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 Scenario 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 | ZRCOPY\_SAMPLE\_ECC\_CODE\_V1 Implementation |  | PWC AI Asset |  |

# 2. Introduction

This document provides the technical specification for the SAP ABAP program ZRCOPY\_SAMPLE\_ECC\_CODE\_V1. The purpose of this document is to outline the structure, logic, and data processing steps implemented in the program, which is designed to retrieve and process data from various SAP standard tables such as MARA, LIPS, T001W, and others. The scope of this document includes the explanation of selection screen elements, data declarations, subroutine logic, and the overall program flow as implemented in the provided ABAP code and includes. The intended audience for this document comprises SAP ABAP developers, technical consultants, and project stakeholders who require a detailed understanding of the program's technical design and functionality.

# 3. Business Requirement Overview

The business requirement addressed by the ZRCOPY\_SAMPLE\_ECC\_CODE\_V1 program is to facilitate comprehensive data extraction and processing from key SAP tables related to materials, deliveries, billing, financials, and organizational data. The program aims to provide a modular and reusable solution for retrieving, validating, and preparing data sets for further business processing or reporting. The objectives include enabling users to select plant and material ranges, fetch relevant master and transactional data, perform necessary validations, and prepare final datasets for downstream consumption. The solution addresses the need for efficient, structured, and automated data handling across multiple SAP modules, supporting business processes that rely on accurate and timely information retrieval.

# 4. Business Process Flow

Start

-> User enters selection criteria (Plant and Material Numbers) on the selection screen

-> System validates plant input and fetches plant data (fetch\_and\_check\_plant)

-> System fetches material master data based on selected materials (fetch\_material)

-> System retrieves delivery item data for selected materials and plant (fetch\_delivery\_items)

-> System fetches condition records (fetch\_konv)

-> System retrieves billing document header (fetc\_vbrk)

-> System retrieves billing document items (fetch\_vbrp)

-> System fetches financial document data (fetch\_bsak)

-> System fetches customer master data (fetch\_j1m0cust)

-> System fetches material storage and commodity code data (fetch\_marc\_stawn)

-> System fetches condition counter data (fetch\_dzaehk)

-> System fetches business place/branch data (fetch\_jbbranch)

-> System counts sales document headers (fetch\_vbuk)

-> System fetches material and storage location data (fetch\_marc\_mard)

-> System performs additional data ordering and processing (fetch\_orderby)

-> System prepares final output dataset (prepare\_final\_data)

-> System populates salary-related data (populate\_salary)

-> End

# 5. Functional Scope

1. In-Scope items

1. The program 'ZRCOPY\_SAMPLE\_ECC\_CODE\_V1' is designed to perform comprehensive data retrieval and processing tasks within the SAP ECC environment. The following functionalities are included in scope:

- Definition and handling of selection screen parameters, specifically for plant (p\_werks) and material numbers (s\_matnr), allowing users to filter data based on these criteria.

- Data declarations and setup of internal tables for storing and processing data related to materials (mara), deliveries (lips), conditions (konv), plants (t001w), and other relevant SAP tables.

- Fetching and validation of plant data from T001W based on user input.

- Retrieval of material master data from MARA filtered by user-selected material numbers.

- Extraction of delivery item data from LIPS, filtered by materials and plant, and organization of this data into internal tables for further processing.

- Fetching of condition records from PRCD\_ELEMENTS (representing KONV data) and storing them in internal tables.

- Retrieval of billing document header (VBRK) and item (VBRP) data, specifically filtering out draft documents.

- Extraction of financial document data from ACDOCA, including company code, fiscal year, document number, line item, GL account, amounts, currency, and posting date.

- Fetching of customer master data from KNA1, specifically customer numbers, and sorting them for further use.

- Retrieval of business place data from P\_BusinessPlace, including company code and business place, and organizing this data.

- Fetching of commodity code and related details from MARC and via class /sapsll/cl\_mm\_cls\_service, including calls to methods for commodity code classification and details.

- Extraction of condition counters from PRCD\_ELEMENTS for further processing.

- Counting of sales document headers in VBAK for reporting or validation purposes.

- Retrieval of material and storage location data from MARC and MARD, including substring operations on material numbers.

- Data processing routines such as sorting, substring extraction, and conditional message construction based on material numbers and types.

- Preparation of final data sets by transferring and organizing relevant fields from delivery items to a final internal table, counting entries, and clearing tables as needed.

- Population of salary-related data by extracting monetary values from ACDOCA.

- Modularization of logic using subroutines (FORM routines) and local classes (e.g., lcl\_data) for encapsulation and reusability.

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 UI or web-based front-end).

- Data manipulation or updates to the underlying SAP tables; all operations are read-only and focused on data retrieval and internal processing.

- Error handling, logging, or exception management beyond basic system checks (e.g., sy-subrc checks without detailed error processing).

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

- Advanced business logic or calculations beyond the described data fetching, substring operations, and simple conditional checks.

- Output formatting, reporting, or data export functionalities (e.g., ALV grid, file download) are not included.

- Authorization checks, security validations, or user-specific data restrictions.

- Performance optimization, parallel processing, or handling of large data volumes beyond standard internal table operations.

- Any enhancements, user exits, BADIs, or custom extension points.

- Implementation of detailed business rules or validations not explicitly described in the provided logic.

# 6. Functional Solution Approach

The business requirement is addressed by designing an ABAP program that facilitates the retrieval, processing, and preparation of data from various SAP standard tables, based on user input from a selection screen. The functional solution begins with a selection screen that allows users to specify a plant and a range of material numbers, ensuring that only relevant data is processed.

Upon execution, the program sequentially performs a series of data retrieval operations through modular subroutines. These subroutines fetch plant details, material master data, delivery items, pricing conditions, billing documents, financial postings, customer and branch information, and other relevant master and transactional data. Each subroutine is responsible for extracting and validating data from its respective source, ensuring data integrity and completeness.

The retrieved data is then processed and transformed as needed, including filtering, sorting, and mapping to internal tables structured to mirror the underlying database tables. Subroutines also handle specific business logic, such as substring extraction for material numbers, conditional checks, and the preparation of final datasets for reporting or further processing. The program concludes by preparing the final output dataset, ensuring that all required information is consolidated and ready for downstream consumption or reporting, thereby meeting the business requirement in a structured and efficient manner.

# 7. Functional Requirements

[Error: Section Functional Requirements not found in LLM output.]

# 8. Interfaces & Integration

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Interface ID | Source System | Target System | Data/Message Type | Frequency/Mode | Description |  |
|  | -------------- | -------------- | -------------- | ------------------- | ---------------- | ------------- |  |

# 9. Output

[Error: Section Output not found in LLM output.]

# 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 (Plant)

- Default Value: None specified

- Mandatory: Not specified

- Business Purpose/Validation: Used to input a single plant value. The field is based on the WERKS field from table T001W. No additional validation or dependencies are mentioned.

2. Field Name: S\_MATNR

- Type: Input (Range/Multiple values)

- Data Element: MARA-MATNR (Material Number)

- Default Value: None specified

- Mandatory: Not specified

- Business Purpose/Validation: Used to input a range of material numbers. The field is based on the MATNR field from table MARA. No additional validation or dependencies are mentioned.

No further user interactions, default values, 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

# 13. Assumptions & Dependencies

# 14. Test Scenario

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Objective | Input Data | Expected Output | Actual Result/Status | Sign-off/Comments |
| TC01 | Validate plant selection screen input and plant data fetch | p\_werks = '1000' | Plant data for WERKS '1000' is fetched from T001W and stored in i\_t001w |  |  |
| TC02 | Validate material number range selection and material data fetch | s\_matnr = ['MAT1', 'MAT2', 'MAT3'] | Material data for MATNR in ['MAT1', 'MAT2', 'MAT3'] is fetched from MARA and stored in imara |  |  |
| TC03 | Fetch delivery items for selected materials and plant | p\_werks = '1000', s\_matnr = ['MAT1', 'MAT2'] | Delivery items from LIPS for MATNR in ['MAT1', 'MAT2'] and WERKS = '1000' are fetched and stored in ilips |  |  |
| TC04 | Prepare final data from delivery items | ilips populated with delivery items | ifinal internal table is populated with vbeln, posnr, matnr, werks from ilips |  |  |
| TC05 | Fetch condition records from prcd\_elements | - | ikonv internal table is populated with knumv from prcd\_elements, sorted by knumv |  |  |
| TC06 | Fetch billing document header where draft is empty | - | lv\_vbeln is populated with vbeln from VBRK where draft = space |  |  |
| TC07 | Fetch billing document items where draft is empty | - | lt\_vbrk internal table is populated with vbeln, posnr from VBRP where draft = space |  |  |
| TC08 | Fetch accounting document data from ACDOCA | - | lt\_data internal table is populated with bukrs, gjahr, belnr, buzei, hkont, dmbtr, wrbtr, waers, budat from ACDOCA, ordered accordingly |  |  |
| TC09 | Fetch customer numbers from KNA1 | - | lt\_data internal table is populated with kunnr from KNA1, ordered by kunnr |  |  |
| TC10 | Fetch MARC stawn and call commodity code services | - | ls\_marc is populated with stawn, expme from MARC; /sapsll/cl\_mm\_cls\_service methods are called |  |  |
| TC11 | Fetch condition counter from prcd\_elements | - | lv\_dzaehk is populated with condition\_counter from prcd\_elements |  |  |
| TC12 | Fetch business place data from P\_BusinessPlace | - | lt\_data internal table is populated with bukrs, branch from P\_BusinessPlace, ordered by bukrs, branch |  |  |
| TC13 | Count number of entries in VBAK | - | lv\_vbak\_cnt is populated with count of entries in VBAK |  |  |
| TC14 | Fetch MARC material numbers and MARD lsobs | lv\_matnr = 'MAT1234567' | lt\_data is populated with matnr from MARC where matnr = substring(lv\_matnr,3,4); lv\_lsobs is populated with lsobs from MARD |  |  |
| TC15 | Fetch material data with ordering and message construction | lv\_matnr, lv\_mtart set; lv\_matnr\_chk4 = '1234' | lt\_table is populated with matnr, mtart, matkl from MARA; message is constructed if lv\_matnr\_chk4 = '1234' |  |  |
| TC16 | Fetch single MARC material number using substring | lv\_matnr = 'MAT1234567' | lv\_marc\_matnr is populated with matnr from MARC where matnr = substring(lv\_matnr,3,3) |  |  |
| TC17 | Populate salary from ACDOCA | acdoca-dmbtr = 1000 | lv\_salary is set to 1000 |  |  |

# 15. Sign-Off

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

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