|  |  |
| --- | --- |
| Product Name | IRIS Exchequer Prospect Manager |
| Proposed Version | v7 |

05/03/2010

**High Level Design**

**Paul Rutherford/Mark Higginson**

**Version**

# Revision and Signoff Sheet

## Revision

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Version | Change Reference |
| 05/03/2010 | Paul Rutherford | 0.1 | First Draft |
| 19/03/2010 | Paul Rutherford/ Mark Higginson | 0.2 | Revision following review and comments by MH. |
| 24/03/2010 | PR/MH | 0.3 | Revision following discussion with MH |
| 26/03/2010 | PR/MH | 0.4 | Added UDF customisation + labels to Interactions section of Technical Spec of Requirement 1. |
| 31/03/2010 | PR/MH | 0.5 | Added Appendix 4 – Low Level Design for Conversion Process.  Revised following KH’s comments. |

## Reviewers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Position | Version Approved | Date |
| Kevin Horlock | ???? |  |  |
| David Rustell | QA Manager |  |  |
| Tansey Roberts | QA |  |  |
| Training / Tech Support????? |  |  |  |

Contents

[Revision and Signoff Sheet 2](#_Toc257811821)

[Revision 2](#_Toc257811822)

[Reviewers 2](#_Toc257811823)

[Purpose of Document 6](#_Toc257811824)

[SDLC Checkpoints That Must Be Covered 6](#_Toc257811825)

[SDLC Checkpoints Currently Covered 6](#_Toc257811826)

[HLD Instructions 7](#_Toc257811827)

[Marketing Requirements Scope 8](#_Toc257811828)

[User Profiles 8](#_Toc257811829)

[Coding Standards 9](#_Toc257811830)

[Source Control Procedures 10](#_Toc257811831)

[Requirements 10](#_Toc257811832)

[Requirement < 6.3.PRO.42/6.3.PRO.47 > – Requirement 1 10](#_Toc257811833)

[Estimates 10](#_Toc257811834)

[Requirement < 6.3.PRO.43 > – Requirement 2 11](#_Toc257811835)

[Estimates 11](#_Toc257811836)

[Requirement < 6.3.PRO.44 > – Requirement 3 12](#_Toc257811837)

[Estimates 12](#_Toc257811838)

[Requirement < 6.3.PRO.45 > – Requirement 4 13](#_Toc257811839)

[Estimates 13](#_Toc257811840)

[Requirement < 6.3.PRO.46 > – Requirement 5 14](#_Toc257811841)

[Estimates 14](#_Toc257811842)

[Requirement < 6.3.PRO.42c > – Requirement 6 15](#_Toc257811843)

[Estimates 15](#_Toc257811844)

[Requirement < 6.3.PRO.42d > – Requirement 7 16](#_Toc257811845)

[Estimates 16](#_Toc257811846)

[Requirement < 6.3.PRO.47f > – Requirement 8 17](#_Toc257811847)

[Estimates 17](#_Toc257811848)

[Iteration Design 18](#_Toc257811849)

[Proposed Development Machine Specification 20](#_Toc257811850)

[Proposed Build Machine Specification 20](#_Toc257811851)

[Proposed Test Machine Environments 21](#_Toc257811852)

[Revised Gate Dates 21](#_Toc257811853)

[Appendix 1 – Technical Design 22](#_Toc257811854)

[Requirement < 6.3.PRO.42/6.3.PRO.47 > – Technical design 22](#_Toc257811855)

[Requirement < 6.3.PRO.43 > – Technical design 31](#_Toc257811856)

[Requirement < 6.3.PRO.44 > – Technical design 33](#_Toc257811857)

[Requirement < 6.3.PRO.45 > – Technical design 34](#_Toc257811858)

[Requirement < 6.3.PRO.46 > – Technical design 36](#_Toc257811859)

[Requirement < 6.3.PRO.42c > – Technical design 37](#_Toc257811860)

[Requirement < 6.3.PRO.42d > – Technical design 39](#_Toc257811861)

[Requirement < 6.3.PRO.47f > – Technical design 41](#_Toc257811862)

[Appendix 2 - Test Approach 42](#_Toc257811863)

[Test Plan Outline 42](#_Toc257811864)

[References 42](#_Toc257811865)

[Test Items (Functions) 42](#_Toc257811866)

[Features to be Tested 42](#_Toc257811867)

[Features not to be Tested 43](#_Toc257811868)

[Approach (Strategy) 44](#_Toc257811869)

[Pass Strategy 44](#_Toc257811870)

[Testing Tools 44](#_Toc257811871)

[Test Development Tools 44](#_Toc257811872)

[Test Execution Tools 44](#_Toc257811873)

[Testing Metrics 44](#_Toc257811874)

[Metrics Collected During Test Development 44](#_Toc257811875)

[Metrics Collected During Test Execution 45](#_Toc257811876)

[Configuration Management 46](#_Toc257811877)

[Test Configurations 46](#_Toc257811878)

[Regression Testing 47](#_Toc257811879)

[Automated Regression Testing 47](#_Toc257811880)

[Manual Regression Testing 47](#_Toc257811881)

[Status Reporting 48](#_Toc257811882)

[Item Pass/Fail Criteria 48](#_Toc257811883)

[Test Deliverables 48](#_Toc257811884)

[Test Environment 49](#_Toc257811885)

[Staffing and Training Needs 49](#_Toc257811886)

[Roles and Responsibilities 49](#_Toc257811887)

[Schedule 51](#_Toc257811888)

[Planning Risks, Contingencies and Assumptions 52](#_Toc257811889)

[Appendix 3 – New Data Records 53](#_Toc257811890)

[Appendix 4 – Low-Level Design for Conversion Process 58](#_Toc257811891)

# Purpose of Document

The purpose of the High level Design (HLD)

1. To design each requirement to the extent that
   1. We are confident with the estimates contained in the document
   2. We provide enough information for testers to produce test scripts
   3. All requirements are fully understood by development and test teams
2. To provide an agreed level of testing in advance of building the product
3. That requirements are designed to such a level that the target completion date can be achieved.

# SDLC Checkpoints That Must Be Covered

The High Level Design (HLD) document is part of the planning stage of the project, the planning stage of the project seeks to pass the following check points. The checkpoints that should be contained in this document are indicated in the first part of the table.

|  |  |  |
| --- | --- | --- |
| Id | Description | In Document |
| P1 | Have High Level Design Documents been completed for all requirements? | High Level Design |
| P2 | Have detailed work estimates been produced? | High Level Design |
| P3 | Has an approved test plan been produced for the release? | High Level Design |
| P4 | Has build frequency been defined and communicated? | High Level Design |
| P5 | Have coding standards been reviewed and published? | High Level Design |
| P6 | Have backup and recovery processes been reviewed and published? | High Level Design |
| P7 | Have source control procedures been reviewed and published? | High Level Design |
| P8 | Are separate build and test environments in place? | Detailed in High Level Design |
| P9 | Has the process for transferring builds to test team/test environment been reviewed and published? | High Level Design |
| P10 | Is a base lined, detailed project plan in place covering all Development Team release activity? | Project Plan |
| P11 | Has a revised target completion date been approved by Product Management and published? | Project Plan |

# SDLC Checkpoints Currently Covered

This section should include the checkpoints from the previous table that are currently satisfied in the current version of this document.

# HLD Instructions

*This document should use the Product Requirements Document (PRD) as its input. Each requirement from the PRD should*

* *Have a technical Analysis Performed on it*
* *Have a test plan (per requirement) completed for it.*
* *Refine the estimates when the high level design is completed (which will be used to create a base lined project plan)*

*The Architect should define the overall architecture, and perform a high level analysis of the deployment considerations*

*The development team should produce a set of agreed coding standard (or point to a set of already published ones) and indicate which platforms will be supported for the release.*

*The test should define the test environments*

*All the development team should define the iteration plan (i.e. which features will be delivered in which iteration)*

# Marketing Requirements Scope

*This section should give a brief overview of the requirements that the HLD will address, it should also point to published requirements list e.g. a SharePoint list or reproduce the agreed requirements from the PRD (In MoSCoW format)*

*Link to copy of PRD*

# User Profiles

*Profiles and personas of typical users can be found in the Marketing Requirements Document for this project.*

*Link to copy of MRD*

# Coding Standards

The standard Exchequer Coding Standards will be used (see ????????).

# Source Control Procedures

The Prospect Manager will be a module of Exchequer with all new code being compiled in to the existing programs. A separate branch will be created in the Source Code Repository to accommodate the changes and this will be merged back into the Exchequer trunk once the project is complete.

# Requirements

*This section should list all requirements for the project; these requirements can be deployment, system, platform, or functional types. The table below represents the details needed for one requirement. One table must be completed for every requirement.*

## Requirement < 6.3.PRO.42/6.3.PRO.47 > – Requirement 1

| Requirement Name and Number | | Trader List – Prospects Tab  Prospects - Opportunities | | 6.3.PRO.42  6.3.PRO.47 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.PRO.42 > – Refer to PRD  < 6.3.PRO.47 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.PRO.42/6.3.PRO.47 > – Technical design*](#_Requirement_<_6.3.PRO.42) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.PRO.43 > – Requirement 2

| Requirement Name and Number | | Toolkits – Prospects Support | | 6.3.PRO.43 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.PRO.43 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.PRO.43 > – Technical design*](#_Requirement_<_6.3.PRO.43) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.PRO.44 > – Requirement 3

| Requirement Name and Number | | Importer – Prospects Support | | 6.3.PRO.44 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.PRO.44 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.PRO.44 > – Technical design*](#_Requirement_<_6.3.PRO.44) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.PRO.45 > – Requirement 4

| Requirement Name and Number | | Data Dictionary – Prospects Support | | 6.3.PRO.45 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.PRO.45 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.PRO.45 > – Technical design*](#_Requirement_<_6.3.PRO.45) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.PRO.46 > – Requirement 5

| Requirement Name and Number | | Prospects – Form Designer Support | | 6.3.PRO.46 |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.PRO.46 > – Refer to PRD | |
| Technical Design | | | [*< 6.3.PRO.46 > – Technical design*](#_Requirement_<_6.3.PRO.46) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.PRO.42c > – Requirement 6

| Requirement Name and Number | | Prospects – Licencing | | 6.3.PRO.42c |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.PRO.42c > – Refer to PRD | |
| Technical Design | | | [*< 6.3.PRO.42c > – Technical design*](#_Requirement_<_6.3.PRO.42c) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.PRO.42d > – Requirement 7

| Requirement Name and Number | | Prospects – User Rights | | 6.3.PRO.42d |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.PRO.42d > – Refer to PRD | |
| Technical Design | | | [*< 6.3.PRO.42d > – Technical design*](#_Requirement_<_6.3.PRO.42d) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

## Requirement < 6.3.PRO.47f > – Requirement 8

| Requirement Name and Number | | Prospects – Workflow Diary | | 6.3.PRO.47f |
| --- | --- | --- | --- | --- |
| Requirement Type | Functional | | | |
| Business Analysis | | | < 6.3.PRO.47f > – Refer to PRD | |
| Technical Design | | | [*6.3.PRO.47f > – Technical design*](#_Requirement_<_6.3.PRO.47) | |
| Estimates *This section should be used to provide a breakdown of estimates for each requirement.*   |  |  |  | | --- | --- | --- | | **Activity** | **Estimate** | **Confidence** | | Deployment changes |  |  | | Low level use case refinement |  |  | | Low level design |  |  | | Low level test case refinement |  |  | | Data changes |  |  | | Business logic changes |  |  | | User interface changes |  |  | | Testing |  |  | | **Total** |  |  | | | | | |

# Iteration Design

*The Iteration Design for the Prospects Manager project has been designed at Product Management’s request to provide a working prototype of the basic functionality at the earliest practical date (Iteration 2) to allow them to start demonstrating the prototype to relevant parties.*

*This means that some requirements are going to have to be split in development to initially provide that basic functionality, with a full implementation of the functionality following later.*

*We have prioritised the complex and high-risk elements within the first 3 iterations.*

*A full build will be issued to Product Management and QA at the completion of each iteration, it is also envisaged that each iteration will include fixes for any issues found in previous iterations.*

*The planned Start and End Dates for each iteration are available in the Excel Project Plan.*

|  |  |  |
| --- | --- | --- |
| ***Iteration 1 – Infrastructure + Basic Prospects*** | | |
| ***Requirement*** | ***Dev Effort (Days)*** | ***Test Effort (Days)*** |
| *Database Changes – Pervasive Edition* | *TBC* | *TBC* |
| *Database Changes – SQL Edition* | *TBC* | *TBC* |
| *Installer Implementation of Database Changes* | *TBC* | *TBC* |
| *Business Objects* | *TBC* | *TBC* |
| *Main Exchequer Window* | *TBC* | *TBC* |
| *Trader List – Customer/Supplier Prospects Tabs* | *TBC* | *TBC* |
| *Customer/Supplier Prospects Dialog* | *TBC* | *TBC* |
| ***Total*** | *TBC* | *TBC* |

|  |  |  |
| --- | --- | --- |
| ***Iteration 2 – Opportunities Transaction*** | | |
| ***Requirement*** | ***Dev Effort (Days)*** | ***Test Effort (Days)*** |
| *Customer/Supplier Prospects Tabs – Quotes Button* | *TBC* | *TBC* |
| *Customer/Supplier Prospects Dialog – Quotes Tab* | *TBC* | *TBC* |
| *Opportunities Transaction Header(Full)* | *TBC* | *TBC* |
| *Opportunities Transaction Line (Full)* | *TBC* | *TBC* |
| ***Total*** | *TBC* | *TBC* |

|  |  |  |
| --- | --- | --- |
| ***Iteration 3 – Other Opportunity Areas*** | | |
| ***Requirement*** | ***Dev Effort (Days)*** | ***Test Effort (Days)*** |
| *Customer/Supplier Prospects Tabs – Convert Prospect* | *TBC* | *TBC* |
| *Customer/Supplier Prospects Tabs – Delete Prospect* | *TBC* | *TBC* |
| *Add a Transaction Dialog* | *TBC* | *TBC* |
| *Purchase/Sales Opportunities Daybooks* | *TBC* | *TBC* |
| *Workflow Diary* | *TBC* | *TBC* |
| ***Total*** | *TBC* | *TBC* |

|  |  |  |
| --- | --- | --- |
| ***Iteration 4 – Other Prospect/Opportunity Areas*** | | |
| ***Requirement*** | ***Dev Effort (Days)*** | ***Test Effort (Days)*** |
| *Object Credit Controller* | *TBC* | *TBC* |
| *Object Drill* | *TBC* | *TBC* |
| *System Setup – Stages* | *TBC* | *TBC* |
| *Sales/Purchase Quotes – Stage/Opportunity Rating Fields* | *TBC* | *TBC* |
| *Telesales* | *TBC* | *TBC* |
| ***Total*** | *TBC* | *TBC* |

|  |  |  |
| --- | --- | --- |
| ***Iteration 5 – Implement Standard Exchequer Functionality*** | | |
| ***Requirement*** | ***Dev Effort (Days)*** | ***Test Effort (Days)*** |
| *Customer/Supplier Prospects Tabs - Complete* | *TBC* | *TBC* |
| *Customer/Supplier Prospects Dialog – Complete* | *TBC* | *TBC* |
| *Form Designer* | *TBC* | *TBC* |
| ***Total*** | *TBC* | *TBC* |

|  |  |  |
| --- | --- | --- |
| ***Iteration 6 – Other Modules*** | | |
| ***Requirement*** | ***Dev Effort (Days)*** | ***Test Effort (Days)*** |
| *Toolkits* | *TBC* | *TBC* |
| *Importer* | *TBC* | *TBC* |
| *Visual Report Writer* | *TBC* | *TBC* |
| ***Total*** | *TBC* | *TBC* |

|  |  |  |
| --- | --- | --- |
| ***Iteration 7 – Misc*** | | |
| ***Requirement*** | ***Dev Effort (Days)*** | ***Test Effort (Days)*** |
| *Licensing* | *TBC* | *TBC* |
| *User Permissions* | *TBC* | *TBC* |
| *Colour/Window Positions* | *TBC* | *TBC* |
| *Update Plug-Ins (Contacts & UserDefined Fields)* |  |  |
| ***Total*** | *TBC* | *TBC* |

# Proposed Development Machine Specification

The following table should be completed for the “reference” development machine that will be used to develop the release.

|  |  |
| --- | --- |
| Machine Specification |  |
| CPU | 2.0 GHz Minimum |
| Memory | 2 GB Minimum |
| Development Tools | Borland Delphi 6.01  Pervasive.SQL v10 WGE  SQL Server 2005 |
| Design Tools | Microsoft Visio |
| Documentation Tools | Microsoft Word |
| Issue Reporting Tools | JIRA  (Account Also Needed) |
| Project Tools | MS Excel 2008 |
| Source Control Tools | Subversion  (Account Also Needed) |

# Proposed Build Machine Specification

The following table should be completed for the “reference” build server that will be used to build the release.

|  |  |
| --- | --- |
| Machine Specification |  |
| OS | Windows XP SP2 Pro – VMWare Virtual Machine |
| CPU | N/A – Virtual Machine |
| Memory | 512 MB Minimum |
| Build Tools | FinalBuilder 6  FinalBuilder Server 6  Borland Delphi v6.01 |
| Redistributables Needed | MDAC  Windows Installer 3.1  Ifx  Etc. |
| Installer Tools | Wise Installation System 9.01 |
| Source Control Tools | Subversion |

# Proposed Test Machine Environments

Here is where the test manager will define a list of test environments what will be needed to ensure testing is performed correctly. These environments will more than likely take their initial state from the system / platform analysis which defined what environments should run on.

It would be advantageous at this stage to define which the main test environments are, and which will be used to smoke test the application.

These environments will be used during development testing and the full testing at the verify stage.

|  |  |
| --- | --- |
| Machine Specification | Test Machine Specification ID |
| CPU | 2.0 GHz Minimum |
| Memory | 2 GB Minimum |
| Virtualised PC OK? | Yes No |
| Operating System |  |
| Service Pack Level for OS |  |
| Applications | *Application list*  *Including list of service pack levels for the application* |
| Settings | *Detail any special settings that are required on the machine, network, internet access, firewall etc.* |
| Client / Server / Or Both? | *Client, Server, Both* |

# Revised Gate Dates

This section should provide a revision of the gate milestone now that the high level design is complete.

**

## Appendix 1 – Technical Design

### Requirement < 6.3.PRO.42/6.3.PRO.47 > – Technical design

| Requirement Name and Number | | Trader List - Prospects Tab/Quotations for Prospects | 6.3.PRO.42/6.3.PRO.47 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| ImplementationTechnologies *The Prospect Manager will be written in Delphi 6, using a three-tier structure of UI-Business Object-Data Access Object. This will allow code written at the business logic level to ignore differences between Pervasive and MS-SQL databases, which will be automatically handled by the Data Access Objects. It will also allow the same Business Objects to be used by the toolkits and the importer, enabling us to avoid duplication of code as much as possible.* Data changes *The following new tables will be added (one set for each company):*  ***Prospect*** *– each record contains the details of one Prospect*  ***Prospect Quote Header*** *– each record contains the header details of a quotation for a prospect, linked to the Prospect record by the primary Prospect key field.*  ***Prospect Quote Line*** *– each record contains one line of a quotation, linked to the Quote Header record by the primary Quote Header key field.*  ***Stage*** *– each record contains a user-defined description of a stage in either the Prospect process or the Prospect Quote process.*  ***Contacts*** *– each record contains one contact for a Prospect. This table will be the same format as the existing Contacts table from the Contacts plug-in with two extra fields for MobilePhoneNo and NoteLine. (According to the PRD it is intended that the Contacts Plug-In table will be extended to include these new fields.)*  *Details of proposed records and indexes in these tables are given in* [*Appendix 3*](#_Appendix_3_–) *below:*  Stage  Contact  Prospect  Opportunity  Stage  Opportunity Detail  *A data access object will be created for each table to encapsulate navigation and read/write access to the table. These data access objects will automatically determine whether the Exchequer installation is using Pervasive or MS-SQL and will access the database accordingly, removing any need for higher level functions to distinguish between database types. Note: The business objects will go direct to SQL Server in the SQL Edition bypassing the SQL Emulator.* Changes to existing tables   *Two new fields will be added to the Document table: Stage; Opportunity Rating.*  *Two new records will be added to ExchqNum.dat to store document numbers for Sales and Purchase Opportunities. (As current functions for getting document numbers use DocTypes as parameters, new functions will need to be added to retrieve SOP and POP document numbers.)* Business Logic Changes *The following new forms will be added to encompass the Prospect functionality: Prospect details form; Prospect Opportunity Daybook; Prospect Opportunity Form; Prospect Opportunity Details form; Stage List; Stage Details.*  *Details of the fields required for these forms can be found in the PRD (v0.7) as follows:*  *Prospect Form: Pages 12-16; Prospect Opportunity & Details forms: Pages 22-23.*  *The Prospect Opportunity Daybook will be modelled upon the standard SQU/PQU daybooks and will have separate tabs for Sales and Purchase Opportunities with Close, Add, View, Edit, Delete, Find, Print, Copy and Convert buttons.*  *The Stage List will be available from the Utilities\System Setup menu in Exchequer and will display a list of available stages with the following fields: Stage Type (for Sales or Purchase Prospect or Opportunity), Stage Name. The list form will have buttons allowing the user to View, Add, Edit, Insert, and Delete a Stage, and to move the Stage up or down in the list. The Stage Details form will be shown when the user is adding, editing or deleting a Stage; it will include the following fields Stage Type; Stage Name;*  *The following functionality will be added for Prospects, Prospect Opportunities and Stages.*  *[Conventions used:*  View-only mode: *a record is shown in a form with all fields set as read-only, the OK button disabled and the Cancel button labelled ‘Close’.*  Add mode: *a new record is displayed in a form with all writeable fields enabled and the OK button and the Cancel button enabled. No fields are initially populated unless specified in the text. Clicking OK will insert the record into the file.*  Edit Mode: *an existing record is locked and displayed in a form with all editable fields enabled and the OK button and the Cancel button enabled. Clicking OK will update the existing record with the details from the form and unlock the existing record.]*  ***Prospect: View; Add; Edit; Delete; Find; Convert to Customer/Supplier; Print Details.***  *Note: The Contact details for a Prospect are taken from the Contacts table using the Contact name as a key.*  ***View:*** *This will be activated from the View button on either of the Prospect tabs on the Customer/Supplier daybook and will display the Prospect form in view-only mode for the prospect selected in the list on the current tab.*  ***Add:*** *This will be activated from the Add button on either of the Prospect tabs on the Customer/Supplier daybook and will display the Prospect form with no details allowing the user to enter a new prospect. It may also be activated from the Add button on the Prospect form in which case the current form will be cleared of all details.*  ***Edit:*** *This will be activated from the Edit button on either of the Prospect tabs on the Customer/Supplier daybook and will display the Prospect form in edit mode for the prospect selected in the list on the current tab, allowing the user to edit the details of the selected Prospect. It may also be activated from the Edit button on the Prospect form in which case the current form will be placed into edit mode.*  *When the prospect form is in Add or Edit mode, pick lists will be available for the Stage, CC, Dept, Location, Ctrl Code, Sales GL Code, Cos GL Code fields. These pick lists will be shown on the exit event of these fields unless the field is blank. Other fields which require validation will be validated on exit, unless the field is empty.*  *When the prospect form is in Add or Edit mode it may be stored, in which case the following fields will be validated: Account Code (must be populated and unique); Company Name (must be populated); Primary Contact (must be populated and unique for this prospect.) Any changes to the details of any of the contacts must be written to the Contacts table.*  ***Delete:*** *This will be activated either from the Delete button on the Customer/Supplier daybook (Prospect tabs) or from the Delete button on the Prospect form. A message will be displayed asking the user to confirm that they wish to delete the Prospect record. On confirmation, the Prospect record will be deleted from the database, as will any Opportunity, Discount, MultiBuy, Contact, Note and Link records associated with that Prospect.*  ***Find:*** *This will be activated either from the Find button on the Customer/Supplier daybook (Prospect tabs) or from the Find button on the Prospect form. It will display the ObjectFind dialog with only the appropriate Prospect tab visible. The search options will be AccountCode, AltSearchCode, CompanyName, and Contact.*  ***Convert:*** *This will be activated from the Convert button on either of the Prospect tabs on the Customer/Supplier daybook. It will create a new Customer or Supplier record (according to the type of the selected Prospect) and copy the details of the selected Prospect record to the new account record. A new unique account code will be suggested but the user will be able to override this. Any Discount, MultiBuy, Contact, Note and Link records associated with the Prospect will be modified to refer to the new account code. Any Opportunities associated with the Prospect will be converted to standard Quotations associated with the new account record.*  *Because of the number of records that are changed in this process, it should be conducted within a database transaction so that, should the process fail at any point, it will be possible to roll back the process to leave the database in the state it was in when the process started, allowing the user to try the conversion again. (But see the discussion in Open Issues below.)*  ***Print:*** *This will be activated from the Print button on the Prospect List or Prospect Record and will display the Print dialog for the Prospect with the default details from the Prospect record and the default form from Form Definition Sets, the user will be able to print the Prospect to Printer, Fax or Email only.*  ***Opportunity – Add; View; Edit; Delete; Find; Copy; Convert; Print.***  ***View:*** *This will be activated from the View button on the Opportunities tab of a Prospect record form or on the Opportunities Daybook and will display the Opportunity form in view-only mode for the selected Opportunity.*  ***Add:*** *This will be activated from the Add button on the Opportunities tab of a Prospect record form or on the Opportunities Daybook and will display the Opportunity form in add mode with no details other than the Prospect Account Code, allowing the user to enter a new prospect. It may also be activated from the Opportunities tab on the Add Transaction form, in which case the Prospect form will be displayed in add mode with no details completed.*  ***Edit:*** *This will be activated from the Edit button on the Opportunities tab of a Prospect record form or on the Opportunities Daybook and will display the Opportunity form in edit mode for the selected Opportunity in the list on the current tab, allowing the user to edit the details of the selected Opportunity.*  ***Delete:*** *This will be activated from the Delete button on the Opportunities tab of a Prospect record form or on the Opportunities Daybook. A message will be displayed asking the user to confirm that they wish to delete the selected Opportunity. On confirmation, the Opportunity record will be deleted from the database along with all of its detail lines.*  ***Copy:*** *This will be activated either from the Copy button on the Opportunities daybook or the Copy button on the Prospect record (Ledger tab). It will copy the selected Opportunity to a new Opportunity.*  ***Convert:*** *This will be activated from the Convert button on the Opportunities tab of a Prospect record form or on the Opportunities Daybook and will display a dialog allowing the user to select whether the selected Opportunity should be converted to a quote, order or invoice. Once the selection has been made and confirmed, the process will trigger the Convert process for the Prospect associated with the selected Opportunity. The only difference in the outcome of the process will be that the selected Opportunity will be converted to the transaction type specified by the user whilst all other Opportunities for the Prospect will be converted to quotes. (Further to MH’s inital review, further discussion of the Convert process has been moved to the Open Issues section below.)*  ***Find:*** *This will be activated from the Find button on the Opportunities tab of a Prospect record form or on the Opportunities Daybook and will display the ObjectFind dialog with only the appropriate Opportunities tab visible. The search options will be OurRef, YourRef, & AltRef (LongYrRef);*  ***Print:*** *This will be activated from the Print button on the Opportunities tab of a Prospect record form or on the Opportunities Daybook and will display the Print dialog for the Opportunity with the default details from the Prospect record and the default form from Form Definition Sets, the user will be able to print the Opportunity to Printer, Fax or Email only.*  ***Opportunity Detail: Add; Edit; Delete;***  ***Add:*** *This will be activated from the Add button on the Data Entry tab of an Opportunity form and will display an Opportunity Detail form in Add Mode.*  ***Edit:*** *This will be activated from the Edit button on the Data Entry tab of an Opportunity form or by double-clicking a detail line and will display the selected detail line in an Opportunity Detail form in Edit Mode.*  ***Delete:*** *This will be activated from the Delete button on the Data Entry tab of an Opportunity form and will display a dialog asking the user to confirm that they wish to delete the selected detail line. On confirmation, the line will be deleted.*  ***Stage: Add; Edit; Delete; Insert; Move up; Move Down.***  ***Add:*** *This will be activated by the Add button on the Stage List form and will display a Stage Detail form in Add Mode.*  ***Edit:*** *This will be activated by the Edit button on the Stage List form (or by double-clicking a item in the list and will display the selected Stage record in a Stage Detail form in Edit Mode.*  ***Delete:*** *This will be activated by the Delete button on the Stage List form and will delete the selected stage. If the stage is currently being used by a Prospect or Opportunity then it will not be possible to delete it.*  ***Insert:*** *This will be activated by the Insert button on the Stage List form and will display a Stage Detail form in Add Mode. Upon storing the Stage it will be added to the list in the position directly above the selected Stage. The position in the list is governed by the psListPos field in the Stage record.*  ***Move up:*** *This will be activated by the Move up button on the Stage List form and will move the selected stage up in the list by one position.*  ***Move down:*** *This will be activated by the Move down button on the Stage List form and will move the selected stage down the list by one position.*  *Business objects will be created for Prospect, Opportunity, Opportunity Detail and Stage. (An object will also be created for Contact, but this object will only be available via the Prospect object.) These objects will encapsulate the data fields of each table as properties and will implement a common set of methods: Add; Update; Delete; Save; Cancel; GetFirst; GetNext; GetLast; GetPrevious; GetEqual; GetGreaterThan; GetLessThan; GetGreaterThanOrEqual; GetLessThanOrEqual. The business objects will use the data access objects referred to above to access the database.* Interactions *In the following areas, access functionality will need to be modified to allow Prospects to be recognised in the same way Customers & Suppliers currently are: Discounts, Multi-Buy Discounts. This will involve defining extra values for TMultiBuyDiscount.mbdOwnerType in MultiBuy.Dat (currently only allows ‘C’, ‘S’ & ‘T’) and for MiscRec.SubType in ExStkChk.Dat to extend the range of CustDiscRec and QtyDiscRec variants (‘C’ & ‘D’ Prefixes.)*  *In the following areas, access functionality will need to be modified to allow Prospects and Prospect Opportunities to be recognised in the same way Customers/Suppliers and Transactions currently are: Notes, Links. This will involve defining extra values for PasswordRec.SubType in ExchqChk.Dat (Notes – Prefix ‘N’) and for MiscRec.SubType in ExStkChk.Dat (Letters/Links - Prefix ‘W’)*  *The following existing forms will need to be modified to allow their use with Prospects:*  ***Trader List*** *– Prospect tabs will be added for Sales Prospects & Purchase Prospects. These tabs will each display a list (using a TMultiList component) of the appropriate prospects. Details of the fields that should be shown in the list are on Page 9 of v0.7 of the Prospect Manager PRD.*  ***Add Transaction Form*** *– Radio buttons for ‘Sales Opportunity’ and ‘Purchase Opportunity’, allowing the user to display the Opportunity form in add mode, will be added at the bottom of the existing Sales/Purchase tabs, separated from the standard transaction buttons by a horizontal line.*  ***ObjectFind Dialog*** *– new options will be added for Prospects and Prospect Opportunities. New tabs and menu items will be added for Prospect, Purchase Prospect and Opportunity and appropriate search functionality will be added. (Search options will be as described for the individual Find buttons above.)*  ***ObjectPriceLookup Dialog*** *– will be amended to work with Prospects and Prospect Opportunities. This dialog currently accepts a Customer or Supplier code to find prices for a stock item. It will be necessary for it also to accept a Sales or Purchase Prospect code.*  ***ObjectDrill***  *– will be amended to work with Prospect Opportunities. In effect, the only change required here will be to display the Prospect for the Opportunity, where the Customer/Supplier would be displayed on a Sales/Purchase transaction.*  ***ObjectCreditController*** *– This will be amended to work with Prospects. This dialog currently accepts a Customer or Supplier code. It will be necessary for it also to accept a Sales or Purchase Prospect code. Details of the data required to be shown in the dialog are on Page 10 of v0.7 of the Prospect Manager PRD.*  ***System Setup*** *– The Custom Fields tab of the System Setup form will be amended to add the user defined fields for Prospects, Opportunities and Opportunity Details.*  ***Customisation*** *– New hook points will be added for validation of the User Defined Fields on the Prospect, Opportunity and Opportunity Detail forms.*  ***Telesales Form*** *– This will be amended to allow its use with Prospects as well as with customers. A radio group will be added before the A/C field in the tab order with the options Customer and Prospect. It will default to having Customer selected. Should Prospect be selected then the pick list shown on exit from the A/C field will list Prospects and it will only be possible to create an Opportunity from the wizard. The radio buttons to the right of the ‘Generate’ caption will be made invisible and a caption of ‘Opportunity’ will be shown in their place. (Selecting a different option in the radio group after an account code has been entered will clear the account code field.)*  *New functionality will need to be added to enable the creation of an Opportunity in place of a standard transaction when required.* Open Issues Although business objects and data access objects will be used to insulate the UI from the database, it will be necessary, in order to achieve acceptable performance in an MS-SQL install, for any forms which have lists to use TMultiList components which access the database directly. Consequently a working SQL Multilist component is a necessary condition for completing this requirement.  It was suggested above that we should convert from a Prospect to a Customer/Supplier in a database transaction, allowing records to be rolled back in the case of problems. However, this may have some implications that need to be explored.  First, it is not clear how many records (and of what size) can be held in a transaction. In Pervasive, I have seen suggestions that contents of a transaction are limited by the size of the engine cache. I could find no mention of a specific size for MS-SQL transactions, but I would assume that there is a limit.  The main issue with size is that for any prospect we have an indeterminate number of ancillary records which need to be modified: 0..n Opportunity Headers, each containing 0..n lines: for each of these, a Transaction Header and the associated Transaction Lines will need to be created and the existing Opportunity & Line records deleted as well as new Note records added to the Transaction Header and Customer/Supplier Record. Each of the following records associated with the Prospect will also have to be updated with the new Account Code: Notes, Discounts, Multi-Buy Discounts and Links.  Second, there is the problem that records in a transaction are locked until the transaction is committed or rolled back (assuming record level locking – if locking is page-level or table-level then there will be further problems.) With most of the records, locking is not an issue as Opportunities, Notes, etc are specific to the Prospect being converted and it is unlikely that other users will be trying to update these records when the Prospect is in the process of being converted. However, there may be an issue when adding Invoices, Orders or Quotes from the Prospect’s Opportunity headers, since this will involve locking Document Number records, preventing any transactions of the same type being added during the conversion process.  The alternative to executing the convert within a database transaction would be to write a custom rollback function which would keep track of all changes made and allow them to be reversed out if necessary.  Note: Our understanding of Pervasive.SQL database transactions is that each table updated in a database transaction is added into a ‘System Transaction’ within the DB Engine preventing any other users from accessing that table until the database transaction is completed. ‘System Transactions’ will also be completed if the number of updates exceeds the Operation Bundle Limit (default = 65,535 updates per file) or passes the Initiation Time Limit (default = 10 seconds). This behaviour would be likely to cause problems in a multi-user system with other users being stalled.  Given the above constraints, it will be necessary to write custom rollback functionality for the conversion. It will be necessary to create a low-level design for this before it is possible to give any estimates. A low-level design for this process is given in [Appendix 4](#_Appendix_4_–). | | | |

### Requirement < 6.3.PRO.43 > – Technical design

| Requirement Name and Number | | Toolkits – Prospect Support | 6.3.PRO.43 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Overview *This details the changes it will be necessary to make to the Exchequer COM and DLL Toolkits to support the Prospect Manager module.* ImplementationTechnologies *Toolkit functionality will be implemented in Delphi 6. DLL Toolkit Record declarations will be written in Delphi 6 and Visual Basic 6. Test/Demonstration programs will be written in Delphi 6 and Visual Basic 6.* Business Logic Changes ***COM Toolkit.***  *The following new interfaces which expose the public fields for the new records and implement the standard toolkit database functions will be added to access the new tables:*  *IProspect; IOpportunity; IOpportunityLines; IOpportunityLine.*  *IProspect and IOpportunity will also have Delete and Print methods.*  *IProspect : IDespatch*  *Property prPrimaryContact : IContact*  *Property prDecisionMaker : IContact*  *Property prInfluencer1 : IContact*  *Property prInfluencer2 : IContact*  *IOpportunity : IDespatch*  *Property opLines : IOpportunityLines*  *IOpportunityLines : IDespatch*  *Property opLine [Index : Integer] : IOpportunityLine*  *These interfaces will be accessed through an IProspectManager interface:*  *IProspectManager : IDespatch;*  *Property SalesProspect : IProspect;*  *Property PurchaseProspect : IProspect;*  *Property SalesOpportunity : IOpportunity;*  *Property PurchaseOpportunity : IOpportunity;*  *Property SalesProspectStage : IStage;*  *Property PurchaseProspectStage : IStage;*  *Property SalesOpportunityStage : IStage;*  *Property PurchaseOpportunityStage : IStage;*  *A new IToolkit4 interface will be added, descending from IToolkit3. This will have a ProspectManager property giving access to the IProspectManager interface.*  ***DLL Toolkit.***  *The following functions will be added to allow access to the new prospect-related tables:*  *Ex\_GetProspect; Ex\_StoreProspect; Ex\_GetOpportunity; Ex\_StoreOpportunity; Ex\_GetStage; Ex\_StoreStage;*  *Toolkit versions of each of the new record types containing all user available fields will be developed with VB-compatible record alignment. These record structures will not be written until the Exchequer structures for the new records (see* [*Appendix 3*](#_Appendix_3_–)*) have been finalised and approved.*  *The new functionality in both toolkits should be implemented, as far as is possible, using the business and data access objects detailed in* [*6.3.PRO.42/6.3.PRO.47*](#_Requirement_<_IBIF_EXL_RQ01>_1) *above.* Open Issues It may be useful to write one or more test/demonstration programs in a .NET language.  It would be useful to develop a suite of automated test programs to cover the following areas:   1. Storing a Prospect record: Check that the correct data has been added to the database. 2. Storing an Opportunity record: Check that the correct data has been added to the database. 3. Retrieving a Prospect record: Check that the correct data has been returned. 4. Retrieving an Opportunity record: Check that the correct data has been returned. 5. Converting a Prospect to a Customer/Supplier: check that the correct data has been added to the database.   In all cases, it would be best if the data was checked by going directly to the database, rather than accessing the database through the toolkits. | | | |

### Requirement < 6.3.PRO.44 > – Technical design

| Requirement Name and Number | | Importer – Prospects Support | 6.3.PRO.44 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Overview *The importer will be amended to allow Prospects and Opportunities to be imported.* ImplementationTechnologies *This will be written using Delphi 6.* Data changes *The following new record types will need to be added in Importer.dat:*  *SP=Sales Prospect*  *PP=Purchase Prospect*  *OH=Opportunity Header*  *OL=Opportunity Line*  *For each record type, all writeable fields must be added to Importer.dat.*  *Note: Importer.dat is a text file and additions can be made using any text editor. For upgrades, .upd files are created which are in the same format as Importer.dat. These are installed to the Import folder in an Exchequer install and are then used by Importer to update the user’s Importer.dat with the new records and fields without overwriting any of the user’s existing settings.* Business Logic Changes *New functions will need to be added to the TImportToolkit.pas to allow the importing of the new records. The only validation needed in the importer itself will be the functionality to validate that data fields are of the right type since, as is the case with existing functions in the class, these new functions will map onto the Dll Toolkit EX\_StoreXXX functions to validate the imported records and store them to the database. This has the implication that the dll toolkit functionality for Prospect Manager (*[*6.3.Pro.43*](#_Requirement_<_6.3.PRO.43)*) must have been completed before the functional changes to Importer can be implemented.* Open Issues | | | |

### Requirement < 6.3.PRO.45 > – Technical design

| Requirement Name and Number | | Data Dictionary - Prospect Support | 6.3.PRO.45 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Overview *The Data Dictionary is a table of fields that can be used in Exchequer Forms and the Visual Report Writer by the users.* ImplementationData changes *The Available Files list against each Data Dictionary Field will be extended to support Customer Prospects, Supplier Prospects, Sales Opportunities and Purchase Opportunities.*  *All viewable fields from the Prospect, Opportunity, Opportunity Detail and Stage tables will be added to the Data Dictionary using the Data Dictionary Editor. The following unique prefixes will be used to identify each category:-*  *PR Prospect fields*  *OH Opportunity Header fields*  *OL Opportunity Line fields*  *SG Stage fields*  *The new fields added to the Document Record for Quotes (Stage and Opportunity Rating) will be added to the Data Dictionary.*  *All new data dictionary fields will be tagged as ‘Not IAO’ to prevent them showing for future versions of IRIS Accounts Office.* Business Logic Changes *The Data Dictionary Edition in Entrprse\Dictionary will be extended to support the four new Available files.*  *In Entrprse \FormDes2\DicLinkU.pas the function Link\_Dict will be amended to deal with the new tables and fields. A new function will be added to the unit for each table in order to return the values from the fields of that table.*  *In Entrprse \FormDes2\DicLnk2U.Pas the function LinkOtherDB will be amended to perform relational links between the existing and new tables.*  *The popup Field Selection dialog in Entrprse\FormDes2\SelFld2.Pas will be amended to list the new tables and fields if the Prospect Manager module is enabled.*  *In order for Visual Report Writer to accommodate the new fields, the following changes will need to be made:*  *Add the new tables to the DataFileL array in VarCnst3.pas*  *Add the new tables to the functions File\_BtrvDrv and File\_CheckKey in RepObjCU.pas*  *Add the new tables to the procedure FileTxLate in DicLinkU.pas* Open Issues | | | |

### Requirement < 6.3.PRO.46 > – Technical design

| Requirement Name and Number | | Prospects – Form Designer Support | 6.3.PRO.46 |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Overview *The Form Designer is an end-user tool for visually designing forms to be printed, it will be extended to support Prospects and Opportunity Transactions. There is also a manual design mode called PCC Forms which work at a lower level using printer control codes which is typically used for multi-part stationary and other forms with can’t be used with modern printers.* ImplementationInteractions *Changes to the COM Toolkit and Form Printing Toolkit will be required to provide similar printing support to programmers.* Business Logic Changes  1. *Form Definition Sets will need to be extended to allow the user to define default forms for Prospects and Opportunity transactions, the following numbers will be used:-*   *65 Customer Prospect Account Details*  *66 Sales Opportunity*  *67 Supplier Prospect Account Details*  *68 Purchase Opportunity*   1. *Two new Form Printing Modes will be added into GlobType.Pas:-*   *fmProspectDetails 31 Prospect Account Details*  *fmOpportunity 32 Opportunity Transaction*   1. *The EFX Form printing routines will be extended to support the new modes. PrintBatch\_Print (PrintU.Pas) will be extended to generate an intelligent attachment name for email attachments. The AdjustTable, Batch\_GetModeKey, Batch\_Print, Include and LinkCust printing routines in PrnBatch.Inc will be extended to support the new modes.* 2. *PCC Form Printing routines in PrintPCC.Pas will be extended in a similar manner to EFX Form Printing.*  Open Issues | | | |

### Requirement < 6.3.PRO.42c > – Technical design

| Requirement Name and Number | | Prospects – Licencing | 6.3.PRO.42c |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Overview *A new Module licence will be added into the Exchequer licensing systems to control access to Prospect Manager.* ImplementationData changes *The CD Licence file and Exchequer Licence file structures will be extended to include a new Prospect Manager System Release Code.*  *Note: This will leave only 5 slots available for future modules.* Business Logic Changes *The following Exchequer related components will need to be changed:-*  *CD Licence Generator*  *The Optional Modules dialog, the Confirm Details dialogs and the human readable section of the Licence File will be extended to include the Prospect Manager module.*  *Licence Viewer*  *The IRIS Exchequer CD Licence Viewer will be extended to display the Prospect Manager module licence status.*  *CD Auto-Run*  *The Read Exchequer CD Licence dialog in the Installation CD Auto-Run application will be extended to display the Prospect Manager module licence status.*  *Installer*  *The CD Licence reading routines in Setup.Dll will be extended to include the Prospect Manager module, the Exchequer Licence writing routines in SQLHelpr.Exe will be modified to apply the Prospect Manager licence, the Install/Upgrade/Add Company processes will be extended to copy the Prospect Manager files if licensed and the Optional Modules dialog in the Installation Wizard will be extended to support Prospect Manager.*  *Multi-Company Manager*  *The Module Licenses and User Counts dialog will be extended to support the Prospect Manager module licence.*  *SecRel*  *The SecRel Security Release Code Utility will be extended to support the Prospect Manager module licence.*  *WebRel*  *The WebRel Release Code web-site will be extended to support the Prospect Manager module licence.*  *VwDat*  *The internal VwDat, Exchequer Licence Viewer Utility, will be extended to support the Prospect Manager module licence.*  *SDK*  *The ex\_CheckModuleReleaseCode function in the Toolkit DLL will be updated to support all modules. The SystemSetup.ssReleaseCodes object in the COM Toolkit will be updated to support all modules.* Open Issues | | | |

### Requirement < 6.3.PRO.42d > – Technical design

| Requirement Name and Number | | Prospects – User Permissions | 6.3.PRO.42d |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Overview *New user permissions will need to be added for the Prospect Manager module. This will involve changes both to Enter1 and to the library GeUpgrde.dll, which is used by the Exchequer installer to add new permission records to the password table (ExchQChk.dat) when upgrading an installation.* ImplementationTechnologies *This will be written in Delphi 6.* Business Logic Changes *The following permissions will be added to both Sales and Purchase groups under a Sub-Group labelled ‘Prospects’:*  *Access to Prospects*  *Add a Prospect*  *Edit a Prospect*  *Delete a Prospect*  *Print a Prospect*  *Convert a Prospect*  *Access to Notes*  *Access to Links*  *Access to Discounts*  *Access to MultiBuy Discounts*  *Access to Opportunities Ledger*  *Access to Opportunities Daybook*  *Create an Opportunity*  *Edit an Opportunity*  *Delete an Opportunity*  *Print an Opportunity*  *For GeUpGrde.dll, these permissions will need to be added to the Permissions array in Entrprse\MultComp\Upgrades\PwUpgrde.pas. For Enter1 they will need to be added to the files Entrprse\R&D\PWary.pas and Entrprse\R&D\UA\_Const.pas.* Open Issues Since converting a Prospect to a Customer/Supplier will also convert any Opportunities for that Prospect to quotations and vice-versa, no separate permission should need to be added for Converting Opportunities.  The permissions Edit, Delete & Print an Opportunity, and Access to Opportunities Daybook were not included in the PRD, but their addition is consistent with the way Exchequer usually works. | | | |

### Requirement < 6.3.PRO.47f > – Technical design

| Requirement Name and Number | | Prospects – Workflow Diary | 6.3.PRO.47f |
| --- | --- | --- | --- |
| Requirement Type | Functional | | |
| Overview *The workflow diary will be amended to allow notes to include a time stamp and an alarm time.* ImplementationTechnologies *This will be written in Delphi 6.* Data changes *The NotesType sub-record in ExchQChk.Dat will be extended to allow details of time to be stored along with the Date and the AlarmDate.* Business Logic Changes *The Note Entry form will be amended to allow the user to enter a time and an alarm time for a dated note. (The time will be in hours and minutes only, using the current system time format.) The dates are currently entered using a TEditDate component which does not currently support times. This situation will need to be changed to support the entry of times as well as dates. (See Open Issues below.)*  *The WorkFlow Diary form will be amended as follows:*  *The Date column will include the time from the record as well as the date.*  *A Filter button will be added. Clicking this will display a dialog containing a list of Exchequer users along with start and end date fields allowing the user to limit the notes shown in the diary to those for a specific user and within a set date range.* Open Issues *On the Note Entry form, dates are currently* *entered using a TEditDate component which does not currently support times. In order to allow the entry of times, while maintaining consistency with the rest of Exchequer we have two options:*   1. Add separate controls to the form for displaying and editing the time and alarm time. This would be the easier option but might result in a more cluttered form. 2. Create a descendant of the TEditDate component which allows times to be entered. This would give a more elegant UI but would require investigation into the workings of TEditDate which would necessarily take longer. | | | |

# Appendix 2 - Test Approach

# Test Plan Outline

This document outlines the testing strategy and approach to be taken regarding the functionality of PROJECT A Release n.nn developed by the Iris Software and Services development team. Add any relevant information as an introduction.

# References

The following list reference documents that support this section of the HLD.

|  |  |  |
| --- | --- | --- |
| Ref | Document(s) | Version |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |

# Test Items (Functions)

The following test items are planned to be tested in PROJECT NAME: [delete or add to the list as appropriate]

* Functional testing of new features – new features will be tested to validate that they have been implemented successfully and meet their requirements
* Installation and upgrade testing – testing of the product installation/un-installation and the ability to upgrade from previous versions. The direct upgrade ability of the two previous versions of ILB (i.e. release 1.9 and 1.10) is supported and will be tested.
* Defect Fixes – Validation of the implementation of defect fixes. Defect fixes are performed by the development team on a priority basis.
* Regression testing – perform regression testing to validate that existing functions are not adversely affected by changes made. Regression testing will primarily be focused at areas that have undergone change, either by new functionality.

Automated regression testing will also be performed during PROJECT NAME.

# Features to be Tested

The following features are planned to be delivered in PROJECT NAME and will be tested. Please refer to the relevant design/requirement specifications for further details of the feature.

Each feature/requirement has been designated a requirement id as this is used in various testing tools.

|  |  |
| --- | --- |
| Req Id | Feature |
| 0001 |  |
| 0002 |  |
| 0003 |  |
| 0004 |  |
| 0005 |  |
| 0006 |  |
| 0007 |  |
| 0008 |  |
| 0009 |  |

Additional testing will be performed in the following areas: : [delete or add to the list as appropriate]

* Installation Testing – validation that the software components install successfully on supported systems and configurations
* Uninstall Testing – validation that un-installation cleanly removes installed files, system settings, registry settings etc
* Upgrade Testing – validation that upgrades from supported existing versions are performed correctly
* Interoperability Testing – Verifying that the product interacts correctly with other products (MS Office, SQL server etc)

# Features not to be Tested

For example:

Testing of the functionality of all new features is planned.

Non-functional testing of the new functionality is not planned to be performed by the ….

Testing via RDP or Citrix connections is not planned for this release.

# Approach (Strategy)

## Pass Strategy

[delete or add to the list as appropriate]

PROJECT NAME is being delivered in n iterations. Each iteration will include a sub-set of the new functionality to be developed

Testing will be split into n cycles to reflect the n iterations into test. Each test cycle will contain a single testing pass (the set of tests assigned to that cycle is planned to be executed once).

Some tests that are executed in cycles one and two will be repeated in subsequent cycles to verify that functionality has not been adversely affected by subsequent functionality additions or defect fixes.

Testing within the A4-A6 gates (known as the ‘verify’ stage within the PROJECT NAME project plan) will consist primarily of defect fix verification, new functionality testing re-tests, installation/upgrade testing and system regression testing. This period will also allow some contingency to perform new functionality testing if the feature could not be fully tested during the earlier cycles.

## Testing Tools

### Test Development Tools

It is planned that all new functionality testing will utilise manual testing techniques.

Regression testing will utilise both manual and automated techniques.

Test scenarios and cases will be developed initially using Excel spreadsheets.

### Test Execution Tools

Test execution recording will initially be performed using Excel spreadsheets

Other tools will be used as required. These may include a screen capture tool to record tester transactions (to assist with defect reproduction).

## Testing Metrics

Testing metrics will be collected during test development and test execution. These metrics will be used to measure the effectiveness of the testing effort and to help improve the testing process.

The ease and effectiveness of metrics collation will be improved by the adoption of a test management and defect tracking tool (Test Link and Jira respectively).

The collection of other metrics may be included as required.

### Metrics Collected During Test Development

The following metrics will be collected during test development:

|  |  |  |
| --- | --- | --- |
| **Metric** | **Description** | **Formula** |
| Test Specification Progress | Comparison of the actual time spent preparing test scenarios, conditions and scripts against the estimated time | Actual v estimated |
| Scripts Passed Review First Time Rate | The relative proportion of test scripts that are successfully peer reviewed at the first attempt.  The rate of first time review passes is an indicator of the quality of the test preparation activity and associated deliverables.  A lower than expected pass rate could be due to poor test preparation quality, and is likely to lead to increased costs and extended timescales. | Scripts Passed Review First Time / Scripts Reviewed First Time |

### Metrics Collected During Test Execution

The following metrics will be collected during test execution:

|  |  |  |
| --- | --- | --- |
| **Metric** | **Description** | **Formula** |
| Test Execution Progress | Comparison of the actual tests executed compared to the plan.  Provides an indication of capability to meet deadlines  Significant variance (i.e. +/- 5%) should be accompanied by an explanatory comment. | Actuals v Plan |
| Test Environment Availability | Availability of test environments expressed as a percentage of test execution time.  Provides an indication of the ability to progress testing.  Significant outages should be accompanied by an explanatory comment. | Percentage of environment availability during working hours |
| Defect Progress | Comparison of the rate at which defects are being raised compared to the rate at which defects are being closed.  If defects are being raised at a rate significantly greater than the rate at which they are being closed, it is an indicator that testing schedules may have to be extended in order to achieve test exit criteria | Defects raised v Defects closed |
| Defect Age | Analysis of the time that outstanding defects have been open, by severity.  Defects outstanding for long periods are an indicator that testing schedules may have to be extended in order to achieve test exit criteria (e.g. no outstanding Severity 1 defects). | Time outstanding, by defect severity |
| Tests Passed First Time Rate | The relative proportion of tests that are successfully executed at the first attempt.  The rate of first time test passes is an indicator of product quality.  A lower than expected pass rate could be due to poor product quality, and is likely to lead to increased costs and extended timescales. | Tests Passed First Time / Tests Executed First Time |

## Configuration Management

Configuration Management is necessary at different areas of the testing project. The correct versions of software, tests, data and code must be available throughout the testing life cycle to ensure the validity of the testing effort.

|  |  |
| --- | --- |
| **Test Area** | **Configuration requirements** |
| Test Preparation | Test cases will be versioned with data relating to the source documentation used to create the test case being recorded. Test author, date and associated details will be recorded. |
| Test Environment | The setup of the test environment will be recorded during test execution. This will include things such as the installed operating system, database, applications and configuration as well as the version of the software under test. |
| Test Execution | Each execution of a test case will record the necessary details such as:- test case version, environment setup and configuration, tester + date/time, defects raised etc. |

## Test Configurations

PROJECT NAME is required to work on a number of different operating systems and setups.

Testing will be performed on a number of these configurations to verify correct operation.

The test environment is not currently able to support x64 windows versions.

Therefore, the majority of functional testing during iterations 1-3 will be performed on the following configuration, with a single server machine been accessed by XP and Vista clients:

|  |  |  |
| --- | --- | --- |
| **Server** | Win 2K3 x86 | SQL 2005 |
| **Client** | Windows XP x86 | Office 2003 |
| **Client** | Windows Vista x86 | Office 2007 |

During the verification testing phase, further test configurations will be utilised and testing will be spread across each configuration. A new test server capable of hosting 64-bit windows versions should have been introduced and testing to a remote SQL instance will also be included (ie SQL installed on a different server to the ILB instance).

The tables below show the server and client configurations that will be utilised during PROJECT NAME verification testing:

|  |  |
| --- | --- |
| **Server** | |
| Windows 2K3 x86 | SQL 2005 |
| Windows 2K3 x64 | SQL 2008 |
| Windows 2K8 x86 | SQL 2005 |
| Windows 2K8 x64 | SQL 2008 |

|  |  |
| --- | --- |
| **Client** | |
| Windows XP x86 | Office 2K3 |
| Windows XP x64 | Office 2K7 |
| Windows Vista x86 | Office 2K3 |
| Windows Vista x64 | Office 2K7 |

## 

## Regression Testing

### Automated Regression Testing

Automated testing is planned to be included during PROJECT NAME.

The automated solution has been planned to be setup and executing the automated tests in Knutsford by mid-July. From this point, the automated tests will be run whenever a new build is taken by the test team (i.e. when the iteration releases are taken and when fix releases are taken during verification).

### Manual Regression Testing

Regression testing will be performed on multiple configurations of the supported platforms and applications (please refer to section 8.5).

Regression testing will be performed by using a mixture of pre-planned test cases and exploratory testing techniques. The exploratory testing will be documented so that a record of the testing completed is maintained and to assist in the creation of future test cases.

## Status Reporting

A status report will be made available to project stakeholders following each cycle of testing. This report will include details of the relevant metrics mentioned above as well as comment on the progress of the testing effort.

Daily reports will be made during test execution to the Test Manager detailing daily execution and defect status.

# Item Pass/Fail Criteria

Testing will have been considered as completed successfully when the following criteria have all been met:

* All planned test cases have been executed and all associated details (application version, test environment, execution date, tester name, result) have been recorded against them
  + Any tests that have not been executed will require dispensation from the appropriate parties
* High severity defects, if any, are all closed, with documentation of the steps taken to correct them and of the steps taken to test the correction.
* Lower severity defects should also be closed, as above. However, any that do not present an unacceptable risk may remain outstanding if dispensation has been agreed by the relevant parties.

# Test Deliverables

The following table lists the test items that will be delivered as part of this plan.

|  |  |
| --- | --- |
| **Deliverable** | **Details** |
| Test Approach | In this document |
| Testing Schedule | MS Project document detailing the tasks, assignees and timescales of the testing effort for PROJECT NAME. |
| Test Cases | Excel spreadsheets containing the test cases produced for PROJECT NAME (test cases may also be produced within Test Link if it is available during PROJECT NAME) |
| Test Results | Excel spreadsheets containing the test execution results for the PROJECT NAME testing (results may also be produced within Test Link if it is available during PROJECT NAME) |
| Status Reports | A status report will be generated at the end of each of the three test cycles |
| Test Completion Report | A test completion report will be generated at the end of the testing project that will summarise the testing performed, the results, the closed and outstanding defects etc. This report should also include a lessons learnt section to help improve the quality of future projects. |

# Test Environment

PROJECT NAME supports a number of operating systems and applications (please refer to section 8.5 for the platforms and applications that will be tested against).

Testing will be performed such that a proportional amount of testing is performed on each operating system / Office version to match the usage within the customer base.

Testing will utilise a Virtual Server environment. Testing may also be executed on local and virtual machines if appropriate. The Virtual Server environment allows greater flexibility in testing over different operating systems and configurations as well as getting more benefit from minimal hardware resources.

# Staffing and Training Needs

The PROJECT NAME test team received product training prior to the start of the project and no specific additional training has been identified as being required. Domain experts may be required to be consulted as needs arise. Additional training may be required for the ILS if testing for that product is required in Knutsford (this will be covered in a separate test plan).

No specific training on testing tools has been identified. The test automation tool ‘Test Complete’ is planned to be used for the first time during PROJECT NAME and the required knowledge of the tool will be sought via online resources and consultation with the RAVE team if required. Knowledge transfer within the team and wider community will be used on additional testing tools used by the team.

# Roles and Responsibilities

The following have been identified as the main roles associated with the PROJECT NAME testing project:

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Responsibilities** |
| ?? | Quality Engineering Manager | * Stakeholder management * Assist test lead * Review and Sign-off testing deliverables * Resource Management * Partake in Go/No-Go decisions |
| ?? | Test Team Lead | * Test planning and management * Produce the testing strategy/approach * Ensure all required elements are in place * Review design specifications * Review testing deliverables * Create and maintain testing schedule * Produce status/progress reports * Produce test closure report * Produce test scenarios/conditions/cases and data * Perform test execution * Define and monitor the testing environment * Partake in Go/No-Go decisions |
| ?? | Tester | * Review design specifications * Peer review testing deliverables * Produce test scenarios/conditions/cases and test data * Perform test execution * Assist in the maintenance of the testing environment |
| ?? | Tester | * Review design specifications * Peer review testing deliverables * Produce test scenarios/conditions/cases and test data * Perform test execution * Assist in the maintenance of the testing environment |
| ?? | Development Manager | * Review testing deliverables * Respond to testing queries * Partake in Go/No-Go decisions |
| ?? | Development Lead | * Review testing deliverables * Respond to testing queries |
| ?? | Product Manager | * Review testing deliverables * Respond to testing queries |
| ?? | Project Manager | * Review testing deliverables * Maintain the overall project plan * Inform test team of project changes/delays that may affect the testing schedule |

# Schedule

The major testing milestones are shown below:

|  |  |  |
| --- | --- | --- |
| Milestone | Comment | Date |
| Start of Test Preparation |  | Date |
| End of Test Preparation | Test preparation will continue during iteration 1 and iteration 2 test execution | Date |
| Start of Iteration 1 Execution | Iteration 1 functionality testing | Date |
| End of Iteration 1 Execution |  | Date |
| Start of Iteration 2 Execution | Iteration 2 functionality testing | Date |
| End of Iteration 2 Execution |  | Date |
| Start of Iteration 3 Execution | Iteration 3 functionality testing | Date |
| End of Iteration 3 Execution |  | Date |
| Start of post-functionality testing | Installation/upgrade testing  Release testing etc | Date |
| End of post-functionality testing |  | Date |
| PROJECT NAME Testing Complete | Test Closure Document completed | Date |
| Test maintenance operations completed | Date |

Test duration estimates have been based on estimates provided by the test team for the duration of preparation and execution activities for each of the testing requirements. In addition ‘time buckets’ have been estimated for the testing of defect fixes, execution of regression tests and performing installation testing.

# Planning Risks, Contingencies and Assumptions

The following risks and related contingencies have been recognised for the PROJECT NAME testing project.

|  |  |  |
| --- | --- | --- |
| **#** | **Risk** | **Contingency** |
| 1 |  |  |
| 2 |  |  |

The following assumptions have been made regarding the testing of PROJECT NAME.

|  |  |
| --- | --- |
| **#** | **Assumption** |
| 1 | Development, training and product management resources will be available to support queries that the test team may have |
| 2 | Development will deliver software iterations to the timescales detailed in the project plan |
| 3 | The defect management system utilised for PROJECT NAME will be available to both testing and development teams |
| 4 | No further requirements will be included in the PROJECT NAME release |

## Appendix 3 – New Data Records

The following describes the record structures and indexes for the new tables:

TDocumentDeliveryMode = (dmPrinted, dmFax, dmEmail); //enumeration defaults to 0, 1, 2

TEmailCompressionMode = (ecNone, ecZip, ecEDZ);

Prospect Table

TProspectRecord = Record

prAccountType : Char; // Customer / Supplier Flag

prCode : string[10]; // Prospect account code

prCompany : string[45]; // Company Name

prType : String[4]; // Free Type Sort Field

prRemitCode : String[10]; // Account Code of Remit Account

prVATRegNo : String[30]; // VAT Registration No.

prAddress : AddrTyp; // Addr1-5

prDeliveryAddress : AddrTyp; // Despatch Addr1-5

prUseDeliveryAddress : Boolean; // Separate Despatch Address

prContact : String[25]; // Contact Name

prPhone : string[30]; // Phone No.

prFax : string[30]; // Phone No.

[[1]](#footnote-1)

prEmailAddress : String[100]; // Email address for Statement/ Remittance

prInvoiceDeliveryMode : TDocumentDeliveryMode; //Invoice delivery mode

prStatementDeliveryMode : TDocumentDeliveryMode; // Statement/Remittance delivery mode. 0 = Printed;1 = Fax;2 = email

prSendHTMLWithXML : Boolean; //When sending XML, send HTML

prEmailCompressionMode : TEmailCompressionMode; //Default Zip attachement 0=no,1= pkzip, 2= edz

prCurrency : Byte;

prVATCode : Char;

prPayTerms : SmallInt;

prCostCentre : String[3];

prDepartment : String[3];

prDiscount : Double;

prDiscountChar : Char;

prSettlementDays : SmallInt; // Default Settlement discount Number of Days

prTheirRefForUs : String[10]; // Our Code with them

prUseTradeTerms : Boolean; // Special Terms

prTradeTerms : TradeTermType; // 2 Lines of Terms

prLocation : String[3]; // Default Multi Loc Stock

AccStatus : Byte; // On Hold, Closed, See notes

prMobileNo : String[30]; // Second Phone No.

prSalesGLCode : LongInt; // Default Nominal Code

prCostOfSalesGLCode : LongInt; // Override COS Nominal

prControlGLCode : LongInt; // Override Default Ctrl Nominal

prUserDefinedFields : Array[1..4] of String[30];

prDefaultFormSet : Byte; // Use form def page for forms

prDefaultSettleDisc : Double; // Default Settlement Discount

prSendReader : Boolean; // On next email, send reader & reset

prPostCode : String[20]; // Separate postcode

prAltCode : String[20]; // Alternative look up code, can be blank

prEECMember : Boolean; // VAT Inclusion for EEC

prVATRegistrationCountry : String[5];

prSSDDeliveryTerms : String[5]; // " Delivery Terms

prSSDModeOfTransport : Byte;

prTagNumber : Byte; // Tag No

prStageFolio : longint;

prRating : String[20];

prActiveDate : LongDate;

prCloseDate : LongDate;

//Internal fields

prNoteLineCount : LongInt; // Note Line Count

prUserLastUpdated : String[10];

prDateLastUpdated : LongDate; // Date last updated

prTimeLastUpdated : String[6]; // Time stamp for record Change

//Contact codes

prPrimaryContact : String[20]; //

prDecisionMaker : String[20];

prInfluencer1 : String[20];

prInfluencer2 : String[20];

Spare : Array[1..709] of Char;

end;

Indexes

prCode //Unique id

prType + prCode

prType + prCompany

prType + prAltCode

prType + prStageFolio + prCode

prType + prArea + prCode

prPhone

prEmailAddress

Opportunity Header Table

TOpportunityRecord = Record

opAcCode : String[10]; // Lookup Cust Code

opOurRef : String[10]; // Doc Number

opFolioNum : Longint; // Audit No.

opCurrency : Byte; // Currency of Document

opYear : Byte;

opPeriod : Byte;

opDate : LongDate; // Doc Date

opDueDate : LongDate;

opAcType : Char; // Cust/Supplier char, to differentiate between Payments / Reciepts Due

opDeliveryAddress : AddrTyp; // Delivery Address

opVatAnalysis : Array[VATType] of Double; // Analysis of VAT

opNetValue : Double; // Net Value of Opportunity

opVat : Double; // Total VAT Content

opManualVat : Boolean; // If Set, prevents re-calclation of VAT

opLineSplit : Array[1..6] of// Store makeup of line totals

Double;

opXrate : CurrTypes; // Exchange Rates

opVariance : Double; // Currency Exchabge Loss/ Gain

opOrigRates : CurrTypes;

opOBaseEquiv : Double; {\* Pre EMU conversion base value }

opUseORate : Byte; {\* Forces the conversion routines to apply non tri rules \*}

opOldORates : CurrTypes; {\* After euro conversion, very original rates are shown\*}

opVATTriR : TriCurType; // Details of VAT Triangulation

opSettleDiscPercent: Double; // Discount Avail/Take

opSettleDiscAmount : Double; // Actual Value of Setle Discount

opSettleDiscDays : SmallInt; // No Days Disc Avail

opSettleDiscTaken : Boolean; // Discount Taken

opDiscountAmount : Double; // Discount Amount

opSSDProcess: Char; // SSD process flag

opSSDTransNat : Byte; // SSD Nature of Transaction

opSSDTransMode : Byte; // SSD Mode of Transport

opSSDDeliveryTerms : String[3]; // SSD Delivery Terms

opTotalWeight : Double; // Order Weight Details

opAltRef : String[20]; // Free format text

opYourRef : String[20]; // Customers Ref for opportunity

opUserID : String[10]; // Operators User Name

opTagNo : Byte; // Doc Marked for something

opControlGL : Longint; // Debtor/Creditor Control Nominal

opJobCode : String[10]; // Default Document Job Code

opAnalysisCode : String[10]; // Default Document Anal Code

opUserField1 : String[30];

opUserField2 : String[30]; // User def fields

opUserField3 : String[30];

opUserField4 : String[30]; // User def fields

opStageFolio : longint;

opRating : String[20];

opIsPrinted : Boolean; // This Document has been printed

opTimeChanged : String[6]; // Last time transaction changed

opTimeCreated : String[6]; // Time transaction created

opLineCount : LongInt; // Line count - internal use

opNoteCount : LongInt; // Notes Line count - internal use

Spare : Array[1..800] of Byte;

end;

Indexes

opAcCode + opTransDate + opFolioNum – for Opportunites Ledger off Prospect record

opOurRef

opYourRef

opFolioNum

opAltRef

**opTransDate + opFolioNum** – to control order within duplicate dates

opYear + opPeriod

opExpiryDate + opFolioNum – to control order within duplicate dates

opStageFolio + opFolioNum – to control order within duplicate stages

Opportunity Detail Table

TOpportunityDetailRec = Record

odFolioRef : LongInt; { Link to Doc Header }

odLineNo : LongInt; { Array / Line No}

odGLCode : LongInt; { Nominal Code }

odCurrency : Byte;

odPeriod : Byte;

odYear : Byte;

odCostCentre : String[3]; { Cost Center / Analysis }

odDepartment : String[3]; { Cost Center / Analysis }

odStockCode : String[20]; { Stock Code }

odABSLineNo : LongInt; { Absolute, non moveable line no. \*}

odAcType : Char; { Account type }

odQty : Double;

odQtyMul : Double; { Item Multiplier Factor }

odNetValue : Double; { Line Total Sans VAT }

odDiscount : Double; { Amount/% }

odDiscountType : Char; { Flag to indicate if disc is amount or % }

odVATCode : Char; { VAT Code }

odVAT : Double; { Line VAT Content }

odCostPrice : Double;

odAcCode : String[10]; { Trader Code }

odLineDate : LongDate;

odDescription : String[60]; { Line Desc }

odJobCode : String[10]; { Job Code }

odAnalysisCode : String[10]; { Job AnalCode }

odXRate : CurrTypes; { Co/VAT Rates }

odWeight : Double; { Weight of Sales Unit }

odKitLink : LongInt; { Folio No. of parent Stock code for kit line }

odLocation : String[3]; { Multi stock location code \*}

odUsePack : Boolean; { Include Qty Mul in Line Calc }

odCOSGLCode : LongInt; { COS Nominal Code to post stock to }

odOurRef : String[10]; { Parent OurRef }

odLineType : Byte; { Doc Line Type Link }

odPackPrice : Boolean; { Price is a total price }

odPackQty : Double; { Qriginal Qty in Case }

odShowCase : Boolean;

odAltStockFolio : LongInt; { Link to Alt db Code }

odUserField1 : String[30]; {\* Line user def 1 \*}

odUserField2 : String[30]; {\* Line user def 2 \*}

odUserField3 : String[30]; {\* Line user def 3 \*}

odUserField4 : String[30]; {\* Line user def 4 \*}

odInclusiveVATCode : Char; {\* Rate on line is inclusive of this rate \*}

odIncNetValue : Double; {Original inclusive figure we are aiming to achieve}

odSSDUplift : Double; {\* Intrastat uplift % \*}

odSSDCountry : String[5]; {\* " Country of origin \*}

odSSDCommodityCode : String[10]; {\* SSD Commodity code \*}

odSSDSPUnit : Double; {\* SSD unit into sales unit \*}

odSSDUseLine : Boolean; {\* Take the ssd values from the line \*}

odPriceMultiplier : Double; {\* Price rate multiplier to get price per \*}

odOBaseEquiv : Double; { Pre EMU conversion base value }

odUseORate : Byte; {\* Forces the conversion routines to apply non tri rules\*}

odCurrTriR : TriCurType; {\* Details of Main Triangulation \*}

odBinQty : Double; {Qty required to satisfy bin qty}

odDiscount2 : Double; // Multi-Buy Discount

odDiscount2Chr : Char; // #0=Amount, %=Percentage

odDiscount3 : Double; // Transaction Based Discount (TTD/VBD)

odDiscount3Chr : Char; // #0=Amount, %=Percentage

odDiscount3Type : Byte; // 0=Undefined, 1=TTD, 2=VBD, 255=Discount Info Line

odECService : Boolean;

odServiceStartDate : LongDate;

odServiceEndDate : LongDate;

Spare : Array[1..359] of Byte;

End;

Indexes

odFolioRef + odLineNo

odFolioRef + odAbsLineNo

odTransDate + odFolioRef + odLineNo

odYear + odPeriod

odAccountCode + odFolioRef + odLineNo

odStageFolio + odFolioRef + odLineNo

odStockCode + odFolioRef + odLineNo

Contact Table (based on Contacts Plug-In)

The Contact record is based on the Contacts Plug-in record and consequently includes address fields. However, in the PRD, the Contacts tab on the Prospect record has no way of entering address fields. Consequently, if it is desirable to store address fields then we will need to expand the available fields on the Contacts tab – possibly by adding a button at the side of each contact which will display a dialog allowing all fields to be entered.

TContactType = (ctPrimaryContact, ctDecisionMaker, ctInfluencer1, ctInfluencer2);

TContactRecType = Record

coAccount : String[10]; // Enterprise Parent Account Code

coCode : String[20]; // Unique Contact Code

coTitle : String[10]; // Title - 'Mr', 'Mrs', 'Miss', 'Ms',etc…

coFirstName : String[30]; // Christian Name - 'Mark', 'Kevin', etc…

coSurname : String[30];

coPosition : String[30]; // Job Title

coSalutation : String[20]; // Method Of Address - 'Jon', 'Mr Frewer'

coContactNo : String[30]; // Telephone Number

coDate : String[8]; // Added Date in YYYYMMDD format

coFaxNumber : String[30]; // Fax Number

coEmailAddr : String[100]; // Email Address

coAddress1 : String[30]; // Delivery Address #1

coAddress2 : String[30]; // Delivery Address #2

coAddress3 : String[30]; // Delivery Address #3

coAddress4 : String[30]; // Delivery Address #4

coPostCode : String[10]; // PostCode

coMobileNo : string[30]; // Mobile Phone Number

coNoteLine : string[100]; // Optional note

coContactType: TContactType; // Primary, Decision maker, Influencers 1 & 2

coSpare : Array [1..380] Of Char;

End;

Indexes as Contacts Plug-in

Stage Table

TProspectStageType = (pstSalesProspect, pstPurchaseProspect,

pstSalesOpportunity, pstPurchaseOpportunity);

TProspectStageRecord = Record

psType : TPropectStageType; //Type of object stage applies to

psFolioNo : longint;

psName : string[100]; //Name of stage

psListPos : Longint; //position in list

Spare : Array[1..512] of Char;

end;

Indexes

psType + psFolioNo

psType + psName

psType + psListPos

## Appendix 4 – Low-Level Design for Conversion Process

**The problem:**

Converting a Prospect to a Customer or Supplier involves adding new transactions to the database as well as making changes to a number of ancillary records for that Prospect - Notes, Links, Discounts, etc. It is possible that the process may not be able to update one or more of these records during the conversion, due to them being locked by another user. It is also possible that the program could crash during the conversion. In either of these cases, we would need to be able to recover and complete the conversion, in order to prevent the database being left in an incomplete state.

**The proposed solution:**

There are two main approaches to this problem: one would be to try to return the database to the position it was in before the process started, by reversing all the steps that were carried out successfully. This has a number of disadvantages, not least of which is that the reversal process itself might be interrupted.

A second approach is to add functionality which keeps any changes made and enables an interrupted conversion to be completed. This is the approach I am taking.

The solution proposed is that, once the new Account Code has been accepted by the user, the Prospect status will be set to a specific value to indicate that it is being converted. The new Account Code will be stored in a field in the Prospect record and the Prospect Record will be stored and locked. We will also store the date and time of the conversion and the User ID of the user running it. This will allow useful information to be given if a different workstation recovers an interrupted conversion.

The principle here is that if the program crashes during the process, then the status on the Prospect will still be set to 'Converting', allowing Exchequer to search for Prospects with that status on startup and complete the conversion process for any it finds which are not locked (locking indicating that the conversion process is currently being run by another workstation.)

(The status of 'Converting' will also be used to prevent users from editing or adding Opportunities or other records for a Prospect on which the conversion process has been started.)

In order to enable the conversion process to be entered either from the start or from a situation where some records have already been converted, it will be written to simply take the ProspectCode and AccountCode as parameters and then go through the following steps:

1. If the Account record does not exist then create it (with a status of On Hold.)
2. For each Opportunity containing the Prospect Code:

Start a Database Transaction

Create an appropriate Exchequer transaction from the opportunity, including all lines.

Add a note to the transaction

Delete the Opportunity and all its lines.

End the database transaction.

3. For each Notes record containing the Prospect Code:

Update the record with the Account Code and Type

4. For each Notes record containing the Opportunity Folio:

Update the record with the Transaction Folio

5. For each Discount record containing the Prospect Code:

Update the record with the Account Code and Type

6. For each MultiBuy record containing the Prospect Code:

Update the record with the Account Code and Type

7. For each Links record containing the Prospect Code:

Update the record with the Account Code and Type

8. Add the Contacts from the Prospect record to the Contacts Plug-In table and delete them from the Prospect Manager Contacts table.

9. Update the Account record status to Active

10. Delete the Prospect Record

If any step fails (eg because it is not possible to update a record) then this should be recorded. At the end of the process, if no fails have been recorded, then the Prospect record will be unlocked and deleted.

If any fails have been recorded then a dialog will be displayed to the user, informing them of that fact and giving them the option to try to finish the remaining steps of the convert at this time. If this option is refused then the Prospect will be unlocked but not deleted and the conversion process can be completed the next time Exchequer starts.

1. [↑](#footnote-ref-1)