Object Overview

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object ID | MM\_E\_4 | Business Process | | MM |
| Object Title | Batch Determination Logics and Short Text | | | |
| Object Description | Batch Determination Logics and Short Text | | | |
| Object Type | Report | Interface | | Conversion |
| Enhancement | Form | | Solution/Tool |
| Priority | High | Medium | | Low |
| Complexity | High | Medium | | Low |
| SAP Module | MM |  | |  |
| SAP Release | S/4 | | | |
| SAP Transaction  and Menu Path | VL01N/VL02N/VL03N | | Required Development Completion Date |  |
| Type of Interface |  | | Interface Run |  |

Document Version History

| Version No. | Date | Version Author | Brief Description of Change (add incident # if appliciable) |
| --- | --- | --- | --- |
| 1.0 | 07/15/2019 | Carlos Gonçalves | Document Creation |

<Minor changes to this document will result in an increase in the second number (1.#>) of the version number

<Major changes to this document will result in an increase in the first number and zeroing the second number (#.0) of the version number>

<Final approval of this document will result in an increase in the first number and zeroing the second number (#.0) of the version number>

Table of Contents

[1. Program Attributes 1](#_Toc300315386)

[2. Authorization Requirements 2](#_Toc300315387)

[3. Technical Flow Diagram 3](#_Toc300315388)

[4. Processing Logic 3](#_Toc300315389)

[5. Reusable Code 4](#_Toc300315390)

[6. Internal Tables 4](#_Toc300315391)

[7. Messages 4](#_Toc300315392)

[8. Text Elements 4](#_Toc300315393)

[9. Subroutines 4](#_Toc300315394)

[10. Assumption in Technical Design 4](#_Toc300315395)

[11. Open Issues in Technical Design 4](#_Toc300315396)

[12. Unit Test Plan 4](#_Toc300315397)

[13. Approvals 5](#_Toc300315398)

[14. Related / Referenced Documents 5](#_Toc300315399)

[15. Attachments 5](#_Toc300315400)

<*Jump to a section by ‘CTRL+click’ on item in Table of Contents*

*To update, right click on Table of Contents items and select ‘Update Field’, then select ‘Update Entire Field’>*

Document Instructions

-All text added will be Arial, size 11

-Use standard COE naming conventions

-Do not remove any sections

If not used, enter ‘N/A’ into section

1. Program Attributes

|  |  |
| --- | --- |
| Smartform |  |
| Report |  |
| Include |  |
| Smartstyle |  |
| Search help |  |
| Domain |  |
| Function module |  |
| Class | ZCL\_IM\_\_SHP\_DELIVERY\_PROC - Imp. class for BAdI imp. Z\_SHP\_DELIVERY\_PROC |
| Badi |  |
| Enhancement implementation | ZSD\_ENH\_SHP\_DELIVERY\_PROC - Check Delivery Batch Sled  ZENH\_DLV\_BATCH\_SLED\_CHECK - Check Batch SLED during Delivery creation  ZENH\_BATCH\_TEXT - check batch text |
| Implementation | Z\_SHP\_DELIVERY\_PROC - Check Batch Sled |

|  |  |
| --- | --- |
| Transaction Code |  |
| Development Class | ZMM |

|  |  |
| --- | --- |
| SAP Tables Read |  |
| Custom Tables |  |

1. Authorization Requirements

<Please fill this section if there are special Authorization Requirements; else mention N/A. Every authorization object needs to be documented to provide the security administrator information on the purpose and use of the object. The following sections are the minimal documentation requirements>

* 1. Definition

<The definition establishes the purpose and or use for the object. Is there a standard object available to be used or custom authorization object to be created>

* 1. System / Modules

Systems

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ECC (Y/N) | PI (Y/N) | BW (Y/N) | SolMan (Y/N) | S/4 (Y/N) | TM (Y/N) | EM (Y/N) | YL (Y/N) |
|  |  |  |  | Y |  |  |  |

* 1. Processing Requirements

<Provide in this (Optional) section the specific security processing requirements including enabling/disabling screen controls based on authorization, granting access to data only from a specific region, etc>

* 1. Account Requirements

<Provide in this section the list of batch accounts, users, job titles or the appropriate security role(s) of users who need access to run the program (or who should not have access)>

Users

\*To add a row, press the ‘tab’ button at end of row

| User (First, Last) | User ID | Job Title | User Group |
| --- | --- | --- | --- |
|  |  |  |  |

Roles

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| New Role Request (Y/N) |  | | Change to Existing Role (Y/N) |  |
| Technical Role Name | |  | | |
| Role Short Name | |  | | |
| Role Long Text | |  | | |
| Reference Role Name | |  | | |
| Name of Approver or DL in Prod | |  | | |

* 1. Authority Checks

<Provide in this (Required) section functional level details about the security check>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Security Object |  | | | |
| New Object Request | (Y/N) | Existing Object | (Y/N) |  |

Fields

|  |  |  |
| --- | --- | --- |
| Field Name: | Value: | Description: |
|  |  |  |
|  |  |  |

* 1. Authorization Group Attributes:

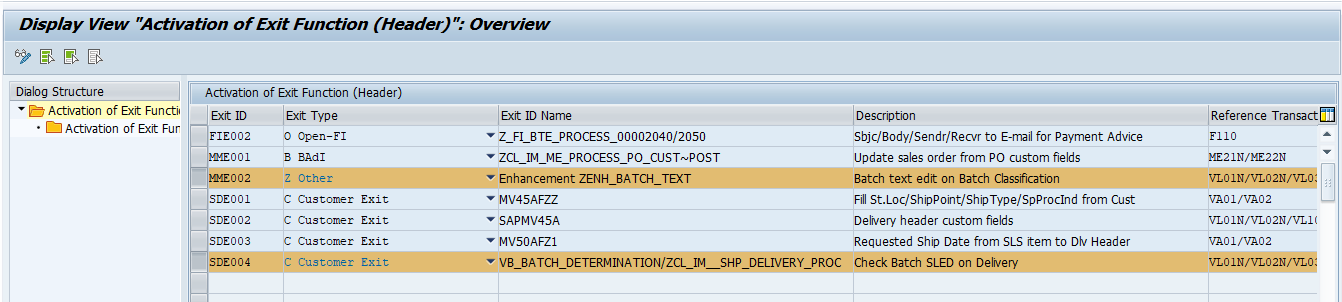
*<Provide in this optional section the authorization group attribute to use for this program>*

* 1. Special Instructions (Comments, SU53, Privacy Concerns)
  2. SAP Manual Configuration Instructions:

Created entry on user exit table for exit handling.

MME002 – for Batch text

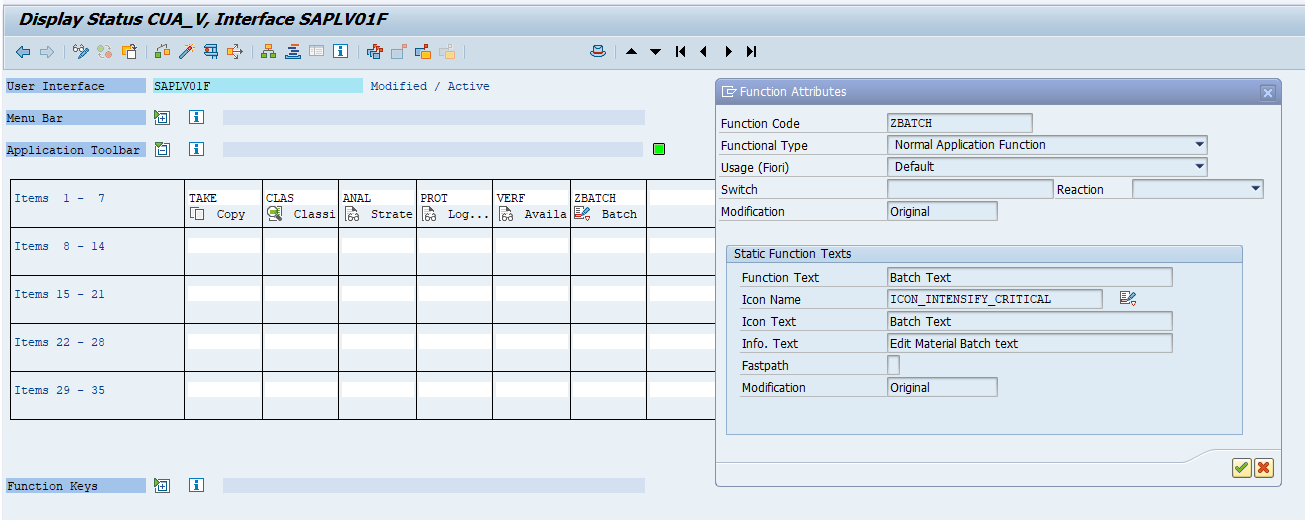
SDE004 – For batch sled check.



* 1. Procedure

*<The procedure section helps to explain how this object is to be used. Examples with field values and explanations should be provided. Provide Example of technical ABAP coding snippet to validate Authorization Requirement. >*

Status CUA\_V of program SAPLV01F was changed to have Batch Text button available.



Batch Sled Check

|  |  |
| --- | --- |
| Function module | VB\_BATCH\_DETERMINATION |
| *"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""$"$\SE:(1) Enhancement Exit EHP\_VB\_BATCH\_DETERMINATION\_02, Enhancement VB\_SFWS\_VB\_BATCH\_DETERMINATION, Start                                                 A* *\*$\*$-Start: (1)---------------------------------------------------------------------------------$\*$\** ENHANCEMENT 1  ZENH\_DLV\_BATCH\_SLED\_CHECK.    *"active version* *\*  break cgoncalves.* constants: lc\_exitid type zbc\_exitid value 'SDE004',            lc\_kval1  type zbc\_k1val  value 'NO-CHECK'.  data: lv\_chkval type zbc\_exitvlow,       lv\_actflg type zbc\_actflg.  types: begin of ty\_mch1,          matnr type mch1-matnr,          charg type mch1-charg,          vfdat type mch1-vfdat,        end of ty\_mch1. data: lt\_mch1 type standard table of ty\_mch1,       ls\_mch1 type ty\_mch1.  *\*\*\* Check if user exit is active on control table* call method zcl\_bc\_chk\_actst\_exit=>chk\_actst\_exit   exporting     im\_exitid = lc\_exitid     im\_chkval = lv\_chkval     im\_k1val  = lc\_kval1   importing     ex\_actflg = lv\_actflg.  *\*\* If active, process exit* if lv\_actflg is not initial.    if not xresult[] is initial.      select matnr charg vfdat            into table lt\_mch1            from mch1            for all entries in xresult            where matnr = xresult-matnr            and charg = xresult-charg            order by primary key.   endif.    loop at i\_bdbatch where menge > 0.     read table xresult with key matnr = i\_bdbatch-matnr                                 werks = i\_bdbatch-werks                                 charg = i\_bdbatch-charg                                 lgort = i\_bdbatch-lgort                                 sobkz = i\_bdbatch-sobkz.     if sy-subrc is initial.       clear ls\_mch1.       read table lt\_mch1 into ls\_mch1 with key matnr = xresult-matnr                                                charg = xresult-charg                                                binary search.       if sy-subrc eq 0.         data(lv\_dif) = ls\_mch1-vfdat - sy-datum.          if lv\_dif between 0 and 30. *\*\*\* The remaining shelf life of batch xxxxx is YY days*           concatenate text-Z01                       xresult-charg text-Z02                       into data(lv\_message) separated by space.           message w100(farr\_inflight\_check) with lv\_message lv\_dif text-z03.         elseif lv\_dif < 0. *\*\*\* Batch xxxxx has been expired on DD/MM/YYYY.*           message w100(farr\_inflight\_check) with Text-z04 xresult-charg                                                  text-z05 ls\_mch1-vfdat.         endif.        endif.     endif.   endloop. endif.  ENDENHANCEMENT. *\*$\*$-End:   (1)---------------------------------------------------------------------------------$\*$\** *\*Selection criteria only?* | |

|  |  |
| --- | --- |
| Class / Method | ZCL\_IM\_\_SHP\_DELIVERY\_PROC / IF\_EX\_LE\_SHP\_DELIVERY\_PROC~DELIVERY\_FINAL\_CHECK |
| method if\_ex\_le\_shp\_delivery\_proc~delivery\_final\_check.      constants: lc\_exitid type zbc\_exitid value 'SDE004',                lc\_kval1  type zbc\_k1val  value 'NO-CHECK'.      data: lv\_chkval type zbc\_exitvlow,           lv\_actflg type zbc\_actflg.      types: begin of ty\_mch1,              matnr type mch1-matnr,              charg type mch1-charg,              vfdat type mch1-vfdat,            end of ty\_mch1.     data: lt\_mch1 type standard table of ty\_mch1,           ls\_mch1 type ty\_mch1.  *\*\*\* Check if user exit is active on control table*     call method zcl\_bc\_chk\_actst\_exit=>chk\_actst\_exit       exporting         im\_exitid = lc\_exitid         im\_chkval = lv\_chkval         im\_k1val  = lc\_kval1       importing         ex\_actflg = lv\_actflg.  *\*\* If active, process exit*     if lv\_actflg is not initial.         if not it\_xlips[] is initial.          select matnr charg vfdat                into table lt\_mch1                from mch1                for all entries in it\_xlips                where matnr = it\_xlips-matnr                and charg = it\_xlips-charg                order by primary key.       endif.        loop at it\_xlikp into data(ls\_xlikp).          loop at it\_xlips into data(ls\_xlips) where vbeln = ls\_xlikp-vbeln                                              and charg <> space.            clear ls\_mch1.           read table lt\_mch1 into ls\_mch1 with key matnr = ls\_xlips-matnr                                                    charg = ls\_xlips-charg                                                    binary search.           if sy-subrc eq 0.             data(lv\_dif) = ls\_mch1-vfdat - sy-datum.              if lv\_dif between 0 and 30. *\*\*\* The remaining shelf life of batch xxxxx is YY days*               concatenate text-z01                           ls\_xlips-charg text-z02                           into data(lv\_message) separated by space.               message w100(farr\_inflight\_check) with lv\_message lv\_dif text-z03.             elseif lv\_dif < 0. *\*\*\* Batch xxxxx has been expired on DD/MM/YYYY.*               message w100(farr\_inflight\_check) with text-z04 ls\_xlips-charg                                                      text-z05 ls\_mch1-vfdat.             endif.            endif.         endloop.        endloop.     endif.   endmethod. | |

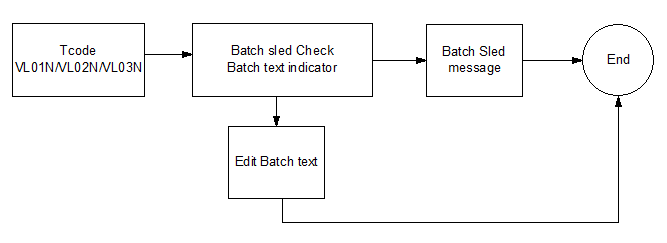
Batch text

Enhancement ZENH\_BATCH\_TEXT.

|  |  |
| --- | --- |
| Include | MILL\_LV01FMO2 |
| *"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""$"$\SE:(1) Form DATEN\_SETZEN\_POS\_MILL, End                                                                                                                   A* *\*$\*$-Start: (1)---------------------------------------------------------------------------------$\*$\** ENHANCEMENT 1  ZENH\_BATCH\_TEXT.    *"active version* *\* Check for Batch text* data: lv\_verm    type tdid     value 'VERM',       lv\_charge  type tdobject value 'CHARGE',       lv\_tx      type char02   value 'TX',       lv\_tdname type stxl-tdname.  constants: lc\_exitid type zbc\_exitid value 'MME002',            lc\_kval1  type zbc\_k1val  value 'NO-CHECK'.  data: lv\_chkval type zbc\_exitvlow,       lv\_actflg type zbc\_actflg.  *\*\*\* Check if exit is active* call method zcl\_bc\_chk\_actst\_exit=>chk\_actst\_exit   exporting     im\_exitid = lc\_exitid     im\_chkval = lv\_chkval     im\_k1val  = lc\_kval1   importing     ex\_actflg = lv\_actflg.  if lv\_actflg is not initial.    clear lv\_tdname.   lv\_tdname(44)    = v01fdk-matnr.  *"Material*   lv\_tdname+44(10) = xv01fdp-charg. *"Batch*  *\*\*\* Check if text exits for the Batch*   select count(\*)                                    *"#EC CI\_SEL\_NESTED*          from stxl          where relid    eq lv\_tx          and   tdspras  eq sy-langu          and   tdname   eq lv\_tdname          and   tdid     eq lv\_verm          and   tdobject eq lv\_charge.    if sy-subrc eq 0. *\*\*\* Set icon indicator*     xv01fdp-mill\_icon\_uc = ICON\_DISPLAY\_NOTE.   endif. endif. ENDENHANCEMENT. *\*$\*$-End:   (1)---------------------------------------------------------------------------------$\*$\** | |

|  |  |
| --- | --- |
| Include | LV01FI00 |
| *\*$\*$-Start: LV01FI00\_07-------------------------------------------------------------------------$\*$\** ENHANCEMENT 2  ZENH\_BATCH\_TEXT.    *"active version* if ok\_code\_old = 'ZBATCH'.    data: lv\_matnr    type mara-matnr,         lv\_id       type thead-tdid,         lv\_language type thead-tdspras,         lv\_name     type thead-tdname,         lv\_object   type thead-tdobject,         ls\_header   type thead,         lt\_lines    type standard table of tline.    constants: lc\_exitid type zbc\_exitid value 'MME002',              lc\_kval1  type zbc\_k1val  value 'NO-CHECK'.    data: lv\_chkval type zbc\_exitvlow,         lv\_actflg type zbc\_actflg.     call method zcl\_bc\_chk\_actst\_exit=>chk\_actst\_exit     exporting       im\_exitid = lc\_exitid       im\_chkval = lv\_chkval       im\_k1val  = lc\_kval1     importing       ex\_actflg = lv\_actflg.    if lv\_actflg is not initial.  *\* Charge von Dynpro lesen*     get cursor field dynfield line dynline.     tab\_index = loop\_i\_100 + dynline - 1.      read table xv01fdp index tab\_index.     if sy-subrc eq 0.        lv\_id = 'VERM'.       lv\_object = 'CHARGE'.       lv\_language = sy-langu.        call function 'CONVERSION\_EXIT\_ALPHA\_INPUT'         exporting           input  = v01fdk-matnr         importing           output = lv\_matnr.        lv\_name(44) = lv\_matnr.       lv\_name+44(10) = xv01fdp-charg.        call function 'READ\_TEXT'         exporting           id                      = lv\_id           language                = lv\_language           name                    = lv\_name           object                  = lv\_object         importing           header                  = ls\_header         tables           lines                   = lt\_lines         exceptions           id                      = 1           language                = 2           name                    = 3           not\_found               = 4           object                  = 5           reference\_check         = 6           wrong\_access\_to\_archive = 7           others                  = 8.        if sy-subrc eq 0.          call function 'EDIT\_TEXT'           exporting             header        = ls\_header *\*           save          = 'X'*             save          = ' '             line\_editor   = 'X'           tables             lines         = lt\_lines           exceptions             id            = 1             language      = 2             linesize      = 3             name          = 4             object        = 5             textformat    = 6             communication = 7             others        = 8.          if sy-subrc eq 0.           call function 'SAVE\_TEXT'             exporting *\*             CLIENT          = SY-MANDT*               header          = ls\_header *\*             INSERT          = ' '*               savemode\_direct = 'X'             tables               lines           = lt\_lines             exceptions               id              = 1               language        = 2               name            = 3               object          = 4               others          = 5.            if sy-subrc <> 0. *\* Implement suitable error handling here*           endif.         endif.       endif.     endif.   endif. endif. ENDENHANCEMENT. *\*$\*$-End:   LV01FI00\_07-------------------------------------------------------------------------$\*$\** | |

1. Technical Flow Diagram



1. Processing Logic

* Batch SLED check.

Check for user exit ID SDE004.

If return flag is not initial.

Based on Delivery item, retrieve batch shelf life expiration date.

Calculate difference days between batch date and current date.

Based on this difference, apply the following rule.

If difference is between 0 and 30, show message “The remaining shelf life of batch xxxxx is YY days”.

If difference is lower than zero, show message “Batch xxxxx has been expired on DD/MM/YYYY.”.

* Batch text.

*Batch Text indicator*

Check for user exit ID MME002.

If return flag is not initial.

Based on material and batch number, check if there is a text, checking on table STXL.

If text exist, set icon ICON\_DISPLAY\_NOTE on field XV01FDP-MILL\_ICON\_UC.

*Batch text edition*

Check for user exit ID MME002.

If return flag is not initial.

Call Function module EDIT\_TEXT.

Call Function SAVE\_TEXT.

1. Reusable Code
2. Internal Tables

|  |  |
| --- | --- |
| Name: | Description |
|  |  |
|  |  |

1. Messages

|  |  |  |
| --- | --- | --- |
| Message Class | Message ID | Message Text |
|  |  |  |
|  |  |  |

1. Text Elements

|  |  |
| --- | --- |
| Name: | Description |
|  |  |
|  |  |

1. Subroutines

|  |  |
| --- | --- |
| Name: | Description |
|  |  |
|  |  |

1. Assumption in Technical Design
2. Open Issues in Technical Design
3. Unit Test Plan

*<Document key test technical aspects of business scenarios that will form basis for acceptance testing>*

Unit Testing

* Unit testing is required to test the functionality that is being built in SAP for this development object.
* Unit testing conditions provide the criteria that needs to be tested to be able to sign off on this object as functioning correctly
* Need to ensure that unit testing accounts for the following (at a minimum)
  + Assumptions are met
  + Constraints are met
  + Security requirements are met

|  |  |  |
| --- | --- | --- |
| Scenario Name | Input Selection Criteria | Expected Result |
| I. General Functional Test Scenarios | | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| II. Negative Functional Test Scenarios | | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| III. Security Authorization Test Scenarios (Positive & Negative) | | |
|  |  |  |

1. Approvals

|  |  |  |
| --- | --- | --- |
| Business | | |
| Business Person Accountable’s Name | Signature | Date |
|  |  |  |
| Technical | | |
| Process Analyst Lead Name | Signature | Date |
|  |  |  |
| Development Lead Name | Signature | Date |
|  |  |  |

1. Related / Referenced Documents

| Ref. ID | Document Name | Version | Brief Description |
| --- | --- | --- | --- |
|  |  |  |  |

1. Attachments

|  |  |  |  |
| --- | --- | --- | --- |
| Attch ID | Document Name | Version | Brief Description |
|  |  |  |  |