Functional Specification Document

# Index

1. Document Information 1

2. Introduction 1

3. Business Requirement Overview 1

4. Business Process Flow 1

5. Functional Scope 1

6. Functional Solution Approach 1

7. Functional Requirements 1

8. Interfaces & Integration 1

9. Output 1

10. UI Requirement 1

11. Authorization & Security 1

12. Error Handling & Notifications 1

13. Assumptions & Dependencies 1

14. Test Scenarios 1

15. Sign-Off 1

# 1. Document Information

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Document Title | Project Name | Client Name | Prepared By (Author, Department) | Document Version & Date |
| ZRCOPY\_SAMPLE\_ECC\_CODE\_V1 Technical Specification |  |  | PWC AI Asset |  |

# 2. Introduction

This document provides the technical specification for the SAP ABAP program ZRCOPY\_SAMPLE\_ECC\_CODE\_V1. It outlines the program's purpose, which is to facilitate data retrieval and processing from various SAP tables using modular subroutines. The scope of this document includes the definition of data structures, selection screens, and the sequence of subroutine executions for comprehensive data extraction and manipulation. The intended audience for this document includes SAP ABAP developers, technical consultants, and project stakeholders involved in the development, review, or maintenance of this program.

# 3. Business Requirement Overview

The business requirements focus on enabling efficient data extraction and processing from multiple SAP standard and custom tables. The objective is to provide a modular and reusable ABAP program that can retrieve, validate, and process data related to materials, deliveries, billing, financials, and other business entities. The solution aims to streamline data handling through well-structured subroutines, ensuring clarity, maintainability, and scalability in the data processing workflow.

# 4. Business Process Flow

No business process flow details are provided in the payload.

# 5. Functional Scope

1. In-Scope items

1. The program ZRCOPY\_SAMPLE\_ECC\_CODE\_V1 is designed to perform a comprehensive set of data retrieval and processing tasks using ABAP. The following functionalities are included within the scope:

- Definition and initialization of selection screen parameters and select-options for plant (p\_werks) and material numbers (s\_matnr), allowing user-driven data filtering.

- Declaration and setup of internal tables and structures for handling data from various SAP standard tables such as T001W (plant), MARA (material master), LIPS (delivery items), BSEG (accounting), and others.

- Modularization of logic using includes and subroutines (FORM routines) for maintainability and reusability.

- Fetching and validation of plant data based on user input (fetch\_and\_check\_plant).

- Retrieval of material master data from MARA based on selected material numbers (fetch\_material).

- Extraction of delivery item data from LIPS, filtered by material and plant (fetch\_delivery\_items).

- Retrieval of condition records from PRCD\_ELEMENTS (fetch\_konv).

- Fetching billing document header (VBRK) and item (VBRP) data, specifically non-draft documents (fetc\_vbrk, fetch\_vbrp).

- Extraction of financial document data from ACDOCA (fetch\_bsak).

- Retrieval of customer master data from KNA1 (fetch\_j1m0cust).

- Fetching of commodity code and related details from MARC and via /SAPSLL/CL\_MM\_CLS\_SERVICE class (fetch\_marc\_stawn).

- Retrieval of condition counters from PRCD\_ELEMENTS (fetch\_dzaehk).

- Extraction of business place data from P\_BusinessPlace (fetch\_jbbranch).

- Counting of sales document headers in VBAK (fetch\_vbuk).

- Fetching material and storage location data from MARC and MARD, including substring operations on material numbers (fetch\_marc\_mard).

- Data retrieval and processing from MARA with substring and ordering logic, and message construction based on material number patterns (fetch\_orderby).

- Fetching a single material number from MARC based on a substring of another material number (fetch\_single).

- Preparation of final output data by processing delivery items and populating a final internal table (prepare\_final\_data).

- Population of salary-related data by extracting monetary values from ACDOCA (populate\_salary).

- Definition and implementation of a local class (lcl\_data) with a public data variable and method for data encapsulation and processing.

- Use of modular includes for code organization (e.g., selection screen, top-of-program declarations, subroutine implementations).

2. Out-of-Scope items

1. The following functionalities are explicitly excluded from the scope:

- Any user interface beyond the standard SAP selection screen (no custom ALV grids, web UIs, or Fiori apps).

- Data manipulation or updates to the database (no INSERT, UPDATE, or DELETE operations are performed; all logic is read-only).

- Error handling, exception management, and user messaging beyond basic ABAP statements (no advanced logging or notification mechanisms).

- Authorization checks or user validation logic.

- Integration with external systems or non-SAP data sources.

- Advanced business logic such as pricing calculations, workflow triggers, or document posting.

- Performance optimization techniques (e.g., parallel processing, buffering) beyond standard Open SQL usage.

- Detailed implementation of the methods in the lcl\_data class (only structure and a simple CLEAR operation are provided).

- Any enhancements, BADIs, user exits, or modifications to standard SAP behavior.

- Output formatting, reporting, or printing logic (no explicit output routines or layouts are defined).

- Handling of language translations or multi-language support for text elements.

- Security, encryption, or data masking features.

# 6. Functional Solution Approach

The business requirement is addressed by designing an ABAP program that orchestrates a sequence of data retrieval and processing steps, leveraging SAP standard tables and modular subroutines. The solution begins with a user-friendly selection screen, allowing users to specify a plant and a range of material numbers, ensuring that only relevant data is processed.

Upon execution, the program validates and fetches plant data, ensuring the provided plant code exists and is valid. It then retrieves material master data based on the user's selection, followed by fetching delivery item data that matches the selected materials and plant. The program continues by extracting condition records, billing document headers and items, vendor open items, customer data, and material storage location details from respective SAP tables. Specialized subroutines handle the retrieval of number ranges, branch information, and sales document statuses, as well as material and storage location data.

Throughout the process, the program utilizes internal tables and structures to organize and temporarily store data, ensuring efficient processing and clear separation of concerns. Subroutines are used to encapsulate logic for data fetching, transformation, and validation, promoting code reusability and maintainability. The final data is prepared by consolidating relevant fields into a dedicated internal table, ready for output or further processing. Additional logic, such as salary population and message construction, is included to support specific business scenarios.

Overall, the functional solution approach ensures a robust, modular, and scalable program that meets the business requirement for comprehensive data extraction, validation, and preparation from multiple SAP sources, all driven by user input and executed in a logical, stepwise manner.

# 7. Functional Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement ID | Requirement Description | Business Rule | Priority | Comments |
| To Be Filled | To Be Filled | To Be Filled | To Be Filled | To Be Filled |

# 8. Interfaces & Integration

No interface details available.

# 9. Output

Based on the provided payload, the program 'ZRCOPY\_SAMPLE\_ECC\_CODE\_V1' is structured to perform a series of data retrieval and processing tasks through multiple subroutines. However, there is no explicit mention of specific reports, extracts, or output files being generated (such as Excel, PDF, CSV, or spool outputs), nor is there any reference to output destinations like SAP AL11, email, or spool.

The program prepares and processes data from various SAP tables (such as MARA, LIPS, T001W, VBRK, VBRP, ACDOCA, KNA1, PRCD\_ELEMENTS, P\_BusinessPlace, MARC, MARD, etc.) and organizes it into internal tables (e.g., ilips, imara, ikonv, ifinal). The subroutines fetch, process, and prepare data, but the payload does not specify any final output format, layout, or destination for the processed data.

Functionally, the purpose of the program is to extract, process, and prepare datasets from multiple SAP tables for further use within the program or for subsequent processing steps. However, without explicit details in the payload, there is no information about user-facing reports, file extracts, or automated outputs being generated or delivered to any destination.

If output requirements exist, they are not described in the provided payload.

# 10. UI Requirement

The selection screen consists of the following UI elements:

1. Field Name: p\_werks

- Type: Input (Single-value)

- Data Element: t001w-werks

- Description: Single-value input field for plant (WERKS) based on table T001W.

- Default Value: None specified.

- Mandatory: Not specified.

- Business Purpose/Validation: Used to input a specific plant code. No additional validation or dependencies mentioned.

2. Field Name: s\_matnr

- Type: Input (Range)

- Data Element: mara-matnr

- Description: Range input for material number (MATNR) based on table MARA.

- Default Value: None specified.

- Mandatory: Not specified.

- Business Purpose/Validation: Allows users to specify a range of material numbers for selection. No additional validation or dependencies mentioned.

No other user interactions, defaults, or validation logic are specified in the provided payload.

# 11. Authorization & Security

|  |  |  |  |
| --- | --- | --- | --- |
| Role/Profile | Authorization Object | Access Level | Description |
| [To Be Filled] | [To Be Filled] | [To Be Filled] | [To Be Filled] |

# 12. Error Handling & Notifications

No error handling or notification requirements specified.

# 13. Assumptions & Dependencies

No assumptions or dependencies specified.

# 14. Test Scenarios

[ERROR: Section content missing]

# 15. Sign-Off

|  |  |  |  |
| --- | --- | --- | --- |
| Role | Name | Signature | Date |
| Prepared By |  |  |  |
| Approved By |  |  |  |
| Client Sign-Off |  |  |  |

Document generated by PWC AI-powered ABAP Tech Spec Assistant.