

# Batch Input

Guía para la creación de un Batch Input

## Guía para la creación de un Batch Input.

Un Batch Input es un proceso automatizado para realizar la ejecución de una transacción. Es muy útil emplearlos a la hora de tener que hacer una carga masiva de datos en SAP, los cuales pueden encontrarse alojados en un archivo o fichero de texto plano. Los Batch Input permiten simular la interacción del usuario con el programa que está siendo ejecutado, codificando los pasos que deberían llevarse a cabo y obteniendo los mismos resultados.

En este ejemplo, crearemos un programa que llame a una transacción estándar del SAP, la cual nos permite modificar ciertos campos de una tabla estándar. Las modificaciones necesarias serán entregadas al sistema a través de un archivo de texto plano, por lo cual llamaremos a una función para poder subir esta información y cargarla en una tabla interna, para luego pasar dichos valores por medio de variables al código que simula una carga manual, la cual es reemplazada por la automatización del Batch Input.

Los datos son:

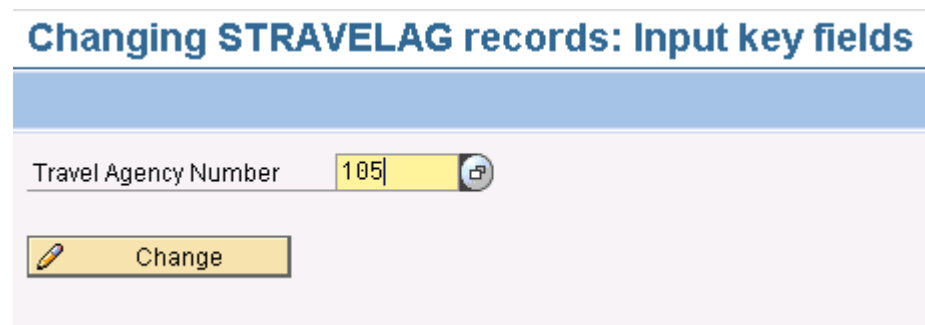
Transacción: BC\_GLOBAL\_STRAV\_EDIT

Tabla: STRAVELAG

<u>Campos de la tabla:</u>	AGENCYNUM	NUM(8)	clave primaria
	POSTCODE	CHAR(10)	

Secuencia de pasos:

Ingresamos a la transacción **BC\_GLOBAL\_STRAV\_EDIT** para entender como trabaja y cuales son los parámetros que debemos ingresar y cuales cambiar.



The screenshot shows a SAP dialog box titled "Changing STRAVELAG records: Input key fields". It has a light blue header bar. Below the header, there is a text input field labeled "Travel Agency Number" containing the value "105". To the right of the input field is a small circular icon with a square inside. Below the input field is a yellow button with a pencil icon and the text "Change".

Si accedemos a la tabla **STRAVELA** por medio de la **SE11**, veremos que **Travel Agency Number** es clave primaria y corresponde al tipo de dato **AGENCYNUM** que es un numérico de 8.

Seleccionamos un número que exista en nuestra tabla (en este caso el 105) y accedemos a la información que está disponible para ser cambiada:

## Changing STRAVELAG records: Changing a record

Travel Agency Number	105
Travel agency name	Travel from Walldorf
Street	Altonaer Str. 24
PO Box	
Postal Code	11111
City	Berlin
Country	DE
Region	11
Tel. no.	+49 30-622860
Travel agency URL	http://www.travel-from-walldorf
Language Key	DE

Vemos que uno de los campos que pueden ser modificados es el **Postal Code**, que, volviendo a la tabla observamos que es del tipo **POSTCODE** y es un char de 10.

Entonces ahora que sabemos que vamos a cambiar y como actúa esta transacción, vamos a acceder a la **SM35** (también puede hacerse directamente por la transacción **SHDB**) para ver cuales son los códigos que hay que seguir y así crear nuestro Batch Input.

Seleccionamos **Recording**

**Batch Input: Session Overview**

Analysis Process Statistics Log **Recording**

Selection criteria  
Sess.: \* From: To: Created by: \*

New Incorrect Processed In Process In Background Being Created Locked

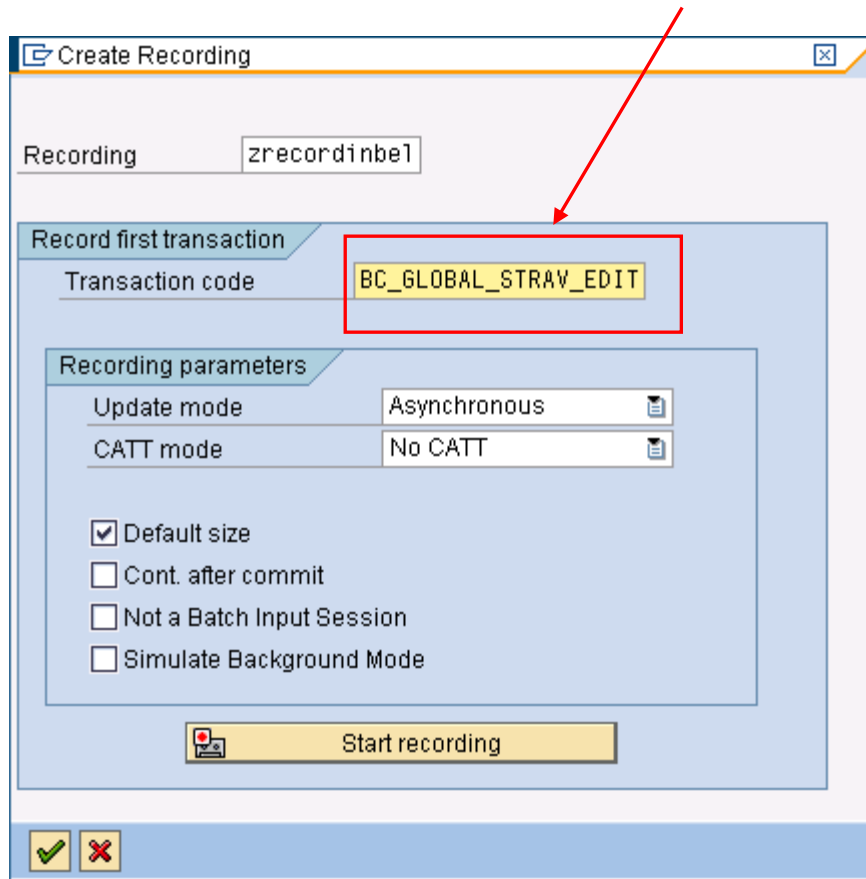
	Session name	Stat...	Created By	Date	Time	Creation Prog...	Lock Date	Authorizat.	Σ Trans.
<input type="checkbox"/>	BEL	<input type="checkbox"/>	BCUSER	28.03.2008	18:25:26	ZCODIGOPST...		BCUSER	1 0
<input type="checkbox"/>	BEL	<input type="checkbox"/>	BCUSER	28.03.2008	18:19:32	ZCODIGOPST...		BCUSER	1 0
<input type="checkbox"/>	BE	<input type="checkbox"/>	BCUSER	28.03.2008	18:17:11	ZCODIGOPST...		BCUSER	1 0
<input type="checkbox"/>	BELEN	<input type="checkbox"/>	BCUSER	28.03.2008	18:16:20	ZCODIGOPST...		BCUSER	1 0

Y Luego en la barra seleccionamos **New Recording**

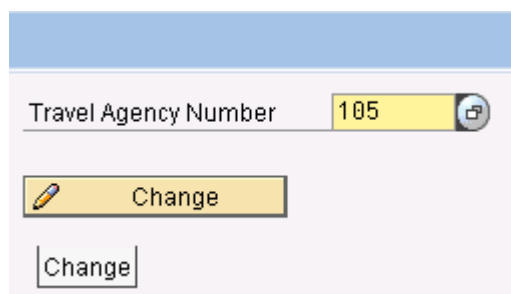
**Transaction Recorder: Recording Overview**

**New recording** Process Session Program Test data Function module

Debemos rellenar el nombre de nuestro recording para, después de una vez creado, buscarlo dentro de la tabla de esta transacción y en **Transaction code** escribir la transacción que queremos reproducir **BC\_GLOBAL\_STRAV\_EDIT**



Se ejecuta la transacción y seleccionamos el **Travel Agency Number** que queremos modificar y luego presionamos en **Change**:





A continuación nos aparecen los datos posibles de modificar. Buscamos el **Postal Code** y modificamos el 1111 por el 99999


Travel Agency Number	105
Travel agency name	Travel from Walldorf
Street	Altonaer Str. 24
PO Box	
Postal Code	99999
City	Berlin
Country	DE
Region	11
Tel. no.	+49 30-622860
Travel agency URL	http://www.travel-from-walldorf
Language Key	DE

Presionamos  y a continuación 

Nos aparecerá el siguiente pop-up y seleccionamos **Yes**

 Back

 Save data first?

Nuevamente presionamos  y volveremos a la transacción **SM35** dónde se mostrara una tabla con todos los pasos que hemos realizado dentro de la transacción **BC\_GLOBAL\_STRAV\_EDIT**

Podemos observar varios datos:

Figura el nombre de la transacción, el nombre del programa que ejecuta dicha transacción y todas las variables y eventos que están surgiendo a medida que navegamos por las distintas screen.

Para nuestro ejercicio sólo necesitaremos los datos remarcados por cuadros

## Transaction Recorder: Change Recording ZRECORDINBEL

	Program	Screen	St...	Field name	Field value
1			T	BDC_GLOBAL_STRAV_EDIT	
2	SAPBC_GLO...	0100	X		
3				BDC_CURSOR	STRAVELAG-AGENCYNUM
4				BDC_OKCODE	/00
5				STRAVELAG-AGENCYNUM	105
6	SAPBC_GLO...	0200	X		
7				BDC_CURSOR	STRAVELAG-POSTCODE
8				BDC_OKCODE	=SAVE
9				STRAVELAG-NAME	Travel from Walldorf
10				STRAVELAG-STREET	Altonaer Str. 24
11				STRAVELAG-POSTCODE	99999
12				STRAVELAG-CITY	Berlin
13				STRAVELAG-COUNTRY	DE
14				STRAVELAG-REGION	11
15				STRAVELAG-TELEPHONE	+49 30-622860
16				STRAVELAG-URL	http://www.travel-from-walldorf
17				STRAVELAG-LANGU	DE

	Program	Screen	St...	Field name	Field value
14				STRAVELAG-REGION	11
15				STRAVELAG-TELEPHONE	+49 30-622860
16				STRAVELAG-URL	http://www.travel-from-walldorf
17				STRAVELAG-LANGU	DE
18	SAPBC_GLO...	0200	X		
19				BDC_CURSOR	STRAVELAG-NAME
20				BDC_OKCODE	=BACK
21				STRAVELAG-NAME	Travel from Walldorf
22				STRAVELAG-STREET	Altonaer Str. 24
23				STRAVELAG-POSTCODE	99999
24				STRAVELAG-CITY	Berlin
25				STRAVELAG-COUNTRY	DE
26				STRAVELAG-REGION	11
27				STRAVELAG-TELEPHONE	+49 30-622860
28				STRAVELAG-URL	http://www.travel-from-walldorf
29				STRAVELAG-LANGU	DE
30	SAPLSP01	0100	X		
31				BDC_OKCODE	=YES
32	SAPBC_GLO...	0100	X		
33				BDC_CURSOR	STRAVELAG-AGENCYNUM
34				BDC_OKCODE	=BACK
35				STRAVELAG-AGENCYNUM	105

Con los cuales generaremos el siguiente código en un reporte nuevo.

Para ello nos dirigimos a la **SE38** y creamos el reporte

The screenshot shows the 'ABAP Editor: Initial Screen' with a toolbar at the top containing icons for file operations, debugging, and variants. Below the toolbar, the 'Program' field is set to 'ZBATCH\_INPUT\_1' and a 'Create' button is visible. A dialog box titled 'ABAP: Program Attributes ZBATCH\_INPUT\_1 Change' is open, displaying various attributes for the program. The 'Attributes' section includes fields for Type (1 Executable program), Status, Application, Authorization Group, Package (\$TMP), Logical database, and Selection screen. There are also checkboxes for Editor lock, Unicode Checks Active, Fixed point arithmetic, and Start using variant. The 'Created' and 'Last Changed' fields are both set to 09.07.2018 by LGONZALE.

Y escribiremos el siguiente programa:

```
*&-----*
*& Report  ZBATCH_INPUT_1
*&
*&-----*
*& Este es un reporte que por medio de un batch input ejecuta una
*& transacción estandar (BC_GLOBAL_STRAV_EDIT, la cual cambiará un
*& campo (POSTCODE) de una tabla standard (STRAVELAG) por medio de un
*& archivo de texto (C:\codigos.txt) invocando a la función GUI_UPLOAD
*&-----*
REPORT zbatch_input_1.

*Estructura interna para almacenar el AGENCYNUM (clave primaria de la
*tabla) y el POSTCODE (campo a modificar) como los datos se reciben
*desde un txt, todos los campos de la estructura son de tipo Char
TYPES: BEGIN OF ty_codigos,
        nro(8)      TYPE c,
        codigo(10)  TYPE c,
      END OF ty_codigos,
```

```

BEGIN OF ty_log,
    light      TYPE icon_d,
    agencynum  TYPE stravelag-agencynum,
    message    TYPE bapiret2-message,
END OF ty_log.

*Estructura con los datos del programa.
DATA: gt_bdc_tab TYPE TABLE OF bdcdata,
      gs_bdc_tab TYPE bdcdata.

*Estructura con mensajes de error y otros
DATA: gt_messtab TYPE TABLE OF bdcmsgcoll,
      gs_messtab TYPE bdcmsgcoll.

DATA: gt_log TYPE TABLE OF ty_log,
      gs_log TYPE ty_log.

*Defino la tabla interna y la working area de la estructura.
DATA: gs_codigos TYPE ty_codigos,
      gt_codigos TYPE TABLE OF ty_codigos.

*Llamamos a la funcion GUI_UPLOAD para traer el documento que tiene
*los códigos que hay que modificar con los datos ordenados y de igual
*longitud que la estructura interna ty_codigos
CALL FUNCTION 'GUI_UPLOAD'
  EXPORTING
    filename      = 'C:\test\codigos.txt'
    filetype      = 'ASC'
  TABLES
    data_tab      = gt_codigos
  EXCEPTIONS
    file_open_error      = 1
    file_read_error      = 2
    no_batch              = 3
    gui_refuse_filetransfer = 4
    invalid_type          = 5
    no_authority          = 6
    unknown_error         = 7
    bad_data_format       = 8
    header_not_allowed    = 9
    separator_not_allowed = 10
    header_too_long       = 11
    unknown_dp_error      = 12
    access_denied         = 13
    dp_out_of_memory      = 14
    disk_full             = 15
    dp_timeout            = 16
    OTHERS                = 17.

IF sy-subrc <> 0.
  MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno
    WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.

*Aquí tenemos definidos los pasos que la transacción SM35 nos mostró
*al ejecutar la transacción BC_GLOBAL_STRAV_EDIT. Estos pasos son
*colocados dentro de un LOOP donde se lee la tabla importada por la
*función anterior (gt_codigos) y es cargada en la working area
*(gs_codigos), la cual se ha completando con los datos del txt para
*luego ser traspasados al código como variables (y no hardcodeadas).

```



```

LOOP AT gt_codigos INTO gs_codigos.

    REFRESH: gt_bdc_tab, gt_messtab.

*Este perform va cargando una tabla interna (gt_bdc_tab) dónde se
*guardarán los datos generados por la SM35, para que, cuando es
*invocada la transacción(BC_GLOBAL_STRAV_EDIT) acceda a dicha tabla
*y simule el trabajo de un usuario.

    PERFORM fill_bdc_tab USING:

        'X' 'SAPBC_GLOBAL_STRAVELAG_EDIT' '0100',
        ' ' 'BDC_OKCODE' '/00', "Ingresa el AGENCYNUM del txt
        ' ' 'STRAVELAG-AGENCYNUM' gs_codigos-nro,

        'X' 'SAPBC_GLOBAL_STRAVELAG_EDIT' '0200',
        ' ' 'BDC_OKCODE' '=SAVE', "Ingresa el POSTCODE del txt
        ' ' 'STRAVELAG-POSTCODE' gs_codigos-codigo,
        'X' 'SAPBC_GLOBAL_STRAVELAG_EDIT' '0200',
        ' ' 'BDC_OKCODE' '=BACK',

        'X' 'SAPLSPO1' '0100',
        ' ' 'BDC_OKCODE' '=YES',

        'X' 'SAPBC_GLOBAL_STRAVELAG_EDIT' '0100',
        ' ' 'BDC_OKCODE' '=BACK'.

*Llamamos a la transacción BC_GLOBAL_STRAV_EDIT para que se ejecute
*de acuerdo*a los datos cargados anteriormente (en la tabla bdc_tab)
*usando el:
*Modo A: modo para ver paso a paso la ejecución automática que nos
*creo el código de la Trx. SM35.
*Modo N: ejecuta la transacción sin la necesidad de que el usuario
*interactúe con la continuidad del mismo
*Modo E: en este caso la transacción sólo requerirá intervención del
*usuario si trae alguna excepción

    CALL TRANSACTION 'BC_GLOBAL_STRAV_EDIT' USING gt_bdc_tab
                                         MODE 'N'
                                         MESSAGES INTO gt_messtab.

    READ TABLE gt_messtab INTO gs_messtab WITH KEY msgtyp = 'E'.
    IF sy-subrc NE 0.

        gs_log-light      = icon_green_light.
        gs_log-agencynum  = gs_codigos-nro.
        gs_log-message    = 'La modificación se realizó con éxito.'.

        APPEND gs_log TO gt_log.
        CLEAR gs_log.

    ELSE.

        LOOP AT gt_messtab INTO gs_messtab.

```

```

CALL FUNCTION 'FORMAT_MESSAGE'
EXPORTING
    id      = gs_messtab-msgid
    lang    = sy-langu
    no      = gs_messtab-msgnr
    v1      = gs_messtab-msgv1
    v2      = gs_messtab-msgv2
    v3      = gs_messtab-msgv3
    v4      = gs_messtab-msgv4
IMPORTING
    msg     = gs_log-message
EXCEPTIONS
    not_found = 1
    OTHERS    = 2.

ENDLOOP.

gs_log-light      = icon_red_light.
gs_log-agencynum = gs_codigos-nro.

APPEND gs_log TO gt_log.
CLEAR gs_log.

ENDIF.

ENDLOOP.

IF gt_log IS NOT INITIAL.

    LOOP AT gt_log INTO gs_log.
        WRITE:/ gs_log-light,
            AT 8 gs_log-agencynum,
            AT 20 gs_log-message.
    ENDLOOP.

ENDIF.

*&-----*
*&      Form  FILL_BDC_TAB
*&-----*
*Carga la tabla bdc_tab de acuerdo a los datos de la SM35
FORM fill_bdc_tab USING dynbegin TYPE bdcdata-dynbegin
                        name      TYPE any
                        value     TYPE any.

    IF dynbegin = 'X'.

        CLEAR gs_bdc_tab.
        MOVE: name TO gs_bdc_tab-program,
              value TO gs_bdc_tab-dynpro,
              'X' TO gs_bdc_tab-dynbegin.

        APPEND gs_bdc_tab TO gt_bdc_tab.

    ELSE.

        CLEAR gs_bdc_tab.
        MOVE: name TO gs_bdc_tab-fnam,
              value TO gs_bdc_tab-fval.

        APPEND gs_bdc_tab TO gt_bdc_tab.

```

```

ENDIF.

ENDFORM.                  " FILL_BDC_TAB

*Fin del programa.

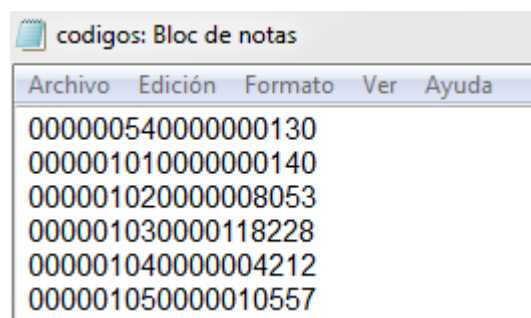
```

Luego de haber compilado y activado el programa, lo ejecutamos y veremos que la transacción **BC\_GLOBAL\_STRAV\_EDIT** es ejecutada automáticamente, ingresando en cada uno de los campos, el código postal que había en el archivo txt.

Ejemplo de BI para Trx. BC_GLOBAL_STRAV_EDIT		
●○○○	00000054	Agency 00000054 not found (table STRAVELAG)
○○■	00000101	La modificación se realizó con éxito.
○○■	00000102	La modificación se realizó con éxito.
○○■	00000103	La modificación se realizó con éxito.
○○■	00000104	La modificación se realizó con éxito.
○○■	00000105	La modificación se realizó con éxito.

Pueden cambiarse el resto de los campos de la tabla, modificar el código y ver como van impactando estos cambios gracias a la ejecución del Batch Input.

Nota: el archivo txt creado, debe contener los datos que se desean cargar en la tabla, repetando la longitud de los mismo, o en caso necesario trabajar los datos recogidos de tal modo que queden de la misma manera que el tipo de dato del campo de la tabla.  
Ejemplo:



```

codigos: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
0000005400000000130
0000010100000000140
00000102000000008053
0000010300000118228
0000010400000004212
0000010500000010557

```

Es recomendable entrar a la transacción SE11 o SE16 para observar que los datos ingresados coinciden con los del txt y que están completos.