

```

1  *&-----*
2  *& Report Z PPTSf FORM1
3  *&-----*
4  *&
5  *&-----*
6  REPORT z_pptsf_form1.
7
8  INCLUDE zdef_form.
9
10 SELECT * FROM sFlight INTO CORRESPONDING FIELDS OF TABLE it_data WHERE connid =
    p_FLIGHT.
11
12 *READ TABLE it_data INTO wa_data INDEX 1.
13
14 IF p_r1 = 'X'.
15     PERFORM show_on_screen .
16
17 ELSEIF p_r2 = 'X'.
18     PERFORM download_pdf.
19
20 elseif p_r3 = 'X'.
21     Perform send_email.
22
23 ENDIF.
24
25
26 FORM show_on_screen.
27
28
29 CALL FUNCTION 'SSF_FUNCTION_MODULE_NAME'
30     EXPORTING
31         formname           = 'ZSF_FORM1'
32     IMPORTING
33         fm_name            = fm_name
34     EXCEPTIONS
35         no_form            = 1
36         no_function_module = 2
37         OTHERS             = 3.
38
39 CALL FUNCTION fm_name
40     EXPORTING
41         control_parameters = lv_control_params
42         output_options     = wa_comp
43         flight_num         = p_flight
44     IMPORTING
45         job_output_info    = lv_output_info
46     TABLES
47         it_data            = it_data
48     EXCEPTIONS
49         formatting_error   = 1
50         internal_error     = 2
51         send_error         = 3
52         user_canceled      = 4
53         OTHERS             = 5.
54
55 * CALL FUNCTION fm_name
56 *     EXPORTING
57 *         flight_num        = p_flight
58 *     TABLES
59 *         it_data          = it_data
60 *     EXCEPTIONS
61 *         formatting error = 1
62 *         internal error   = 2
63 *         send_error       = 3
64 *         user_canceled    = 4
65 *         OTHERS           = 5.
66
67 ENDFORM.
68
69 FORM download_pdf.
70
71
72 CALL FUNCTION 'SSF_FUNCTION_MODULE_NAME'
73     EXPORTING

```

```

74      formname                = 'ZSF_FORM1'
75  *      VARIANT              = ' '
76  *      DIRECT CALL          = ' '
77  IMPORTING
78      fm_name                  = fm_name
79  EXCEPTIONS
80      no_form                  = 1
81      no_function_module      = 2
82      OTHERS                   = 3.
83
84  CALL FUNCTION fm_name
85  EXPORTING
86      control_parameters      = lv_control_params
87      flight_num              = p_flight
88  IMPORTING
89      job_output_info         = lv_output_info
90  TABLES
91      it_data                  = it_data
92  EXCEPTIONS
93      formatting_error         = 1
94      internal_error           = 2
95      send_error               = 3
96      user_canceled           = 4
97      OTHERS                   = 5.
98
99  CALL FUNCTION 'CONVERT_OTF'
100  EXPORTING
101      format                   = 'PDF'
102  *  IMPORTING
103  *      bin_file               = lv_file
104  TABLES
105      otf                      = lv_output_info-otfdata
106      lines                    = lt_lines
107  EXCEPTIONS
108      err_max_linewidth        = 1
109      err_format                = 2
110      err_conv_not_possible    = 3
111      err_bad_otf              = 4
112      OTHERS                   = 5.
113
114
115  CALL FUNCTION 'GUI_DOWNLOAD'
116  EXPORTING
117      filename                  = 'C:\Users\user7\Desktop\smartf.PDF'
118      filetype                  = 'BIN'
119  TABLES
120      data_tab                  = lt_lines
121  EXCEPTIONS
122      file_write_error         = 1
123      no_batch                  = 2
124      gui_refuse_filetransfer  = 3
125      invalid_type              = 4
126      no_authority              = 5
127      unknown_error            = 6
128      header_not_allowed       = 7
129      separator_not_allowed    = 8
130      filesize_not_allowed     = 9
131      header_too_long          = 10
132      dp_error_create           = 11
133      dp_error_send             = 12
134      dp_error_write            = 13
135      unknown_dp_error         = 14
136      access_denied             = 15
137      dp_out_of_memory         = 16
138      disk_full                 = 17
139      dp_timeout                = 18
140      file_not_found           = 19
141      dataprovider_exception    = 20
142      control_flush_error       = 21
143      OTHERS                    = 22.
144
145  ENDFORM.
146
147

```

```

148 FORM send_email.
149
150 CALL FUNCTION 'SSF_FUNCTION_MODULE_NAME'
151 EXPORTING
152   formname           = 'ZSF_FORM1'
153 IMPORTING
154   fm_name            = fm_name
155 EXCEPTIONS
156   no_form             = 1
157   no_function_module = 2
158   OTHERS              = 3.
159
160 CALL FUNCTION fm_name
161 EXPORTING
162   control_parameters = lv_control_params
163   output_options     = wa_comp
164   flight_num         = p_flight
165 IMPORTING
166   job_output_info    = lv_output_info
167 TABLES
168   it_data            = it_data
169 EXCEPTIONS
170   formatting_error   = 1
171   internal_error     = 2
172   send_error         = 3
173   user_canceled      = 4
174   OTHERS             = 5.
175
176 CALL FUNCTION 'CONVERT_OTF'
177 EXPORTING
178   format              = 'PDF'
179 IMPORTING
180   bin_file            = lv_file
181 TABLES
182   otf                 = lv_output_info-otfdata
183   lines               = lt_lines
184 EXCEPTIONS
185   err_max_linewidth   = 1
186   err_format          = 2
187   err_conv_not_possible = 3
188   err_bad_otf         = 4
189   OTHERS              = 5.
190
191
192 CALL FUNCTION 'SCMS_XSTRING_TO_BINARY'
193 EXPORTING
194   buffer              = lv_file
195 *   APPEND TO TABLE    = ' '
196 * IMPORTING
197 *   OUTPUT_LENGTH       =
198 TABLES
199   binary_tab = lt_binary_tab.
200
201 TRY.
202   CALL METHOD cl_bcs=>create_persistent      "Creating send request
203   RECEIVING
204     result = lo_bcs.
205   CATCH cx_send_req_bcs.
206 ENDTRY.
207
208 TRY.
209   CALL METHOD cl_sapuser_bcs=>create
210   EXPORTING
211     i_user = 'AKARMAKAR'
212   RECEIVING
213     result = lo_user.
214   CATCH cx_address_bcs.
215 ENDTRY.
216
217 TRY.
218   CALL METHOD lo_bcs->add_recipient
219   EXPORTING
220     i_recipient = lo_user
221 *   i_express   =

```

```

222 *           i_copy           =
223 *           i_blind_copy     =
224 *           i_no_forward    =
225
226     CATCH cx_send_req_bcs.
227 ENDTRY.
228
229     wa_text-line = TEXT-004.
230 APPEND wa_text TO lt_text.
231 CLEAR wa_text.
232
233     wa_text-line = TEXT-005.
234 APPEND wa_text TO lt_text.
235 CLEAR wa_text.
236
237     wa_text-line = TEXT-006.
238 APPEND wa_text TO lt_text.
239 CLEAR wa_text.
240
241     wa_text-line = TEXT-007.
242 APPEND wa_text TO lt_text.
243 CLEAR wa_text.
244
245 TRY.
246     CALL METHOD cl_document_bcs=>create_document
247     EXPORTING
248         i_type      = 'RAW'
249         i_subject   = TEXT-003
250 *           i_length =
251 *           i_language      = SPACE
252 *           i_importance    =
253 *           i_sensitivity   =
254         i_text      = lt_text
255 *           i_hex          =
256 *           i_header       =
257 *           i_sender       =
258 *           iv_vsi_profile  =
259 *           iv_vsi_scan_off =
260     RECEIVING
261         result      = lo_doc.
262     CATCH cx_document_bcs.
263 ENDTRY.
264
265     CONCATENATE TEXT-008 p_flight INTO lv_subject.
266
267 TRY.
268     CALL METHOD lo_doc->add_attachment
269     EXPORTING
270         i_attachment_type      = 'PDF'
271         i_attachment_subject   = lv_subject
272         i_att_content_hex      = lt_binary_tab.
273     CATCH cx_document_bcs.
274 ENDTRY.
275
276 TRY.
277     CALL METHOD lo_bcs->set_document
278     EXPORTING
279         i_document = lo_doc.
280     CATCH cx_send_req_bcs.
281 ENDTRY.
282
283 TRY.
284     CALL METHOD lo_bcs->set_send_immediately
285     EXPORTING
286         i_send_immediately = 'X'.
287     CATCH cx_send_req_bcs.
288 ENDTRY.
289
290 TRY.
291     CALL METHOD lo_bcs->send
292     RECEIVING
293         result = lv_result.
294     CATCH cx_send_req_bcs.
295 ENDTRY.

```

```
296  
297     IF lv_result IS NOT INITIAL.  
298         COMMIT WORK.  
299     ENDIF.  
300 ENDFORM.
```