

**Version 1.1**

**Date: 2018/11/15**

Abstract

**Describes the post sales order release process to pick goods ordered by customers in an efficient manner suitable to the Engineparts business and operational cycle.**

**The original solution was designed and implemented as a BSc Honors project that scored well**

Optimised Business Aligned Picking sub-system

**Table of Contents**

Document approval and distribution list 2

1. Introduction 3

2. Audience 3

3. Dependencies 4

4. Objectives 4

5. Business Rules 4

6. Detail description of functionality 5

7. Picking process 6

7.1 Dashboard 6

7.2 Creating a picker job 7

8. Future considerations: 10

9. Database diagram 11

9.1 SQL Stored Procedures 11

10. Programs 15

10.1 MS Windows Executables 15

11. Acceptance 16

# Document approval and distribution list

|  |  |  |  |
| --- | --- | --- | --- |
| **Description** | **Name / Title** | **Signature** | **Date** |
| **Document Type / purpose** | | | |
| Prepared by |  |  |  |
| Reviewed by |  |  |  |
| Approved by |  |  |  |

# Introduction

The picking process aims at extracting stock from a specific warehouse based on instructions issued via a confirmed sales order.

To perform the function in an optimised manner that recognises the following elements:

* There are defined delivery route departure times. The picker workforce needs to be modulated to ensure that all queued picking orders are completed to meet these route departure times.
* In some instances, there are multiple departure times per day and for local deliveries maybe even every 30 minutes.
* Picker resources can be allocated to one or more delivery routes as a means of optimising the workforce.
* Picking orders can be grouped into picking jobs and allocated to individual picker staff. This gains better traction on the picker workload.

The general rule is to allow more picking orders per picking job where the route departure time is less demanding but to reduce the number of picking order per job and to allocate more resources per imminently departing route.

* To assist the picking dashboard view, picking orders are grouped by route and colour coded according to the departure time of the route.
  + Red implies critical – departure time is very imminent
  + Orange raise the picking urgency
  + Lower urgency
  + No colour no urgency

In some instances, there may be reason for the picking of goods that do not form part of the sales process i.e. where goods are to be returned to suppliers or goods are to be destroyed due to expiry or faulty. For these instances the ***Returns to Vendor*** (RTV) process manages the stock holding and isolation followed by the generating of a ***Sales Order*** with ***very specific*** picker instructions. This is described in greater detail in the ***Returns to Vendor (RTV)*** part of the documentation.

# Audience

Receiving

Warehouse

Despatch

Suppliers

# Dependencies

The existence of the customer order fulfilment (picking) is dependent on sales and the sales order process

To ensure that customer order fulfilment can effectively be conducted, there is a dependency on proper warehouse management hence warehouse management

Once the picking part of the customer order fulfilment is done, there is a dependency on the despatching and delivery system

Once the picking has concluded, there is a dependency on the sub-ledger accounts receivable for the shipped goods to be invoiced against

# Objectives

The objective of this sub system is to ensure that ordered goods (Sales Ordering) are accurately picked following an pre-defined priority with dynamic optimisation of sales order consolidation to picking jobs. Concurrently, there are option to dynamically adjust workforce allocations and workforce workload to optimise production to meet with distribution ***time to depart*** schedules.

The picking sub-system design and 1st implementation formed part of a BSc student development program where the best of Honours students received parcelled projects for their dissertation

# Business Rules

When a sale order is created, no stock is placed on reserve, only recorded as a potential demand. Applying the 1st come 1st serve principal.

At the time of releasing a sales order, the sales order defined stock requirements are placed on reserve such that the stock is not available for sale anymore.

At this time a sales order translates into a picking order status and the sales order can be viewed but not changed anymore.

Should there be a reason for it, the picking order can be cancelled, and the sales order consequently is also cancelled.

Should there not be enough stock on hand to fulfil the sales order defined requirement, the quantity picked becomes the ruling quantity for later invoice generation.

It is a business requirement and so implemented that in some instances stocking items may only be sold in sets. This rule is enforced at sales order time and again as part of the picking process in the following manner:

* 4 items required but only 3 in stock and the stocking item has no set restriction, the 3 items are confirmed as picked.
  + For such events, the rule is that the picker is required to refer the shortage to the relevant storeman for immediate investigation.
  + Should the storeman find the missing item(s), the picker can complete the order
  + During the investigation period the picking order is placed on hold with the relevant reason code for management review of warehouse efficiencies.
  + Should the storeman not find the required items, the picker picked quantity remains. However, the non-conformance is injected into the daily cycle count process as an exception to be investigated
* 3 items requested a set rules of 4 or nothing is in place then the sales person will not be allowed to order the 3 items and the quantity will be zero. By way of an explanation is that some exotic vehicles pistons are always supplied and consumed in sets. Breaking a set of pistons may results in the rest of the set becoming unsaleable.

Although this is a sales order functional requirement, it is raised here for the reader to understand the underlying rules and reasons why

* Where items on hand are found to be non-conformant, this is referred to the storeman for investigation with immediate effect.
  + The picking order is placed on hold
  + The store-man conducts the investigation and if the goods are non-conformant, a bin to bin transfer for the non-forming goods is processed to a non-selling bin and the storeman then physically moves the goods accordingly
  + When opportune, the position is investigated in line with documentation under ***warehouse management.***
  + The non-conforming items event injects a stock check for the next daily cycle count where a more detailed investigation is launched

# Detail description of functionality

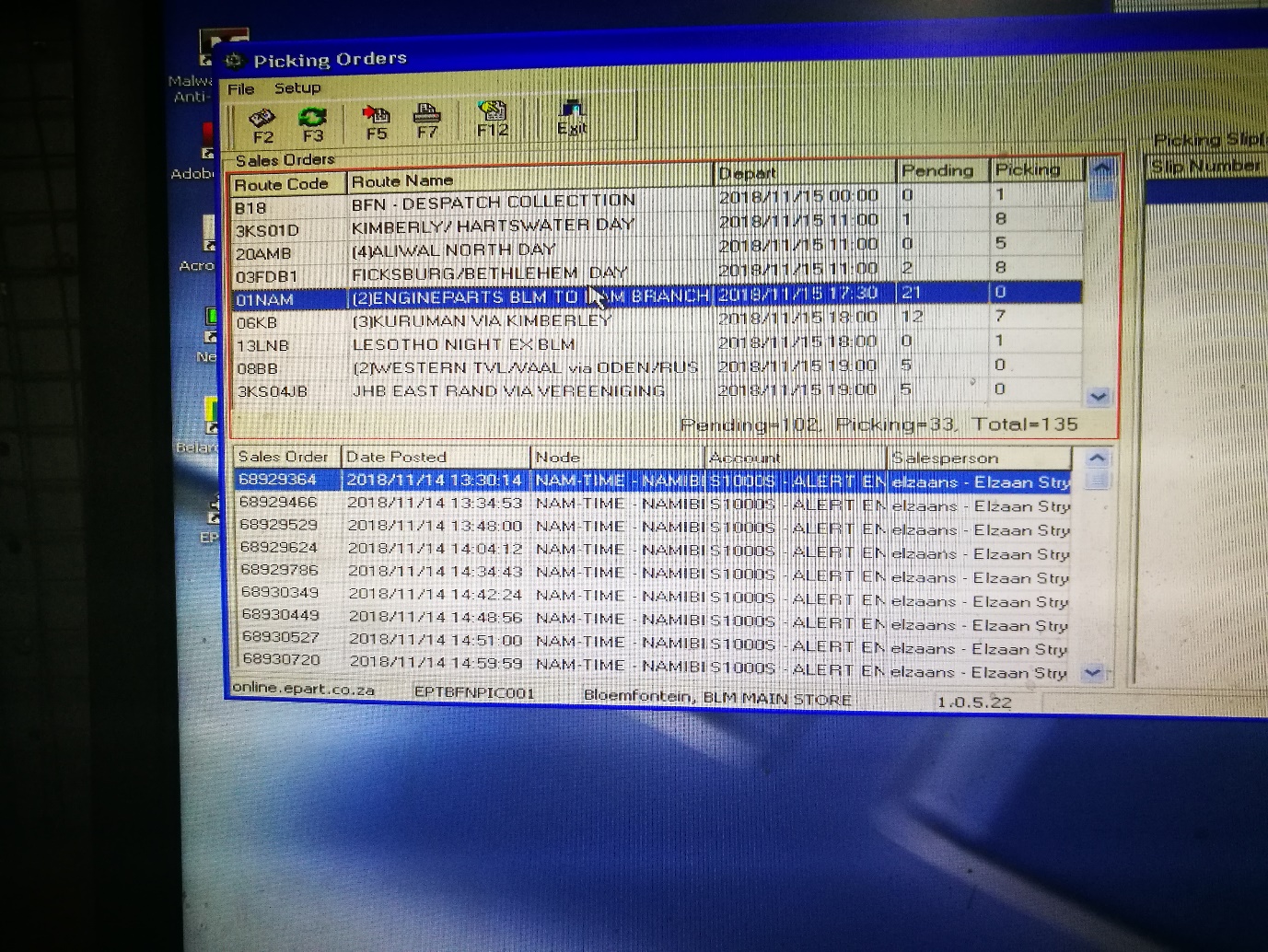
* 1. Customer sales order would have to have received a valid delivery node allocation at the time of sales order creation / before releasing the sales order.
  2. To note these node codes are verified against a route to which it is allocated
  3. The sub system auto calculates the correct route allocation based of best time to departure, in the event of the node code being allocated to multiple route codes.
  4. It is possible for the sales staff to alter the delivery node code, subject to approval and permission.
  5. It is possible to pre-allocate any number of delivery nodes to a customer from which sales stall and self-help customers can select the appropriate option
  6. Restricting delivery node overrides is part of the solution due to illegally using on-line customer to place order for delivery to the criminal’s address
  7. It is a management responsible to ensure that routes and related node codes form an efficient / permissible combination
  8. These node / route combinations are agreed to with one or more delivery service providers, including own fleet facilities.
  9. ***Customer to collect*** is a valid route and node combination that depicts the collection counter
  10. Picking sub system. This sub system is delivery node and route allocation aware and provides the following feedback to related users:
  11. Picking jobs within a parameter defined time to complete is displayed without any emphasis
  12. Picking jobs within a parameter defined time period to route departure not completed are displayed in orange raising the system urgency to complete to meet with the planned departure time
  13. As soon as a route departure time has almost been reached (parameter defined) these are displayed in red

# Picking process

### Dashboard

Orders available for picking is displayed on a display screen dashboard and is updated and refreshed in real-time without requiring human input

The following image provides a in-production view of picking orders to be completed

**Image 7.1 Picking orders in queue by route and per route the related sales orders**

The display screen allows for real-time changing of views by clicking on the column heading it re-sorts the content. To view by earlier / latest departing time sorted a mouse click on ***Depart*** will action the re-sort.

Using this functionality, the user is able to schedule the workforce by volume of work and time to complete.

To note are the ***Pending and Picking*** columns.

***Pending*** represents sales orders not in pickers hands

***Picking*** represents picking orders work in progress

### Creating a picker job

For accountability and performance measurement, a picker is required to announce him- or herself by keying in a 2-letter code. The reason for the short code is efficiency as pickers are time constrained.

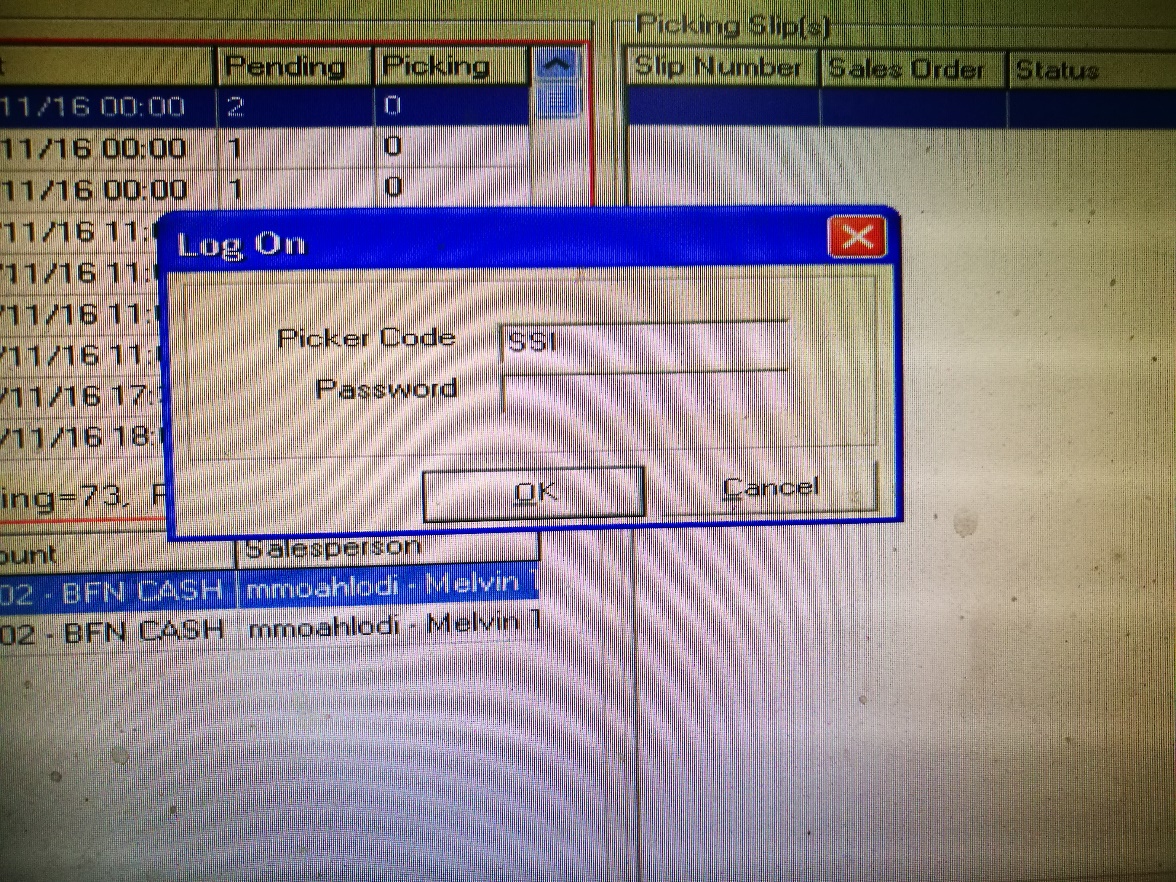
The notion of security does not play a role as the picker resource is motivated by rewards based on performance. Conversely not achieving minimum standards of performance may eventually lead to dismissal. However, there is provision for the password but is not used.

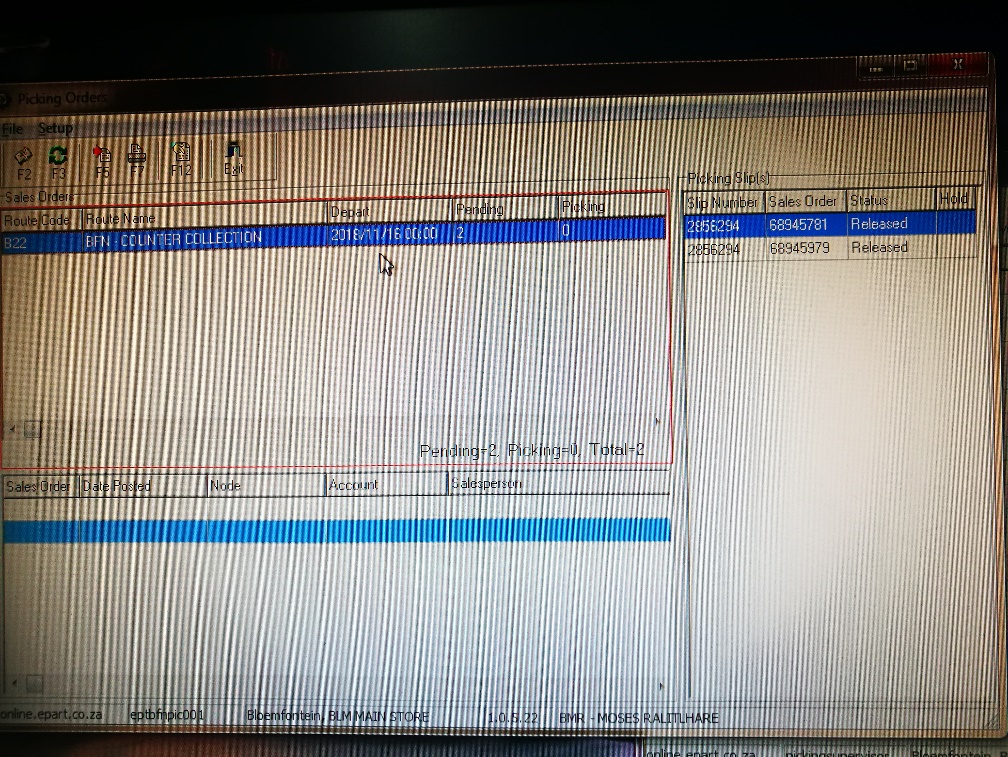
The picker is then allowed to select 1 or more sales orders from the list of available routes. To note, the picker sees a limited set of routes once signed in as determined by the supervisor.

The maximum number of sales orders that a picker can select is determined by the supervisor

The route allocation per picker and the maximum number of orders per picker is set by the supervisor in real time to accommodate the specific workload to ensure that pending routes depart on time

Pickers are measure by items picked and often would rather select order with many line items which is fine unless the route departure is imminent. The workforce is trained to look at the size of the orders being viewed on the dashboard and to make a call which to select against route departure.

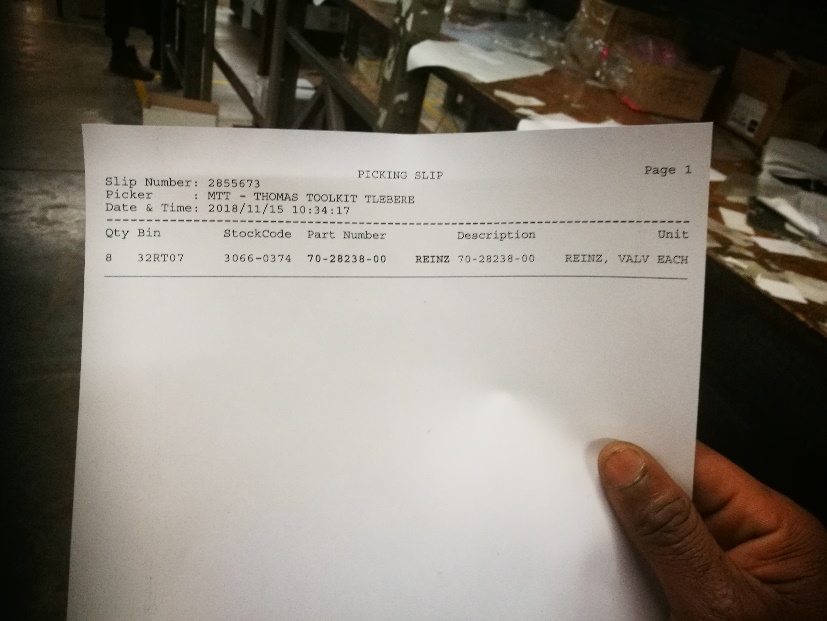
**Image 7.2.1 provides a view of the picker sign-in screen**

**Image 7.2.2 provides a view of a picking job being assembled**

Note the right-hand side with heading ***Picking Slips*** and inside the list are 2 picking slips allocated to the picker as a picking job

As soon as the picker has completed his / her selection, function key ***F7*** updates the database with the selection and immediately prints a picking slip as depicted in the following image

**Image 7.2.2 provides a view of a picking slip**

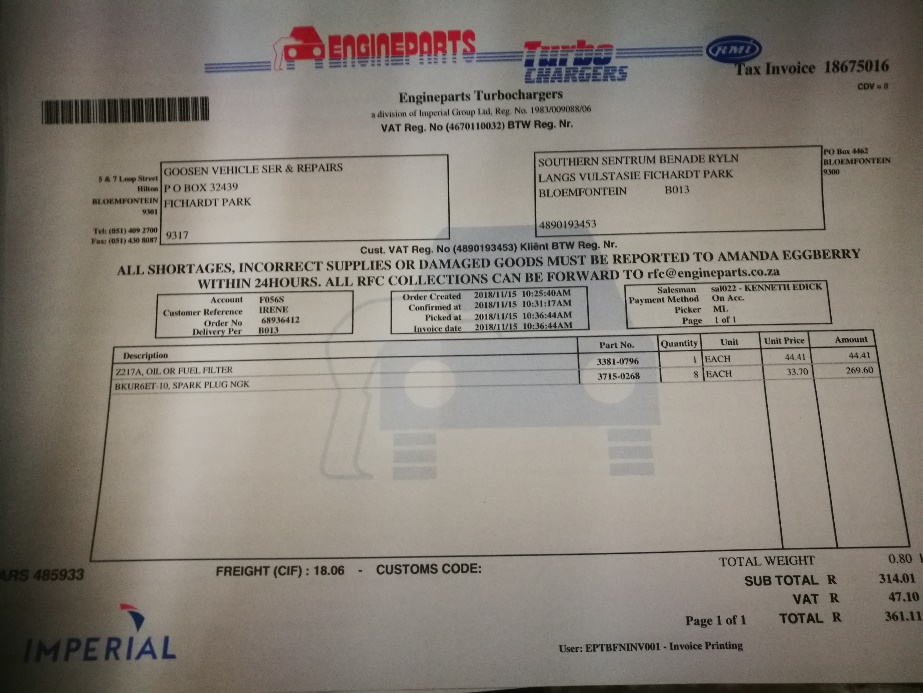


As son as the picker has completed the picking task, a packing slip is required to collate the items per sales order.

This procedure has largely been changed in favour of not printing the packing slip and rather printing the invoice only to save on paper, printing equipment and consumables.

Any errors post invoice is corrected through transactional entries i.e. RFC. The invoice is annotated and signed indicating the error.

The incidents of errors is so small that the practice has been an acceptable change by customers. However, the omission of the packing slip it still remains a negative change when measured against best practice.

**Image 7.2.3 The following image provide a view of the invoice as printed**

To note is ONLY 1 ***Tax Invoice*** with the same reference will be printed. Any reprints will reflect ***Copy Tax Invoice*** in line with the legislated SARS stipulations.

# Future considerations:

Although, at the time, the functionality was advanced for its time, no attention to optimisation has been given to part of the system. As taken up in other document sets pertaining to deploying mobility, it would be a small step forward to enable the workforce with mobile scanners in place of paper.

The supervisor dashboard can be enhanced to pre-allocate work to each mobile device and as a picker completes the picking process the mobile device can be auto loaded via WiFi with the next picking job.

Some statistical detail could be made available to the supervisor to ensure that the work is equitably shared and that the departure times are met with as expected.

The omission of printing the packing slip currently can be re-instated using a mobile device scanning and verification process, pre-invoice print. The incidents of having to pass credits to compensate for human errors will be much reduced with improved customer experience.

To further enhance the process and saving on consumables, consideration can be given to supress printing of invoices in favour of backed systems performing an electronic transmission rather. This is standard practice with all the on-line portal organisations today.

# Database diagram

# SQL Stored Procedures

|  |  |
| --- | --- |
| **Name** | **Description** |
|  |  |
| pckPickingJobHomeFindBySalesOrderCtrlRef |  |
| pckGetInvoicePrint |  |
| pckPickerReport |  |
| pckPickingJobRefresh |  |
| pckSortOrders |  |
| PckJobHdrValidateInsert |  |
| pckCheckOrderInvoice |  |
| pckPickingOrderFieldChange |  |
| pckCreateInvoice |  |
| pckGetAllJobHdr |  |
| pckGoingLiveSetup |  |
| pckSortBusyOrders |  |
| pckPickingOrderLineHomeFindByCtrlRef |  |
| pckPickingOrderHomeFindByPickerRoutes |  |
| pckAddAllRoutes |  |
| pckRptPickerStats |  |
| pckPickingOrderLineRefresh |  |
| pckPickingOrderHomeFindByPickerCode |  |
| pckAutoCreateJob |  |
| pckAdjustJob |  |
| pckPickingOrderHomeFindByCtrlRef |  |
| pckRecalcRoute |  |
| pckPickingOrderHomeFindByBranchLoc |  |
| pckGetInvoicePrint\_1 |  |
| PckRouteSalesOrderRefresh |  |
| pckFakeSo |  |
| pckPickingOrderRemoveFromPickingSlip |  |
| pckPickingOrderHomeFindByPickingSlipNumber |  |
| PckControlSearchPickingSlipNumber |  |
| pckGetDeliveryNote |  |
| pckPickingOrderStartPicking |  |
| pckGetJobPrint |  |
| pckPickingOrderStopPicking |  |
| pckAdd |  |
| pckPickingSecureZoneFieldChange |  |
| pckGroupAddAllRoutes |  |
| pckPickingSecureZoneHomeCreate |  |
| pckRptCounterPickerPerformance |  |
| PckRptAdjustments |  |
| pckPickingSecureZoneHomeDestroy |  |
| pckOrdersNotAdjusted |  |
| pckPickingSecureZoneHomeFindByBranchLoc |  |
| pckPickingSlip |  |
| pckPickingSecureZoneRefresh |  |
| pckRemoveRoutes |  |
| pckPickingOrderCreateInvoice |  |
| PckRptPackingSlip |  |
| pckPickingSlipHeader |  |
| PckRptPickingSlip\_old |  |
| pckPickingSlipHomeCreate |  |
| pckPickingSlipDetail |  |
| pckSalesOrderSearch |  |
| pckPickingSlipIsOpen |  |
| pckStkTranDet |  |
| pckPickingSlipNotes |  |
| pckStockOffHold |  |
| pckRptPickRouteStats |  |
| pckRouteGroupAddRouteCode |  |
| pckRouteGroupGetRouteCodes |  |
| PckJobDetOnHoldChange |  |
| pckRouteGroupHomeCreate |  |
| pckRouteGroupHomeDestroy |  |
| pckPickingEventHomeFindByType |  |
| pckRouteGroupHomeFindByBranchLoc |  |
| pckRptValueOfUnPickedItems |  |
| pckRouteGroupRefresh |  |
| pckGetRouteSalesOrderHdr |  |
| pckRouteGroupRemoveRouteCode |  |
| pckWaitingOrders |  |
| pckJobDetOnInsertSalTran |  |
| pckJobDetOnDeleteSalTran |  |
| pckPickingOrderLineSetPckQtyBOFix |  |
| PckGetNextJobNo |  |
| pckGetAllRoutes |  |
| pckAdjustOrder |  |
| pckGetAllLabels |  |
| pckStock |  |
| pckDsRouteGroupHdrIU |  |
| pckTranDetNextInvoiceLineId |  |
| pckPickingOrderHomeFindByPickerRoutesX |  |
| pckTranDet |  |
| pckInvoiceJob |  |
| pckPickerFieldChange |  |
| PckJobDetUpdateHdrStatus |  |
| pckJobDetStatusChange |  |
| pckZone |  |
| pckGetPackPrint |  |
| pckRptPickingOnHold |  |
| pckAddGroupRoutes |  |
| pckPickingOrderLineSetPckQty |  |
| PckRptPickerStatsTest |  |
| PckRptIncompleteOrders |  |
| pckConstraints |  |
| pckInvoiceProcess |  |
| pckPickingOrderSetOnHold |  |
| pckGetAllJobs |  |
| pckGetGraph |  |
| pckGetHoldStock |  |
| pckGetJobPrintMessages |  |
| pckGetCDV |  |
| pckGetJobs |  |
| pckGetLabelJobs |  |
| pckPickingOrderRefresh |  |
| pckCounterLogInvoiceOutScan |  |
| pckGetInvoiceList |  |
| pckInvoiceMessages |  |
| pckPickingOrderAddToPickingSlip |  |
| pckNextPickingJob |  |
| pckCounterLogSlipPickingScan |  |
| pckAddSalesOrder |  |
| pckPickingOrderReroute |  |
| pckCounterLogSlipSecurityScan |  |
| pckSearchJob |  |
| pckSearchOrder |  |
| pckReprintInvoiceLog |  |
| pckSearchOrder1 |  |
| PckRerouteAllUnpickedOrders |  |
| pckPickingSlipLineHomeFindByNumber |  |
| pckSecureSlipDetailTest |  |
| pckDsRouteGroupDet |  |
| pckDsRouteGroupHdr |  |
| pckPicker |  |
| PckMaintRouteGroupsSearchRoute |  |
| pckGetInvoiceMessages |  |
| pckRptInvoiceMail |  |
| pckGetPackingMessages |  |
| pckCheckDeleted |  |
| pckHold |  |
| pckRptPickStats |  |
| PckRouteSalesOrdersRefresh |  |
| pckPickingSlipLineSetPckQty |  |
| pckGetInvoicePrint1 |  |
| pckSecureSlipHeader |  |
| pckGetJobHdr |  |
| pckViewRouteSalesOrderDet |  |
| pckSecureSlipGetZoneCodes |  |
| pckControl |  |
| pckJobDet |  |
| pckPickerHomeFindByBranchLocStatus |  |
| pckBaggingLabelGetZoneCodes |  |
| pckSecureSlipDetail |  |
| pckBaggingLabelDetail |  |
| pckRptPickerStatsDetail |  |
| pckPackingSlipHeader |  |
| pckJobHdr |  |
| pckPackingSlipDetail |  |
| PckJobDetValidateDelete |  |
| pckRptReprintOriginalInvoice |  |
| PckJobDetValidateInsert |  |
| pckPrtInvoiceJob |  |
| PckJobDetValidateStatus |  |
| PckJobDetValidateUpdate |  |
| pckJobDet\_with\_invalid\_pckJobHdr |  |
| PckJobDetDeleteStkTran |  |
| pckRouteGroupFieldChange |  |
| pckInvoicePreview |  |
| pckJobDetStatus |  |
| StkSalesOrderDetDetPck\_Old |  |
| pckRptInvoiceJob |  |
| pckPickerHomeCreate |  |
| pckZoneGroup |  |
| pckCounterScanLog |  |
| pckPickerHomeDestroy |  |
| pckPickerRefresh |  |
| pckPickingEventRefresh |  |
| pckPackingSlip |  |

# Programs

# MS Windows Executables

|  |  |
| --- | --- |
| pckBaggingReprint | Re-print bagging labels for small parts |
|  |  |
| pckJobs | List picking jobs not completed |
| pckMaintainReasons | Defined list of usable reasons i.e. job on hold etc |
| pckMaintainZones | Picking is divided into zone areas for specific pickers |
| pckMaintainRoutes | Picker is given work for one or more routes |
| pckReleaseOrders | One or more sales orders released to picking |
| pckReprintInvoice | Only one invoice may be original – this is marked as copy |
| pckReprintOriginalInvoice | For those customers that insist – controlled |
| pckRouteGroups | Routes are grouped together to optimise picking |
| pckSalesOrderTracking | Track sales order progress through ePart |
| pckSupervisor | Allocate permission to picker as supervisor to manage team |
| pckControl | Main controller to manage all picking processes |
| pckMaintainPickers | CRUD of picker resource |
| pckMaintainRouteGroups | CRUD to allocate routes into groupings |
| pckOrders | List all orders in the system with appropriate filtering |

# Acceptance

I hereby confirm that I have been fully informed of the documents content and, received training to understand how the detailed instructions are to be applied

Name …………………………………………………………………………….

Job Title ………………………………………………………………………….

Signed ……………………………………………………………………………

Date ………………………………………………………………………………