Database Definition.

The ePART system is divided into many subsystems.

We are going to use an approach of at first defining all major ERP (non-financial) processes and then zoom into them defining the data flows and which database elements are being used.

A 3-letter identification code (sometimes a 2 letter code) is used to identify each subsystem.

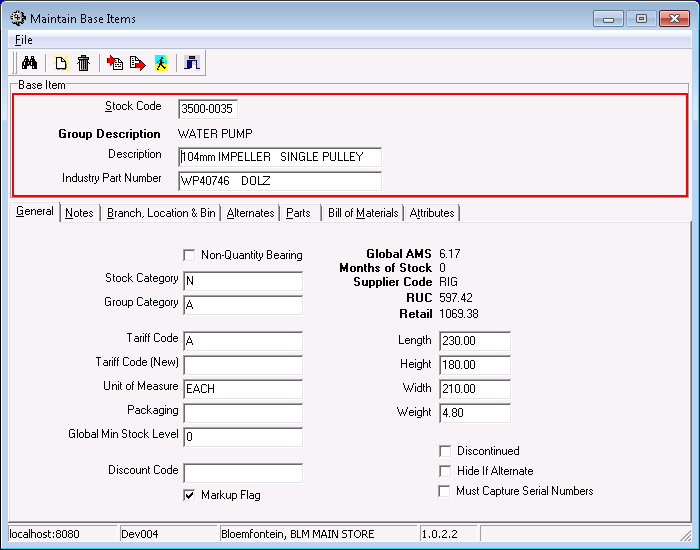
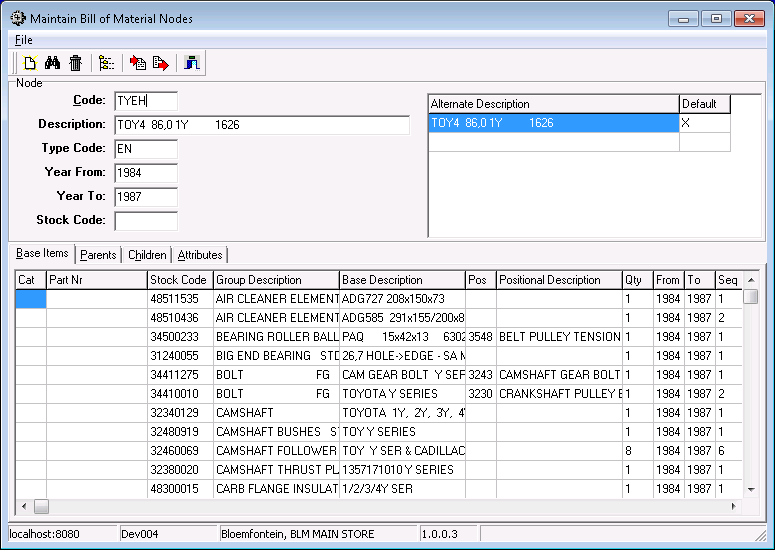
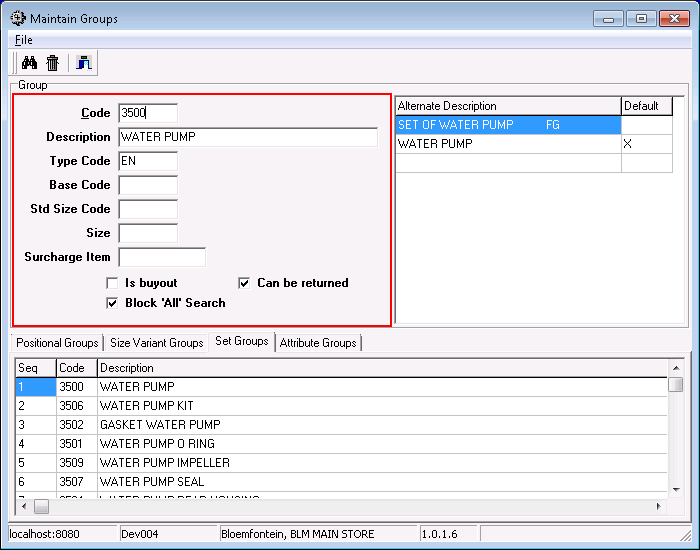
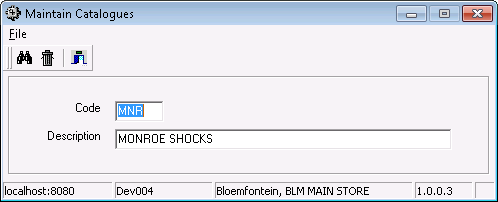
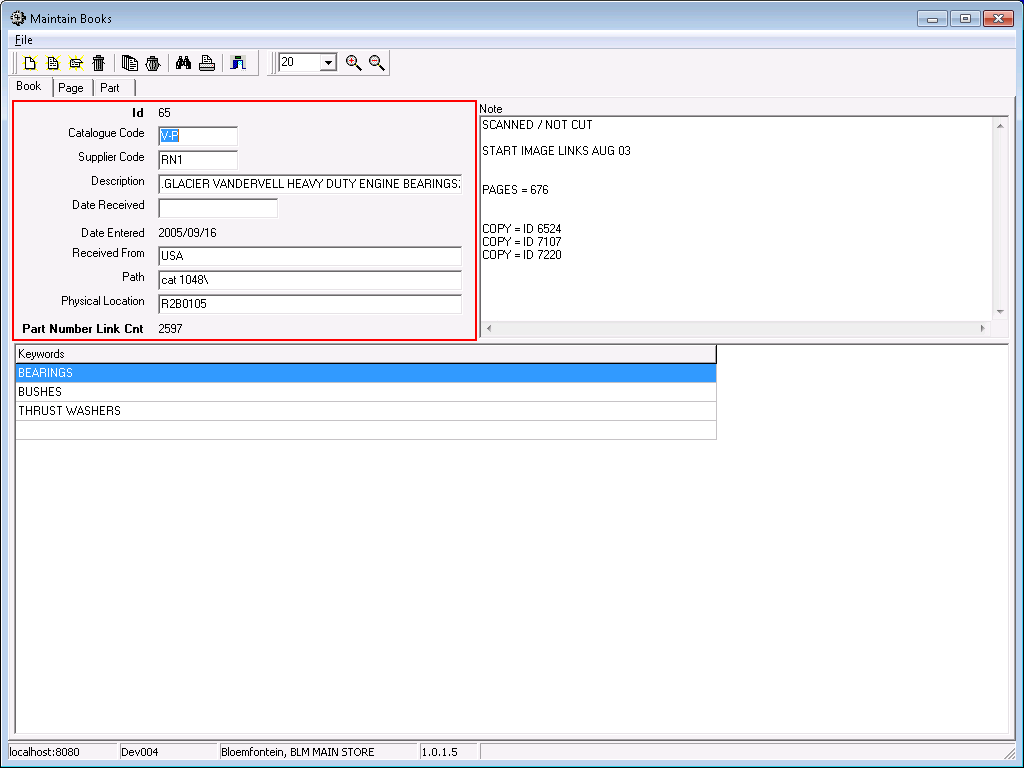
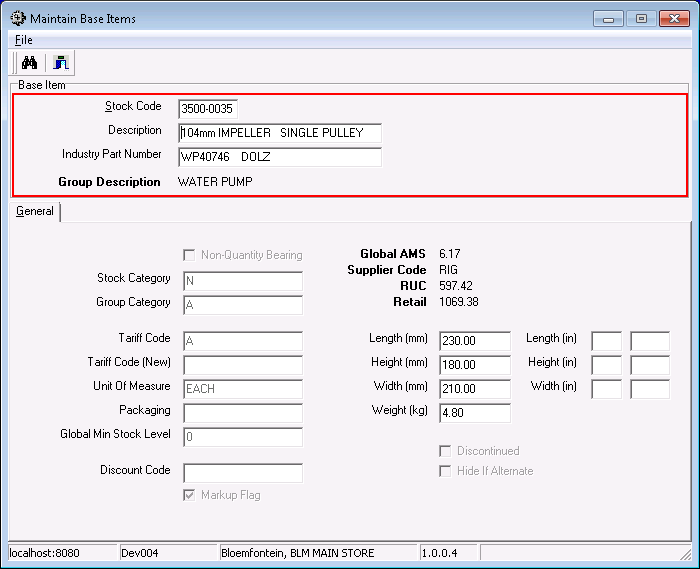
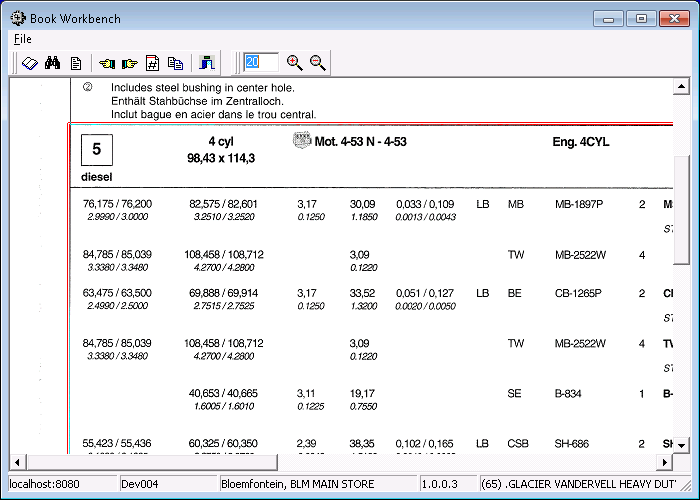
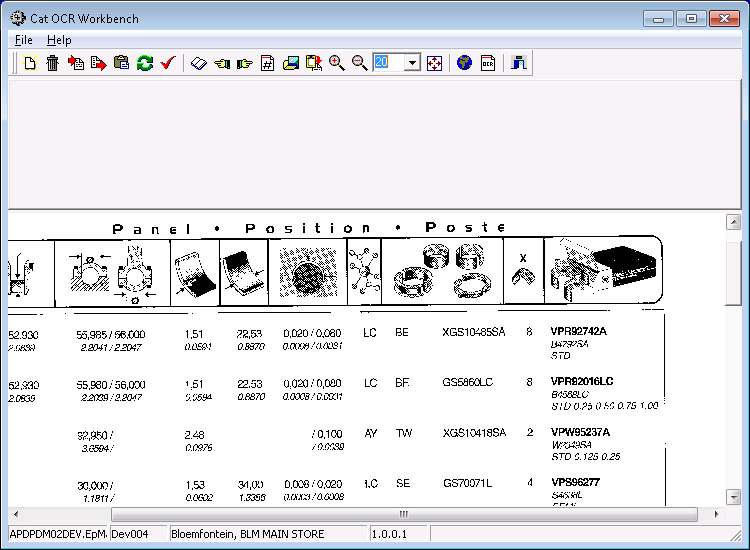
Financial (non-ERP) processes are not described for it is in process of being deprecated in favour of SAGE X3.

|  |  |  |
| --- | --- | --- |
| **Subsystem** | **Code** | **Purpose** |
| Catalogue subsystem | CAT | This subsystem contains all parts and their applications. |
| Pricing subsystem | CAT | Record which part can be bought where and at what price. |
| Purchasing subsystem | STK  REC  PUR | Placement of orders and tracking of shipments. |
| Receiving | REC | Capturing of supplier invoices and labelling and binning of stock. |
| Warehousing | STK | Organizing and planning of warehouse usage.  Stock location. |
| Sales | STK  SAL | Placement of sales orders. |
| Picking | PCK | Retrieving stock from warehouse.  Checking stock.  Packing of stock into parcels.  Printing invoices. |
| Despatch | DS | Load group allocation.  Load onto vehicles.  Proof of delivery notes.  Distribution of parcels. |
| Buyouts | STK | Purchase and Sales of non-stock items. |
| Stock taking | STT | Taking inventory of stock on hand. |
| Company | CO | Maintenance of general data being used by various subsystems e.g. creation of user accounts or of new branches. |
| General | EP | General purpose libraries  Automatic e-mails and reports |
| Management reports | MAN | Sensitive reports.  Sales performance targets. |
| Incident Log | INC | Originally meant as an incident logging subsystem, it is now also used for account notes and automatic messages. |

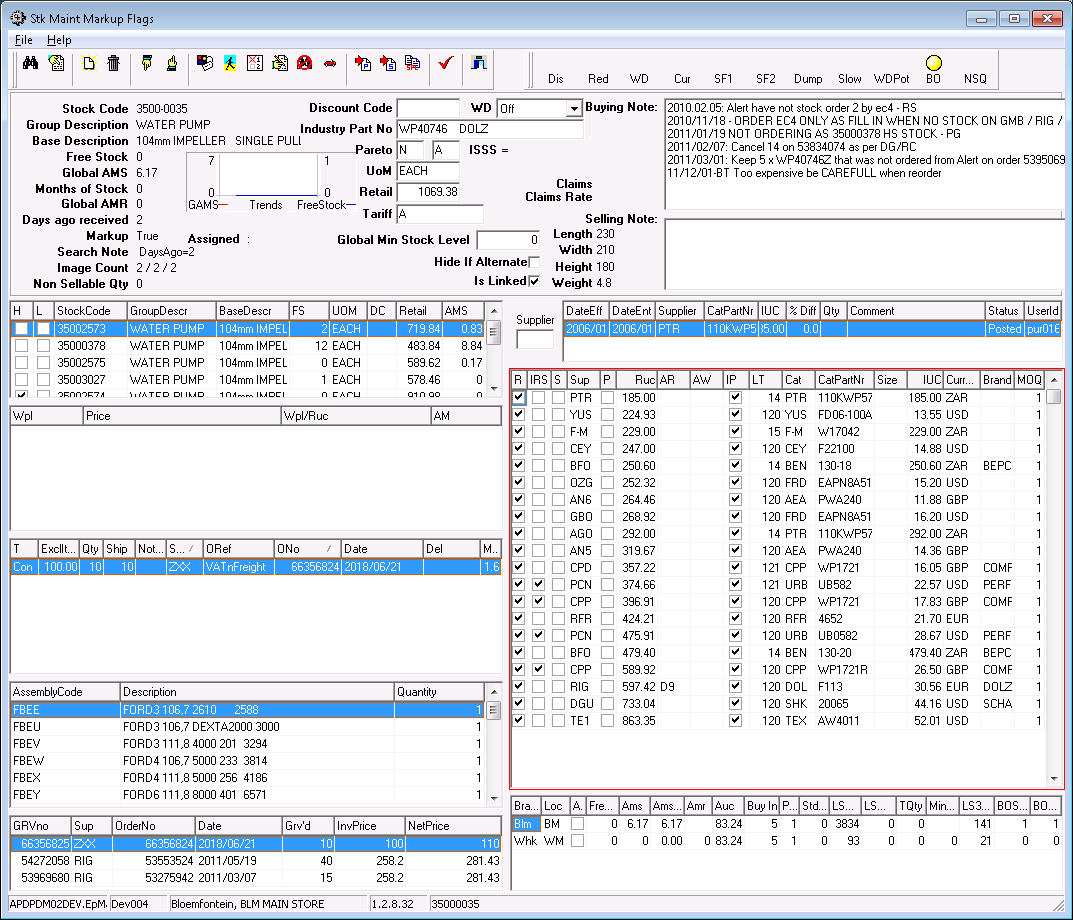
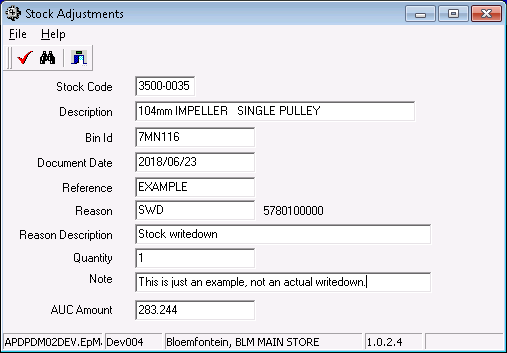
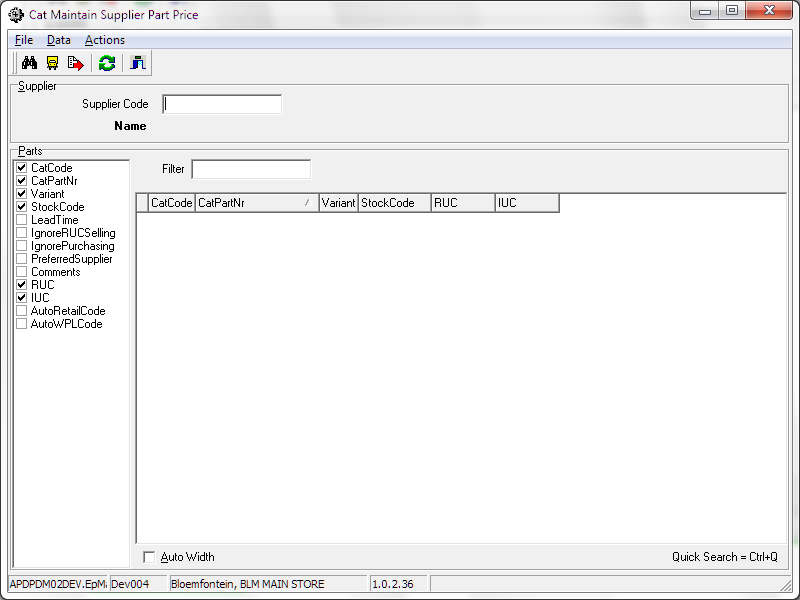
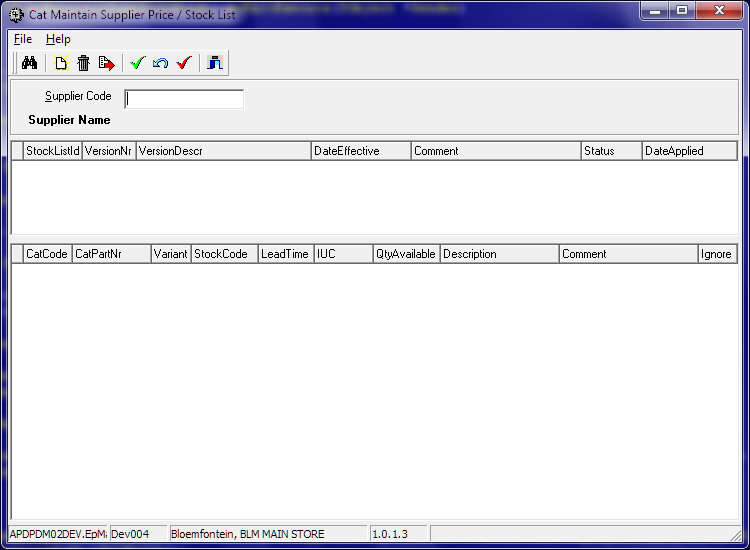
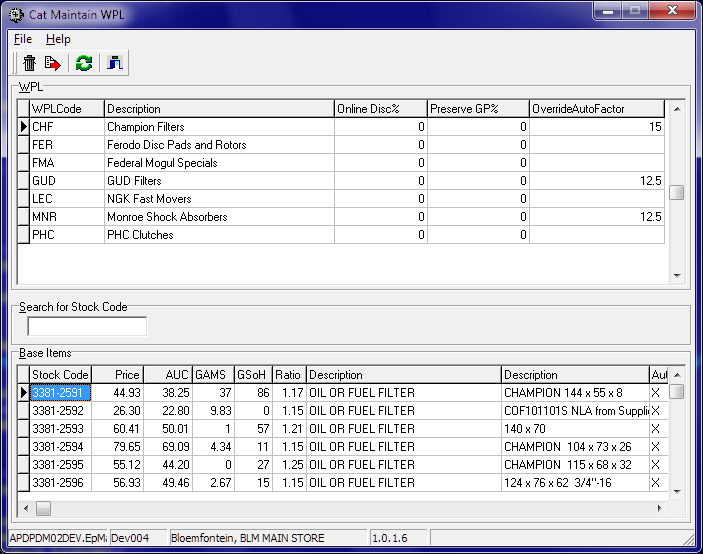
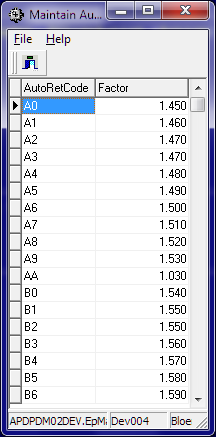
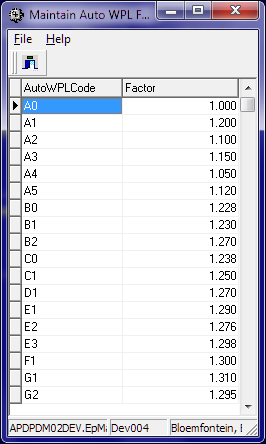
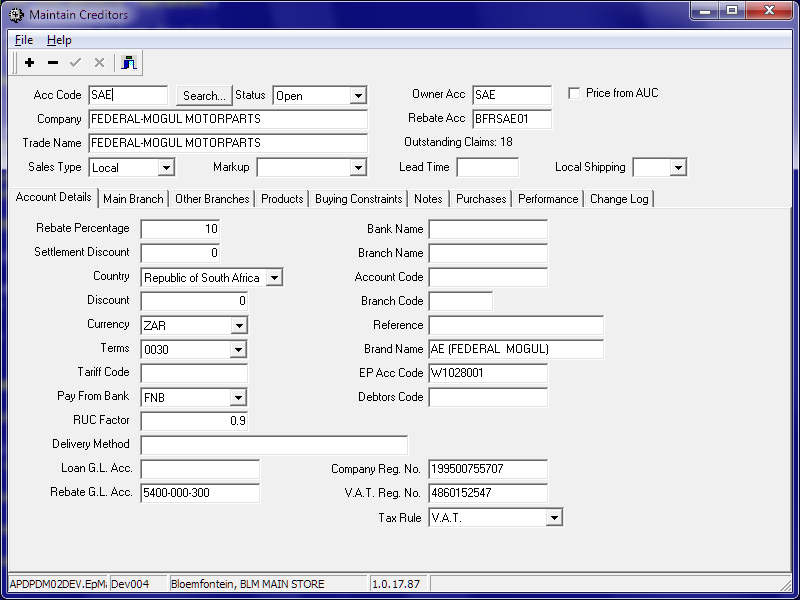
A brief note on security

Access to an application is given to a user by adding a link to the application on the user’s menu. Some applications also require certain *permission codes* to access some of their functionality, or occasionally to switch off certain functionality. There are also sometimes access restrictions to write transactions into specific date periods; these are called *user periods*.

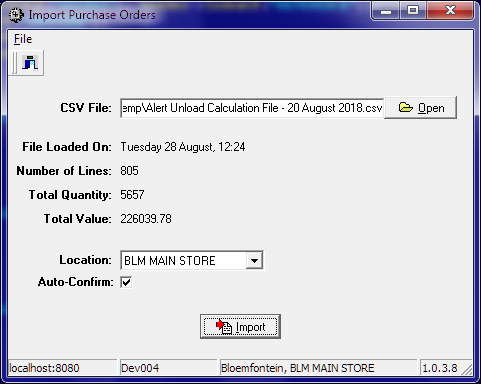
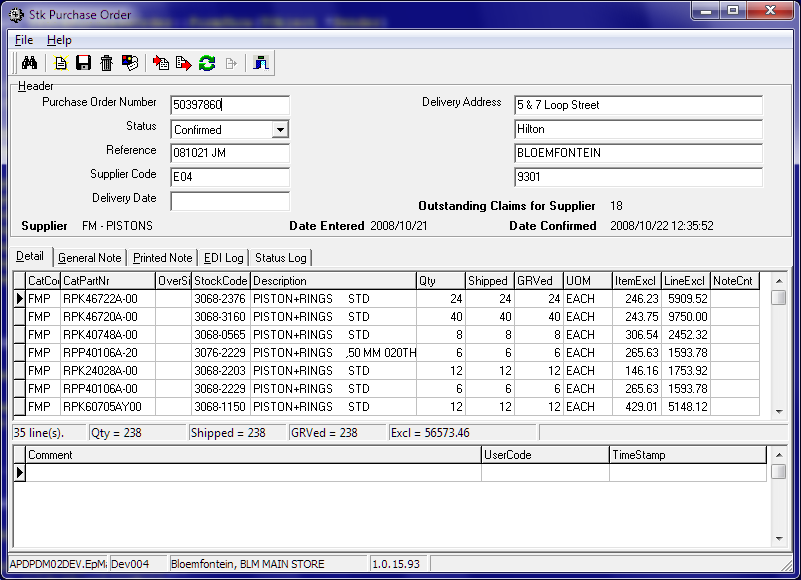
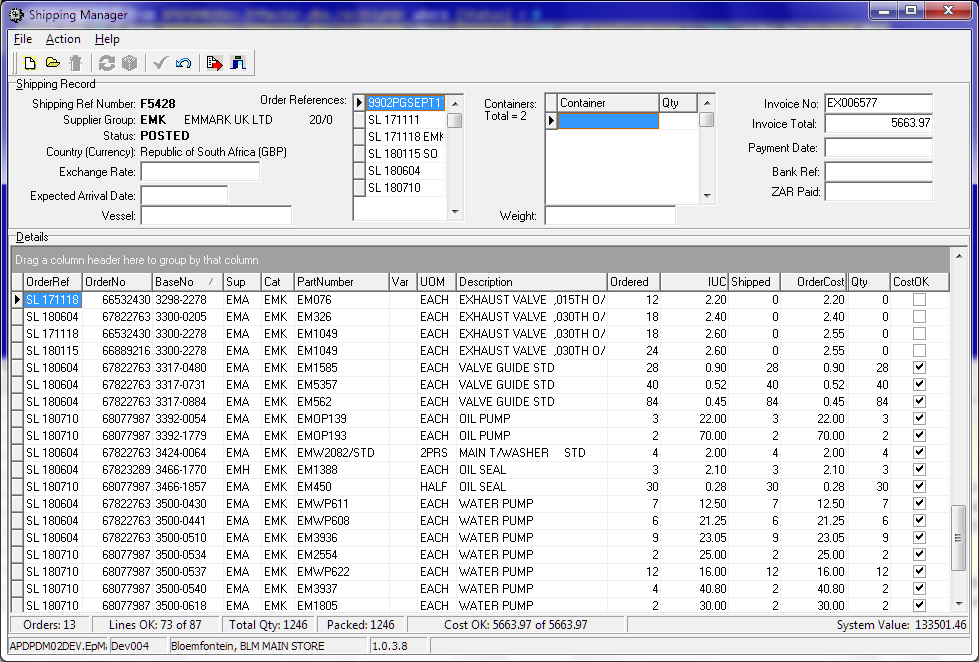
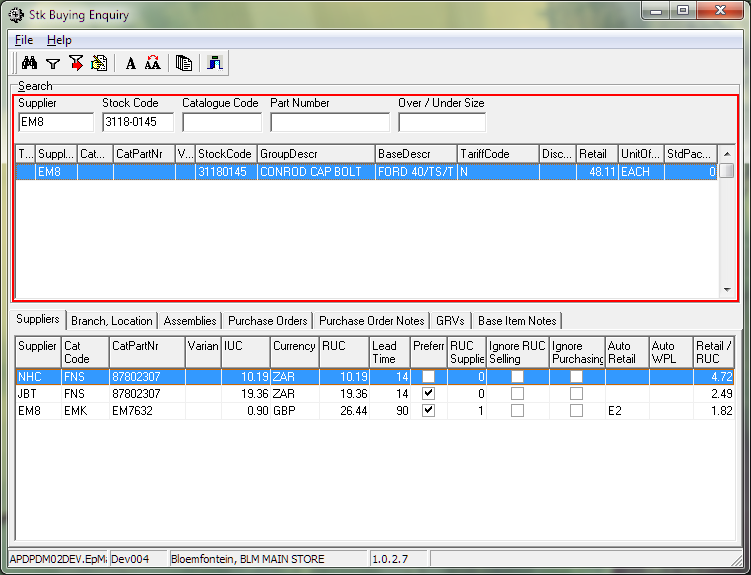
Catalogue subsystem

* Introduction:
  + The purpose of the catalogue subsystem is to keep track of all known parts that we could stock. This allows us to have data related to each part with which to determine what needs to be stocked, regardless of whether we have ever stocked it before.
  + The data is sectioned into catalogues, which are related to the catalogue books received from potential suppliers, and parts, which are groupings of different suppliers’ part numbers matching identical physical items. This is called a “base item”, or just “item”.
  + Many of the catalogue books were disassembled and scanned, then processed into the system as images associated with each part. There was also OCR involved; ABBYY FineReader was used, but the license has expired in the meantime.
  + We also maintain a large database of vehicle models and their respective sub-assemblies, so that we can help a customer to select the right parts to purchase. This is called a “bill of materials” (or “B.o.M.”).
  + Similarly, we have lists of alternate parts for each part, in case we do not have an item in stock, or if the customer needs a cheaper option.
  + Items are divided into groups, for statistical and descriptive reasons.
  + The groups are divided into “super sets”, which in turn is divided into “super set groups”, which are the largest subdivisions.
* Frequently used maintenance apps:
  + catMaintBaseItem.exe  
    
    - Here new base items are created and maintained.
    - They are also associated to branches, indicating whether a branch can stock that base item, and in what pack sizes they are sold and are allowed to be returned etc.
    - The ‘discontinued’ flag is maintained here, but, since it is hardly used by the system, the data is not considered to be accurate. This could be considered to be a vicious circle.
    - This is also the primary location to associate base items to alternate items which may be used as substitutes in some cases.
  + catMaintBomNode.exe  
    
    - This is where the Bill of Materials tree is maintained, by creating nodes, linking them together and adding base items where appropriate.
    - The Stock Code field in the top left panel is only for when the node can be purchased pre-assembled into an item. So far primarily used when a cylinder head is sold with rings rather than without.
  + catMaintGroup.exe  
    
    - Used to maintain most of the properties of item groups.
    - The Group Code is the first four digits of the base items’ Stock Codes.
    - This would be a good place to add the ability to maintain the links between groups and super sets.
  + catMaintCatalogue.exe  
    
    - Apparently only used to create new catalogue codes.
  + catMaintBook.exe  
    
    - Used to create and maintain catalogue images as scanned from physical books.
  + catMaintBaseItemMeasure.exe  
    
    - A version of catMaintBaseItem.exe that has been reduced to only maintain the measurements, and to clarify which units should be used.
  + catBookWorkbench.exe  
    
    - This application can be used to associate parts of page images to catalogue part numbers, although for doing so in bulk catMaintBook.exe would be better.
  + catOCRWorkbench.exe  
    
    - This application was used to apply OCR to scanned catalogue pages.
    - We lack the ABBYY FineReader components that are necessary to recompile it.
    - However, this application also has various data import abilities, including the ability to import data from a CSV file into a supplier price list.
* Description of main tables:
  + catCatalogue
  + catBook
  + catPage
  + catPagePart
  + catPart
  + catBaseItem
* Description of main stored procedures:
* Description of main triggers:

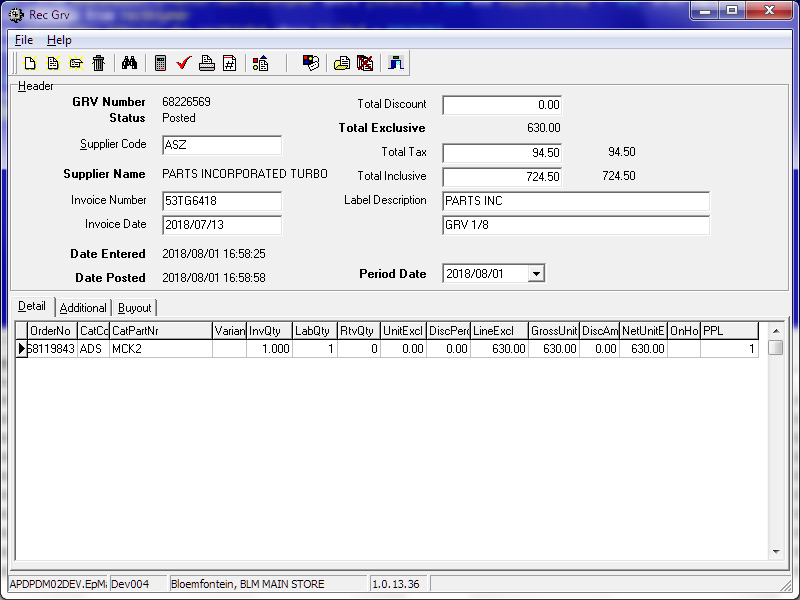
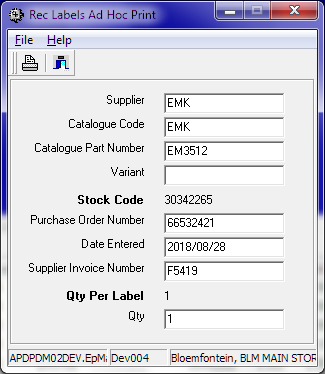
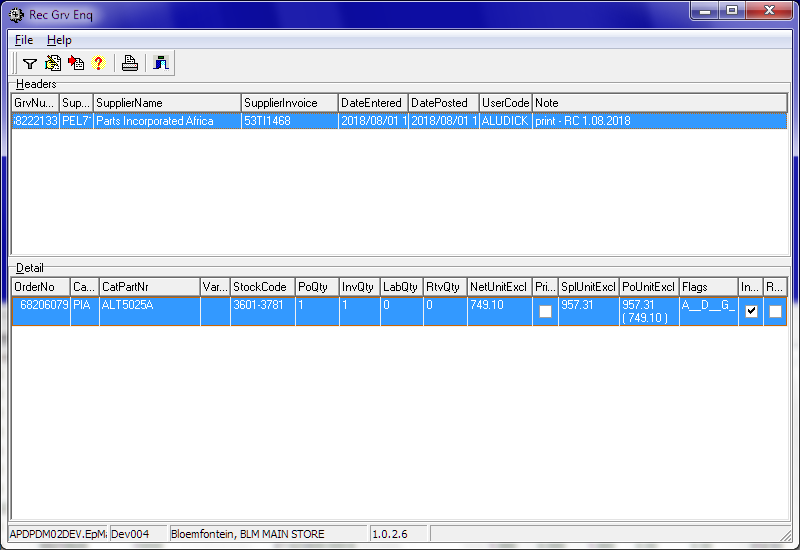
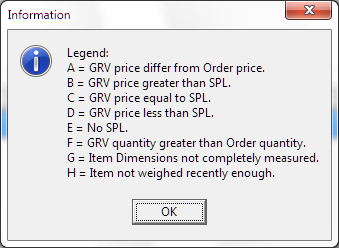
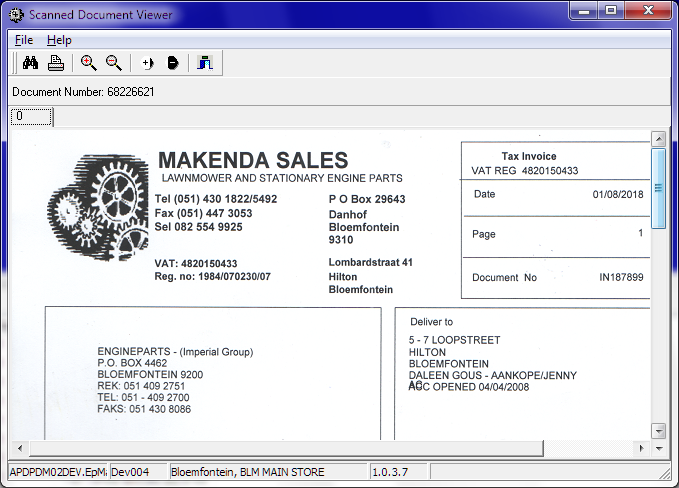
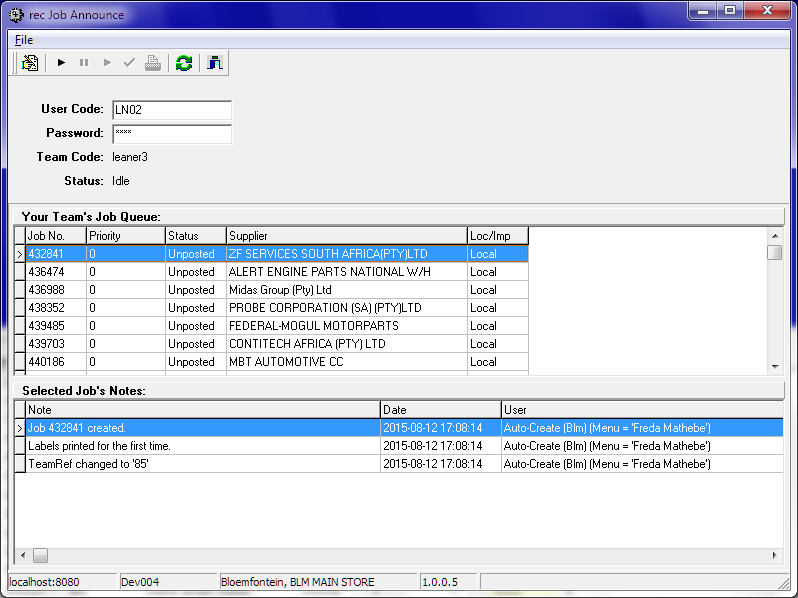
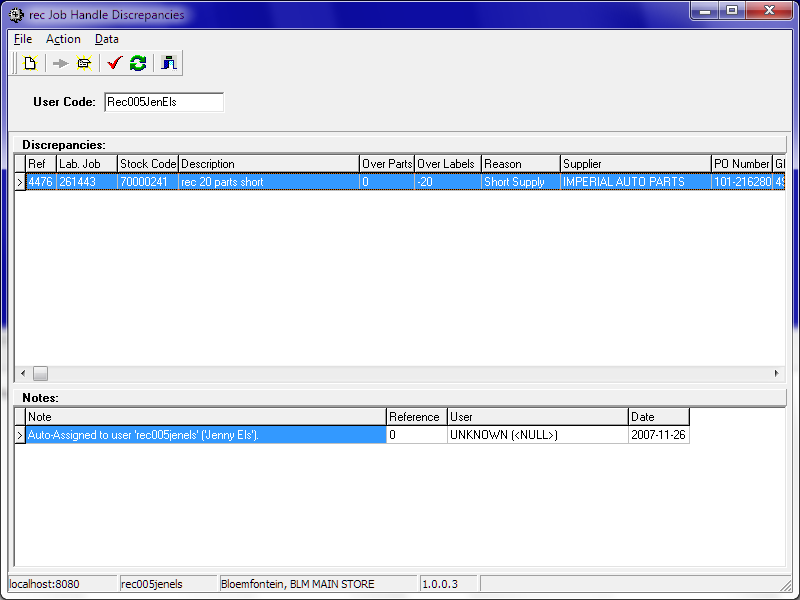
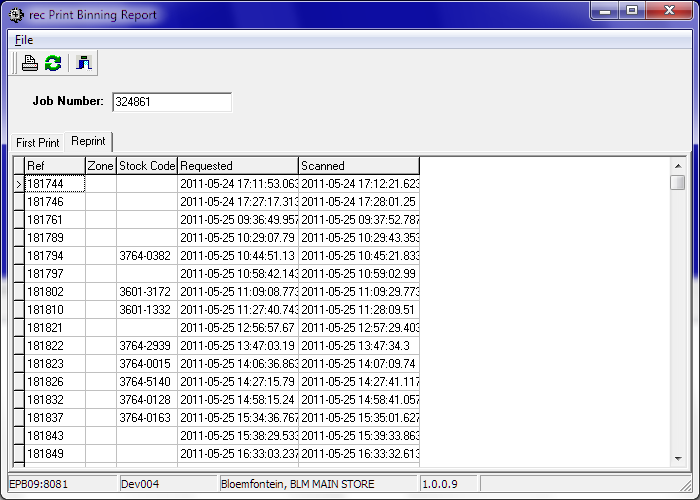
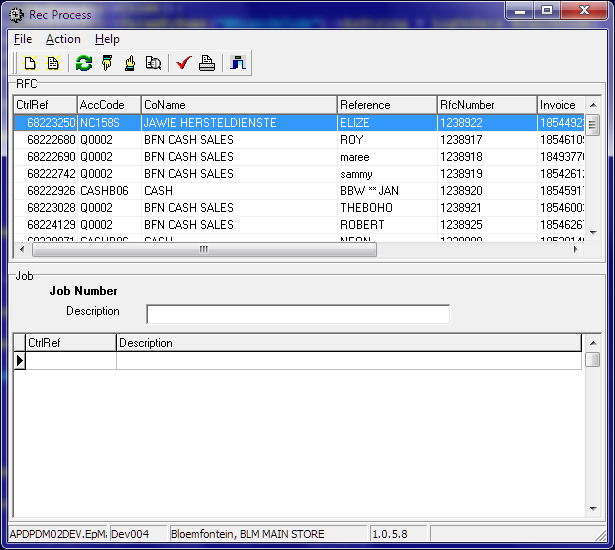
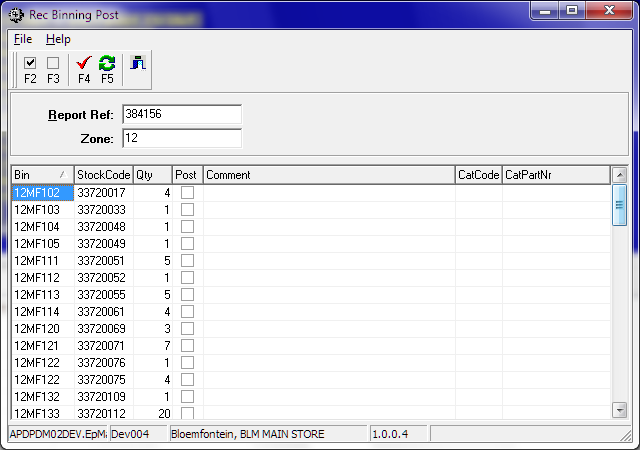
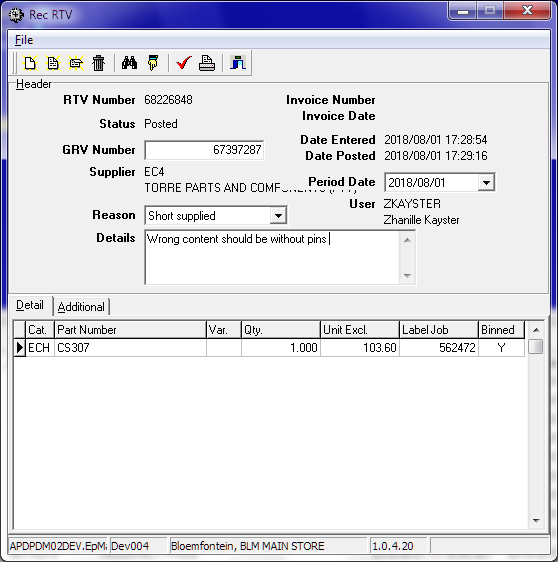
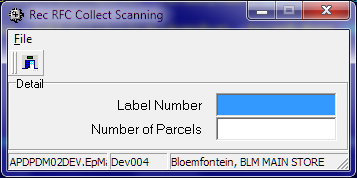
Pricing subsystem

* Introduction:
  + The primary pricing calculation consists of two parts: the mark-up and the discount.
  + Each supplier provides a price list (SPL).
  + The purchase price, in the supplier’s currency, before any modifications, is the IUC.
  + After rebates, freight costs and currency conversion, this is the RUC. (Replacement Unit Cost)
  + The mark-up percentage (a.k.a. automatic retail factor) is added to the RUC of the “preferred supplier” to get the “List Price” (a.k.a. system price or retail price)
  + The discount is dependent on the customer, the super set group, and the stock category.
  + The list price minus the discount is the selling price.
* Frequently used business applications:
  + stkMaintMarkupFlags.exe  
    
    - Central application gathering information from all over the system to aid in pricing and purchase of stock.
    - Explaining everything that this application does will require a separate document.
    - However, it is useful to know that this application shows information of one base item at a time, using working tables that are populated when one chooses to look it up. One then makes changes in those working tables and press the ‘Post’ button to have it all applied to the actual data.
  + stkJrnlQtyAuc.exe  
    
    - The same application as discussed in the warehousing subsystem, except here it is used to change the average unit cost price (AUC) of an item. (The permission code STK\_JRNL is needed for the user to do this along with access to this application.)
    - To do this requires the creation of two stock adjustment journals:
      1. One to remove the stock at the current AUC.
      2. Another to bring the stock back at the new AUC.
    - When there is no stock of the item, but the price still needs to be adjusted, the following is done instead:
      1. Adjust one item into stock at the new AUC.
      2. Remove it again, also at the new AUC.
    - The reason code that is specified determines which general ledger account is contra-affected when the journal is integrated. (The primary GL account affected is stock control, or the base item’s override for stock control if it has one.)
    - The reason codes are stored in [coBranchLocationReason] and the associations between users and reasons in [coLinkUserJrnlReason]
* Frequently used maintenance apps:
  + catMaintainSupplierPartPrice.exe  
    
    - This application can list all of a supplier’s associated parts, so that they can be maintained together.
    - The list on the left controls which columns are displayed on the right.
    - The “Supplier Code” field controls which supplier’s parts are shown in the grid. The “Search” function can be used if the supplier’s name is known, but not their account code.
    - The “Filter” field can be used to narrow the selection by filtering on the catalogue part number.
    - When in the grid, the “Quick Search” can be used to quickly find a value in the currently selected column.
    - The “Set Lead Time” button is a shortcut to change a supplier’s global lead time value, and will cause an update of the lead time on all of that supplier’s parts.
    - The “Export” button can be used to export the whole list as a CSV file, filtered only by the supplier code.
  + catMaintainSuppStockList.exe  
    
    - This application is used to view new price lists from suppliers, compare the prices with the current prices in the system, and apply the new pricing in the system in bulk.
    - The actual data is imported using catOCRWorkbench.exe, but the list must first be created in this application.
  + catMaintainWPL.exe  
    
    - WPL stands for Wholesale Price List
    - A WPL can be assigned to zero or more customers, and then the prices on said list will override the normal selling price calculation.
    - This application is used to create new WPLs, and assign prices to items on them.
    - There is also the ability to automatically maintain a ratio between the WPL’s price and the retail price; very similar to the standard discount procedure, except with finer control. This ratio is called the “automatic WPL factor”.
  + catMaintainAutoRetailFactor.exe  
    
    - The “automatic retail factor” is essentially the mark up applied to the replacement unit cost (RUC) to calculate the list price (a.k.a. retail price).
    - This application reads from and writes to the [catAutoRetailFactor] table directly; without any validation or logic in itself.
    - However, on that table is a trigger to initiate the automatic recalculation of the prices affected.
  + catMaintainAutoWPLFactor.exe  
    
    - The “automatic WPL factor” is used to automatically update the prices on the WPLs when that is preferred to having a constant price.
    - This application reads from and writes to the [catAutoWPLFactor] table directly; without any validation or logic in itself.
    - However, on that table is a trigger to initiate the automatic recalculation of the prices affected.
  + crMaintainCreditors.exe  
    
    - This application was originally used to maintain all the master data on supplier accounts.
    - Since much of this is now maintained in the Sage X3 system and then imported into ePART, the affected fields have been disabled.
    - However, there is still much ERP-related or non-financial data, which therefore does not come from X3, and so needs to be maintained in this application.
    - The tabs labelled “Performance” and “Change Log” are not data entry tabs, but rather allows one to view delivery performance and master data changes, respectively.
* Description of main tables:
  + catSupplierPart
  + drDiscountMatrix
* Description of main stored procedures:
  + catCalcNetPrice
* Description of main triggers:
  + catBaseItemRecalcInsert

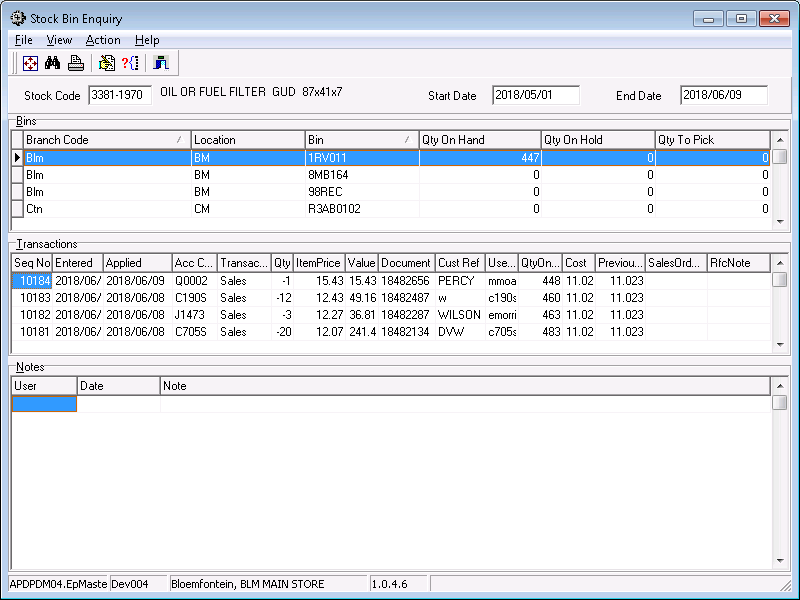
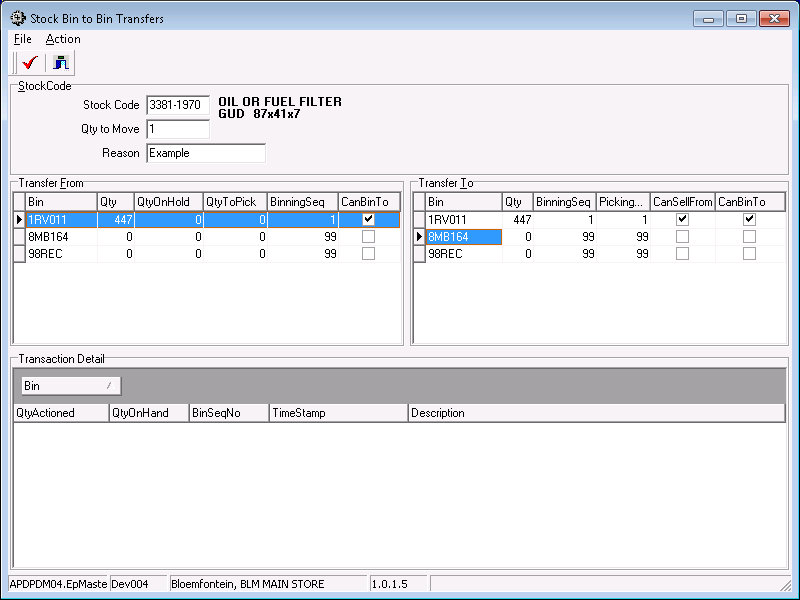
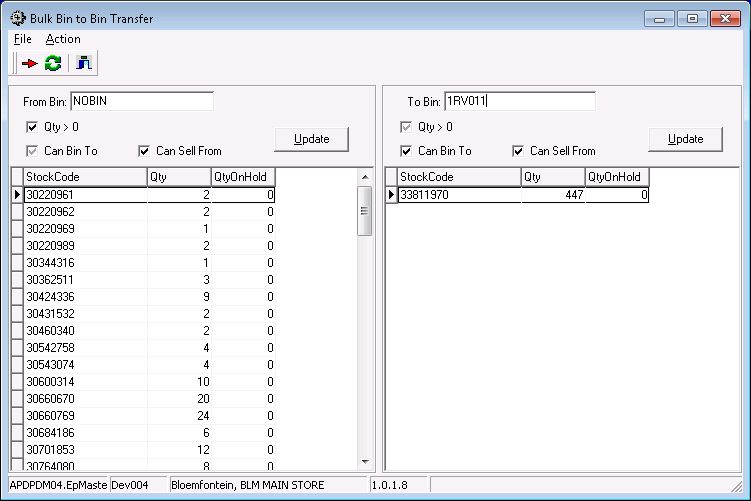
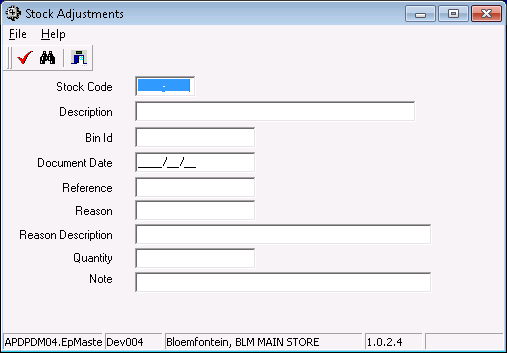
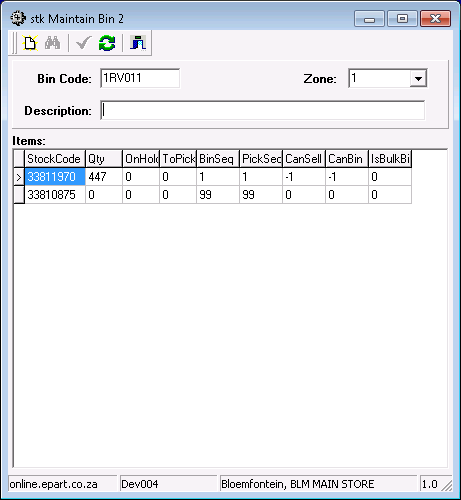
Purchasing subsystem

* Introduction:
  + The purpose of the purchasing subsystem is to keep track of our purchase orders at our various suppliers.
  + It can also send purchase orders to our suppliers via EDI, fax, or e-mail (PDF).
  + The shipping applications can capture details of an incoming shipment and can create the needed GRVs in bulk when it arrives.
* Frequently used business applications:
  + purImportOrders.exe  
    
    - This is used to import the suggested purchase orders that we receive daily from Optimiza.
    - It reads the data from a CSV file with the following mandatory columns:
      1. StockCode
      2. Supplier (or SupplierCode, or AccCode)
      3. Quantity (or Qty)
      4. Reference (or DocRef)
    - It also supports the following optional columns:
      1. Price
      2. CatCode
      3. CatPartNr
      4. Variant (or Size)
    - If the price column is not included, the price will be read from the supplier’s price list.
    - Similarly, leaving out either the catalogue code or the catalogue part number will cause both of these fields and the variant to be read from the supplier’s price list as well.
  + stkPurchaseOrder.exe  
    
    - This application allows one to view, edit, monitor and manage purchase orders.
    - It also shows a log of what changes have been made to the currently selected line, by whom, and when.
  + ~~stkPOShip.exe~~
  + recShipping.exe  
    
    - Large indent orders can arrive in multiple shipments and in multiple containers, with multiple invoices, and, as such, require multiple GRVs and a central location from which to monitor their progress.
    - This application allows one to manage these large orders, and to automatically generate the eventual GRVs.
    - The actual exchange rate can be entered here to override the usual estimate, and will then be recorded onto the GRVs when they are created.
    - Prices can be verified here and the parts with differences accepted or rejected.
  + stkBuyingEnquiry.exe  
    
    - I am unfamiliar with this application, but it looks to be useful for determining whether a part should be purchased, and from whom.
    - However, it is mostly used by receiving users. I guess this is used to look up incoming purchases when the salespeople have queries.
* Frequently used reports:
  + catRptNAAMSAStats.rpt
  + purRptInvoicesByDate.rpt
  + purRptSupplierSalesTargets.rpt
  + purRptOutstandingBackOrders.rpt
  + stkRptPurchaseOrder.rpt
  + stkRptBackPOrderByStockCode\_NewLayout.rpt
  + stkRptPurchaseOrderDayTotal.rpt
  + stkRptPurchaseOrderBackOrder.rpt
* Description of main tables:
* Description of main stored procedures:
* Description of main triggers:

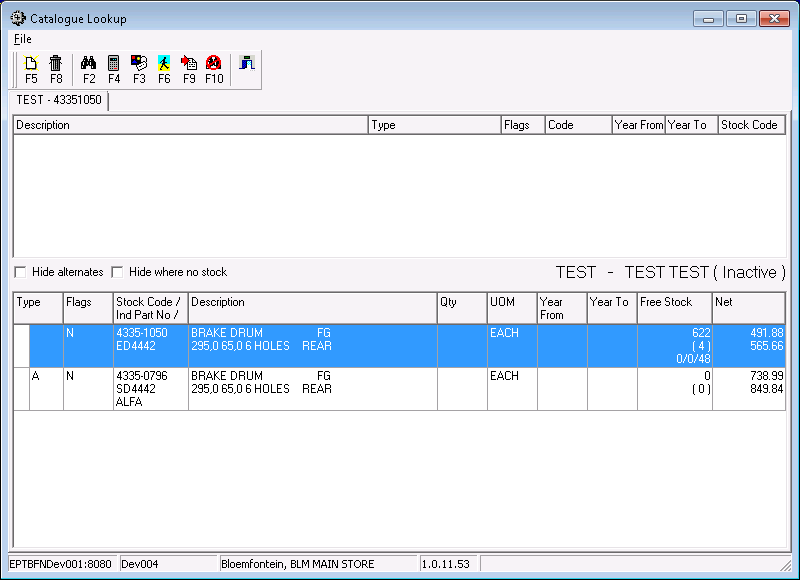
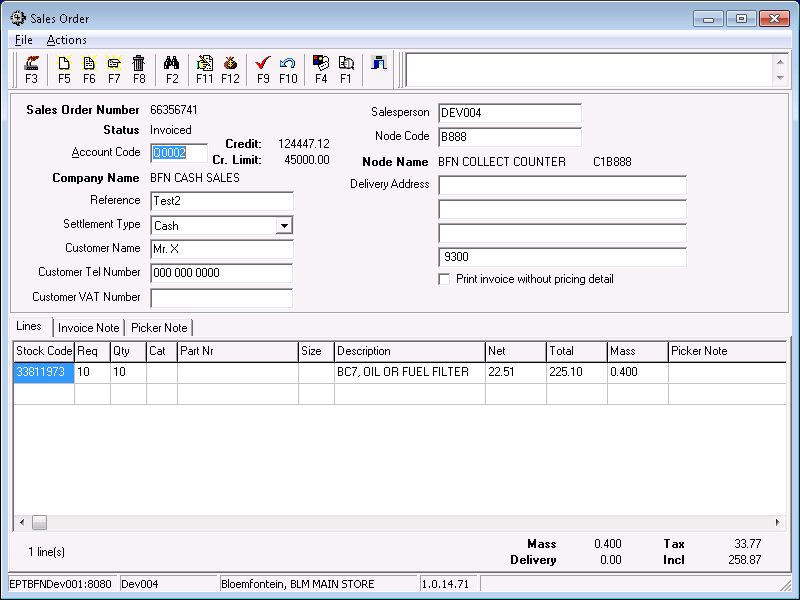
Receiving subsystem

* Introduction:
  + When stock is received, we need to capture the supplier’s invoice, stick labels on the items and put them into the correct stock bins.
  + The GRV application captures the invoice detail, so it must match the invoice exactly, regardless of what stock was received.
  + When the GRV is posted, the supplier invoice is generated in the creditors subsystem.
  + The GRV application can also upload a scanned image of the invoice for later retrieval via the creditors subsystem.
  + After the GRV is posted, the application can print the labels.
  + Depending on the binning process used at the specific branch, printing of the labels will create the necessary labelling jobs and/or binning jobs required.
  + In the case of branches where binning is not done, and also in the case of buyouts at any branch, the binning process is completed immediately behind the scenes.
  + In Bloemfontein, we currently use version 2 of the labelling jobs and version 2 of the binning jobs.
    - When the labels are printed a labelling job is entered into the recJob and recJobLine tables, and an unposted binning transaction is entered into stkTranHdr and stkTranDet.
    - The job’s progress is reported using recJobAnnounce.exe
    - When the labelling job is done and discrepancies have been recorded, the binning report is printed.
    - The binning report’s bar code needs to be scanned when the labelled items pass from the labelling area to the warehouse.
    - When the binning is done the discrepancies are captured and the binning job is released. The unposted binning transaction is updated to match the binning job, discrepancies are sent to the discrepancy bin and these transactions are all posted.
  + In Windhoek, we currently use version 1 of the labelling jobs and version 1 of the binning jobs.
    - Here there is no separate labelling job, only the unposted binning transaction.
    - When the labelling is finished, the binning report is printed manually using a menu option pointing to GenericCrystalReport.exe with parameters indicating recBinningReport2.rpt
    - This report has no bar code and does not need to be scanned.
    - When the binning is done the discrepancies are captured and the binning job is released. The unposted binning transaction is updated to match the binning job, discrepancies are sent to the discrepancy bin and these transactions are all posted.
  + Buyouts can currently only be received in Bloemfontein. The amount of work needed to change this has not yet been determined.
  + When a customer returns stock for credit (RFC), an RFC Collection Order can be created, so that the delivery truck will stop there for the pickup. The invoice for this collect order must be scanned at receiving when the stock is received back.
* Frequently used business applications:
  + ~~stkBuyingEnquiry.exe~~
  + recGrv.exe  
    
    - When stock arrives, it is accompanied by an invoice or a delivery note.
    - This document has to be captured as a Goods Received Voucher (GRV).
    - Therefore, the GRV must match with the documentation received, regardless of what physical items were received.
    - Any such discrepancies will be handled during labelling or binning.
    - The original document should be scanned for record-keeping, and then this application can upload it to the server and link it to the GRV in the database.
    - The values in the “Total Tax” and “Total Inclusive” fields are used to verify that the document has been captured correctly.
    - In the case of mismatches between the document and the purchase order, the lines involved will go on hold.
    - When the document has been captured completely, the GRV can be posted, creating the financial transactions on the supplier’s account and preventing any further changes to the GRV.
    - After the document is posted, labels can be printed for any items that are not on hold and does not already have labels. This creates a labelling job in branches using version 2 of the labelling process.
    - In the case of buy-out items, the labels printed are different, and the parts skip any labelling or binning process. The buy-out system will then automatically create a sales order for the received parts. The picking job will indicate that the pickers should pick up the parts at receiving, by making use of the “buy-out bin”, which is hard-coded to have the bin code: “BUYOUT”.
  + recLabelsAdHocPrint.exe  
    
    - For various reasons it might occasionally be needed to print labels that are not associated with a GRV.
    - After the supervisor has determined that it is safe to do so, they may use this application to do that.
    - Until it is decided to log this, the most reliable way of determining who printed an ad hoc label is to compare the date on the label with the user log for when this application was run. However, this is not reliable at all.
  + recGrvEnq.exe  
    
    - When there are problems with an item on a GRV, the line goes on hold, and this, the GRV Enquiry application, is used to analyse and handle such queries.
    - To determine what is wrong, one looks in the “Flags” column. The meanings of the various flags can be seen by pressing the “Legend” button, but I repeat it here:  
      
  + recGrvView.exe  
    
    - This is the application used for looking up the scanned documents associated with a GRV.
    - It relies on the Envision library of C++Builder components to display, zoom and change the brightness of the images.
  + recJobAnnounce.exe  
    
    - In branches using labelling job version 2, this application is used by the labelling team leaders to manage their team’s activity and performance.
    - From here they can pick a labelling job from the queue, start it, pause or continue it, record discrepancies and then print the binning report.
    - For the purpose of performance measurement, a job’s timer starts when the job is started, stops when it is paused, starts again when it is continued, but only stops finally when the binning report is scanned where they hand the parts over to a binning team.
    - This might seem exploitable, however, we also calculate a team’s idle time.
  + recJobHandleDiscrepancies.exe  
    
    - In branches using labelling job version 2, this application is supposed to be used to record actions taken to resolve a labelling discrepancy.
    - However, there are currently no users set up to use it.
    - The user setup data is stored in the table [recLabelDiscUser]
  + recPrintBinningReport.exe  
    
    - In branches using labelling job version 2, this application shows all the binning reports printed for a specified labelling job, and can be used to reprint them if necessary.
    - It can also be used to print an original binning report if necessary, but normally that would be printed in the labelling job “Announce” application.
    - The “Ref” column displays the binning job number of the binning job associated with each binning report.
  + recProcess.exe  
    
    - Depending of the value of the “-t” parameter specified in the menu option, this application can be used to create three kind of receiving job batches: RFCs, IBTs or GRVs
    - The valid values for said parameter is respectively: “RFC”, “TRANSFER”, and “GRV”
    - This will affect what columns are displayed in the top grid, as well as what data.
    - A labelling job can be created and lines from the top grid assigned onto the job being displayed at the bottom.
    - The purpose is to avoid creating many small jobs; as the label paper is quite expensive, it is good to print a lot of labels on the same page. Up to 27 labels can be printed on each page.
    - For RFCs and IBTs, this is the only way to create the required receiving jobs.
  + recBinningPost.exe  
    
    - Once the binner returns from binning the items, the actual quantities binned need to be entered into this application, along with comments where available, and the binning job posted.
    - This posting process finalizes the binning job and makes the stock available for sale.
    - Any differences between the actual quantities of items binned and the quantities on the binning job is recorded as a binning discrepancy. They remain available to be binned, in case the items are found. To cancel the binning, an RTV must be posted.
  + recRtv.exe  
    
    - When the GRV/Invoice differs from the actual stock received, a decision needs to be made regarding what to do with it.
    - If the stock was missing (short supplied), needs to be sent back, or otherwise should not be paid yet, an RTV needs to be created and posted. If the item has not yet been binned, the binning job and labelling job gets adjusted accordingly.
    - Otherwise, an RTV transaction is entered into the stock sub-ledger, removing stock from the designated RTV bin.
    - If the stock is to be accepted instead, an additional GRV needs to be entered and posted, leading to another labelling and binning job, of course.
    - Both of the above transaction types appear the same in the creditors sub-ledger, except for the sign and the reference to the original document.
    - The reason selected in that drop-down box controls whether or not the purchase order is reduced when the RTV is posted.
  + recRFCCollectScanning.exe  
    
* Frequently used maintenance apps:
  + recMaintainJobs.exe
  + recMaintainUserTeamTable.exe
  + recMaintainLabelJobReason.exe
* Frequently used reports:
  + stkRptGRVsBinningState.rpt
  + recRptRTVPrint.rpt
  + recBinningReport2.rpt
  + recJobExceptionsV3.rpt
  + recRptLabellingTeamTimes.rpt
  + recRptBinnerPerformance.rpt
  + recJobPriorityV3.rpt
  + recGRVEnqHdr.rpt
  + recRptShipBinsRequired.rpt
  + recRptGRVUserProductionPerc.rpt
* Description of main tables:
  + recGRVHdr
  + recGRVDet
  + recRTVHdr
  + recRTVDet
  + recJob
  + recJobLine
* Description of main stored procedures:
* Description of main triggers:

Warehousing subsystem

* Introduction:
  + This subsystem exists mostly to contain the stock-affecting transactions.
  + It also allows one to move stock around, find space for stock and answer queries about stock.
* Frequently used business applications:
  + stkBinEnquiry.exe  
    
    - Note: This application’s source is not in the “Stock” folder, but rather in “Transaction Enquiry”
    - Can show all the transactions affecting one bin, including what effect they had on stock quantity and value.
  + stkBin2Bin.exe  
    
    - Used to create a bin-to-bin transfer transaction.
    - This allows a quantity of items of one type to be moved from one bin to another while guaranteeing that the total quantity and value is not affected.
  + stkBin2BinBulk.exe  
    
    - Allows one to create a batch of bin-to-bin transfers, filtering the stock by various flags.
  + stkJrnlQtyAuc.exe  
    
    - This is the same application used in Pricing to adjust cost prices, but here it is used to record lost stock (e.g. picking failure), found stock (e.g. stock check surplus), damaged stock etc.
    - The first difference is that these users do not have the permission code STK\_JRNL, and so the panel with the AUC entry field is hidden.
    - The second difference is in the reason codes that the users are allowed to specify.
    - The link between users and reason codes is stored in the table called [coLinkUserJrnlReason]
* Frequently used maintenance apps:
  + StkMaintainBin2.exe  
    
    - The purpose of this application is to create new bins.
    - However, that happens very rarely.
* Frequently used reports:
  + stkRptBinTransactions.rpt
  + stkRptWarehouseByStockRange.rpt
  + stkRptUnsellableItems.rpt
  + stkRptWarehouseByBinRange.rpt
  + stkRptNoSellFromBinsWithStk.rpt
  + stkRptMultiBinStockCodes.rpt
* Description of main tables:
* Description of main stored procedures:
* Description of main triggers:

Sales subsystem

* Introduction:
  + Creating a sales order involves looking up the right part for the customer’s purpose and selling it to them for the right price.
  + The applications catLookup2.exe and salOrder.exe are made to work together:
    - In catLookup2.exe you can add a part to an existing sales order or to a new sales order.
    - When you then switch focus to salOrder.exe, it will automatically show the latest sales order whereto you added an item.
  + Whether a customer wants a specific part to put in a specific vehicle, or are buying stock for their shop using a catalogue, in either case or any other case catLookup2.exe will help one to identify the correct part.
  + In salOrder.exe you can see the total selling price of the order, and some users can do price overrides to further entice the customer.
  + To record lost sales, there are separate columns for the requested quantity and for the quantity being sold.
  + Unfortunately, most of the lost sales reports are still using the lost sales based on failed search queries. (Stored in stkItem). It may be a good idea to simply replace that calculation, rather than changing the reports.
  + When the sales order is ready, it is released (a.k.a. posted). It may also be released automatically if it has been open (unreleased) for longer than two hours, or if it has more than 10 line items.
  + During release:
    - If the order has more than 10 line items, it is split into multiple orders. This is logged in the table salTranLogSplit.
    - If delivery charges are enabled and the customer buys less than 20k per month, and the route or node qualifies, a delivery charge is added. (This is not currently enabled.)
* Frequently used business applications:
  + catLookup2.exe (added the picture)  
    
    - Used to look up the correct parts and add them to a sales order.
  + salOrder.exe  
    
    - Allows the sales person to review the order, and change the delivery address.
    - They can also adjust or add items to the order from here.
    - Allows authorised users to adjust the order’s selling prices.
    - When the order is ready, they are encouraged to release it using this application, rather than waiting for it to eventually be released automatically.
* Frequently used maintenance apps:
  + salMaintainUserSSGDiscounts.exe
* Frequently used reports:
  + stkRptQuote.rpt
  + stkRptSalespersonNetTurnover.rpt
  + stkRptBranchLocTurnover.rpt
  + stkRptBranchLocLogicalTurnover.rpt
  + stkRptSalesByTown6month.rpt
  + stkRptBranchLocCashTurnover.rpt
  + salRptBuyoutsPerDayBranch.rpt
  + salRptExternalTransfersWeekly.rpt
  + salRptTurnoverByAccountBusinessType.rpt
  + salRptGroupCodeSalesHistoryByAccount.rpt
  + salRptSellingPricesForAcc.rpt
  + salRptTurnoverByBranchBusinessType.rpt
  + salRptCustomerGroupCodeSales.rpt
  + salRptGroupCodeSalesHistoryByStockCode.rpt
  + salRptSalesmanSalesHistory.rpt
* Description of main tables:
* Description of main stored procedures:
* Description of main triggers:

Picking subsystem

* Introduction:
  + After a sales order has been released, the items need to be picked from stock, packed into parcels, and invoices printed. During this process the items are still in stock, but they are on hold.
  + Using pckOrders.exe, the picker requests a picking job, and a picking slip is printed. The system automatically calculates which sales orders to add to the picking job according to which route group the picker is assigned to work on, and possibly other criteria.
  + When the picker has retrieved the items, they use pckOrders.exe to print the packing slip. This allows them to separate the parts by sales order before packing.
  + When the packing is completed pckOrders.exe is also used to print the invoices. The items are now removed from stock.
* Frequently used business applications:
  + pckControl.exe
    - Used by the supervisor to adjust picking jobs in the case of unexpected stock-outs, or if the customer changes their mind, or whatever.
    - Also used to monitor the pickers’ performance, and to put a job on hold when the pickers are on break.
  + pckOrders.exe
    - Used by the pickers to request picking jobs, print picking slips and packing slips, and finally the invoices.
  + pckReprintInvoice.exe
  + pckReprintOriginalInvoice.exe
    - If the printer fails during invoice printing, the supervisor may need to create a new original invoice using this application.
* Frequently used maintenance apps:
  + pckMaintPickers.exe
    - Used to create new pickers, to change pickers’ passwords and to assign pickers to route groups.
  + pckMaintRouteGroups.exe
    - Used to allocate despatch routes to picking’s route groups.
* Frequently used reports:
  + pckRptInvoiceLaser.rpt
  + pckGetInvoicePrint.rpt
  + pckRptPickerStats.rpt
  + pckRptReprintOriginalInvoice.rpt
  + pckRptAdjustments.rpt
  + pckRptIncompleteOrders.rpt
  + pckRptInvoiceLaser\_tmpDebtorsReprintMnthEnd.rpt
* Description of main tables:
* Description of main stored procedures:
* Description of main triggers:

Despatch subsystem

* Introduction:
  + When the items are in parcels and the invoices are printed, it is time to label the parcels and load them into the delivery vehicles.
  + To print a label using dsCreateLabels.exe, the invoice’s bar code is scanned, and a number is typed to indicate how many parcels need labels for this one invoice. The invoice is then placed inside one of the boxes, they are all sealed, and the labels are stuck on them.
  + The despatch area is divided into lanes. Each lane leads to one cage, called a loading bay. In the system the lane and the bay are one object, stored in dsLoadBay.
  + Each delivery address has a node code, which represents a physical place where the parcels will be delivered.
  + Each node is assigned to one or more routes, and each route to one or more load groups.
  + Each load group departs at a specific time and from a specific bay. The parcels are assigned to the next load group to depart containing an active route containing the delivery node.
  + When the parcels arrive at the correct bay, the cage’s inner door is open and each parcel is scanned as it is put into the cage. (dsOutScan2.exe)
  + When all the parcels for the delivery is accounted for, the inner door is closed, the outer door is opened and each parcel is scanned as it is loaded into the vehicle.
  + When this is done, dsLoadGroupDepart.exe is used to print all the delivery documentation:
    - Proof of Delivery documents (P.o.D.)
    - Trip sheet
    - Cover sheet
    - C.o.D. Reconciliation
    - Manifest
  + When the delivery vehicle returns, the PoDs are scanned and processed, and the scanned images are stored on a server (currently EPTBK02) for later recall.
* Frequently used business applications:
  + dsViewPODs.exe
  + dsLoadGroupDepart.exe
  + dsReprint.exe
  + dsOutScan2.exe
  + dsCreateLabels\_jhb.exe
  + dsCreateLabels.exe
  + dsDeleteLabel.exe
  + dsPrintNamInvoice.exe
  + dsAlterDeliveryNodeCode.exe
* Frequently used maintenance apps:
  + dsMaintainLoadGroups.exe
  + dsMaintainRoutes.exe
  + dsMaintainNodes.exe
* Frequently used reports:
  + dsRptInvnoParcels.rpt
  + dsPODs.rpt
  + dsTripSheet.rpt
  + dsRptRouteCover.rpt
  + dsRptRouteManifest.rpt
  + dsRptAccCodeParcels.rpt
  + dsRptCustDelAddrByTown.rpt
  + dsRptParcelToInvoices.rpt
  + dsRptCoverSheetScanByDate.rpt
  + dsReportNodeRoute.rpt
  + dsRptRouteParcelCnt.rpt
  + dsRptRouteStatistics.rpt
  + dsRptRouteCustomerProfit.rpt
  + dsRptParcelDeleteList.rpt
  + dsRptRouteAvgDelTime.rpt
  + dsRptRouteProfit.rpt
  + dsRptBranchParcelCnt.rpt
  + dsRptCODRecon.rpt
  + dsRptPODsArriveLate.rpt
  + dsRptListViaParcels.rpt
  + dsRptPODsReturnLate.rpt
* Description of main tables:
* Description of main stored procedures:
* Description of main triggers:

Buyouts subsystem

* Introduction:
  + When a customer requires an item that we do not stock, or an item that is currently out of stock, a sales person has the option to create a buyout.
  + The buyout process has the following steps:
    1. Capturing
    2. Automated Purchase Order(s)
    3. Receiving
    4. Automated Sales Order
    5. Picking
  + After picking, it is handled as an ordinary invoice, except that all the items on the invoice have stock code 9540-0000.
  + The buyout process cannot be cancelled properly at any point before invoicing.
  + Also, the buyout items’ GRVs cannot be RTVed, and the sales invoice cannot be RFCed.
  + If a buyout needs to be reversed, this is currently handled through the claims subsystem.
* Frequently used business applications:
  + stkBuyout.exe
  + stkDocumentLookup.exe
  + stkBuyoutOutstanding.exe
* Frequently used maintenance apps:
* Frequently used reports:
  + salRptBuyoutsPerDayBranch.rpt
  + manRptStkBuyOuts.rpt
  + stkRptBuyoutsOutstanding.rpt
  + stkRptBuyOuts.rpt
* Description of main tables:
* Description of main stored procedures:
* Description of main triggers:

Stock take subsystem

* Introduction:
* Frequently used business applications:
  + sttCountSheets.exe
  + sttWallToWallCapture.exe
  + sttWallToWallAdjust.exe
  + sttAdHocCounting.exe
* Frequently used maintenance apps:
* Frequently used reports:
  + sttRptCountSheetWallToWall.rpt
  + sttDiscrepencyReport.rpt
  + sttRptCountSheetAutoMail.rpt
  + sttRptAdjustmentLog.rpt
  + sttRptW2WAdjustmentList.rpt
  + sttRptWallToWallProgress.rpt
  + sttRptCountHistoryForPart.rpt
  + sttRptDayCountersNotCaptured.rpt
  + sttRptDiscrepancyWallToWall.rpt
  + sttRptWallToWallControlSheet.rpt
* Description of main tables:
* Description of main stored procedures:

Company subsystem

* Introduction:
  + This subsystem is used to provide data that is shared by all subsystems, like:
    - Users
    - Menu structures
    - Permissions and roles
    - Branches
    - Countries and currencies
    - Public holidays
    - Tax rules
  + It also provides a centralized place to store programming structures:
    - Global constants
    - Unique identifiers
    - Invoice numbers
    - RFC numbers
    - All other document reference numbers
* Frequently used maintenance apps:
  + coMaintainUser.exe
  + coMaintainCurrency.exe
  + coMaintainTariffElement.exe
  + coMaintainEmployees.exe
* Frequently used reports:
  + coRptUserMenu.rpt
* Description of main tables:
  + coUser
    - User names, logins, e-mail addresses and other silent parameters.
  + coPermission
    - Definitions of the various permission codes, and descriptions as to their purpose.
    - The actual implementation of the permissions falls to the relevant stored procedures and applications.
  + coUserPermission
    - Linking table to assign permissions to users.
  + coCompany
    - A list of companies that are using the current instance of ePART.
    - Used primarily for separating the companies’ accounts and ledgers.
  + coBranch
    - A list of branches per company and their addresses and parameters.
  + coCountry
    - A list of countries’ names, short hands and links to currencies.
  + coCurrency
    - A list of currencies, short hands and their estimated exchange rates (to use on GRVs.)
  + coPublicHoliday
    - A list of public holidays, used during users’ performance measurements to exclude days where they could not be expected to work.
    - Updated annually by a database job, except for the Easter weekend, which has been filled in manually up to 2034.
  + coUserLog
    - A log of users logging in, running applications or changing their passwords.
  + coPasswordLog
    - A log of users’ previous passwords to prevent too frequent re-use, and to determine when to force them to change it.
    - Also sometimes useful for verifying a person’s identity.
  + coPerformanceLog
    - A log of users running Crystal reports and how long they took to run.
  + coCtrl
    - Most of this table is being phased out
    - It might still contain a few useful global constants
    - The primary use that will remain is for generated new CtrlRefs.
  + coDocRef
    - Was created to provide sequential invoice and RFC numbers
    - Not currently used in EPT-ZA
    - The procedure that generates these numbers needs to be improved; it sometimes generates the same number more than once due to concurrency.
  + coSequence
    - This table is used for all other unique sequential numbers.
* Description of main stored procedures:
* Description of main triggers:

General subsystem

* Introduction:
  + This subsystem is a catch-all for all applications that would not fit in any of the subsystems, while not being complex enough to warrant their own subsystems.
  + It also holds all the Generic applications, which typically have a stored procedure name as a parameter and builds their own user interface in accordance with how the procedure will be used.
* Frequently used business applications:
  + epMail.exe
  + epMenu.exe
  + epLoad.exe
  + GenericCrystalReport.exe
  + GenericExcelReport.exe
  + GenCSVExpInt.exe
  + GenCSVImpInt.exe
  + GenericLookup.exe
* Frequently used maintenance apps:
* Description of main tables:
* Description of main stored procedures:

Management reports subsystem

* Introduction:
  + This subsystem consists of a large number of sensitive reports
  + Also some reports that should not be run often as they are very heavy on database usage
  + And a few maintenance applications for data that are used by various reports.
* Frequently used maintenance apps:
  + manMaintainSalPerf.exe
  + manMaintInvoiceMessage.exe
  + manMaintainSuperGroups.exe
* Frequently used reports:
  + manRptSalPerf.rpt
  + manRptRepsCallingV2.rpt
  + manRptListCustomersByTownOrNode.rpt
  + manRptTurnoverPerDebtorWPLSuperSet.rpt
  + manRptCustomerSalesHistoryTopYXMnths.rpt
  + manRptCustomerTurnoverBySalesman.rpt
  + manRptTurnoverPartitionbyAccSSG.rpt
  + manRptOnlineCustomer3MonthSalesV2.rpt
  + manRptTurnoverPerDebtorSSGYears.rpt
  + manRptStkBuyOuts.rpt
  + manRptNewDebtorsSalesXMnths.rpt
  + manRptSalPerfOnlineDet.rpt
  + manRptSalPerfSuperCode.rpt
  + manRptSalPerfOnlineDetByTown.rpt
  + manRptSalPerfCounterDailyRecon.rpt
  + manRptPivStockCodePeriodPurchasesBySupplier.rpt
  + manRptChargeDeliveriesByAccount.rpt
  + manRptSalesOrderActivityTimes.rpt
  + manRptOnlineCustomer3MonthSalesV3.rpt
  + manRptPivGroupCodePeriodSales.rpt
  + manRptPivGroupCodePeriodSalesSummary.rpt
  + manRptPivStockCodePeriodPurchases.rpt
  + manRptStkTrendExceptionsSummary.rpt
* Description of main tables:
* Description of main stored procedures:

Incident logging subsystem

* Introduction:
  + This subsystem was originally meant to facilitate communication between the departments through the logging of various incidents, providing messaging and setting up reminders.
  + Currently, only the debtors department are using it for its original purpose, in the form of debtors’ notes.
  + This subsystem is found in a separate database on the same server: EPIncidentLog
  + Because of this, anyone who wants to run the Debtors Maintenance application also needs access to EPIncidentLog
* Description of main tables:
  + incLog
* Description of main stored procedures: