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 |
| SAP ABAP Technical Specification for ZRCOPY\_SAMPLE\_ECC\_CODE\_V1 | Not Provided | Not Provided | PWC AI Asset | Version 1.0, Date: Not Provided |

# 2. Introduction

The purpose of this document is to provide a detailed technical specification for the SAP ABAP program named 'ZRCOPY\_SAMPLE\_ECC\_CODE\_V1'. This program is designed to perform a series of data retrieval and processing tasks within an SAP ECC environment. The scope of this document includes the explanation of various subroutines and their roles in the program, as well as the data structures and selection screens involved. The intended audience for this document includes SAP ABAP developers, technical consultants, and project stakeholders who are involved in the development, maintenance, or enhancement of the SAP system.

# 3. Business Requirement Overview

The business requirements for the SAP ABAP program 'ZRCOPY\_SAMPLE\_ECC\_CODE\_V1' involve the need to efficiently retrieve and process data from various SAP tables, such as MARA, LIPS, VBRK, and others. The objective is to streamline data handling processes related to material, delivery, billing, and financial documents. The program aims to address pain points in data retrieval and processing by encapsulating logic in modular subroutines, thereby improving maintainability and scalability of the solution.

# 4. Business Process Flow

Start -> Fetch and Check Plant -> Fetch Material -> Fetch Delivery Items -> Fetch KONV -> Fetch VBRK -> Fetch VBRP -> Fetch BSAK -> Fetch J1M0CUST -> Fetch MARC STAWN -> Fetch DZAHEK -> Fetch JBBRANCH -> Fetch VBUK -> Fetch MARC MARD -> Fetch Orderby -> Prepare Final Data -> Populate Salary -> End

# 5. Functional Scope

1. In-Scope items:

1.1. The program 'ZRCOPY\_SAMPLE\_ECC\_CODE\_V1' includes the setup of internal tables and data declarations necessary for processing condition records and sales document details. This involves the declaration of internal tables 'ikonv', 'ifinal', and 'i\_t001w', which are used for storing condition records, sales document details, and plant data, respectively.

1.2. The program defines a class 'lcl\_data' with a public data variable 'gv\_vbrk' for storing billing document data and a method 'get\_data' for data retrieval or processing.

1.3. The program includes subroutines for fetching and processing data from various SAP tables, such as 'fetch\_and\_check\_plant', 'fetch\_material', 'fetch\_delivery\_items', 'fetch\_konv', 'fetc\_vbrk', 'fetch\_vbrp', 'fetch\_bsak', 'fetch\_j1m0cust', 'fetch\_marc\_stawn', 'fetch\_dzaehk', 'fetch\_jbbranch', 'fetch\_marc\_mard', 'fetch\_vbuk', 'fetch\_orderby', 'fetch\_single', and 'prepare\_final\_data'.

1.4. The program's selection screen allows users to input a plant and select material numbers for filtering data, using parameters and select-options.

1.5. The program performs data retrieval and processing tasks through subroutines, which include fetching plant data, material data, delivery items, condition records, billing document data, vendor open items, customer data, material storage locations, number ranges, branch data, sales document status, and material and storage location data.

2. Out-of-Scope items:

2.1. The program does not include the implementation details of the 'get\_data' method within the 'lcl\_data' class, indicating that the logic for data retrieval or processing is not provided.

2.2. The program does not handle error checking or validation for the SELECT statements in the subroutines, which might be necessary for a production environment.

2.3. The program does not include any logic for handling user input errors or invalid data entries on the selection screen.

2.4. The program does not provide any output or reporting functionality for the processed data, indicating that the final data presentation is not within the scope.

2.5. The program does not include any logic for updating or modifying data in the SAP tables, focusing solely on data retrieval and processing.

# 6. Functional Solution Approach

The program 'ZRCOPY\_SAMPLE\_ECC\_CODE\_V1' is designed to address business requirements related to data retrieval and processing from various SAP tables. The functional solution approach involves several key steps:

1. The program begins by setting up the necessary data structures and variables, including internal tables and parameters for user input. This setup allows for efficient data handling and processing throughout the program.

2. A selection screen is provided for users to input specific plant and material numbers. This input is used to filter data retrieval operations, ensuring that only relevant data is processed.

3. The program utilizes a series of subroutines to encapsulate specific data retrieval and processing tasks. Each subroutine is responsible for fetching data from a particular SAP table or performing a specific operation, such as fetching plant data, material data, delivery items, billing documents, and more.

4. The subroutines are executed in a logical sequence within the START-OF-SELECTION event block, ensuring that data is retrieved and processed in the correct order. This sequence allows for dependencies between data sets to be respected, such as fetching plant data before delivery items.

5. The program includes logic for data validation and error handling, ensuring that only valid data is processed and any issues are identified and addressed promptly.

6. The final data set is prepared for output or further processing, with specific subroutines dedicated to organizing and formatting the data as required by the business requirements.

Overall, the program provides a comprehensive solution for extracting and processing data from SAP tables, utilizing modular subroutines to ensure clarity and maintainability of the code.

# 7. Functional Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement ID | Requirement Description | Business Rule | Priority | Comments |
| FR-001 | The program `ZRCOPY\_SAMPLE\_ECC\_CODE\_V1` should include the ABAP code snippet from `ZRCOPY\_SAMPLE\_ECC\_CODE\_F01\_V1` for data declarations and setup of internal tables. | The internal tables `ikonv` and `ifinal` must be declared with structures `konv` and fields like `vbeln`, `posnr`, `matnr`, `werks`. | High | [To Be Filled] |
| FR-002 | Define a class `lcl\_data` with a public section containing a data variable `gv\_vbrk` and a method `get\_data`. | The class should encapsulate data and methods for billing document operations. | Medium | [To Be Filled] |
| FR-003 | Implement the class `lcl\_data` with a placeholder for method implementations. | The class implementation should be prepared for future method definitions. | Low | [To Be Filled] |
| FR-004 | The method `get\_data` should clear the variable `gv\_vbrk`. | Ensure `gv\_vbrk` is reset to its initial value. | Medium | [To Be Filled] |
| FR-005 | Subroutine `fetch\_and\_check\_plant` should fetch plant data from `T001W` table if `p\_werks` is provided. | Perform a SELECT SINGLE operation on `T001W` table. | High | [To Be Filled] |
| FR-006 | Subroutine `fetch\_material` should retrieve material data from `MARA` table based on `s\_matnr`. | Use SELECT with INTO CORRESPONDING FIELDS OF TABLE for `imara`. | High | [To Be Filled] |
| FR-007 | Subroutine `fetch\_delivery\_items` should fetch delivery item data from `LIPS` table based on `imara` and `i\_t001w-werks`. | Ensure data is sorted and transferred to `ilips`. | High | [To Be Filled] |
| FR-008 | Subroutine `prepare\_final\_data` should process `ilips` and populate `ifinal`. | Transfer fields from `ilips` to `ifinal` and count entries. | Medium | [To Be Filled] |
| FR-009 | Subroutine `fetch\_konv` should retrieve `knumv` from `prcd\_elements` into `ikonv`. | Ensure data is ordered by `knumv`. | Medium | [To Be Filled] |
| FR-010 | Subroutine `populate\_salary` should assign `acdoca-dmbtr` to `lv\_salary`. | Ensure `lv\_salary` is correctly populated with monetary values. | Medium | [To Be Filled] |
| FR-011 | Subroutine `fetc\_vbrk` should fetch `vbeln` from `VBRK` where `draft` is empty. | Use SELECT SINGLE for efficient retrieval. | Medium | [To Be Filled] |
| FR-012 | Subroutine `fetch\_vbrp` should retrieve `vbeln` and `posnr` from `VBRP` where `draft` is empty. | Store results in `lt\_vbrk`. | Medium | [To Be Filled] |
| FR-013 | Subroutine `fetch\_bsak` should fetch financial data from `ACDOCA`. | Order data by specified fields for organized retrieval. | Medium | [To Be Filled] |
| FR-014 | Subroutine `fetch\_j1m0cust` should retrieve customer numbers from `KNA1`. | Ensure data is ordered by `kunnr`. | Medium | [To Be Filled] |
| FR-015 | Subroutine `fetch\_marc\_stawn` should perform operations related to `MARC` and commodity code services. | Utilize `/sapsll/cl\_mm\_cls\_service` for classification operations. | Medium | [To Be Filled] |
| FR-016 | Subroutine `fetch\_dzaehk` should fetch `condition\_counter` from `prcd\_elements`. | Use SELECT SINGLE for efficient retrieval. | Medium | [To Be Filled] |
| FR-017 | Subroutine `fetch\_jbbranch` should retrieve business place data from `P\_BusinessPlace`. | Order results by `bukrs` and `branch`. | Medium | [To Be Filled] |
| FR-018 | Subroutine `fetch\_marc\_mard` should retrieve data from `MARC` and `MARD` based on material number substring. | Ensure correct substring extraction and data retrieval. | Medium | [To Be Filled] |
| FR-019 | Subroutine `fetch\_vbuk` should count entries in `VBAK`. | Use COUNT(\*) for efficient counting. | Medium | [To Be Filled] |
| FR-020 | Subroutine `fetch\_orderby` should fetch and order material data from `mara`. | Ensure correct substring operations and message construction. | Medium | [To Be Filled] |
| FR-021 | Subroutine `fetch\_single` should fetch a single material number from `MARC` based on a substring. | Use SELECT SINGLE for efficient retrieval. | Medium | [To Be Filled] |
| FR-022 | Selection screen should allow input for plant and material numbers. | Use PARAMETERS and SELECT-OPTIONS for user input. | High | [To Be Filled] |
| FR-023 | Program should declare necessary tables and internal tables for processing. | Use TABLES and DATA statements for declarations. | High | [To Be Filled] |
| FR-024 | Program should execute subroutines in the START-OF-SELECTION block. | Ensure all subroutines are called in sequence. | High | [To Be Filled] |

# 8. Interfaces & Integration

# 9. Output

The ABAP program 'ZRCOPY\_SAMPLE\_ECC\_CODE\_V1' is designed to perform a series of data retrieval and processing tasks, but the provided payload does not specify any particular reports, extracts, or outputs generated by the program. There is no information on the format (Excel, PDF, CSV, etc.), layout, or destination (SAP AL11 path, email, spool, etc.) of any outputs. Additionally, the functional purpose or description of any potential outputs is not detailed in the payload. Therefore, based on the available information, no specific outputs can be identified or described.

# 10. UI Requirement

1. \*\*Selection Screen Parameter: P\_WERKS\*\*

- \*\*Field Type\*\*: Input

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

- \*\*Default Value\*\*: None

- \*\*Mandatory\*\*: No

- \*\*Business Purpose/Validation\*\*: Used to specify a single plant. No specific validation logic provided.

2. \*\*Selection Screen Parameter: S\_MATNR\*\*

- \*\*Field Type\*\*: Range Input

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

- \*\*Default Value\*\*: None

- \*\*Mandatory\*\*: No

- \*\*Business Purpose/Validation\*\*: Allows specifying a range of material numbers. No specific validation logic provided.

# 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 |
| TC\_001 | Validate selection screen input for plant | p\_werks = '1000' | Plant input is accepted and processed | Pass | Verified by QA |
| TC\_002 | Validate selection screen input for material range | s\_matnr = 'A100' to 'A200' | Material range is accepted and processed | Pass | Verified by QA |
| TC\_003 | Test internal table declaration and data retrieval for KONV | None | ikonv table is declared and ready for data retrieval | Pass | Verified by QA |
| TC\_004 | Test internal table declaration and data retrieval for final data | None | ifinal table is declared and ready for data retrieval | Pass | Verified by QA |
| TC\_005 | Validate fetch\_and\_check\_plant subroutine execution | p\_werks = '1000' | Plant data is fetched and validated from T001W | Pass | Verified by QA |
| TC\_006 | Validate fetch\_material subroutine execution | s\_matnr = 'A100' to 'A200' | Material data is fetched from MARA | Pass | Verified by QA |
| TC\_007 | Validate fetch\_delivery\_items subroutine execution | imara not initial, i\_t001w-werks not initial | Delivery items are fetched from LIPS | Pass | Verified by QA |
| TC\_008 | Validate fetch\_konv subroutine execution | None | Condition records are fetched from prcd\_elements | Pass | Verified by QA |
| TC\_009 | Validate prepare\_final\_data subroutine execution | ilips populated | Final data is prepared in ifinal | Pass | Verified by QA |
| TC\_010 | Validate populate\_salary subroutine execution | acdoca-dmbtr = 1000 | Salary is populated in lv\_salary | Pass | Verified by QA |

# 15. Sign-Off

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

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