

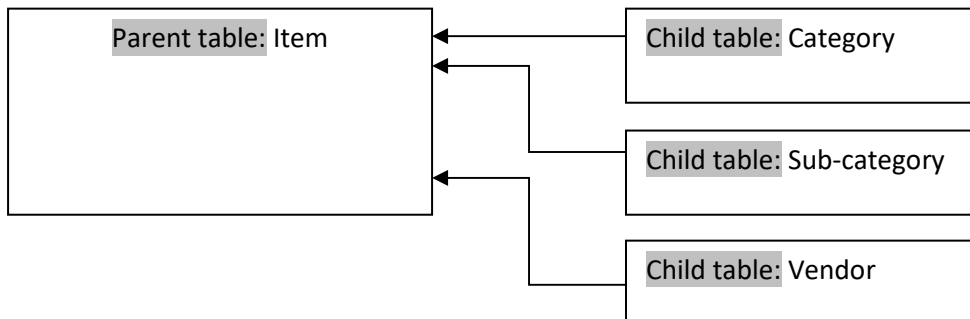
Importing into CounterPoint SQL using DB Import

Importing can greatly simplify updating your system. BUT if done incorrectly it can replace good data with bad. Importing will add or change existing data. There is no 'Undo' function. You should always try this on a practice company first and backup your live company before importing.

Sometimes you would like to add or update a large number of items, customers or other data in CounterPoint SQL (CPSQL). Here I will cover the basics and give a few pointers on ways to do it. You can import data into any CPSQL table.

You first need to understand that many tables in CPSQL have child tables, and in fact these child tables could have their own child tables.

Parent - child table relationships: The parent table is the primary table. A child table is an underlying table that the parent depends on. For example you have the Item table. The item table may refer to a category TOOL. There is a category table that will have TOOL in it. TOOL must exist in the child table before you can enter it in the parent table.



This is just a example, the Item table actually has more child tables than are shown above.

What is important to remember when importing is that the parent table can't refer to data in a child table that isn't there. For example you can't import (or even add) an item with a category that does not exist in the category table. It must exist in the category table first. Only then you can import an item that refers to this category.

Every table has required field(s). By default these are shown in blue. Every table also has a unique key field. By default this is a bold blue color. The Key field is a unique value. No two records can have the same key value. Examples of Keyfields are a customer number or item number. Think of these like a phone number. If two people had the same phone number how could you call the person you wanted. CPSQL needs a way to distinguish every record, the Key field is used for this purpose.

You can have a Template item and Template customer. When importing, CPSQL will use the Template item or Template customer if they exist to fill in fields not in the import file. Typically the template item or customer is not a real record. It is used expressly to auto fill fields when creating new records. If field information is in both the import and template record, the import takes precedence.

If you don't have a Template Item or Template Customer then all data will need to come from your import file. You can check to see what your Template item and customer is by going to / Setup / System / Workgroups /. I highly recommend using a TEMPLATE Item/Customer. This will vastly cut down on the amount of data to import and reduce the chance for errors.

Assuming you have set up templates for Items and Customers, the lists below are the fields that will be imported from those template records, if those fields are not contained in the import file. If a field does not exist in either the template file or the import file, the field will be left blank. If a field is required and is not in the template or import files you will get an error and the record will not be imported.

- Item file: Discountable, Taxable, Category, Sub-category, Account code, Primary Vendor, Stocking unit, Status, Status date, Label code, Mix-and-Match, All the descriptions on the description tab, Item attributes, Units, Other tab, Substitute items tab.
- Customer fields: Everything except credit card info

There are two import methods. 'Add new' and 'Update'.

☒ Regardless of the method used, the import file must have the key field.

Add New –Use when adding new records to the table. If only the 'Add new' record method is used and a record already exists with the same key field, the record will not be imported and you will receive an error.

Update –Used to change an existing record. With the Update record method you have two modes:

1. **Update** - If update is selected only values in the import file will be used to update the record.
2. **Replace** - If Replace is used, **all** fields in the record will be replaced. Any fields in the record that do not exist in the import file will be cleared. You might think of the Replace mode as first deleting the record, then recreating it from the import file. This method is very rarely used.

If only the Update method is used and the item does not already exist in the table, no new record is created and you will receive an error. Other records that do exist will be updated.

If both Add New and Update are used, then CPSQL will update existing and add new records from the same import file.

- Add new – New records will be added to the table (If item already exist an error is generated)
- Update – A matching record must already exist in the table (if item does not exist an error is generated)
 - Update Records – Only fields that exist in the import file are replaced
 - Replace records – Every field will be replaced. If it does not exist in the import file the field will be cleared.

When errors are generated only those lines with errors are not imported. All other lines will be imported.

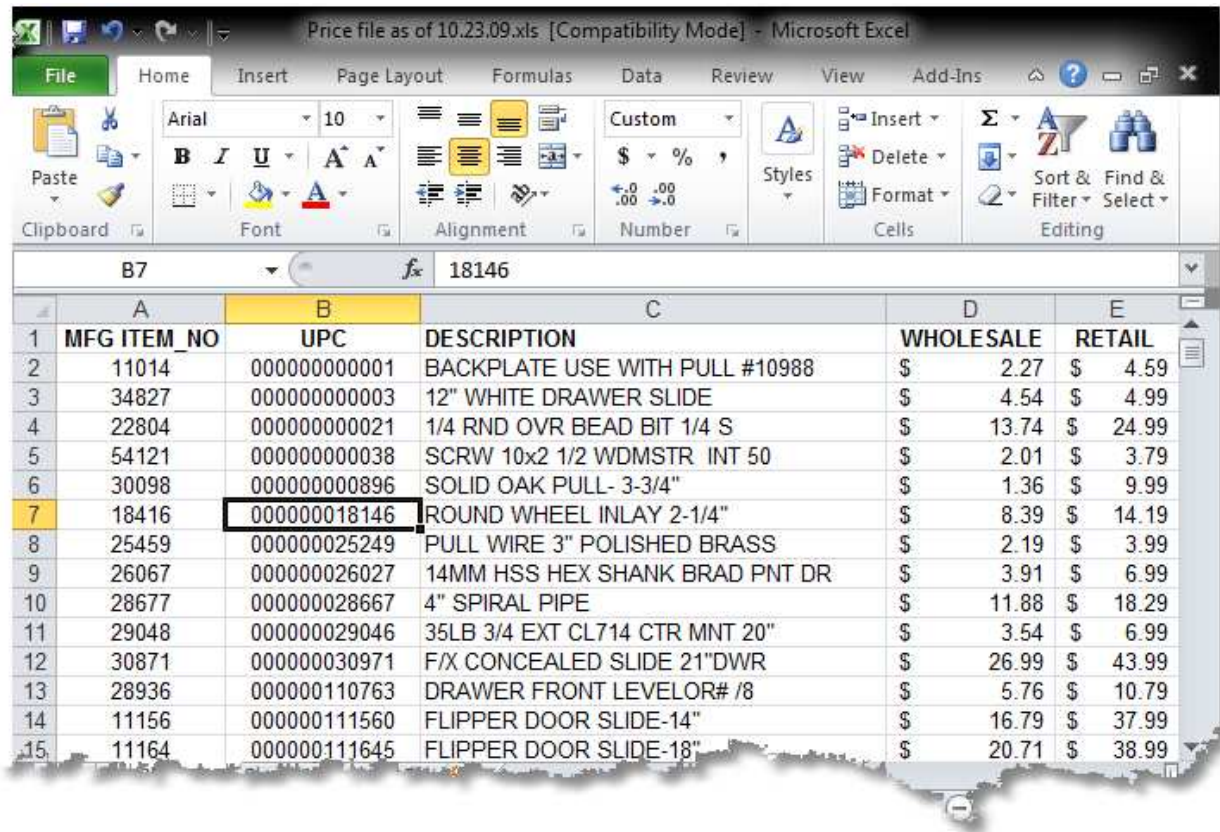
Keep in mind that errors are not always a problem.

Example 1: You may carry only part of a line offered by a vendor. The vendor may send you a price change file for you to update your system. This file contains every item offered by this vendor. When you import and use Update, you will receive errors on those items you don't carry. This is fine. You've updated the items you do carry and you don't want to import thousands of items you don't carry.

Example 2: The vendor adds new items and they send you their complete catalog file. You then import with the Add New. You will get errors on all those items you already have, but it will add any items you didn't have in your system. This is exactly what you want. You've added the new items but did not update items you already had in your system. Over time you've added better descriptions; added your own categories and sub-categories, etc. You don't want this information replaced.

CPSQL imports from CSV files. This is a universal format supported by most programs. I'll use Microsoft Excel in the following examples.

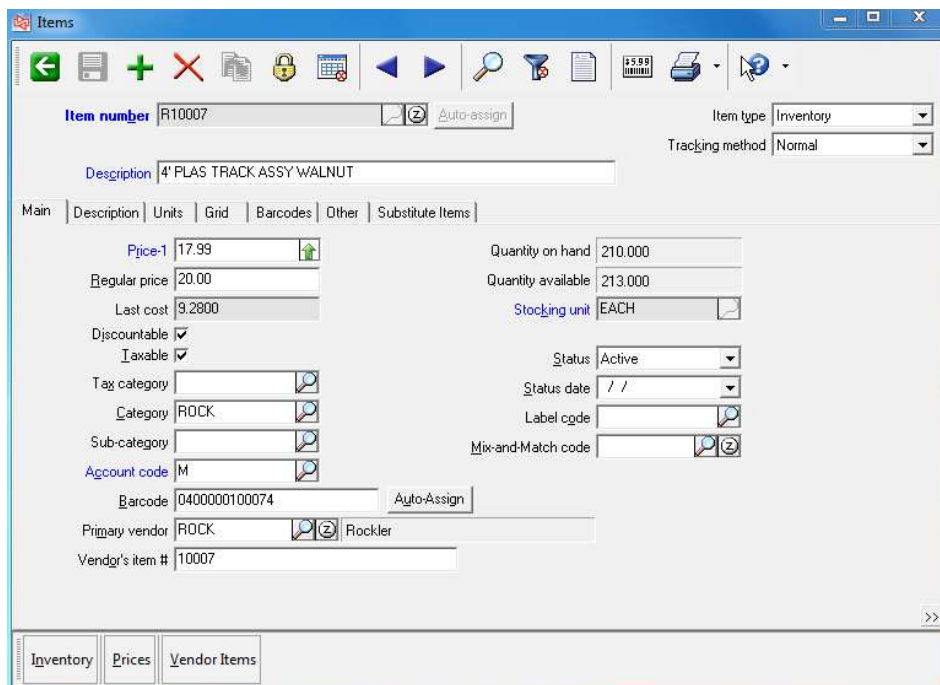
Below is a sample spreadsheet for items.



	A	B	C	D	E
1	MFG ITEM_NO	UPC	DESCRIPTION	WHOLESALE	RETAIL
2	11014	000000000001	BACKPLATE USE WITH PULL #10988	\$ 2.27	\$ 4.59
3	34827	000000000003	12" WHITE DRAWER SLIDE	\$ 4.54	\$ 4.99
4	22804	000000000021	1/4 RND OVR BEAD BIT 1/4 S	\$ 13.74	\$ 24.99
5	54121	000000000038	SCRW 10x2 1/2 WDMSTR INT 50	\$ 2.01	\$ 3.79
6	30098	0000000000896	SOLID OAK PULL- 3-3/4"	\$ 1.36	\$ 9.99
7	18416	00000018146	ROUND WHEEL INLAY 2-1/4"	\$ 8.39	\$ 14.19
8	25459	000000025249	PULL WIRE 3" POLISHED BRASS	\$ 2.19	\$ 3.99
9	26067	000000026027	14MM HSS HEX SHANK BRAD PNT DR	\$ 3.91	\$ 6.99
10	28677	000000028667	4" SPIRAL PIPE	\$ 11.88	\$ 18.29
11	29048	000000029046	35LB 3/4 EXT CL714 CTR MNT 20"	\$ 3.54	\$ 6.99
12	30871	000000030971	F/X CONCEALED SLIDE 21"DWR	\$ 26.99	\$ 43.99
13	28936	000000110763	DRAWER FRONT LEVELOR# /8	\$ 5.76	\$ 10.79
14	11156	000000111560	FLIPPER DOOR SLIDE-14"	\$ 16.79	\$ 37.99
15	11164	000000111645	FLIPPER DOOR SLIDE-18"	\$ 20.71	\$ 38.99

How does CPSQL know what column to import into which field? It uses the column heading names to determine this. CPSQL import looks for specific names in the heading. How do you know what these names are. All you need to do is open up the maintenance form for the type of data you want to import. For example, if you want to import items, go to / Inventory / Items /. To import customers, go to / Customers / Customers /.

What you see at this point are the friendly names. But this is not what is used in the tables themselves.



Item number: R10007

Description: 4' PLAS TRACK ASSY WALNUT

Price: 17.99

Quantity on hand: 210.000

Regular price: 20.00

Quantity available: 213.000

Last cost: 9.2800

Stocking unit: EACH

Discountable: ☒

Tag category: []

Category: ROCK

Sub-category: []

Account code: M

Barcode: 0400000100074

Primary vendor: ROCK

Vendor's item #: 10007

Status: Active

Status date: / /

Label code: []

Mix-and-Match code: []

Inventory | Prices | Vendor Items

Just right click on the title bar of the window.

Then click “Labels as FieldNames/ControlNames”

You will then see the actual names used by CPSQL.

The screenshot shows the 'IM_ITEM' form in CPSQL. The title bar has a context menu open with the following options: Restore, Move, Size, Minimize, Maximize, Close (Alt+F4), Labels as FieldNames/ControlNames, Labels as ControlNames, Show default labels, FormClass: TfrmItems, Print form, Default size, Last Maintenance Information, Table Information, and Customize toolbars. The form fields include: *item_no (10007), *item_typ (Inventory), *trk_meth (Normal), *descr (4' PLAS TRACK ASSY WALNUT), *prc_1 (99), *reg_prc (20.00), *lst_cost (9.2800), *is_discrbl (checked), *is_txbl (checked), *tax_catg_cod, *catg_cod (ROCK), *subcat_cod, *acot_cod (M), *barcod (0400000100074), *item_vend_no (ROCK), *vend_item_no (10007), *stk_unit (EACH), *stat (Active), *stat_dat (//), *lcl_cod, and *mix_match_cod. The bottom of the form has tabs for dxbInventory, dxbPrices, and dxbVendorItems.

At the top you will see the actual table name.

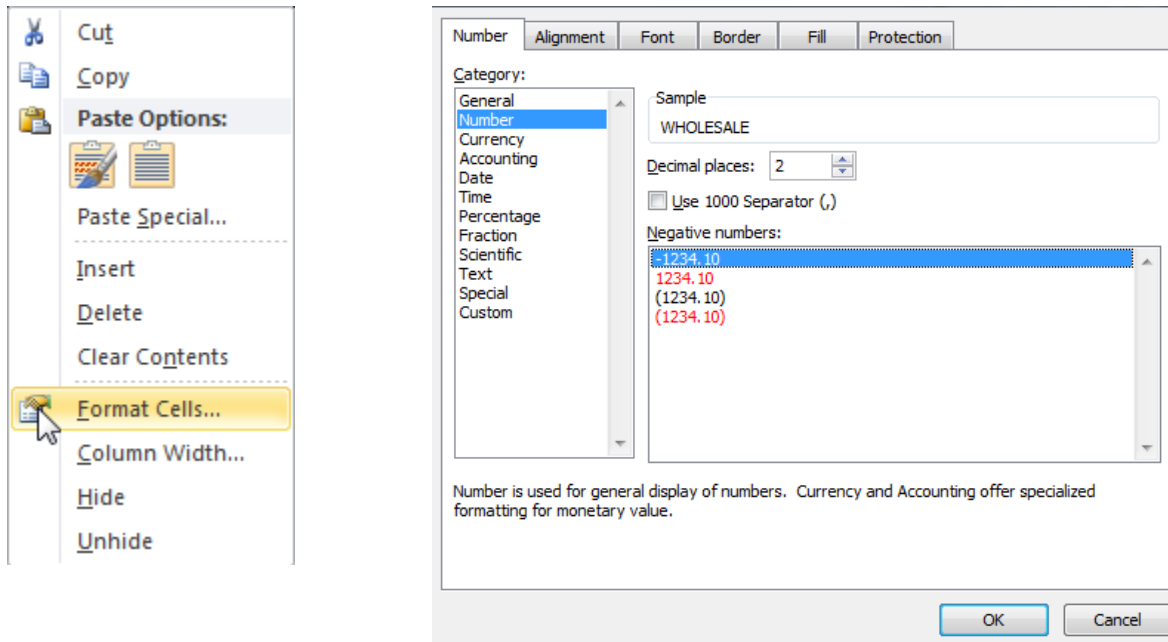
Then each field will display the actual field name.

So now you know the table name and the field names to use for column headings on your spreadsheet. The spreadsheets you receive will probably not have the correct column names so you will need to change them.

A	B	C	D	E
MFG ITEM NO	UPC	DESCRIPTION	WHOLESALE	RETAIL
11014	000000000001	BACKPLATE USE WITH PULL #10988	\$ 2.27	\$ 4.59
34827	000000000003	12" WHITE DRAWER SLIDE	\$ 4.54	\$ 4.99
22804	000000000021	1/4 RND OVR BEAD BIT 1/4 S	\$ 13.74	\$ 24.99
54121	000000000038	SCRW 10x2 1/2 WDMSTR INT 50	\$ 2.01	\$ 3.79
30098	000000000086	SOLID OAK PULL- 3-3/4"	\$ 1.36	\$ 9.99
18416	000000018146	ROUND WHEEL INLAY 2-1/4"	\$ 8.39	\$ 14.19
25459	000000025249	PULL WIRE 3" POLISHED BRASS	\$ 2.19	\$ 3.99
26067	000000026027	14MM HSS HEX SHANK BRAD PNT DR	\$ 3.91	\$ 6.99
28677	000000028667	4" SPIRAL PIPE	\$ 11.88	\$ 18.29
29048	000000029046	35LB 3/4 EXT CL714 CTR MNT 20"	\$ 3.54	\$ 6.99
30871	000000030971	F/X CONCEALED SLIDE 2"DWR	\$ 26.99	\$ 43.99
28936	000000110763	DRAWER FRONT LEVELOR# /8	\$ 5.76	\$ 10.79
11156	000000111560	FLIPPER DOOR SLIDE-14"	\$ 16.79	\$ 37.99
11164	000000111645	FLIPPER DOOR SLIDE-18"	\$ 20.71	\$ 38.99

Change MFG ITEM_NO to ITEM_NO, Change Description to DESCR, etc. Also you can see \$ in front of the prices. This

must be changed to only show numbers. If you right click on the column, and select Format Cells..., you can change the settings. Select Number, uncheck “Use 1000 Separator (,)” and pick -1234.10 as show below.

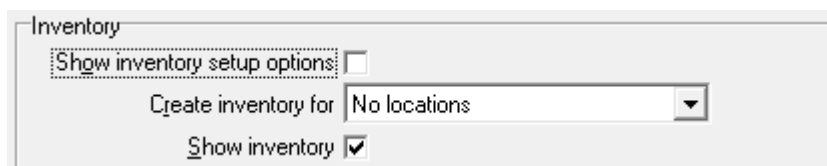


You may find it is nessassary to create formulas to get the results you want. For example you may want the description to contain a manufacturer’s prefix. So if the manufacturer is “Hewlet Packard” and the description is “High speed printer”. You want your description to be “HP – High speed printer”. You would add a new column with the heading DESCR, and use a fomula to create the description you want.

When using formulas, I recommend you save the spreadsheet in native Excel format before saving it in CSV format. If you find it nessassary to modify your fomulas you will need to use the original Excel format. When you save as CSV, you lose all formulas.

Some data may have leading zeros. For example barcodes or phone numbers. At times Excel will strip leading zeros making this values invalid. You need to find a method to prevent this. You can format the cells as text, or you can use Excel’s import feature and then specify the column type as text.

You can control whether inventory records will be created automatically when creating items manually or through import. In / Setup / Inventory / -- Control on the Item Setup tab.



Import will create inventory if you specify ‘All’ or a specific location in the ‘Create inventory for’ dialog.

If you have the control file set to auto create inventory and enter a last cost as well as enter a Vend for the item and a vendor item number the import process will create recoreds in the Item (IM_ITEM), Inventory (IM_INV), Vendor Item (PO_VEND_ITEM) files.

Handy fomulas: These formulas are examples only. You will need to modify them to fit your needs.

Example 1: Use if you have a list of names that you want to split into First and Last names. This will also check for middle initial, for example “John C. Smith” The example assumes the full name is in column A and that the middle initial has a period after it.

To get first name and middle initial if one exist.

=IF(ISERROR(FIND(".",A3,1)),LEFT(A3,FIND(" ",A3,1)-1),LEFT(A3,FIND(".",A3,1)))

To get last name

=IF(ISERROR(FIND(".",A1,1)),RIGHT(A1,LEN(A1)-FIND(" ",A1,1)),RIGHT(A1,LEN(A1)-FIND(".",A1,1)-1))

Example 2: You need to join two columns or add text to a value. You have column A and column B that you want to joined and add a dash between them.

=CONCATENATE(A1,"-",B1)

Example 3: Remove characters from text string. You have a phone number 890-5500 and you want to remove the dash. This will remove the first occurane only.

=REPLACE(target,1st character,# of characters, replacement text)

=REPLACE(A1,FIND("-",A1),1,"")

To replace all occurrences of a character. SUBSTITUTE(text, old_text, new_text, instance_num). If instance_num is used then only that occurrence is substituted. If instance_num is not included all occurrences are substituted.

Example 1-800-555-6666. SUBSTITUTE(A1,"-", "") returns “18005556666”

Messages and Errors: When importing you may receive messages and/or errors. Example: below is part of the IM_ITEM table in CPSQL.

ITEM_NO	DESCR	PRC_1					
12345	Item one	1.00					
67890	Item two	2.00					
ABCDE	Item three	3.00					

Example 1: If you are using a spreadsheet supplied by a vendor, one of the most common messages is the following.

- ❖ **“Some target fields are not in the database table you've selected. Data for these fields cannot be imported. Do you want to import the others?”**

This means is that you have one or more columns in your import file that do not exist in the CPSQL table. It could be that there are fields you do not want to import or it could be that you did not change the column headers to those required by CPSQL. If the case is you don't want to import these columns than you can ignore the message and import.

Example 2: Import Update with the following file

ITEM_NO	PRC_1		
XYZ	9.50		

You will receive this message:

- ❖ Skipped: 1

The record was not updated because the item in the import file does not exist in the table. There was nothing to update.

Example3: Import with Add New with the following file

ITEM_NO	PRC_1	CATEG_COD	
XYZ	9.50	NEW	

- ❖ ;line:1 field:1 message: Error during database post process

ACCES is not a valid subcategory of NEW.

XYZ,9.5,NEW

Errors: 1

Due to the way Microsoft SQL works, sometimes the message may seem odd, but it will still allow you to find the problem. This import record error refers to a Category NEW and the sub-category is not valid. “NEW” is not a valid Category at all. The sub-category comment is a little bit of a red herring, but it still helps you to realize that Category is the problem.

Example 4: Import with Update using the following file

ITEM_NO	PRC_1	CATEG_COD	
XYZ	\$9.50	NEW	

;line:1 field:2 message: Error converting data.

'\$9.50 ' is not a valid floating point value for field 'Price-1 (PRC_1)'

ADM-TL2,\$9.50 ,,B7,MAIN

Errors: 1

This error is because the price has a \$ in it. For all number fields CPSQL only wants numbers and where needed a decimal point. No dollar signs or comas are allowed in a numeric field(prices). See above for reformatting the field.

The following table is IM_INV

ITEM_NO	LOC_ID	BIN_1	
12345	MAIN		
67890	MAIN		
ABCDE	MAIN		

Example 1a: Import new BIN_1 location for an item using the following import file.

ITEM_NO	LOC_ID	BIN_1	
XYZ	9.50	B7	

❖ ;line:1 field:1 message: Error during database post process

Foreign key violation

The INSERT statement conflicted with the FOREIGN KEY constraint "FK_IM_INV_IM_ITEM". The conflict occurred in database "DemoGolf839", table "dbo.IM_ITEM", column 'ITEM_NO'

[Insert]

[SELECT TOP 1 * FROM [IM_INV]

ORDER BY IM_INV.ITEM_NO ASC, IM_INV.LOC_ID ASC]

XYZ,MAIN,B7

Errors: 1

This error is generated because the Item file has no XYZ item. Before you can add a Inventory record a Item record must first exist. You can create an Item without having inventory, but you can't create inventory without the item. The Foreign Key message refers to the fact the table your importing into relies on the item existing in the IM_ITEM table.

Note: If you've been paying attention you may wonder why a similar problem generated different errors during import. When adding an item that referred to a category that did not exist and when adding an inventory record for an item that did not exist we got different errors. The reason is an item does not have to have a category, but if it does the category must exist. An inventory record must have an item record. So in the first case the item was simply referring to a record that did not exist. While in the second case it's a database requirement that item records must exist for each inventory record.

As you start out using the import feature you will find you start to import a file and get an error because the table is dependent on a different table. You then go to the child table and start an import and discover it depends on another table. Don't get frustrated. You will find you will soon get the hang of it and in no time be an import wizard.

You can also give us a call and we will be happy to help with your imports. Whether it's showing you how to import or we can do it for you.

The most common imports are customers and items. Customers is pretty straight forward. But Items import can get more involved. When adding new items your adding to up to 5 tables. Items (IM_ITEM), Inventory (IM_INV), Barcodes (IM_BARCOD), Price (IM_PRC) and Vendor Items (PO_VEND_ITEM) tables.

To make your item import go as smoothly as possible do the following.

- (1) Use a Template item for all the fields you don't have in your import file. (Check this item thoroughly). Delete any field(s) you don't want in your new items.
- (2) In your Inventory Control file set Create inventory for {All locations} (Setup > Inventory > Control)
- (3) In your import file have the following fields [ITEM_NO, DESCR, PRC_1, LST_COST, ITEM_VEND_NO, VEND_ITEM_NO. You can have more but these are the most important ones.
- (4) Then import into the Item table (IM_ITEM)

If you do this the import process will add the item your item file, create an inventory record, price file record, vendor item record, and if you import a barcode the barcode file.

You will notice that the import may be slower than other imports. That is because CounterPoint is creating records in up to 5 tables and validating all the fields in every table. For this reason use a fastere computer if available. Do no be concerned if durring the import the import dialog says 'Not responding'. This just means the workstation is waiting for the server to complete the task.