Template Batch Input:  
  
DATA: TI\_TABELA TYPE TABLE OF BDCDATA.  
  
FORM F\_REALIZAR\_BATCH\_INPUT.  
  
PERFORM F\_PREENCHER\_CAMPOS USING '' '' '' '' ''.  
PERFORM F\_PREENCHER\_CAMPOS USING '' '' '' '' ''.  
PERFORM F\_PREENCHER\_CAMPOS USING '' '' '' '' ''.  
  
CALL TRANSACTION 'SM30' USING TI\_TABELA MODE 'A'.  
  
ENDFORM.  
  
FORM F\_PREENCHER\_CAMPOS   
USING P\_PROGRAM  
P\_DYNPRO  
P\_DYNBEGIN  
P\_FNAM  
P\_FVAL.  
  
DATA: WA\_TABELA TYPE BDCDATA.  
  
WA\_TABELA-PROGRAM = P\_PROGRAM.  
WA\_TABELA-DYNPRO = P\_DYNPRO.  
WA\_TABELA-DYNBEGIN = P\_DYNBEGIN.  
WA\_TABELA-FNAM = P\_FNAM.  
WA\_TABELA-FVAL = P\_FVAL.  
  
APPEND WA\_TABELA TO TI\_TABELA.  
  
ENDFORM.

Batch Input - Básico

*"  Vamos agora tentar explicar um Batch Input, sabe quando tem aquele  
" funcionário que tem um monte de dados dentro de um arquivo externo como  
" um arquivo .xls ou .txt ou qualquer outro formato fora do SAP e este  
" funcinário fala que os dados estão dentro de um arquivo e que pra  
" cadastrar todos dentro da SAP demoraria dias, nessa hora o funcional  
" escuta em sua mente sombria a palavra BATCH INPUT, a primeira coisa  
" que o funcional fala é organize todos os dados em um único arquivo  
" pulando uma linha por registro e separando os dados por ';' ou 'Space'  
" recomendo ';' pois as vezes contem dados com espaço dentro e isso pode  
" atrapalhar o processo de Input no Sistema.  
"  Crie um arquivo que servirá como entrada de dados com as informações:  
  
" 8000000001;nome teste1;rua teste1;São Paulo1;12345671  
" 8000000002;nome teste2;rua teste2;São Paulo2;12345672  
  
"  Lembrando que a ordem é muito importante pois eh gravado na ordem.  
  
"  Tem um outro arquivo muito importante tb, o famoso SHDB contendo o  
" mapeamento do processo por este arquivo vc consegue ver quais são os  
" dados necessários para realizar com sucesso o processo qualque dúvida  
" http://abapjuniores.blogspot.com/2011/04/shdb-basico.html.  
"  Segue abaixo um programa comentado básico para um batch imput funciona.  
  
\*&---------------------------------------------------------------------\*  
REPORT zmj\_batch.  
  
  
" Tipos ....  
TYPES:  
" Dados que serão informados pelo arquivo.  
  BEGIN OF ty\_arqui,  
    cliente TYPE rf02d-kunnr,        " Código do Cliente  
    nome    TYPE kna1-name1,         " Nome do cliente  
    rua     TYPE kna1-stras,         " Rua do cliente  
    end     TYPE kna1-ort01,         " Endereço do cliente  
    tel     TYPE kna1-telf1,         " telefone do cliente  
  END OF ty\_arqui,  
  
" Batch input nova estrutura do campo de tabela  
  BEGIN OF ty\_bdcdata,  
    program   TYPE bdcdata-program,  " Pool de módulos BDC  
    dynpro    TYPE bdcdata-dynpro,   " NÚmero de tela BDC  
    dynbegin  TYPE bdcdata-dynbegin, " Início BDC de uma tela  
    fnam      TYPE bdcdata-fnam,     " Nome do campo  
    fval      TYPE bdcdata-fval,     " Valor do campo BDC  
  END OF ty\_bdcdata,  
  
" Relação informativa do log  
  BEGIN OF ty\_message,  
    cliente TYPE rf02d-kunnr,        " Código do cliente  
    msgty   TYPE message-msgty,      " Tipo da mensagem  
    msgno   TYPE message-msgno,      " Numero da mensagem  
    msgtx   TYPE message-msgtx,      " Descrição da mensagem  
  END OF   ty\_message  
    .  
  
" Tabelas Internas ....  
DATA: it\_arqui    TYPE TABLE OF ty\_arqui,  
      it\_bdcdata  TYPE TABLE OF ty\_bdcdata,  
      it\_msg      TYPE TABLE OF bdcmsgcoll WITH HEADER LINE,  
      it\_message  TYPE TABLE OF ty\_message  
        .  
" Tabela que recebe as informações crua do arquivo que será tratado  
DATA: BEGIN OF t\_line OCCURS 0,  
        linha(108) TYPE c,  
      END OF t\_line  
        .  
  
" Estruturas ...  
DATA: st\_arqui   TYPE ty\_arqui,  
      st\_bdcdata TYPE ty\_bdcdata,  
      st\_message TYPE ty\_message  
        .  
  
" Variaveis ....  
DATA: vg\_mode(1) TYPE c VALUE 'N', " informa o Modo do Call Transaction  
      vg\_texto(100) TYPE c,        " Texto para o Indicator  
      vg\_s TYPE c VALUE 'S',       " Informa o Update do call Transaction  
      mensg LIKE message VALUE IS INITIAL, " variavel que recebe retorno  
      msgno LIKE sy-msgno  
        .  
  
  
" Tela de Seleção ....  
" texto (  
SELECTION-SCREEN BEGIN OF BLOCK a WITH FRAME TITLE text-001.  
PARAMETERS:    p\_dest TYPE string, " Texto ( Arquivo txt: )  
               p\_log  TYPE string  " Texto ( Arquivo log: )  
                 .  
SELECTION-SCREEN END OF BLOCK a.  
  
" Quando for requisitado um valor no 'Parameter faça ...  
AT SELECTION-SCREEN ON VALUE-REQUEST FOR p\_dest.  
  PERFORM z\_busca\_arquivo.  
  
" Quando for requisitado um valor no 'Parameter faça ...  
AT SELECTION-SCREEN ON VALUE-REQUEST FOR p\_log.  
  PERFORM z\_mostra\_local.  
  
"  Começa aki ...  
START-OF-SELECTION.  
  PERFORM: z\_baixa\_arquivo,  
           z\_trata\_arquivo,  
           z\_monta\_shdb,  
           z\_manda\_arquivo  
           .  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  z\_busca\_arquivo  
\*&---------------------------------------------------------------------\*  
  "  Pega o valor contido no parameters e add o texto '.txt' no final assim o  
  " o nome do arquivo fica com a extensão desejada.  
  FORM z\_busca\_arquivo.  
  CONCATENATE p\_dest '.txt' INTO p\_dest.  
  CALL FUNCTION 'WS\_FILENAME\_GET'  
       EXPORTING  
            def\_filename = ' '  
            def\_path     = 'C:\'  
            mask         = ',Texto,\*.txt,Todos,\*.\*.'  
            mode         = 'O'  
            title        = 'Arquivo de Entrada'(004)  
       IMPORTING  
            filename     = p\_dest  
       EXCEPTIONS  
            OTHERS.  
  
ENDFORM.                    " z\_busca\_local  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  z\_mostra\_local  
\*&---------------------------------------------------------------------\*  
  " Mostra o local onde será gravado o arquivo de Log.  
FORM z\_mostra\_local .  
  CALL METHOD cl\_gui\_frontend\_services=>directory\_browse  
    EXPORTING  
      window\_title         = 'Informar o caminho para gerar o arquivo'  
      initial\_folder       = 'C:\'  
    CHANGING  
      selected\_folder      = p\_log  
    EXCEPTIONS  
      cntl\_error           = 1  
      error\_no\_gui         = 2  
      not\_supported\_by\_gui = 3  
      OTHERS               = 4.  
  " Nome do Arquivo do Log contendo o endereço+NOME1+HORA+.TXT.  
  CONCATENATE  p\_log '\' sy-uname '\_' sy-uzeit '.TXT' INTO p\_log  
          .  
  
ENDFORM.                    " Z\_MOSTRA\_LOCAL  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  Z\_MONTA\_ARQUIVO  
\*&---------------------------------------------------------------------\*  
"  Pega o arquivo externo relacionado e coloca os dados na 'T\_LINE  
FORM z\_baixa\_arquivo.  
  CALL FUNCTION 'GUI\_UPLOAD'  
    EXPORTING  
      filename = p\_dest  
    TABLES  
      data\_tab = t\_line  
    EXCEPTIONS  
      OTHERS   = 17.  
  
ENDFORM.                    " Z\_MONTA\_ARQUIVO  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  Z\_TRATA\_ARQUIVO  
\*&---------------------------------------------------------------------\*  
"  Será os dados conforme sequencia de campos e add na tabela principal  
FORM z\_trata\_arquivo.  
  LOOP AT t\_line.  
    " o Split quebra a linha e separa por delimitador ';' colocando cada dado  
    " em cada campo sequencialmente.  
    SPLIT t\_line AT ';' INTO: st\_arqui-cliente  
                              st\_arqui-nome  
                              st\_arqui-rua  
                              st\_arqui-end  
                              st\_arqui-tel  
                              .  
    " Add registro na tabela.  
    APPEND st\_arqui TO it\_arqui.  
  
  ENDLOOP.  
ENDFORM.                    " Z\_TRATA\_ARQUIVO  
\*&---------------------------------------------------------------------\*  
\*&      Form  z\_monta\_shdb  
\*&---------------------------------------------------------------------\*  
FORM z\_monta\_shdb.  
  
  LOOP AT it\_arqui INTO st\_arqui.  
    " cria uma variavel pra informar qual cliente está porcessando no  
    " perform z\_sapgui\_progress\_indicator.  
    CONCATENATE 'Processando o Cliente -' st\_arqui-cliente  
    INTO vg\_texto SEPARATED BY space.  
    "  informa o processo atual  
    PERFORM z\_sapgui\_progress\_indicator USING vg\_texto.  
    " É aki que o bixo pega, lembra dakele arquivo SHDB que enviaram pra vc  
    " é aki que ele começa a faze sentido, oq nós estamos fazendo aki e criando  
    " uma tabela com as informações conforme o SHDB só mundando a informação  
    " que vc quer que mude conforme o registro.  
    "  Depois de terminar os performs z\_preenche\_bdc vc vai dar uma olhada  
    " na tabela it\_bdcdata pq ela vai estar igualzinha com o SHDB que  
    " enviaram pra vc.  
    " Crie um 'Perform pra cada tela que tiver no SHDB.  
    PERFORM z\_preenche\_bdc USING:  
  
      'X'    'SAPMF02D'       '0101',  
      ' '    'BDC\_CURSOR'     'RF02D-D0110',  
      ' '    'BDC\_OKCODE'     '/00',  
      ' '    'RF02D-KUNNR'    st\_arqui-cliente,  
      ' '    'RF02D-D0110'    'X'.  
  
    PERFORM z\_preenche\_bdc USING:  
  
      'X'    'SAPMF02D'        '0110',  
      ' '    'BDC\_CURSOR'      'KNA1-TELF1',  
      ' '    'BDC\_OKCODE'      '/00',  
      ' '    'KNA1-NAME1'     st\_arqui-nome,  
      ' '    'KNA1-STRAS'     st\_arqui-rua,  
      ' '    'KNA1-ORT01'     st\_arqui-end,  
      ' '    'KNA1-TELF1'      st\_arqui-tel.  
  
    PERFORM z\_carrega\_transacao.  
    PERFORM z\_imprime\_mensagem.  
  
    CLEAR it\_bdcdata.  
  
  ENDLOOP.  
ENDFORM.                    " z\_monta\_shdb  
\*&---------------------------------------------------------------------\*  
\*&      Form  Z\_PREENCHE\_BDC  
\*&---------------------------------------------------------------------\*  
"  Se Dynbegin = 'X' ele preenche as informações da tela, senão ele preenche  
" o campo e o dado dela. prontio.  
FORM z\_preenche\_bdc  USING dynbegin  
                           name  
                           value.  
  IF dynbegin = 'X'.  
    MOVE: name      TO st\_bdcdata-program,  
          value     TO st\_bdcdata-dynpro,  
          dynbegin  TO st\_bdcdata-dynbegin.  
    APPEND st\_bdcdata TO it\_bdcdata.  
  ELSE.  
  
    MOVE: name  TO st\_bdcdata-fnam,  
          value TO st\_bdcdata-fval.  
    APPEND st\_bdcdata TO it\_bdcdata.  
  
  ENDIF.  
"  prepara a estrutura para o Loop.  
  CLEAR st\_bdcdata.  
ENDFORM.                    " Z\_PREENCHE\_BDC  
\*&---------------------------------------------------------------------\*  
\*&      Form  Z\_CARREGA\_TRANSACAO  
\*&---------------------------------------------------------------------\*  
FORM z\_carrega\_transacao .  
"  Chama a trasação informada usando a tabela 'IT\_BDCDATA que na verdade  
" é um shdb que vai automátiza o processo até o ultimo registro, existe  
" dois metodos de ver o processo, no 'MODE se colocar 'N' vc não visualiza  
" o processo, agora se vc coloca 'A' no 'MODE vc terá que apertar 'Enter  
" por processo e depois de efetuar o processo por registro ele popula a  
" tabela 'IT\_MSG com os dados de retorno,se foi realizado com exito, ou se  
" deu algum problema.  
  CALL TRANSACTION 'XD02' USING it\_bdcdata  
                          MODE  vg\_mode  
                          UPDATE vg\_s  
                          MESSAGES INTO it\_msg  
                            .  
  
ENDFORM.                    " Z\_CARREGA\_TRANSACAO  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  Z\_IMPRIME\_MENSAGEM  
\*&---------------------------------------------------------------------\*  
FORM z\_imprime\_mensagem.  
"  Dá 'Loop na tabela de retorno da chamada da transação e alimenta outra  
" tabela com a retorno referenciado com o Cliente  
  LOOP AT it\_msg.  
    msgno = it\_msg-msgnr.  
    "  Function que faz mostrar a mensagem  
    CALL FUNCTION 'WRITE\_MESSAGE'  
      EXPORTING  
        msgid         = it\_msg-msgid  
        msgno         = msgno  
        msgty         = it\_msg-msgtyp  
        msgv1         = it\_msg-msgv1  
        msgv2         = it\_msg-msgv2  
        msgv3         = it\_msg-msgv3  
        msgv4         = it\_msg-msgv4  
        msgv5         = ' '  
     IMPORTING  
\*       ERROR         =  
        messg         = mensg  
\*       MSGLN         =  
              .  
  ENDLOOP.  
  
  st\_message-cliente = st\_arqui-cliente.  
  st\_message-msgty   = mensg-msgty.  
  st\_message-msgno   = mensg-msgno.  
  st\_message-msgtx   = mensg-msgtx.  
" popula a tabela principal de mensagem que será o Log de erro.  
  APPEND st\_message TO it\_message.  
  
\*   WRITE: / , st\_arqui-cliente ,  
\*              mensg-msgtx,  
\*              mensg-msgty  
\*           .  
  
ENDFORM.                    " Z\_IMPRIME\_MENSAGEM  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  Z\_MANDA\_ARQUIVO  
\*&---------------------------------------------------------------------\*  
FORM z\_manda\_arquivo .  
"  Cria um arquivo externo conforme 'Filename com as informações da 'Data\_tab  
  CALL FUNCTION 'GUI\_DOWNLOAD'  
    EXPORTING  
      filename = p\_log  
    TABLES  
      data\_tab = it\_message  
    EXCEPTIONS  
      OTHERS   = 17.  
ENDFORM.                    " Z\_MANDA\_ARQUIVO  
  
\*&---------------------------------------------------------------------\*  
\*&      Form  Z\_SAPGUI\_PROGRESS\_INDICATOR  
\*&---------------------------------------------------------------------\*  
FORM z\_sapgui\_progress\_indicator  USING texto.  
  
  CALL FUNCTION 'SAPGUI\_PROGRESS\_INDICATOR'  
    EXPORTING  
      percentage = 10  
      text       = texto.  
  
ENDFORM.                    " Z\_SAPGUI\_PROGRESS\_INDICATOR  
  
"  Debugem isso, estudem, façam de outra forma pois não eh tão facil assim,  
" apertem F1, muda o 'MODE no 'CALL TRANSACTION pra ver processo por processo  
" abusem deste programa, olhem abaixo o SHDB possivel para este Batch input,  
" provavelmente deve ter mais campo mostrando valores mas nós não precisavamos  
" cadatrar outros campos, e qualquer dúvida e soh fala, t+.  
  
  
"                                               T XD02  BS AA X   F  
"SAPMF02D 0101  X  
"                                                 BDC\_CURSOR  RF02D-D0110  
"                                                 BDC\_OKCODE  /00  
"                                                 RF02D-KUNNR 8000000001  
"                                                 RF02D-D0110 X  
"SAPMF02D 0110  X  
"                                                 BDC\_CURSOR  KNA1-TELF1  
"                                                 BDC\_OKCODE  /00  
"                                                 KNA1-NAME1  teste  
"                                                 KNA1-STRAS  rua teste  
"                                                 KNA1-ORT01  São Paulo  
"                                                 KNA1-TELF1  12345678*