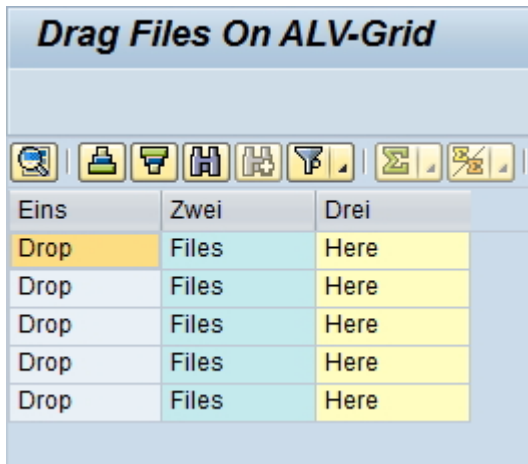


## Drop Files in ALV-Grid



Seit etwa 2012 ist es möglich, Dateien aus dem Explorer in ein ALV-Grid zu ziehen und die Namen dort zu verarbeiten (SAP Note 1748468). Ob das in deinem SAP-System bereits funktioniert, kannst du daran erkennen, dass die Klasse `CL_GUI_ALV_GRID` die Methode `GET_DROPPED_EXTERNAL_FILES` hat.

Das Event `DROP_EXTERNAL_FILES` muss registriert werden und mit der Methode `DRAG_ACCEPT_FILES( 1 )` muss die Funktionalität aktiviert werden.

Es kann das gesamte ALV-Grid-Control als „Ablagefläche“ genutzt werden, also auch dort, wo keine Zeilen vorhanden sind. Werden die Dateien auf einer Zelle abgelegt, dann kann diese Zelle bestimmt werden. In dem Beispielcoding werden die Dateien der Spalte hinzugefügt, auf der sie abgelegt wurden.

## Coding

```
REPORT zz_drop_files_on_grid.
```

```
*-----*
*          CLASS lcl_appl DEFINITION
*-----*
```

```
CLASS lcl_appl DEFINITION.
```

```
    PUBLIC SECTION.
```

```
*== type for Grid-Demonstration: Columns EINS, ZWEI and DREI
    TYPES: BEGIN OF ty_files,
            eins TYPE string,
```

```
        zwei TYPE string,  
        drei TYPE string,  
    END OF ty_files.
```

```
*== table containing the data (text/ files)  
    CLASS-DATA gt_data TYPE STANDARD TABLE OF ty_files.
```

```
*== ALV-Grid  
    CLASS-DATA gr_grid TYPE REF TO cl_gui_alv_grid.
```

```
*== setup ALV-Grid  
    CLASS-METHODS init.
```

```
*== get dropped files and display in grid  
    CLASS-METHODS get_dropped_files FOR EVENT drop_external_files OF  
cl_gui_alv_grid
```

```
        IMPORTING files.
```

```
ENDCLASS.                "lcl_appl DEFINITION
```

```
*-----*  
*          CLASS lcl_appl IMPLEMENTATION  
*-----*  
CLASS lcl_appl IMPLEMENTATION.
```

```
    METHOD init.
```

```
*== local data  
    DATA lt_fcat          TYPE lvc_t_fcat.  
    FIELD-SYMBOLS <fcat> LIKE LINE OF lt_fcat.  
    FIELD-SYMBOLS <data> LIKE LINE OF gt_data.
```

```
    CHECK gr_grid IS INITIAL.
```

```
*== set initial data  
    DO 5 TIMES.  
        APPEND INITIAL LINE TO gt_data ASSIGNING <data>.  
        <data>-eins = 'Drop'.  
        <data>-zwei = 'Files'.  
        <data>-drei = 'Here'.  
    ENDDO.
```

```
*== Create Grid-control  
    CREATE OBJECT gr_grid  
        EXPORTING  
            i_parent = cl_gui_container=>screen0.
```

```
*== build field catalog  
    APPEND INITIAL LINE TO lt_fcat ASSIGNING <fcat>.  
    <fcat>-fieldname = 'EINS'.  
    <fcat>-reptext   = 'Eins'.  
    <fcat>-style     = 1.  
    <fcat>-datatype  = 'STRG'.
```

```

APPEND INITIAL LINE TO lt_fcat ASSIGNING <fcat>.
<fcat>-fieldname = 'ZWEI'.
<fcat>-reptext   = 'Zwei'.
<fcat>-style     = 2.
<fcat>-datatype  = 'STRG'.
APPEND INITIAL LINE TO lt_fcat ASSIGNING <fcat>.
<fcat>-fieldname = 'DREI'.
<fcat>-reptext   = 'Drei'.
<fcat>-style     = 4.
<fcat>-datatype  = 'STRG'.

*== display grid
gr_grid->set_table_for_first_display(
    CHANGING it_fieldcatalog = lt_fcat
             it_outtab      = gt_data ).

*== Allow drop files action
gr_grid->drag_accept_files( 1 ).

*== set handler to react on file drop
SET HANDLER get_dropped_files FOR gr_grid ACTIVATION abap_true.

ENDMETHOD.                                     "init

METHOD get_dropped_files.

*== local data
DATA lt_files          TYPE filetable.
FIELD-SYMBOLS <file> LIKE LINE OF lt_files.
DATA lv_row_id        TYPE i.
DATA lv_col_id        TYPE i.
FIELD-SYMBOLS <data> LIKE LINE OF gt_data.

*== get dropped files in a table
gr_grid->get_dropped_external_files(
    IMPORTING files      = lt_files
              row_id     = lv_row_id
              col_id     = lv_col_id
    EXCEPTIONS OTHERS   = 3 ).

*== put file names in grid
IF lt_files IS NOT INITIAL.
    CLEAR gt_data.
    LOOP AT lt_files ASSIGNING <file>.
        APPEND INITIAL LINE TO gt_data ASSIGNING <data>.
        CASE lv_col_id.
            WHEN 1. <data>-eins = <file>.
            WHEN 2. <data>-zwei = <file>.
            WHEN 3. <data>-drei = <file>.
        ENDCASE.
    ENDLLOOP.

```

```

*== refresh display
    gr_grid->refresh_table_display( ).
ENDIF.

ENDMETHOD.                                "get_dropped_files

ENDCLASS.                                  "lcl_appl
IMPLEMENTATION

START-OF-SELECTION.

*== call screen
    CALL SCREEN 100.

*-----*
*  MODULE 100 OUTPUT
*-----*
MODULE pbo_100 OUTPUT.

*== Use system status (so no status must be created; we only need "Leave
Screen")
    SET PF-STATUS 'STLI' OF PROGRAM 'SAPMSSY0'.
*== init controls/ application
    lcl_appl=>init( ).

ENDMODULE.                                "100 OUTPUT

*-----*
*  MODULE 100 INPUT
*-----*
MODULE pai_100 INPUT.
    CASE sy-ucomm.
        WHEN 'BACK' OR '%EX' OR 'RW'.
            SET SCREEN 0. LEAVE SCREEN.
    ENDCASE.
ENDMODULE.                                "100 INPUT

```