

SDN Community Contribution

(This is not an official SAP document.)

Disclaimer & Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.

Table of Content

ABAP Code WebViewer Utility	1
Table of Content	2
Applies To:	2
Article Summary	2
ABAP Code WebViewer Utility by www.rmtiwari.com	2
Benefits:	2
Steps to create this utility program in your SAP system:	3
Code :	3
Disclaimer & Liability Notice	Error! Bookmark not defined.

Applies To:

ABAP Reports / Function Module codes shared on Web.

Article Summary

As ABAP developers, we find it difficult to read ABAP codes shared on web since navigation is not possible by clicking on Form Routines / Variables. Also documentations / comments & Codes are not separated using color schemes. e.g. Blue for comments.

ABAP Code WebViewer is a utility tool, written in ABAP, which converts the ABAP Report / Function Module Codes into an HTML file and downloads it on the chosen directory of your PC. Further, developers can share the code on websites by directly by sending it in an HTML format.

By: Ram Manohar Tiwari
Title: ABAP Code WebViewer
Date: 10 Apr 2005

ABAP Code WebViewer Utility by www.rmtiwari.com

This utility tool, written in ABAP, converts the ABAP Report / Function Module Codes into an HTML file and downloads it on the chosen directory of your PC. Further, developers can share the code on websites by directly by sending it in an HTML format.

This utility is developed on MiniSAP and should be compatible with SAP4.7. Though with slight modifications to remove the value added aspects will make it compatible with any versions of SAP.

Demo Link:

Click on [ABAP Code WebViewer by www.rmtiwari.com](http://www.rmtiwari.com) to check out the HTML code generated by this utility for itself.

Benefits:

ABAP Codes downloaded by this utility will have better readability on web as:

- ◆ It retains the blue color of comments and documentation with the code and hence HTML version looks exactly like the ABAP editor code.
- ◆ The generated HTML will be search engine friendly as it will have its own title and meta search keywords. Code Sharer has the option of providing his own <title> tag suitable for the program and relevant meta keywords. It enables the shared code to be ranked on search engines the way it is intended by the code sharer.
- ◆ Navigation is provided in the generated HTML using anchor tags. Hence, just like ABAP editor, viewers can click on a variable to navigate directly at its DATA definition.
- ◆ Similarly, viewers can click on PERFORM calls to reach directly at the defined FORM routine.

Steps to create this utility program in your SAP system:

- Create a new ABAP Report and upload the text version (Code given below) of this program.
- Create text elements for the new program.

Text Symbols:

- 001 Choose either a Report or a Function Module
- 002 Choose Output Directory on your PC
- 003 Make your code Search Engine Friendly on web.

Selection Text:

- P_DESC Program Description
- P_KEYW Keywords (Separated by comma)
- P_OUTDIR Output Directory
- P_PROG Report
- P_PROGF Function Module
- Check and activate.
- Execute and enter selection parameters.

Code :

```
REPORT  Z_RMTIWARI_ABAP_CODE_WEBVIEWER .
```

```
*-----
*
*      ABAP Code WebViewer Beta Version - Date - 20.03.2005
*-----
*
*      Written By: Ram Manohar Tiwari
*-----
*
*      Presented By: http://www.rmtiwari.com
*-----
*
```

```

* This utility tool converts the ABAP Report / FM Code into an html
* file and downloads it on the chosen directory of your PC.
* Further, developers can share the code on websites by directly
* sending it in an HTML format.
* The code downloaded by this utility is more readable on web for an
* ABAP developer as:
*     1. It retains the blue color of comments and documentation with
*         the code and hence HTML version looks exactly like the ABAP
*         editor code.
*
*     2. The generated HTML will be search engine friendly as it will
*         have its own title and meta search keywords. This way search
*         engines can directly find the code on your site. Code Sharer
*         has the option of providing his own <title> suitable for the
*         program and relevant meta keywords to find it on web.
*
*     3. Just like ABAP editor, viewers can click on a variable to
*         reach its directly at its DATA definition.
*
*     4. Similarly, viewers can click on PERFORM calls to reach
*         directly at the defined FORM routine.
*-----
-
* This utility is developed on MiniSAP and should be compatible with
* SAP4.7 version. Though with slight modifications to remove
* the value added aspects will make it compatible with any versions of
* SAP.
*-----
-

* SELECTION SCREEN
*-----
-
SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME TITLE text-001.
PARAMETERS : p_prog      TYPE trdir-name,
              p_progf     TYPE tfdir-funcname.
SELECTION-SCREEN END OF BLOCK b1 .

SELECTION-SCREEN BEGIN OF BLOCK B2 WITH FRAME TITLE text-002.

SELECTION-SCREEN BEGIN OF LINE.
PARAMETERS: P_OUTDIR LIKE rlgrap-filename obligatory.
SELECTION-SCREEN END OF LINE.
SELECTION-SCREEN END OF BLOCK B2.

SELECTION-SCREEN BEGIN OF BLOCK b3 WITH FRAME TITLE text-003.

PARAMETERS: P_DESC      TYPE PFELINE.

PARAMETERS: P_KEYW      TYPE PFELINE.

SELECTION-SCREEN END OF BLOCK b3 .

*-----
*
* Table declaration

```

```

*-----
*
TABLES: tfdir,
        trdir.

*-----
-

data : gv_directory LIKE rlgrap-filename,
        gv_file_name TYPE STRING,
        gv_html_file_name TYPE trdir-name.
DATA : BEGIN OF gt_html OCCURS 0,
        rec(200) TYPE c,
        END OF gt_html.

DATA : gv_html_wa          like line of gt_html.

*-----
*
* Type declaration
*-----
*
* Type for report
TYPES: BEGIN OF ty_report_wo_index,
        line(300),
        END OF ty_report_wo_index.

* Type for Report with line index
TYPES: BEGIN OF ty_report_with_index,
        index TYPE i,
        line(300),
        END OF ty_report_with_index.

*-----
*
* CONSTANTS
*-----
*
CONSTANTS : gc_false(1)      TYPE c VALUE ' ',
            gc_true(1)       TYPE c VALUE 'X',
            gc_quotes1(1)    TYPE c VALUE ' ',
            gc_asterik(1)    TYPE c VALUE '*',
            gc_comma(1)      TYPE c VALUE ',',
            gc_quotes(1)     TYPE c VALUE '"',
            gc_percent(1)    TYPE c VALUE '%'.

*-----
*
* Data declaration
*-----
*

DATA : gt_rep_table          TYPE STANDARD TABLE OF ty_report_wo_index WITH
                                NON-UNIQUE DEFAULT KEY INITIAL SIZE 500,
        gv_rep_wa            TYPE ty_report_wo_index,
        gv_rep_wa_next       TYPE ty_report_wo_index.

```

```

DATA : gv_prog_name      TYPE trdir-name.
*-----
*
AT SELECTION-SCREEN ON VALUE-REQUEST FOR p_prog.
    PERFORM get_report_name USING      'P_PROG'
                                CHANGING p_prog.

AT SELECTION-SCREEN ON VALUE-REQUEST FOR p_progf.
    PERFORM get_report_name USING      'P_PROGF'
                                CHANGING p_progf.

AT SELECTION-SCREEN ON VALUE-REQUEST FOR p_outdir.
    PERFORM get_directory_name USING    'P_OUTDIR'
                                CHANGING P_OUTDIR.

AT SELECTION-SCREEN.
    PERFORM selection_screen_validations.

START-OF-SELECTION.

* read the report code lines in an internal table
READ REPORT gv_prog_name INTO gt_rep_table.
IF sy-subrc <> 0.
    MESSAGE e001(AQ) WITH
        'Code is blank'.
    EXIT.
ENDIF.

* Read the lines of report one by one and convert to HTML.
PERFORM convert_code_to_html TABLES gt_rep_table
                                gt_html.

PERFORM prepare_file_name      USING    p_outdir
                                gv_html_file_name
                                CHANGING gv_file_name.

PERFORM download_html_file_on_pc tables    gt_html
                                using      gv_file_name.

PERFORM show_html_file using gv_file_name.

END-OF-SELECTION.

*&-----
*&
*&      Form  get_report_name
*&-----
*&
*&      text

```

```

*-----
*
* --> p1      text
* <-- p2      text
*-----
*
FORM get_report_name using      x_report_field_name
                           changing y_report_name.

PERFORM read_screen_values USING      x_report_field_name
                           CHANGING y_report_name.

IF x_report_field_name eq 'P_PROG'.

    CALL FUNCTION 'REPOSITORY_INFO_SYSTEM_F4'
        EXPORTING
            object_type          = 'PROG'
            object_name          = y_report_name
            suppress_selection    = 'X'
            without_personal_list = ' '
        IMPORTING
            object_name_selected = p_prog
        EXCEPTIONS
            cancel                = 01.

    ELSEIF x_report_field_name eq 'P_PROGPF'.
        CALL FUNCTION 'REPOSITORY_INFO_SYSTEM_F4'
            EXPORTING
                object_type          = 'FUNC'
                object_name          = y_report_name
                suppress_selection    = ' '
                without_personal_list = 'X'
            IMPORTING
                object_name_selected = p_progpf
            EXCEPTIONS
                cancel                = 01.
        ENDIF.

ENDFORM.                                " get_directory_name

*&-----
*&
*&      Form  get_file_name
*&-----
*&
*&      text
*&-----
*&
*& --> p1      text
*& <-- p2      text
*&-----
*&
FORM get_directory_name using      x_file_field_name
                           changing y_directory_name.

class CL_GUI_FRONTEND_SERVICES definition load.

```

```

data : lv_file_path type rlgrap-filename,
      lv_file_name type rlgrap-filename,
      lv_outdir    type string,
      lv_directory type string.

CONSTANTS : lc_window_title type string
            value 'Select HTML download folder'.

PERFORM read_screen_values USING    x_file_field_name
                                CHANGING y_directory_name.

lv_directory = y_directory_name.

CALL METHOD CL_GUI_FRONTEND_SERVICES=>DIRECTORY_BROWSE
EXPORTING
  WINDOW_TITLE      = lc_window_title
  INITIAL_FOLDER    = lv_directory
CHANGING
  SELECTED_FOLDER   = lv_outdir
EXCEPTIONS
  CNTL_ERROR        = 1
  ERROR_NO_GUI      = 2
  NOT_SUPPORTED_BY_GUI = 3.

if sy-subrc eq 0.
  p_outdir = lv_outdir.
endif.
ENDFORM.                    " get_directory_name
*&-----*
*&      Form  read_screen_values
*&-----*
*      text
*-----*
*      -->P_FIELD    text
*      -->P_VALUE    text
*-----*
FORM read_screen_values USING    x_field
                                CHANGING y_value.

DATA: lv_dynpname LIKE d020s-prog,
      lv_dynpnumb LIKE d020s-dnum.

DATA: BEGIN OF lt_dynpvaluetab OCCURS 1.
      INCLUDE STRUCTURE dynpread.
DATA: END   OF lt_dynpvaluetab.

lv_dynpname = sy-repid.
lv_dynpnumb = sy-dynnr.

REFRESH lt_dynpvaluetab.
lt_dynpvaluetab-fieldname = x_field.
APPEND lt_dynpvaluetab.

CALL FUNCTION 'DYNP_VALUES_READ'
EXPORTING

```

```

        dyname                = lv_dynpname
        dynumb                = lv_dynpnumb
TABLES
        dynpfields            = lt_dynpvaluetab
EXCEPTIONS
        invalid_abapworkarea = 1
        invalid_dynprofield  = 2
        invalid_dynprname    = 3
        invalid_dynpronummer = 4
        invalid_request       = 5
        no_fielddescription   = 6
        invalid_parameter     = 7
        undefind_error        = 8
        double_conversion     = 9
        OTHERS                 = 10.
IF sy-subrc = 0.

        READ TABLE lt_dynpvaluetab INDEX 1.
        MOVE: lt_dynpvaluetab-fieldvalue TO y_value.

ENDIF.

ENDFORM.                    " read_screen_values
*&-----
*
*&      Form  selection_screen_validations
*&-----
*
*      text
*-----
*
*  -->  p1      text
*  <--  p2      text
*-----
*
FORM selection_screen_validations .
        IF NOT p_prog  IS INITIAL AND
           NOT p_progf IS INITIAL.

                MESSAGE e001(AQ) WITH
                'Please use either program of function module'.
        ELSEIF p_prog  IS INITIAL AND
                p_progf IS INITIAL.
                MESSAGE e001(AQ) WITH
                'Please provide any of the program or function module name'.
        ELSEIF NOT p_progf IS INITIAL.
                SELECT SINGLE *
                FROM    tfdir
                WHERE    funcname = p_progf.
                IF sy-subrc EQ 0.
                        CONCATENATE tfdir-pname 'U'(793) tfdir-include INTO gv_prog_name.
                        gv_prog_name = gv_prog_name+3(30).
                        CONCATENATE p_progf '.html' into gv_html_file_name.
                ELSE.
                        MESSAGE e001(AQ) WITH
                        'Function module does not exit'.
                ENDIF.
        ENDIF.

```

```

ELSE.
    SELECT SINGLE *
    FROM trdir
    WHERE name = p_prog.
    IF sy-subrc NE 0.
        MESSAGE e001(AQ) WITH
        'This Report does not exist'.
    ELSE.
        gv_prog_name = p_prog.
        CONCATENATE p_prog '.html' into gv_html_file_name.
    ENDIF.
ENDIF.

ENDFORM.                                " selection_screen_validations

*&-----*
*&      Form  prepare_file_name
*&-----*
*      text
*-----*
*      -->x_directory      text
*      -->x_program_name text
*      -->y_file_name      text
*-----*
FORM prepare_file_name USING x_directory
                           x_file
                           CHANGING y_file_name.

CONCATENATE x_directory '\ ' x_file into y_file_name.

ENDFORM.                                " prepare_file_name
*&-----*
*
*&      Form  download_html_file_on_pc
*&-----*
*      text
*-----*
*
*      --> p1      text
*      <-- p2      text
*-----*
*
FORM download_html_file_on_pc tables  yt_download
                                using  x_outfile type string.

* Use gui_download if file is located on the local PC.
* WS_download only works in foreground
  IF sy-batch EQ 'X'.
    MESSAGE e001(AQ) WITH
    'This program cannot be executed in background'.
*   ERROR: Unable to download locally stored files when running in
*   background
  ELSE.

```

```

CALL FUNCTION 'GUI_DOWNLOAD'
  EXPORTING
    FILENAME          = x_outfile
    FILETYPE          = 'ASC'
  TABLES
    DATA_TAB         = yt_download
  EXCEPTIONS
    FILE_WRITE_ERROR  = 1
    NO_BATCH          = 2
    GUI_REFUSE_FILETRANSFER = 3
    INVALID_TYPE      = 4
    NO_AUTHORITY      = 5
    UNKNOWN_ERROR     = 6
    HEADER_NOT_ALLOWED = 7
    SEPARATOR_NOT_ALLOWED = 8
    FILESIZE_NOT_ALLOWED = 9
    HEADER_TOO_LONG   = 10
    DP_ERROR_CREATE    = 11
    DP_ERROR_SEND      = 12
    DP_ERROR_WRITE     = 13
    UNKNOWN_DP_ERROR   = 14
    ACCESS_DENIED      = 15
    DP_OUT_OF_MEMORY   = 16
    DISK_FULL          = 17
    DP_TIMEOUT         = 18
    FILE_NOT_FOUND     = 19
    DATAPROVIDER_EXCEPTION = 20
    CONTROL_FLUSH_ERROR = 21
    OTHERS             = 22.

* Status of download
CASE sy-subrc.
  WHEN 0.
    MESSAGE I002(AQ) WITH
      'Code converted to HTML and downloaded as ' x_outfile.

  WHEN OTHERS.
*    Upload unsuccessful - error message
    MESSAGE ID SY-MSGID TYPE 'E' NUMBER SY-MSGNO
      WITH SY-MSGV1 SY-MSGV2 SY-MSGV3 SY-MSGV4.
ENDCASE.
ENDIF.

ENDFORM.                  " download_html_file_on_pc
*&-----
*
*&      Form  convert_code_to_html
*&-----
*
*      text
*-----
*
*      -->P_GT_REP_TABLE  text
*      -->P_GT_HTML      text
*-----
*
FORM convert_code_to_html  TABLES xt_REP_TABLE

```

yt_HTML.

```
DEFINE add_html.  
  yt_html = &1.  
  APPEND yt_html.
```

END-OF-DEFINITION.

```
DATA : lv_next_line_index type i.  
DATA : lv_rep_wa_caps      type ty_report_wo_index,  
      lv_inline_comment   type ty_report_wo_index,  
      lv_form_counter(4)  type n,  
      lv_anchor(4)        type n.
```

```
DATA : lt_form_index type standard table of ty_report_with_index,  
      lt_form_index_wa like line of lt_form_index,  
      lv_form_name(72) type c,  
      lv_string(72)    type c.
```

```
* Add Header to HTML  
add_html '<HTML>'.  
add_html '<head>'.  
add_html '<title>'.  
add_html 'ABAP Code WebViewer by www.rmtiwari.com'.  
add_html ' View :'.  
add_html gv_prog_name.  
add_html '</title>'.  
add_html ' '.  
add_html '<meta name="description"'.  
add_html ' content="'.  
add_html P_DESC.  
add_html '">'.  
add_html '<meta name="keywords"'.  
add_html 'content=" ' .  
add_html 'www.rmtiwari.com,ABAP Code WebViewer,ABAP Code,'.  
add_html P_KEYW .  
add_html '">'.  
  
add_html '<link type="text/css">'.  
add_html '</head>'.  
add_html '<BODY>'.  
add_html '<PRE>'.  
  
LOOP AT xt_rep_table INTO gv_rep_wa.
```

```
* TRANSLATE gv_rep_wa TO UPPER CASE.  
  
* Modify words in ABAP code which are tags for HTML.  
REPLACE '<' with '< ' into gv_rep_wa.  
REPLACE '>' with '> ' into gv_rep_wa.  
  
* For Comments  
IF gv_rep_wa(1) eq gc_asterik.  
  concatenate '<FONT COLOR="BLUE">' gv_rep_wa '</FONT>'  
  INTO yt_html.  
  add_html yt_html.
```

```

*   For Code Lines.
    ELSE.

*       Remove inline comments
        SPLIT gv_rep_wa at '"' into lv_rep_wa_caps lv_inline_comment.
        TRANSLATE lv_rep_wa_caps TO UPPER CASE.
        SHIFT lv_rep_wa_caps LEFT DELETING LEADING space.
*       Replace Form routines and performs with anchors .
        if lv_rep_wa_caps(7) eq 'PERFORM'.

            SPLIT lv_rep_wa_caps+6 at space
              into lv_inline_comment
                  lt_form_index_wa-line lv_inline_comment.
            REPLACE '.' with '' into lt_form_index_wa-line.

            READ TABLE lt_form_index into lt_form_index_wa
              with key line = lt_form_index_wa-line.
            if sy-subrc eq 0.
                lv_anchor = lt_form_index_wa-index.
            else.

*               Populate form index table
                lv_form_counter = lv_form_counter + 1.
                lv_anchor        = lv_form_counter.
                lt_form_index_wa-index = lv_form_counter.

                append lt_form_index_wa to lt_form_index.
            endif.

            CONCATENATE '<A HREF=' '"#F' lv_anchor '>'
              INTO yt_html.

            CONCATENATE yt_html gv_rep_wa into yt_html.
            concatenate lt_form_index_wa-line '</A>' into lv_string.
            replace lt_form_index_wa-line in yt_html
              with lv_string IGNORING CASE .
            add_html yt_html.

        ELSE.
            IF lv_rep_wa_caps(4) eq 'FORM'.

                SPLIT lv_rep_wa_caps+3 at space
                  into lv_inline_comment lv_form_name lv_inline_comment.
                REPLACE '.' in lv_form_name with ''.

                READ TABLE lt_form_index into lt_form_index_wa
                  with key line = lv_form_name.
                if sy-subrc eq 0.
                    lv_anchor = lt_form_index_wa-index.
                    CONCATENATE '<A NAME="F' lv_anchor '> </A>'
                      INTO yt_html.
                    add_html yt_html.
                endif.
            endif.

        add_html gv_rep_wa.

```

```

        endif.
    ENDIF.
ENDLOOP.

* Close HTML Tags
add_html '</PRE>'.
add_html '</BODY>'.
add_html '</HTML>'.

*-----
* Write Anchors for Variables.

DATA : lt_var_list type standard table of RSYMB with header line.
DATA : lv_var_anchor(200) type c,
      lv_var_number(4) type n,
      lv_variable(200) type c.

* Type for Report with line index
TYPES: BEGIN OF ty_var_index,
      index TYPE i,
      line(300),
      first_time,
      END OF ty_var_index.

DATA : lt_var_index type standard table of ty_var_index,
      lt_var_index_wa type ty_var_index,
      lv_first_time type C,
      lv_length type i,
      lv_line_length type i.

lv_first_time = 'X'.
* Get Variables list.
LOAD REPORT gv_prog_name PART 'SYMB' INTO lt_var_list.

LOOP AT yt_html INTO gv_rep_wa.
    LOOP AT lt_var_list.
        SEARCH lt_var_list-TXOF FOR '%'.
        if sy-subrc eq 0 or
            gv_rep_wa(5) eq '<FONT'.
            continue.
        endif.
        READ TABLE lt_var_index into lt_var_index_wa
        with key line = lt_var_list-TXOF.
        if sy-subrc eq 0.
            lv_first_time = lt_var_index_wa-first_time.
            lv_var_number = lt_var_index_wa-index.
        else.
            lv_first_time = 'X'.
            DESCRIBE TABLE lt_var_index lines lv_var_number.
            lt_var_index_wa-index = lv_var_number.
            lt_var_index_wa-line = lt_var_list-TXOF.
            lt_var_index_wa-first_time = 'X'.
            APPEND lt_var_index_wa to lt_var_index.
        endif.
    endloop.
endloop.

```

```

if lv_first_time eq 'X'.

    concatenate '<A NAME="' lv_var_number '"'>' lt_var_list-TXOF
    '</A>'
    into lv_string.

else.
    concatenate '<A HREF="#" lv_var_number '"'>' lt_var_list-TXOF
    '</A>'
    into lv_string.
endif.

concatenate lt_var_list-TXOF ' ' into lv_variable
separated by space.

SEARCH gv_rep_wa FOR lv_variable .
IF sy-subrc eq 0.
    lv_length = sy-fdpos - 1.
    IF lv_length >= 0 .
        CHECK gv_rep_wa+lv_length(1) eq ' ' or
            gv_rep_wa+lv_length(1) eq ':'.
    ENDIF.

    lv_length = strlen( lv_variable ) + sy-fdpos.
    lv_line_length = strlen( gv_rep_wa ).
    IF lv_length le lv_line_length .
        CHECK gv_rep_wa+lv_length(1) eq ' ' or
            gv_rep_wa+lv_length(1) eq '.' or
            gv_rep_wa+lv_length(1) eq '(' or
            gv_rep_wa+lv_length(1) eq '-' or
            gv_rep_wa+lv_length(1) eq ','.
    ENDIF.

CASE lv_variable.
    WHEN 'BLOCK' OR 'BEGIN' OR 'END'.
        SEARCH gv_rep_wa FOR 'OF BLOCK'.
        CHECK sy-subrc ne 0.

    WHEN 'WITH' OR 'FRAME' OR 'TITLE'..
        SEARCH gv_rep_wa FOR 'WITH FRAME TITLE'.
        CHECK sy-subrc ne 0.

    WHEN 'LINE' OR 'BEGIN' OR 'END'.
        SEARCH gv_rep_wa FOR 'OF LINE'.
        CHECK sy-subrc ne 0.
        SEARCH gv_rep_wa FOR 'LIKE LINE OF'.
        CHECK sy-subrc ne 0.

    WHEN 'PARAMETERS'.
        SEARCH gv_rep_wa FOR 'PARAMETERS'.
        CHECK sy-subrc ne 0.

ENDCASE.

REPLACE lt_var_list-TXOF in gv_rep_wa with lv_string

```

```

            IGNORING CASE .
        if sy-subrc eq 0.
            lt_var_index_wa-FIRST_TIME = ' '.
            MODIFY lt_var_index
            from lt_var_index_wa transporting FIRST_TIME
            where line = lt_var_list-TXOF.

            Modify yt_html from gv_rep_wa.

        endif.
    endif.
ENDLOOP.

ENDLOOP.

*-----

ENDFORM.                " convert_code_to_html
*&-----
*
*&      Form  show_html_file
*&-----
*
*      text
*-----
*
*      -->X_FILE_NAME  text
*-----
*
FORM show_html_file  USING      X_FILE_NAME.

data : lv_url(200) type c.

lv_url = x_file_name.
CALL FUNCTION 'CALL_BROWSER'
    EXPORTING
        URL                        = lv_url.

ENDFORM.                " show_html_file

```