00095

|  |
| --- |
| **System Name>**  **<Change Control Reference Number>** |

**Table of Contents**

Glossary of Terms 3

Abbreviations 3

Definitions 3

1 Introduction 4

2 Business Requirements 4

3 High-Level Design 4

3.1 System Architecture 4

4 Functional Requirements 4

4.1 Data Requirements 4

4.2 Configuration 5

4.2.1 Configuration Step <n> 5

4.3 Development 5

5 Non-Functional Requirements 6

5.1 Transaction Volume 6

5.2 Performance Requirements 6

5.3 Security Requirements 6

5.4 Software Quality Attributes 6

5.5 History Requirements 6

5.6 Archiving Requirements 6

5.7 Frequency 6

5.8 Dependencies 6

5.9 Access Rights (Roles and Profiles) 6

6 Other Requirements 7

7 Interface Requirements 7

7.1 Interface Data Structure 7

7.1.1 Interface Data Schema 7

7.1.2 Sample file 7

7.2 User Interfaces 7

7.3 Hardware Interfaces 8

7.4 Software Interfaces 8

7.5 Communications Interfaces 8

7.6 Module Interfaces 8

8 Testing Requirements 8

8.1 Positive Test Scenarios 8

8.2 Negative Test Scenarios 8

8.3 Regression Test Scenarios 8

9 Documents to be updated 8

10 References 9

10.1 Reference Documentation (include reference to test pack) 9

11 Functional Specification Signoff 9

# Glossary of Terms

## Abbreviations

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| RFS | Request for service |
|  |  |
|  |  |

## Definitions

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| System | It can be application, or middleware |
|  |  |
|  |  |

N.B\* All the text highlighted in red must be replaced with the actual details when this document is used.

# 

# Introduction

The Functional Specification clearly defines what the functionality of the requested change will be by providing all the information that is necessary to design the requested change. The functional specification should describe the business requirements, the input, processing and output required to meet these User Requirement Specification.

# Business Requirements

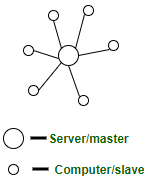
<Provide a detailed description of the business requirements as stated in section A of the change control form. Define what is in scope and what is not covered in scope and document any activities/processes that this change is dependent on.>

1. **Purpose** - <What the function is intended to accomplish?>
2. **Input** - <What inputs will be accepted, in what format the inputs arrive, sources for the inputs, and other input characteristics?>
3. **Process** - <The steps to be performed, algorithms, formulas, or techniques to be used. However, Software implementation details are not included;> and
4. **Output** - <Desired outcomes such as the output form (e.g. report layout), the destination of the output, output volume and timing, error handling procedures, and units of measure.>

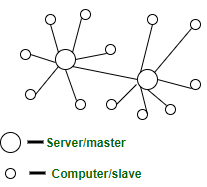
Purpose of this functional specification is to describe how to avoid, loss or lack of data not re-entering, after a momenteous availability and locality in that detail to a transaction falls away, and can now be retained for several business cases.

General description that led to finding the missing functionality stems from systems progress, from centralized to de-centralized and then distributed. In most cases systems designs address their own capability, and target architecture and infrastructure. Transaction recording or then detail thereto becomes available again irrespective of all three types of systems, enabling a fourth.

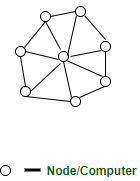
* Centralized systems are systems that use client/server architecture where one or more client nodes are directly connected to a central server. This is the most commonly used type of system in many organizations where a client sends a request to a company server and receives the response.



* In decentralized systems, every node makes its own decision. The final behavior of the system is the aggregate of the decisions of the individual nodes. Note that there is no single entity that receives and responds to the request.



* In decentralized systems, every node makes its own decision. The final behaviour of the system is the aggregate of the decisions of the individual nodes. Note that there is no single entity that receives and responds to the request.



# High-Level Design

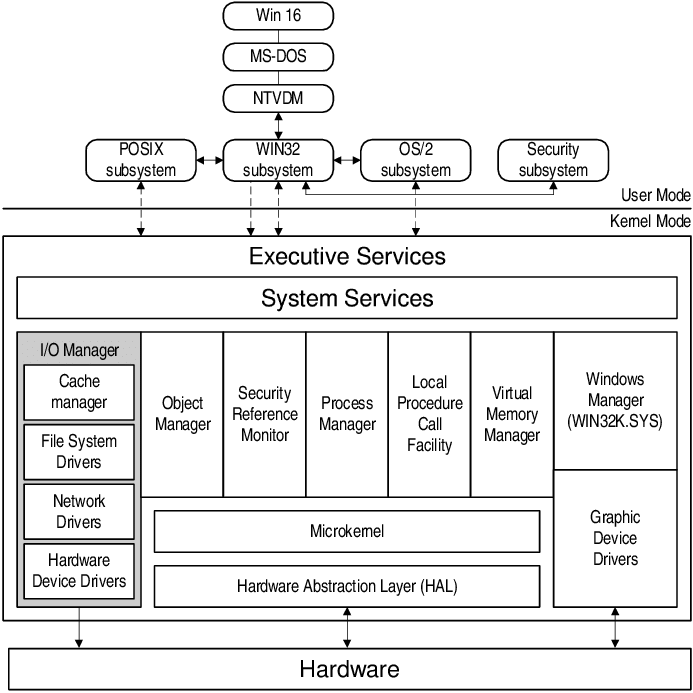
<This section should set out the high-level design of the system. It should include one or more system models showing the relationship between system components and the systems and its environment.

Refer back to the Investigation and Impact Analysis Document and complete the required architecture sections below. Section 3.1 System Architecture can be used as an example. >

Windows Services are non-UI software applications that run in the background. Windows services are usually started when an operating system boots and scheduled to run in the background to execute some tasks. Windows services can also be started automatically or manually. You can also manually pause, stop and restart Windows services.

## System Architecture

<This section describes a high-level overview of the anticipated system architecture showing the distribution functions across system modules and the relationship between them. System architecture is a representation of the system change to be developed, the process and discipline for effectively implementing the design(s) for such a system change.>



# Functional Requirements

<This section contains all the system requirements at a level of detail sufficient to enable designers to design a change to satisfy those requirements, and testers to test that the change satisfies those requirements. These requirements should include at a minimum a description of every input into the system, every output (response) from the system and all functions performed by the system in response to an input or in support of an output.>

Incorporation of Area Based Data, using SCIE (Service Communication Information Exchange) Service

**Functionalities:**

First Functionality :

* ‘NT Service’ Provide creation of lines from Gui’s

- Sub Functionality: (TCP)

* Lines to pair as a list, locality for both added.

Second Functionality :

* Service between Parties (Toobar Button Activated)
  + Generic Object Services (BC-SRV-GBT)

The generic Object Services offer functions that are available in various SAP applications

* + Web Services Tools

The JST Web services component contains tools for developing and interacting with Java Web

services.

Third Functionality:

* Z<PROGRAM> Provide search for Services-Rendered, Items-Sold per Category.

## Data Requirements

<This section defines a complete list of Data requirements. This would include any master data or processing data that needs to be defined beforehand; cleansed and/or uploaded (migrated) as part of this change request.

## 

## Configuration

<This section contains all the Configuration steps of the change request. For each of the configuration steps the following must be specified:

* Purpose
* Menu path / transaction
* Pre-requisites
* Current settings
* Proposed settings>

### Configuration Step <n>

<This section contains the individual configuration step details>.

#### Purpose

<This section describes the purpose of this configuration step from both the system and business process context>.

#### Menu Path /Transaction Code

|  |  |
| --- | --- |
| **Via Menus** | SAP Customizing Implementation Guide 🡪Cross-Application Components 🡪 Quality Issue Management 🡪 Basic Settings🡪 Generic Settings for Reference Objects🡪Define Object Types |
| **Via Transaction Code** | SPRO |
| **Table updated** | See transaction SE10 |

#### Pre-requisites

<This section describes any pre-requisite configuration or data requirements that will be needed before this configuration step can be completed

#### Current settings

<This section describes the current settings before any changes are made emanating from this change request

#### Proposed changes

<This section describes the proposed configuration changes that this change request will implement

## Development

Specify any requirements regarding objects to be developed, programming that needs to be changed or created. The categories relevant for this section are any of the following (RICEFW):

* Reports
* Interfaces
* Conversion programs
* Enhancements
  + User exits
  + Enhancement sport
  + Custom tables / screens
* Forms
* Workflow

Development Phases:

1. Departmental
   1. Terminal
2. Business

b) Mobile Phones

c) Cash Registers

To effect the SDLC of SCIE Service, it is recommended to Bring an Application of Choice to participate

(see below).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| B2C | B2B | Courrier | Parts/Spares | Accounting | HoursWorked | App |
|  | Web-Page |  |  |  |  |  |
|  | Excel |  |  |  |  |  |
|  | Java |  |  |  |  |  |
|  | CashRegs |  |  |  |  |  |
|  | Sap Gui |  |  |  |  |  |
|  | Py |  |  |  |  |  |
|  | WinForms |  |  |  |  |  |
|  | Adobe |  |  |  |  |  |
|  | Nix\* |  |  |  |  |  |
| Web-Page |  |  |  |  |  | iOS |
| Excel |  |  |  |  |  | Android |
| Java |  |  |  |  |  | Win |
| CashRegs |  |  |  |  |  | Blackberry |
| Sap Gui |  |  |  |  |  | HUAWEI |
| Py |  |  |  |  |  | LG |
| WinForms |  |  |  |  |  | .. |
| Adobe |  |  |  |  |  |  |
| Nix\* |  |  |  |  |  |  |
|  |  |  |  |  |  | Plant |

# Non-Functional Requirements

## Transaction Volume

<Specify data volume requirements>

Deliverables from NT Service, SCIE :

* Line from Device, Line from Device
* List of Lines Sided, locality added
* Pairing of Lines

Deliverables after data Incorporation :

* Detail on Detail available 90/90 data retention via Z<PROGRAM>, providing search functionality.

Deliverable Toolbar Activation :

* Correspondence Inter Apllication, regarding Notes including CPA, or PSP-TASK

Deliverable Stack Based : (Not yet Available)

* Generate Tasks from Incorporated data, workload assessment.
* Follow-up
* Update Query
* Corrections
* Errors requiring Attention
* Route for Tax

## Performance Requirements

<If there are performance requirements for the requested change under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding the implementation of the requested change or protection of the data used or created by the change. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the system. Define any security or privacy certificat  
ions that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the requested change that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## History Requirements

<Specify history requirements, what historical period data must be available and the level of granularity required (daily/monthly)>

## Archiving Requirements

<Specify archiving requirements. How old can data be before it is archived? Does archived data need to be accessible?>

## Frequency

<Specify frequency requirements>

## Dependencies

<Specify systems that will be impacted by this change and the impact.>

## Access Rights (Roles and Profiles)

<Specify roles and profiles requirements. Define the permissions for each role that is impacted. >

# Other Requirements

<Define any other requirements not covered elsewhere in the functional specification. This might include database requirements, legal requirements, reuse objectives, and so on. Add any new sections that are pertinent to the requested change.>

# Interface Requirements

## Interface Data Structure

|  |  |
| --- | --- |
| Interface ID: | <Integration team to add interface ID> |
| Source System: | <Name> |
| Target System: | <Name> |
| Integration Mechanism: | <ALE, JCAPS, PI, etc.> |
| Interface Name: | <example ZPMRMIGO> |
| Interface Description: | <Describe the purpose of the interface> |
| Data Format: | <example idoc, flat file> |
| Direction: | <Inbound or Outbound> |

### Interface Data Schema

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Field Description** | **Data Type** | **Length** | **Mandatory** | **\*Business Rule** |
| AUFNR |  | String | 12 | Yes | 1 |
| WARPL |  | String | 12 | Yes | 1 |
| WAPOS |  | String | 16 | No | 2 |
| KTEXT |  | String | 40 | Yes | 3 |
| TPLNR |  | String | 18 | Yes | 3 |
| EQUNR |  | String | 30 | Yes | 4 |
| VAPLZ |  | String | 8 | Yes | 5 |

<Note: The table above is an example. Please remove the text and complete the table as far as possible.>

***Business Rules*** (\*Define business rules if there is any logic required in the mapping)

1. <Description of the business rule>
2. <Description of the business rule>
3. <Description of the business rule>

### Sample file

<Embed sample file of the message if available i.e. standard idoc, flat file structure, etc.>

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed.>

## Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## Software Interfaces

<Describe the connections between this software product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Identify data that will be shared across software components.>

## Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

## Module Interfaces

<Describe the connections between modules within a software product.>

# Testing Requirements

<This section describes the test data required for testing the requested change and the conditions that must be satisfied by the change. Refer to the testing requirements document.>

## Positive Test Scenarios

|  |  |  |
| --- | --- | --- |
| **Condition #** | **Test Description** | **Test Case Reference** |
|  |  |  |
|  |  |  |

## Negative Test Scenarios

|  |  |  |
| --- | --- | --- |
| **Condition #** | **Test Description** | **Test Case Reference** |
|  |  |  |
|  |  |  |

## Regression Test Scenarios

|  |  |  |
| --- | --- | --- |
| **Module** | **Description of impact** | **Test Case Reference** |
|  |  |  |
|  |  |  |

# Documents to be updated

<Business processes, training documents etc.>

# References

<List all documents referenced elsewhere in the functional specification. This information can be provided by reference to an appendix or to another document. As a minimum the Process Control Manual (PCM) needs to be referenced>

Library: SCIE Service library type ‘NT Service’ Service.csproj available on request.

## Reference Documentation (include reference to test pack)

|  |  |
| --- | --- |
| **Name** | **Location** |
|  |  |
|  |  |

# Functional Specification Signoff

This document was reviewed and approved in terms of accuracy and completeness by the following stakeholders:

**Required Approvals:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Name** | **Signature** | **Date** |
| **Specialist** |  |  |  |
| **Functional / Team Lead** |  |  |  |

**Optional Approvals:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Name** | **Signature** | **Date** |
| **<Insert Role>** |  |  |  |
| **<Insert Role>** |  |  |  |