IRIS Exchequer MS SQL Edition

Product Improvement Plan v0.2

Mark Roke August 2009

MS SQL Edition – Identified areas for improvement

1. Reporting – Standard reports and Visual Report Writer (All)
    - Daybook Pre-posting reports (Adam Rouilly)
2. Transaction Line Entry – Kitted BOMS (ASA)
3. Common Routines
    - Daybook Process – Deliver Picked Orders (Sales Daybook)
    - Daybook Process – Invoice All Deliveries (Sales Daybook)
    - Daybook Process – Deliver Received Orders (Purchase Daybook)
    - Daybook Process – Invoice Received Orders (Purchase Daybook)
4. Toolkits – reduce reliance on emulator

1. Reporting

Research and development time to be allocated to produce a set of default reports using Reporting Services that would replace the current hard coded standard reports.

- Replace the current hard coded reports with a set of Reporting Services Reports (RSR)
- RSR's accessed via IRIS Exchequer menu structure
- Ability to copy RSR for customisation by users
- Default RSR to be available at all times

Based on a straw poll conducted in July 09, we should aim to include the top five reports

Report	Votes
Aged Reports	12
TB	11
Stock Valuation	8
GL History	8
Bank rec	5

A similar approach too should be considered for the Sales / Purchase Daybook Reports accessed from the daybooks. Preliminary investigations have revealed performance gains by using reporting services

Basis	IRIS Exchequer	Direct SQL (Management Studio) report	Direct SQL Reporting Services	Diff (Direct SQL using Management Studio & IRIS Exchequer)	Diff (Direct SQL using reporting Services & IRIS Exchequer)
Document	6.33	1.63	4.13	-4.7	-2.2
GL	240	3.68	6.18	-236.32	-233.82

Timings in seconds

NB – for Reporting Services and average of 2.5 seconds has been added to allow for the start up of Internet Explorer which is where the reports are published.

Indicative project start date: December 09

2. Transaction Line Entry – Kitted BOMS

Research and development time to be allocated to assess the feasibility of replacing current code/routines and stored procedures to improve performance. (Reported by ASA)

When adding a BOM stock code to a transaction, where the BOM stock record is set to “explode components” IRIS Exchequer will add the individual component items to the resulting transaction as individual transaction lines and at this point, stock records and job costing records are updated. On storing the transaction line, the system will then look up each individual component and where necessary update any quantity totals. The adding of the stock code element takes approximately 6 seconds, with the line store taking 5 seconds in the MS SQL edition.

Further to a high level review of the system, to improve the performance in this area, three separate routines would need deeper investigation with a strong possibility that these routines would need replacing with stored procedures.

Indicative project start date: January 10

Delivery of improvements

It should be considered following the success of the v6.00.001S update pack released in August 09 to use the same deployment method for these improvements

3. Common Routines

Research and development time to be allocated to assess the feasibility of replacing current code/routines and stored procedures to improve performance.

The daybook process routines, whilst functionally sound, when compared to the Pervasive edition there is performance degradation. This was most noticeable during data investigations for our customer ASA.

Indicative project start date: January 10

4. Toolkits / Customisation

Research and development time to be allocated to assess the feasibility of replacing current code/routines and stored procedures to improve performance.

Validation of line or translation store is where significant performance issues can be seen. This has been seen where the cost centre / department validation plug-in is present.

During the development of the SQL edition the topic of poor performance when using the toolkits was constantly

Indicative project start date: January 10

General Performance

The common cause of performance issues within the MS SQL editor is due to the dependency on the SQL emulator. This is further compounded by database design of the IRIS Exchequer product. In its current design which can be described as non-SQL friendly, and because of this, calls made to the database via the emulator in many cases are having to process more lines of code to return information than if the emulator layer was not present.

Database normalisation would be the first step in bring performance improvements to the MS SQL edition. Replacing existing code with stored procedures may bring some benefit to the system and during the development of the product a number of core routines were replaced (details further on in this document). In some cases though, performance gains were minimal. Where gains were achieved; the performance was still less than that of Pervasive.

Removing the emulator completely from the system would require both database normalisation and total re-writes of code which is not a viable solution.

As part of the ongoing product improvement plan it is my recommendation that following on from the report currently being compiled by both application and engineer support (Aug09) that additional identified system areas are combined with those in this report are prioritised and built into the development schedule. If possible a set period of 6 months minimum is allocated to tackle as many items as is practically possible.

It may become necessary to adopt an “update pack” delivery approach similar to that of 6.00.001S (July 09) but to minimise disruption to customers and ease pressure on the engineer support team, a separate release of MS SQL edition may be deemed more appropriate.

Current Stored Procedures

Posting

PostToHistory

Replaces the LPost_To_Hist2 function in Exchequer which updates the history for all types.
Note: Also effects Check All Stock, Check All Accounts, GL Views, Check Account Stock Analysis.

PostToYearToDate

Replaces the LPost_To_CYTDHist2 function in Exchequer which creates the YTD History.
Note: Also effects Check All Stock, Check All Accounts, GL Views, Check Account Stock Analysis.

Stock Take with Locations

Filter
Copies the Bin Location on the Stock Location record to the Stock record for the Stock Take.

Freeze

Updates the balance and stock level information on stock items for a specified location.

Revalue Stock

Updates the balances on the stock items for a specified location from history.

Check All Stock

DeleteLinks
Called by Re_CalcStockLevels to remove any FIFO Entries for a Stock Item.

DeleteAuditHistory

Called from Re_CalcStockLevels to delete or reset the posted stock levels

Stk_CheckLocHist

Called from Re_CalcStockLevels to run through the Stock Location records resetting the balances.
Can only be used on Non Purged systems, purged systems will have to stick with the existing code.

LCheck_NoteNo

Called from TCheckStk.ProcessFromCheck this routine runs through the Notes for the Stock Item and returns the next available note number.

Check Account

NOTE: Check All Accounts just calls Check Account repeatedly.