The **Date Format** has been set to MM/DD/YYYY.



Import Definitions
- Block Definitions
- Fixture Information
- Style-1
- Style-2
- UDA Information

Description: Block Definitions
Type: Fixtures
Source Type: CSV File
Frequency: 30
Hierarchy Level: 0

Ignore PK violations □ Reload Lookups ☑
Delete After Import □ Skip Existing Check □
Archive After Import □ Rename with Date □

File: Fixtures.csv
Start Line: 1
Delimiter Character: ,
Quote Character: "
Decimal Character: .
Date Format: MM/DD/YYYY

Database:
Table/View Name:

Import Fields

	Description	Table Name	Group	Field Name	Field Size	Data Type	Lookup	Default	Sequence
1	Block ID	AVTTB_BLOCK_DEFINITION	1	BLK_ID	0	Double			0
2	Block Name	AVTTB_BLOCK_DEFINITION	1	BLK_NAME	64	Text	Block ID - PK		2
3	Block Description	AVTTB_BLOCK_DEFINITION	1	BLK_DESC	160	Text			3
4	Block X Size	AVTTB_BLOCK_DEFINITION	1	BLK_XSIZ	0	Double			4
5	Block Y Size	AVTTB_BLOCK_DEFINITION	1	BLK_YSIZ	0	Double			5
6	Block Z Size	AVTTB_BLOCK_DEFINITION	1	BLK_ZSIZ	0	Double			6
7	Block Layer	AVTTB_BLOCK_DEFINITION	1	LAY_ID	0	Double	Layer ID		7
8	Block Type	AVTTB_BLOCK_DEFINITION	1	BLT_ID	0	Double	Block Type - PK		8
9	Block Shape	AVTTB_BLOCK_DEFINITION	1	BLS_ID	0	Double	Block Shape - ID		9
10	Block Insertion Point	AVTTB_BLOCK_DEFINITION	1	BLI_ID	0	Double			10
11	Block Level	AVTTB_BLOCK_DEFINITION	1	LVL_ID	0	Double	Level		11

This is because we are importing date in the US format of Month//Day/Year.

Had we been importing UK dates, the DD/MM/YYYY format would have been required (Day/Month/Year).

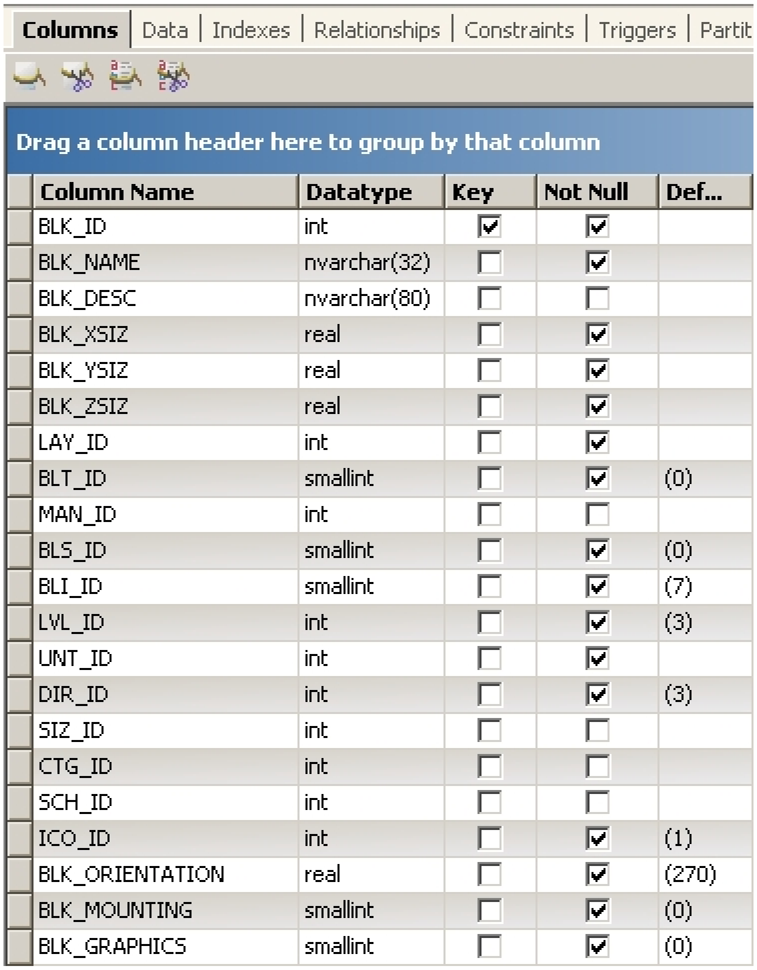
Adding the Block Definitions

Adding Block Definitions

Adding to the AVTTB_BLOCK_DEFINITIONS table (and also adding to the AVTTB_BLOCK_GROUP_LINK table) is the largest import definition in this example.

> **Note**: It would have been perfectly valid to update the AVTTB_BLOCK_GROUP_LINK table using a separate Import Definition - it has been included in this Import Definition to illustrate the use of Groups.

The AVVTB_BLOCK_DEFINITIONS table is a large one:



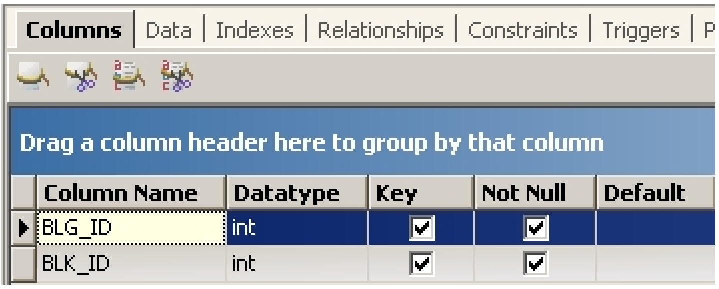
Drag a column header here to group by that column

Column Name	Datatype	Key	Not Null	Def...
BLK_ID	int	☑	☑	
BLK_NAME	nvarchar(32)	□	☑	
BLK_DESC	nvarchar(80)	□	□	
BLK_XSIZ	real	□	☑	
BLK_YSIZ	real	□	☑	
BLK_ZSIZ	real	□	☑	
LAY_ID	int	□	☑	
BLT_ID	smallint	□	☑	(0)
MAN_ID	int	□	□	
BLS_ID	smallint	□	☑	(0)
BLI_ID	smallint	□	☑	(7)
LVL_ID	int	□	☑	(3)
UNT_ID	int	□	☑	
DIR_ID	int	□	☑	(3)
SIZ_ID	int	□	□	
CTG_ID	int	□	□	
SCH_ID	int	□	□	
ICO_ID	int	□	☑	(1)
BLK_ORIENTATION	real	□	☑	(270)
BLK_MOUNTING	smallint	□	☑	(0)
BLK_GRAPHICS	smallint	□	☑	(0)

Note: The above screen shot only shows part of the Block Definition (AVTTB_BLOCK_DEFINITION table).

A significant number of fields in this table will be populated from the spreadsheet, both nullable and non-nullable. Some non-nullable fields will be populated by means of their defaults and will not require importing via Data Importer.

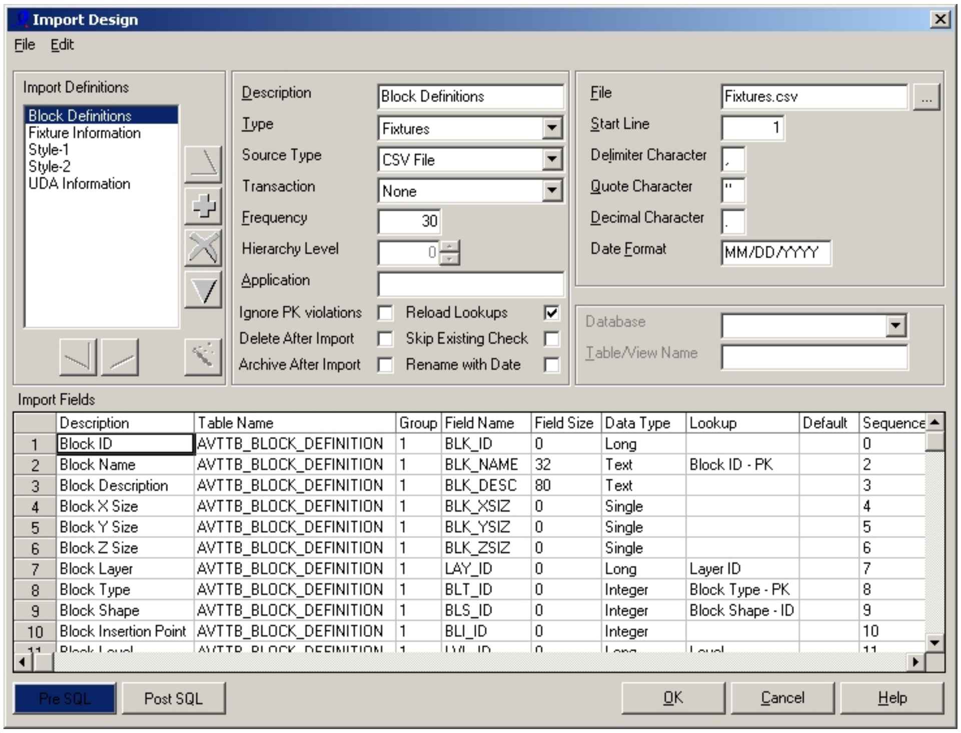
The AVTTB_BLOCK_GROUP_LINK table is a smaller one.



Drag a column header here to group by that column

Column Name	Datatype	Key	Not Null	Default
BLG_ID	int	☑	☑	
BLK_ID	int	☑	☑	

It is then possible to set up the Import Fields:



Import Definitions: Block Definitions, Fixture Information, Style-1, Style-2, UDA Information

Description	Block Definitions
Type	Fixtures
Source Type	CSV File
Transaction	None
Frequency	30
Hierarchy Level	0
Application	
File	Fixtures.csv
Start Line	1
Delimiter Character	,
Quote Character	"
Decimal Character	.
Date Format	MM/DD/YYYY
Database	
Table/View Name	

Ignore PK violations □ Reload Lookups ☑
Delete After Import □ Skip Existing Check □
Archive After Import □ Rename with Date □

Import Fields

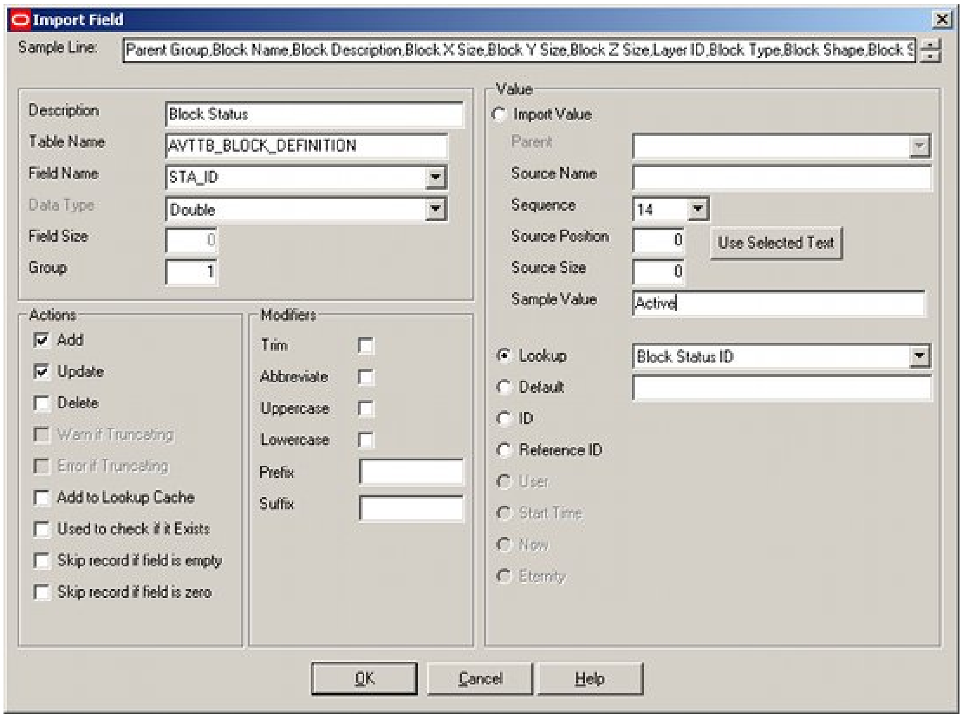
	Description	Table Name	Group	Field Name	Field Size	Data Type	Lookup	Default	Sequence
1	Block ID	AVTTB_BLOCK_DEFINITION	1	BLK_ID	0	Long			0
2	Block Name	AVTTB_BLOCK_DEFINITION	1	BLK_NAME	32	Text	Block ID - PK		2
3	Block Description	AVTTB_BLOCK_DEFINITION	1	BLK_DESC	80	Text			3
4	Block X Size	AVTTB_BLOCK_DEFINITION	1	BLK_XSIZ	0	Single			4
5	Block Y Size	AVTTB_BLOCK_DEFINITION	1	BLK_YSIZ	0	Single			5
6	Block Z Size	AVTTB_BLOCK_DEFINITION	1	BLK_ZSIZ	0	Single			6
7	Block Layer	AVTTB_BLOCK_DEFINITION	1	LAY_ID	0	Long	Layer ID		7
8	Block Type	AVTTB_BLOCK_DEFINITION	1	BLT_ID	0	Integer	Block Type - PK		8
9	Block Shape	AVTTB_BLOCK_DEFINITION	1	BLS_ID	0	Integer	Block Shape - ID		9
10	Block Insertion Point	AVTTB_BLOCK_DEFINITION	1	BLI_ID	0	Integer			10
11	Block Level	AVTTB_BLOCK_DEFINITION	1	LVL_ID	0	Long	Level		11

There are twenty one Import fields to be defined - some of which provide illustrations of how to correctly use Data Importer.

These are discussed in the next few topics.

Configuring the Import Definition - Using SQL in the Lookup

There are **several points of interest** in the Block Definition Import Definition. One of these is in the Block Status Import Field.



Sample Line: Parent Group,Block Name,Block Description,Block X Size,Block Y Size,Block Z Size,Layer ID,Block Type,Block Shape,Block S

Description	Block Status
Table Name	AVTTB_BLOCK_DEFINITION
Field Name	STA_ID
Data Type	Double
Field Size	0
Group	1

Actions: ☑ Add ☑ Update □ Delete □ Warn if Truncating □ Error if Truncating □ Add to Lookup Cache □ Used to check if it Exists □ Skip record if field is empty □ Skip record if field is zero

Modifiers: Trim □ Abbreviate □ Uppercase □ Lowercase □ Prefix Suffix

Value:

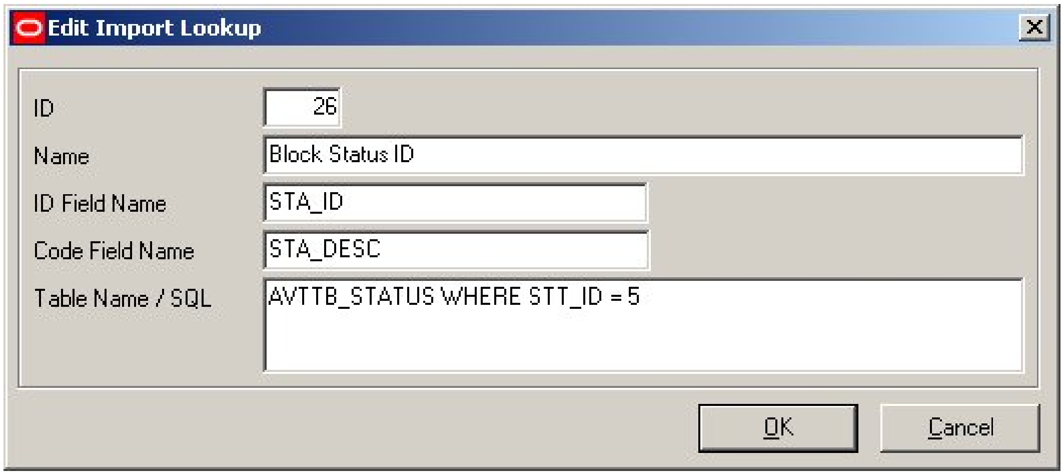
○ Import Value

Parent	
Source Name	
Sequence	14
Source Position	0
Source Size	0
Sample Value	Active

● Lookup: Block Status ID
○ Default
○ ID
○ Reference ID
○ User
○ Start Time
○ Now
○ Eternity

Lookup using SQL

One requirement is to look up the Status for the block and convert it to a Status ID (STA_ID).

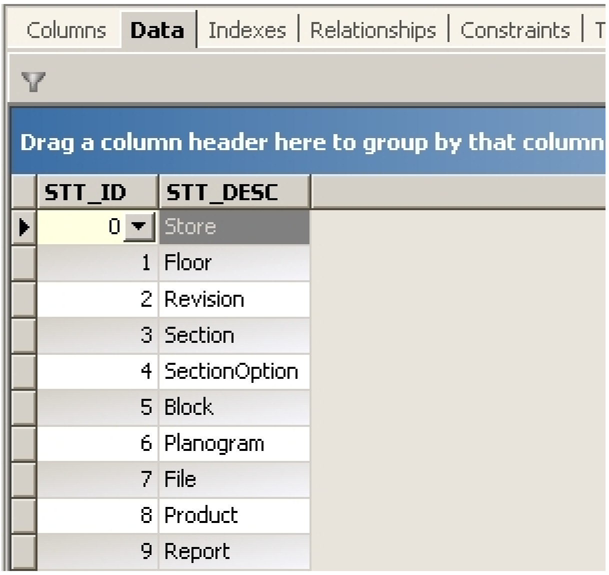


ID	26
Name	Block Status ID
ID Field Name	STA_ID
Code Field Name	STA_DESC
Table Name / SQL	AVTTB_STATUS WHERE STT_ID = 5

The status will be read in from the .csv file as a text string - for example Current.

However, there are a number of status types so there may be several types of object that could have Current status.

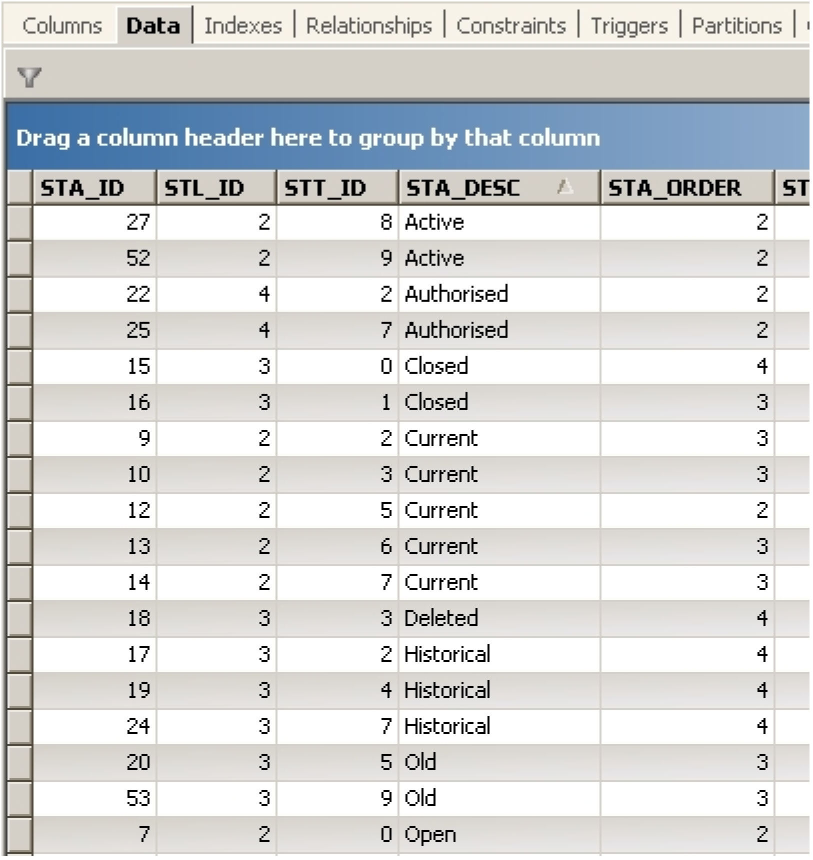
Accordingly, statuses have been assigned to Status Types - found in the AVTTB_STATUS_TYPE table.



STT_ID	STT_DESC
0	Store
1	Floor
2	Revision
3	Section
4	SectionOption
5	Block
6	Planogram
7	File
8	Product
9	Report

We can see blocks have been assigned a Status Type (STT_ID) of 5.

Looking at the AVTTB_STATUS table shows us that there are indeed several objects that can have a Current status assigned to them.



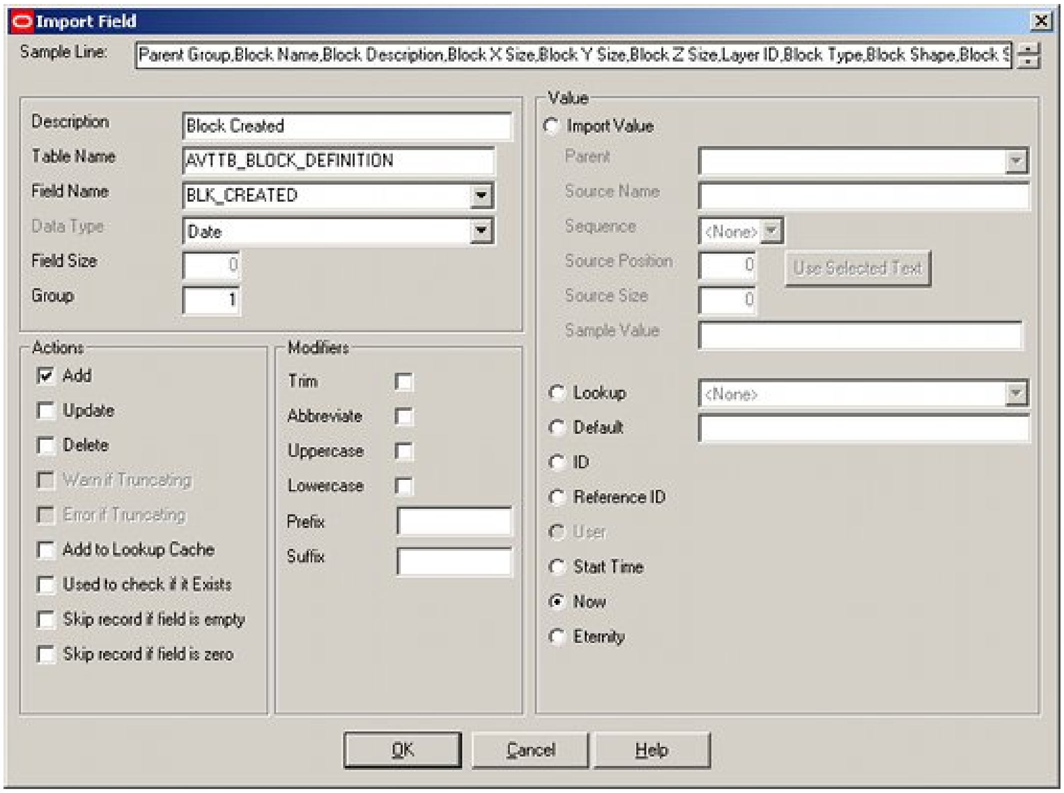
STA_ID	STL_ID	STT_ID	STA_DESC	STA_ORDER	ST
27	2	8	Active	2	
52	2	9	Active	2	
22	4	2	Authorised	2	
25	4	7	Authorised	2	
15	3	0	Closed	4	
16	3	1	Closed	3	
9	2	2	Current	3	
10	2	3	Current	3	
12	2	5	Current	2	
13	2	6	Current	3	
14	2	7	Current	3	
18	3	3	Deleted	4	
17	3	2	Historical	4	
19	3	4	Historical	4	
24	3	7	Historical	4	
20	3	5	Old	3	
53	3	9	Old	3	
7	2	0	Open	2	

Accordingly, the Lookup looks in the AVTTB_STATUS table WHERE STT_ID = 5 and returns the appropriate Status ID (STA_ID).

This restricts the lookup to statuses associated with blocks, and prevents the lookup returning values associated with other objects such as Floors or Files.

Configuring the Import Definition - Using Dates

Using **Dates** requires some care.



Import Field

Sample Line: Parent Group,Block Name,Block Description,Block X Size,Block Y Size,Block Z Size,Layer ID,Block Type,Block Shape,Block S

Description: Block Created
Table Name: AVTTB_BLOCK_DEFINITION
Field Name: BLK_CREATED
Data Type: Date
Field Size: 0
Group: 1

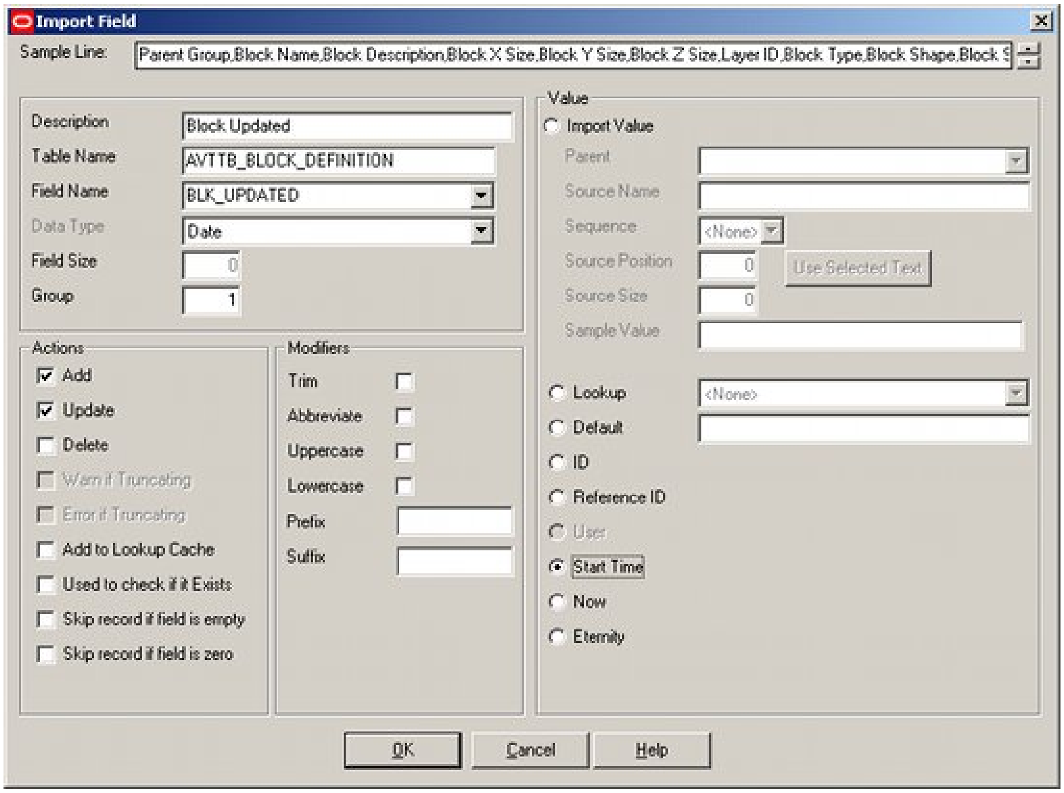
Actions
☑ Add
□ Update
□ Delete
□ Warn if Truncating
□ Error if Truncating
□ Add to Lookup Cache
□ Used to check if it Exists
□ Skip record if field is empty
□ Skip record if field is zero

Modifiers
Trim □
Abbreviate □
Uppercase □
Lowercase □
Prefix:
Suffix:

Value
○ Import Value
Parent:
Source Name:
Sequence: <None>
Source Position: 0
Source Size: 0
Sample Value:
○ Lookup <None>
○ Default
○ ID
○ Reference ID
○ User
○ Start Time
● Now
○ Eternity

In this example, we are adding a Block Created date (BLK_CREATED).

As it is the creation date, we only select the Add option. We do not wish this field to change if the record is updated, so the Update option is left unchecked.



Import Field

Sample Line: Parent Group,Block Name,Block Description,Block X Size,Block Y Size,Block Z Size,Layer ID,Block Type,Block Shape,Block S

Description: Block Updated
Table Name: AVTTB_BLOCK_DEFINITION
Field Name: BLK_UPDATED
Data Type: Date
Field Size: 0
Group: 1

Actions
☑ Add
☑ Update
□ Delete
□ Warn if Truncating
□ Error if Truncating
□ Add to Lookup Cache
□ Used to check if it Exists
□ Skip record if field is empty
□ Skip record if field is zero

Modifiers
Trim □
Abbreviate □
Uppercase □
Lowercase □
Prefix:
Suffix:

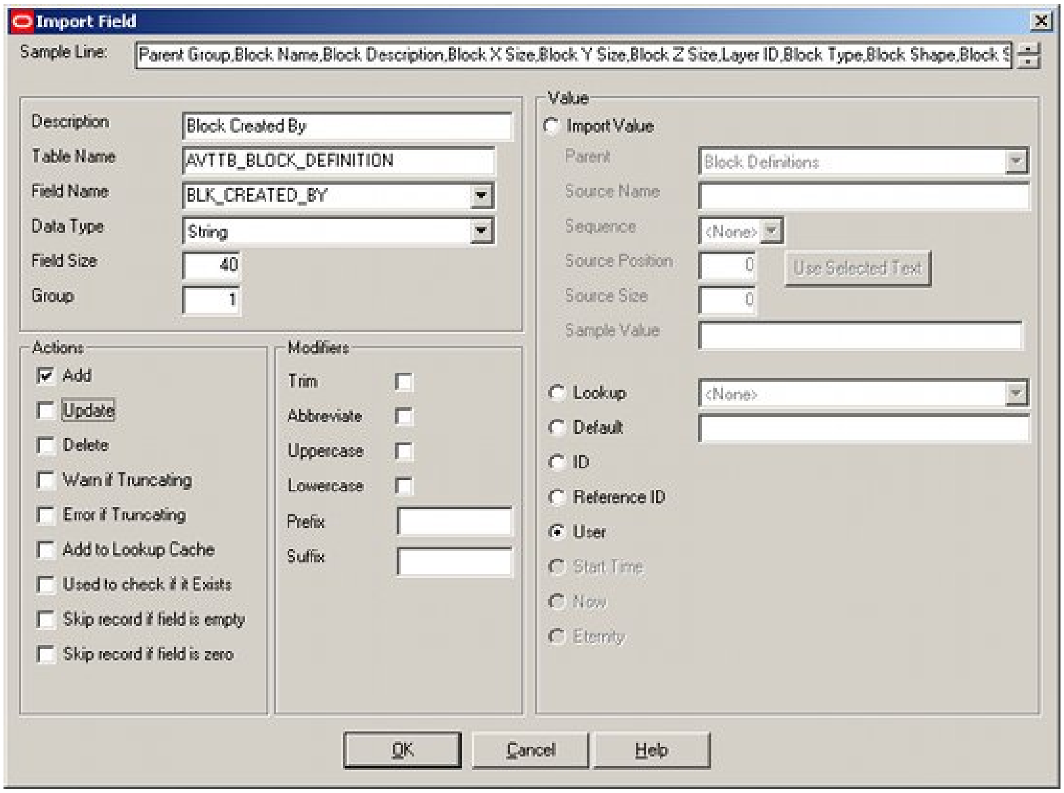
Value
○ Import Value
Parent:
Source Name:
Sequence: <None>
Source Position: 0
Source Size: 0
Sample Value:
○ Lookup <None>
○ Default
○ ID
○ Reference ID
○ User
● Start Time
○ Now
○ Eternity

For the Block Updated field (BLK_UPDATED) we would select both Add and Update. This is because if we are adding a new record, we wish the block to have the Block Updated field set to the same time as the Block Created field if the record is being added for the first time. If we are updating the record we wish the Block Updated field to be updated to the time the record was modified.

There are two options for time - Start Time and Now. Start Time will result in the record being given the date and time the import process started, now will result in the record being given the date and time the specific record was actually imported.

Configuring the Import Definition - Using Users

The **name of the user** importing the record can be appended to the record using the User option.



Import Field

Sample Line: Parent Group,Block Name,Block Description,Block X Size,Block Y Size,Block Z Size,Layer ID,Block Type,Block Shape,Block S

Description: Block Created By
Table Name: AVTTB_BLOCK_DEFINITION
Field Name: BLK_CREATED_BY
Data Type: String
Field Size: 40
Group: 1

Actions
- ☑ Add
- □ Update
- □ Delete
- □ Warn if Truncating
- □ Error if Truncating
- □ Add to Lookup Cache
- □ Used to check if it Exists
- □ Skip record if field is empty
- □ Skip record if field is zero

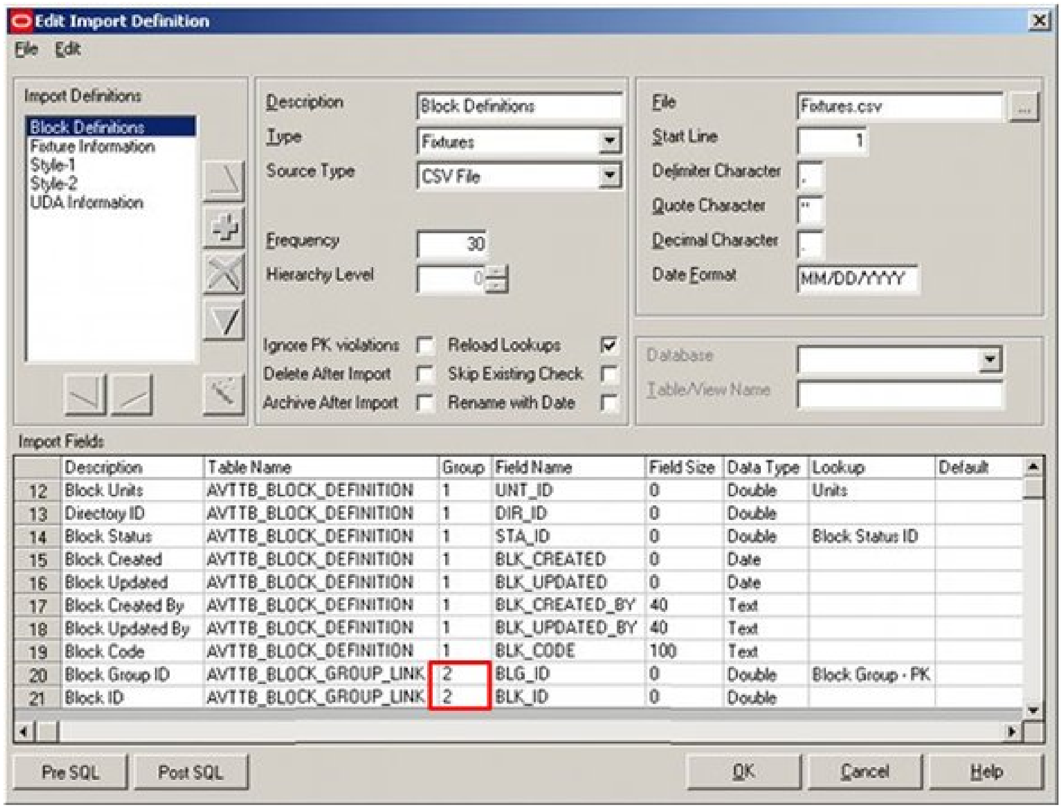
Modifiers
- Trim □
- Abbreviate □
- Uppercase □
- Lowercase □
- Prefix:
- Suffix:

Value
- ○ Import Value
  - Parent: Block Definitions
  - Source Name:
  - Sequence: <None>
  - Source Position: 0
  - Source Size: 0
  - Sample Value:
- ○ Lookup: <None>
- ○ Default:
- ○ ID
- ○ Reference ID
- ● User
- ○ Start Time
- ○ Now
- ○ Eternity

This will import the user name of the person carrying out the import into the pertinent table in the database.

Configuring the Import Definition - Using Groups

The **Block Definitions** Import definition provides an example of using Groups.



Import Definitions: Block Definitions, Fixture Information, Style-1, Style-2, UDA Information

Description: Block Definitions

Type: Fixtures

Source Type: CSV File

Frequency: 30

Hierarchy Level: 0

Ignore PK violations □ Reload Lookups ☑

Delete After Import □ Skip Existing Check □

Archive After Import □ Rename with Date □

File: Fixtures.csv

Start Line: 1

Delimiter Character: ,

Quote Character: "

Decimal Character: .

Date Format: MM/DD/YYYY

Database:

Table/View Name:

Import Fields

	Description	Table Name	Group	Field Name	Field Size	Data Type	Lookup	Default
12	Block Units	AVTTB_BLOCK_DEFINITION	1	UNT_ID	0	Double	Units	
13	Directory ID	AVTTB_BLOCK_DEFINITION	1	DIR_ID	0	Double		
14	Block Status	AVTTB_BLOCK_DEFINITION	1	STA_ID	0	Double	Block Status ID	
15	Block Created	AVTTB_BLOCK_DEFINITION	1	BLK_CREATED	0	Date		
16	Block Updated	AVTTB_BLOCK_DEFINITION	1	BLK_UPDATED	0	Date		
17	Block Created By	AVTTB_BLOCK_DEFINITION	1	BLK_CREATED_BY	40	Text		
18	Block Updated By	AVTTB_BLOCK_DEFINITION	1	BLK_UPDATED_BY	40	Text		
19	Block Code	AVTTB_BLOCK_DEFINITION	1	BLK_CODE	100	Text		
20	Block Group ID	AVTTB_BLOCK_GROUP_LINK	2	BLG_ID	0	Double	Block Group - PK	
21	Block ID	AVTTB_BLOCK_GROUP_LINK	2	BLK_ID	0	Double		

Towards the bottom of the list of definitions, the designated group has been changed from 1 to 2. This is because the table the data is being inserted into has changed from AVTTB_BLOCK_DEFINITION to AVTTB_BLOCK_GROUP_LINK - each table for which data is being inserted into within a particular Import Definition must be assigned to a specific group.

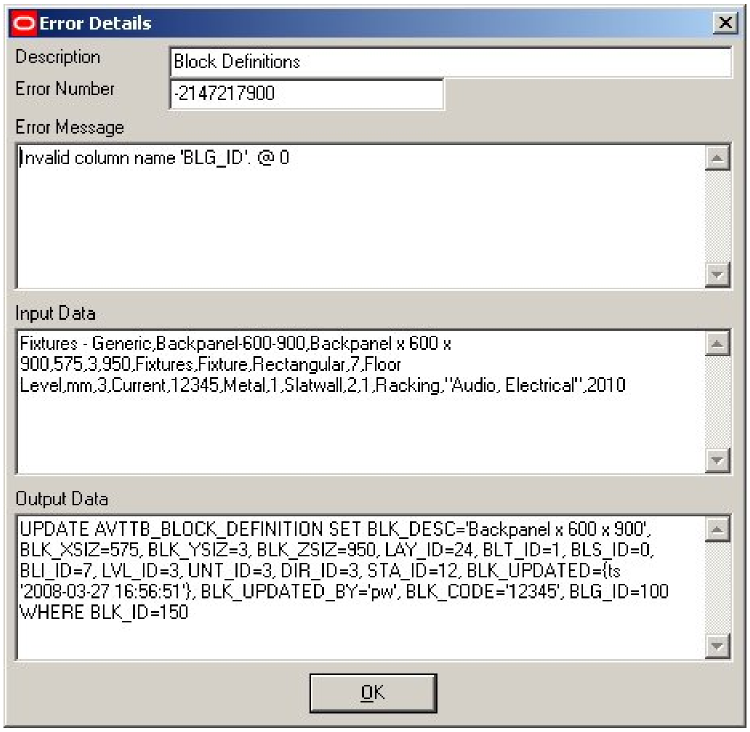
Data Importer works by creating SQL Update statements which are then run against the database.

Because data is to be inserted into two different tables, two different SQL Update statements will be produced. Using Groups ensures that each update statement contains the data pertinent to the table the data will be inserted into.

Example of Errors: An error was deliberately created by changing the Group for the Block Group ID from 2 to 1.

Data Importer then tried to insert the data into AVTTB_BLOCK_DEFINITION table (Group 1) rather than the AVTTB_BLOCK_GROUP_LINK table (group 2).

This produced the following error:



Description: Block Definitions

Error Number: -2147217900

Error Message:

```
Invalid column name 'BLG_ID'. @ 0
```

Input Data:

```
Fixtures - Generic,Backpanel-600-900,Backpanel x 600 x
900,575,3,950,Fixtures,Fixture,Rectangular,7,Floor
Level,mm,3,Current,12345,Metal,1,Slatwall,2,1,Racking,"Audio, Electrical",2010
```

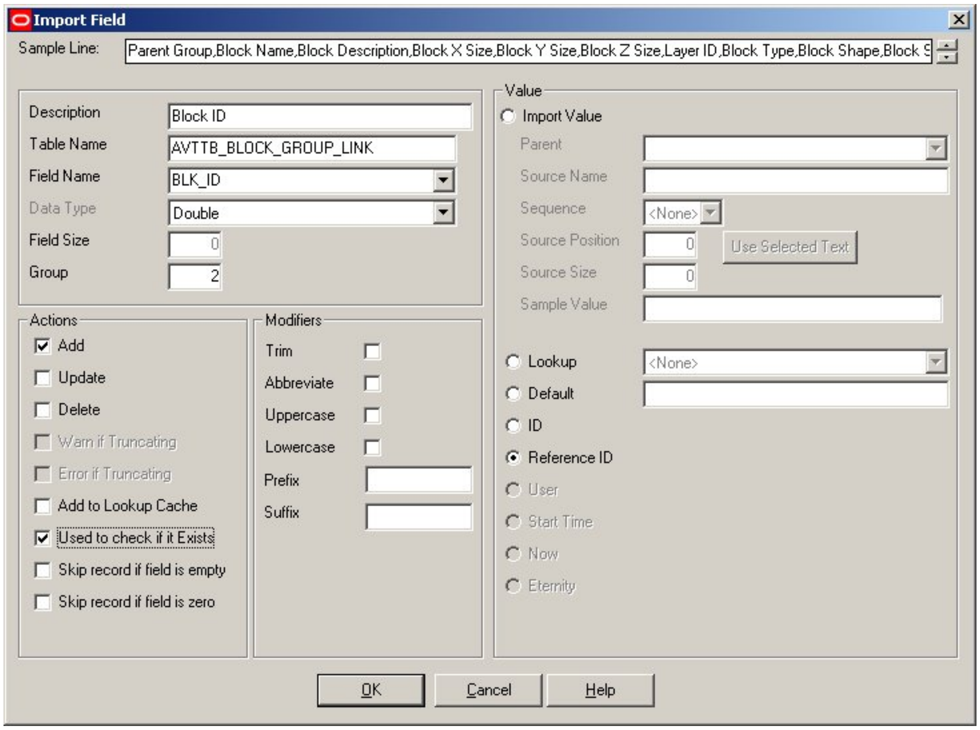
Output Data:

```
UPDATE AVTTB_BLOCK_DEFINITION SET BLK_DESC='Backpanel x 600 x 900',
BLK_XSIZ=575, BLK_YSIZ=3, BLK_ZSIZ=950, LAY_ID=24, BLT_ID=1, BLS_ID=0,
BLI_ID=7, LVL_ID=3, UNT_ID=3, DIR_ID=3, STA_ID=12, BLK_UPDATED={ts
'2008-03-27 16:56:51'}, BLK_UPDATED_BY='pw', BLK_CODE='12345', BLG_ID=100
WHERE BLK_ID=150
```

(Data Importer tried to insert data into a BLG_ID column in the AVTT_BLOCK_DEFINITION table - and that column only exists in the AVTTB_BLOCK_GROUP_LINK table).

Configuring the Import Definition - Reference ID's

Reference ID's must be used when an ID has previously been looked up in a specific Import Definition.



Sample Line: Parent Group,Block Name,Block Description,Block X Size,Block Y Size,Block Z Size,Layer ID,Block Type,Block Shape,Block S

Description	Block ID
Table Name	AVTTB_BLOCK_GROUP_LINK
Field Name	BLK_ID
Data Type	Double
Field Size	0
Group	2

Actions

- ☑ Add
- □ Update
- □ Delete
- □ Warn if Truncating
- □ Error if Truncating
- □ Add to Lookup Cache
- ☑ Used to check if it Exists
- □ Skip record if field is empty
- □ Skip record if field is zero

Modifiers

- Trim □
- Abbreviate □
- Uppercase □
- Lowercase □
- Prefix
- Suffix

Value

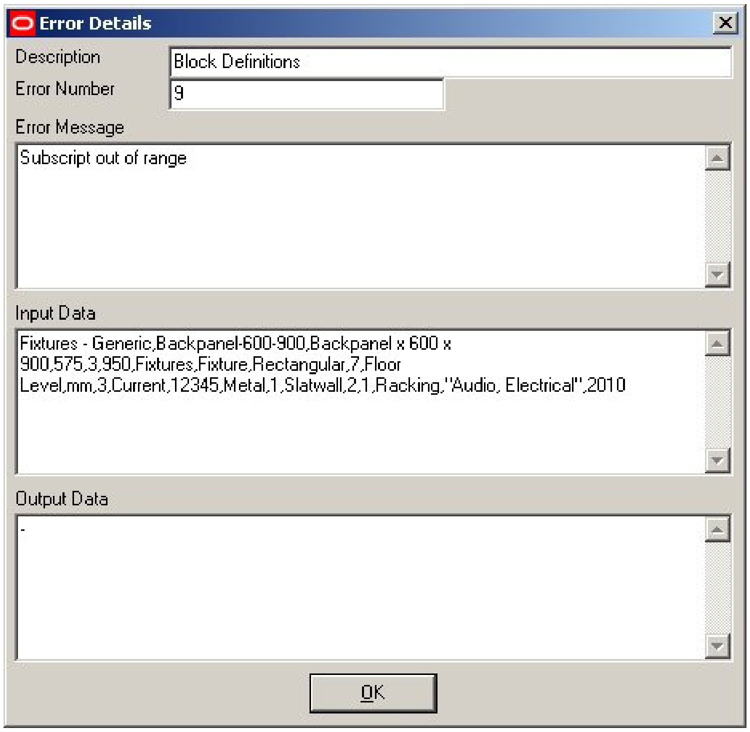
- ○ Import Value
  - Parent
  - Source Name
  - Sequence: <None>
  - Source Position: 0
  - Source Size: 0
  - Sample Value
- ○ Lookup: <None>
- ○ Default
- ○ ID
- ● Reference ID
- ○ User
- ○ Start Time
- ○ Now
- ○ Eternity

In the example above we have previously used a lookup to convert a Block Name (BLK_NAME) into a Block ID (BLK_ID).

This information is already available in this definition and using the lookup for a second time within the Block Definitions import definition will cause an error.

(If it is necessary to use a lookup for Block name in another import definition [for example Fixture Information] then it may be used again for the first timer in that import definition, with a Reference ID being required for the second time of use in that import definition).

> **Note**: A typical error message that might result from using a lookup for the second time instead of a Reference ID might look as follows:



Description: Block Definitions

Error Number: 9

Error Message:

```
Subscript out of range
```

Input Data:

```
Fixtures - Generic,Backpanel-600-900,Backpanel x 600 x
900,575,3,950,Fixtures,Fixture,Rectangular,7,Floor
Level,mm,3,Current,12345,Metal,1,Slatwall,2,1,Racking,"Audio, Electrical",2010
```

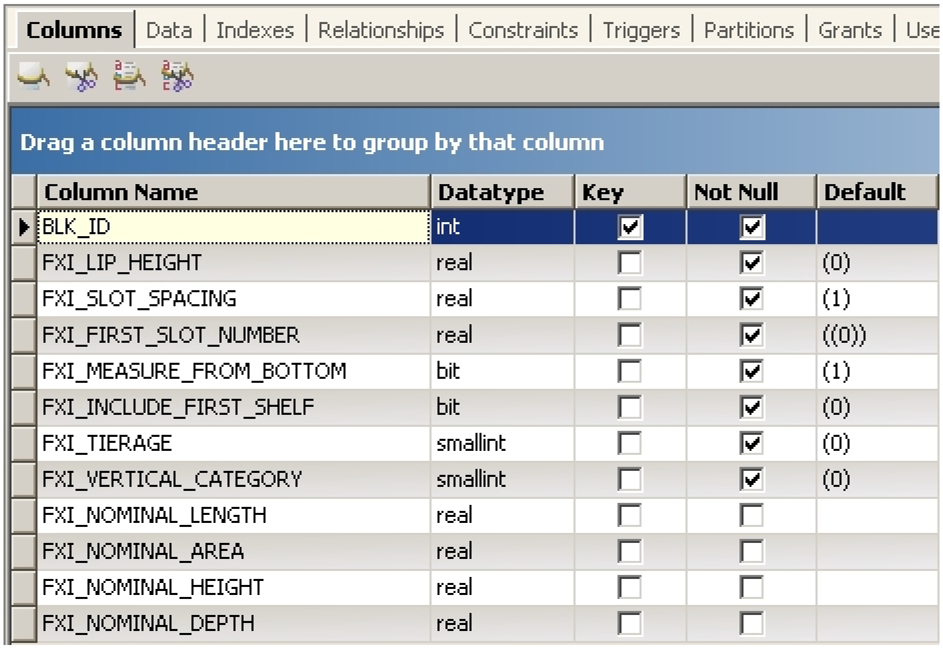
Output Data:

```
-
```

Adding the Fixture Information

Adding Fixture Information

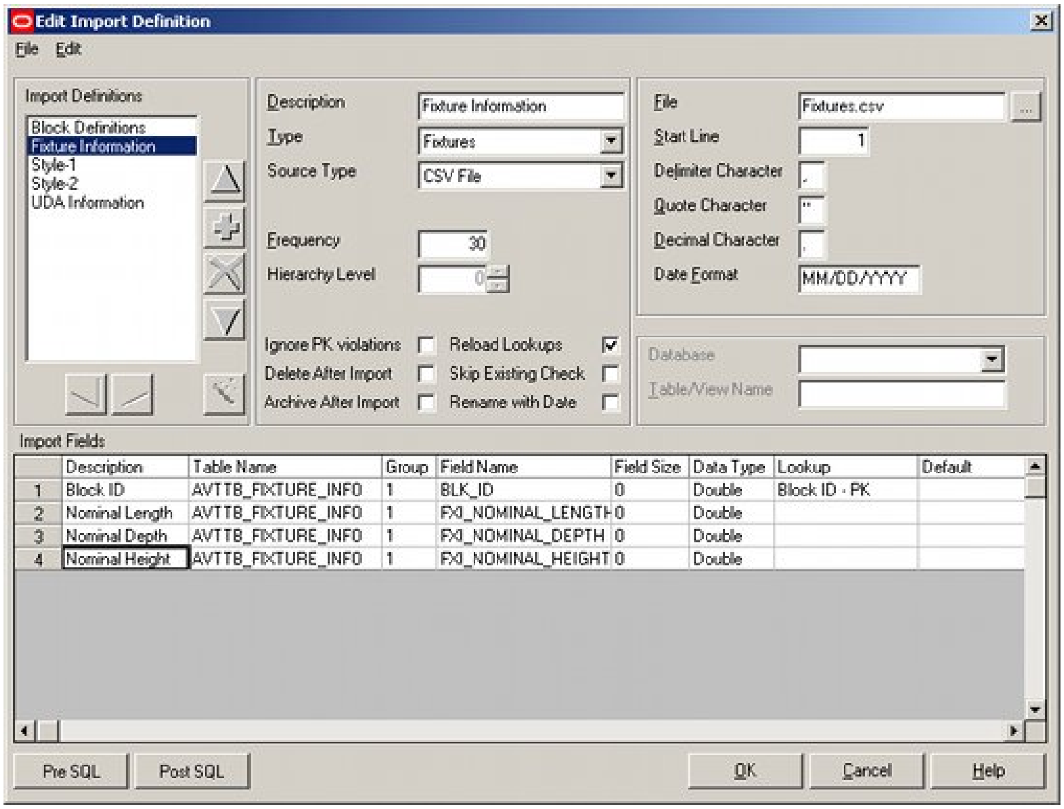
There are a number of columns in the AVTTB_FIXTURE_INFO table:



Columns | Data | Indexes | Relationships | Constraints | Triggers | Partitions | Grants | Use
Drag a column header here to group by that column

Column Name	Datatype	Key	Not Null	Default
BLK_ID	int	☑	☑	
FXI_LIP_HEIGHT	real	□	☑	(0)
FXI_SLOT_SPACING	real	□	☑	(1)
FXI_FIRST_SLOT_NUMBER	real	□	☑	((0))
FXI_MEASURE_FROM_BOTTOM	bit	□	☑	(1)
FXI_INCLUDE_FIRST_SHELF	bit	□	☑	(0)
FXI_TIERAGE	smallint	□	☑	(0)
FXI_VERTICAL_CATEGORY	smallint	□	☑	(0)
FXI_NOMINAL_LENGTH	real	□	□	
FXI_NOMINAL_AREA	real	□	□	
FXI_NOMINAL_HEIGHT	real	□	□	
FXI_NOMINAL_DEPTH	real	□	□	

However, Data Importer is only going to be used to insert values into four of them, the other columns having default values that will be used.



Import Definitions: Block Definitions, Fixture Information, Style-1, Style-2, UDA Information

Description: Fixture Information
Type: Fixtures
Source Type: CSV File
Frequency: 30
Hierarchy Level: 0

Ignore PK violations □ Reload Lookups ☑
Delete After Import □ Skip Existing Check □
Archive After Import □ Rename with Date □

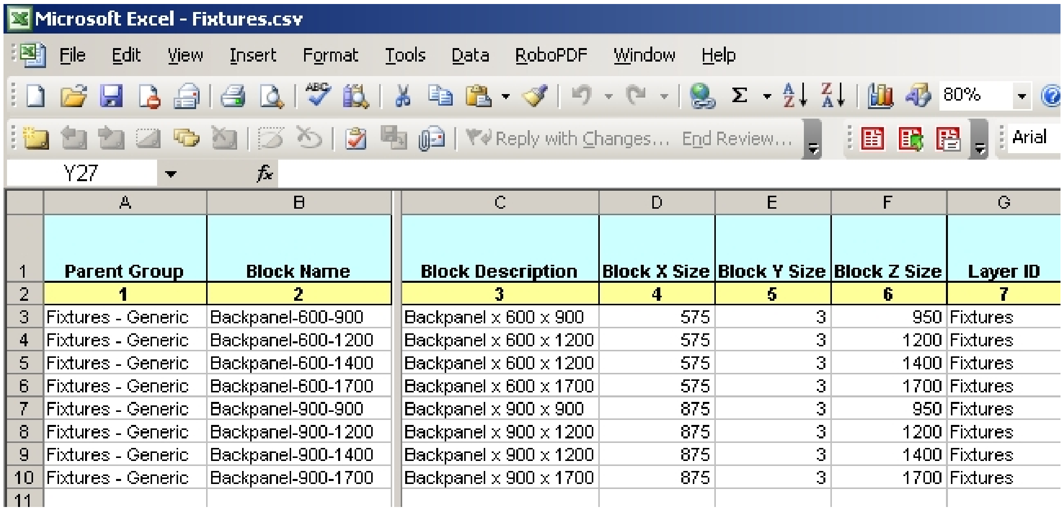
File: Fixtures.csv
Start Line: 1
Delimiter Character: ,
Quote Character: "
Decimal Character: .
Date Format: MM/DD/YYYY

Database:
Table/View Name:

Import Fields

	Description	Table Name	Group	Field Name	Field Size	Data Type	Lookup	Default
1	Block ID	AVTTB_FIXTURE_INFO	1	BLK_ID	0	Double	Block ID - PK	
2	Nominal Length	AVTTB_FIXTURE_INFO	1	FXI_NOMINAL_LENGTH	0	Double		
3	Nominal Depth	AVTTB_FIXTURE_INFO	1	FXI_NOMINAL_DEPTH	0	Double		
4	Nominal Height	AVTTB_FIXTURE_INFO	1	FXI_NOMINAL_HEIGHT	0	Double		

In this instance, it has been chosen to insert values from columns 4, 5 and 6, duplicating the actual dimensions.


Microsoft Excel - Fixtures.csv
File Edit View Insert Format Tools Data RoboPDF Window Help
80%
Reply with Changes... End Review...
Arial
Y27
fx

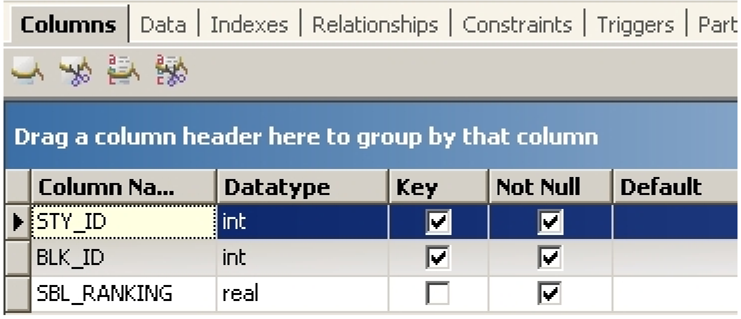
	A	B	C	D	E	F	G
1	Parent Group	Block Name	Block Description	Block X Size	Block Y Size	Block Z Size	Layer ID
2	1	2	3	4	5	6	7
3	Fixtures - Generic	Backpanel-600-900	Backpanel x 600 x 900	575	3	950	Fixtures
4	Fixtures - Generic	Backpanel-600-1200	Backpanel x 600 x 1200	575	3	1200	Fixtures
5	Fixtures - Generic	Backpanel-600-1400	Backpanel x 600 x 1200	575	3	1400	Fixtures
6	Fixtures - Generic	Backpanel-600-1700	Backpanel x 600 x 1700	575	3	1700	Fixtures
7	Fixtures - Generic	Backpanel-900-900	Backpanel x 900 x 900	875	3	950	Fixtures
8	Fixtures - Generic	Backpanel-900-1200	Backpanel x 900 x 1200	875	3	1200	Fixtures
9	Fixtures - Generic	Backpanel-900-1400	Backpanel x 900 x 1200	875	3	1400	Fixtures
10	Fixtures - Generic	Backpanel-900-1700	Backpanel x 900 x 1700	875	3	1700	Fixtures
11							

Adding the Style Relationships

Adding Styles

Adding **Styles** is relatively straight forward.

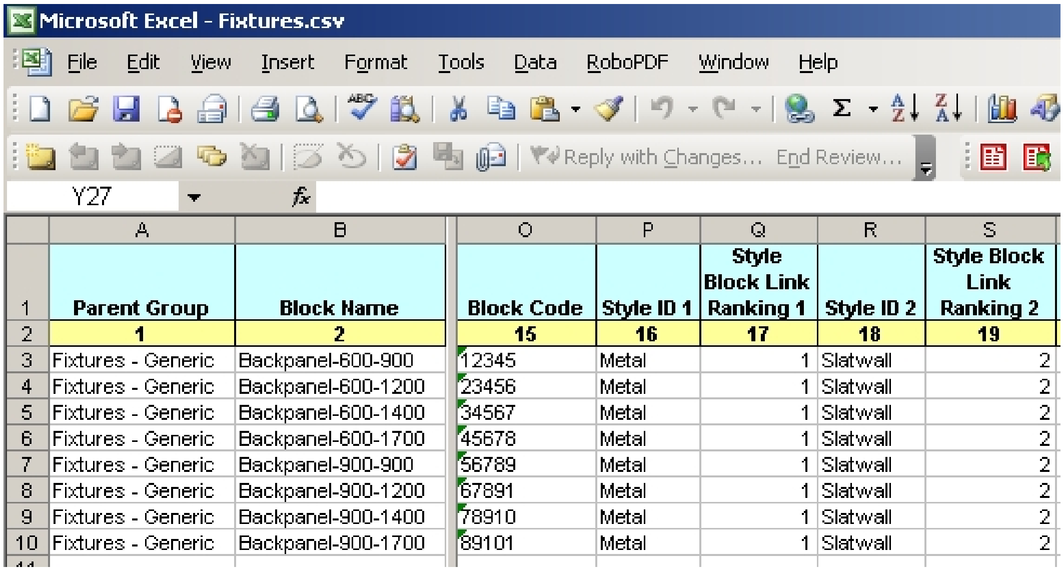
The **STYLE_BLOCK_LINK** table contains just three columns:



Column Na...	Datatype	Key	Not Null	Default
STY_ID	int	☑	☑	
BLK_ID	int	☑	☑	
SBL_RANKING	real	☐	☑	

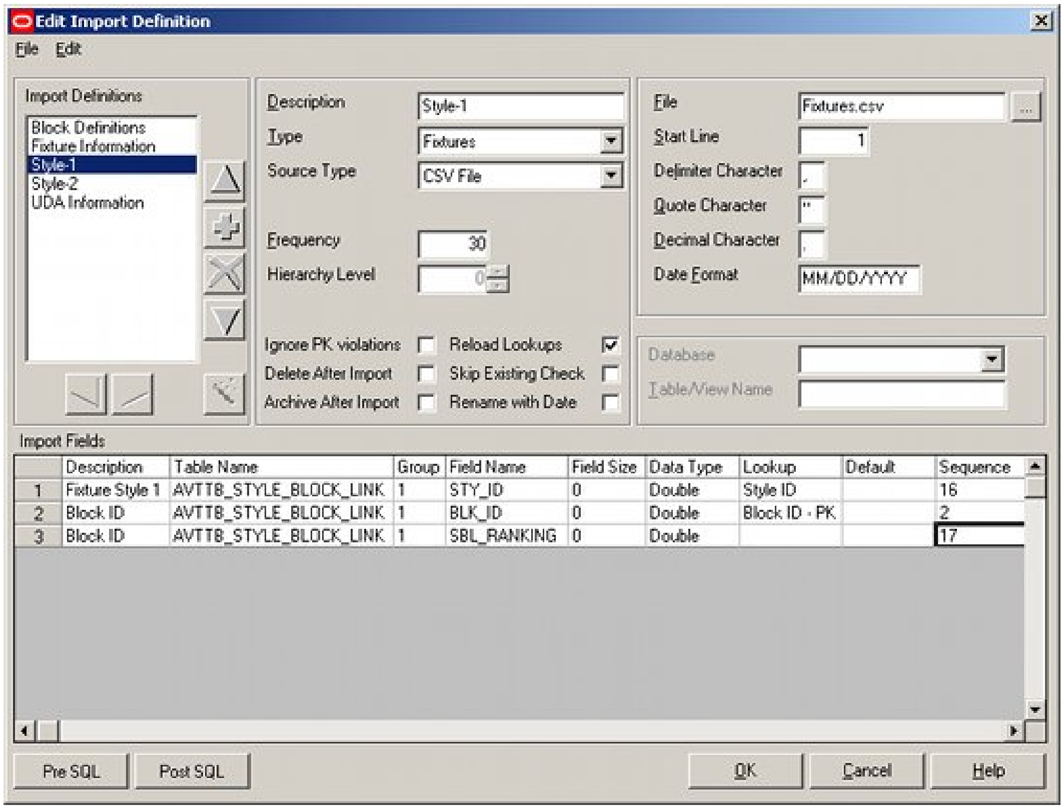
However, it is possible to insert many styles for a specific fixture, each forming a record in the table.

The spreadsheet is configured to add two styles for each block (columns 16 - 19).


Microsoft Excel - Fixtures.csv
File Edit View Insert Format Tools Data RoboPDF Window Help
Reply with Changes... End Review...
Y27
fx

	A	B	O	P	Q	R	S
1	Parent Group	Block Name	Block Code	Style ID 1	Style Block Link Ranking 1	Style ID 2	Style Block Link Ranking 2
2	1	2	15	16	17	18	19
3	Fixtures - Generic	Backpanel-600-900	12345	Metal	1	Slatwall	2
4	Fixtures - Generic	Backpanel-600-1200	23456	Metal	1	Slatwall	2
5	Fixtures - Generic	Backpanel-600-1400	34567	Metal	1	Slatwall	2
6	Fixtures - Generic	Backpanel-600-1700	45678	Metal	1	Slatwall	2
7	Fixtures - Generic	Backpanel-900-900	56789	Metal	1	Slatwall	2
8	Fixtures - Generic	Backpanel-900-1200	67891	Metal	1	Slatwall	2
9	Fixtures - Generic	Backpanel-900-1400	78910	Metal	1	Slatwall	2
10	Fixtures - Generic	Backpanel-900-1700	89101	Metal	1	Slatwall	2

Because we are inserting two separate records for each fixture, we require two separate import definitions, Style-1 and Style-2.



Description: Style-1
Type: Fixtures
Source Type: CSV File
Frequency: 30
Hierarchy Level: 0

File: Fixtures.csv
Start Line: 1
Delimiter Character: ,
Quote Character: "
Decimal Character: .
Date Format: MM/DD/YYYY

Database:
Table/View Name:

□ Ignore PK violations
☑ Reload Lookups
□ Delete After Import
□ Skip Existing Check
□ Archive After Import
□ Rename with Date

Import Fields

	Description	Table Name	Group	Field Name	Field Size	Data Type	Lookup	Default	Sequence
1	Fixture Style 1	AVTTB_STYLE_BLOCK_LINK	1	STY_ID	0	Double	Style ID		16
2	Block ID	AVTTB_STYLE_BLOCK_LINK	1	BLK_ID	0	Double	Block ID - PK		2
3	Block ID	AVTTB_STYLE_BLOCK_LINK	1	SBL_RANKING	0	Double			17

Each is straightforward, using a lookup to covert the block names and style names to their respective ID's, and bringing in a value from the spreadsheet.

> **Note**: It is possible to set up the spreadsheet for many styles - say five per fixture.

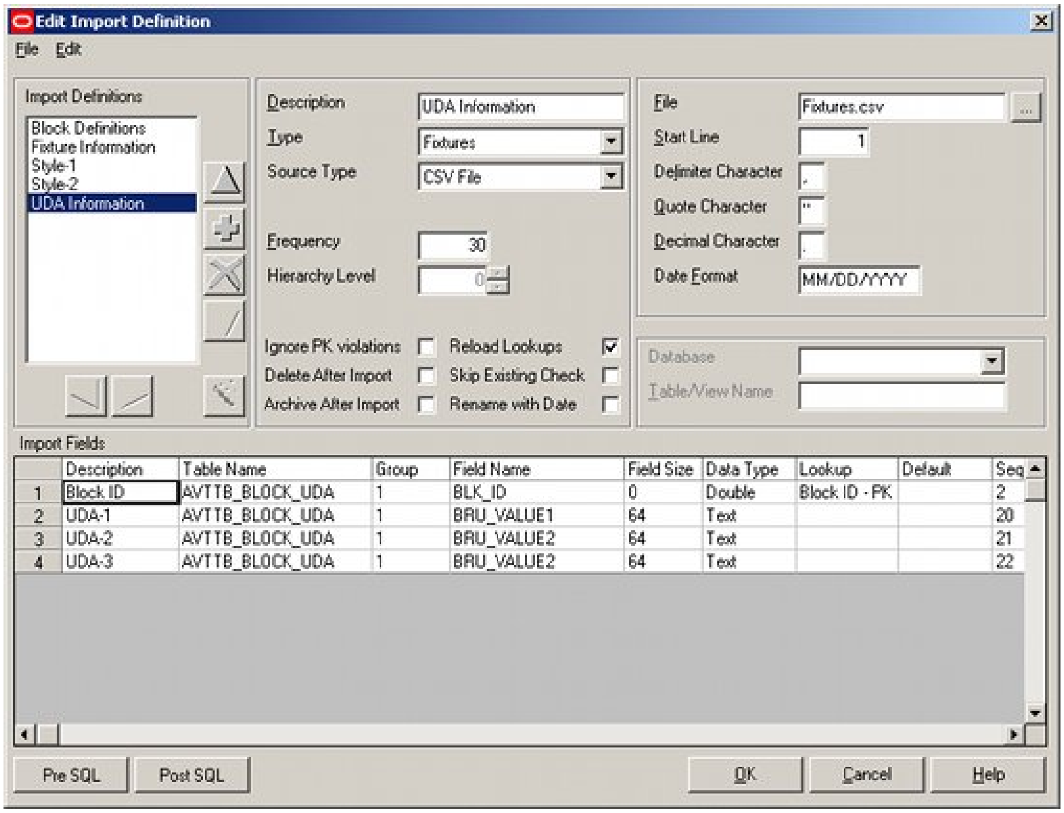
Five separate Style import definitions could then be set up (Style-1, Style-2, Style-3, etc).

It would then be possible to use the Skip record if field is empty option in the Import Field definition to ignore records for fixtures with less than the maximum number of styles defined. For example, if a fixture only had the first three styles defined in the spreadsheet, Data Importer would skip importing Style-4 and Style-5.

Adding the UDA Information

Adding UDA Information

The **UDA Definition** is straightforward for this example, containing just four Import fields. These will be used to insert information into the AVTTB_BLOCK_UDA table.



Import Definitions: Block Definitions, Fixture Information, Style-1, Style-2, UDA Information

Description: UDA Information
Type: Fixtures
Source Type: CSV File
Frequency: 30
Hierarchy Level: 0

File: Fixtures.csv
Start Line: 1
Delimiter Character: ,
Quote Character: "
Decimal Character: .
Date Format: MM/DD/YYYY

Ignore PK violations ☐ Reload Lookups ☑
Delete After Import ☐ Skip Existing Check ☐
Archive After Import ☐ Rename with Date ☐

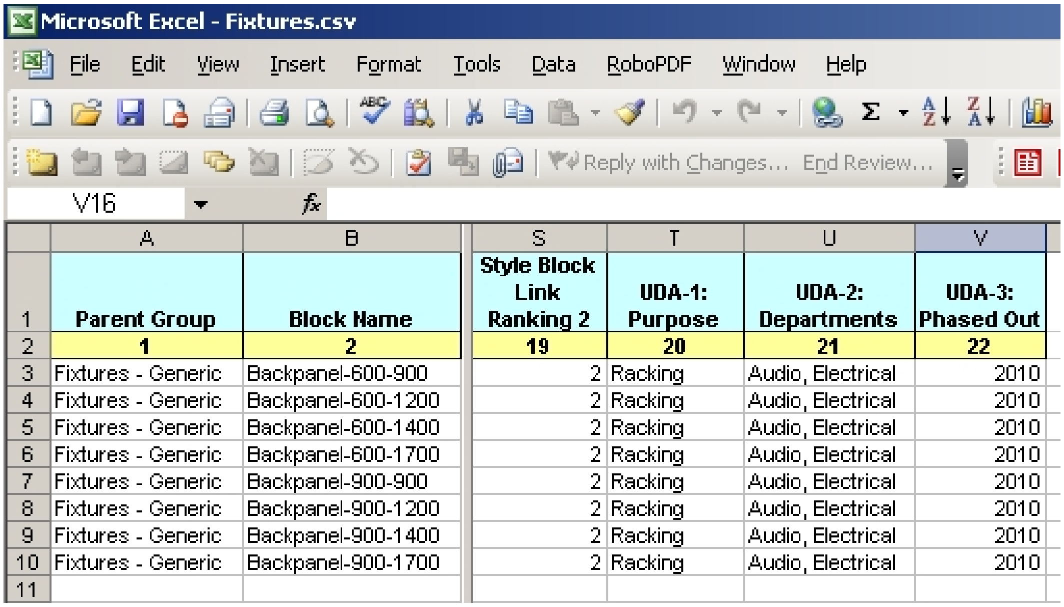
Database:
Table/View Name:

Import Fields

	Description	Table Name	Group	Field Name	Field Size	Data Type	Lookup	Default	Seq
1	Block ID	AVTTB_BLOCK_UDA	1	BLK_ID	0	Double	Block ID - PK		2
2	UDA-1	AVTTB_BLOCK_UDA	1	BRU_VALUE1	64	Text			20
3	UDA-2	AVTTB_BLOCK_UDA	1	BRU_VALUE2	64	Text			21
4	UDA-3	AVTTB_BLOCK_UDA	1	BRU_VALUE2	64	Text			22

The first import field reads the Block Name from the 2nd column of the spreadsheet and converts it to a lookup to insert the Block ID (BLK_ID) into the AVTTB_BLOCK_UDA table.

The second, third and fourth import fields read data from columns 20, 21 and 22 of the spreadsheet and insert it into the appropriate fields in the AVTTB_BLOCK_UDA table (BRU_VALUE1, BRU_VALUE2 and BRU_VALUE3).



	A	B	S	T	U	V
1	Parent Group	Block Name	Style Block Link Ranking 2	UDA-1: Purpose	UDA-2: Departments	UDA-3: Phased Out
2	1	2	19	20	21	22
3	Fixtures - Generic	Backpanel-600-900	2	Racking	Audio, Electrical	2010
4	Fixtures - Generic	Backpanel-600-1200	2	Racking	Audio, Electrical	2010
5	Fixtures - Generic	Backpanel-600-1400	2	Racking	Audio, Electrical	2010
6	Fixtures - Generic	Backpanel-600-1700	2	Racking	Audio, Electrical	2010
7	Fixtures - Generic	Backpanel-900-900	2	Racking	Audio, Electrical	2010
8	Fixtures - Generic	Backpanel-900-1200	2	Racking	Audio, Electrical	2010
9	Fixtures - Generic	Backpanel-900-1400	2	Racking	Audio, Electrical	2010
10	Fixtures - Generic	Backpanel-900-1700	2	Racking	Audio, Electrical	2010
11						

7

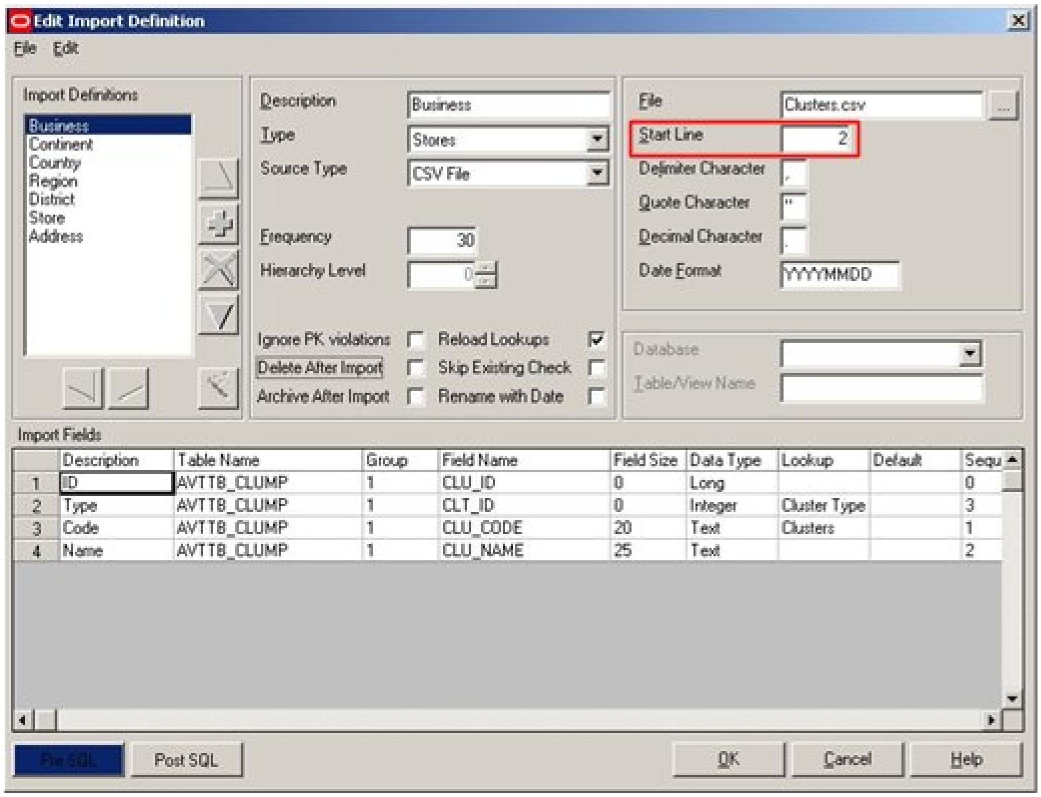
Points to Watch

CSV Formats

Important: It is important to save the CSV file in the format that is appropriate for the system it will be used on. If you do not save the CSV file in the format appropriate for the system you are working on, errors may result. For example, if you are working on a computer with the Windows operating System, and you want to use the file on that computer, it is important to save the file in the CSV (Windows) format. If however, you are using Windows and want to use the text file on a Macintosh computer, save the file in the CSV (Macintosh) format. If you are using a Macintosh computer and want to use the text file on a system running Windows or Windows NT, save the file in the CSV (Windows) format.

Incorrect Start Line

If the **Start Line** is incorrect, Data Importer will read the wrong data



Import Definitions: Business, Continent, Country, Region, District, Store, Address

Description: Business
Type: Stores
Source Type: CSV File
Frequency: 30
Hierarchy Level: 0

Ignore PK violations □ Reload Lookups ☑
Delete After Import □ Skip Existing Check □
Archive After Import □ Rename with Date □

File: Clusters.csv
Start Line: 2
Delimiter Character: ,
Quote Character: "
Decimal Character: .
Date Format: YYYYMMDD

Database:
Table/View Name:

Import Fields

	Description	Table Name	Group	Field Name	Field Size	Data Type	Lookup	Default	Sequ
1	ID	AVTTB_CLUMP	1	CLU_ID	0	Long			0
2	Type	AVTTB_CLUMP	1	CLT_ID	0	Integer	Cluster Type		3
3	Code	AVTTB_CLUMP	1	CLU_CODE	20	Text	Clusters		1
4	Name	AVTTB_CLUMP	1	CLU_NAME	25	Text			2

The start line will typically be set to 1 during configuring the data Import.

It should be set to where the first line of actual data is (typically 2 or 3) before data is imported into the database using data importer.

If it is left at 1; errors will result as Data Importer reads in data incompatible with the database.