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 |  | PWC AI Asset |  |

# 2. Introduction

This document serves as the technical specification for the SAP ABAP program "ZRCOPY\_SAMPLE\_ECC\_CODE\_V1". The purpose of this document is to provide a comprehensive overview of the program's structure, logic, and data processing routines as implemented in the ECC environment. The scope includes detailed explanations of the program's modular components, such as selection screens, data declarations, internal tables, subroutines, and class definitions, as well as the sequence of data retrieval and processing steps. The intended audience for this document includes SAP ABAP developers, technical consultants, and project stakeholders who require an in-depth understanding of the program's technical design and operational flow.

# 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 various SAP standard tables such as MARA, LIPS, T001W, VBRK, VBRP, ACDOCA, KNA1, and others. The program is designed to address the need for efficient retrieval, validation, and transformation of master and transactional data related to materials, deliveries, billing documents, financial postings, and plant information. The objectives of the proposed solution are to streamline the process of gathering relevant business data, perform necessary checks and transformations, and prepare the data for further reporting or integration purposes. The solution modularizes the logic into reusable subroutines and leverages selection screens for user-driven data filtering, thereby improving maintainability and user experience.

# 4. Business Process Flow

Start

->

User enters Plant (p\_werks) and Material Number(s) (s\_matnr) on Selection Screen

->

PERFORM fetch\_and\_check\_plant

[Checks if p\_werks is provided, fetches plant data from T001W]

->

PERFORM fetch\_material

[Fetches material data from MARA for selected material numbers]

->

PERFORM fetch\_delivery\_items

[Fetches delivery item data from LIPS based on materials and plant]

->

PERFORM fetch\_konv

[Fetches condition records from prcd\_elements]

->

PERFORM fetc\_vbrk

[Fetches billing document header from VBRK where draft is blank]

->

PERFORM fetch\_vbrp

[Fetches billing document items from VBRP where draft is blank]

->

PERFORM fetch\_bsak

[Fetches financial data from ACDOCA]

->

PERFORM fetch\_j1m0cust

[Fetches customer numbers from KNA1]

->

PERFORM fetch\_marc\_stawn

[Fetches MARC data and calls commodity code service classes]

->

PERFORM fetch\_dzaehk

[Fetches condition counter from prcd\_elements]

->

PERFORM fetch\_jbbranch

[Fetches business place data from P\_BusinessPlace]

->

PERFORM fetch\_vbuk

[Counts entries in VBAK]

->

PERFORM fetch\_marc\_mard

[Fetches material numbers from MARC and storage data from MARD]

->

PERFORM fetch\_orderby

[Processes material data from MARA with substring logic and conditional messaging]

->

PERFORM prepare\_final\_data

[Transfers and prepares final data set from delivery items]

->

PERFORM populate\_salary

[Populates salary value from ACDOCA]

->

End

# 5. Functional Scope

1. In-Scope items:

1.1. The program ZRCOPY\_SAMPLE\_ECC\_CODE\_V1 is designed to perform a series of data retrieval and processing tasks using subroutines (PERFORM statements) and includes modular code via INCLUDE statements.

1.2. The solution covers the following functional areas:

- Selection screen input for plant (p\_werks) and material numbers (s\_matnr), allowing users to specify the plant and a range or list of materials for processing.

- Retrieval and validation of plant data from T001W based on user input (fetch\_and\_check\_plant).

- Fetching of material master data from MARA for selected materials (fetch\_material).

- Extraction of delivery item data from LIPS, filtered by materials and plant, and population of an internal table (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, including company code, fiscal year, document number, line item, GL account, amounts, currency, and posting date (fetch\_bsak).

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

- Fetching of commodity code and related details from MARC and via /sapsll/cl\_mm\_cls\_service class methods (fetch\_marc\_stawn).

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

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

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

- Fetching material numbers from MARC based on substrings and retrieving LSOBS from MARD (fetch\_marc\_mard).

- Fetching and processing material data from MARA with substring logic and conditional message construction (fetch\_orderby).

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

- Preparation of final data by transferring and appending relevant fields from delivery items to a final internal table, counting entries, and clearing the table (prepare\_final\_data).

- Populating salary data by extracting monetary values from ACDOCA (populate\_salary).

- Use of internal tables and structures for temporary data storage and processing, including ikonv, ifinal, ilips, imara, and i\_t001w.

- Object-oriented encapsulation of data and logic via the local class lcl\_data, with a public method get\_data for clearing billing document data.

1.3. The solution is modular, with each subroutine handling a specific data extraction or processing task, and is designed for extensibility and clarity.

2. Out-of-Scope items:

2.1. The program does not include any user interface beyond the selection screen; there is no ALV grid, classical report output, or interactive processing.

2.2. No update, insert, or delete operations are performed on SAP database tables; all operations are read-only.

2.3. There is no error handling, logging, or messaging logic detailed in the provided code snippets.

2.4. No authorization checks or user validation are implemented.

2.5. The solution does not cover integration with external systems or interfaces.

2.6. There is no explicit handling of performance optimization, parallel processing, or background job scheduling.

2.7. No business logic for calculation, aggregation, or transformation of data beyond simple data movement and assignment is included.

2.8. The implementation of the get\_data method in the lcl\_data class is limited to clearing a variable; no further data processing or retrieval logic is present.

2.9. The solution does not include any custom enhancements, BADIs, user exits, or SAP enhancement framework usage.

2.10. There is no documentation of output formatting, printing, or export to external formats (such as Excel or PDF).

# 6. Functional Solution Approach

1. The program ZRCOPY\_SAMPLE\_ECC\_CODE\_V1 is designed to extract, process, and prepare data from various SAP standard tables (such as MARA, LIPS, T001W, VBRK, VBRP, ACDOCA, KNA1, VBAK, MARD, PRCD\_ELEMENTS, and custom tables/views) based on user input from a selection screen. The selection screen allows users to specify a plant (P\_WERKS) and a range of material numbers (S\_MATNR), which serve as the primary filters for subsequent data retrieval.

2. Upon execution, the program sequentially performs a series of data retrieval and processing steps encapsulated in subroutines (PERFORMs). These steps include:

- Validating and fetching plant data (from T001W) if a plant is specified.

- Fetching material master data (from MARA) for the selected material numbers.

- Retrieving delivery item data (from LIPS) filtered by the selected materials and plant.

- Extracting condition records (from PRCD\_ELEMENTS) and billing document data (from VBRK and VBRP) for further processing.

- Gathering financial and customer data (from ACDOCA and KNA1), as well as additional master data such as business place (from P\_BusinessPlace), sales document status (from VBAK), and storage location data (from MARC and MARD).

- Performing specialized data retrievals and transformations, such as substring operations on material numbers, commodity code classification via class methods, and conditional message construction.

3. The program consolidates the retrieved and processed data into internal tables structured to mirror the relevant SAP tables. It then prepares a final dataset (IFINAL) by mapping and transferring the necessary fields from the intermediate tables (such as ILIPS), counting the number of records, and clearing the table for reuse if needed. The approach ensures modularity, reusability, and clarity by encapsulating each functional step in a dedicated subroutine, allowing for easy maintenance and potential extension of the solution to accommodate additional business requirements.

# 7. Functional Requirements

[Error: Section not found in LLM output.]

# 8. Interfaces & Integration

No interface details found in the provided payload.

# 9. Output

Based strictly on the provided payload, the ABAP program 'ZRCOPY\_SAMPLE\_ECC\_CODE\_V1' is structured to perform a series of data retrieval and processing tasks using multiple subroutines. The payload does not explicitly specify any output format (such as Excel, PDF, CSV), layout, or destination (such as SAP AL11, email, or spool). There is also no direct mention of report generation or extract files.

However, the following can be inferred regarding outputs:

- The program prepares internal tables (such as 'ifinal', 'ilips', 'imara', 'ikonv') by fetching and processing data from various SAP tables (e.g., MARA, LIPS, T001W, PRCD\_ELEMENTS, VBRK, VBRP, ACDOCA, KNA1, P\_BusinessPlace, MARC, MARD, VBAK).

- The subroutine 'prepare\_final\_data' processes data into the internal table 'ifinal', which is likely intended for final reporting or further processing, but the payload does not specify the output medium or format.

- The selection screen allows users to input parameters (plant and material numbers) to filter data, but there is no mention of how the results are presented to the user.

- There is no information about outputting data to files, sending emails, or printing to spool.

Functional purpose:

- The main functional purpose is to extract, process, and prepare data from various SAP standard tables for further use, possibly for reporting, analysis, or integration with other processes. The final processed data is stored in internal tables, but the payload does not specify any external output or report generation.

In summary, the payload does not define any explicit reports, extracts, or outputs in terms of format, layout, or destination. All data processing appears to be internal within the program, with the final data likely residing in internal tables for subsequent use.

# 10. UI Requirement

The selection screen consists of the following UI elements:

1. Parameter: P\_WERKS

- Field Type: Input field (single value)

- Data Element: T001W-WERKS (Plant)

- Default Value: Not specified

- Mandatory: Not specified

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

2. Select-Option: S\_MATNR

- Field Type: Range input (multiple values or intervals)

- Data Element: MARA-MATNR (Material Number)

- Default Value: Not specified

- Mandatory: Not specified

- Business Purpose/Validation: Used to input one or more material numbers or ranges. No additional validation or dependencies mentioned.

No further user interactions, default values, mandatory flags, 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 found in the provided payload.

# 13. Assumptions & Dependencies

1. The program relies on the existence and correct structure of standard SAP tables such as T001W, MARA, LIPS, BSEG, VBRK, VBRP, ACDOCA, KNA1, PRCD\_ELEMENTS, VBAK, MARD, MARC, and P\_BusinessPlace.

2. The selection screen parameters (p\_werks and s\_matnr) are assumed to be provided by the user for filtering data.

3. The includes (e.g., ZRCOPY\_SAMPLE\_ECC\_CODE\_TOP\_V1, ZRCOPY\_SAMPLE\_ECC\_CODE\_F01\_V1) must be present and contain the expected declarations and subroutines.

4. The custom class /sapsll/cl\_mm\_cls\_service must exist in the system for commodity code operations.

5. The program assumes that the user has the necessary authorizations to access all referenced tables and objects.

6. The structure and fields of internal tables (e.g., ikonv, ifinal, ilips, imara) are assumed to match the referenced database tables or custom structures.

# 14. Test Scenario

[Error: Section not found in LLM output.]

# 15. Sign-Off

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

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