Title: Billing Details

Version: 1.0

Date: 18.01.2018

| Project Heading |
| --- |
| Functional Specification ID: Assignment 1 |
| Associated Spec(s): N/A |
| Implementation Phase: Development |

Document History:

|  |  |  |  |
| --- | --- | --- | --- |
| Conversion Revision | Revision Date | Revision Description | Author |
| 1.0 | 18.01.2018 | Initial Review | Group 1 |
|  |  |  |  |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| Document Approvals | | |
| Approver Name | Approver Sign-Off | Date of Sign-Off |
| Vandana Koleshwar |  |  |
| Prashant Kasturi |  |  |
|  |  |  |

Table of Contents

Program Header 3

1 What to bring to a Code Review 4

2 Code Review Checklist 4

## Program Header

|  |  |
| --- | --- |
| Program Name: | ZG1\_ASSIGN1 |
| Program Desc: | Sales and Purchase order information |
| Code Reviewer: | Group 1 |
| Process Owner: | Vandana Koleshwar |
| Technical Owner: | Group 1 |
| Developer: | Group 1 |

# What to bring to a Code Review

Functional Specification

Technical Spec including Test Plan

The Extended Program check and Code Inspector

For new programs, send these to the code reviewer a few days ahead of the code review if possible

# Code Review Checklist

|  |  |  |
| --- | --- | --- |
| QA Description | Self  QA | QA Approval |
| Format/Standard Code | Yes |  |
| Readability –  separate header/comment lines for data/table/parameter … sections | Yes |  |
| Standard Report Header & Trailer --  Use standard functions for report header & trailer (if program produces report) | No |  |
| Check Return Code --  Check sy-subrc after each DB call, function module and for all possible commands | Yes |  |
| Extended Program Check --  Remove unused variables and dead code from the program. Run ‘Extended Program Check’ for this purpose.  Code Inspector –  Run Code Inspector and bring the printout to code review meeting. | Yes |  |
| Naming Convention |  |  |
| Program name - starting with “Z” - starting with “Z” | Yes |  |
| Function group - starting with “Z” | N/A |  |
| Function module - starting with “Z\_” | N/A |  |
| Variable - starting with “g\_” for global variables  “L\_” for local variables | Yes |  |
| Subroutine – ZF\_ | N/A |  |
| Constants - starting with “C\_” | Yes |  |
| Parameters - starting with “P\_” | Yes |  |
| Internal table - starting with “I\_” | Yes |  |
| Select options – starting with “S\_” | Yes |  |
| Record structure - starting with “LA\_” | N/A |  |
| Transaction code - starting with “Z” | Yes |  |
| Data dictionary objects - starting with “Z” | N/A |  |
| Variant – starting with “Z” | N/A |  |
| Range – starting with “R\_” | N/A |  |
| Documentation |  |  |
| Program Header | Yes |  |
| Program Name, Title, Author, Creation Date | Yes |  |
| Description of program | Yes |  |
| List of tables | Yes |  |
| Modification Log | N/A |  |
| In-line or Header Comments | Yes |  |
| Performs, Includes, Call Function points | Yes |  |
| Modification points with CTR # | N/A |  |
| Flags & Counters (comments on each possible value) | Yes |  |
| Security |  |  |
| Authorization Object | N/A |  |
| Performance Improvement (If Applicable) |  |  |
| Is the program using SELECT \* statements?  Convert them to SELECT column 1 column 2… or use Projection views unless all the fields are used. | No |  |
| Are Check Statements for table fields embedded in a SELECT… ENDSELECT loop?  Incorporate the CHECK statements into the WHERE clause of the SELECT statement. | No |  |
| Do Selects on non-key fields use an appropriate DB index or is the table buffered?  Create index for the table in the data dictionary or buffer tables if they are read only or read mostly.  (Request to create index or buffer a table must be submitted to Technical Team Lead) | No |  |
| Is the program using nested SELECTS to retrieve data?  Convert nested SELECTS to a database views or SELECT… FOR ALL ENTRIES in ITAB or joins. | No |  |
| Are there SELECTS without WHERE condition against files that grow constantly (BSEG, MSEG, VBAK…)?  Program design is wrong. Back to the drawing board. | No |  |
| Are SELECT accesses to master data files buffered (i.e. no duplicate accesses with the same key)?  Buffer accesses to master data fields by storing the data in an internal table and filling the table with the READ TABLE…. BINARY SEARCH method. | No |  |
| Is the program using SELECT…. APPEND ITAB.. ENDSELECT techniques to fill internal tables?  Change processing to read data into an internal table (SELECT VBELN AUART… into TABLE I\_itab..) | No |  |
| Is the program doing calculations/summations that can be done on the database via SUM, AVG, MIN or MAX functions for the SELECT statements?  Use the calculation capabilities of the database via SELECT SUM (….) | N/A |  |
| Is the program using a SELECT ORDER BY statements?  Data should be read into an internal table first and then sorted, unless there is an appropriate index for the order by fields. | No |  |
| Use SELECT SINGLE if all the key fields are known. | N/A |  |
| Use SELECT SINGLE if you only want one record. | N/A |  |
| Limit joins to 3 or less tables for performance issues. | Yes |  |
| Use SELECT <fields> from DBTABLE into.. TABLE <Internal table name> directly without using ENDSELECT statement. (Ex: MARA, BKPF, BSEG, MKPF, MSEG etc ) | Yes |  |
| Use views instead of nested SELECT, if available. | N/A |  |
| Internal Table is defined with “OCCURS 0” if the maximum size of the table is potentially >= 8KB. If the data to be read is estimated to be under 8K, then specify the number of lines in the OCCURS statement (i.e., itab1 occurs 10) is effective. | N/A |  |
| Put conditions which will eliminate most records first with IF…AND | Yes |  |
| Put success conditions first with IF…OR | Yes |  |
| Put most probable conditions first with IF or CASE | Yes |  |
| Choose the most efficient logical database possible. If logical databases are used, read tables in the same order as they are listed. | Yes |  |
| Specify the fields while SORTing the internal table. | No |  |
| To find out the number of entries in the internal table, use DESCRIBE statement instead of looping the table for record counting purpose | N/A |  |
| Specify sorted fields for READ….Binary search | N/A |  |
| Use DO LOOP only when the number of loop passes is known. Otherwise, use EXIT or STOP command to prevent endless looping. | N/A |  |
| Use the in list clause wherever possible as opposed to using BETWEEN. | N/A |  |
| OPEN the dataset before the read/write operation, and always CLOSE the file when file processing finished. | N/A |  |
| Messages are included at appropriate places to indicate illegal operation, no authorization, no data found, etc. | Yes |  |
| Header line of Internal Table is cleared before moving data into it. | N/A |  |
| Internal tables are freed when no longer needed especially when the tables are large or the program is a batch program. | N/A |  |
| When searching for a fragment in a long string, use the special operators CO, CA, CS instead of programming the operations. | N/A |  |
| When deleting the leading spaces in a string, use SHIFT…LEFT DELETING LEADING SPACE. | N/A |  |
|  |  |  |
| Avoid: |  |  |
| Using shift inside a while-loop | Yes |  |
| Using move-corresponding for just a few fields | N/A |  |
| Using DB field names if value will be changed | N/A |  |
| Using COLLECT statement when not summarizing numeric fields | N/A |  |
| Hard codes in programs | Yes |  |
|  |  |  |



Comments: N/A