White Paper

Converting Data into ALERE® Business Applications

by
Robert Mohr
Director of Technical Services
TIW Technology, Inc.

Updated November 2014

769 Youngs Hill Rd
Easton, PA 18040
610-258-5161
www.tiwcorp.com

Converting Data to ALERE Accounting and Manufacturing

This document will assist in verifying data converted using the Quickbooks, Pro or VisionPoint to ALERE Conversion Programs or in manually converting data from other packages to the ALERE packages. Please advise us if you find errors or omissions, or if you have suggestions to add.

General Information

Before Conversions

Before doing data conversions either manually or with a conversion program, it is useful to take a look and determine what if any data needs to be imported!

With PRO and Visionpoint conversions be sure to Clear Flags, and Pack and Reindex first!

Before importing data, you might want to consider deleting Cancelled or Voided orders, as often they were voided because of improperly entered information. We have found that deleting these often removes 70-80% of the bad data.

With PRO and Visionpoint conversions be sure to Clear Flags, and Pack & Reindex first!

The best time to make changes to things like a company's Item Number Policy (they have one, don't they?), Chart of Accounts, or Customer ID Policies is before starting to use a new package!

When entering screens, the highest document number will be presented first. If companies have used various document numbers over the years, and older documents have higher numbers than newer documents, you should consider a change to the numbering system once again, or possibly renumber or delete the old higher number orders. This greatly reduces frustration levels for new users, as current data is then the first thing displayed when they enter screens.

Sometimes due to problems with data in previous packages, a company may be better off only bringing in 'Static' data such as Customers, Suppliers, Inventory and Chart of Accounts. For some companies, it may be best to bring over no previous data. The Pro to ALERE and VP to ALERE programs both provide the ability to bring over just the Static data, or all data.

As of the date of this document, the VP and Pro Conversion programs convert to a special ALERE format, this is then updated to current with the TIW File Update utility. A copy of the ALERE conversion tables is included in the converison program download.

YOU MUST USE THE SUPPLIED CONVERSION DATA SET OR WILL GET ERRORS!

TIW offers Conversion Programs to convert data from Pro Series (Versions 3 to current) and VisionPoint (Series 6 to current) to the ALERE Accounting System. The compiled versions of these programs may be downloaded from our web site at: http://www.tiwcorp.com/downloads/download.html. As well, the ALERE Accounting package has built into it a Quickbooks to ALERE conversion. Additionally, the ALERE Accounting package has several imports for Excel files built into it, such as General Ledger COA, Inventory and Key Changes. Other data can easily be imported directly using Foxpro.

If you have need to modify the VP or Pro conversion programs, source versions are available; contact TIW's Marketing Department for information on them.

We recommend that for either the VP or Pro conversions, that you operate from a copy of the actual data. The programs use exclusive locks, and take some time to process all the way through. Note that with some of the older VP versions and some of the newer Pro versions, it may be necessary to make minor changes to the data structure to get the conversions to run. See the individual instructions or contact TIW Technical Services with any questions.

The 5.0 and higher versions of ALERE Accounting contain a Quickbooks conversion utility. This utility uses a third party tool to export the data from Quickbooks to an importable state. To run this utility, it is necessary to register the QODBC interface program. There is a free 30 day trial period. After that time it is necessary to contact TIW for a full key. The conversion will import static data such as Customers, Suppliers, and Item Numbers.

Both ALERE Accounting and Manufacturing have data dictionaries built into them. Go to the Manager Module, under General Reports to view or print one. shopLink uses the same tables as ALERE Manufacturing, so you can use that data dictionary.

Be aware that because of the powerful abilities of ALERE Manufacturing's Modular and Variable BOM's many companies will decide that is better to create new BOM's rather than to import their old ones. Make sure that they have studied this option before doing any BOM work. ALERE Mfg also has the ability to import BOM's from Excel files.

We also have a conversion program that converts from the ALERE version of RIATA to the ALERE InTouch Module.

With all conversions it is recommended that you limit what is imported. Clearly importing 20 years of old data will create a tremendous amount of data in the ALERE system. Most companies don't really need to have more than a couple years of data, so only import what they need. As well, realize that GL data from a batch system will be importing the batches, not the detail, so keeping years of old data serves little purpose, better to create an opening GL entry the sums up all the old transactions.

Before converting, make sure that all possible open unapplied payments have been applied, and that all modules are closed and released to GL.

Note that throughout the package there are import routines that allow importing GL Transactions, Inventory Items, Prospects, BOM's, etc from Excel files.

Before and after conversions, make Backups!

After Conversions

If the conversion includes complete data such as Orders and Invoices, you may want to update some of the DONEDATE fields on the order headers. Generally, if the order is complete, more than a year old, and doesn't have a DONEDATE record, it would be sufficient to make the DONEDATE equal to the DUEDATE. Doing this before running the File Update utility will save quite a bit of time. Files to check: SLHEADER, APHEADER, and APBILLS.

After converting data from another package, whether done manually or with a conversion program, it is necessary to test the new data to make sure that all necessary information was imported correctly. This document will attempt to aid you in this process.

After doing conversions, make sure you have run the appropriate program utilities:
For ALERE Accounting all of the following should be run:

- From Manager> System Tools: Run TIW File Update
- Pack & Reindex, then run Manager> Utilities> Remove Duplicates
- From Manager> Utilities: Reconcile Balances (Multiple runs until you show no errors)
- On the FoxPro Command Line: DO GLTEST (GLTEST is located in the Developers Kit directory)

For ALERE Manufacturing or shopLink

- From Manager> System Tools: Run TIW File Update
- On the FoxPro Command Line: DO IDUPDATE
- On the FoxPro Command Line: DO BMTAGFIX
- On the FoxPro Command Line: DO ROUTERECONCILE
- From Manager> Utilities: Reconcile Balances

Many things you need to check are obvious, here are some of the most important for Accounting:
- Chart of Accounts
- Company Defaults (Setup screen, sets Cost method to be used for Inventory)
- Balance Sheet
- Income Statements
- Aged Payables and Receivables
- Prepayments
- Outstanding (Open) Sales Orders and Purchase Orders
- Company Codes
- Bank Reconciliation
- Unrecognized Payables and Receivables
- Inventory count and corresponding GL Balances

Usually the easiest way to do this is to print out all of these reports from the User's old accounting system. The ALERE reports can all be displayed to screen, so you may want to do it on screen, or print out reports to compare. Finally you need to verify that the "Next Order Number" is valid for all the items listed in the Manager Module, Utilities. Note that this includes non order items like CASHID.

A conversion will rarely be 100% accurate. It happens, but don't bet on it. But don't panic, correcting the data is generally not difficult, although it may be a little time consuming. Make sure that the customer is aware that errors may become visible over a period of time, so they should not panic either! Even issues where balances are doubled or tripled are generally not difficult to fix.

Since the ALERE packages rely heavily on SQL commands, they are sensitive to duplicate records. When importing data, it is possible to get duplicates in many places. It is most important to remove all duplicate records! Remember that duplicates might be the same Key ID record entered in both Upper and Lower Case. Duplicates will often be quite visible when running reports. For example if all the line items showing on a Sales Order have duplicates, the most likely places to look for a duplicate record are:

- COADDR
- SLCUST
- COTERMS
- COCODES

NOTE: Current versions of ALERE have a duplicate removal Utility in the Manager Module.

If only certain line items are duplicated, you probably need to be looking in Inventory tables:

- IMMASTER
- IMLOCN
- IMSTOCK
- IMSERIAL
- IMSOURCE

Think about what is duplicated. If all the orders for only a couple customers have all line items duplicated, it is likely a problem with the Customer Record, Company Codes or Terms. If other customers' records using the same terms are OK, you have eliminated that one.

A couple simple SELECT statements that will help finding duplicates:

```
SELECT coid, locid, COUNT(*) as mcount FROM coaddr GROUP BY 1,2 having mcount>1
SELECT item, count(*) as mcount from immaster group by item having mcount>1
SELECT item, locid, count(*) as mcount FROM imlocn GROUP BY 1,2 having mcount>1
SELECT routeno, count(*) from roheader group by routeno having mcount>1
```

We have seen several sets of data where GL transactions had more than 2 decimal places in the postings. To find this try:

```
SELECT * from glpost where INT(amount*100)/100 <> amount into cursor temp
```

After making changes to databases directly, make sure to again run the Reconcile Balances to update other databases. Reconcile Balances works on a 'Bottom Up' concept. It will look at the lowest level data, and update the database next up the ladder. Some examples:

- IMLOCN updates from IMSTOCK
- IMMASTER updates from IMLOCN
- GLTOTALS updates from GLPOST
- GLCHART updates from GLTOTALS
- COCHECK updates from GLCHART
- SLCUST updates from SLHEADER
- SLHEADER updates from SLLINES

This is the reason that you may need to run the Reconcile utility several times before everything 'Passes'. This makes correcting problems quite simple, as a change to GLPOST or IMSTOCK will flow upward to the relevant databases.

Before and after changing data, don’t forget to make Backups!

ALERE Accounting Activity Logs

Activity Logs

ALERE Accounting uses Activity Logs to track certain processes.

- IMACTLOG tracks all Inventory movements, and cost changes, and is the source for reports like the As Of Inventory Report. It provides much of the Activity information for the Inventory, Inventory Log of Activity and Inventory Future Activity Screens. If there are problems in the Inventory As Of Report or these screens, this is the first place to look.

- GLITEM tracks GL activity of specific Inventory Issues and Receipts. This allows the system to determine specific accounts hit by individual items in a massed transaction.

- COACTLOG tracks every order processed. COACTLOG is in the process of being phased out. It currently supports only the Customer and Supplier Transaction reports.

ALERE General Ledger Module

Chart of Account

ALERE uses a single database for the setup of the GL Chart of Accounts (GLCHART). It is very important to check the setup of the Chart of Accounts!
The RECTYPE field will have a * for Categories, a + for Subcategories, and numbers 1-6 for the Account Segment. Sometimes when importing data, the Categories and Subcategories don’t come over, and must be entered manually, or extra ones come over that must be deleted.

Note: It is quite common to have extra Categories or Subcategories come over in a conversion! Generally these come over with the flag as Category or Subcategory, but is missing the AC-CTNO and/or HILIMIT number. This will cause various reports to give incorrect values.

The starting point of the range for Categories and Subcategories is in the ACCTNO field. The end of the range is in the HILIMIT field. Overlapping ranges will cause a range of account numbers to have their balances doubled or tripled. Account numbers outside the defined ranges will not show on these reports. Make sure here that you do not have overlaps of the account number ranges. The Account Balances will be updated by Reconcile Balances.

After verifying the COA, check the setup of Fiscal Periods, making sure you have years set up as far into the past as necessary, and at least one year into the future.

Carefully check accounts that are set up in the Default GL Accounts and Custom GL Accounts screens. You MUST fill in all the account numbers in the Default GL Accounts screen for ALERE to function properly! If you do not have all accounts filled in, you will get ‘Invalid account’ error messages as you post transactions. Note that this message also is used if an account is flagged as inactive. If you look at the Default and Custom GL screens, inactive or invalid accounts will be highlighted in Red.

Generally, each field in the Default GL Screen should be a different account number! Exceptions to this are the 2 Checking Accounts. Very carefully evaluate any that are duplicated to be sure that their usage is valid! If you have questions, feel free to call TIW Technical Support. A mistake here will generate a lot of incorrect transactions down the road. We have often seen customers set up things like Inventory and Inventory Clearing as the same account. This makes it almost impossible to track down problems when posting inventory.

ALERE uses Custom GL Accounts to override the Default GL Account setup. There is no need to set up Custom GL Accounts for processes using all the default accounts. If an item (or Customer) uses all default accounts, just leave its GL Group blank, it will then automatically use the Defaults. There is no need to put 'DEFAULT' everywhere. By the same logic, when you set up Custom GL Accounts, only enter account numbers that are DIFFERENT than the default, leave the ones that are the same as the defaults blank. Use the Inquiry Tab on the Custom GL Screen to verify that the accounts you expect are being used. Look at WHAT custom GL groups are based on. Note that if a Customer or Item is set to ANY CUSTOMER or ANY ITEM, that means it applies to ALL. This is an area that we get lots of questions on, so check it carefully. ALERE will be posting all your transactions based on these settings!

NOTE! <u>ALERE does not accept any GL transactions that are not balanced.</u> Importing data could certainly bring over out of balance records. Starting with ALERE v5.1, the Developer's Source Kit has an updated utility (GLTEST) that verifies that GL Transactions are in balance, and have the correct sign. If you do not use the GLTEST Utility, make sure that you run the Trial Balance for EVERY period imported.

Correcting unbalanced entries will normally be done directly in the GLPOST database. If changes are being made directly to GLPOST, be aware that since ALERE reflects Par Value, there are fields for both POSTTYPE (Debit or Credit), and AMOUNT (which could be positive or negative). We suggest that you again run the GLTEST utility after making any manual changes to the GLPOST data. Of course GLTEST is a great tool to use if you suspect that you have corruption in the GLPOST records.

<u>Make sure that GL postings do not contain fractional cents!</u>

GL Periods

After converting or importing data, make sure to verify that the records in GLPERIOD are correct. Not only for the current year, but however far back you are importing data. Make sure to check for gaps and overlaps!

Since ALERE is a date based, real time system, it does not need to track GL transactions by period. All transactions are posted by date only. The 'Periods' are used in GLTOTALS and reports for tracking total postings and are methods set up on the reporting and display functions. This means it is actually possible to change the period definition of the year you are currently posting if necessary. You can easily close a year short, or even change the dates for closing each period.

If you change the period definition, be sure to run the Reconcile Balances utility multiple times to update the GLTOTALS. If it is ever necessary, this logic can also be used to change dates of a transaction. Update the dates on GLHEADER and GLPOST, then run Reconcile Balances. Remember, of course, if you change dates of transactions within periods for which the accounting reports have already been run, it will be necessary to reprint those reports.

Data Flow on Manual GL Transactions

Let’s look at the flow of data when a manual GL Transaction is posted.

- A transaction will update: GLHEADER, GLPOST, GLTOTALS, GLCHART, COINFO, and possibly GLPERIOD.

DOCTYPE’s on GL Transactions

The DOCTYPE on GL Transactions will indicate where it came from. These same codes are used throughout the package.

- AP Accounts Payable
- AR Sales Invoices
- BS Blanket Orders
- CM Credit Memos
- DM Debit Memo
- IA Inventory Adjustment
- IP AR Payment
- IR Receipt
- JE Manual Journal Entry
- MO Manufacturing Order
- OP AP Payment
- OS Shipment
- PO Purchase Order
- PR Purchase Return
- QP Quoted Purchase
- QS Quoted Sales
- RE Reversing Entry
- RP Recurring Purchase Order
- RS Recurring Sales Orders
- SO Sales Orders
- SP Sales Prepay (No longer used)
- SR Sales Return Orders
- TO Transfer Order
- YE Year End Entry

ALERE Inventory Module

Inventory Items

In the Inventory data, make sure that all inventory items are set up in both IMMASTER and IMLOCN. If they are Stocking items they should also be in IMSTOCK. If the customer is using LIFO or FIFO costing, you will also need to populate IMTIER. Inventory Supplier records are in IMSOURCE. You can have as many suppliers as desired for each item.

Only Serialized items should be listed in IMSERIAL. Only Lot or Serialized items that are specifi-cally allocated to unshipped Sales Orders should be listed in IMALLOC. Make sure to check these two databases for unneeded data, and delete any unneeded records.

Changing Cost Methods

To change the inventory Cost method, it is as simple as going into WSINFO and changing the ACCOST field.

- A for Average Cost
- S for Standard Cost
- F for FIFO Cost
- L for LIFO Cost

Before doing the change in WSINFO, make sure you have made any corrections that will be necessary:

- If changing from Ave to Std, use Inventory Adjustments to make Std Cost equal 100% of Ave Cost, Then change the ACCOST to “S”
- If changing from Std to Ave, use Inventory Adjustments to make Ave Cost equal 100% of Std Cost. Then change the ACCOST to “A”
- To change to LIFO or FIFO, first use Inventory Adjustments to get correct values in Ave and Std costs. Then make a copy of IMLOCN, changing the field names to match those in IMTIER, and simply append the modified IMLOCN into IMTIER. Then be sure to set the TIERID to a right justified ‘1’ for each item. Finally change the ACCOST to “F” or “L”.
- To change FROM LIFO or FIFO, use the Reconcile Balance Utility to update Average Cost. If going to STD Cost, then use the Inventory Adjustments to set Std Cost equal to 100% of Ave Cost. Then change the ACCOST to “S” or “A”. Finally Blank the data in IMTIER.

By updating in this manner, there is no GL impact to the actual conversion from one method to another, as all changes were properly done within the system first.

Data Flow on Inventory Adjustments

Let’s look at the flow of data when an Inventory Adjustment is posted.

- Processing an Adjustment to Price updates IMMASTER
- Processing an Adjustment to Cost updates IMMASTER, IMLOCN, IMACTLOG, GL-HEADER, GLPOST, GLTOTALS, GLCHART, GLITEM, possibly GLPERIOD. Note that LIFO or FIFO companies cannot change Average Cost.
- Processing an Adjustment to Quantity updates IMMASTER, IMLOCN, IMSTOCK, IMACT-LOG, IMTIER (if LIFO or FIFO), IMSERIAL (if Serialized), GLHEADER, GLPOST, GLTO-TALS, GLCHART, GLITEM, possibly GLPERIOD.

ALERE Accounting Manager Module

Global Codes

All ALERE Accounting product wide Codes are set up in COCODES.
Make sure that INVENTORY TYPE codes are set up for Finance Charges and Shipping as used in the Manager Module> Company Defaults, and that these codes are assigned to the appropri-ate items in Inventory.

Set up whatever codes are associated with the actual Inventory Items in IMMASTER. These may be GL codes or Sales and Pricing Codes. Again remember that you don’t have to set up codes for the ‘Default’ items, just the non standard ones.

Make sure there are no duplicates in the Codes. Remember to look for items that have the Key Field (CODENAME) in both upper an lower case. All Key Field entries should always be in all UPPER Case.

Under Global Definitions verify all the Code Definitions. Currently the Global Codes are:

AGP	Approval Group
BUY	Buyer
CCD	Customer codes
CLS	Product Class
COS	Company Status
CPL	Pricing Level- Grouping of customers with special pricing
CUR	Currency
EMA	Email Message Text
FAS	FASB Code
FOB	Free on Board Code
GLI	GL Item Group- GL accounts set to specific items
GLR	GL Account Rules
GLS	GL Sales Group- GL accounts set to specific customers
ICD	Inventory MISC Code
IND	Industry
ITM	Price Group- Grouping of inventory items for special pricing
KCD	Contact MISC Codes
KCF	Profile Custom (MISC) Fields
PCD	Purchase MISC Codes
PRT	Printer Codes
REM	Order Comments
RJG	Recurring Journal Entry Group
RLS	Release Group- Grouping of Recurring Orders for release purposes
RUL	Locator Rule
SCD	Sales MISC Codes
SLS	Sales Person
SLN	Sales Lines MISC Codes
STM	Sales Team
SUP	Supplier Code
TRA	Traits
TYP	Item Type, Required to be one for Ship and one for Finance Charge items
UOM	Unit of Measure
VIA	Ship Via

Also under Global Definitions, verify or set up

- Unit of Measure Conversions (COUNITS)
- Terms (COTERMS)
- Tax Districts (COTXRATE & COTAXTBL) If your customers are taxable, it is mandatory to create tax districts, and assign them to the customers locations.

Often these codes and other Global Definitions can be added to Customer or Supplier records in mass if not brought over (or not brought over properly) via the conversion. Filling in as many Codes as possible in the Customer and Supplier records will save a user much time and frustration when using ALERE. In COADDR I would make sure the following are populated for a Customer record.

- SHIPVIA (Carrier)
- SHIPFROM (Location)
- SHIPFOB
- SALESMAN
- REMARKID

Also for Customers in SLCUST consider entering

- INDUSTRY
- TERMS
- FINANCE

For a Supplier record, in COADDR I would fill in

- RECVVIA (Carrier)
- RECVLOCN (Location)
- RECVFOB
- BUYER
- REMARKID

Also for Suppliers in APSUPL consider entering

- TERMS
- RANK

Filling in these fields will make the User's job much easier with just a little extra effort during conversion. Note that the ALERE Key Change will allow you to change or merge Terms.

Checking Accounts

Checking Accounts are set up under Banking Actions> Checking Accounts. Note that 'non cash' accounts can be set up here for processing things like bad debits. This data is stored in CO-CHECK. If payments are going to be made by company credit card, there should also be a Liability account set up to transfer liability from a Supplier to the Credit Card Account.

Company ID and Locations

All Company Addresses are stored in the COADDR.DBF. This includes

- The Company's Internal Locations
- Customers
- Suppliers
- Others (might be used for employees, Potential customers etc.)

Note that each location for a company will be an additional record in COADDR. Some rules for records in COADDR:

- If a company has multiple locations, there will be one record in COADDR for each.
- If a Company has multiple locations, they must all have a LOCID to identify each location (one cannot be blank).
- Any customer location that can have shipments made to it will be flagged in ISSHIPTO.
- Any customer location that has bills sent to it will be flagged in ISBILL.
- Any supplier location that can receive POs should be flagged in ISPOTO.
- Any supplier location that receives payments should be flagged ISREMIT.
- One customer and or supplier record must be marked MAINLOCN.
- Only one record for a customer and/or supplier may be marked MAINLOCN.
- All Internal locations will be flagged as ISMYCO.
- If a Customer or Supplier only has one location, blank is an acceptable LOCID, however we have found that Users understand functioning better if a LOCID is set up for all records.

Additional Rules for Versions before 5.0:

- If a company is a customer ISCUSTCO must be checked for all of that customer's locations.
- If a company is a supplier ISSUPLCO must be checked for all of that supplier's locations.
- If a Customer is also a Supplier the ISCUSTCO and ISSUPLCO fields must be checked for ALL of their locations.
- A customer's default ship to location should be flagged in MAINSHIP. Note that in version 5.0, it is replaced by the SHIPID and SHIPLOC in SLCUST
- A suppliers default PO to location should be flagged MAINPOTO. Note that in versions 5.0, it is replaced by the POTOID and POTOLOC in APSUPL

There are also secondary databases that track Customer specific information (SLCUST) and Supplier specific information (APSUPL). Before v5.0, there would be one record per company in SLCUST or APSUPL. Starting with v5.0 there is a record in SLCUST for every Bill To Location, and one record for each Remit To Location in APSUPL.

With current versions of ALERE, there is also a Contact database, COPEOPLE. The TIW File Update will automatically create COPEOPLE records for the primary contacts in COADDR.

Bank Reconciliation

Bank Reconciliation affects

- APCASH
- SLCASH
- COBANK

Cleared checks are flagged with the date that they are cleared in the xxCASH database. To simplify the first Bank Reconciliations in ALERE after importing data, you want to flag all checks before a certain date as being cleared (generally a couple months prior to the conversion date).

To do that, in APCASH and SLCASH:

- Select the checks earlier than the desired date.
- Set the BANKDATE to that date for all checks earlier than desired date that have BANKSTAT other than V (Void). Set the BANKSTAT for these records to C (Cleared). This will clear these items, so that when Bank Reconciliation is run, you don't have to clear all these old records.
- Do this in both SLCASH and APCASH
- Set the balance for the first entry in COBANK for each checking account. To do this:

  1 Create the first Bank Reconcilliation in ALERE's Bank Rec Screen. Just select some items that need to be Cleared, and SAVE. Don't worry at this point that the ending balance is incorrect.
  2 In the Bank Reconciliation Screen, set the date well into the future, click REFRESH and ALL.
  3 Take the "Ending Balance" number, (from the Reconcile Column) subtract from it the actual balance of that account, Subtract that number from the amount in the Starting Balance field, and enter the resulting number in the CLOSEAMT field in COBANK, replacing the number that was there.
  4 After saving, go back to the Bank Reconciliation Screen, and again set the date in the future and select ALL. The Ending Balance should now be the same as the Bank Balance. This should work unless there are duplicate records in the xxCASH tables. Do this for each checking account set up.

- One additional thing to do in SLCASH. Any Voided Checks, the Amount field should be set to 0 to make sure they don't affect reconcilliations.

Note that Bank Reconciliation does not link in any way to General Ledger, or the checking account balance. It is only a tool for verifying your bank's statement.

Order Numbers

The COINFO database contains the 'Next order number' for all types of orders. Make sure that the numbers are the next higher available number for each type of order. This can be directly accessed in the Manager Module under Utilities.
Many times companies will have used different ordering number schemes over the years. We recommend that the current order numbers should be higher than any older order numbers. In some cases you might want to renumber older orders to a lower, blank range of order numbers, or if possible they might be deleted. Also check for non numeric order numbers. We have seen some data come in that had dashes and other non-numeric characters in the order numbers. These will need to be corrected.

ALERE Sales and Purchases Modules

Orders

For the Sales Module based orders, the DOCTYPEs in SLHEADER, SLLINES and SLCASHTR refer to

- AR Sales Invoices
- BS Blanket Orders
- CM Credit Memos

- RS Recurring Orders
- SO Sales Orders
- SP Sales Prepay (No longer used)
- SR Sales Return Orders

For the Purchase module, Purchases, the DOCTYPEs in APHEADER and APLINES refer to

- PO Purchase Order
- PR Purchase Return
- QP Quoted Purchase
- RP Recurring Purchase Order

For the Purchase module, Payables, the DOCTYPEs in APBILLS and APFROMPO and AP-CASHTR refer to:

- AP Accounts Payable
- DM Debit Memo
- PD Purchase Discount (APCASHTR Only)
- PP Purchase Prepayment (APCASHTR Only)
- SP Sales Prepayment Refund (APCASHTR Only)

Normally when data is imported, there would not be any outstanding Approved Payables; if there are, they would go into APAPPROVE.

Order Addresses

There are several address codes on each order. SLHEADER has:

• BILLTO, BILLLOCN	(Where bill goes)
• SHIPTO, SHIPLOCN	(Where shipment goes)
• SHIPFROM	(Your shipping location)
• REFTO, REFLOCN, REFADDR	(Reference, new in 5.0, refaddr is memo holding detail)

SLLINES has:

• SHIPID, SHIPFROM	(Where shipped from, Drop Ship would be Supplier ID. ShipId is new in 5.0)

APHEADER has:

• RECVAT, RECVLOCN	(Where shipment goes, RECVAT new in 5.0)
• POTO, POLOCN	(Where PO goes)
• PAYTO, PAYLOCN	(Where Payment goes)
• REFTO, REFLOCN, REFADDR	(Reference, new in 5.0, refaddr is memo holding detail)

Starting in 5.0 it is critical that APHEADER has a RECVAT record. This is the company ID that the order will be received at. This supports Drop Shipping, as well as receiving at your own location (RECVAT= ISMYCO). As well in SLLINES, there is now a SHIPTO as well as a SHIP-LOCN. The SHIPID in SLLINES is the company the item is being shipped from, either a Supplier for Drop Ship Orders, or your own company for normal orders.

Payments and Receipts (including Prepayments)

Payments are tracked in APCASH and APCASHTR. Sales Receipts are tracked in SLCASH and SLCASHTR. In both cases the xxCASH database tracks the total $ value, the xxCASHTR tracks the specific DOCID that the $ is applied against.

Customer Prepayments are also tracked with the SLCASH and SLCASHTR. Errors in this area are easiest to correct by adding the appropriate correcting records to SLCASH and SLCASHTR. Remember to run Reconcile!

Sometimes Sales Prepayments will not be carried over in a conversion, or ones that were already used may be carried over. If you have a prepayment showing that was already used, you can simply delete the record for that prepayment in SLCASHTR (not in SLCASH). Make sure to verify the amount! Remember that you could have a situation where part of a prepayment was used, but not all. In this case you could edit the amount of the “SP” record in SLCASHTR, or you could add records to SLCASH and SLCASHTR to indicate a use of the “SP”. Most other packages do not support payments against a Sales Order, so after converting the data, check SLCASHTR for

Normal processing in ALERE will receive the prepayment in SLCASH & SLCASHTR, and then have entries for how it was applied in the same tables.

If a Invoice shows as partially paid, but the Aging report or Statement shows the full amount, the problem is likely in the SLCASHTR database. As payments are made on an invoice, the specific application of them will be made in SLCASHTR. Check if there is a payment record. If there is, correct the amount. If there is not, add a record to SLCASH and SLCASHTR. Note that a DONE-DATE before the dates on the aging report would inhibit the record from showing there, but it may still show on the Statement.

Done Date

There is a DONEDATE field in the various Order Headers. This field is very important for AP and AR Aging reports. Since they are ‘As Of’ reports, they need to know when an order is paid. The DONEDATE provides this information. If orders are incorrectly showing up (or not showing) for a particular date, it usually means that the DONEDATE needs to be corrected or entered. Additionally you have the xxCASHTR tables that track payments to specific orders/invoices. When entering a DONEDATE for completed orders, it is generally more accurate to make it equal to the Due Date than Today’s Date.

Custom Pricing

Special Sales Pricing is all set up in SLPRICE.DBF. If special pricing has been imported, make sure that the STATUS is ‘A’ for Active pricing. Often it is better to set up Pricing in ALERE rather than importing it, as there are often simpler ways to set it up than a Users existing methods.

Data Flow on Sales or Purchase Orders

Let’s look at the flow of data when a Sales Order is processed. (Or other Sales Process, such as: Invoice, Credit Memo, Return Order, Recurring Order, etc.). Note that GLPERIOD could be updated by any posting. Based on Closing Grace Days, it will update the period status as necessary.

- Creation of the SO updates SLHEADER, COACTLOG and COINFO.
- Adding lines, on SAVE updates SLLINES and SLHEADER gets the total updated. If Lot or Serial number items are allocated IMALLOC is also updated. If an item is configured, SALESCFG and possibly SALESMTL.
- Accepting the SO updates the status in SLHEADER, Allocates in IMLOCN, and updates customer totals and dates in SLCUST.
- Issuing the inventory on a SO or manual Invoice updates IMACTLOG, IMMASTER, IMLOCN, IMSTOCK, IMSERIAL (for Serialized Items), IMTIER (if LIFO or FIFO), SLLINES, GLHEADER, GLITEMS, GLPOST, GLTOTALS, GLCHART. If a Kit, SALESMTL.
- Creation of the Invoice will impact: SLHEADER, SLLINES, GLHEADER, GLPOST, GLTOTALS, GLCHART, COACTLOG, COINFO.
- Payment of Invoice will impact: SLHEADER, SLCASH, SLCASHTR, SLCUST, GL-HEADER, GLPOST, GLTOTALS, GLCHART, COINFO.

The flow on Purchase Orders is similar

- Creation of the PO updates APHEADER, COACTLOG and COINFO.
- Adding lines, on SAVE updates APLINES and APHEADER gets the total updated.
- Accepting the PO updates the status in APHEADER, places On Order in IMLOCN, and updates supplier totals and dates in SLCUST.
- Receiving the PO updates IMACTLOG, IMMASTER, IMLOCN, IMSTOCK, IMSERIAL (for Serialized Items), IMTIER (if LIFO or FIFO), SLLINES, GLHEADER, GLLINES, GLPOST, GLTOTALS, GLCHART.
- Creation of the Payable will impact: APHEADER, APLINES, APBILLS, APFROMPO, GLHEADER, GLPOST, GLTOTALS, GLCHART, COACTLOG, COINFO.
- Payment of Invoice will impact: APBILLS, APCASH, APCASHTR, SLCUST, GLHEADER, GLPOST, GLTOTALS, GLCHART, COINFO.

The flow on Drop Shipment of Sales Orders is similar to the above:

- From the Status Tab Create the PO. Updates the SOLINES.POQYT plus PO Data as above
- From the SO Status Tab, or Inventory Drop Ship Recording, Record the Drop Ship. Processes as Receiving the PO plus Shipping the SO

ALERE InTouch Module

The InTouch Module shares the use of the COADDR and COPEOPLE tables. There is a COMISC which contains the profile UDF fields, and COCONLOG which contains Contact Notes. There are additional tables for Prospects COLDADDRESS, COLDNAME, COLDMISC, COLDLOG to hold the same information. As well there are 2 tables for Prospect Quotes; KSHEADER and KSLINES. The TIW File Update will populate the COPEOPLE table from the COADDR information. If running the RiataToTIW conversion program, contact info will be added to COPEOPLE from Riata. There is a second utility program that will take any company record that has never had any activity, and convert them to the Prospect data. This utility can be run even if not converting from Riata. (CONVERTTOPROSPECT.FXP)

CORULES defines the Mass Emailing Rules. COEMLOG contains a log of emails sent.

ALERE Manufacturing BOM Module or ALERE Configuration Module

Basic BOMS

While TIW does not offer a conversion program for BOMs, simple BOMs are easy to convert. There is a BMREV database that holds all the information for the Parent BOM, and the BMSL database holds the information for the Children. Using standard FoxPro tools, it is a simple matter to populate these two databases. The BMREV database contains a field called BOMTAG. When importing or appending data to this table, leave this field blank. After you're done importing data, run TIW File Update to properly populate this field. As well, they will populate the INACTIVE fields when they are blank. Note that ALERE Mfg also allows importing BOM's via Excel.

The BOM Types in BMREV are

1	Component
2	Modular
3	Alternate
4	Phantom
5	Variable
6	Kit

ALERE Manufacturing supports all of these.
ALERE Accounting Configurator supports creating types 2, 5, and 6.
shopLink supports 1, 3, 4, and 6.

One caution: some packages 'echo' the BOMPARENT item number in the CHILD item list. If this is the case, make sure that you delete this extra item from BMSL.

NOTE: The current BOM Module has a Import from Excel routine to import BOM's

Multi Level BOMS

Multi level BOMS are simply made up of a lower level Parent BOM (BMREV) being a Child Item (BMSL) of the Parent above it. There is nothing special that needs to be done to accomplish this. Just add the lower level Parent to the higher level Child list.

Configurable BOMS

Generally if a company is planning to use Modular or Variable BOMs, they will be entering the BOM information manually, so there is no need to deal with BMFGCFG or BMCFGSAV. If it is necessary to work with these databases, be aware that they use 3 digit ASCII characters to uniquely identify BOMs from the BMREV database. These get concatenated to provide references through multiple levels of the BOM in the configuration tables. For example, from the standard Mikes Bike BOM

BOMTAG	Item Number	Description	Status
!!"	MBBA01	Bike Assembly (to be configured)	Top Level Item

!!E	MBYF01	Men’s or Woman’s Frame	1st level Modular
!!F	MBYFM1	What Color Men’s Frame	Mod Answer/ 2nd Level Mod
!!G	MBYFW1	What Color Woman’s Frame	Mod Answer/ 2nd Level Mod
!!5	MBFR19	Men’s Frame White	Answer to 2nd level Modular

In BMFGCFG the saved configuration for the default choices on MBBA01 would be:

- The BOMTAG would be !!”
- The BOMTYPE would be 2 (Modular)
- The AITEM would be MBFR19
- The BOMLINK would be !!E!!F

BMCFGSAV stores information on Modulars more than a level deep. For the above example:

- The BOMTAG would be !!E
- The QITEM would be MBYFM1 (MBYFW1 if Woman’s selected)
- The BOMTYPE would be 2 (Modular)
- The BOMLINK would be !!E!!F

Generally you want to save the configurations in the BOM Config tab to populate this database.

The same BOM Tag logic is used in the SALESCFG database which contains configuration information for configured Sales Orders. It is also used in WOFGCFG for Work Order Configurations, and PMFGCFG for Planning purposes.

Remember that moving all the BOM’s from one TIW Company to another is as simple as copying all the BMxxx files, then running the TIW File Update routine. Appending the files is more difficult as you need to blank the BMTAG field on the records being added, then run the update after appending.

If a company will be running the ALERE Configurator without linking to ALERE Manufacturing, the BOM Files in ALERE are identical to those that would be in Manufacturing. If ALERE Accounting is linked to ALERE Manufacturing, it will use the BOM files in the Manufacturing company data directory. If ALERE Accounting Configurator is being run without links to ALERE Manufacturing or shopLink, the same BOM files as above will be found in the ALERE Accounting company data directory.

Specific information about converting VP Manufacturing Data to ALERE:

Copy MABILLxx database from VP Data to the new ALERE Data Directory.
Using FoxPro, open the MABILLxx database that you copied to your ALERE company data. (set to use exclusively), then on the command line, enter: **MODI STRU**

Delete all indexes.
Change the field names in the MABILLxx:
ASSM=ITEMPARENT
ITEM=ITEMCHILD
SEQNO=FINDNO
BEGDATE=ACTIVE
ENDDATE=INACTIVE
QTY=QTY_ASSY

On the FoxPro command line enter the following commands:
SELECT FROM MABILLxx WHERE FINDNO=0 INTO DBF TEMP1
USE?
open BMREV database in your ALERE company data.
APPEND FROM TEMP1
select the path to your modified MABILLxx
REPLACE ALL BOMTYPE WITH 1
REPLACE ALL ALOCFQ WITH 1
REPLACE ALL FGTYPE WITH 1
REPLACE ALL SOCCMETH WITH 1
REPLACE ALL SOCPMETH WITH 1
SELECT FROM MABILLxx WHERE FINDNO!=0 INTO DBF TEMP2
USE?
open BMSL database in your TIW company data.
APPEND FROM TEMP2
again, select the path to your modified MABILLxx
QUIT FoxPro

Go into ALERE Program and run TIW FILE UPDATE. This will populate Inactive dates and BOMTAGs.
In BOM> Utilities, run Mark Orphans for Delete.
Run PLAN to check for circular BOMs. If any are found, correct or delete.

ALERE Manufacturing (Work) Order Module

Work Orders

Often times it is not necessary to import Work Orders, as by running PLAN, you can have the system create these orders for you. If you are importing Orders, you will need to populate:

- SOHEADER (Order Header)
- SOROUTE (Order Route)
- SOMATER (Material List)
- SOTRAN (Processed Transactions)
- WOFGCFG (Configuration information if item is configured)

After importing the data, from Manager> Access FoxPro, run ‘DO IDUPDATE’. This will create the necessary records in IDLOTSN for Lot or Serial Numbered items. This needs to be run even if the company is not using Lot or Serial Numbers.

After creating the orders from PLAN, use the Mass Transaction screen to mass FPO all of the necessary orders.

Data Flow on Work Orders

Let’s look at the flow of data when a Work Order is processed.

- Adding an order updates SOHEADER, SOROUTE, SOMATER. If configured, WOF-GCFG.
- FPO updates SOHEADER, SOMATER, IDLOTSN and Inventory Control for Allocations

and Order amounts.

- Transactions and Completions of orders update SOHEADER, SOROUTE, SOMATER, SOTRAN and tables listed under Inventory Adjustments. Also IDLOTSN if transaction is Lot or Serial Item.
- Archiving orders generally takes info from the SOxx tables to SOYxx table names. Exception, IDLOTSN goes to SOYIDLSN. SOMATER and WOFGCFG are not saved on archive.

ALERE Manufacturing Route Module

Route Steps

Again there are only two tables that generally need to be imported

- ROHEADER carries the Route Header information
- RODETAIL contains the individual Route Steps.

Note that while there is a Revision Number for the Route Header, unlike BOMs, ALERE Manufacturing does not track multiple revisions of the Route. This is not necessary as each Work Order contains a complete copy of the Route that was used for that order.

Route Material

Most companies will assign all the material on a Work Order to the first Route Operation. If this is the case, the system will automatically assign material the first time an order is processed for a particular Item and Route. This information is stored in ROMATER.

Route to Item Linkage

Setting up the Route that will be used by default for any particular item is very simple. Populate APNRN with the Parent BOM Item Number in PARTNO and the Route in ROUTENO. This is also important if the customer wants to include Route information in the Costed BOM Report.

Route Library

It may also be requested to populate ROLIB which provides the Route Library function. This information is basically the same as the RODETAIL information with an indentifier tag.

ALERE Manufacturing Machine Module

Machines

The setup of these modules is simple enough that most companies will manually enter the data. Work Centers could be created by imports to MALC. Note that the Start Time and Hours available are all entered in seconds from midnight. It is easiest to leave these fields blank, and use the utility CHANGE WORK CENTERS to assign times to them in mass.

Closed Dates

The Closed Dates are simply stored in MAEXEMPT. Again the times are seconds from Midnight. So if you are closed all day the EXTSTART will be 0, and the EXAVAIL will be 86400

ALERE Manufacturing Cost Module

Labor Grades

Labor Grades are set up in LABORGR. It is important to note that every labor grade that is set up must have an employee record of $STANDARD set for that grade before it can be used.

Product Classes

Product Classes are set up in PRODCL. A Product Class should be set up for every Product Class used in the accounting package for manufactured items.

ALERE Manufacturing Plan and Schedule Modules

These are process modules, so there is no import of data necessary.

ALERE Manufacturing General

Reconcile Balances

The Reconcile Balances Utility in ALERE Manufacturing looks at the SOMATER and SALESMTL databases to determine what should be Allocated and On Order in linked accounting packages. It will update the accounting tables to match this information. Note that if ALERE Manufacturing is linked to ALERE Accounting, the SALESMTL and SALESCFG databases will be in the ALERE Company Data. The files of the same name in the ALERE Manufacturing Data will not be used.

Converting data from ALERE Accounting LAN to ALERE Accounting CS

If converting from another accounting system, our conversion programs will put the new ALERE data into standard LAN tables. After converting the data, and testing it (It is generally easier to test and correct issues in LAN), you are ready to convert to the Client Server file structure. Make sure that you have Microsoft SQL Server 2000 or newer or the free MSDE (Microsoft SQL Server Desktop Engine (having a 2GB data limit)) installed before running the conversion. Also make sure you have set up the ODBC Data Source. (See ALERE Manual)

Enter ALERE, in the company you wish to convert. Go to Manager> System Tools> Access FoxPro. On the command line enter:

```
DO LanToCS WITH “<SQLDBNAME>”
```

Where <SQLDBNAME> is the name of the SQL Server database that will receive the converted data. When prompted, select the path of the LAN data to be converted. After conversion make sure that all users have appropriate rights to the data. Follow the instructions in the ALERE Manual to complete the link to the converted data.

If running ALERE Accounting and Manufacturing, starting with Manufacturing v10.0, if you run accounting in SQL, you need to set up the MFG in SQL as well. Use the above procedure to convert the MFG data to a SQL database.

General

Remember that there are always current Data Dictionaries in the ALERE packages to help you with conversions or modifications.

<u>YOU MUST USE THE SUPPLIED CONVERSION DATA SETS WITH OUR CONVERSION PROGRAMS OR YOU WILL GET ERRORS!</u>

<u>DID YOU MAKE BACKUPS???</u>

Contact us

If you have specific questions about conversions, please contact TIW Technical Services.

By email at: tech@tiwcorp.com
By fax at: 610-258-6217
By phone at: 610-258-5161.

We are always happy to receive any suggestions you may have or hints on how you have made this process easier.

Version Specific Data Changes that need to be made:

VP Version specific data issues:

VP9, add the following fields:

ARGLAC.FINCH	C10
ARGLAC.CREDIT	C10
ARCUST.FAXNO	C25
ARYITR.MODULE	C2
ARITRN.MODULE	C2
ARRECR.TAXABLE	C1
GLENTR.GLSTAT	C1
GLYENT.GLSTAT	C1
ARCASH.GLACCNT	C10
ARYCSH.GLACCNT	C10

ARADRS file, all fields
Add these to the data to be converted.

Pro Version specific data issues:

Pro 5 and earlier add the fields:

ARCUST.URL	M
APVEND.URL	M

Pro 7.0 and higher:

ARCUST.EMAIL	Convert from Memo to C40
APVEND.EMAIL	Convert from Memo to C40

Pro 7.4 and higher:
Periods get truncated in conversion. The easiest way to fix is just to correct period ID's in ALERE GLPERIOD after conversion. If they are on a Calendar year, you might want to copy and modify the GLPERIOD table from the sample company.

add the following fields:

ARMAST.CURRENT	C1
SOMAST.CURRENT	C1
APMAST.CURRENT	C1
POMAST.CURRENT	C1

Additional Reference:
See ALSO the Data Dictionary in the ALERE Manager Module!
ALERE Accounting Table Mappings:

Key table pairings (one to many mapping)

ApBills.DocID + ApBills.DocType -> ApFromPo.DocID + ApFromPo.DocType
ApCash.CashID -> ApCashTr.CashID
ApHeader.DocID + ApHeader.DocType -> ApLines.DocID + ApLines.DocType
BmRev.ItemParent + BmRev.Rev -> BmSL.ItemParent + BmSL.Rev
GLHeader.JournlID -> GLPost.JournlID
ImHeader.DocID -> ImLines.DocID
ImMaster.Item -> ImLocn.Item
ImMaster.Item -> ImSource.Item
ImLocn.Item + ImLocn.LocID -> ImStock.Item + ImStock.LocID
ImStock.Item + ImStock.StockID -> ImSerial.Item + ImSerial.StockID
SlCash.CashID -> SlCashTr.CashID
SlHeader.DocID + SlHeader.DocType -> SlLines.DocID + SlLines.DocType

ApApprove

ApApprove.DocID + ApApprove.DocType -> ApBills.DocID + ApBills.DocType
ApApprove.CoID + ApApprove.LocID -> CoAddr.CoID + CoAddr.LocID
ApApprove.CoID + ApApprove.LocID -> ApSupl.CoID + ApSupl.LocID
(if ApApprove.DocType = "AP", "DM", "PP")
ApApprove.CoID + ApApprove.LocID -> SlCust.CoID + SlCust.LocID
(if ApApprove.DocType = "SP")

ApBills

ApBills.PayTo + ApBills.PayLocn -> ApSupl.CoID + ApSupl.LocID
ApBills.PayTo + ApBills.PayLocn -> CoAddr.CoID + CoAddr.LocID
ApBills.Terms -> CoTerms.TermID
ApBills.GLAP -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
ApBills.OriginID + ApBills.Origin -> ApHeader.DocID + ApHeader.DocType
(if ApBills.Origin = "BP", "PO", "PR", "QP" or "RP")
ApBills.OriginID + ApBills.Origin -> SlHeader.DocID + SlHeader.DocType
(if ApBills.Origin = "CM")

ApCash

ApCash.CashID -> ApCashTr.CashID
ApCash.CashAcct -> CoBank.Account
ApCash.CashAcct -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
ApCash.JournlID -> GLHeader.JournlID
ApCash.CoID + ApCash.CoLocn -> CoAddr.CoID + CoAddr.LocID
ApCash.CoID + ApCash.CoLocn -> ApSupl.CoID + ApSupl.LocID
(if any associated ApCashTr.DocType = "AP", "DM" or "PP")
ApCash.CoID + ApCash.CoLocn -> SlCust.CoID + SlCust.LocID
(if any associated ApCashTr.DocType = "SP")

ApCashTr

ApCashTr.CashID -> ApCash.CashID
ApCashTr.DocID + ApCashTr.DocType -> ApBills.DocID + ApBills.DocType

ApFromPo

ApFromPo.DocID + ApFromPo.DocType -> ApBills.DocID + ApBills.DocType
ApFromPo.Item -> ImMaster.Item
ApFromPo.DocID + ApFromPo.DocType -> ApHeader.DocID + ApHeader.DocType
ApFromPo.LineNum -> ApLines.LineNum
(where associated ApBills.OriginID + ApBills.Origin = ApLines.DocID + ApLines.DocType)

ApHeader
ApHeader.RecvAt + ApHeader.RecvLocn -> CoAddr.CoID + CoAddr.LocID
ApHeader.PoTo + ApHeader.PoLocn -> CoAddr.CoID + CoAddr.LocID
ApHeader.PayTo + ApHeader.PayLocn -> ApSupl.CoID + ApSupl.LocID
ApHeader.PayTo + ApHeader.PayLocn -> CoAddr.CoID + CoAddr.LocID
ApHeader.ShipVia + “VIA” -> CoCodes.CodeName + CoCodes.CodeType
ApHeader.FOB + “FOB” -> CoCodes.CodeName + CoCodes.CodeType
ApHeader.Terms -> CoTerms.TermID
ApHeader.TaxDist -> CoTaxTbl.TaxDist
ApHeader.Buyer + “BUY” -> CoCodes.CodeName + CoCodes.CodeType
ApHeader.OriginID + ApHeader.Origin -> ApHeader.DocID + ApHeader.DocType
(if ApHeader.Origin = “BP”, “PO”, “PR”, “QP” or “RP”)
ApHeader.OriginID + ApHeader.Origin -> SlHeader.DocID + SlHeader.DocType
(if ApHeader.Origin = “SO” or “SR”)
ApHeader.RlsGroup + “RLS” -> CoCodes.CodeName + CoCodes.CodeType
ApHeader.RefTo + ApHeader.RefLocn -> CoAddr.CoID + CoAddr.LocID

ApLines
ApLines.DocID + ApLines.DocType -> ApHeader.DocID + ApHeader.DocType
ApLines.OriginLi -> ApLines.LineNum
(where associated ApHeader.OriginID + ApHeader.Origin = ApLines.DocID + ApLines.DocType)
ApLines.Item -> ImMaster.Item
ApLines.GLInvtID +”ITM” -> CoCodes.CodeName + CoCodes.CodeType
ApLines.SuplItem -> ImSource.SuplItem
(where associated ApHeader.PoTo + ApHeader.PoLocn = ImSource.CoID + ImSource.LocID)
ApLines.BuyUM + “UOM” -> CoCodes.CodeName + CoCodes.CodeType
ApLines.GLAcrPay -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)

ApSupl
ApSupl.CoID + ApSupl.LocID -> CoAddr.CoID + CoAddr.LocID
ApSupl.SuplCode + “SUP” -> CoCodes.CodeName + CoCodes.CodeType
ApSupl.Terms -> CoTerms.TermID
ApSupl.DistID + “S” -> GLDist.DistID + GLDist.DistType
ApSupl.PoToID + ApSupl.PoToLoc -> CoAddr.CoID + CoAddr.LocID

BmCfgSave
BmCfgSave.BomTag -> BmRev.BomTag

BmFgCfg
BmFgCfg.ItemPart -> ImMaster.Item
BmFgCfg.ItemPart + BmFgCfg.Rev -> BmRev.ItemParent + BmRev.Rev
BmFgCfg.BomTag -> BmRev.BomTag

BmRev
BmRev.ItemParent -> ImMaster.Item
BmRev.FGParent + BmRev.Rev -> BmRev.ItemParent + BmRev.Rev

BmSL
BmSL.ItemParent -> ImMaster.Item
BmSL.ItemParent + BmSL.Rev -> BmRev.ItemParent + BmRev.Rev

BmSL.ItemChild -> ImMaster.Item

CoActLog

CoActLog.CoID -> CoAddr.CoID (where CoAddr.MainLocn=.T.)
CoActLog.DocID + CoActLog.DocType -> ApBills.DocID + ApBills.DocType
(if CoActLog.DocType = “AP” or “DM”)
CoActLog.DocID + CoActLog.DocType -> ApHeader.DocID + ApHeader.DocType
(if CoActLog.DocType = “BP”, “PO”, “PR”, “QP” or “RP”)
CoActLog.DocID + CoActLog.DocType -> SlHeader.DocID + SlHeader.DocType
(if CoActLog.DocType = “AR”, “BS”, “CM”, “QS”, “RS”, “SO” or “SR”)
CoActLog.SourceID + CoActLog.Source -> ApBills.DocID + ApBills.DocType
(if CoActLog.Source = “AP” or “DM”)
CoActLog.SourceID + CoActLog.Source -> ApHeader.DocID + ApHeader.DocType
(if CoActLog.Source = “BP”, “PO”, “PR”, “QP” or “RP”)
CoActLog.SourceID + CoActLog.Source -> SlHeader.DocID + SlHeader.DocType
(if CoActLog.Source = “AR”, “BS”, “CM”, “QS”, “RS”, “SO” or “SR”)

CoAddr

CoAddr.RemarkID + “REM” -> CoCodes.CodeName + CoCodes.CodeType
CoAddr.RevLocn -> CoAddr.LocID (where CoAddr.IsMyCo = .T.)
CoAddr.FOB +”FOB” -> CoCodes.CodeName + CoCodes.CodeType
CoAddr.Buyer + “BUY” -> CoCodes.CodeName + CoCodes.CodeType
CoAddr.ShipFrom -> CoAddr.LocID (where CoAddr.IsMyCo = .T.)
CoAddr.ShipVia + “VIA” -> CoCodes.CodeName + CoCodes.CodeType
CoAddr.ShipFOB + “FOB” -> CoCodes.CodeName + CoCodes.CodeType
CoAddr.TaxDist -> CoTaxTbl.TaxDist
CoAddr.SalesMan + “SLS” -> CoCodes.CodeName + CoCodes.CodeType
CoAddr.GLSaleCo “GLS” -> CoCodes.CodeName + CoCodes.CodeType
CoAddr.BillID + CoAddr.BillLoc -> CoAddr.CoID + CoAddr.LocID
CoAddr.BillID + CoAddr.BillLoc -> SlCust.CoID + SlCust.LocID
CoAddr.RemitID + CoAddr.RemitLoc -> ApSupl.CoID + ApSupl.LocID
CoAddr.RemitID + CoAddr.RemitLoc -> CoAddr.CoID + CoAddr.LocID
CoAddr.AgentID + CoAddr.AgentLoc -> CoAddr.CoID + CoAddr.LocID
CoAddr.CoStatus + “COS” -> CoCodes.CodeName + CoCodes.CodeType

CoBank

CoBank.Account -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)

CoCheck

CoCheck.GLAcct -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)

CoTaxTbl

CoTaxTbl.State -> CoTxRate.RateID
CoTaxTbl.County -> CoTxRate.RateID
CoTaxTbl.City -> CoTxRate.RateID
CoTaxTbl.Local1 -> CoTxRate.RateID
CoTaxTbl.Local2 -> CoTxRate.RateID
CoTaxTbl.Local3 -> CoTxRate.RateID

CoTxRate

CoTxRate.Account -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)

CoUnits

CoUnits.UM1 + “UOM” -> CoCodes.CodeName + CoCodes.CodeType

CoUnits.UM2 + “UOM” -> CoCodes.CodeName + CoCodes.CodeType

GLCustom

GLCustom.CustGr -> CoAddr.CoID (if GLCustom.SalesCode = “S”)
GLCustom.CustGr + “GLS” -> CoCodes.CodeName + CoCodes.CodeType
(if GLCustom.ItemCode = “G”)
GLCustom.Location -> CoAddr.LocID (where IsMyCo = .T.)
GLCustom.ItemGr -> ImMaster.Item (if GLCustom.ItemCode = “S”)
GLCustom.ItemGr + “ITM” -> CoCodes.CodeName + CoCodes.CodeType
(if GLCustom.ItemCode = “G”)
GLCustom.GLAcrPay -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLCustom.GLAR -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLCustom.GLCOGS -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLCustom.GLCPPay -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLCustom.GLInvt -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLCustom.GLIClear -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLCustom.Rev -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLCustom.PurVa -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLCustom.SDisc -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)

GLDeflt

GLDeflt.GLAR -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLDeflt.GLCash -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLDeflt.GLCFC -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLDeflt.GLInTran -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLDeflt.GLInvt -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLDeflt.GLIClear -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLDeflt.GLVPPay -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLDeflt.GLAP -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLDeflt.GLCPPay -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLDeflt.GLTaxOwe -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLDeflt.GLRev -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLDeflt.GLSDisc -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLDeflt.GLRetain -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLDeflt.GLCOGS -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLDeflt.GLAcrPay -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLDeflt.GLPDisc -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLDeflt.GLPurVar -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLDeflt.GLSaleCh -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLDeflt.GLTaxExp -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLDeflt.GLCShip -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLDeflt.GLVShip -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLDeflt.GLBankFee -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)

GLDist

GLDist.GLAP -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)

GLErrLog

GLErrLog.DocID + GLErrLog.DocType -> ApBills.DocID + ApBills.DocType
(if GLErrLog.DocType = “AP” or “DM”)
GLErrLog.DocID + GLErrLog.DocType -> ApHeader.DocID + ApHeader.DocType
(if GLErrLog.DocType = “BP”, “PO”, “PR”, “QP” or “RP”)
GLErrLog.DocID + GLErrLog.DocType -> SlHeader.DocID + SlHeader.DocType
(if GLErrLog.DocType = “AR”, “BS”, “CM”, “QS”, “RS”, “SO” or “SR”)

GLErrLog.Account -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)

GLHeader

GLHeader.DocID + GLHeader.DocType -> ApBills.DocID + ApBills.DocType
(if GLHeader.DocType = “AP” or “DM”)
GLHeader.DocID + GLHeader.DocType -> ApHeader.DocID + ApHeader.DocType
(if GLHeader.DocType = “BP”, “PO”, “PR”, “QP” or “RP”)
GLHeader.DocID + GLHeader.DocType -> SlHeader.DocID + SlHeader.DocType
(if GLHeader.DocType = “AR”, “BS”, “CM”, “QS”, “RS”, “SO” or “SR”)
GLHeader.DocID -> ImlHeader.DocID (if GLHeader.DocType = “TO”)
GLHeader.DocID -> ImActLog (if GLHeader.DocType = “IA”, “IR” or “OS”)
GLHeader.DocID -> SlCash.CashID (if GLHeader.DocType = “IP”)
GLHeader.DocID -> ApCash.CashID (if GLHeader.DocType = “OP”)
GLHeader.DocID -> GLHeader.JournlID (if GLHeader.DocType = “JE” or “YE”)

GLItem

GLItem.JournlID -> GLHeader.JournlID
GLItem.Item -> ImMaster.Item

GLMerge

GLMerge.CoID -> CoAddr.CoID (where CoAddr.IsMyCo = .T. AND CoAddr.MainLocn = .T.)

GLPost

GLPost.JournlID -> GLHeader.JournlID
GLPost.Account -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)

GLTotals

GLTotals.Account -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
GLTotals.PeriodID + GLTotals.FiscalYr -> GLPeriod.PeriodID + GLPeriod.FiscalYr

ImActLog

ImActLog.DocID + ImActLog.DocType -> ApBills.DocID + ApBills.DocType
(if ImActLog.DocType = “AP” or “DM”)
ImActLog.DocID + ImActLog.DocType -> ApHeader.DocID + ApHeader.DocType
(if ImActLog.DocType = “BP”, “PO”, “PR”, “QP” or “RP”)
ImActLog.DocID + ImActLog.DocType -> SlHeader.DocID + SlHeader.DocType
(if ImActLog.DocType = “AR”, “BS”, “CM”, “QS”, “RS”, “SO” or “SR”)
ImActLog.DocID -> ImlHeader.DocID (if ImActLog.DocType = “TO”)
ImActLog.DocID + ImActLog.DocType + ImActLog.LineNum ->
ApLines.DocID + ApLines.DocType + ApLines.LineNum
(if ImActLog.DocType = “BP”, “PO”, “PR”, “QP” or “RP”)
ImActLog.DocID + ImActLog.DocType -+ ImActLog.LineNum ->
SlLines.DocID + SlLines.DocType + SlLines.LineNum
(if ImActLog.DocType = “AR”, “BS”, “CM”, “QS”, “RS”, “SO” or “SR”)
ImActLog.DocID -> ImlHeader.DocID + ImActLog.LineNum (if ImActLog.DocType = “TO”)
ImActLog.Item -> ImMaster.Item
ImActLog.Item + ImActLog.StockID -> ImStock.Item + ImStock.StockID
ImActLog.LocID -> CoAddr.LocID (where CoAddr.IsMyCo = .T.)
ImActLog.CoID -> CoAddr.CoID (where CoAddr.N\MainLocn = .T.)

ImAlloc

ImAlloc.DocID + ImAlloc.DocType -> ApHeader.DocID + ApHeader.DocType
(if ImAlloc.DocType = “PO” or “PR”)
ImAlloc.DocID + ImAlloc.DocType -> SlHeader.DocID + SlHeader.DocType

(if ImAlloc.DocType = “AR”, “CM”, “SO” or “SR”)
ImAlloc.DocID -> ImlHeader.DocID (if ImAlloc.DocType = “TO”)
ImAlloc.DocID + ImAlloc.DocType + ImAlloc.LineNum ->
ApHeader.DocID + ApHeader.DocType + ApHeader.LineNum
(if ImAlloc.DocType = “PO” or “PR”)
ImAlloc.DocID + ImAlloc.DocType + ImAlloc.LineNum ->
SlHeader.DocID + SlHeader.DocType + SlHeader.LineNum
(if ImAlloc.DocType = “AR”, “CM”, “SO” or “SR”)
ImAlloc.DocID+ ImAlloc.LineNum -> ImlHeader.DocID + ImHeader.LineNum
(if ImAlloc.DocType = “TO”)
ImAlloc.Item -> ImMaster.Item
ImAlloc.Item + ImAlloc.StockID -> ImStock.Item + ImStock.StockID
ImAlloc.LocID -> CoAddr.LocID (where CoAddr.IsMyCo = .T.)

ImHeader

ImHeader.ShipLocn -> CoAddr.LocID (where CoAddr.IsMyCo = .T.)
ImHeader.RecvLocn -> CoAddr.LocID (where CoAddr.IsMyCo = .T.)
ImHeader.ShipVia + “VIA” -> CoCodes.CodeName + CoCodes.CodeType

ImLines

ImLines.DocID -> ImHeader.DocID
ImLines.Item -> ImMaster.Item
ImLines.ShipFrom -> CoAddr.LocID (where CoAddr.IsMyCo = .T.)
ImLines.StockUM + “UOM” -> CoCodes.CodeName + CoCodes.CodeType
ImLines.GLInvtID + “ITM” -> CoCodes.CodeName + CoCodes.CodeType

ImLocn

ImLocn.Item -> ImMaster.Item
ImLocn.LocID -> CoAddr.LocID (where CoAddr.IsMyCo = .T.)
ImLocn.Supplier + ImLocn.SuplLocn -> CoAddr.CoID + CoAddr.LocID
ImLocn.Supplier + ImLocn.SuplLocn -> ApSupl.CoID + ApSupl.LocID
ImLocn.Item + ImLocn.Supplier + ImLocn.SuplLocn -> ImSource.Item + ImSource.CoID + ImSource.LocID
ImLocn.Item + ImLocn.DefStore + ImLocn.DefBin -> ImStock.Item + ImStock.Store + ImStock.Bin
ImLocn.SellUM + “UOM” -> CoCodes.CodeName + CoCodes.CodeType

ImMaster

ImMaster.ProdClas + “CLS” -> CoCodes.CodeName + CoCodes.CodeType
ImMaster.StockUM + “UOM” -> CoCodes.CodeName + CoCodes.CodeType
ImMaster.BuyUM + “UOM” -> CoCodes.CodeName + CoCodes.CodeType
ImMaster.SellUM + “UOM” -> CoCodes.CodeName + CoCodes.CodeType
ImMaster.PriceGrp + “ITM” -> CoCodes.CodeName + CoCodes.CodeType
ImMaster.GLInvGr + “GLI” -> CoCodes.CodeName + CoCodes.CodeType
ImMaster.ItemType + “TYP” -> CoCodes.CodeName + CoCodes.CodeType

ImSerial

ImSerial.Item -> ImMaster.Item
ImSerail.LocID -> CoAddr.LocID (where CoAddr.IsMyCo = .T.)
ImSerial.Item + ImSerial.LocID -> ImLocn.Item + ImLocn.LocID
ImSerial.Item + ImSerial.Stock.StockID -< ImStock.Item + ImStock.StockID
ImSerial.Item + ImSerial.RecvLog -> ImActLog.Item + ImActLog.SNLogID
(to be changed to ImActlog.Item + ImActlog.LogID)

ImSerial.Item + ImSerial.ShipLog -> ImActLog.Item + ImActLog.SNLogID
(to be changed to ImActlog.Item + ImActlog.LogID)

ImSnap

ImSnap.Item -> ImMaster.Item
ImSnap.StockUM + “UOM” -> CoCodes.CodeName + CoCodes.CodeType

ImSource

ImSource.Item -> ImMaster.Item
ImSource.CoId + ImSource.LocID -> ApSupl.CoID + ApSupl.LocID
ImSource.CoId + ImSource.LocID -> CoAddr.CoID + CoAddr.LocID
ImSource.BuyUM + “UOM” -> CoCodes.CodeName + CoCodes.CodeType

ImStock

ImStock.Item -> ImMaster.Item
ImStock.LocID -> CoAddr.LocID (where CoAddr.IsMyCo = .T.)

ImTier

ImTier.Item -> ImMaster.Item

SalesCfg

SalesCfg.SalesNo + SalesCfg.DocType -> SlHeader.DocID + SlHeader.DocType
SalesCfg.SalesNo + SalesCfg.DocType + SalesCfg.LineNum ->
SlLines.DocID + SlLines.DocType + SlLines.LineNum
SalesCfg.AItem -> ImMaster.Item

SalesMtl

SalesMtl.SalesNo + SalesMtl.DocType -> SlHeader.DocID + SlHeader.DocType
SalesMtl.SalesNo + SalesMtl.DocType + SalesMtl.LineNum ->
SlLines.DocID + SlLines.DocType + SlLines.LineNum
SalesMtl.PartNo -> ImMaster.Item

SlCash

SlCash.CashID -> SlCashTr.CashID
SlCash.CashAcct -> CoBank.Account
SlCash.CashAcct -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)
SlCash.JournlID -> GLHeader.JournlID
SlCash.CoID + SlCash.LocID -> CoAddr.CoID + CoAddr.LocID
SlCash.CoID + SlCash.LocID -> ApSupl.CoID + ApSupl.LocID
(if any associated SlCashTr.DocType = “AP”, “DM” or “PP”)
SlCash.CoID + SlCash.LocID -> SlCust.CoID + SlCust.LocID
(if any associated SlCashTr.DocType = “AR”, “CM”, “SO” or “SP”)
SlCash.LinkID -> SlCash.CashID (if SlCash.BankStat = “C” or “O”)
SlCash.LinkID -> GlHeader.JournlID (if SlCash.BankStat = “X”)

SlCashTr

SlCashTr.CashID -> SlCash.CashID
SlCashTr.DocID + SlCashTr.DocType -> ApBills.DocID + ApBills.DocType
(if SlCashTr.DocType = “AP”, “DM” or “PP”)
SlCashTr.DocID + SlCashTr.DocType -> SlHeader.DocID + SlHeader.DocType
(if SlCashTr.DocType = “AR”, “CM”, “SO” or “SP”)

SlCust

SlCust.CoID + SlCust.LocID -> CoAddr.CoID + CoAddr.LocID
SlCust.PrLevel + “CPL” -> CoCodes.CodeName + CoCodes.CodeType

SlCust.Indusrty + “IND” -> CoCodes.CodeName + CoCodes.CodeType
SlCust.Code1 + “CD1” -> CoCodes.CodeName + CoCodes.CodeType
SlCust.Code2 + “CD2” -> CoCodes.CodeName + CoCodes.CodeType
SlCust.Code3 + “CD3” -> CoCodes.CodeName + CoCodes.CodeType
SlCust.Code4 + “CD4” -> CoCodes.CodeName + CoCodes.CodeType
SlCust.ShipId + SlCust.ShipLoc -> CoAddr.CoID + CoAddr.LocID
SlCust.Terms -> CoTerms.TermID

SlHeader

SlHeader.BillTo + SlHeader.BillLocn -> CoAddr.CoID + CoAddr.LocID
SlHeader.BillTo + SlHeader.BillLocn -> SlCust.CoID + SlCust.LocID
SlHeader.ShipTo + SlHeader.ShipLocn -> CoAddr.CoID + CoAddr.LocID
SlHeader.ShipVia +”VIA” -> CoCodes.CodeName + CoCodes.CodeType
SlHeader.FOB +”FOB” -> CoCodes.CodeName + CoCodes.CodeType
SlHeader.Terms -> CoTerms.TermID
SlHeader.OriginID + SlHeader.Origin -> SlHeader.DocID + SlHeader.DocType
SlHeader.SalesMan + “SLS” -> CoCodes.CodeName + CoCodes.CodeType
SlHeader.ShipFrom -> CoAddr.LocID (where CoAddr.IsMyCo = .T.)
SlHeader.GLSaleID +”GLS” -> CoCodes.CodeName + CoCodes.CodeType
SlHeader.RlsGroup + “RLS” -> CoCodes.CodeName + CoCodes.CodeType
SlHeader.RefTo + SlHeader.RefLocn -> CoAddr.CoID + CoAddr.LocID

SlLines

SlLines.DocID + SlLines.DocType -> SlHeader.DocID + SlHeader.DocType
SlLines.Item -> ImMaster.Item
SlLines.ShipID + SlLines.ShipLocn -> CoAddr.CoID + CoAddr.LocID
SlLines.SellUM + “UOM” -> CoCodes.CodeName + CoCodes.CodeType
SlLines.GLInvtID + “GLI” -> CoCodes.CodeName + CoCodes.CodeType
SlLines.GLAcct -> GlChart.AcctNo (where GlChart.RecType = 1, 2, 3, 4, 5 or 6)

SlPrice

SlPrice.Customer -> CoAddr.CoID (where CoAddr.MainLocn = .T.)
(if SlPrice.CustCode = “S”)
SlPrice.Customer + “CPL” -> CoCodes.CodeName + CoCodes.CodeType
(if SlPrice.CustCode = “L”)
SlPrice,PartGr -> ImMaster.Item
(if SlPrice.PartCode = “S”)
SlPrice.PartGr + “ITM” -> CoCodes.CodeName + CoCodes.CodeType
(if SlPrice.PartCode = “G”)
SlPrice.Location -> CoAddr.LocID (where CoAddr.IsMyCo = .T.)