Choco mambo db

TECHNICAL DESIGN DOCUMENT

21 March 2014

Robert Kelly

Contents

[PREFACE 2](#_Toc383200131)

[Purpose of this document 2](#_Toc383200132)

[Introduction 2](#_Toc383200133)

[Background 2](#_Toc383200134)

[Purpose 2](#_Toc383200135)

[Scope 3](#_Toc383200136)

[Definitions 3](#_Toc383200137)

[Program Overview 4](#_Toc383200138)

[Program Characteristics 4](#_Toc383200139)

[Program Context 4](#_Toc383200140)

[Types of data and Information that will be displayed in the interfaces 4](#_Toc383200141)

[Program Design 5](#_Toc383200142)

[Design methods and standards 5](#_Toc383200143)

[Class Maps 5](#_Toc383200144)

[Class Map Overview 5](#_Toc383200145)

[Address Class 6](#_Toc383200146)

[Business Contact Class 6](#_Toc383200147)

[Customer Class 6](#_Toc383200148)

[DB Connections Class 7](#_Toc383200149)

[Employee Class 7](#_Toc383200150)

[Order Class 8](#_Toc383200151)

[Order Lines Class 8](#_Toc383200152)

[Product Class 9](#_Toc383200153)

[Person 9](#_Toc383200154)

[Raw Ingredients Class 9](#_Toc383200155)

[Sale Class 10](#_Toc383200156)

[Sale Lines Class 10](#_Toc383200157)

[Stock Class 11](#_Toc383200158)

[Supplier Class 11](#_Toc383200159)

[Form Standard Design 11](#_Toc383200160)

[Documentation Standards 12](#_Toc383200161)

[Naming conventions 12](#_Toc383200162)

[Programing Standards 12](#_Toc383200163)

[Software development tools 12](#_Toc383200164)

[Document Control 13](#_Toc383200165)

# PREFACE

## Purpose of this document

This document is used to display the technical requirements and design standards that are used in the accompanying program. This document will provide a guide into developing and maintaining the aforementioned program.

# Introduction

## Background

“Choco Mambo” is a chocolate factory that buys raw ingredients from several suppliers and produces a variety of chocolate treats. Choco Mambo has several employees that works in 4 departments: Production, Office, Sales, and Delivery. The chocolate treats have their own name and code and they are sold to a number of retail stores by a sales agent.

Choco Mambo would like to keep track of their purchases of raw ingredients from their suppliers, and they would also like to keep records of their employees and the retailers who purchase their chocolates. The company would also like to keep track of the value and date of sales or purchases made as well as who they sold what chocolates treats to which retailers or bought what ingredients from which suppliers.

The company would like to keep the details of where the employees live and their contact details as well as their annual salary. In regards to retailers, the company would like to track the location of each retailers and the sales agent who manages their accounts.

Recently, the company opened their branches in Perth and Melbourne and they would like to track only the purchases made by each branch.

The security of the system is important to the company and they would like to assign different access rights to different users. Some users may have limited access to the features of the application, for example, some users may be able to enter product details but not enter the sales; some users may be able to enter sales but can only read the product details etc., or some users may have administration rights where they can access every features of the application. Each user will have their own user logins and one user will only be allocated with one login; there will be no cases where one user will have multiple access rights.

## Purpose

The purpose of this program is to provide efficient storage of data kept by “Choco Mambo”. The purpose of this program is to also enable the “User” to easily modify the information stored.

## Scope

By the completion of the project the program will have the following Characteristics:

1. Create, Read, Update and Delete data on the following
   1. Employees
   2. Customers
   3. Suppliers
   4. Products
   5. Raw Ingredients
   6. Sales
   7. Orders
2. Be user friendly, including the following
   1. Standard form layout
   2. Easy to read font
   3. “Easy on the eyes” colour scheme
   4. Clearly labelled buttons
3. Have user permissions classed by the following:
   1. Write – Can write to the table
   2. Read – Information is “read only”
   3. Deny – User is denied access to the information.
4. Have user logins including
   1. Username
   2. Password
   3. Whether is admin or not.

## Definitions

|  |  |
| --- | --- |
| CRUD | Create, Read, Update and Delete |
| Class Map | A method of Uml displaying the class layout of a program |
| User Friendly | A method of form design used to describe making the form easy to use |
| Class | An object oriented term used to group a set of related methods |
| Method | A small section of code used to perform a function |
| Coupling | In programming terms means the level that methods are dependent on each other |
| Cohesion | A programming term used to describe the naming of objects to describe their function |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Program Overview

## Program Characteristics

The Characteristics of the complete program includes:

1. The Ability to Create, Update, Read and delete data.
2. Handle multiple users at one time.
3. Handle data formats from the clients’ side.
4. User friendly.
5. Easy form flow.
6. Dynamic search box.
7. Login required.
8. Handle and log errors

# Program Context

## Types of data and Information that will be displayed in the interfaces

Data types displayed will include:

1. String (Text)
2. Long (Whole number)
3. Decimal (Number with decimal)

Information displayed will include:

1. Employee Details.
2. Customer Details.
3. Supplier Details.
4. Product Details.
5. Raw Ingredient Details.
6. Sale Details.
7. Order Details.

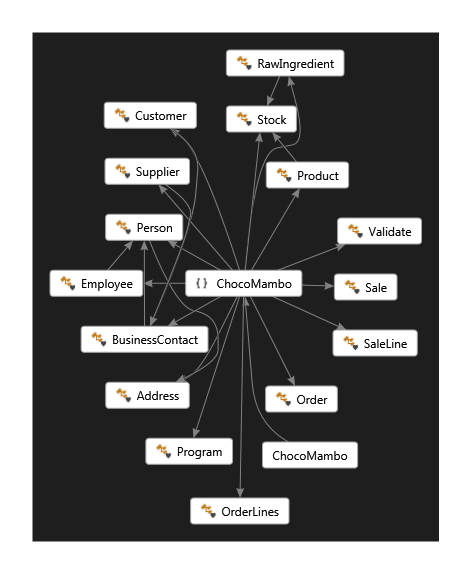
Note: Information displayed can include sensitive material therefore security measures are to be taken.

# Program Design

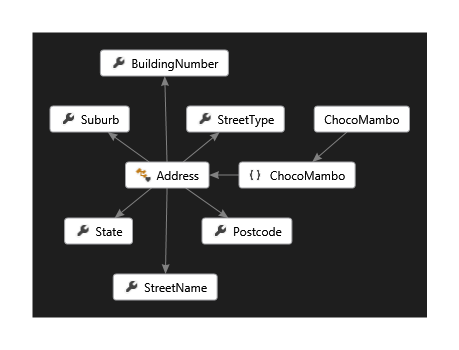
## Design methods and standards

## Class Maps

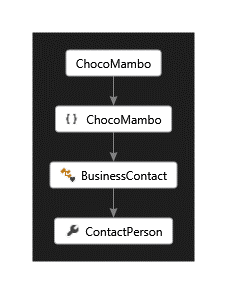
### Class Map Overview



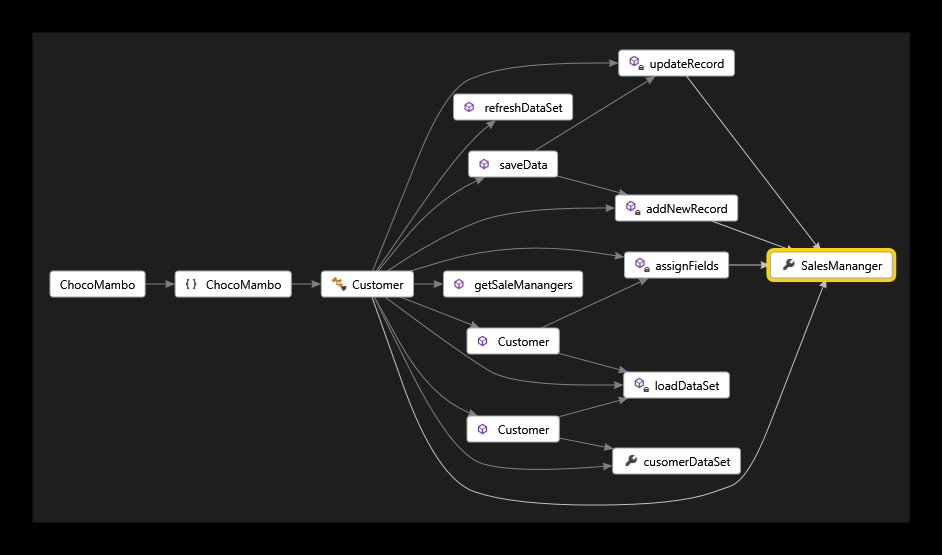
### Address Class



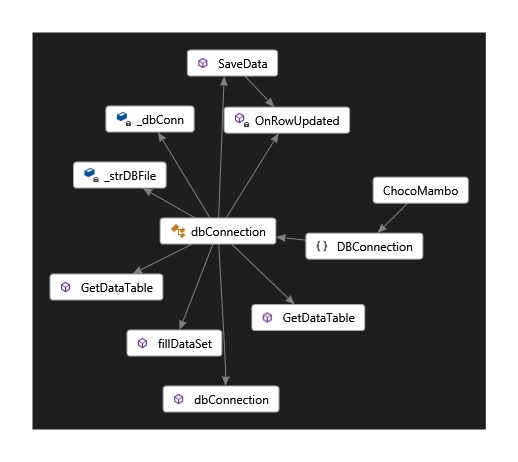
### Business Contact Class



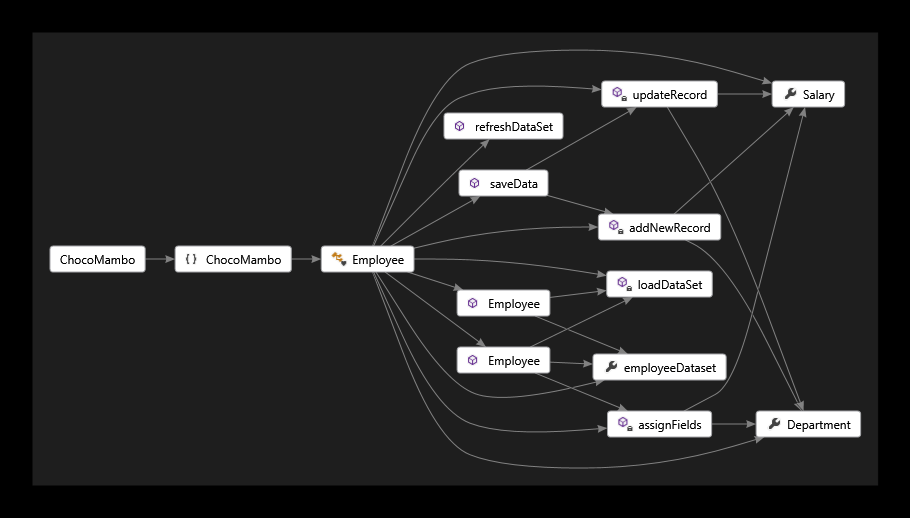
### Customer Class



### DB Connections Class



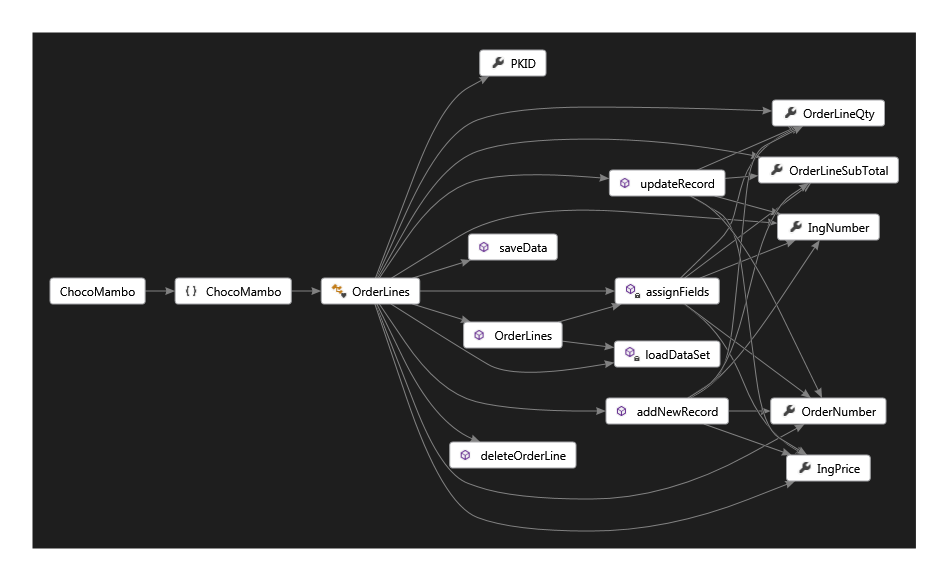
### Employee Class



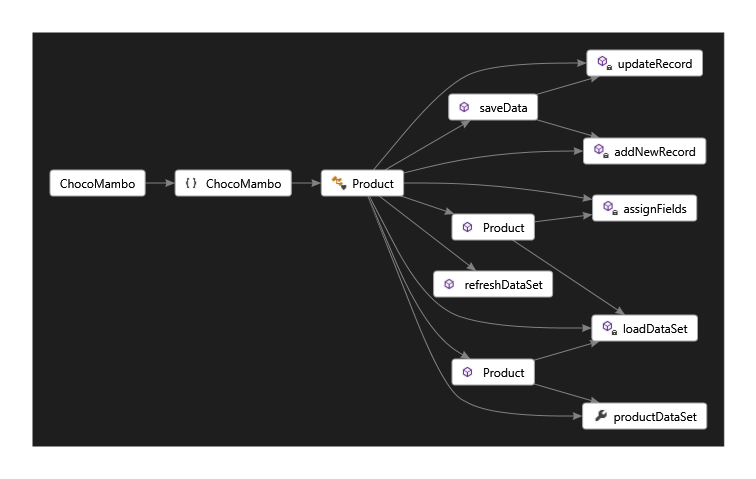
### Order Class

#### 

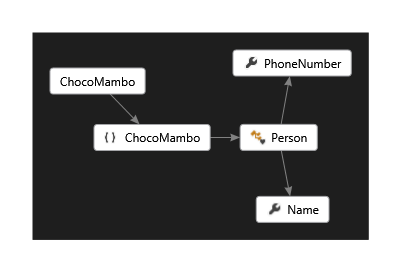
### Order Lines Class



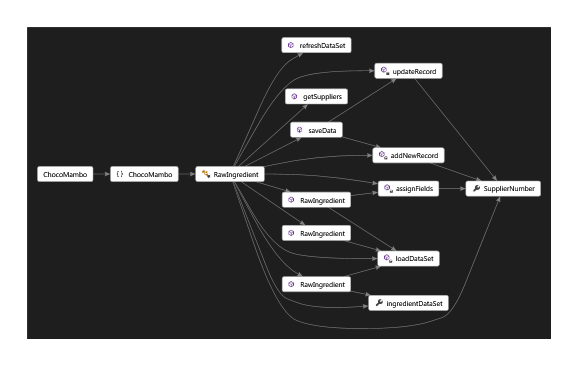
### Product Class



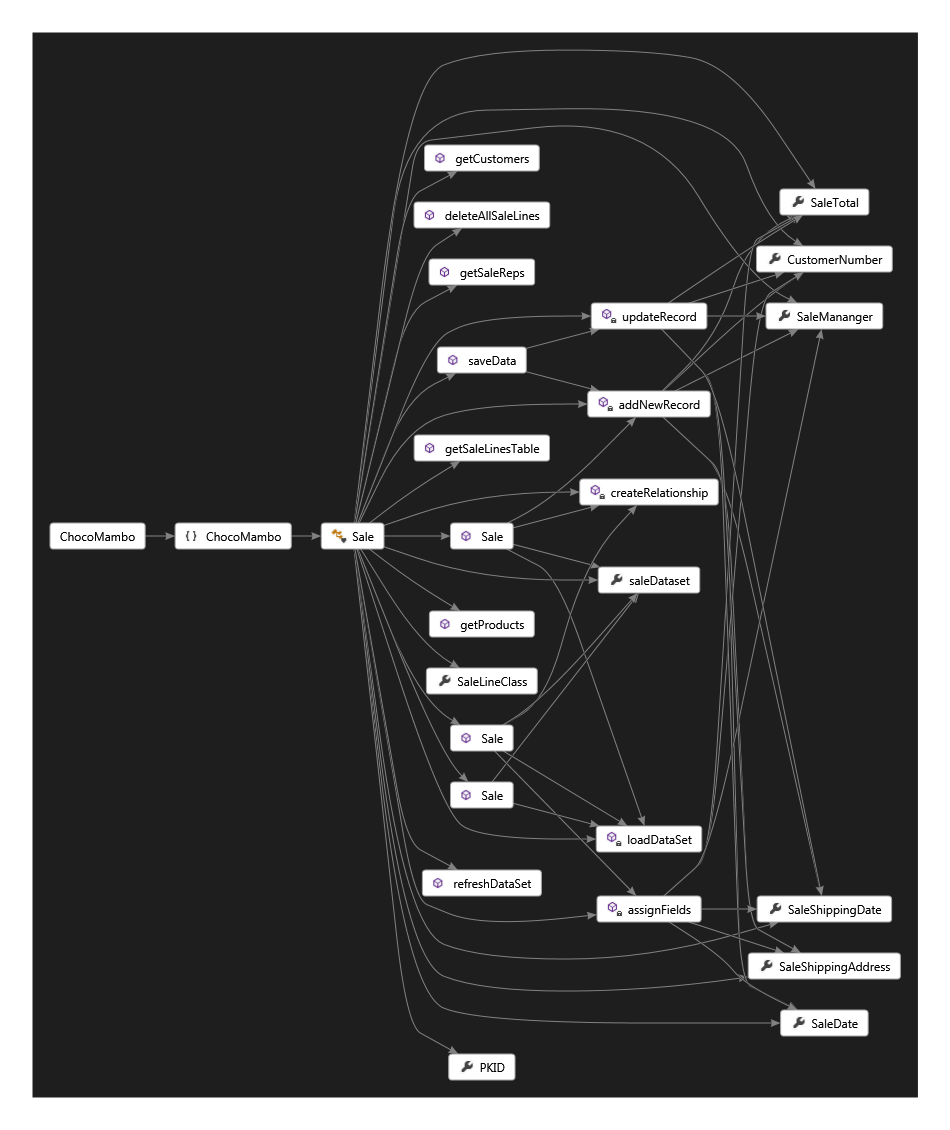
### Person



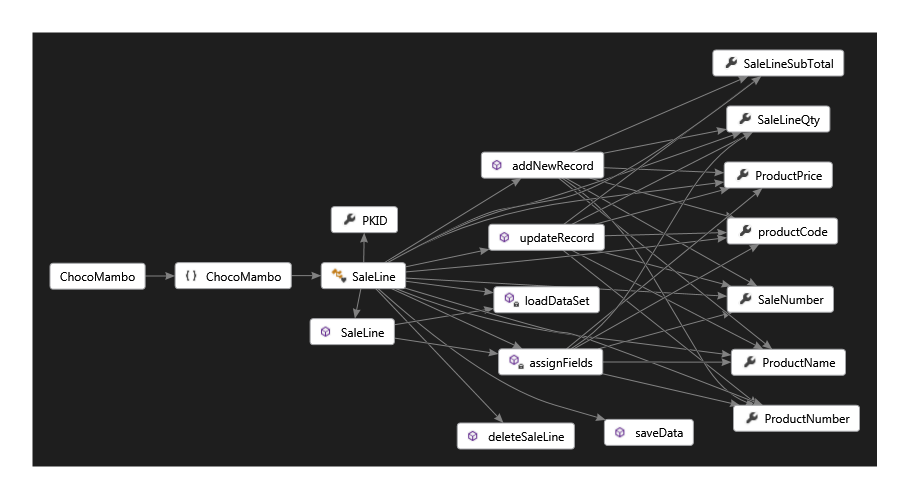
### Raw Ingredients Class



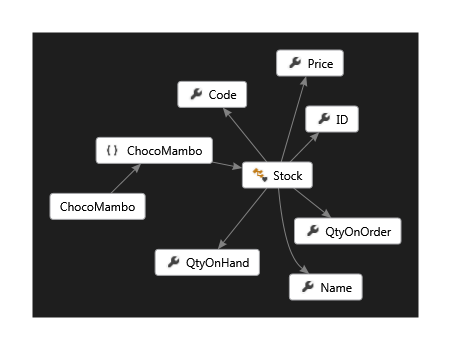
### Sale Class



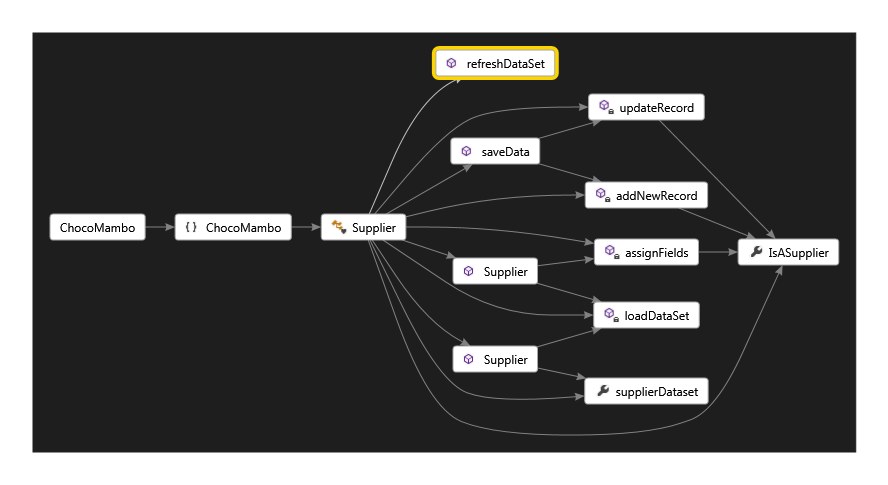
### Sale Lines Class



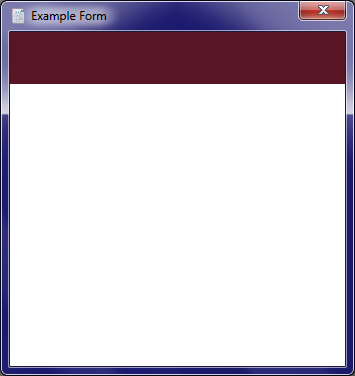
### Stock Class



### Supplier Class



## Form Standard Design

All forms must be built with the following standard:

1. Red panel across the top.
2. Frame border set to “Fixed 3D”.
3. Maximise and Minimise Buttons set to “False”.
4. Icon set.
5. Form name set.
6. Background colour set to White

## Documentation Standards

The program MUST contain the following documentation:

1. High level of internal documentation, Including Pre-Condition, Post-Condition and Description. Example See fig A.
2. User Manual, Including screen shots and troubleshooting.
3. Installation and uninstallation guides
4. Technical document, Including Preface, Introduction, Program Overview, Program Context, Program Design and Document Control.

## Naming conventions

Accompanying document naming convention:

ProjectName\_DocumentName, Example “ChocoMambo\_TechnicalDocument”

Other supporting resources should be named as follows:

Head folder: Project name.

Subfolder: Containing file category Example “TechnicalScreenshots”

Images: img\_ImageDescription Example “img\_LoginScreenShot”

## Programing Standards

When developing or updating this program the following standards must be adhered to.

1. All form names must start with frm\_
2. All objects in the form must have a name starting with a description of the object. Example TextBox1 will be named txt\_One, ComboBox2 will be named cbo\_Two
3. Rule two also applies to objects in classes with a few modifications. Global class variables must start with an underscore followed by an abbreviation of the object and a clear description on how the variable will be used. Examples – a String used to hold a name will be named \_strName, A long used to hold an id will be \_lngID and a DataTable used to hold employee information in the employee class will be named \_dtEmployeeInformation.
4. Getters and Setters must be used where ever possible.
5. Classes that hold the same information must have a super class that can be inherited from. Example a product class and an ingredient class could both contain some of the same fields they could be “Name”, “ID”, “Price”, “QtyOnHand”. These will be placed as properties in a super class called “Stock”. Both the product class and the ingredient class will inherit the Stock class.
6. All code written must have loose coupling and high cohesion.

## Software development tools

This program was developed in Visual studio 2013 ultimate. Any further updates should be done in visual studio 2013. In the case that a copy of visual studio 2013 in unavailable a substitute program may be suitable. See your project manager for approval.

# Document Control

Title: Choco Mambo Technical Document

Issue: Issue 1

Date: 21 March 2014

Author: Robert Kelly

Distribution: MSIT, Alan Jovellanos.

Filename: CMTechnicalDocument.Doc

Control: To be released as a whole Document with a copy of the Choco Mambo Database.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nature of Signoff | Person | Signature | Date | Role |
| Authors | Robert Kelly |  |  | Project Manager |
| Reviewers | Michael Borodin |  |  | Consultant |

|  |  |  |  |
| --- | --- | --- | --- |
| Date: | Version | Author | Change Details |
| 17 March 2014 | 1.0 | Robert Kelly | First Draft |
| 20 March 2014 | 1.5 | Robert Kelly | Final Draft |
| 21 March 2014 | 2.0 | Robert Kelly | Published Document |
|  |  |  |  |
|  |  |  |  |