

**Version 1.1**

**Date: 2019/03/07**

Abstract ***Stock modelling forms part of an essential function within Engineparts due to the number of stock items that carry stock on hand. The replenishment cycle needs to operate an autonomous manner whenever possible with minor user input largely aimed to fine tune the selection and re-ordering process.***

***For new stocking items, multiple warehouse replenishment models need to be defined until sales / demand history has become effective***

Stock Workbench

Optimised Stock Modelling Integrating several sub-system data sources

**Table of Contents**

1. Document approval and distribution list 2

2. Introduction 3

3. Audience 3

4. Objectives 3

5. Database entities and relationships 5

6. Programs 6

6.1 MS Windows Executables 6

6.2 SQL Stored Procedures 6

7. Acceptance 8

# Document approval and distribution list

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

# Introduction

Due to the great number of stocking items and broad based demand planning, much of the purchasing decisions is driven by a semi-automated process based on sales demand, supplier fulfilment rate, available funding etc.

The function o the workbench is to give a broad yet definitive view of all the factors that have an influence on each stocking item’s strategic employment in Engineparts. Some of the elements presented:

* Supplier details of the specific stocking item such as industry part references, lead time country of origin
* Pricing available
* Rebate conditions
* Actual cost after rebate. This is required to be taken into consideration during cost price calculation as to not do so results in a market related selling price that is lower than the acquisition price
* Current state of active supplier orders.
* When will Engineparts run out of stock given delayed replenishment at current demand
* Sales demand trend graph
* Stock levels in the various warehouse locations (Branches)
* Comparative demands (AMD) of each warehouse location to see if over / under stocked.

More recently, post liquidation, an advanced stock demand modelling system was acquired and now forms [art of the bulk of purchasing done by Engineparts from internal (Imperial based) suppliers. However, for other suppliers, Engineparts remains reliant on the AMS / workbench modelling methodology.

# Audience

* Management
* Purchasing

# Objectives

To allow a ***workbench style of access to all pertinent aspects*** relating to the organisations need to manage its ability to:

* Not over-stock – purchase in line with actual demand. However, a reasonable level of over stock is good provided a bulk purchasing agreement would yield better than normal beneficial purchase pricing
* Not run out of stock to sell –
  + Early warning of low stocking levels
  + Identify alternate sources of supply
  + Fill in from local suppliers in case of delayed imports
* Manage the best pricing possible for both purchasing and selling
* Manage the even spreading of available stock on hand to the various points of selling and distribution – multi warehouse model.
* Manage import orders and the various stages it needs to follow
* Define sales discounts in various categories to meet with customer business profiles and to allow support of customers that operate in differing market segments. This would imply that auto electrical customers should earn a ***better*** discount that customers in engine overhauling by way of an example.
* There are many more aspects that could be added to this list.

The core driver of this part of the supply chain management in ePart is demand and fulfilment.

Sales demand is expressed as an AMS factor (3-month Average Monthly Sales). This is a calculated value using sales for the past period where every month’s sales are folded into the current AMS factor.

The main reason for using this method rather than re-calculating demand from the immediate past 3 months of turnover is that there can be an intrinsic demand without any stock availability. This will skew the AMS is based on the immediate past 3 month.

Additionally, the AMS recalculation is NOT recalculated when stock availability reaches zero. This is to preserve the purchasing model in the event of supply failure.

Considering that ePart is designed to be multi-company multi-warehouse capable, demand is apportioned to each point of ***consumption,*** carrying and own AMS factor deriving a collective demand model.

Each point of consumption with its own demand model provides input to the purchasing levels and frequency based on several factors exposed in the ***workbench*** program.

Some of these factors are GP%, AMS, supplier capacity / reliability to supply etc.

The notion should be for the purchaser to minimise stock holding without negatively affecting sales availability. Using the ePart workbench empowers buying staff to make effective and informed decisions to balance investment vs availability; there is a cost in purchasing too much stock and loss of GP if unable to supply.

# Database entities and relationships

# Programs

# MS Windows Executables

|  |  |
| --- | --- |
| **Name** | **Description** |
| stkMaintMarkupFlags.exe | The stock workbench application. |

# SQL Stored Procedures

The design of the workbench application’s database access mostly follows a pattern of:

1. Extract data into working tables
2. The user works on the working tables
3. The changes that the user made are posted from the working tables into live data

As such, most of the stored procedures are for putting data into the working tables. In conjunction with the database diagram, it should be easy to tell which these are, and the other procedures are named with clarity in mind:

|  |  |
| --- | --- |
| **Name** | **Note** |
| stkMaintMkpFlagsSupplierStockLists |  |
| stkMaintMkpFlagsLinkedTo |  |
| stkMaintMkpFlagsPost |  |
| stkMaintMkpFlagsSPLLine |  |
| stkMaintMkpFlagsWPLLine |  |
| stkMaintMkpFlagsWPLNew |  |
| stkMaintMkpFlagsDetBin |  |
| stkMaintMkpFlagsDetAsm |  |
| stkMaintMkpFlagsDetPartNr |  |
| stkMaintMkpFlagsDetPOHdr |  |
| stkMaintMkpFlagsDetPODet |  |
| stkMaintMkpFlagsDetGRVs |  |
| stkMaintMkpFlagsDetSusShipped |  |
| stkMaintMkpFlagsDetSusLabeling |  |
| stkMaintMkpFlagsDetSusTransfer |  |
| stkMaintMkpFlagsDetLostSales |  |
| stkMaintMkpFlagsAMSLookup |  |
| stk12MonthSalesFromCubeForStockCode |  |
| recPurchaseHistory |  |
| stkMaintMkpFlagsReturnsByMonth |  |
| stkWritedownHistory |  |
| stkMkpDetPOChangeAdd |  |
| stkGraphGAMS |  |
| stkGraphFreeStock |  |
| stkMaintMkpFlagsChangeHist |  |
| stkMaintMkpFlagsGetFlagButtons |  |
| stkMaintMkpFlagsGetAttribsMain |  |
| stkMaintMkpFlagsGetAttribsDetail |  |
| catBaseItemAttributeChangeField |  |
| stkMaintMkpFlagsGetWDHistory |  |
| **Name** | **Note** |
| catLogLostSaleManually |  |
| catBoMNodeHomeFindByStockCode |  |
| catBoMNodeHomeFindChildren |  |
| catBoMBaseItemHomeFindByBoMNodeCode |  |
| catSupplierPartMove | Move pricing history. |
| catSupplierPartSplit | Split pricing history. |
| stkLogWritedownReview |  |
| stkWritedownReviewHistory |  |
| stkMaintMkpFlagsBuyoutDetails |  |
| stkGetBuyoutSupplierInvoice |  |
| stk12MonthBuytouSalesForStockCode | Beware the spelling error on this one. |
| coRptLog | For logging report use and performance. |
| stkMaintMkpFlagsListHdr | The “list” contains the search results. |
| stkMaintMkpFlagsListDet |  |
| stkMaintMkpFlagsListAlt |  |
| stkMaintMkpFlagsListNew |  |
| stkMaintMkpFlagsListDelete |  |
| stkMaintMkpFlagsListHdrSetSelected |  |
| stkMaintMkpFlagsListHdrGetSelected |  |
| stkMaintMkpFlagsListDetSetSelected |  |
| stkMaintMkpFlagsListDetGetSelected |  |
| stkMaintMkpFlagsImport |  |
| catGetOpenSalesOrders |  |
| catAddToSalesOrder |  |
| stkMaintMkpFlagsAddToPO |  |
| stkMaintMkpFlagsAddToPOPopulate |  |
| stkMaintMkpFlagsAddToPOPost |  |

# Acceptance

I hereby confirm that I have been fully informed of the document’s content and received training to understand how the detailed instructions are to be applied:

Name ……………………………………………………………………………

Job Title ……………………………………………………………………………

Signed ……………………………………………………………………………

Date ……………………………………………………………………………