High Level Software Design

Specification

For

MixCont Lab Software Interface

*Revision 3.0*

**Table of Contents**

1. Introduction 4

1.1 Purpose 4

1.2 System Overview 4

1.2.1 MixCont Lab Software 4

1.2.2 Rubber Mix Solution 4

1.3 Definitions and Acronyms 4

2. Software Existing and Used 5

2.1 Mixcont System Database 5

2.2 Recipe Management System 5

3. Existing Process Flow 5

4. Proposed Process Flow 6

5. Implementation Approach 6

5.1 Implementation Constraints 6

6. Environmental Needs 7

7. Interface Table Requirements 7

7.1 RMS Database User Authentications 7

7.1.1 Test Profile Transaction (Table Name: interf) 8

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Name** | **Reason For Changes** | **Date** |
| 1.0 | BSK | First Issue | 14-Mar-‘14 |
| 2.0 | BSK | Table Structure Changed | 19-Mar-‘14 |
| 3.0 | BSK | Reviewed Version with MixCont | 21- Mar-‘14 |

**Approved By**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Signature** | **Department** | **Date** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

This high level design document, will detail the implementation techniques and the interface requirements for data exchange between the MixCont Lab Software and the Recipe Management System for Black Line and White Line.

## System Overview

### MixCont Lab Software

This is existing lab software currently used by the client, for the testing the quality parameter of the finished goods.

### Rubber Mix Solution

This is shop floor control software, which control the production process of black line and white line production lines.

## Definitions and Acronyms

|  |  |
| --- | --- |
| Acronyms | Explanation |
| RMS | Rubber Mix Solution |
| Movex | ERP system |
| ODBC | Open Database Connectivity (Data client software) |
| MIXCont | Quality Lab Software supplied by Others |

# Software Existing and Used

## Mixcont System Database

|  |  |  |  |
| --- | --- | --- | --- |
| Application | Software | Version | Platform |
| Database | MS SQL 200? |  | ? |

## Recipe Management System

|  |  |  |  |
| --- | --- | --- | --- |
| Application | Software | Version | Platform |
| Database | MS SQL 2012 |  | Window Server 2008 R2 SP1 64 Bit |

# Existing Process Flow

1. A Manufacturing Order is created in Movex System.
2. The information regarding the Manufacturing Order number, batch number, article details, test profile details are exchange from Movex system to MixCont Lab Software.
3. The finished goods are tested in the quality lab, the test parameter against the test profile are captured and registered against the manufacturing order number and batch no.
4. Based on the quality information and qualify criteria requirement, the lab technician approves the finished good produced.
5. The passed and accepted finished good is shipped to customer.

# Proposed Process Flow

1. A Manufacturing Order is created in Movex System. (existing)
2. The information regarding the Manufacturing Order number, batch number, article details, test profile details are exchange from Movex system to MixCont Lab Software. (existing)
3. The finished goods are tested in the quality lab, the test parameter against the test profile are captured and registered against the manufacturing order number and batch no. (existing)
4. Based on the quality information and qualify criteria requirement, the lab technician approves the finished goods produced. (existing)
5. The test value related to the manufacturing order and batch number will be exchange to RMS. (New)
6. The quality status of the manufacturing order will be exchange to the RMS. (New)
7. The RMS system updates the quality status as received from Mix Cont Lab for the manufacturing order in the Movex for shipping and invoicing.

# Implementation Approach

RMS will define a dedicated transaction table in RMS database schema and expose them to MixCont for populate the data.

MixCont system will populate the data at some defined frequency.

## Implementation Constraints

*Will be update later if any*.

# Environmental Needs

*Will be update later if any*.

# Interface Table Requirements

## RMS Database User Authentications

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.no | Field Name | Use Value | Remark |
| 1 | Table Schema Name | **TRELLEBORG** |  |
| 2 | User Name | **mixcont** |  |
| 3 | Password | **mixcont@1234** |  |
| 4 | Environment Server Name | **RMS** |  |
| 5 | Privilege Assigned | Insert /Update Only | Insert/Update Record Set |

### Test Profile Transaction (Table Name: interf)

|  |  |
| --- | --- |
| Table Name | interf |
| Use | This table is used to capture the test result of the each test code for a manufacturing ordering. |
| Scenario | During quality testing for each manufacturing order |
| Exceptions | If any profile testing fail, they may opt for retrial. The latest trail result will supersede the previous results. |
| Frequency | *Every one hour once from day start* |

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.no | Field Name | Data Type | Remark |
| 1 | VHMFNO | Char(15) | Manufacturing order number. |
| 2 | VHMBNO | Int | Batch Number |
| 3 | VHPRNO | Char(15) | Article number of finished goods. |
| 4 | TPCODE | Char(15) | Test Profile |
| 5 | TCODE | Char(5) | Test Code |
| 6 | TVER | Char(15) | Test Version |
| 7 | TTRAIL | Smallint | Test Trails Count , |
| 8 | TAVALUE | Float | Test Actual Value Captured |
| 9 | TDATETIME | DATETIME | Test Date and Time |
| 10 | USL | Float | Upper Set Limit( Customer Specification) |
| 11 | Float | Float | Lowe Set Limit ( Customer Specification) |
| 12 | UCL | Float | Upper Control Limit (Target Spec ) |
| 13 | LCL | Float | Lower Control Limit ( Target Spec) |
| 14 | UNITS | Char(10) | Measurement Units |
| 15 | QCSTATUS | Char(08) | Quality Status , Pass ,Fail Accepted , |
| 16 | EXPORTTIME | DATETIME | Time DataTransfer to RMS Sys |

Notes:

**Script Used for Creating Interface Table MSSQL2012**

*USE TRELLEBORG*

*SET ANSI\_NULLS ON*

*GO*

*SET QUOTED\_IDENTIFIER ON*

*GO*

*SET ANSI\_PADDING ON*

*GO*

*CREATE TABLE [dbo].[interf](*

*[VHMFNO] [char](15) COLLATE SQL\_Latin1\_General\_CP1\_CI\_AS NULL,*

*[VHMBNO] [int] NULL,*

*[VHPRNO] [char](15) COLLATE SQL\_Latin1\_General\_CP1\_CI\_AS NULL,*

*[TPCODE] [char](15) COLLATE SQL\_Latin1\_General\_CP1\_CI\_AS NULL,*

*[TCODE] [char](5) COLLATE SQL\_Latin1\_General\_CP1\_CI\_AS NULL,*

*[TVER] [char](15) COLLATE SQL\_Latin1\_General\_CP1\_CI\_AS NULL,*

*[TTRAIL] [smallint] NULL,*

*[TAVALUE] [float] NULL,*

*[TDATETIME] [datetime] NULL,*

*[LSL] [float] NULL,*

*[USL] [float] NULL,*

*[UCL] [float] NULL,*

*[LCL] [float] NULL,*

*[UNITS] [char](10) COLLATE SQL\_Latin1\_General\_CP1\_CI\_AS NULL,*

*[QCSTATUS] [char](8) COLLATE SQL\_Latin1\_General\_CP1\_CI\_AS NULL,*

*[ExportTime] [datetime] NOT NULL*

*) ON [PRIMARY]*

*GO*

*SET ANSI\_PADDING OFF*