Technical Specification

[Project Name]

|  |  |
| --- | --- |
| **Filename:** | Technical\_Spec\_Template |
| **Last Save Date:** | Friday, March 10, 2017 |
| **Author(s):** | First Last (alias) |
| **File Location:** | [\\server\folder](mailto:ITLCHelp@microsoft.com) |
| **Project Information Classification:** | HBI, MBI, LBI, PII, or HSPII [select information classification] |
|  | |

Table of Contents

[1 Introduction 8](#_Toc325640246)

[2 System Architecture 8](#_Toc325640247)

[2.1 Distributed System View 8](#_Toc325640248)

[2.1.1 Application Context Diagram 8](#_Toc325640249)

[2.1.2 Functional Components 8](#_Toc325640250)

[2.2 Deployment View 10](#_Toc325640251)

[2.2.1 Logical Server Definition 10](#_Toc325640252)

[2.3 System Quality Attributes 10](#_Toc325640253)

[2.3.1 Performance 11](#_Toc325640254)

[2.3.2 Dependability 11](#_Toc325640255)

[2.3.3 Security 12](#_Toc325640256)

[3 Application Subject Area Context Model 12](#_Toc325640257)

[*4* Maintenance Impact Analysis *(This section is best practice and is required for projects that have an Enterprise Architecture Stakeholder)* 12](#_Toc325640258)

[4.1 <Name of first feature goes here> 13](#_Toc325640259)

[4.1.1 Areas of Impact 13](#_Toc325640260)

[4.1.2 Technology Area - Web Pages / Windows Application 13](#_Toc325640261)

[4.1.3 Technology Area - Middle Tier Objects 13](#_Toc325640262)

[4.1.4 Technology Area – Database Changes 14](#_Toc325640263)

[4.2 <Name of second feature goes here> 18](#_Toc325640264)

[5 Application Detailed Design *(This section is best practice and is required for projects that have an Enterprise Architecture Stakeholder)* 18](#_Toc325640265)

[5.1 End to End Dynamic Models 19](#_Toc325640266)

[5.1.1 Example use case: GetAgreementList 19](#_Toc325640267)

[5.1.2 <Use Case Name Goes Here> 20](#_Toc325640268)

[5.2 Technology Dependencies 21](#_Toc325640269)

[5.2.1 .NET Framework Version Compatibility Issues 21](#_Toc325640270)

[5.2.2 Server software dependencies 21](#_Toc325640271)

[5.2.3 Client software dependencies 21](#_Toc325640272)

[5.3 Security Roles 21](#_Toc325640273)

[6 Front End 21](#_Toc325640274)

[6.1 Web Application Front End 22](#_Toc325640275)

[6.1.1 Physical Site Map 22](#_Toc325640276)

[6.1.2 Common UI Technical Specification 23](#_Toc325640277)

[6.1.3 Web Pages and Web Controls 23](#_Toc325640278)

[6.1.4 Localization 24](#_Toc325640279)

[6.1.5 Configuration 24](#_Toc325640280)

[6.1.6 Presentation Layer Class definitions 24](#_Toc325640281)

[6.1.7 Collaboration Layer Class definitions 24](#_Toc325640282)

[6.1.8 Error Handling 25](#_Toc325640283)

[6.1.9 Security 25](#_Toc325640284)

[6.2 Windows Application Front End 25](#_Toc325640285)

[6.2.1 Screen Navigation map 25](#_Toc325640286)

[6.2.2 Object Model 25](#_Toc325640287)

[6.2.3 Screens and User Controls 26](#_Toc325640288)

[6.2.4 Localization 26](#_Toc325640289)

[6.2.5 Common Components 26](#_Toc325640290)

[6.2.6 Configurations 26](#_Toc325640291)

[6.2.7 Error Handling 26](#_Toc325640292)

[6.2.8 Security 27](#_Toc325640293)

[7 Messaging or Persistence Schema 27](#_Toc325640294)

[7.1 <XML Schema Name> 27](#_Toc325640295)

[8 Middle Tier Objects (Service and Business Layers) 28](#_Toc325640296)

[8.1 <Web Service Name> 28](#_Toc325640297)

[8.1.1 Web Service Description 28](#_Toc325640298)

[8.1.2 Directory Structure 28](#_Toc325640299)

[8.1.3 Configuration 28](#_Toc325640300)

[8.1.4 Security 28](#_Toc325640301)

[8.2 Components (Business Objects) 28](#_Toc325640302)

[8.2.1 Object Map 29](#_Toc325640303)

[8.2.2 Class definitions 29](#_Toc325640304)

[8.2.3 Localization 29](#_Toc325640305)

[8.2.4 Configuration 29](#_Toc325640306)

[8.2.5 Security 29](#_Toc325640307)

[8.2.6 Error Handling 29](#_Toc325640308)

[9 Database Objects 30](#_Toc325640309)

[9.1 Entity Relationship Diagram 30](#_Toc325640310)

[9.1.1 Tables used in Application 30](#_Toc325640311)

[9.1.2 Meta Data Table Structure 30](#_Toc325640312)

[9.2 Tables 31](#_Toc325640313)

[9.3 Views 31](#_Toc325640314)

[9.4 Indexes 31](#_Toc325640315)

[9.4.1 <Index Name> 31](#_Toc325640316)

[9.5 Triggers 32](#_Toc325640317)

[9.5.1 <Trigger Name> 32](#_Toc325640318)

[9.6 User Defined Functions 32](#_Toc325640319)

[9.6.1 <User Defined Function Name 1> 32](#_Toc325640320)

[9.6.2 <User Defined Function Name 2> 32](#_Toc325640321)

[9.7 Stored Procedures 32](#_Toc325640322)

[9.7.1 <Stored Procedure 1> 32](#_Toc325640323)

[9.7.2 <Stored Procedure 2> 33](#_Toc325640324)

[9.8 DTS or SQL Integration Server Packages 33](#_Toc325640325)

[9.8.1 <DTS Package 1> 33](#_Toc325640326)

[9.8.2 <DTS Package 2> 34](#_Toc325640327)

[9.9 Package Execution 35](#_Toc325640328)

[9.10 Assumptions and Limitations 35](#_Toc325640329)

[9.11 Scripts 35](#_Toc325640330)

[10 Batch Jobs/Services 35](#_Toc325640331)

[10.1 Batch Job/Service Schedule and Dependencies 35](#_Toc325640332)

[10.2 Security 37](#_Toc325640333)

[10.3 <Batch Job/Service Name> 37](#_Toc325640334)

[10.3.1 Overview 37](#_Toc325640335)

[10.3.2 Security 38](#_Toc325640336)

[10.3.3 Parameters 39](#_Toc325640337)

[10.3.4 <Step Name> 39](#_Toc325640338)

[11 Registry Settings/INI files/.config files 39](#_Toc325640339)

[12 Reporting 39](#_Toc325640340)

[12.1 <Report Name> 40](#_Toc325640341)

[12.1.1 Dependencies 40](#_Toc325640342)

[12.1.2 Data/Metadata 40](#_Toc325640343)

[12.1.3 Processing/Display 40](#_Toc325640344)

[12.1.4 Delivery/Archiving 40](#_Toc325640345)

[13 Notification Services 40](#_Toc325640346)

[13.1 Notification Subscription 40](#_Toc325640347)

[13.2 Notification Generation 40](#_Toc325640348)

[13.3 Notification Formatting 40](#_Toc325640349)

[14 Windows Services 40](#_Toc325640350)

[15 System Integration Points 40](#_Toc325640351)

[15.1 Data Mapping 41](#_Toc325640352)

[15.2 Data Capture Rules 41](#_Toc325640353)

[16 Internationalization 41](#_Toc325640354)

[16.1 Globalization 41](#_Toc325640355)

[16.2 Localization 41](#_Toc325640356)

[16.2.1 Languages 41](#_Toc325640357)

[16.3 Market Customization 42](#_Toc325640358)

[17 Special Considerations 42](#_Toc325640359)

[17.1 Volume Considerations 42](#_Toc325640360)

[17.2 Testing Considerations 42](#_Toc325640361)

[17.3 Production Support Considerations 42](#_Toc325640362)

[17.4 Integration with MSOps.com Portal 42](#_Toc325640363)

[17.5 Performance & Response Time 42](#_Toc325640364)

[17.6 Setup and Deployment Strategy 42](#_Toc325640365)

[17.7 Software Development Environment 43](#_Toc325640366)

[17.7.1 Version Control System 43](#_Toc325640367)

[17.7.2 Build Procedures 43](#_Toc325640368)

[17.7.3 Deployment Procedures 43](#_Toc325640369)

[Appendix A Glossary/ Definitions 44](#_Toc325640370)

[Appendix B Related Documents/References 45](#_Toc325640371)

[Appendix C Environments 46](#_Toc325640372)

[Appendix D : Back-End (SAP) 47](#_Toc325640373)

[D.1 Development Overview/Approach *[REQUIRED for SAP]* 47](#_Toc325640374)

[D.1.1 Requirements Summary 47](#_Toc325640375)

[D.1.2 Assumptions 47](#_Toc325640376)

[D.1.3 Dependencies / Constraints 47](#_Toc325640377)

[D.1.4 Applications, Objects or Transactions Affected 47](#_Toc325640378)

[D.1.5 SAP Object Attributes 47](#_Toc325640379)

[D.1.6 Transaction Volume 47](#_Toc325640380)

[D.1.7 Standards 47](#_Toc325640381)

[D.1.8 Estimation of Effort 47](#_Toc325640382)

[D.1.9 Code Management Tool 48](#_Toc325640383)

[D.2 High Level System Architecture *[REQUIRED for SAP]* 48](#_Toc325640384)

[D.2.1 Technical Flow Diagram 48](#_Toc325640385)

[D.2.2 Technical Flow Description 48](#_Toc325640386)

[D.3 Data Element List *[REQUIRED for SAP]* 48](#_Toc325640387)

[D.4 Interface Type *[REQUIRED for SAP]* 48](#_Toc325640388)

[D.5 SAP RFC Connection Information *[REQUIRED for SAP]* 49](#_Toc325640389)

[D.6 Detailed SAP Interface & Conversion Information *[REQUIRED for SAP]* 49](#_Toc325640390)

[D.6.1 Interface Parameters [Programs] *[REQUIRED for SAP Interfaces][AS NEEDED for SAP Conversions]* 49](#_Toc325640391)

[D.6.2 Interface Parameters [RFCs] *[REQUIRED for SAP Conversions & Interfaces]* 49](#_Toc325640392)

[D.6.3 High-level processing flow & Developer comments *[REQUIRED for All Development Objects]* 51](#_Toc325640393)

[D.6.4 Processing and Operational Considerations 51](#_Toc325640394)

[D.6.5 Execution Mode [Real-time, Batch] *[REQUIRED for SAP]* 51](#_Toc325640395)

[D.6.6 Dependencies *[REQUIRED for SAP]* 51](#_Toc325640396)

[D.6.7 Batch Requirements 52](#_Toc325640397)

[D.7 Security and Authorization *[REQUIRED for SAP]* 52](#_Toc325640398)

[D.7.1 Account Requirements 52](#_Toc325640399)

[D.7.2 Processing Requirements 52](#_Toc325640400)

[D.7.3 Authority Checks 52](#_Toc325640401)

[D.7.4 Authorization Group Attribute 52](#_Toc325640402)

[D.7.5 Threat Modeling 52](#_Toc325640403)

[D.8 External Programs 53](#_Toc325640404)

[D.9 Technical SAP Requirements *[REQUIRED for SAP]* 53](#_Toc325640405)

[D.9.1 SAP Database Objects (created/changed) 53](#_Toc325640406)

[D.9.2 Custom SAP Tables 53](#_Toc325640407)

[D.9.3 Custom SAP Views 53](#_Toc325640408)

[D.9.4 SAP Table Indexes 53](#_Toc325640409)

[D.9.5 Testing Considerations for SAP 53](#_Toc325640410)

[Appendix E Document Change History 54](#_Toc325640411)

[Appendix F Reviewers 55](#_Toc325640412)

**Directions for using template**:

Read the Guidance Text (*Arial blue italic font in brackets*) to understand the information that should be placed in each section of this template. Then delete the Guidance and replace the placeholder within <<Begin text here>> with your response. There may be additional Guidance in the Appendix of some documents, which should also be deleted once it has been used

Note that the top portions of this document are useful for High Level Design review as performed by the EA team and/or Architecture Review Teams within various Centers of Excellence. The following sections must be filled out for this high level design review: Chapter 1 (Introduction), Chapter 2 (System Architecture), and Chapter 3 (Application Subject Area Context Model).

Legend:

* All sections, unless specified below, follow same applicability as its parent
* Y – Required content, Empty – Optional content
* H – High level Design and D - Detailed Design



# Introduction

[Prior to reviewing this document please refer to the Vision and Scope document, the Business Requirements document and the Functional Specification. See [Appendix: Related Documents/References](#Related_Documents_References) for links to these documents.]

<<Begin text here>>

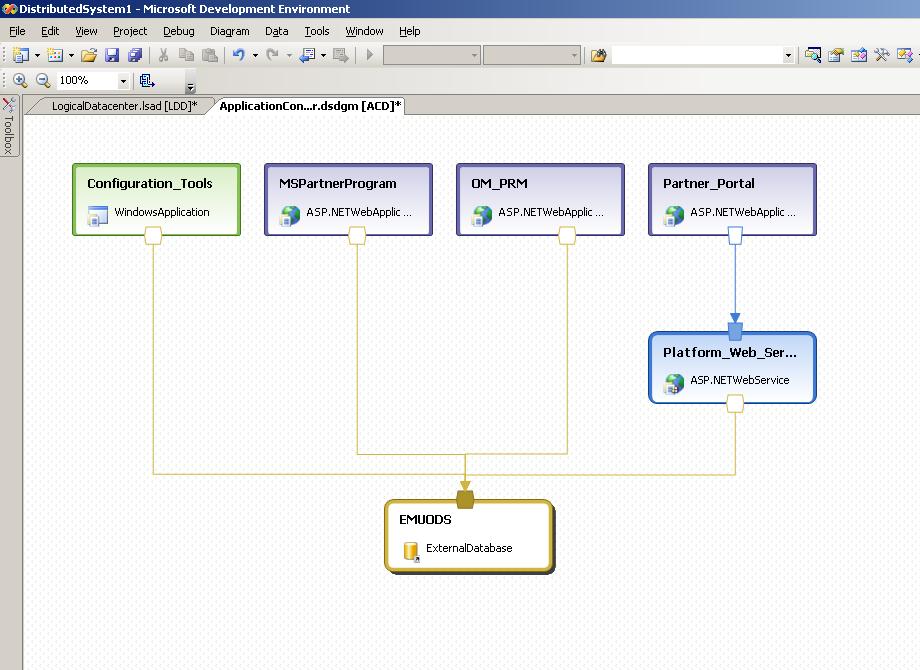
# System Architecture

## Distributed System View

### Application Context Diagram

[Provide “Application Context Diagram” which illustrated both “internal” and “external” applications. The internal applications are individually deployable applications being defined within the solution. Each internal application typically results in a single project that defines a host-able executable (such as a Web application or a Windows application. The external applications are outside of the solution domain but they provide key services and data to internal applications. In a simplest format, Application Connection Diagram defines application in a distributed context. The application architects and designers can extend the Application Connection Diagram to represent the definition of behavior provided by a service, whether by internal applications or external applications. For example, a web service endpoint can be expanded to include the definition of operations for that web service. By extending Application Connection Diagram this way, the diagram defines both “distributed application architecture” of overall solution and “logical application architecture” of a chosen application. See sample below]

<<Begin text here>>



### Functional Components

[Repeat the Functional Component Description section for each row in the following table. An element should exist in the table for each application or functional component described in the above model.]

<<Begin Text Here>>

|  |  |  |
| --- | --- | --- |
| **Component Name** | **Responsibility** | **Execution Environment** |
|  |  |  |
|  |  |  |
|  |  |  |

<<Begin text here>>

* + - 1. Functional Component Description

[Provide a high level description for each functional component as illustrated “Application Connection Diagram”. Application architects must provide the system use case survey and dynamic models for each functional application component to illustrate primary responsibilities]

* + - 1. System Use Case Survey

<<Begin text here>>

|  |  |  |
| --- | --- | --- |
| Use Case Number | Use Case Name | System Use Case Trigger |
| SC01 | New File Processing | Notification arrives from component Y that a new transaction is ready to process. |
|  |  |  |
|  |  |  |

* + - 1. Dynamic Models

*[At the design level, the dynamic model should illustrate only high level layers or components and should provide notes about their responsibilities, mechanisms for communication, and system quality attributes. It would be typical to illustrate system-to-system communications in a diagram in this section, as long as you make it clear the order of the communications, not just the existence of a communication link.]*

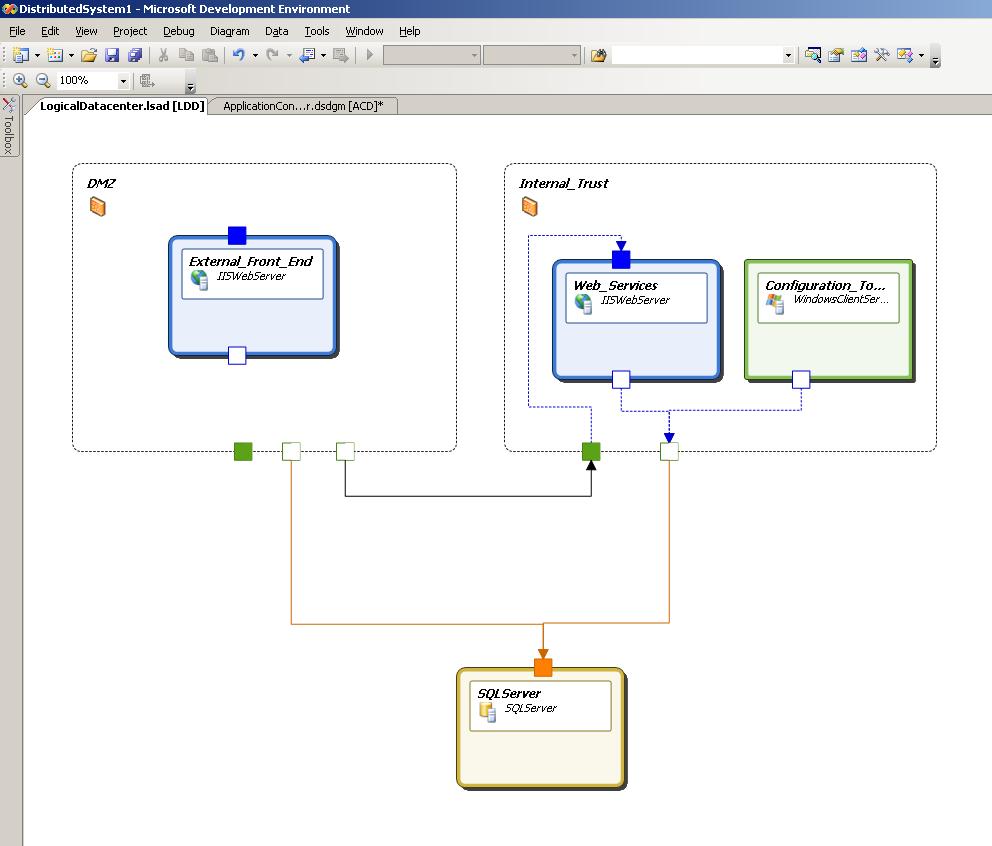
The following dynamic model illustrates a chain of feeds used to deliver product data from its source (MSProduct) to the POD application.

DataFeeds

## Deployment View

[Insert “UML Deployment Model” or “VS Deployment diagram” to represent interconnected logical servers in a context of “logical structure of a datacenter. This diagram communicates important information to the developer about the target deployment environment and to Operations architects about development view of application deployment. This diagram is important to developers because it allows developers to understand application boundaries to the extent they can be more effective in designing “security”, “trust” and “configuration” aspect of the application. See sample below. The Deployment model needs to describe the details that a designer is ‘asserting’ as ‘these things must exist for my design to work.’ The deployment architects would not add detail to this document on how to accomplish that hosting environment]

<<Begin text here>>



### Logical Server Definition

[Add a row in this table for every server in the “Logical Datacenter Diagram”]

|  |  |  |  |
| --- | --- | --- | --- |
| **Server Logical Name** | **Software environment** | **Security Environment** | **Hardware and Networking** |
| AppServer1 | Windows Server 2003 with IIS | Active Directory in Partners Extranet | Hardware Load Balanced |
| ExternalDatabase1 | Windows Server 2003, SQL Server 2005 | Active Directory in Partners Extranet | Dual homed with network connections to Redmond domain for replication |

## System Quality Attributes

*[The goal of writing down the System Quality Attributes is as an input to the Architecture Tradeoff Analysis Method (ATAM) process. The ATAM process allows for architects and senior designers to review the design of this application to provide a quality check to the architectural tradeoffs that are visible in the design. Therefore, in addition to outlining the attribute itself, this section needs to discuss the tradeoffs that were made in the name of each of the attributes.]*

<<begin text here>>

### Performance

*[Performance refers to responsiveness: either the time required to respond to specific events (latency) or the number of events processed in a given interval of time (throughput).How much demand can be placed on the system while meeting latency and throughput requirements, also called capacity. How can the demand change over time and what should happen when the system capacity exceeds and not all events can be responded in a timely manner? Also describe the environment under which this performance will be verified.]*

<<begin text here>>

### Dependability

*[Dependability is a property of the computer system such that reliance can justifiably placed on the service it delivers. Provide enough information to determine how dependability will be evaluated for the sake of architectural tradeoff analysis. .]*

<<begin text here>>

* + - 1. Availability

*[Readiness for usage. Describe the trade-offs that you made in the design either in favor of greater availability or that resulted in an expected loss of availability for the sake of another attribute.]*

<<begin text here>>

* + - 1. Reliability

*[Continuity of service. Describe the trade-offs that you made in the design either in favor of greater reliability or that resulted in an expected loss of reliability for the sake of another attribute.]*

<<begin text here>>

* + - 1. Safety

*[Non-occurrence of catastrophic consequences on the environment. Describe the trade-offs that you made in the design either in favor of greater safety or that resulted in an expected loss of safety for the sake of another attribute.]*

<<begin text here>>

* + - 1. Maintainability

*[Aptitude to undergo repairs and evolution or brittleness when changes are made. Describe the trade-offs that you made in the design either in favor of greater maintainability or that resulted in an expected loss of maintainability for the sake of another attribute.]*

<<begin text here>>

### Security

*[This section is not to describe the techniques used to secure the application. This section is used to describe the trade-off decisions that the designer made when considering security with respect to the other attributes of the architecture.]*

* + - 1. Confidentiality

*[Confidentiality is the property that data be inaccessible to unauthorized users, for example user roles and their* access to the system. *Describe the trade-offs that you made in the design either in favor of greater confidentiality or that resulted in an expected loss of confidentiality for the sake of another attribute.]*

<<begin text here>>

* + - 1. Integrity

*[Integrity is the property that data be resistant to unauthorized modification. Describe the trade-offs that you made in the design either in favor of greater integrity or that resulted in an expected loss of integrity for the sake of another attribute.]*

<<begin text here>>

# Application Subject Area Context Model

[A diagram that illustrates the data connections between the application data subject areas both within the application boundary and that are communicated with other applications. Do not show messaging infrastructure like EAI hubs on this diagram, as the channel configuration is not salient at this level. Please show local cache requirements if the application architecture requires local cache of data for performance reasons. This diagram also serves as a high-level logical ER diagram to help guide db design. On maintenance projects, this diagram is required.]

<<Begin text here>>

# Maintenance Impact Analysis *(This section is best practice and is required for projects that have an Enterprise Architecture Stakeholder)*

[This chapter is to be used specifically where the project is going to modify an existing system. This section will contain an analysis of the sections of existing code that need to be changed to support the new requirements imposed by the project. Separate the sections in this chapter first by technologies that need to be assigned to different teams. Therefore, if the project will change data feeds, SAP components, and application code, then you will have three sections since different teams will work on each one. Then within each section, describe the changes needed on a feature-by-feature basis. Therefore, within application changes, to support “feature X” describe the database, messaging, middleware, and user interface changes needed. For projects that are primarily maintenance of existing systems, this chapter may be used to replace the sections that describe the design in a detailed sense (e.g. chapters on Front end, Middleware, and Database objects).]

<<Begin text here For example:>>

We are making the following changes to support the new features for Net Invoicing.

## <Name of first feature goes here>

*[Describe the details of this feature]*

### Areas of Impact

*[Describe which areas of the current system are affected for this feature. For each area affected, describe the changes that need to be done. Also, think about consequence of this change to other parts of the system and explicitly state the areas that are not affected.]*

### Technology Area - Web Pages / Windows Application

*[Describe which web pages, windows application, etc., needs to be changed and why? While you describe the specific changes, include the rationale on why such a design change decision is made. Also note, the error conditions that must be verified, unit testing that must be done]*

* + - 1. **Web Page / Windows Application**
         1. Description

*[Insert description of page/control, Physical file name, Class name, Namespace, Assembly it belongs to, properties and methods of the class, etc.]*

|  |  |  |
| --- | --- | --- |
| **Function / Module Name** | **Description** | **Reason / Comments for Change** |
|  |  |  |
|  |  |  |

### Technology Area - Middle Tier Objects

*[Describe which services, objects, etc., needs to be changed and why? While you describe the specific changes, include the rationale on why such a design change decision is made. Also note, the error conditions that must be verified, unit testing that must be done]*

* + - 1. **Web Services**
         1. Web Service 1

Description

*[Insert description of web service, purpose of it.]*

|  |  |  |
| --- | --- | --- |
| **Function / Module Name** | **Description** | **Reason / Comments for Change** |
|  |  |  |
|  |  |  |

* + - * 1. Web Service 2

Description

*[Insert description of page/control, Physical file name, Class name, Namespace, Assembly it belongs to, properties and methods of the class, etc.]*

|  |  |  |
| --- | --- | --- |
| **Function / Module Name** | **Description** | **Reason / Comments for Change** |
|  |  |  |
|  |  |  |

* + - 1. **Components**
         1. Component Name 1

Description

*[Insert description of objects and the business use of these objects]*

|  |  |  |
| --- | --- | --- |
| **Function / Method Name** | **Description** | **Reason / Comments for Change** |
|  |  |  |
|  |  |  |

* + - * 1. Component Name 2

Description

*[Insert description of objects and the business use of these objects]*

|  |  |  |
| --- | --- | --- |
| **Function / Method Name** | **Description** | **Reason / Comments for Change** |
|  |  |  |
|  |  |  |

### Technology Area – Database Changes

* + - 1. **Tables**

[This section contains the list of tables that are getting changed for this feature. Also any Domain table specified under this section should also contain the values that are to be inserted in that table.]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table Name** | **Column Name** | **Add / Update/ Delete** | **Description** | **Data type** | **Reason for Change** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

* + - 1. **Views**

[This section should contain only the list of views that requires a change for this feature]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **View Name** | **Column Name** | **Add / Update/ Delete** | **Description** | **Reason for Change** |
|  |  |  |  |  |
|  |  |  |  |  |

* + - 1. **Indexes**

*[Fill in the table below for each index that is include for this feature]*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Index Name** | **Column Name** | **Table Name** | **Index Type** | **Add / Delete** | **Reason for Change** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

* + - 1. **Triggers**

*[Fill in the table below for each trigger that is include for this feature]*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Trigger Name** | **Table Name** | **Trigger Type** | **Add / Update/ Delete** | **Description** | **Reason for Change** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

* + - 1. **User Defined Functions**
         1. User Defined Functions < Name 1>

Description

*[Describe the reason for having this user defined function and do mention why we need a change on this function]*

Parameters (Input , Output and Return Values)

Pseudocode

* + - * 1. User Defined Functions < Name 2>

Description

*[Describe the reason for having this user defined function and do mention why we need a change on this function]*

Parameters (Input , Output and Return Values)

Pseudocode

* + - 1. **Stored Procedures**
         1. Stored Procedures <Name 1>

Purpose

[Define the purpose of the procedure.]

<<Begin text here>>

Parameters (Input and Output) and Return Codes

[Specify input and output parameters including data types. Specify the value and meaning of any return codes the procedure will return.]

<<Begin text here>>

Called From

[Provide details on how the stored procedure is called from within the application. A stored procedure may be called from a web page, from a batch job or from another stored procedure.]

<<Begin text here>>

Processing/Pseudocode

[Specify the processing the stored procedure will perform. Pseudo code may also be provided.]

<<Begin text here>>

Error Handling

[Specify error handling within the stored procedure.]

<<Begin text here>>

* + - * 1. Stored Procedures <Name 2>

Purpose

[Define the purpose of the procedure.]

<<Begin text here>>

Parameters (Input and Output) and Return Codes

[Specify input and output parameters including data types. Specify the value and meaning of any return codes the procedure will return.]

<<Begin text here>>

Called From

[Provide details on how the stored procedure is called from within the application. A stored procedure may be called from a web page, from a batch job or from another stored procedure.]

<<Begin text here>>

Processing/Pseudocode

[Specify the processing the stored procedure will perform. Pseudo code may also be provided.]

<<Begin text here>>

Error Handling

[Specify error handling within the stored procedure.]

<<Begin text here>>

* + - 1. **DTS Packages / Integration Services**

[Complete the following section for each DTS Package or SSIS services.]

<<Begin text here>>

* + - * 1. <DTS Package 1>

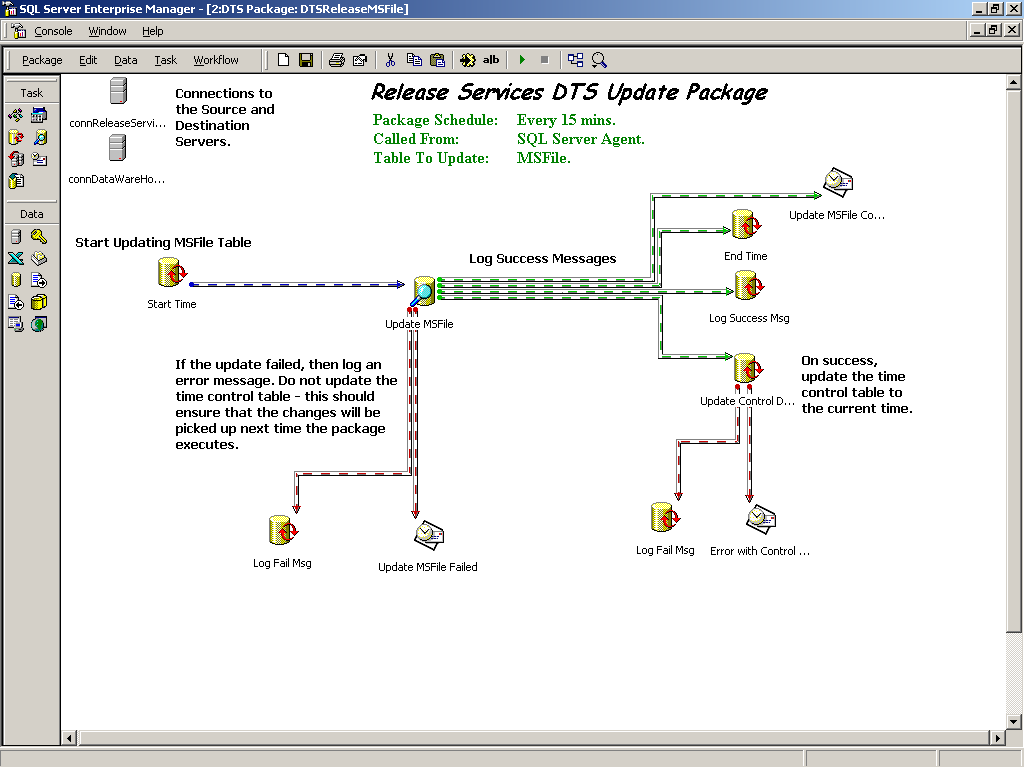
Purpose

[Define the purpose of the DTS package. Specify the source and destination of the data. Specify where the DTS package will be located.]

<<Begin text here>>

Processing/Pseudocode

[Specify the logic/structure of the DTS Package will perform in narrative form or Pseudocode. A screen shot may also be provided. See below for a sample screenshot and narrative for an update package from the RSM application documentation:



* A Starting message is logged
* A DTS DDQ (Data Driven Query) task is executed. This is the task that actually performs the transforms, updates/inserts.
* If the DDQ task fails, a message is logged.
* If the DDQ task completes successfully, messages are logged and the control table DTSLastRunTime is updated with the current time for the current destination table being update]

<<Begin text here>>

Error Handling

[Specify error handling within the DTS Package.]

<<Begin text here>>

* + - * 1. <DTS Package 2>

Purpose

<<Begin text here>>

Processing/Pseudocode

<<Begin text here>>

Error Handling

<<Begin text here>>

* + - 1. **Package Execution**

[This table outlines how the DTS or SSIS packages will be executed and controlled. Need to mention only the packages or services that are going to be changed for this feature]

<<Begin text here>>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NTBatch Job Name | Type | Packages Executed | Parameters | Return Values | Called From |
| DTS\_RSM | Controlling Job | MSFile, ServiceRequest,UserAccount | Char (1) – Expects ‘F’ for full transfer or ‘I’ for incremental transfer. | NTBatch Success OR  Failure. | NTBatch Schedule |
| DTS\_Pak1 | Sub Job | SRContact, SRFileID, UsedIn | Char (1) – Expects ‘F’ or Blank | NTBatch Success OR  Failure. | DTS\_RSM Job |
|  |  |  |  |  |  |

* + - 1. **Data Conversion Scripts**

[Use this section to define scripts that should be created to add/update/delete data or add/modify/delete database objects or any migration scripts that are required.]

<<Begin text here>>

## <Name of second feature goes here>

[repeat all of the sections from 4.1 into this area if the maintenance project will implement more than one new or updated feature. If it will not, delete this header.]

# Application Detailed Design *(This section is best practice and is required for projects that have an Enterprise Architecture Stakeholder)*

*[For projects that are creating new components or substantially rewriting existing components from the ground up, this section should be used to provide information on the component or application. Normally, this section and the ones that follow are not used for maintenance projects where the change is viewed as ‘relative to existing design’. In that sense, section 4 is mutually exclusive with sections 5 and onward. When filling out this template, if you find you do not need a section, leave the chapter heading in place, but feel free to delete the body of that chapter. ]*

## End to End Dynamic Models

*[This section is used to present UML diagrams that illustrate how the system components will be used in an end-to-end context, usually with respect to a use case. Each diagram is filed under the header of the use case that it is illustrating with a block of text describing which choices from the use case are illustrated. It is normal for one use case to yield as many models as the designers feel are necessary to describe the interaction. Not every use case requires a unique model Repeat the following section for each use case modeled.]*

### Example use case: GetAgreementList

This particular diagram illustrates the interaction needed when the front end needs the list of agreements from the database (use case 16: Get List of Agreements).

AgreementViewerPOCIn this diagram, the ‘sunny day flow’ is illustrated. The user provides sufficient and correct information in order to retrieve a list of agreements, and the list is correctly returned to the front end.

### <Use Case Name Goes Here>

<<place text that gives a brief overview of the use case.>>

<<place the dynamic model diagram here. Use the EMF format to keep the diagram size small.>>

<<place text here that describes which choices from the use case are illustrated in the model.>>

## Technology Dependencies

### .NET Framework Version Compatibility Issues

[All applications need to consider their position with respect to the particular version of the .Net framework that is being leveraged, and how that version will affect the application both during development and on into maintenance. Use this section to describe any constraints, installation issues, backward or forward compatibility issues, or emerging technologies that may affect the way the application is developed, tested, deployed or maintained]

<<Begin text here>>

### Server software dependencies

*[Use this section to list out the dependencies on particular server products or versions of server products that will need to be present for installation and execution to succeed. Note, in this section, how each of these technologies are installed and whether or not the support team is expected to keep updating the server software when new versions of this technology are released from the product groups. External (third party) technologies need to include details about underlying things that they depend upon.]*

<<Begin text here>>

### Client software dependencies

*[Use this section to list out the dependencies on particular client products or versions of client products that will need to be present for installation and execution to succeed. Note, in this section, how each of these technologies are installed and whether or not the support team is expected to keep updating the client software when new versions of this technology are released from the product groups. External (third party) technologies need to include details about underlying things that they depend upon.]*

<<Begin text here>>

## Security Roles

*[Use this section for the tangible mapping of security roles in Active Directory or an intermediate mapping for portals to specific pages or capabilities within the application(s). This is part of the detailed design. Let the definition of high-level business roles live in the BRD or functional spec without copying that information into this document.]*

|  |  |  |
| --- | --- | --- |
| **Role or Group ID** | **Module or Page** | **Level of Access or Capability** |
|  |  |  |
|  |  |  |

# Front End

*[Different applications may leverage one or more front-end technology or may in fact not have a front end. Use only the section that is appropriate for your application. Leave the level 2 section header for any section that you do not need. For example, if you have a web app that does not have a windows or smart client front end, use section 6.1, leave the headers for sections 6.2 and 6.3 but delete their bodies.]*

## Web Application Front End

### Physical Site Map

[Provide a Physical ‘site map’ of the application being developed. See below for an example from the MERT 1.8 application documentation:

This diagram represents the physical site for MERT 1.8. The name of each page is listed along with any frames that each page appears with. The arrows represent links used to access each page.]

<<Begin text here>>





### Common UI Technical Specification

[In this section, specify functionality, which will be common to all UI Web Pages/Screens. The headings provided are for a web application and may be altered to suit the application being developed.]

<<Begin text here>>

* + - 1. Web-site Directory Structure

[Provide details on web-site directory structure. The example below is an extract from the PIRSHome application documentation.

The web site has the following directory structure:

\ - Root directory containing homepage

\crt - Files containing the CRT files and business owned content

\Images - Images Directory

\Include - Directory with include files

\pirsUpload - Directory containing the file upload components.]

<<Begin text here>>

* + - 1. Style sheet

[For most web sites, a cascading style sheet will be included in all the web pages used in the site. It is used to define the presentation and formatting for the entire site. In this section specify the styles to be included in the CSS File.]

<<Begin text here>>

* + - 1. Page Templates

*[For this section, describe the standard layout templates that will be used for all pages in the app, including visual diagrams for navigation, header/footer, logos, and other site art.]*

<<Begin text here>>

### Web Pages and Web Controls

[Repeat following subsection for each web page and control]

<<Begin text here>>

* + - 1. Description[of individual web pages or controls]

[Insert description of page/control, Physical file name, Class name, Namespace, Assembly it belongs to, properties and methods of the class, etc.]

<<Begin text here>>

### Localization

[Insert localization information for the web application, including:

* Localization model (eg. .NET Resource Manager)
* Language selection method(s) (eg. sniff browser language; select from picklist; etc.)
* Export/import of localized strings]

<<Begin text here>>

### Configuration

[Provide web.config and Register Settings for this web application]

<<Begin text here>>

### Presentation Layer Class definitions

[This section is to be used by classes that are called directly by the web application code-behind classes. The presentation layer contains code for laying out the user interface and simple validation of incoming data. Note that the services layer will also validate data. Repeat following sections for each class; alternatively, link to UML diagram]

* + - 1. Class description

[For each class provide:

Description,

Physical file name,

Class Name,

Namespace,

Assembly,

Base class (derived from),

Properties,

Methods,

Other information deemed important]

<<Begin text here>>

### Collaboration Layer Class definitions

[This section is to be used by classes that are called by presentation layer objects, including the presentation layer classes and the web application code-behind classes. The collaboration layer is the layer responsible for managing the service proxies and for calling services in a traditional SOA application. Repeat following sections for each class; alternatively, link to UML diagram]

<<Begin text here>>

* + - 1. Class description

[For each class provide:

Description,

Physical file name,

Class Name,

Namespace,

Assembly,

Base class (derived from),

Properties,

Methods,

Other information deemed important]

<<Begin text here>>

### Error Handling

* + - 1. Server Side

[Outline any server side error handling that will be common to all web pages.]

<<Begin text here>>

* + - 1. Client Side

[Outline any client side error handling that will be common to all web pages.]

<<Begin text here>>

### Security

* + - 1. Authentication

<<Begin text here>>

* + - 1. Authorization

<<Begin text here>>

* + - 1. Encryption

<<Begin text here>>

* + - 1. Cross Scripting and Input Security

[For web applications: specify the steps that are being taken to ensure that no malicious code can be input into the pages. A number of procedures are used to avoid cross scripting issues including: checking the referring page is on the same server, validation of inputs and limiting the range of inputs. Describe checks being made for data passing from an untrusted area/domain to a trusted one.]

<<Begin text here>>

## Windows Application Front End

### Screen Navigation map

[Insert a diagram that illustrates the windows forms, controls, dialog box and their navigational relationships.]

<<Begin text here>>

### Object Model

[Insert UML dioagram describe the object model of screen and user control classes]

<<Begin text here>>

### Screens and User Controls

[Repeat following sections for each screen and user control]

<<Begin text here>>

* + - 1. Screen or Control definition

[For each class provide:

Description,

Physical file name,

Class Name,

Namespace,

Assembly,

Base class (derived from),

Properties,

Methods,

Other information deemed important]

<<Begin text here>>

### Localization

[Insert localization information for the windows application, including:

* Localization model (eg. .NET Resource Manager)
* Language selection method(s) (eg. sniff browser language; select from picklist; etc.)
* Export/import of localized strings]

<<Begin text here>>

### Common Components

[Repeat following sections for each screen and user control]

<<Begin text here>>

* + - 1. Class definition

[For each class provide:

Description,

Physical file name,

Class Name,

Namespace,

Assembly,

Base class (derived from),

Properties,

Methods,

Other information deemed important]

<<Begin text here>>

### Configurations

[Insert key app.exe.config and registry settings for this Windows application]

<<Begin text here>>

### Error Handling

[Insert error handling information]

<<Begin text here>>

### Security

* + - 1. Authentication

<<Begin text here>>

* + - 1. Authorization

<<Begin text here>>

* + - 1. Encryption

<<Begin text here>>

# Messaging or Persistence Schema

[An XML schema diagram (Visual studio XML Schema diagram) may be included here or may be referenced as a separate document. This must be included for any messages that will be sent from this system to an EAI infrastructure or that are being introduced by this project to be sent from another system to be consumed by the new code. Message interaction should already be described in the Architecture section. Any description of schema used for data persistence should include a description of how that data is stored and managed in its persisted state. ]

<<Begin text here>>

## <XML Schema Name>

[For each schema describe the purpose, define its structure and specify what database table or UI data field the data is derived from. The sample schema below and sample table entries are derived from the RIO application documentation:

<CST TYPE='ShipTo'>

<NBR></NBR>

<NM1> </NM1>

<POS></POS>

</CST>]

<<follow below or insert>>

|  |  |  |
| --- | --- | --- |
| XML Tag Name | XML Tag Name Description | Source field name |
| CST | **C**u**st**omer | N/A |
| @TYPE | Is an attribute of element CST. Contains the type of customer – Sold-To / Ship-To. | N/A |
| NBR | Customer **N**um**b**e**r** | TblCustomer.CustNo |
| NM1 | Ship-To **N**a**m**e **1** | TblCustomer.Name1 |
| POS | Ship-To **Pos**tal Code | TblCustomer.PostalCode |

# Middle Tier Objects (Service and Business Layers)

## <Web Service Name>

### Web Service Description

* + - 1. Interface

[Insert or link to wsdl xml for web services interface; alternatively insert interface definitions via UML diagram]

<<Begin text here>>

* + - 1. Implementation

[Insert following info:

Physical file name,

Class Name,

Namespace,

Assembly,

Properties,

Methods]

<<Begin text here>>

### Directory Structure

[Include vroot]

<<Begin text here>>

### Configuration

[Insert Web.config and Registry setting information]

<<Begin text here>>

### Security

[Insert security model how web services authenticate clients and authorize access levels]

<<Begin text here>>

* + - 1. Authentication

<<Begin text here>>

* + - 1. Authorization

<<Begin text here>>

* + - 1. Encryption

<<Begin text here>>

## Components (Business Objects)

[Provide the overall class diagrams for the different objects and interfaces. Treat web service as regular classes.]

<<Begin text here>>

### Object Map

[Insert Sequence diagram and UML diagram, the sequence diagram is to understand objects in action to respond to execution path of specific scenarios, UML is to document class interaction in designer’s view]

<<Begin text here>>

### Class definitions

[Repeat following sections for each class; alternatively, link to UML diagram]

<<Begin text here>>

* + - 1. Class description

[For each class provide:

Description,

Physical file name,

Class Name,

Namespace,

Assembly,

Base class (derived from),

Properties,

Methods,

Other information deemed important]

<<Begin text here>>

### Localization

[Insert localization information for the business objects, including:

* Localization model (eg. .NET Resource Manager)
* Language selection method(s) (eg. sniff browser language; select from picklist; etc.)
* Export/import of localized strings]

<<Begin text here>>

### Configuration

[Insert registry settings]

<<Begin text here>>

### Security

* + - 1. Authentication

<<Begin text here>>

* + - 1. Authorization

<<Begin text here>>

* + - 1. Encryption

<<Begin text here>>

### Error Handling

[Describe error handling info. Use a diagram tool such as Visio UML or Whidbey Class designer and include link here.]

<<Begin text here>>

# Database Objects

[The following table lists the database objects, their location and a brief description. Database object types include indexes, triggers, user defined functions, stored procedures, DTS packages etc. If Erwin is being used to generate reports on Tables and Views it is not necessary to list Tables and Views in this table.]

<<Begin text here>>

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Database Location | Description |
| e.g. SpStoredProcedureName | Stored Procedure |  |  |
| e.g. DTSPackageName | DTS Package |  |  |
|  |  |  |  |

## Entity Relationship Diagram

### Tables used in Application

[A data schema (ER diagram) may be included here or may be referenced as a separate document. The sample below is a subset of the RIO ERwin diagram. On maintenance projects where the changes to the database are minimal, this diagram is not required. This ER diagram would contain the list of tables and relationships pertaining to the application]

<<Begin text here>>



### Meta Data Table Structure

[This section requires the ER diagram for the tables that are associated with Meta Data where applicable. Metadata tables are tables that drive business rules for the use of the application without ever being displayed in any report or user interface component. Examples include tables with permissions for screens, controls or page objects by user group]



## Tables

[The table structures should be included in the Database Schema. Include an ER Diagram showing for each table its name, purpose and structure. See Appendix G ERwin Reports for more details on how to generate this report. Please indicate what master keys or systems this data may have come from.

Also any Domain table specified under this section should also contain the values that are to be inserted in that table.]

## Views

[The views should be included in the Database Schema. Include an ER Diagram showing for each view it's name, purpose and structure. See Appendix G Data Schema for more details on how to generate this report.]

<<Begin text here>>

## Indexes

### <Index Name>

[For each index describe its purpose and type, list the fields indexed. Pseudo code may also be provided. For example:

CREATE CLUSTERED INDEX ciUsedPID ON [dbo].tblUseProductKeys] (PIDGroupID, SequenceNumber)]

<<Begin text here>>

## Triggers

### <Trigger Name>

[For each trigger describe its purpose, specify when it executes and provide a specification for the trigger logic. Pseudo code may also be provided]

<<Begin text here>>

## User Defined Functions

### <User Defined Function Name 1>

[For each User Defined Function describe its purpose, specify when it executes and provide a specification for the function logic. Pseudo code may also be provided]

<<Begin text here>>

### <User Defined Function Name 2>

[New section for each function]

<<Begin text here>>

## Stored Procedures

[Complete the following section for each stored procedure.]

### <Stored Procedure 1>

* + - 1. Purpose

[Define the purpose of the procedure.]

<<Begin text here>>

* + - 1. Parameters (Input and Output) and Return Codes

[Specify input and output parameters including data types. Specify the value and meaning of any return codes the procedure will return.]

<<Begin text here>>

* + - 1. Called From

[Provide details on how the stored procedure is called from within the application. A stored procedure may be called from a web page, from a batch job or from another stored procedure.]

<<Begin text here>>

* + - 1. Processing/Pseudocode

[Specify the processing the stored procedure will perform. Pseudo code may also be provided.]

<<Begin text here>>

* + - 1. Error Handling

[Specify error handling within the stored procedure.]

<<Begin text here>>

### <Stored Procedure 2>

[Repeat for each stored procedure]

* + - 1. Purpose

<<Begin text here>>

* + - 1. Parameters (Input and Output) and Return Codes

<<Begin text here>>

* + - 1. Called From

<<Begin text here>>

* + - 1. Processing/Pseudocode

<<Begin text here>>

* + - 1. Error Handling

<<Begin text here>>

## DTS or SQL Integration Server Packages

[Complete the following section for each DTS Package.]

<<Begin text here>>

### <DTS Package 1>

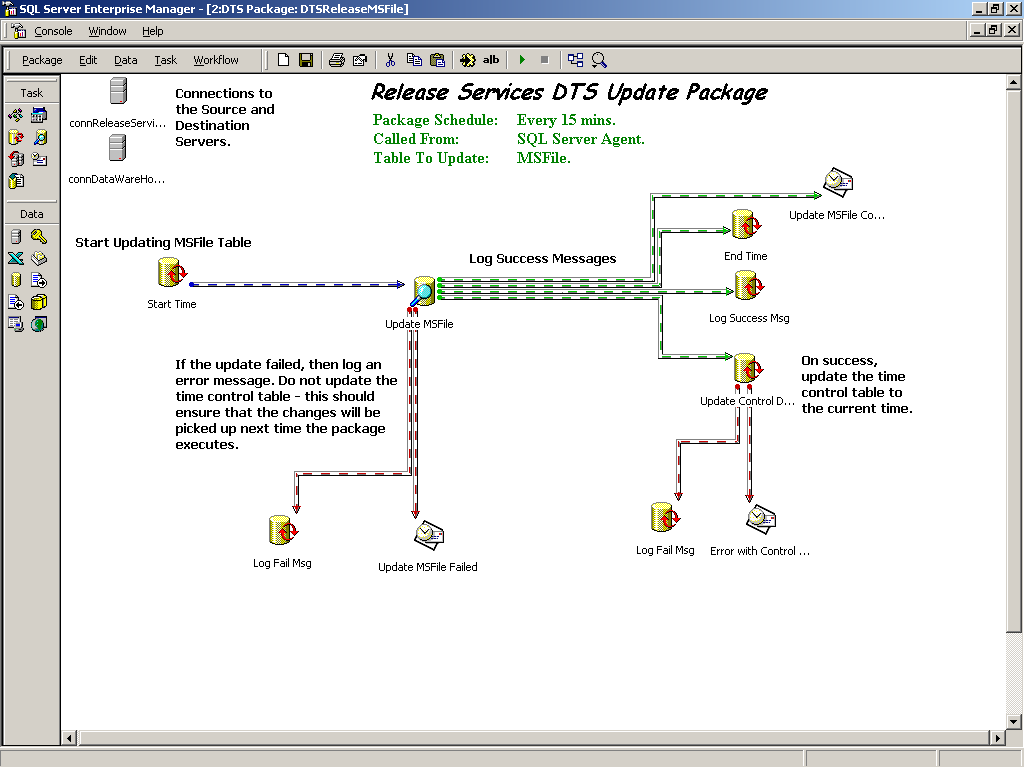
* + - 1. Purpose

[Define the purpose of the DTS package. Specify the source and destination of the data. Specify where the DTS package will be located.]

<<Begin text here>>

* + - 1. Processing/Pseudocode

[Specify the logic/structure of the DTS Package will perform in narrative form or Pseudocode. A screen shot may also be provided. See below for a sample screenshot and narrative for an update package from the RSM application documentation:



* A Starting message is logged
* A DTS DDQ (Data Driven Query) task is executed. This is the task that actually performs the transforms, updates/inserts.
* If the DDQ task fails, a message is logged.
* If the DDQ task completes successfully, messages are logged and the control table DTSLastRunTime is updated with the current time for the current destination table being update]

<<Begin text here>>

* + - 1. Error Handling

[Specify error handling within the DTS Package.]

<<Begin text here>>

### <DTS Package 2>

* + - 1. Purpose

<<Begin text here>>

* + - 1. Processing/Pseudocode

<<Begin text here>>

* + - 1. Error Handling

<<Begin text here>>

## Package Execution

[This table outlines how the DTS packages will be executed and controlled.]

<<Begin text here>>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NTBatch Job Name | Type | Packages Executed | Parameters | Return Values | Called From |
| DTS\_RSM | Controlling Job | MSFile, ServiceRequest,UserAccount | Char (1) – Expects ‘F’ for full transfer or ‘I’ for incremental transfer. | NTBatch Success OR  Failure. | NTBatch Schedule |
| DTS\_Pak1 | Sub Job | SRContact, SRFileID, UsedIn | Char (1) – Expects ‘F’ or Blank | NTBatch Success OR  Failure. | DTS\_RSM Job |
|  |  |  |  |  |  |

## Assumptions and Limitations

<<Begin text here>>

## Scripts

[Use this section to define scripts that should be created to add/update/delete data or add/modify/delete database objects.]

<<Begin text here>>

# Batch Jobs/Services

## Batch Job/Service Schedule and Dependencies

[Provide detail on the Job/Service Schedule and dependencies. Ensure that any variations to the schedule for certain days of the week or fiscal month end are also captured. This can be achieved in a number of ways. Three examples are provided below for reference.

Example 1 is a sub-section from the AIM project batch schedule documentation. For this project the batch schedule and dependencies are being captured in MS-Project. If the batch schedule is maintained in this way the MS-Project file should be referenced in this section and a link to the file included in Appendix E Related Documents/References.



Example 2 is a table that might be used to capture the same information. Sample data has been provided.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NTBatch Job Name | Type | Parameters | Return Values | Called From | Dependencies |
| Control.cmd | Controlling Job | Char (1) – Expects ‘M’ for month end or 'D' for normal daily run | NTBatch Success OR  Failure. | NTBatch Schedule at 8pm PST daily. | Cannot proceed if backup.cmd is still running. Sleeps 10 mins and checks again. |
| Import.cmd | Data Import | N/A | NTBatch Success OR  Failure. | Control.cmd | None |
|  |  |  |  |  |  |

Example 3 is from the RIO application documentation. The RIO Service is a Visual Basic EXE that handles a variety of tasks. Its main purpose is to validate and import retail orders into the RIO database and submit orders to SAP unattended. This diagram shows the relationships between the various jobs of the RIO Service.]



<<insert here>>

## Security

[Detail any access privileges (e.g., share permissions required, username for dev and test environments, etc.) For security reasons, do not include password for production environment. A table may be included to capture access privileges required. See the example above from the RIO application documentation.]

<<Begin text here>>

## <Batch Job/Service Name>

[Complete this section for each Batch Job or Service.]

<<Begin text here>>

### Overview

[Insert a description of the batch job or service. If desired the logic can be represented diagrammatically. See the example below from the RIO Application documentation.

**Job 150 – Query SAP Order status**

This job is responsible for querying order status within SAP and updating RIO status. In the event that RIO loses connection with SAP after submitting the order to SAP then it must regain communication in order to provide a link to Lucille for a given order.]



<<Begin text here>>

### Security

[Detail any access privileges (share permissions required, username, passwords for dev and test environments etc. For security reasons do not include password for production environment. A table may be included to capture access privileges required. See the example below from the RIO application documentation.]

<<Begin text here>>

| Resource Name | Alias | Permissions | Security | Reason |
| --- | --- | --- | --- | --- |
| RIOArchiveShare | NT Account running the RIOService executable.  (e.g. Redmond\batchxod) | Change, Read | Modify | MertService must be able to read, write, and delete from the archive share. |
| RIOArchiveShare | Redmond\itg-ntbatch | Read | Read | NTMail uses itg-ntbatch to send out email. If there are file attachments then itg-ntbatch must be able to read the attached file. |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

### Parameters

| Parameter | Purpose | Value |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

### <Step Name>

[For each step in a batch job/service, detail the following two sections.]

<<Begin text here>>

* + - 1. Description

[Describe the batch job/service step.]

<<Begin text here>>

* + - 1. Error Handling/Recoverability

[Describe the any error handling within the batch job/service step. Describe any housekeeping steps that should be performed if the step fails to ensure the job/service can be restarted safely by the support team.]

<<Begin text here>>

# Registry Settings/INI files/.config files

[If applicable describe how Registry settings, INI file, or .config file will be used to configure the application. The example entry in the table below is from the MERT application documentation and shows one of the key values that the MertService reads from an INI file:]

<<Begin text here>>

| AppSettings Key | Value | Purpose |
| --- | --- | --- |
| EnableDebugMode | FALSE | When set to TRUE causes the Service to log entry to every function. This can be used to trace the callstack when problems occur. |
|  |  |  |
|  |  |  |

# Reporting

[If the reports have been implemented as part of the core system use this table to cross reference the Reports as specified in the logical design (Functional Spec) to the physical system components that implement them as specified in this document. For example in the System Components column list the appropriate report web page or batch job along with any stored procedures used.]

<<follow below or insert>>

|  |  |  |
| --- | --- | --- |
| Report Name | Report Description | System Components |
|  |  |  |
|  |  |  |
|  |  |  |

## <Report Name>

[If the reports have not been implemented as part of the core system complete this section for each report. First provide a brief description of the report and then complete the following sections for each report:]

<<Begin text here>>

### Dependencies

[Clearly identify any dependencies and specify in detail how the report will check that all required information is in place before executing.]

<<Begin text here>>

### Data/Metadata

[Identify data sources, reference a stored procedure if one is used to generate the data for display. If a metadata layer is used to generate the report specify the metadata structure and content in this section.]

<<Begin text here>>

### Processing/Display

[Define any processing performed on the source data before display as well as any display specific processing in this section. Provide implementation logic in narrative or pseudocode form.]

<<Begin text here>>

### Delivery/Archiving

[Provide technical detail on the report delivery mechanism as well as the archiving strategy for the report.]

<<Begin text here>>

# Notification Services

[When SQL Notification Services are used, configuration files define the notification settings. Use this section to provide sufficient information to illustrate the design of the notifications.]

## Notification Subscription

<<Begin text here>>

## Notification Generation

<<Begin text here>>

## Notification Formatting

<<Begin text here>>

# Windows Services

<<Begin text here>>

# System Integration Points

[Use this table to cross reference the system interfaces as specified in the logical design (Functional Spec) to the physical system components that implement them as specified in this document. For example in the System Components column list the appropriate batch job, DTS Package or COM object along with any stored procedures used.

<<Begin text here>>

|  |  |  |
| --- | --- | --- |
| System Name | Interface Description | System Components |
|  |  |  |
|  |  |  |
|  |  |  |

## Data Mapping

<<Begin text here>>

## Data Capture Rules

[Determine necessary business rules required to accurately capture and convert data. Determine whether data will stay compliant with existing business rules]

<<Begin text here>>

# Internationalization

*[Specify technical globalization and localization requirements and designs related to the entire application. This section should summarize all measures taken to build a globalized, localizable, and localized system or changes to the system, rather than rely on globalization and localization comments scattered throughout the document. All of the Globalization/Localization documents referenced in this section are at the* [*MSIT Globalization and Localization site*](http://sharepoint/sites/globalizeIT)*.*

*For more information about globalization and localization standards and practices within MSIT,*

* *visit* [*http://sharepoint/sites/globalizeIT*](http://sharepoint/sites/globalizeIT)
* *or contact* [*MSIT Globalization-Localization Info*](http://itlc/) *(mailto:locinfo)]*

## Globalization

*[Specify globalization and localizability requirements or designs determined through investigation of the internationalization requirements documented in the BRD and Functional Specification.]*

<<Begin text here>>

## Localization

### Languages

*[Language requirements documented in the BRD and Functional Specification.]*

|  |  |  |
| --- | --- | --- |
| **Language** | **Country** | **Culture Code (e.g., en-US)** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Market Customization

*[Feature-specific market customization details should be documented in the corresponding section of this document. In this section, include any market-specific information not documented elsewhere in the document.]*

<<Begin text here>>

# Special Considerations

[This section is a placeholder for any additional technical considerations related to the application being developed. Some sample headings are included below.]

<<Begin text here>>

## Volume Considerations

<<Begin text here>>

## Testing Considerations

<<Begin text here>>

## Production Support Considerations

<<Begin text here>>

## Integration with MSOps.com Portal

<<Begin text here>>

## Performance & Response Time

<<Begin text here>>

## Setup and Deployment Strategy

<<Begin text here>>

## Software Development Environment

### Version Control System

### Build Procedures

### Deployment Procedures

1. Glossary/ Definitions

Below is a list of common terms and their definitions that are used throughout this document:

| Term | Definition |
| --- | --- |
|  |  |
|  |  |
|  |  |

1. Related Documents/References

| Document | Document Location |
| --- | --- |
| Vision and Scope |  |
| Business Requirements Document |  |
| Functional Specification |  |
|  |  |
|  |  |

1. Environments

[Include in this section known environmental information. At design time test and production environment details may not be known.]

|  |  |  |  |
| --- | --- | --- | --- |
| Component | Development | Test | Production |
| e.g. IIS |  |  |  |
| e.g. Database |  |  |  |
| e.g. Reporting DB |  |  |  |
| e.g. Batch job group |  |  |  |
| e.g. File share |  |  |  |
| e.g. VSS Location |  |  |  |

1. : Back-End (SAP)

Additional Guidance Text has been added (*Arial GREEN italic font in brackets*) for SAP specific – labelled “REQUIRED for SAP”, (*Arial RED italic font in brackets*) non-SAP specific – labelled “NOT APPLICABLE for SAP”, and (*Arial ORANGE italic font in brackets*) for Optional SAP – labelled “AS NEEDED for SAP”.

* 1. Development Overview/Approach *[REQUIRED for SAP]*
     1. Requirements Summary

*[Please provide a high level description of the development requirement that will be addressed.]*

* + 1. Assumptions

*[Describe any assumptions that have been made in the process of completing this design. What functionality is expected of configuration or other developments that pass information to this development and/or retrieve the information processed by it (performance, triggers, exceptions, etc.)?]*

* + 1. Dependencies / Constraints

[*Indicate any dependencies or constraints that may impact development, in terms of requirements from internal or external applications or teams, limited access to legacy system, time constraints or data restrictions. Also, please specify schedule dependencies e.g. interface or batch jobs that must run prior to execution]*

* + 1. Applications, Objects or Transactions Affected

*[Please provide a list of SAP objects or transactions, or any other Applications which will be affected by the development.]*

* + 1. SAP Object Attributes

|  |  |
| --- | --- |
| **Development Class** |  |
| **Message Class** |  |
| **Program ID** |  |
| **Program Type** |  |
| **Module** |  |

* + 1. Transaction Volume

*[Please provide an indication of the expected number of records that will need to be read and displayed using this Interface]*

* + 1. Standards

*[Detail any programming, error handling and archiving standards, which will be utilized in development of the Interface.]*

* + 1. Estimation of Effort

*[Detail the amount of hours required to complete this requirement, including time for specification review prep, attending the review, writing the code, unit testing and code review.]*

* + 1. Code Management Tool

[Identify code management tool (e.g. VSS)]

SAP has an existing code management repository, which includes object version control and transport management functionality.

* 1. High Level System Architecture *[REQUIRED for SAP]*
     1. Technical Flow Diagram

*[Please insert a technical flow diagram depicting the Interface technical flow. Please include the Source and Target systems, along with the direction of data flow.]*

**Ex:**



* + 1. Technical Flow Description

*[Please provide a description of the technical flow. Where appropriate this should include selection criteria, initialization routines, data retrieval, validation, processing and manipulation.]*

<<Begin text here>>

* 1. Data Element List *[REQUIRED for SAP]*

*[Please provide a detailed listing of the data elements that will be used in this interface, including both the interface references and internal objects.]*

* 1. Interface Type *[REQUIRED for SAP]*

*[Please provide details of the interface type (Inbound/Outbound) and the proposed data load method (e.g. IDoc, Transaction Call, BAPI/RFC.]*

* 1. SAP RFC Connection Information *[REQUIRED for SAP]*

[Provide in this section the login information that will be used for connecting to SAP through this RFC. Refer to section 4.1 for details on any Web Services that may be associated with this interface.]

<<Begin text here>>

* 1. Detailed SAP Interface & Conversion Information *[REQUIRED for SAP]*
     1. Interface Parameters [Programs] *[REQUIRED for SAP Interfaces][AS NEEDED for SAP Conversions]*

[This section may be removed if the object is RFC-based and does not require file-based or selection screen parameters.]

[Provide in this section a listing of all import parameters, all import/export files including layouts and mappings between them, and all error exceptions with this SAP Interface Program. Include a screen shot as it becomes available.]

* + - 1. Selection Screen Details [Programs] [REQUIRED for SAP – Conversions, Conversions & Reporting]

*[Please attach sample selection screen layout if applicable.]*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Table field / Check Box / Radio Button** | **Type/Length or SAP Reference** | | **Select-Option(S) or Parameter(P)** | **Comments** | **Default Value** | | **Required Field?** | |
| S\_VBELN | Field | VBAP-VBELN | S | | Billing Doc # | |  | | X |
| S\_KUNNR | Field | KNA1-KUNNR | S | | Customer # | |  | |  |
| P\_START\_DATE | Field | DATS(8) | P | | Begin Date | | SY-DATUM | |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Table field / Check Box / Radio Button** | **Type/Length or SAP Reference** | | **Select-Option(S) or Parameter(P)** | | **Comments** | **Default Value** | **Required Field?** |
| P\_PATH | Field | CHAR(200) | P | | FilePath | |  | X |
| P\_IFILE | Field | CHAR(50) | P | | Input FileName | |  | X |
| P\_OFILE | Field | CHAR(50) | P | | Output FileName | |  | X |
| P\_NOTIFY | CheckBox | CHAR(1) | P | | Send Email | | X |  |

* + - 1. File Layout [REQUIRED for SAP Conversions & Interfaces]

*[Provide in this section a sample layout for any Input or Output files, including data type attributes and field names.*]

* + 1. Interface Parameters [RFCs] *[REQUIRED for SAP Conversions & Interfaces]*

*[This section may be removed if the object is program-based and does not include an RFC interface]*

[Provide in this section a listing of all import/export parameters, all import/export tables, and all exceptions associated with this Function/RFC. To maintain consistency, please include in the name of the parameters and tables IN/OUT depending on the direction of the data. Also include a screen shot as it becomes available.]

* + - 1. SAP RFC Import Parameters

[Provide in this section all of the Import Parameters needed to process this Function/RFC. Use the following table as an example.]

| **Name** | **Description** | **Type/Length or SAP Reference** | **Comments** | **Required Field?** |
| --- | --- | --- | --- | --- |
| IN\_BILL\_DOC | Billing Document Number | VBAP-VBELN |  | X |
| IN\_CUST\_NBR | Customer Number | KNA1-KUNNR |  |  |
| IN\_START\_DATE | Beginning Date | DATS(8) | Required if IN\_CUST\_NBR is blank. |  |

* + - 1. SAP RFC Export Parameters

[Provide in this section all of the Export Parameters needed to process this Function/RFC. Use the following table as an example.]

| **Name** | **Description** | **Type/Length or SAP Reference** | **Comments** | **Required Field?** |
| --- | --- | --- | --- | --- |
| OUT\_CUST\_NBR | Customer Number | KNA1-KUNNR | Provide leading zeroes. |  |
| OUT\_CUST\_NAME | Customer Name | KNA1-NAME1 |  |  |

* + - 1. SAP RFC Table Parameters [REQUIRED for SAP Interfaces][AS NEEDED for SAP Conversions]

[Provide in this section all of the table information including structures passed into and out of the Function/RFC. Tables and structures can be inbound and/or outbound. Use the following tables as examples.]

**Import Tables**

IN\_MATERIALS

| **Name** | **Description** | **Type/Length or SAP Reference** | **Comments** | **Required Field?** |
| --- | --- | --- | --- | --- |
| IN\_MATERIAL | Material Number | VBAP\_MATNR |  | X |

IN\_DOCUMENTS

| **Name** | **Description** | **Type/Length or SAP Reference** | **Comments** | **Required Field?** |
| --- | --- | --- | --- | --- |
| IN\_BILL\_DOC | Billing Document Number | VBAP-VBELN |  | X |
| IN\_DOC\_ITEM | Billing Document Item | VBAP-POSNR |  |  |

**Export Tables**

OUT\_DETAILS

| **Name** | **Description** | **Type/Length or SAP Reference** | **Comments** | **Required Field?** |
| --- | --- | --- | --- | --- |
| OUT\_BILL\_DOC | Billing Document Number | VBAP-VBELN |  |  |
| OUT\_DOC\_ITEM | Billing Document Item | VBAP-POSNR |  |  |
| OUT\_MATERIAL | Material Number | VBAP-MATNR |  |  |
| OUT\_DESCRIPTION | Material Description | MAKT-MAKTX |  |  |

*Return Detail*

| **Name** | **Description** | **Type/Length or SAP Reference** | **Comments** | **Required Field?** |
| --- | --- | --- | --- | --- |
| BAPIRET2 | Return structure | BAPIRET2 | Standard return structure for SAP BAPIs |  |

\* Required Export Table for RFCs wrapped with a Web Service.

The above return structure (or similar) is required for all RFCs that will be wrapped with a Web Service.  The purpose of this structure is to allow a return code with corresponding text description to be passed from the RFC into the Web Service.  This will aid the SAP Technical Support team to debug/troubleshoot Web Service issues in production without having to immediately request that the SAP code be run through the debugger; a specific return code/description will allow the support resource to quickly narrow the focus to resolve the issue.

* + - 1. SAP RFC Exceptions [REQUIRED for SAP – All Developments]

[Provide in this section all of the Exceptions and Return Codes that could result through the processing of this Function/RFC. Use the following table as an example.]

| **Return Codes** | **Exceptions** |
| --- | --- |
| 0 | Success |
| 1 | Required Field Missing |
| 2 | No Data Returned |

* + 1. High-level processing flow & Developer comments *[REQUIRED for All Development Objects]*

*[Please provide details on the processing flow/logic that the developer should implement, including any specific concerns that should be considered when coding the solution.]*

*Topics include:*

* *Validation logic*
* *Table joins*
* *Information lookups (including common functions to utilize)*

*Example:*

***Inbound Interface:***

1. *Open input file / read input parameters for an RFC*
2. *Perform authority check. [provide Auth Check info]*
3. *Make sure all the required fields are populated.*
4. *Perform data validations.*
5. *Do look ups and gather all the necessary data required to do a function call or call transaction.*
6. *Do the necessary formatting for date fields and currency fields as needed.*
7. *Perform function call or call transaction.*
8. *Error processing: either send out an email or write out error data*
9. *At the end output data and write status message to screen in case of programs*.
   * 1. Processing and Operational Considerations
     2. Execution Mode [Real-time, Batch] *[REQUIRED for SAP]*
     3. Dependencies *[REQUIRED for SAP]*

*[Please provide details of any dependencies for the Interface to process correctly and perform at its optimum.]*

* + 1. Batch Requirements

*[Include details required for batch processing (i.e. variant set-up, reconciliation requirements, etc.)]*

* *Submission*
* *Scheduling Considerations*
* *Run Frequency*
* *Start Date*
* *Start Time*
* *Estimated Volume Per Run*
* *Parameters / File Dependencies*
* *Job Dependencies*
* *Constraints*
* *Variant Required*
  + *Field Text*
  + *Suggested Value*
  + *Description*
  1. Security and Authorization *[REQUIRED for SAP]*

[Provide in this section the account and processing requirements, authority checks, and authorization group attributes to be used for this Interface Program. If you don’t know the authorization groups attribute to use, please contact a member of the SAP Security Team who will help determine the correct one along with any other security items needed for this Interface Program.

When adding an authorization groups attribute (requested by the Security Development Team) and the entry doesn't yet exist in TPGP please use SE16 to create the entry and add that single record to your transport: R3TR - TABU - TPGP with full key.

* + 1. Account Requirements

[Provide in this section a list of batch accounts, users, or the appropriate security role(s) of users who need access to run the program (or who should not have access).]

* + 1. Processing Requirements

*[Provide in this section the specific security processing requirements including enabling/disabling screen controls based on authorizations, granting access to data only from a specific region, etc.]*

* + 1. Authority Checks

*[Provide in this optional section the security object along with the field values to check based on the above requirements.]*

* + 1. Authorization Group Attribute

*[Provide in this section the authorization group attribute to use for this Interface Program. This section is required.]*

* + 1. Threat Modeling

[Provide in this section the security vulnerabilities in the design. A Threat Model should be provided for all v1.0 applications and for any existing applications with significant functionality changes. If there are no security vulnerabilities in the design, please leave this section blank.]

* 1. External Programs

*[List all external programs and function modules called. This will not be common. Ex: calling an external RFC for Credit Card processing validation.]*

* 1. Technical SAP Requirements *[REQUIRED for SAP]*
     1. SAP Database Objects (created/changed)

[The following table lists the database objects, their system location and a brief description. SAP Database object types include tables, structures, views and indexes.]

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Database Location | Description |
| e.g. ZTAB | Table | R/3 (MS1, MST, MSS) | New Custom Table |
| e.g. KNA1-Z001 | Index | R/3 (MS1, MST, MSS) | New Index |
|  |  |  |  |

* + 1. Custom SAP Tables

[The table structures should be included in the Database Schema. Include for each table it’s name, purpose and structure. Please indicate what master keys or systems this data may have come from.]

<<Begin text here>>

* + 1. Custom SAP Views

[The views should be included in the Database Schema. Include for each view it's name, purpose and structure.

<<Begin text here>>

* + 1. SAP Table Indexes
       1. <Table Name/Index Name>

[For each index describe its purpose and type, list the fields indexed.

<<Begin text here>>

* + 1. Testing Considerations for SAP
* Test for successful generation and delivery of error/success logs and messages.
* If a status email is required, ensure it is sent to the appropriate contact(s).
* Test for normal daily/weekly/month end volumes of X records.
* Validate field values and formatting.
* For inbound processes, validate calling transaction.
* Test for incomplete processing where processed row count and/or checksum values do not equal exceptions.
* Expected performance characteristics: should run to completion in X minutes with Y volume.

1. Document Change History

| Version No. | Date | Name (Alias) | Description of Change |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Reviewers

Below is a list of the project team members and required reviewers

|  |  |  |  |
| --- | --- | --- | --- |
| Person | Role | Contact | Reviewed Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |