

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

Jerry Devore
Accenture
<http://www.accenture.com>

Douglas A. Volz
Douglas Volz Consulting, Inc.
www.volzconsulting.com

Introduction

With the development of Release 12 many new features have been introduced. For example, shared services are easier to implement due to having user responsibilities that can span operating units as well as common definitions for bank accounts and tax rules. But what about some long-standing challenges with inter-company reporting and accounting requirements for inventory? Is it better than R11i? What gaps still exist with Subledger Accounting, local vs. corporate financial reporting, inter-company COGS and profit in inventory? Learn from the Release 12 experiences of Douglas Volz Consulting and CSC Consulting; Jerry DeVore has been active in Supply Chain and Cost industry roles for over 25 years, implementing Oracle over the past seven years. Doug Volz is a former Cost Accounting Manager who co-designed Oracle Cost Management and leads the OAUG MFG Discrete SIG Cost Group, he has over 20 years with the Oracle applications and 30 years combined industry, design and consulting experience.

For further information, Doug may be reached at doug@volzconsulting.com (or www.volzconsulting.com) and Jerry at jerry.devore@gmail.com. We welcome your comments and questions.

Scope of this Paper

This paper covers the following topics:

- Brief Discussion for Subledger Accounting
- Use of Multiple Ledgers (as related to intercompany accounting)
- Solutions for Accrual, Relief and Reporting for Profit in Inventory
- Solutions for booking and elimination of Inter-Company COGS

This paper addresses many of the underlying concepts for resolving inter-company accounting requirements. However this paper is not intended to be a full discourse on how to use Oracle Applications for inter-company accounting; the primary focus is inventory valuation and elimination of intercompany profit in inventory.

Background for Inter-Company Inventory Accounting

Oracle started to introduce profit in inventory concepts with Release 11.5.9. Starting with this point release you could accrue the profit from the sending organization into the receiving organization without some form of customization or manual journal entries. But the applications design has not been completed, as the Oracle Cost Management and related modules do not have the underlying logic in place to relieve the inter-company profit in inventory for when the goods are shipped to the customer or otherwise scrapped or disposed of. In addition you cannot report your profit in inventory based upon your onhand quantities. It simply does not exist. In Release 12 many new financial features were added but unfortunately this area continues to be a point of contention.

The example in this paper is from a Release 12.06 global pharmaceutical company. They use Discrete Costing with multiple inventory orgs, currencies, operating units and ledgers. Goods are transferred from the manufacturing sites

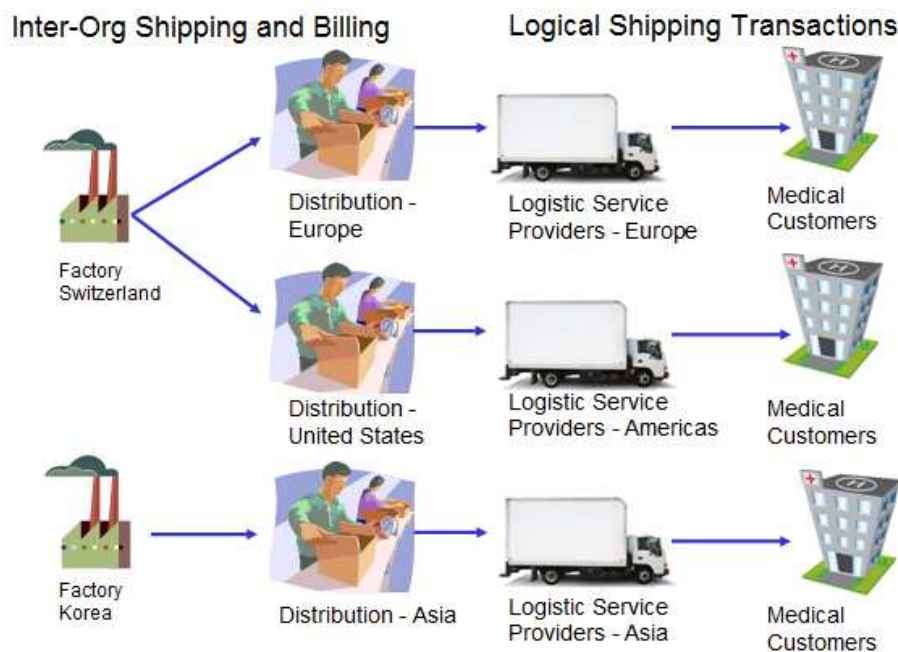
Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

to internal or 3rd party distribution centers using internal requisitions and internal sales orders. Due to achieving certain currency thresholds (or limits to the amount of currency in a Ledger), the example firm was able to use secondary ledgers for most local currency and financial reporting requirements. Inter-company pricing is done using the standard Oracle Advanced Pricing functionality, with normal requirements to report inventory at gross (including profit in inventory) and at net after removing profit in inventory. And they also have normal requirements to report and eliminate profit in inventory and eliminate inter-company cost of goods sold.

Before the creation of the described profit in inventory solution the example firm had significant difficulties recording and eliminating profit in inventory. Their auditors were becoming more and more impatient, with the possibility of several audit exceptions by year-end. The techniques in this paper gave them a reliable, auditable solution which satisfied both their internal management and their external audit requirements. This paper offers insights to how this was accomplished and what roadblocks had to be overcome.

Sample Supply Chain Background (example only):

In the below diagram pharmaceutical goods are manufactured at two locations, Switzerland and Korea, with each factory supplying the product including the API or *Active Pharmaceutical Ingredient* to the distribution centers, usually in capsule form. At the Distribution Centers the capsules are packaged for local markets with appropriate labeling and indicators. The goods are then shipped to the third party logistics service providers (LSPs), who in turn ship the goods to the end customers. Both the Distribution Centers and the LSPs are represented as Inventory Organizations, with internal requisitions and sales orders used to transfer the goods from the Factory to the Distribution Center to the LSPs. And because the LSPs are usually in different countries, using Logical Shipping Transactions for each cross-border movement enables accurate Intrastat Reporting.

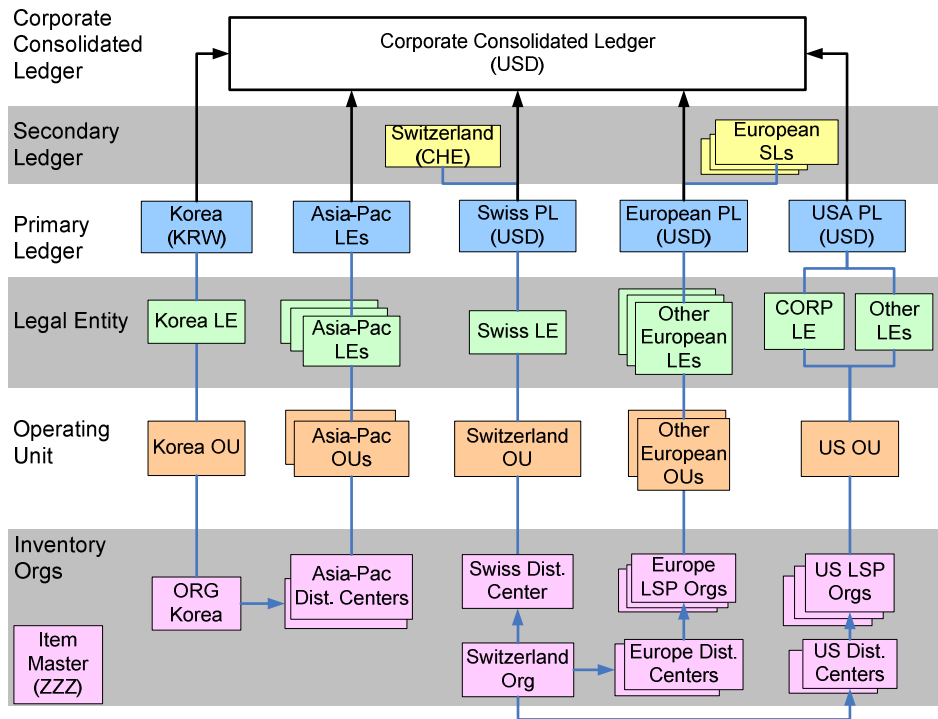


Sample Financial Relationships (example only):

In the below example, with the exception of Korea and other Asia-Pac Ledgers, all other primary ledgers are in USD. Doing so removes the currency conversion and reporting differences caused by foreign currency. This was accomplished with the standard mapping functionality from Subledger Accounting. Corporate reporting is simplified and secondary ledgers used for local reporting requirements.

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

Note: the example client was able to use secondary ledgers for local reporting requirements due to achieving certain currency threshold requirements. Your situation may be different.



Subledger Accounting and Release 12

Release 12 marks a significant departure from earlier releases. Many of the original design limitations, such as inability to journal easily across ledgers (called sets of books in earlier releases), difficult setup and use of shared service centers, separate business tax sub-systems, and separate customer and supplier masters have been resolved. Oracle Process Manufacturing (OPM) has been merged into the Inventory module, with added features for dual unit of measure control, advanced lot control, material status and enhanced inventory allocation rules brought into the inventory module... to name a few of the changes. Discrete Costing has better revenue recognition and deferred COGS capabilities and major enhancements to inventory accrual accounting and reporting.

With the introduction of Subledger Accounting (SLA) you now have one unified view of all the Oracle Application Subledgers across all modules with accounting entries. Whether you are looking for accounting entries from Receivables, Payables, Assets, Costing (Process and Discrete) and any other module that has accounting entries, they are now all in one place. All affected modules have one standardized "Create Accounting" process, standardized methods on when and how to transfer and post to the G/L and all subledgers use the Multi-Org Access Control (MOAC) features for access to multiple operating units and ledgers.

Key Inter-Company Concepts and Definitions

The following concepts and definitions may be helpful in understanding this paper.

Key Subledger Accounting Concepts and Definitions

Release 12 Ledgers and the Four C's

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

For Release 12 the concept of Sets of Books has been replaced with **Ledgers** and **Ledger Sets**. A **Ledger** is made up of one or more Legal Entities, and each legal entity represents the legal corporate structures or companies with a business group, as required for fiscal or tax reporting. Unlike Release 11i, Release 12 Operating Units are assigned to Ledgers and a default legal context. Multiple Operating Units and multiple Legal Entities can be assigned to the same Ledger.¹ **Ledger Sets** gives you the ability to group ledgers together and allow users to access all of the ledgers in the same group. Ledger sets is used by Multi-Org Access Control (MOAC). The use of ledger sets also streamlines the set up process; for example a single set of rules could be defined for all IFRS ledgers and attached to each ledger in turn, with the local GAAP ledgers having the appropriate country specific rules. The following actions can also be done by Ledger Sets: open/close periods, create journals, translate balances and report & view information².

Each ledger has four C's:

- Currency
- Chart of Accounts
- Calendar
- *Convention* (accounting convention)

Convention means the method of accounting for the transactions, whether it is cash or accrual, encumbrance or other accounting convention or method. Oracle provides a number of accounting conventions; with SLA you will copy a seeded one from Oracle and change it to meet your unique requirements. Having the four C's gives you the ability to have primary and secondary ledgers; with SLA you may have a primary ledger (such as a reporting currency ledger) and multiple secondary ledgers, each with different accounting conventions.

This ability to support different conventions happens both for the Release 12 subledger accounting entries and the primary and secondary ledgers. In doing so Release 12 provides better support for MRC – multiple reporting currencies – at both a general ledger and subledger level of detail. You can have any number of detailed subledgers for each of your reporting currency ledgers and map these to additional subledgers to multiple general ledgers (using different currencies as desired).

Profit in Inventory

When you sell goods to a customer you (hopefully) earn a profit on these goods. For example your cost may be \$10 per unit and you sell it for \$20 per unit, thereby making \$10 in profit. For global firms the Manufacturing Sites frequently supply the internal Distribution Centers. The Manufacturing Sites will “sell” the goods to the Distribution Centers at a profit, thereby correctly recording the local cost of goods sold, inter-company revenue and cost of goods sold. Likewise the Distribution Center receives the goods at the marked-up cost for the goods, reflecting the internal cost from the Manufacturing Site. This is all well and good for local reporting to your tax authorities as from a local tax perspective your local profit is based upon your local costs, regardless if the goods come from an outside supplier or from an internal supplier.

But when you consolidate, or roll up your local results into your Corporate General Ledger you must “eliminate” or remove the Inter-Company Profit in Inventory. Did your company really make \$10 each time you shipped goods from your Manufacturing Site to your Distribution Center? Of course not. You don't make money from a corporate, worldwide point of view until you sell your goods to an end-customer – someone who is not part of the same company as your firm.

This internal profit which is created by selling and shipping goods from one internal location to another is called Profit in Inventory (PII) or Intercompany Profit (ICP). The recording, recognition and relief of Profit in Inventory affects your overall Income and Balance Sheet statements and must be done correctly by accurate financial

¹ Paraphrased from *The Release 12 Primer*, chapter 2, Financials Product Family, page 18, Solution Beacon, LLC.

² Ibid, page 52.

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

reporting. Much of this paper will describe current functionality and solutions for deficiencies with this functionality.

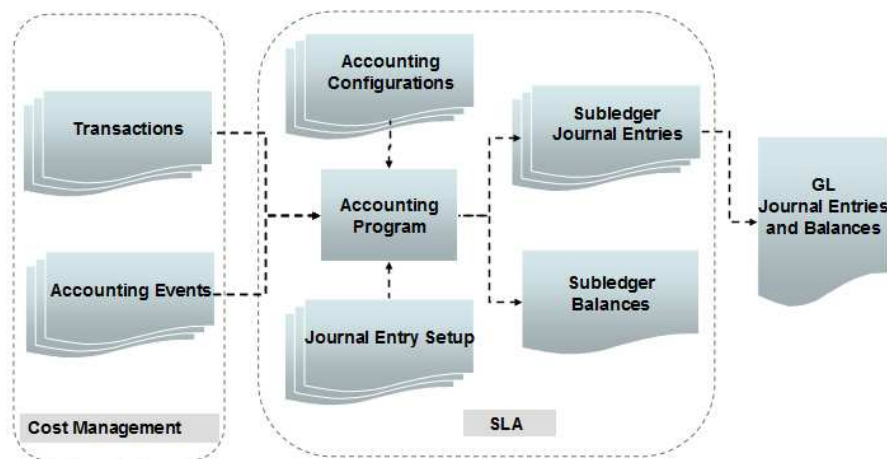
Wow – Release 12 Features that Work!

Subledger for Product Line Accounting, Variance Recognition, Intercompany Accounting

In Release 12 SLA provides a rule-based method for deriving the full accounting string or “Accounting Flexfield” across the accounting modules. For discrete costing, in prior releases there were partial and separate solutions for COGS (Account Generator) and Sales Revenue (Auto Accounting) which did not talk to each other. Many of the accounting entries for cost variances were not flexible. Now with SLA Cost Management has this built-in flexibility and uses the same rule engine to derive the full accounting string. This rule engine is called the Accounting Methods Builder and is an integral part of SLA.

In summary, SLA:

- Provides Rule-based accounting engine, toolset & repository (subledger) supporting both Oracle E-Business Suite and external modules
- Allows multiple accounting representations for a single business event, resolving conflicts between corporate and local fiscal accounting requirements
- Retains the detail and traceability back to the original source transactions, allowing different summarization options in the General Ledger, with full audit trail and reconciliation (in the subledger accounting model)
- Introduces a common data model and User Interface across subledgers, replaces various disparate 11i setups, providing single repository/subledger for financial and management analysis
- With the SLA flexibility over your Accounting Flexfield you can drive the account number information as needed by your intercompany requirements.

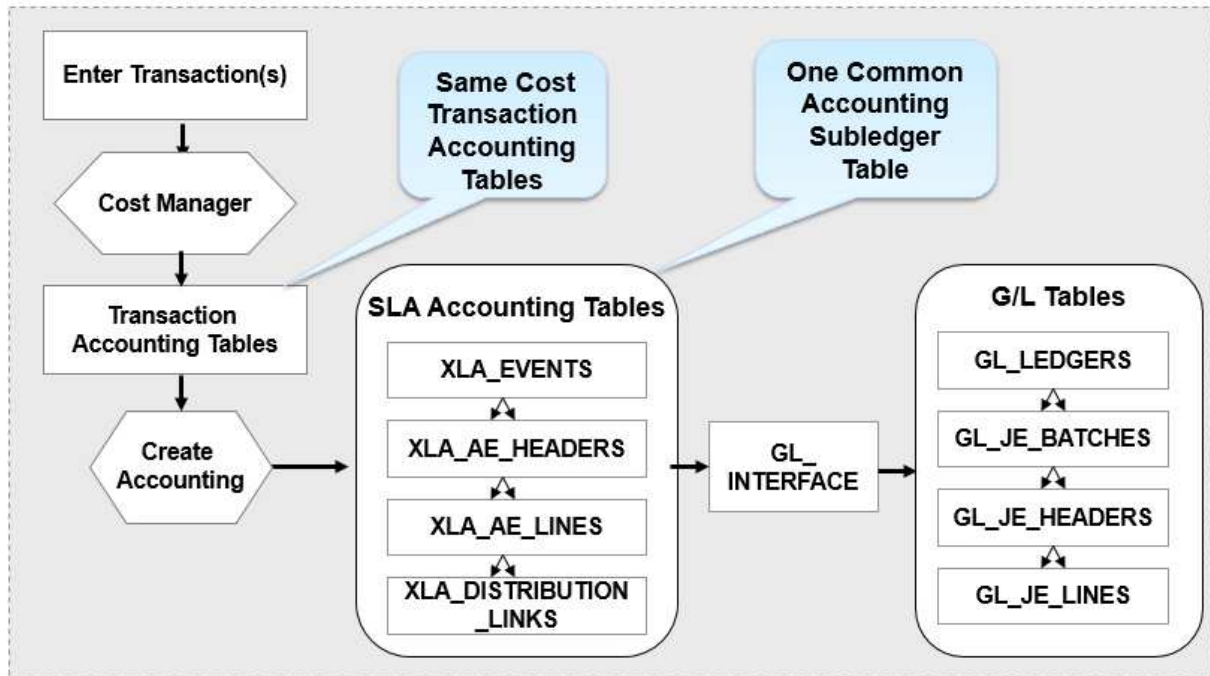


R12 Underlying Table Structure for Cost, SLA and G/L

With Release 12 there have been important changes to the underlying tables. First, there is a new table structure to support the SLA functionality, of which a small sample includes XLA_EVENTS, XLA_AE_HEADERS, XLA_AE_LINES and XLA_DISTRIBUTION_LINES. The XLA_EVENTS corresponds to (or mirrors) the Discrete and Process transactions. XLA_AE_HEADERS and XLA_AE_LINES hold the new subledger accounting entries. XLA_DISTRIBUTION_LINES stores the link between transactions and subledger journal entry lines. Note for the General Ledger the GL_SETS_OF_BOOKS table as been replaced with the GL_LEDGERS table.

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

Here is a conceptual diagram for many of the related tables:



COGS Recognition

In Release 12 the customer shipment entries go to deferred COGS. Oracle Cost Management credits inventory and credits a deferred COGS account, part of your Inventory Parameters setup. After you invoice your customers, you run the “Generate COGS Recognition Entries” program to transfer the balances from the deferred COGS to the final COGS account. This program uses the same percent as used for your revenue recognition, to also recognize your COGS amounts. When you recognize 100% of your revenue you will recognize 100% of your COGS, and this can happen over multiple accounting periods depending on how your customers accept your product shipments. And you can run the “Generate COGS Recognition Entries” program even after you close the inventory accounting period, as the steps for running it are:

- Ship product to your customer and create your normal Sales Order Issue/Return transactions
- Autoinvoice your customer and run Revenue Recognition Master Program in Oracle A/R
- Run the “Record Order Management Transactions” concurrent process to pick up and cost all of your uncosted sales order issue and RMA return transactions. It creates a record for each new order line in the costing COGS recognition matching table and creates the material accounting distributions for sale order issue and RMA return transactions
- Or run the Inventory Cost Processor, as it will automatically run the Record Order Management Transactions for you and create the material accounting distributions as well
- Then run “Generate COGS Recognition Entries” to recognize your true COGS

	Inventory	Deferred COGS	“True” COGS
1. Customer Shipment	10	10	
2. COGS Recognition		10	10

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

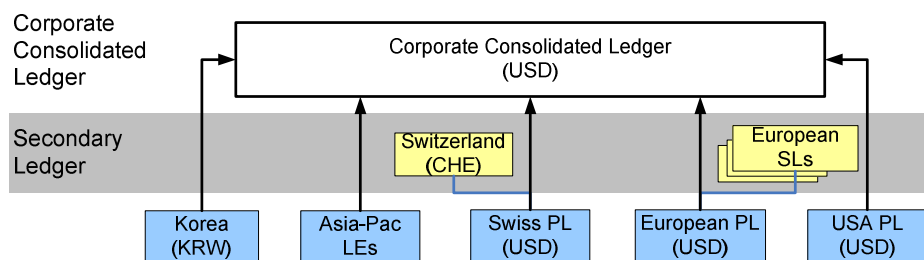
→ **Note:** We have had trouble with the Cost Cut-Off date and COGS Recognition programs, please ensure you have the latest patches for the COGS Recognition programs.

Primary and Secondary Ledgers

For legal and management reporting SLA supports multiple accounting requirements in a single instance. This is accomplished by applying different sets of rules to different sets of transactions or accounting for the same transaction with multiple methods (into multiple ledgers). For example you could have a reporting currency ledger for French reporting requirements (Le Plan Comptable – French National Accounting Code) using a French business calendar, French chart of accounts and the French interpretation of IFRS (International Financial Reporting Standards). And then you could define a secondary ledger for the French ledger, which will share the same currency COA, calendar and accounting method as those at HQ. And by using a combination of mapping sets and/or different Subledger Accounting Methods, you can convert the reporting currency ledger for French reporting into the secondary ledger for the French ledger, which in turn will be consolidated into the headquarters ledger with other countries and ledgers.

With Release 12 you can map primary ledgers to secondary ledgers even if you have different:

- Calendar or Accounting Periods
- Chart of Accounts
- Currencies
- Accounting Convention (the combination of all of the above)



Note: the example client was able to use secondary ledgers for local reporting requirements due to achieving certain currency threshold requirements. Your situation may be different.

Release 12 Inter-Company Challenges

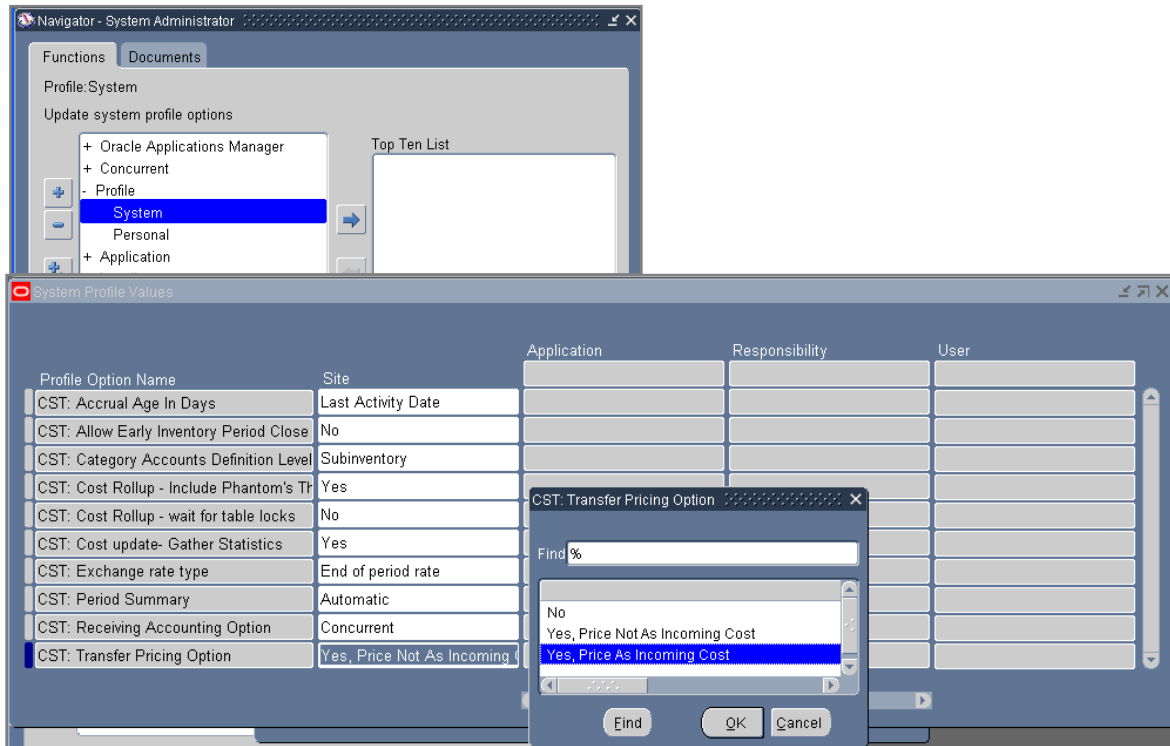
The following section describes several issues with inter-company accounting in Release 12.

Profit in Inventory

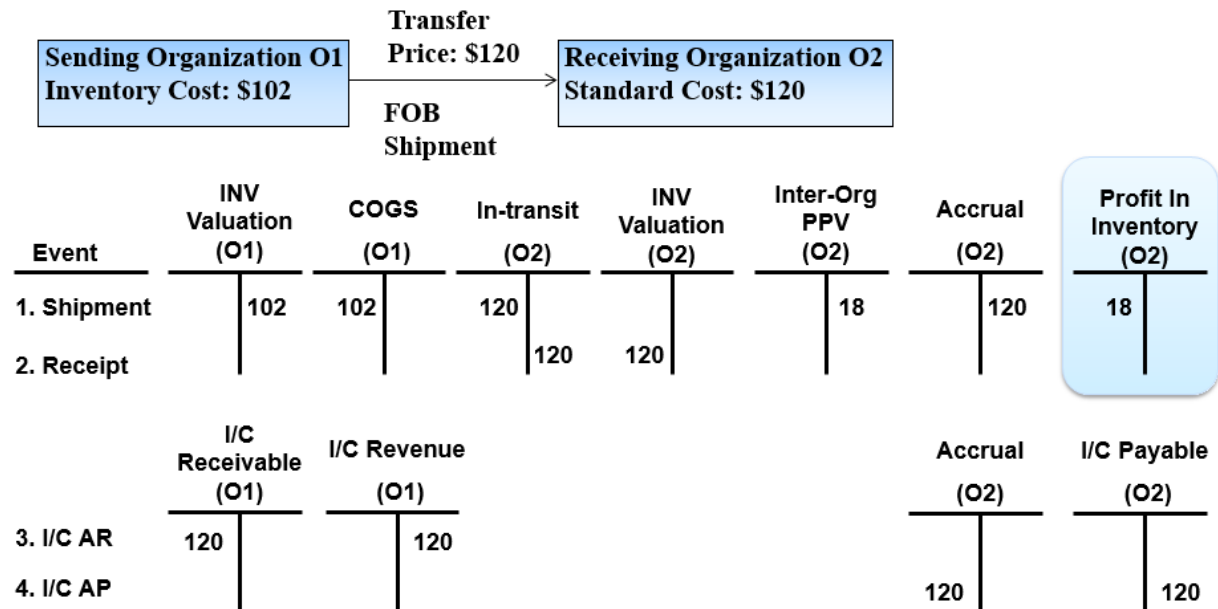
As described in the Background section of this paper, Oracle started to introduce profit in inventory concepts with Release 11.5.9. Starting with this point release you could accrue the profit from the sending organization into the receiving organization without some form of customization or manual journal entries. To try to use the standard Oracle functionality you need to set the Cost profile CST: Transfer Pricing Option. It has three values:

- **No** – transfer goods like an inter-organization transfer, same as Release 10 and 11
- **Yes, Price Not as Incoming Cost** – accrue ICP when the goods are received into your organization
- **Yes, Price as Incoming Cost** – ICP is not accrued, any differences go to Purchase Price Variance. Usually the Price is the same as the Receiving Org's cost and you have no PPV.

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?



When you set the CST: Transfer Pricing Option profile to **Yes, Price Not as Incoming Cost**, the accounting is as follows (FOB is set to FOB Shipment in this example):



In the above example the Receiving Organization has \$18 of Profit in Inventory, derived from the difference between the Sending Org's standard cost of \$102 and the Transfer Price of \$120. Note too that the Transfer Price is equal to the Receiving Org's standard cost (\$120 = \$120). This is normal and necessary for these two values to be the same.

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

But now that you have accrued for Profit in Inventory, how do you relieve it? Unfortunately Oracle does not have this capability. The current design assumes you will manually relieve your profit in inventory balances and correct your Intercompany Cost of Sales accounts. On top of this you don't have a Profit in Inventory Value Report to help you do this.

Two Business Scenarios for Profit in Inventory

So what are the common scenarios faced by businesses? Should you manage your Profit in Inventory in your local books or manage it at a more global level in your consolidation ledger?

If you have significant product costs and need to track your "true" profits at a local level, you need to report your local cost margins net of Profit in Inventory. To do this you have to transact into your material accounting, to both accrue the Profit in Inventory for when the goods are received, and also relieve it for:

- Miscellaneous issues
- Cost of Goods Sold / Shipments
- Inter-company Cost of Goods Sold
- WIP Scrap, WIP Variances, PPV, etc.

Potential Solutions for Profit in Inventory Adjusted in Local Books

Our client example did not need to adjust Profit in Inventory on a transactional level in the local books. But the following techniques have been used since Release 10:

Use a material overhead sub-element in the Frozen or Average cost, as this allows you to earn Profit in Inventory on Purchase Order and Inter-Org Receipt Transactions, as part of standard functionality. Alternatively you can set the CST: Transfer Pricing Option as **Yes, Price Not as Incoming Cost**. For the relief of Profit in Inventory you will need to create custom programs to add additional transactions to the material accounting transactions (MTL_TRANSACTION_ACCOUNTS). In effect you will add additional rows to the existing material accounting entries, correcting COGS and other transactions.

You will also need to create custom Profit in Inventory and WIP Valuation Reports, and most likely, create custom Cost Margin Reports to include or exclude Profit in Inventory adjustments. Example report layouts are in following sections of this paper for the Valuation Reports. And you should also create custom material and WIP distribution accounting reports to use the accounts from Subledger Accounting (see example SQL in the Appendix section).

Caveats

- You will need to customize the Margin Analysis Report to pick up the new COGS entries correctly.
- Overall, creating new material accounting entries to relieve Profit in Inventory is time-consuming to do and potentially expensive
- If you make major changes to your business model this customization may become a potential throwaway

Caveats with Average Costing

- For the end-of-month profit in inventory value report you need to calculate the average Profit in Inventory item cost for each month. This is not a simple task to do.
- This calculation has to compare the transactional selling price with the organization's moving average item cost, again, increasing the complexity
- And these custom programs and tables will need to accumulate Profit in Inventory values by organization by item for each month, and most likely, store these values in a cost type
- And note that for each month the average Profit in Inventory item cost may be different

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

Potential Solutions for Profit in Inventory Adjusted in Local Books

But why would you only manage your Profit in Inventory at a global level? What are the driving factors? You would consider an across the board approach if:

- You do not need to track your local margins net of Profit in Inventory
- You don't want your local country/ local tax authorities to see "real" margins
- You still need end of month Profit in Inventory Value Reports, and it requires the ability to support the consolidated entry with material transaction history or month-end inventory value reporting

This is the approach our client took. The report samples in this paper reflect these requirements; and we estimate that this approach takes less than 25% of the time it would take to do the transaction by transaction approach. The overall approach is:

1. Calculate your Profit in Inventory item costs

- Create an item cost report to show the frozen costs in the source organization and compare it to the target organization standard costs, based on the Sourcing Rules.
- Use this intercompany item cost report to derive the intercompany profit for your items, for input into your intercompany profit cost type
 - Conversely you could compare the transfer price list to the sending org's standard cost
 - For Average Costing you would have to calculate the end of period intercompany profit per item and put it into the respective cost type
 - For Standard Costing you set the amount of intercompany profit per item and not recalculate until you change your transfer prices or your standard costs

2. Store your Profit in Inventory Item Costs

- Store your ICP values in a non-Frozen cost type, called 'ICP' for example, in each inventory organization that has Profit in Inventory
- Usually you will enter a negative ICP amount, as it reduces your gross inventory value

3. Create Reports for Manual Journal Entries

- Add ability to calculate Receiving quantities at month-end, as this information is missing from the month-end inventory cost snapshot (CST_PERIOD_CLOSE_SUMMARY)
- If using WIP, add ability to calculate ICP Value in WIP at month-end
- Create custom reports for intercompany profit in inventory value at month-end for Receiving, Subinventories, WIP and Intransit
- Calculate the Subinventories and Intransit values by multiplying the non-Frozen ICP cost type by the stored quantities in the Costing Snapshot table (CST_PERIOD_CLOSE_SUMMARY)
- Calculate the Receiving and WIP values by multiplying the non-Frozen ICP cost type by the calculated quantities for Receiving and WIP components in inventory

The general formula for Profit in Inventory is:

$$\begin{array}{ccc} \boxed{\text{This Month's ICP Inventory Value Reports}} & - & \boxed{\text{Last Month's ICP Inventory Value Reports}} = \boxed{\text{Monthly Change in ICP Value}} \end{array}$$

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

Sample Reports for Profit in Inventory

Regardless of the approach, whether you adjust your local books or your consolidated books, the following reports will be very useful.

ICP Item Costs Report

ICP Item Costs Based on Sourcing Rules													
Item Number	Src Org	Assignmt Set	Src Curr Code	Src Item Cost	Curr Conv Rate	Conv Src Item Cost	To Org	To Curr Code	To Org Item Cost	Calc. ICP To Org	ICP Percent	PROD ICP Item Cost	
900990	200	Main	EUR	2.31492	1.29594	3.00000	203	EUR	4.00000	(1.00000)	(25.0)	(1.00000)	
100200	203	Main	EUR	10.00000	1.29594	12.95940	101	USD	15.00000	(2.04060)	(13.6)	(2.04060)	
100300	204	Main	EUR	7.00000	1.29594	9.07158	305	USD	25.00000	(15.92842)	(63.7)	(15.92842)	
100300	204	Main	EUR	7.00000	1.29594	9.07158	309	USD	25.00000	(15.92842)	(63.7)	(15.92842)	
100300	204	Main	EUR	7.00000	1.29594	9.07158	316	USD	25.00000	(15.92842)	(63.7)	(15.92842)	

Notes: 1 EUR of the ICP for item 100300 comes from Organization 200,
In Organization 203, in WIP, the component 900990 is used to make 100300.

You will take the ICP (Intercompany Profit) Item Costs as above (based on sourcing rules in this example), and store them in a cost type. These item costs are derived by comparing the Source Org's (Src Org) item costs with the To Org's item costs, by analyzing the sourcing rules to determine which Org ships to which Org. The difference is the Intercompany Profit in Inventory item cost. You could also calculate this by comparing the price lists against the standard costs, but since the Transfer Price usually equals the standard cost for the To Org (or Receiving Org as in the earlier example), comparing item costs is easier to do.

Onhand Inventory ICP Value Report

The Onhand Inventory ICP Value Report (ICP = Inter-Company Profit) multiplies the ICP item costs times the onhand quantity at month-end to calculate the month-end Profit in Inventory Values. This report gives you the month-end Profit in Inventory Values, for comparison against last month's values. Sample SQL is available in the Appendix section.

ICP Item Costs Based on Sourcing Rules													
Item Number	Src Org	Assignmt Set	Src Curr Code	Src Item Cost	Curr Conv Rate	Conv Src Item Cost	To Org	To Curr Code	To Org Item Cost	Calc. ICP To Org	ICP Percent	PROD ICP Item Cost	
900990	200	Main	EUR	2.31492	1.29594	3.00000	203	EUR	4.00000	(1.00000)	(25.0)	(1.00000)	
100200	203	Main	EUR	10.00000	1.29594	12.95940	101	USD	15.00000	(2.04060)	(13.6)	(2.04060)	
100300	204	Main	EUR	7.00000	1.29594	9.07158	305	USD	25.00000	(15.92842)	(63.7)	(15.92842)	
100300	204	Main	EUR	7.00000	1.29594	9.07158	309	USD	25.00000	(15.92842)	(63.7)	(15.92842)	
100300	204	Main	EUR	7.00000	1.29594	9.07158	316	USD	25.00000	(15.92842)	(63.7)	(15.92842)	

Notes: 1 EUR of the ICP for item 100300 comes from Organization 200,
In Organization 203, in WIP, the component 900990 is used to make 100300.

Onhand Inventory ICP Value Report

Month-End Snapshot (quantities) are from October 2009

Period Name	Org Code	Co	Acct	Prod Grp	Item Number	Curr	Gross Item Cost	ICP Item Cost	Net Item Cost	UOM	Qty	Inv. Onhand Value	Total ICP Onhand Value	Total Net Onhand Value
2009-10	203	1012	1301	1110	100300	EUR	10.00000	(1.00000)	9.00000	EA	2,450	24,500.00	(2,450.00)	22,050.00
2009-10	305	1048	1330	1172	100300	USD	25.00000	(15.92842)	4.56319	CTN	483	12,075.00	(7,693.43)	4,381.57
2009-10	309	1048	1330	1172	100300	USD	25.00000	(15.92842)	4.56319	CTN	649	16,225.00	(10,337.54)	5,887.46
2009-10	316	1048	1330	1172	100300	USD	25.00000	(15.92842)	4.56319	CTN	1,445	36,125.00	(23,016.57)	13,108.43
2009-10	316	1048	1350	1172	100300	USD	25.00000	(15.92842)	4.56319	CTN	4	100.00	(63.71)	36.29

Note: FG Product 2, 100300, has 1.29595 USD (1 EUR) ICP from organization 200 and the balance from organization 203

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

WIP ICP Value Report

Similar to the Onhand Inventory ICP Value Report, the WIP ICP Value Report multiplies the ICP item costs times the calculated WIP component quantities for month-end, to derive the month-end Profit in Inventory Values. Note that WIP stored onhand quantities do not exist; you will have to create custom SQL logic to compare the total component quantities issued to a WIP job against the component quantities consumed through the completion of your WIP assemblies. This component quantity information can be found in the WIP table called WIP_REQUIREMENT_OPERATIONS, as it tracks the component requirements. The table called WIP_DISCRETE_JOBS stores the number of assemblies completed at a point in time.

WIP ICP Value Report																
Notes: Only includes jobs with a Status of: Released, Complete, Complete - No charges, On hold, or Pending scheduled.																
Only includes jobs where the FG Scheduled Qty > FG Qty Completed																
Org Code	Co	Acct	Prod Grp	WIP Class	WIP Job	Status	FG Qty Completed	FG Item Number	Component Number	Curr	ICP Item Cost	Qty Per FG	Total Req Qty	Qty Issued	Est. Qty Left in WIP	ICP in WIP
203	1012	1315	1110	PROD	12345	Complete	2,304	100300	900990	EUR	(1.00000)	3	6,912	6,984	72	(72.00)
203	1012	1315	1110	PROD	12346	Complete	529	100300	900990	EUR	(1.00000)	3	1,587	1,632	45	(45.00)
203	1012	1315	1110	PROD	34567	Complete	1,070	100300	900990	EUR	(1.00000)	3	3,210	3,264	54	(54.00)
203	1012	1315	1110	PROD	49505	Released	1,639	100300	900990	EUR	(1.00000)	3	4,917	4,995	78	(78.00)
Note: Component 900990 comes from another internal organization, 200, with an ICP amount of 1 EUR per unit																

WIP ICP Value Report

As a double-check or a confirmation of the ending balances, last month's ending ICP balances minus this month's ending balances should be equal to:



To produce this report you will add up the monthly material transactions for quantities and values, for your Subinventory and Intransit entries (where the accounting line type is Inventory or Intransit, 1 or 14), and multiply the quantities by the ICP Item Costs. Here is a sample for this report:

ICP Material Transactions Report for October 2009 (Oct 1 - 31, 2009)													
Org Code	Co	Acct	Prod Grp	Item Number	Transaction Name	Curr	Gross Item Cost	ICP Item Cost	UOM	Qty	Inventory Value Change	ICP Value Change	Net Inventory Change
204	1012	1330	1172	100300	Intransit Shipment	EUR	7.00000	(1.00000)	CTN	-8,224.0	(57,568.00)	8,224.00	(49,344.00)
204	1012	1330	1172	100300	WIP Completion	EUR	7.00000	(1.00000)	CTN	8,224.0	57,568.00	(8,224.00)	49,344.00
224	1012	1320	1172	100300	Intransit Receipt	EUR	7.00000	(1.00000)	CTN	-8,224.0	(57,568.00)	8,224.00	(49,344.00)
305	1048	1330	1172	100300	Sales Order Pick	USD	25.00000	(15.92842)	CTN	-120.0	(3,000.00)	1,911.41	(1,088.59)
305	1048	1330	1172	100300	Sales Order Pick	USD	25.00000	(15.92842)	CTN	8.0	200.00	(127.43)	72.57
305	1048	1330	1172	100300	Sales Order Pick	USD	25.00000	(15.92842)	CTN	3.0	75.00	(47.79)	27.21
305	1048	1330	1172	100300	Sales Order Pick	USD	25.00000	(15.92842)	CTN	10.0	250.00	(159.28)	90.72
305	1048	1330	1172	100300	Sales Order Pick	USD	25.00000	(15.92842)	CTN	20.0	500.00	(318.57)	181.43
305	1048	1330	1172	100300	Sales Order Pick	USD	25.00000	(15.92842)	CTN	15.0	375.00	(238.93)	136.07
305	1048	1330	1172	100300	Sales Order Pick	USD	25.00000	(15.92842)	CTN	120.0	3,000.00	(1,911.41)	1,088.59
305	1048	1330	1172	100300	Sales order issue	USD	25.00000	(15.92842)	CTN	-160.0	(4,000.00)	2,548.55	(1,451.45)
305	1048	1330	1172	100300	Sales order issue	USD	25.00000	(15.92842)	CTN	-15.0	(375.00)	238.93	(136.07)
305	1048	1330	1172	100300	Sales order issue	USD	25.00000	(15.92842)	CTN	-20.0	(500.00)	318.57	(181.43)

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

ICP Material Transactions Report

You will use the monthly change in your Profit in Inventory to adjust your inter-company cost of goods sold. But what should you use as the offset account to your reductions (or increases) to inventory? It is not always inter-company cost of goods sold. To help in this regard, similar to the above report, for your non-inventory accounting entries (where the accounting line type is not Inventory or Intransit, not 1 or 14), and multiply the quantities by the ICP Item Costs. Here is a sample for this report:

ICP Material Transactions Report for October 2009 (Oct 1 - 31, 2009)											
(Non-Inventory Entries Only)											
Org Code	Co	Acct	Sub-Acct	Prod Grp	Item Number	Transaction Name	Curr	Qty	Inventory Value Change	ICP Value Change	Net Inventory Change
910	1021	5001	4201	1000	100248	COGS Recognition	EUR	-4	(21,816.64)	40.00	(21,776.64)
910	1021	5001	4201	1000	100248	COGS Recognition	EUR	-6	(32,724.96)	60.00	(32,664.96)
910	1021	5001	4201	1000	100248	COGS Recognition	EUR	-34	(185,441.44)	340.00	(185,101.44)
910	1021	5001	4201	1000	100248	COGS Recognition	EUR	-572	(3,120,827.21)	5,720.00	(3,115,107.21)
910	1021	5001	4201	1000	100248	Logical Sales Order Issue	EUR	572	3,120,827.21	(5,720.00)	3,115,107.21
910	1021	5001	4201	1000	100248	Logical Sales Order Issue	EUR	34	185,441.44	(340.00)	185,101.44
910	1021	5001	4201	1000	100248	Logical Sales Order Issue	EUR	6	32,724.96	(60.00)	32,664.96
910	1021	5001	4201	1000	100248	Logical Sales Order Issue	EUR	4	21,816.64	(40.00)	21,776.64

WIP Component ICP Variance Report

If you use WIP your WIP components may have ICP in the item costs. For example, the previous WIP ICP Value Report shows the component 900990 with an ICP item cost of 1.00 EUR. Assuming you are using WIP jobs, when you close your jobs you will reduce your ICP value in WIP as part of the WIP variances. But your WIP variances are overstated by the ICP amount. If you do not account for this properly your variances will be overstated as well as your Profit in Inventory. Here is a sample report:

WIP ICP Variance Report															
Notes: Only includes jobs closed in the current accounting period															
Org Code	Co	Cost Ctr	Acct	Prod Grp	WIP Class	WIP Job	FG Qty Completed	Component Number	Curr	ICP Item Cost	Qty Per FG	Total Req Qty	Actual Qty Issued	Est. Qty in WIP Variances	ICP in WIP Variances
203	1012	9900	1315	1110	PROD	12342	1,800.0	900990	EUR	(1.00000)	3.0	5,400.0	5,700.0	300.0	(300.00)
203	1012	9900	1315	1110	PROD	12344	1,100.0	900990	EUR	(1.00000)	3.0	3,300.0	3,500.0	200.0	(200.00)
203	1012	9900	1315	1110	PROD	40620	1,000.0	900990	EUR	(1.00000)	3.0	3,000.0	3,000.0	0.0	0.00
203	1012	9900	1315	1110	PROD	40621	1,000.0	900990	EUR	(1.00000)	3.0	3,000.0	3,200.0	200.0	(200.00)
203	1012	9900	1315	1110	PROD	40622	1,000.0	900990	EUR	(1.00000)	3.0	3,000.0	3,000.0	0.0	0.00

Note: Component 900990 comes from another internal organization, 200, with an ICP amount of 1 EUR per unit

Caveats with Profit in Inventory Solutions

- Wise to use 3rd party tools to speed ICP item cost data entry. More4Apps has MS Excel solutions for fast item cost entry, using the Open Item Cost interfaces
- This solution models your Profit in Inventory adjustments but it is not 100% accurate. Why?
 - Currency changes will cause inaccuracies (i.e. Standard cost based on 1.3 USD/Euro, but inter-org transfers use the current daily transaction rate of 1.5)
 - If you have multiple Profit in Inventory Item Costs you will need to manually adjust your calculations as the ICP item cost in a particular inventory org may be a composite or average of ICP item costs
- Transfer price changes will cause inaccuracies
 - The weighted average ICP has to be kept by organization, or,
 - Revalue your ICP amounts with the latest transfer price changes

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

- In the ideal world, store the ICP item costs:
 - by item, org
 - and keep track of the average ICP item costs to accurately relieve profit in inventory and keep track of the original currency rate and ICP amounts in use

Time Sensitive Setups

When you enter your shipping network, intercompany transaction flows and inter-company relationships the setups are immediately effective. You cannot stage your changes for future use, there are no effective dates. This makes cutting in business changes (such as a new model introduction) more difficult to do. Here are some screen shots showing these Oracle forms.

Shipping Network Form

Shipping Networks (E2)

Organization: E2 Vision Sweden
Scope: From or To Organization: [v] Find

Shipping Networks

Main | Transfer, Distance | Primary Accounts | Secondary Accounts | Other Accounts | Subcontracting | Transfer Price

From	To	Transfer Type	Elemental Visibility Enabled	FOB	Manual Receipt at Expense Destination	Receipt Routing	Internal Order Required
E1	E2	Direct	<input type="checkbox"/>	Direct	<input type="checkbox"/>	[v]	<input type="checkbox"/>
E2	E1	Direct	<input type="checkbox"/>	Direct	<input type="checkbox"/>	[v]	<input type="checkbox"/>
E2	E5	Intransit	<input type="checkbox"/>	Receipt	<input type="checkbox"/>	Direct	<input type="checkbox"/>
E2	E52	Intransit	<input type="checkbox"/>	Receipt	<input type="checkbox"/>	Direct	<input checked="" type="checkbox"/>
E2	E6	Direct	<input type="checkbox"/>	Direct	<input type="checkbox"/>	[v]	<input type="checkbox"/>
E2	E8	Direct	<input type="checkbox"/>	Direct	<input type="checkbox"/>	[v]	<input type="checkbox"/>
E5	E2	Intransit	<input type="checkbox"/>	Receipt	<input type="checkbox"/>	Direct	<input checked="" type="checkbox"/>

Organization Name

From: Vision Germany
To: Vision Sweden

New Open

Intercompany Transactions Flows and Intercompany Relations Form

Intercompany Transaction Flows

Start	End	Type	Ship From/To Organization	Qualifier
Vision Sweden	Vision France	Shipping	E2	[v]
Vision Netherlands	Vision Germany	Shipping	E4	[v]
Vision Germany	Vision Netherlands	Shipping	E1	[v]
Vision Industries	Vision Italy	Shipping	E3	[v]
Vision Belgium	Vision Operations	Shipping	E8	[v]

Nodes

Seq	From Operating Unit	Organization	To Operating Unit	Organization	Details Required
1	Vision Sweden	E2	Vision France	E5	<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>

Intercompany Relations

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

Release 12 Challenges for Inventory Reconciliation

Typically when you have global inventory you have more inventory reconciliation issues. There are some improvements in Release 12, but the following issues persist:

- The differences between the perpetual and accounted-for inventory balances are trapped in a table
- You can see it on the Period Close Reconciliation Report, but
 - Most users and analysts don't know about it
 - This report is limited to a single warehouse
 - And it shows all items whether or not in stock

Ideally you would have a Multi-Org Perpetual vs. Accounted-For Report, such as:

Out-Of-Balance Reporting

Sample Out of Balance Items												
Org	Period Name	Co	Acct	Prod Grp	Item	Desc.	UOM	Qty	Onhand Value	Accounted Value	Difference	
103	2009-10	1010	1330	1120	100001	Desc. 1	CRD	100	68,328.27	68,323.71	4.56	
103	2009-10	1010	1330	1120	100002	Desc. 2	CRD	126	318,061.67	318,055.37	6.30	
308	2009-10	1048	1330	1110	100003	Desc. 3	CTN	2117	670,576.31	674,082.35	(3,506.04)	
308	2009-10	1048	1330	1110	100004	Desc. 4	CTN	205	91,723.55	94,260.35	(2,536.80)	
308	2009-10	1048	1330	1110	100005	Desc. 5	CTN	1320	39,430.20	39,923.58	(493.38)	
											(6,525.36)	

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

This report is based on the month-end cost snapshot table `CST_PERIOD_CLOSE_SUMMARY`. When you close the inventory accounting period the Period Close Reconciliation Report runs and it populates this table. It stores the month-end perpetual quantities by taking a read of the onhand and intransit quantities at the time you close the inventory accounting period, and should the next inventory accounting period be open, it will subtract away any quantity changes due to transactions that were played in the new month. It also takes last month's perpetual quantity, adds together the monthly accounted for material transactions to come up with the cumulative accounted quantities and values. This report is used as a reconciling item for balancing your Inventory Value reports to your General Ledger.

- This is the same source for the month-end quantities for the **Onhand Inventory ICP Value Report**.

Subledger Accounting and Inventory Valuation Reporting and Reconciliation

Due to the flexibility of Subledger Accounting, Oracle Cost Development felt it was necessary to remove the material valuation accounts from the Inventory Value Reports, as with Subledger Accounting, you can no longer use the Subinventory or Material Parameter setups to determine the valuation accounts. So for example if you run the Inventory Value Report, choose the Option **Summary**, all you will get is the totals by Subinventory and in Total. You will not see any inventory valuation accounts.

For the same reasons they disabled the Receiving Value by Destination Account Report, thinking that the Subledger Accounting Account Balances and other reports would be sufficient.

- So if you are accruing your expense receipts at time of receipt (on-line accruals for expenses) and also have inventory, it is very difficult to see if you have inventory or expenses in your Receiving Value. You can no longer run the Receiving Value by Destination Account Report.
- Do note that Release 12 does offer valuable reconciliation improvements. All of the Inventory Valuation Reports have a "as of date" concept. You can run these reports and backdate them to a desired transaction date. The report will get the current quantities and values and subtract away subsequent transactions until it reaches the desired "as of date".

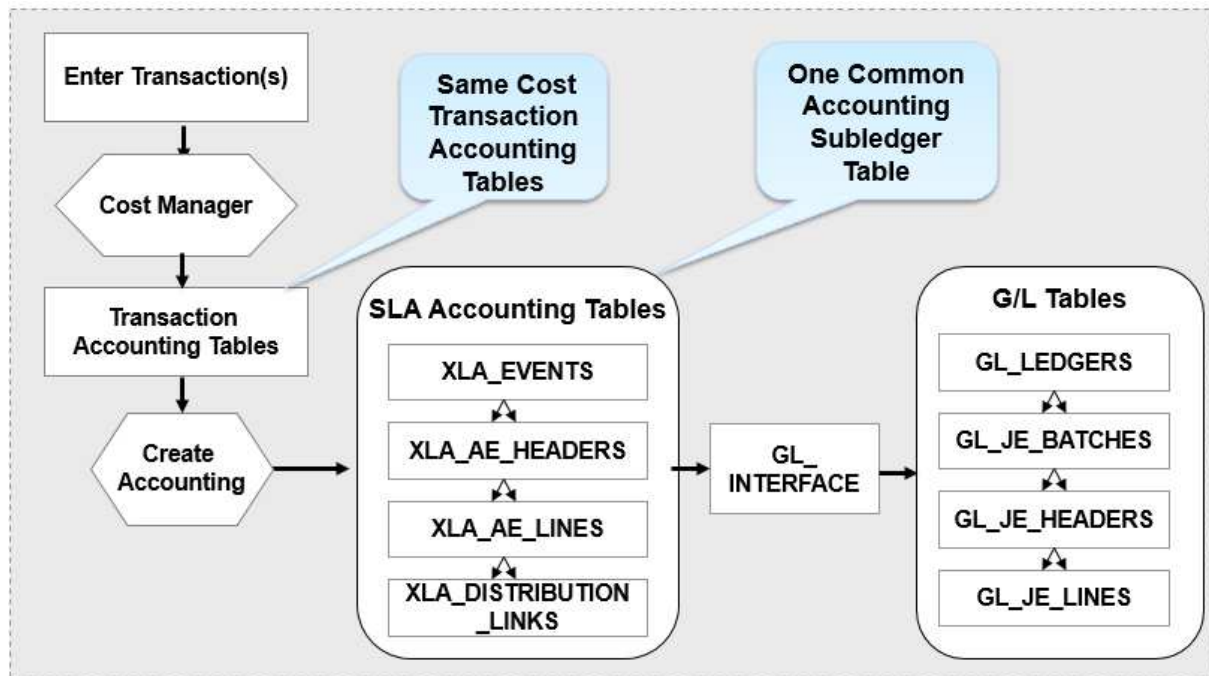
Subledger Accounting and Material and WIP Distribution Inquiries and Reports

As mentioned earlier, in Release 12 there have been important changes to the underlying tables. But as outlined below the Cost Transaction tables from the prior release still exist. And this may cause some confusion if you run an existing Cost Transaction Report (i.e. for Discrete Costing, such as the Material Account Distribution Summary Report), especially after you have used SLA to modify the accounting string. The old Cost Distribution tables will use the old account string and the new SLA tables will use the revised account string (`CODE_COMBINATION_ID`). After Create Accounting is run the Cost Transactions also reside in the SLA table structures. In effect there is another cost accounting layer in Release 12, as the tables from the prior release are still available.

To complicate matters further the **CST: Account Summarization** profile has been removed. Unlike Releases 10, 11 and 11i, if you use Standard Costing you can no longer summarize your entries across cost elements, you must book all your cost accounting entries by cost element. If the new subledger inquiries and reports showed the needed operational details (such as item, transaction type, quantity and so forth) you could summarize across cost elements in your Subledger Accounting information and you would not care. But currently up to Release 12.2 you cannot see operational details on the Subledger Accounting inquiries and reports so you are stuck using the older Material Distribution and WIP Distribution inquiries and reports. And these transaction accounting distribution reports and inquiries do not show the cost element either.

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

Here is a conceptual diagram for many of the related tables:



Conclusion

Subledger Accounting offers good flexibility for allowing you to set up your Accounting Flexfields as desired on your transactions and in doing so solves many account classification issues.

But overall we still have the same inter-company accounting challenges as Release 11i, such as:

- Non-Shippable Items
- Non-Invoiced Items
- Internal Fulfillment returns (lack of a “RTV” transaction for internal orders)

And we still have challenges on ship dates vs. invoice dates. If you require separate customer acceptance dates, your invoice dates should be based on your customer acceptance dates, not ship dates. This is a current issue as of 6-Mar-2010. To get around this we created a batch program to intercept the invoice lines before Autoinvoice runs, in order to correctly set the invoice date.

In addition, inter-company profit solutions are not standard functionality. You need to plan ahead in order to give yourself enough time to create workable solutions. This paper gives insight and recommendations for approaches you might take to resolve these requirements.

And with the nature of global inventory, inventory reconciliation issues are amplified. And unfortunately, Release 12 both hurt and helped with inventory reconciliation:

- Inventory value reports have as “as of date” which is very good indeed!
- But most reports have no valuation accounts
- Still only single warehouse
- Period Close Reconciliation Report not well used

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

Thanks for allow us to share our experiences with you.

Acknowledgements

Jerry and Doug wish to thank for following individuals for their contributions to this paper:

- Bini Mammen – Consultant, Next Generation Consultants
- Oracle Cost Management Development, for meeting with Doug to discuss these challenges
- Hans Kolbe, lead for the OAUG Multi-Org SIG

Additional information is available from:

- “*Manufacturing Accounting by Product Line For Release 11*”. OAUG Paper from Doug Volz. This paper explains how the underlying manufacturing transactions work in Oracle; including a deeper look at material transaction types, sources and transaction actions.

About the Authors

Both Jerry DeVore and Douglas Volz have over 25 years experience, including industry positions and the use of the Oracle Applications. Jerry DeVore is a Senior Consultant with CSC. His 25 years of industry experience as a plant controller and Supply Chain manager make him uniquely aware of requirements in Manufacturing Cost and Supply Chain. He has been involved with the conversion of multiple legacy systems both peer to peer and to Oracle ERP solutions. Jerry has implemented Oracle for the last seven years and these implementations included a variety of manufacturing operations from continuous process manufacturing to batch operations to large distribution and inventory management solutions. Most recently he was the Manufacturing Cost/Supply Chain lead for an Oracle Release 12 implementation, with a global footprint, having multiple currencies, ledgers, operating units and inventory organizations.

Doug is a former Cost Accounting Manager from multiple industries with many of them having a global footprint. His industry experience plus his Oracle design experience with Oracle Cost Management and related applications gives him insights into solving these inter-company and Cost Accounting reconciliation challenges.

Doug is President of Douglas Volz Consulting, Inc., founded in 2005 with the goal of helping Oracle customers use and implement Oracle Financial and Manufacturing applications. As a Senior Architect and Advisor for Oracle Application projects, Doug has 30 years experience, including 5 years in Oracle Development (co-designing Oracle Cost Management), 11 years in industry in Cost and Accounting Management positions and 14 years international consulting . His Manufacturing and Cost systems experience covers project management, business process/systems design, delivery and consulting services, for both Oracle Corporation, and multiple international consulting firms. Prior to his systems career, Mr. Volz also held numerous management accounting positions for telecommunications, defense, and electronics companies.

Doug has led the OAUG Discrete Manufacturing Cost Sub-Committee since 2007. He also advises and participates on the Oracle Customer Advisory Board for Fusion Costing and for the Fusion SCM Strategy Council. In addition, Doug has presented papers at OpenWorld, UKOUG and OAUG conferences over the past 14 years on Business Transformation, Change Management and Cost Accounting topics.

Doug may be reached at doug@volzconsulting.com or at www.volzconsulting.com; Jerry may be reached at jdevore@CSC.com.

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

APPENDIX

- 1. Accounting Line Types**
- 2. Technical Overview for Material and WIP Transaction Tables**
- 3. Sample SQL Code for Release 12 Material Distribution Report**
- 4. Sample SQL Code for Release 12 WIP Distribution Report**
- 5. Sample SQL Code for the Onhand Inventory ICP Value Report**

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

1. Accounting Line Types

The following accounting line types describe the purpose for the accounting entries. In Release 11i only Oracle Cost Management used this concept (MTL_TRANSACTION_ACCOUNTS, WIP_TRANSACTION_ACCOUNTS), and to a lesser degree the A/R distribution tables. In Release 12 the Receiving Accounting table also uses accounting line types:

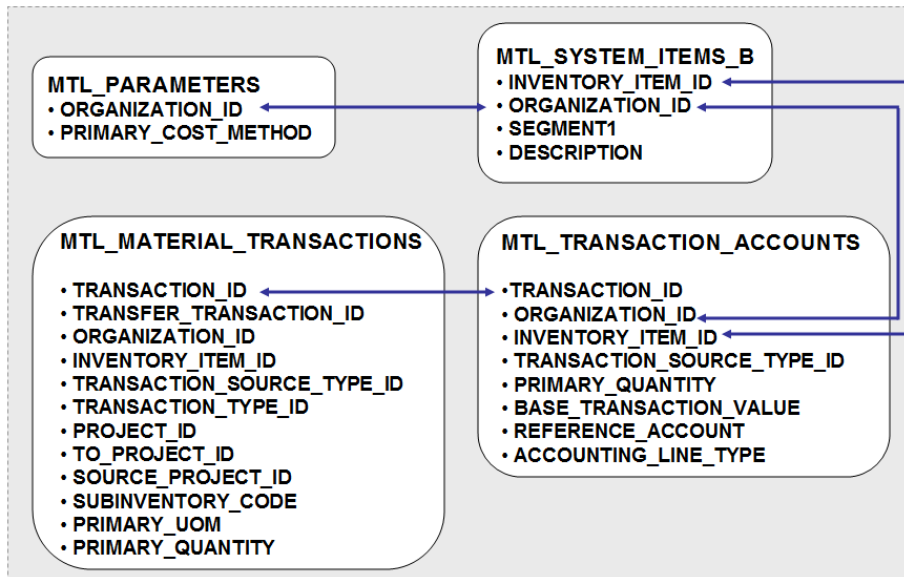
Lookup Type	Lookup Code	Meaning
CST_ACCOUNTING_LINE_TYPE	1	Inv valuation
CST_ACCOUNTING_LINE_TYPE	2	Account
CST_ACCOUNTING_LINE_TYPE	3	Overhead absorption
CST_ACCOUNTING_LINE_TYPE	4	Resource absorption
CST_ACCOUNTING_LINE_TYPE	5	Receiving Inspection
CST_ACCOUNTING_LINE_TYPE	6	Purchase price variance or rate variance
CST_ACCOUNTING_LINE_TYPE	7	WIP valuation
CST_ACCOUNTING_LINE_TYPE	8	WIP variance
CST_ACCOUNTING_LINE_TYPE	9	Inter-org payables
CST_ACCOUNTING_LINE_TYPE	10	Inter-org receivables
CST_ACCOUNTING_LINE_TYPE	11	Inter-org transfer credit
CST_ACCOUNTING_LINE_TYPE	12	Inter-org freight charge
CST_ACCOUNTING_LINE_TYPE	13	Cost variance
CST_ACCOUNTING_LINE_TYPE	14	Intransit Inventory
CST_ACCOUNTING_LINE_TYPE	15	Encumbrance Reversal
CST_ACCOUNTING_LINE_TYPE	16	Accrual
CST_ACCOUNTING_LINE_TYPE	17	Invoice Price Variance
CST_ACCOUNTING_LINE_TYPE	18	Exchange Rate Variance
CST_ACCOUNTING_LINE_TYPE	19	Special Charge Expense
CST_ACCOUNTING_LINE_TYPE	20	Expense
CST_ACCOUNTING_LINE_TYPE	21	Lot based job split issue
CST_ACCOUNTING_LINE_TYPE	22	Lot based job split return
CST_ACCOUNTING_LINE_TYPE	23	Lot based job merge issue
CST_ACCOUNTING_LINE_TYPE	24	Lot based job merge return
CST_ACCOUNTING_LINE_TYPE	25	Lot based job bonus issue
CST_ACCOUNTING_LINE_TYPE	26	Lot based job bonus return
CST_ACCOUNTING_LINE_TYPE	27	Lot based job quantity update issue
CST_ACCOUNTING_LINE_TYPE	28	Lot based job quantity update return
CST_ACCOUNTING_LINE_TYPE	29	Estimated Scrap Absorption
CST_ACCOUNTING_LINE_TYPE	30	Profit in inventory
CST_ACCOUNTING_LINE_TYPE	31	Clearing
CST_ACCOUNTING_LINE_TYPE	32	Retroactive Price Adjustment
CST_ACCOUNTING_LINE_TYPE	33	Shikyu Variance
CST_ACCOUNTING_LINE_TYPE	34	Inter-org Profit
CST_ACCOUNTING_LINE_TYPE	35	Cost of Goods Sold
CST_ACCOUNTING_LINE_TYPE	36	Deferred Cost of Goods Sold
CST_ACCOUNTING_LINE_TYPE	37	Standard Cost Update Adjustment
CST_ACCOUNTING_LINE_TYPE	99	Unknown

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

2. Inventory and WIP Transaction Tables

The manufacturing accounting entries are held in the tables MTL_TRANSACTION_ACCOUNTS and WIP_TRANSACTION_ACCOUNTS (the PO receipt accruals are held in RCV_TRANSACTIONS and RCV_RECEIVING_SUB_LEDGER, but are not part of this discussion). The underlying table structures are:

Inventory Transaction Tables



MTL_PARAMETERS holds the most basic definition of the Inventory Organization or warehouse, holding the costing method, default inventory valuation accounts and default inventory controls for the warehouse. MTL_PARAMETERS join to the complex HR Organization tables; for more information see the view definition for ORG_ORGANIZATION_DEFINITIONS.

MTL_SYSTEM_ITEMS_B defines your items in the item master. Depending on your Key Flexfield definition, the item number is usually held in the SEGMENT1 column, with the INVENTORY_ITEM_ID and the ORGANIZATION_ID the key defining columns for this table. Each inventory organization has its own set of items, grouped together by ORGANIZATION_ID. A child table, MTL_SYSTEM_ITEMS_TL, holds the foreign language translations for your item descriptions. And typically a “master” Item Master organization is set up, holding all of the items for each and every inventory organization.

MTL_MATERIAL_TRANSACTIONS holds the material transactions for inventory and WIP, including transfers to Project Accounting (into non-inventory related projects), as well as into Project Manufacturing projects. The unique reference is the TRANSACTION_ID for each row. The ORGANIZATION_ID tells you which inventory organization the transaction belongs to for most transactions (except for certain organization to organization transfers, called Direct Inter-Org Transfers). So it is much safer to join the Item Master’s ORGANIZATION_ID to the Material Accounting Table, MTL_TRANSACTION_ACCOUNTS, as the ORGANIZATION_ID in this table always represents the correct inventory warehouse. The material transaction types are defined in a table called MTL_TRANSACTION_TYPES, which includes the TRANSACTION_SOURCE_TYPE_ID (source of the transaction, such as Purchase Order, Sales Order and so forth) and the TRANSACTION_ACTION_ID (how the transaction affects the inventory balances, issuing versus receiving the quantities into inventory).

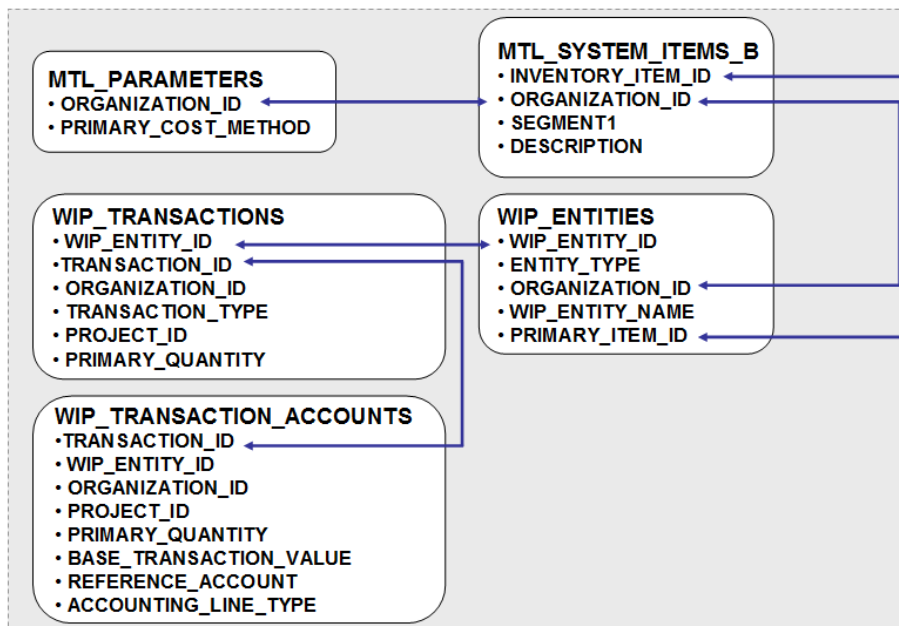
And these columns are not easy to use as their meanings vary by TRANSACTION_SOURCE_TYPE_ID, TRANSACTION_ACTION_ID and MTL_TRANSACTION_TYPE_ID. For example, for a given project transfer, to correctly pick up the PROJECT_ID or TO_PROJECT_ID or SOURCE_PROJECT_ID information you need to first look at the TRANSACTION_ACTION_ID. And for Miscellaneous Project Transfers the logic is reversed for

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

choosing the PROJECT_ID or TO_PROJECT_ID or SOURCE_PROJECT_ID information. The same is true for the Subinventory Code, as sometimes you use the SUBINVENTORY_CODE column and sometimes the TRANSFER_SUBINVENTORY_CODE column, depending on the TRANSACTION_ACTION_ID and TRANSACTION_SOURCE_TYPE_ID.

MTL_TRANSACTION_ACCOUNTS holds the material accounting entries for the parent material transactions table. For each row in MTL_MATERIAL_TRANSACTIONS there can be none to many rows in MTL_TRANSACTION_ACCOUNTS. It joins to MTL_MATERIAL_TRANSACTIONS using the TRANSACTION_ID, but generally speaking you should never join these two tables by ORGANIZATION_ID. MTL_TRANSACTION_ACCOUNTS joins to the item master MTL_SYSTEM_ITEMS_B table using the INVENTORY_ITEM_ID and the ORGANIZATION_ID columns. The BASE_TRANSACTION_VALUE column is the Debit (positive number) or the Credit (negative) number for the accounting entry. The ACCOUNTING_LINE_TYPE gives you the purpose of the accounting entry, such as Inventory (1), Intransit (14), WIP (7) and Account (2) for entries such as Cost of Goods Sold. Please see the appendix for a list of Release 12 Accounting Line Types. And for Release 11i, the REFERENCE_ACCOUNT column holds the code_combination_id found in the GL_CODE_COMBINATIONS table (the table which defines the Accounting Flexfield values). Note that in Release 12 Subledger Accounting has the final say for the Accounting Flexfield values. If using Release 12 you should join to the Subledger Accounting tables to get the correct account numbers.

WIP Transaction Tables



The WIP job is defined in WIP_ENTITIES, listing the ENTITY_TYPE (Discrete job, Repetitive Schedule, EAM Job and so forth). The name of the job is in the WIP_ENTITY_NAME column and the assembly being made on the job, or the repetitive assembly is referenced in the PRIMARY_ITEM_ID column. (Note: other tables are used to define WIP jobs, repetitive schedules, EAM jobs and work-order completions, but this is beyond the scope of this paper.)

WIP_TRANSACTIONS holds the WIP resource, outside processing, cost updates and item-based overhead transactions for your discrete jobs, repetitive schedules and work-orderless completions, as well as the straight-forward PROJECT_ID column. Unlike MTL_MATERIAL_TRANSACTIONS, TRANSACTION_TYPE is based on a lookup code and the ORGANIZATION_ID reliably references the inventory organization. WIP_TRANSACTIONS also references the assembly being worked upon by joining the WIP_ENTITY_ID to WIP_ENTITIES.PRIMARY_ITEM_ID column.

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

WIP_TRANSACTION_ACCOUNTS holds the WIP Resource, Overhead, Cost Update and Variance accounting entries for the parent WIP transactions table (WIP component issues, returns and Assembly completions are in the Material Transaction tables). For each row in WIP_TRANSACTIONS there can be none to many rows in WIP_TRANSACTION_ACCOUNTS. It joins to WIP_MATERIAL_TRANSACTIONS using the TRANSACTION_ID. MTL_TRANSACTION_ACCOUNTS joins to the item master MTL_SYSTEM_ITEMS_B table using the INVENTORY_ITEM_ID and the ORGANIZATION_ID columns. The BASE_TRANSACTION_VALUE column is the Debit (positive number) or the Credit (negative) number for the accounting entry. And the ACCOUNTING_LINE_TYPE gives you the purpose of the accounting entry, such as Inventory (1), Intransit (14), WIP (7) and Account (2) for entries such as WIP Scrap Account. Please see the appendix for a list of Release 12 Accounting Line Types. And for Release 11i, the REFERENCE_ACCOUNT column holds the code_combination_id found in the GL_CODE_COMBINATIONS table (the table which defines the Accounting Flexfield values). Note that in Release 12 Subledger Accounting has the final say for the Accounting Flexfield values. If using Release 12 you should join to the Subledger Accounting tables to get the correct account numbers.

3. Sample Release 12 Material Distribution Report

```

/* +-----+
-- | Copyright 2009-2010 Douglas Volz Consulting, Inc.
-- | All rights reserved.
-- | Permission to use this code is granted provided the original author is
-- | acknowledged. No warranties, express or otherwise is included in this
-- | permission.
-- +-----+
-- |
-- | Original Author: Douglas Volz (davolz@comcast.net)
-- |
-- | Program Name: XXX_XLA_MTL_ACCTS_DETAIL_REPT.sql
-- |
-- | Parameters:
-- | P_OPERATING_UNIT      -- operating unit parameter
-- | P_TRX_DATE_FROM      -- starting transaction date
-- | P_TRX_DATE_TO        -- ending transaction date
-- |
-- | Description:
-- | Report to get the material accounting entries by account,
-- | transaction type and accounting line type.
-- |
-- | Version Modified on Modified by Description
-- | =====
-- | 1.0 06 Nov 2009 Douglas Volz Initial Coding
-- | 1.1 11 Nov 2009 Douglas Volz Added Org Code and transaction ID
-- | 1.2 12 Nov 2009 Douglas Volz Added item and description
-- | 1.3 06 Jan 2010 Douglas Volz Made dates a parameter
-- | 1.4 12 Jan 2010 Douglas Volz Added quantity and unit cost columns
-- +-----+*/

select gl.name "Ledger",
       haou2.name "Operating Unit",
       mp.organization_code "Org Code",
       oap.period_name "Period Name",
       gccl.code_combination_id "CCID",
       ml.meaning "Acct Line Type",
       mtt.transaction_type_name "Txn Name",
       mta.transaction_id "Txn Id",
       gccl.segment1 "Co",
       gccl.segment2 "Cost Ctr",
       gccl.segment3 "Acct",
       gccl.segment4 "Sub-Acct",
       gccl.segment5 "Prod Grp",
       gccl.segment6 "Loc",
       gccl.segment7 "InterCo",
       gccl.segment8 "Future1",
       gccl.segment9 "Future2",
       msi.segment1 "Item",
       msi.description "Description",

```

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

```

DECODE(mta.accounting_line_type, 7, 'WIP', 14, '', 1,
DECODE (mmt.transaction_action_id,
        2, DECODE (SIGN (mta.primary_quantity),
                    -1, mmt.subinventory_code,
                    1, mmt.transfer_subinventory,
                    mmt.subinventory_code
                ),
        3, DECODE (mmt.organization_id,
                    mta.organization_id, mmt.subinventory_code,
                    mmt.transfer_subinventory
                ),
        21, DECODE (SIGN (mta.primary_quantity),
                    -1, mmt.subinventory_code,
                    1, mmt.transfer_subinventory,
                    mmt.subinventory_code
                ),
        22, DECODE (SIGN (mta.primary_quantity),
                    -1, mmt.subinventory_code,
                    1, mmt.transfer_subinventory,
                    mmt.subinventory_code
                ),
        28, DECODE (SIGN (mta.primary_quantity),
                    -1, mmt.subinventory_code,
                    1, mmt.transfer_subinventory,
                    mmt.subinventory_code
                ),
        mmt.subinventory_code
    )
) "Subinv",
mta.primary_quantity "Qty",
sum(nvl(al.ACCOUNTED_DR,0) - nvl(al.ACCOUNTED_CR,0)) /
    decode((mta.primary_quantity), 0,1,(mta.primary_quantity)) "Unit Cost",
sum(nvl(al.ACCOUNTED_DR,0) - nvl(al.ACCOUNTED_CR,0)) "Amount"
from
inv.mtl_transaction_accounts mta,
inv.mtl_material_transactions mmt,
inv.mtl_transaction_types mtt,
inv.mtl_system_items_b msi,
inv.org_acct_periods oap,
apps.GL_CODE_COMBINATIONS_KFV gccl,
inv.mtl_parameters mp,
apps.mfg_lookups ml,
hr.HR_ORGANIZATION_INFORMATION hoi,
hr.hr_all_organization_units haou, -- inv_organization_id
hr.hr_all_organization_units haou2, -- operating unit
gl.gl_ledgers gl,
xla.xla_transaction_entities ent,
xla.xla_events xe,
xla.xla_distribution_links xdl,
xla.xla_ae_headers ah,
xla.xla_ae_lines al
-- =====
-- Material Transaction, org and item joins
-- =====
where mta.transaction_id      = mmt.transaction_id
and   mmt.transaction_type_id = mtt.transaction_type_id
and   mta.organization_id    = msi.organization_id
and   mta.inventory_item_id   = msi.inventory_item_id
and   mp.organization_id      = msi.organization_id
-- =====
-- Inventory Org accounting period joins
-- =====
and   oap.period_name        = ah.period_name
and   oap.organization_id    = mta.organization_id
-- =====
-- Version 1.3, added lookup values to see more detail
-- =====
and   ml.lookup_type         = 'CST_ACCOUNTING_LINE_TYPE'
and   ml.lookup_code         = mta.accounting_line_type
-- =====
-- Material Transaction date and accounting code joins
-- =====

```


Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

```

and      mta.transaction_date      >= to_date('&P_TRX_FROM_YYYY_MM_DD', 'YYYY/MM/DD HH24:MI:SS')
-- P_TRX_DATE_FROM
and      mta.transaction_date      <  to_date('&P_TRX_TO_YYYY_MM_DD', 'YYYY/MM/DD HH24:MI:SS')+1
-- P_TRX_DATE_TO
-- =====
-- using the base tables to avoid the performance issues
-- with org_organization_definitions and hr_operating_units
-- =====
and      hoi.org_information_context = 'Accounting Information'
and      hoi.organization_id         = mta.organization_id
and      hoi.organization_id         = haou.organization_id -- this gets the organization name
and      haou2.organization_id       = to_number(hoi.org_information3)
-- this gets the operating unit id
and      gl.ledger_id               = to_number(hoi.org_information1) -- get the ledger_id
-- =====
-- SLA table joins to get the exact account numbers
-- =====
AND      ent.entity_code             = 'MTL_ACCOUNTING_EVENTS'
AND      ent.application_id          = 707
AND      xe.application_id           = ent.application_id
AND      xe.event_id                 = xdl.event_id
AND      ah.entity_id                = ent.entity_id
AND      ah.ledger_id                = ent.ledger_id
AND      ah.application_id            = al.application_id
AND      ah.application_id            = 707
AND      ah.event_id                  = xe.event_id
AND      ah.ae_header_id              = al.ae_header_id
AND      al.application_id            = ent.application_id
AND      al.ledger_id                 = ah.ledger_id
AND      al.AE_HEADER_ID              = xdl.AE_HEADER_ID
AND      al.AE_LINE_NUM               = xdl.AE_LINE_NUM
AND      xdl.application_id           = ent.application_id
AND      xdl.source_distribution_type = 'MTL_TRANSACTION_ACCOUNTS'
AND      xdl.source_distribution_id_num_1 = mta.inv_sub_ledger_id
AND      gccl.code_combination_id     = al.code_combination_id
-- =====
group by
      gl.name,
      haou2.name,
      mp.organization_code,
      oap.period_name,
      gccl.code_combination_id,
      ml.meaning,
      mtt.transaction_type_name,
      mta.transaction_id,
      gccl.segment1,
      gccl.segment2,
      gccl.segment3,
      gccl.segment4,
      gccl.segment5,
      gccl.segment6,
      gccl.segment7,
      gccl.segment8,
      gccl.segment9,
      msi.segment1,
      msi.description,
      DECODE(mta.accounting_line_type, 7, 'WIP', 14, '', 1,
      DECODE (mmt.transaction_action_id,
      2, DECODE (SIGN (mta.primary_quantity),
      -1, mmt.subinventory_code,
      1, mmt.transfer_subinventory,
      mmt.subinventory_code
      ),
      3, DECODE (mmt.organization_id,
      mta.organization_id, mmt.subinventory_code,
      mmt.transfer_subinventory
      ),
      21, DECODE (SIGN (mta.primary_quantity),
      -1, mmt.subinventory_code,
      1, mmt.transfer_subinventory,
      mmt.subinventory_code

```

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

```

    ),
    22, DECODE (SIGN (mta.primary_quantity),
               -1, mmt.subinventory_code,
               1, mmt.transfer_subinventory,
               mmt.subinventory_code
    ),
    28, DECODE (SIGN (mta.primary_quantity),
               -1, mmt.subinventory_code,
               1, mmt.transfer_subinventory,
               mmt.subinventory_code
    ),
    mmt.subinventory_code
)
),
mta.primary_quantity
order by 1,2;

```

4. Sample Release 12 WIP Distribution Report:

```

/* +-----+
-- | Copyright 2009-2010 Douglas Volz Consulting, Inc.
-- | All rights reserved.
-- | Permission to use this code is granted provided the original author is
-- | acknowledged. No warranties, express or otherwise is included in this
-- | permission.
-- +-----+
-- |
-- | Original Author: Douglas Volz (davolz@comcast.net)
-- |
-- | Program Name: XXX_XLA_WIP_ACCTS_DETAIL_REPT.sql
-- |
-- | Parameters:
-- | P_OPERATING_UNIT      -- operating unit parameter
-- | P_TRX_DATE_FROM      -- starting transaction date
-- | P_TRX_DATE_TO        -- ending transaction date
-- |
-- | Description:
-- | Report to get the WIP accounting entries by account, transaction
-- | type and accounting line type.
-- |
-- | Version Modified on Modified by Description
-- | =====
-- | 1.0 06 Nov 2009 Douglas Volz Initial Coding
-- | 2.0 11 Nov 2009 Douglas Volz Fixed join for primary_assembly_id
-- | 3.0 11 Nov 2009 Douglas Volz Added Org Code and transaction ID
-- | 4.0 12 Nov 2009 Douglas Volz Added item and description
-- | 5.0 05 Jan 2010 Douglas Volz Added org joins for wac and wdj, to avoid
-- | cross-joining
-- +-----+*/

```

```

select
    gl.name "Ledger",
    haou2.name "Operating Unit",
    mp.organization_code "Org Code",
    oap.period_name "Period Name",
    ml.meaning "Acct Line Type",
    ml2.meaning "WIP Transaction Type",
    wta.transaction_id "Txn Id",
    gccl.segment1 "Co",
    gccl.segment2 "Cost Ctr",
    gccl.segment3 "Acct",
    gccl.segment4 "Sub-Acct",
    gccl.segment5 "Prod Grp",
    gccl.segment6 "Loc",
    gccl.segment7 "InterCo",
    gccl.segment8 "Future1",
    gccl.segment9 "Future2",
    DECODE(wta.accounting_line_type, 7, 'WIP', 14, 'Intransit', 1, 'Subinventory') "Account
Type",

```

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

```
msi.segment1 "Item",
msi.description "Description",
wac.class_code "WIP Class",
sum(nvl(al.ACCOUNTED_DR,0) - nvl(al.ACCOUNTED_CR,0)) "Amount"
from wip.wip_transaction_accounts wta,
wip.wip_transactions wt,
wip.wip_accounting_classes wac,
wip.wip_discrete_jobs wdj,
apps.mfg_lookups ml2,
inv.mtl_system_items_b msi,
inv.org_acct_periods oap,
apps.GL_CODE_COMBINATIONS_KFV gccl,
inv.mtl_parameters mp,
apps.mfg_lookups ml,
hr.HR_ORGANIZATION_INFORMATION hoi,
hr.hr_all_organization_units haou, -- inv_organization_id
hr.hr_all_organization_units haou2, -- operating unit
gl.gl_ledgers gl,
xla.xla_transaction_entities ent,
xla.xla_events xe,
xla.xla_distribution_links xdl,
xla.xla_ae_headers ah,
xla.xla_ae_lines al
-- =====
-- Material Transaction, org and item joins
-- =====
where wta.transaction_id = wt.transaction_id
and wta.organization_id = msi.organization_id
-- fix for version 2.0, use wdj not wt
and wdj.primary_item_id = msi.inventory_item_id
--and wt.primary_item_id = msi.inventory_item_id
-- end fix for version 2.0
-- and wt.transaction_type = 4 -- WIP cost update
and mp.organization_id = msi.organization_id
-- get all of them to see the offset accounts
--and wta.accounting_line_type in (1,14)
-- 1 = Inventory valuation entries
-- 14 = Intransit valuation entries
-- =====
and wac.class_code = wdj.class_code
and wdj.wip_entity_id = wt.wip_entity_id
and wac.organization_id = wdj.organization_id
-- =====
-- Inventory Org accounting period joins
-- =====
and oap.acct_period_id = wt.acct_period_id
-- =====
-- Version 1.3, added lookup values to see more detail
-- =====
and ml.lookup_type = 'CST_ACCOUNTING_LINE_TYPE'
and ml.lookup_code = wta.accounting_line_type
and ml2.lookup_type = 'WIP_TRANSACTION_TYPE_SHORT'
and ml2.lookup_code = wt.transaction_type
-- =====
-- Material Transaction date and accounting code joins
-- =====
and wta.transaction_date >= to_date('&P_TRX_FROM_YYYY_MM_DD', 'YYYY/MM/DD HH24:MI:SS')
-- P_TRX_DATE_FROM
and wta.transaction_date < to_date('&P_TRX_FROM_YYYY_MM_DD', 'YYYY/MM/DD HH24:MI:SS')+1
-- P_TRX_DATE_TO
-- =====
-- using the base tables to avoid the performance issues
-- with org_organization_definitions and hr_operating_units
-- =====
and hoi.org_information_context = 'Accounting Information'
and hoi.organization_id = wta.organization_id
and hoi.organization_id = haou.organization_id -- this gets the organization name
and haou2.organization_id = to_number(hoi.org_information3)
-- this gets the operating unit id
and gl.ledger_id = to_number(hoi.org_information1) -- get the ledger_id
-- =====
```

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

```
-- SLA table joins to get the exact account numbers
-- =====
AND   ent.entity_code           = 'WIP_ACCOUNTING_EVENTS'
AND   ent.application_id        = 707
AND   xe.application_id         = ent.application_id
AND   xe.event_id              = xdl.event_id
AND   ah.entity_id             = ent.entity_id
AND   ah.ledger_id            = ent.ledger_id
AND   ah.application_id        = al.application_id
AND   ah.application_id        = 707
AND   ah.event_id              = xe.event_id
AND   ah.ae_header_id          = al.ae_header_id
AND   al.application_id        = ent.application_id
AND   al.ledger_id            = ah.ledger_id
AND   al.AE_HEADER_ID          = xdl.AE_HEADER_ID
AND   al.AE_LINE_NUM           = xdl.AE_LINE_NUM
AND   xdl.application_id        = ent.application_id
AND   xdl.source_distribution_type = 'WIP_TRANSACTION_ACCOUNTS'
AND   xdl.source_distribution_id_num_1 = wta.wip_sub_ledger_id
AND   gccl.code_combination_id = al.code_combination_id
-- =====
group by
    gl.name,
    haou2.name,
    mp.organization_code,
    oap.period_name,
    ml.meaning,
    ml2.meaning,
    wta.transaction_id,
    gccl.segment1,
    gccl.segment2,
    gccl.segment3,
    gccl.segment4,
    gccl.segment5,
    gccl.segment6,
    gccl.segment7,
    gccl.segment8,
    gccl.segment9,
    -- DECODE(wta.accounting_line_type, 7, 'WIP', 14, 'Intransit', 1, 'Subinventory'),
    msi.segment1,
    msi.description,
    wac.class_code
order by 1,2;
```

5. Sample Release 12 ICP Inventory Value Report:

```
/* +-----+
-- | Copyright 2009 Douglas Volz Consulting, Inc.
-- | All rights reserved.
-- | Permission to use this code is granted provided the original author is
-- | acknowledged. No warranties, express or otherwise is included in this
-- | permission.
-- | +-----+
-- |
-- | Original Author: Douglas Volz (davolz@comcast.net)
-- |
-- | Program Name: XXX_ICP_INVENTORY_VAL_REPT.sql
-- |
-- | Parameters:
-- | P_PERIOD_NAME           -- Accounting period you wish to report for
-- |                         This parameter is also used to get the correct
-- |                         currency conversion rate
-- | P_ICP_COST_TYPE         -- The name of the cost type that has that
-- |                         month's ICP costs
-- |
-- | =====
-- | NOTE: THIS SCRIPT USES THE ITEM COSTS FROM THE MONTH-END SNAPSHOT;
-- |       THIS SCRIPT ONLY NEEDS THE ICP COSTS IN THE ICP COST TYPE.
-- |
```

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

```

-- | =====
-- |
-- | Description:
-- | Report to show month-end inventory values in the period end snapshot that is
-- | created when you close the inventory periods. This report also shows the
-- | material account for each subinventory and intransit location.
-- |
-- | Version Modified on Modified by Description
-- | =====
-- | 1.0      27 Sep 2009 Douglas Volz Initial Coding
-- | 1.1      28 Sep 2009 Douglas Volz Added a sum for the ICP costs from cicd
-- | 1.2      01 Oct 2009 Douglas Volz Removed the addition of cpcs.ROLLBACK_INTRANSIT_VALUE
-- |                                     from the first query (had zero value) and
-- |                                     removed the addition of cpcs.ROLLBACK_VALUE from
-- |                                     the second Intransit query, which was doubling
-- |                                     up the intransit value.
-- | 1.3      19 Nov 2009 Douglas Volz Added the G/L Ledger Name to the report
-- | 1.4      03 Mar 2010 Douglas Volz Made the ICP cost type a parameter
-- |                                     and set the company account number based on
-- |                                     the item's cost of goods sold account
-- | =====+*/
--
-- =====
-- The first select gets the period-end quantities from the subinventories
-- =====
select gl.name "Ledger",
       haou2.name "Operating Unit",
       oap.period_name "Period Name",
       mp.organization_code "Org Code",
       gcc2.segment1 "Co",
       gccl.segment2 "Cost Ctr",
       gccl.segment3 "Acct",
       gccl.segment4 "Sub-Acct",
       gcc2.segment5 "Prod Grp",
       gccl.segment6 "Loc",
       gccl.segment7 "InterCo",
       gccl.segment8 "Future1",
       gccl.segment9 "Future2",
       cpcs.subinventory_code "Subinv/Intransit",
       msi.segment1 "Item Number",
       msi.description "Description",
       gl.currency_code "Curr",
       round((nvl(cpcs.ROLLBACK_VALUE,0)) /
             decode(nvl(cpcs.ROLLBACK_QUANTITY,0), 0, 1,
                   nvl(cpcs.ROLLBACK_QUANTITY,0)),5) "Gross Item Cost",
-- Revision for version 1.1
nvl((select sum(nvl(cicd.ITEM_COST,0))
     from bom.cst_item_cost_details cicd,
          bom.cst_cost_types cct,
          bom.bom_resources br
     where cicd.inventory_item_id = msi.inventory_item_id
           and cicd.organization_id = msi.organization_id
           and br.resource_id = cicd.resource_id
           and br.resource_code = 'ICP'
           and cct.cost_type_id = cicd.cost_type_id
           and cct.cost_type = '&P_ICP_COST_TYPE'),0) "ICP Item Cost", -- P_COST_TYPE
round((nvl(cpcs.ROLLBACK_VALUE,0)) /
      decode(nvl(cpcs.ROLLBACK_QUANTITY,0), 0, 1,
            nvl(cpcs.ROLLBACK_QUANTITY,0)),5) +
-- Revision for version 1.1
(nvl((select sum(nvl(cicd.ITEM_COST,0))
     from bom.cst_item_cost_details cicd,
          bom.cst_cost_types cct,
          bom.bom_resources br
     where cicd.inventory_item_id = msi.inventory_item_id
           and cicd.organization_id = msi.organization_id
           and br.resource_id = cicd.resource_id
           and br.resource_code = 'ICP'
           and cct.cost_type_id = cicd.cost_type_id
           and cct.cost_type = '&P_ICP_COST_TYPE'),0)) "Net Item Cost",
-- P_COST_TYPE

```

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

```

msi.primary_uom_code "UOM",
sum(nvl(cpcs.ROLLBACK_QUANTITY,0)) "Quantity",
sum(nvl(cpcs.ROLLBACK_VALUE,0)) "Inv. Onhand Value",
round(sum(nvl(cpcs.ROLLBACK_QUANTITY,0)) *
-- Revision for version 1.1
      nvl((select sum(nvl(cicd.ITEM_COST,0))
           from   bom.cst_item_cost_details cicd,
                  bom.cst_cost_types cct,
                  bom.bom_resources br
           where  cicd.inventory_item_id = msi.inventory_item_id
                  and  cicd.organization_id = msi.organization_id
                  and  br.resource_id = cicd.resource_id
                  and  br.resource_code = 'ICP'
                  and  cct.cost_type_id = cicd.cost_type_id
                  and  cct.cost_type = '&P_ICP_COST_TYPE'),0),2)
      "Total ICP Onhand Value", -- P_COST_TYPE

sum(nvl(cpcs.ROLLBACK_VALUE,0)) +
round(sum(nvl(cpcs.ROLLBACK_QUANTITY,0)) *
-- Revision for version 1.1
      nvl((select sum(nvl(cicd.ITEM_COST,0))
           from   bom.cst_item_cost_details cicd,
                  bom.cst_cost_types cct,
                  bom.bom_resources br
           where  cicd.inventory_item_id = msi.inventory_item_id
                  and  cicd.organization_id = msi.organization_id
                  and  br.resource_id = cicd.resource_id
                  and  br.resource_code = 'ICP'
                  and  cct.cost_type_id = cicd.cost_type_id
                  and  cct.cost_type = '&P_ICP_COST_TYPE'),0),2)
      "Total Net Onhand Value" -- P_COST_TYPE

from   bom.cst_period_close_summary cpcs,
       inv.org_acct_periods oap,
       inv.mtl_parameters mp,
       inv.mtl_system_items_b msi,
       inv.mtl_secondary_inventories msub,
       gl.gl_code_combinations gcc1, -- subinventory accounts
       gl.gl_code_combinations gcc2, -- product group info from COGS account
       hr.hr_organization_information hoi,
       hr.hr_all_organization_units haou,
       hr.hr_all_organization_units haou2,
       gl.gl_ledgers gl

-- =====
-- Inventory accounting period joins
-- =====
where  oap.acct_period_id = cpcs.acct_period_id
and    oap.organization_id = mp.organization_id
and    oap.period_name = '&PERIOD_NAME'
-- P_PERIOD_NAME
-- =====
-- Subinventory, mtl parameter, item master and period close snapshot joins
-- =====
and    msub.SECONDARY_INVENTORY_NAME = cpcs.subinventory_code
and    msub.organization_id = cpcs.organization_id
and    mp.organization_id = cpcs.organization_id
and    mp.organization_id = msi.organization_id
and    msi.organization_id = cpcs.organization_id
and    msi.inventory_item_id = cpcs.inventory_item_id
-- =====
-- Accounting code combination joins
-- =====
and    msub.material_account = gcc1.code_combination_id
and    msi.cost_of_sales_account = gcc2.code_combination_id
-- =====
-- Organization joins to the HR org model
-- =====
and    hoi.org_information_context = 'Accounting Information'
and    hoi.organization_id = mp.organization_id
and    hoi.organization_id = haou.organization_id -- this gets the organization name
and    haou2.organization_id = to_number(hoi.org_information3)
-- this gets the operating unit id
and    gl.ledger_id = to_number(hoi.org_information1) -- get the ledger_id

```

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

```

-- =====
-- limit the rows returned-don't get zero rows
-- =====
and      nvl(cpcs.ROLLBACK_QUANTITY,0) <> 0
group by
    gl.name,
    haou2.name,
    oap.period_name,
    mp.organization_code,
    gcc2.segment1,
    gccl.segment2,
    gccl.segment3,
    gccl.segment4,
    gcc2.segment5,
    gccl.segment6,
    gccl.segment7,
    gccl.segment8,
    gccl.segment9,
    cpcs.subinventory_code,
    msi.segment1,
    msi.description,
    gl.currency_code,
    round((nvl(cpcs.ROLLBACK_VALUE,0)) /
        decode(nvl(cpcs.ROLLBACK_QUANTITY,0), 0, 1,
            nvl(cpcs.ROLLBACK_QUANTITY,0)),5),
    msi.inventory_item_id,
    msi.organization_id,
    msi.primary_uom_code
UNION ALL
-- =====
-- The second select gets the period-end quantities from intransit
-- =====
select gl.name "Ledger",
    haou2.name "Operating Unit",
    oap.period_name "Period Name",
    mp.organization_code "Org Code",
    gcc2.segment1 "Co",
    gccl.segment2 "Cost Ctr",
    gccl.segment3 "Acct",
    gccl.segment4 "Sub-Acct",
    gcc2.segment5 "Prod Grp",
    gccl.segment6 "Loc",
    gccl.segment7 "InterCo",
    gccl.segment8 "Future1",
    gccl.segment9 "Future2",
    'Intransit' "Subinv/Intransit",
    msi.segment1 "Item Number",
    msi.description "Description",
    gl.currency_code "Curr",
    round((nvl(cpcs.ROLLBACK_INTRANSIT_VALUE,0)) /
        decode(nvl(cpcs.ROLLBACK_QUANTITY,0), 0, 1,
            nvl(cpcs.ROLLBACK_QUANTITY,0)),5) "Gross Item Cost",
    -- Revision for version 1.1
    nvl((select sum(nvl(cicd.ITEM_COST,0))
        from bom.cst_item_cost_details cicd,
            bom.cst_cost_types cct,
            bom.bom_resources br
        where cicd.inventory_item_id = msi.inventory_item_id
            and cicd.organization_id = msi.organization_id
            and br.resource_id = cicd.resource_id
            and br.resource_code = 'ICP'
            and cct.cost_type_id = cicd.cost_type_id
            and cct.cost_type = '&P_ICP_COST_TYPE'),0)
        "ICP Item Cost", -- P_COST_TYPE
    round((nvl(cpcs.ROLLBACK_INTRANSIT_VALUE,0)) /
        decode(nvl(cpcs.ROLLBACK_QUANTITY,0), 0, 1,
            nvl(cpcs.ROLLBACK_QUANTITY,0)),5) +
    -- Revision for version 1.1
    (nvl((select sum(nvl(cicd.ITEM_COST,0))
        from bom.cst_item_cost_details cicd,
            bom.cst_cost_types cct,

```

Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

```

        bom.bom_resources br
    where  cicd.inventory_item_id = msi.inventory_item_id
    and    cicd.organization_id   = msi.organization_id
    and    br.resource_id        = cicd.resource_id
    and    br.resource_code      = 'ICP'
    and    cct.cost_type_id      = cicd.cost_type_id
    and    cct.cost_type         = '&P_ICP_COST_TYPE'),0))
                                                "Net Item Cost", -- P_COST_TYPE

msi.primary_uom_code "UOM",
sum(nvl(cpcs.ROLLBACK_QUANTITY,0)) "Quantity",
sum(nvl(cpcs.ROLLBACK_INTRANSIT_VALUE,0)) "Inv. Onhand Value",
round(sum(nvl(cpcs.ROLLBACK_QUANTITY,0)) *
-- Revision for version 1.1
    nvl((select sum(nvl(cicd.ITEM_COST,0))
        from  bom.cst_item_cost_details cicd,
             bom.cst_cost_types cct,
             bom.bom_resources br
        where  cicd.inventory_item_id = msi.inventory_item_id
        and    cicd.organization_id   = msi.organization_id
        and    br.resource_id        = cicd.resource_id
        and    br.resource_code      = 'ICP'
        and    cct.cost_type_id      = cicd.cost_type_id
        and    cct.cost_type         = '&P_ICP_COST_TYPE'),0),2)
                                                "Total ICP Onhand Value", -- P_COST_TYPE

sum(nvl(cpcs.ROLLBACK_INTRANSIT_VALUE,0)) +
round(sum(nvl(cpcs.ROLLBACK_QUANTITY,0)) *
-- Revision for version 1.1
    nvl((select sum(nvl(cicd.ITEM_COST,0))
        from  bom.cst_item_cost_details cicd,
             bom.cst_cost_types cct,
             bom.bom_resources br
        where  cicd.inventory_item_id = msi.inventory_item_id
        and    cicd.organization_id   = msi.organization_id
        and    br.resource_id        = cicd.resource_id
        and    br.resource_code      = 'ICP'
        and    cct.cost_type_id      = cicd.cost_type_id
        and    cct.cost_type         = '&P_ICP_COST_TYPE'),0),2)
                                                "Total Net Onhand Value" -- P_COST_TYPE

from  bom.cst_period_close_summary cpcs,
      inv.org_acct_periods oap,
      inv.mtl_parameters mp,
      inv.mtl_system_items_b msi,
      gl.gl_code_combinations gcc1, -- subinventory accounts
      gl.gl_code_combinations gcc2, -- product group info from COGS account
      hr.hr_organization_information hoi,
      hr.hr_all_organization_units haou,
      hr.hr_all_organization_units haou2,
      gl.gl_ledgers gl
-- =====
-- Inventory accounting period joins
-- =====
where  oap.acct_period_id      = cpcs.acct_period_id
and    oap.organization_id    = mp.organization_id
and    oap.period_name        = '&PERIOD_NAME'
-- P_PERIOD_NAME
-- =====
-- Subinventory, mtl parameter, item master and period close snapshot joins
-- =====
and    cpcs.subinventory_code  IS NULL
and    mp.organization_id      = cpcs.organization_id
and    mp.organization_id      = msi.organization_id
and    msi.organization_id     = cpcs.organization_id
and    msi.inventory_item_id   = cpcs.inventory_item_id
-- =====
-- Accounting code combination joins
-- =====
and    mp.intransit_inv_account = gcc1.code_combination_id
and    msi.cost_of_sales_account = gcc2.code_combination_id
-- =====
-- Organization joins to the HR org model
-- =====

```


Does Rel. 12 Solve Global Inter-Company Issues for Multiple Ledgers, Profit in Inventory and COGS?

```
and    hoi.org_information_context    = 'Accounting Information'
and    hoi.organization_id            = mp.organization_id
and    hoi.organization_id            = haou.organization_id  -- this gets the organization name
and    haou2.organization_id          = to_number(hoi.org_information3)
-- this gets the operating unit id
and    gl.ledger_id                  = to_number(hoi.org_information1) -- get the ledger_id
-- =====
-- limit the rows returned-don't get zero rows
-- =====
and    nvl(cpcs.ROLLBACK_QUANTITY,0) <> 0
group by
    gl.name,
    haou2.name,
    oap.period_name,
    mp.organization_code,
    gcc2.segment1,
    gccl.segment2,
    gccl.segment3,
    gccl.segment4,
    gcc2.segment5,
    gccl.segment6,
    gccl.segment7,
    gccl.segment8,
    gccl.segment9,
    'Intransit',
    msi.segment1,
    msi.description,
    gl.currency_code,
    round(nvl(cpcs.ROLLBACK_INTRANSIT_VALUE,0) /
        decode(nvl(cpcs.ROLLBACK_QUANTITY,0), 0, 1,
            nvl(cpcs.ROLLBACK_QUANTITY,0)),5),
    msi.inventory_item_id,
    msi.organization_id,
    msi.primary_uom_code
order by 1,2,3,4,6;
```