\*&---------------------------------------------------------------------\*  
\*& Report ZGRAPH  
\*&---------------------------------------------------------------------\*  
\*&  
\*&---------------------------------------------------------------------\*  
  
report y\_test\_email\_with\_chart.  
  
type-pools: abap .  
  
tables: sflight .  
  
constants: image\_name\_01 type string value 'Sailing.jpg' .  
constants: image\_name\_02 type string value 'chart\_01.jpg' .  
constants: c\_series\_01 type string value 'series\_01' .  
  
\*----------------------------------------------------------------------\*  
types: begin of tp\_suma\_1 .  
         include type sflight as sflight renaming with suffix \_sflight .  
types: end of tp\_suma\_1 .  
  
types: tp\_suma\_1\_tab type table of tp\_suma\_1  .  
\*----------------------------------------------------------------------\*  
\*----------------------------------------------------------------------\*  
class cl\_htm definition .  
  
  public section.  
  
    constants: cr\_lf type string value '<cr\_lf>' .  
  
    data: htm\_string type string .  
  
    methods: append  
      importing fragment type any .  
  
  private section.  
  
    methods: edit\_value  
      importing fragment type any  
      changing  buffer   type string .  
  
endclass.                    "cl\_htm DEFINITION  
\*----------------------------------------------------------------------\*  
\*----------------------------------------------------------------------\*  
class cl\_htm implementation.  
\*----------------------------------------------------------------------\*  
  method: append .  
  
    data: buffer type string .  
  
    me->edit\_value( exporting fragment = fragment changing buffer = buffer ) .  
  
    condense buffer .  
  
    concatenate htm\_string buffer into htm\_string .  
  
  endmethod.                    "append  
\*----------------------------------------------------------------------\*  
  method: edit\_value .  
  
    data: inttype type inttype .  
  
    describe field fragment type inttype .  
  
    data: edited type so\_text255 .  
  
    write fragment to edited left-justified .  
  
    if buffer eq cl\_htm=>cr\_lf .  
      return .  
    endif .  
  
    case inttype .  
      when 'I' or 'P' .  
        if fragment lt 0 .  
          write fragment to edited+1 no-sign left-justified .  
          write '-' to edited+0(1) .  
          buffer = edited .  
        endif .  
      when others .  
        buffer = fragment .  
    endcase .  
  
  endmethod.                    "edit\_minus\_value  
\*----------------------------------------------------------------------\*  
endclass.                    "cl\_htm IMPLEMENTATION  
\*----------------------------------------------------------------------\*  
\*----------------------------------------------------------------------\*  
class cl\_str definition .  
  
  public section .  
  
    class-methods: to\_string  
      importing input         type any  
      returning value(output) type string .  
  
endclass .                    "cl\_string DEFINITION  
\*----------------------------------------------------------------------\*  
\*----------------------------------------------------------------------\*  
class cl\_str implementation.  
  
  method: to\_string .  
  
    data: buffer type char64 .  
    data: string type string .  
  
    write input to buffer left-justified.  
  
    replace all occurrences of ',' in buffer with '' .  
  
    output = buffer .  
  
  endmethod .                    "to\_string  
  
endclass .                    "cl\_str IMPLEMENTATION  
\*----------------------------------------------------------------------\*  
\*----------------------------------------------------------------------\*  
selection-screen begin of block block02 with frame .  
  
  parameters: p\_fname  type ssfscreen-fname default 'SF\_EXAMPLE\_01' .  
  
  selection-screen skip .  
  
  parameters: p\_sndr\_1 type adsmtp-smtp\_addr .  
  
  selection-screen skip .  
  
  parameters: p\_rcpn\_1 type adsmtp-smtp\_addr .  
  
  selection-screen skip .  
  
  parameters: p\_sbjct type so\_obj\_des .  
  
  selection-screen skip .  
  
  select-options:  s\_carrid   for sflight-carrid .  
  select-options:  s\_connid   for sflight-connid .  
  select-options:  s\_fldate   for sflight-fldate .  
  
  selection-screen skip .  
  
  parameters: p\_debug  type debug\_flg .  
selection-screen end of block block02 .  
\*----------------------------------------------------------------------\*  
  
initialization .  
  perform at\_initialization .  
  
at selection-screen output .  
  perform at\_selection\_screen\_output .  
  
at selection-screen .  
  perform at\_selection\_screen\_input .  
  
start-of-selection .  
  perform at\_start\_of\_selection .  
  
\*----------------------------------------------------------------------\*  
form at\_initialization .  
  
  data: st\_address type bapiaddr3 .  
  
  data: it\_return type bapirettab .  
  
  call function 'BAPI\_USER\_GET\_DETAIL'  
    exporting  
      username = sy-uname  
    importing  
      address  = st\_address  
    tables  
      return   = it\_return.  
  
  p\_sndr\_1 = st\_address-e\_mail .  
  p\_rcpn\_1 = st\_address-e\_mail .  
  
  concatenate 'Hi there-' sy-datum '-' sy-uzeit '(EX)' into p\_sbjct .  
  
  data: it\_seltexts type table of rsseltexts .  
  data: st\_seltexts like line of it\_seltexts .  
  
  st\_seltexts-name = 'P\_SNDR\_1' .  
  st\_seltexts-kind = 'P' .  
  st\_seltexts-text = 'Sender mail '  .  
  append st\_seltexts to it\_seltexts .  
  
  st\_seltexts-name = 'P\_RCPN\_1' .  
  st\_seltexts-kind = 'P' .  
  st\_seltexts-text = 'Recipient mail 1 '  .  
  append st\_seltexts to it\_seltexts .  
  
  call function 'SELECTION\_TEXTS\_MODIFY'  
    exporting  
      program                     = sy-cprog  
    tables  
      seltexts                    = it\_seltexts  
    exceptions  
      program\_not\_found           = 1  
      program\_cannot\_be\_generated = 2  
      others                      = 3.  
  
endform .                    "at\_initialization  
\*----------------------------------------------------------------------\*  
form at\_selection\_screen\_output .  
endform .                    "at\_selection\_screen\_output  
\*----------------------------------------------------------------------\*  
form at\_selection\_screen\_input .  
endform .                    "at\_selection\_screen\_input  
\*----------------------------------------------------------------------\*  
form at\_start\_of\_selection .  
  
  perform mail\_1\_main .  
  
endform .                    "at\_start\_of\_selection  
\*----------------------------------------------------------------------\*  
form mail\_1\_main .  
  
  data: ob\_gbt\_multirelated\_service type ref to cl\_gbt\_multirelated\_service.  
  
  create object ob\_gbt\_multirelated\_service.  
  
  "setup the html document for body of email - references the multi-part  
  "attachments setup below to display inline.  
  perform mail\_1\_prep\_1  
    changing  
      ob\_gbt\_multirelated\_service .  
  
  "get pic and add as binary multi-part attachment to email (will show inline)  
  perform mail\_1\_prep\_2  
    changing  
      ob\_gbt\_multirelated\_service .  
  
  "generate chart and add as binary multi-part attachment to email (will show inline)  
  perform mail\_1\_prep\_3  
    changing  
      ob\_gbt\_multirelated\_service .  
  
  
  
  "create a multi-part email with BCS  
  data: ob\_document\_bcs type ref to cl\_document\_bcs.  
  
  ob\_document\_bcs = cl\_document\_bcs=>create\_from\_multirelated(  
    i\_subject = p\_sbjct  
    i\_multirel\_service = ob\_gbt\_multirelated\_service ).  
  
  "add chart as email attachment as well  
  perform mail\_1\_prep\_4  
    changing  
      ob\_document\_bcs .  
  
  "call a smartform, convert to pdf and add as attachment  
  perform mail\_1\_prep\_5  
    changing  
      ob\_document\_bcs .  
  
  
  "standard BCS sending stuff...  
  data: ob\_bcs type ref to cl\_bcs.  
  data: ob\_sapuser\_bcs type ref to cl\_sapuser\_bcs.  
  data: ob\_cam\_address\_bcs type ref to cl\_cam\_address\_bcs .  
  
  ob\_bcs = cl\_bcs=>create\_persistent( ).  
  
  call method ob\_bcs->set\_document( ob\_document\_bcs ).  
  
  ob\_cam\_address\_bcs = cl\_cam\_address\_bcs=>create\_internet\_address( p\_sndr\_1 ).  
  
  call method ob\_bcs->set\_sender  
    exporting  
      i\_sender = ob\_cam\_address\_bcs.  
  
  data: smtp\_addr type adsmtp-smtp\_addr .  
  
  ob\_cam\_address\_bcs = cl\_cam\_address\_bcs=>create\_internet\_address( p\_rcpn\_1 ).  
  
  call method ob\_bcs->add\_recipient  
    exporting  
      i\_recipient = ob\_cam\_address\_bcs.  
  
  data: result type abap\_bool .  
  
  call method ob\_bcs->send  
    exporting  
      i\_with\_error\_screen = abap\_true  
    receiving  
      result              = result.  
  
  commit work .  
  
endform .                    "mail\_1\_main  
\*----------------------------------------------------------------------\*  
form mail\_1\_prep\_1  
  changing  
    ob\_gbt\_multirelated\_service type ref to cl\_gbt\_multirelated\_service .  
  
  data: content type xstring.  
  
  "get the pic from the mime repository - not sure why this is here - its done later!  
  perform mail\_1\_image\_1  
    changing  
      content .  
  
  data: ob\_htm type ref to cl\_htm .  
  
  create object ob\_htm.  
  
  call method ob\_htm->append  
  exporting  
    fragment = :  
      '<P>'  ,  
      'Hi there' ,  
      '</P>' .  
  
\* Works in outlook.  
  
  call method ob\_htm->append  
    exporting  
      fragment = :  
        '<img src="cid:' , image\_name\_01 , '" alt="' , image\_name\_01 , '" align="middle" />'.  
  
  call method ob\_htm->append  
    exporting  
      fragment = '<br><br>'.  
  
  call method ob\_htm->append  
    exporting  
      fragment = :  
        '<img src="cid:' , image\_name\_02 , '" alt="' , image\_name\_02 , '" align="middle" />'.  
  
  call method ob\_htm->append  
    exporting  
      fragment = '<br><br>'.  
  
\* Does not work in outlook  
  
\*  DATA: encoded TYPE string .  
\*  
\*  CALL METHOD cl\_http\_utility=>if\_http\_utility~encode\_x\_base64  
\*    EXPORTING  
\*      unencoded = content  
\*    RECEIVING  
\*      encoded   = encoded.  
  
\*  CALL METHOD ob\_htm->append  
\*    EXPORTING  
\*      fragment = :  
\*        '<img src="data:image/gif;base64,' , encoded  ,  
\*        ' "alt="Happy birthday" align="middle" width="304" height="228" />'.  
  
  data: it\_soli type soli\_tab .  
  
  call method cl\_bcs\_convert=>string\_to\_soli  
    exporting  
      iv\_string = ob\_htm->htm\_string  
    receiving  
      et\_soli   = it\_soli.  
  
  call method ob\_gbt\_multirelated\_service->set\_main\_html  
    exporting  
      content     = it\_soli  
      filename    = 'main\_html.htm'  
      description = 'main\_html.htm'.  "Title  
  
\*  CALL METHOD cl\_document\_bcs=>create\_document  
\*    EXPORTING  
\*      i\_type    = 'HTM'  
\*      i\_subject = p\_sbjct  
\*      i\_text    = it\_soli  
\*\*     i\_hex     = it\_solix  
\*    RECEIVING  
\*      result    = ob\_document\_bcs.  
  
endform .                    "mail\_1\_prep\_1  
\*----------------------------------------------------------------------\*  
form mail\_1\_prep\_2  
  changing  
    ob\_gbt\_multirelated\_service  type ref to cl\_gbt\_multirelated\_service .  
  
\* From http://scn.sap.com/docs/DOC-42618  
  
  data: content type xstring.  
  
  "get the pic fropm the mime repository  
  perform mail\_1\_image\_1  
    changing  
      content .  
  
  data: it\_solix type solix\_tab .  
  
  call method cl\_bcs\_convert=>xstring\_to\_solix  
    exporting  
      iv\_xstring = content  
    receiving  
      et\_solix   = it\_solix.  
  
  data: description type so\_obj\_des .  
  data: obj\_len type so\_obj\_len .  
  
  obj\_len = xstrlen( content ).  
  
  call method ob\_gbt\_multirelated\_service->add\_binary\_part  
    exporting  
      content      = it\_solix  
      filename     = image\_name\_01  
      extension    = 'JPG'  
      description  = description  
      content\_type = 'image/jpg'  
      length       = obj\_len  
      content\_id   = image\_name\_01.  
  
endform .                    "mail\_1\_prep\_2  
\*----------------------------------------------------------------------\*  
form mail\_1\_prep\_3  
  changing  
    ob\_gbt\_multirelated\_service  type ref to cl\_gbt\_multirelated\_service .  
  
  data: xstring type xstring.  
  
  perform do\_igs\_chart  
    changing  
      xstring .  
  
  data: it\_solix type solix\_tab .  
  
  call method cl\_bcs\_convert=>xstring\_to\_solix  
    exporting  
      iv\_xstring = xstring  
    receiving  
      et\_solix   = it\_solix.  
  
  data: description type so\_obj\_des .  
  data: obj\_len type so\_obj\_len .  
  
  obj\_len = xstrlen( xstring ).  
  
  call method ob\_gbt\_multirelated\_service->add\_binary\_part  
    exporting  
      content      = it\_solix  
      filename     = image\_name\_02  
      extension    = 'GIF'  
      description  = description  
      content\_type = 'image/gif'  
      length       = obj\_len  
      content\_id   = image\_name\_02.  
  
endform .                    "mail\_1\_prep\_3  
\*----------------------------------------------------------------------\*  
form mail\_1\_prep\_4  
  changing  
    ob\_document\_bcs type ref to cl\_document\_bcs .  
  
  data: xstring type xstring.  
  
  perform do\_igs\_chart  
    changing  
      xstring .  
  
  data: it\_solix type solix\_tab .  
  
  call method cl\_bcs\_convert=>xstring\_to\_solix  
    exporting  
      iv\_xstring = xstring  
    receiving  
      et\_solix   = it\_solix.  
  
  data: attachment\_subject type so\_obj\_des .  
  
  message s000(oo) with 'attachment chart' space into attachment\_subject .  
  
  try.  
      call method ob\_document\_bcs->add\_attachment  
        exporting  
          i\_attachment\_type    = 'gif'  
          i\_attachment\_subject = attachment\_subject  
          i\_att\_content\_hex    = it\_solix.  
    catch cx\_document\_bcs .  
  endtry.  
  
endform .                    "mail\_1\_prep\_4  
\*----------------------------------------------------------------------\*  
form mail\_1\_prep\_5  
  changing  
    ob\_document\_bcs type ref to cl\_document\_bcs .  
  
  data: customer type scustom .  
  data: bookings type ty\_bookings .  
  data: connections type ty\_connections   .  
  
  data: formname type tdsfname .  
  data: fm\_name type rs38l\_fnam .  
  
  formname = p\_fname .  
  
  call function 'SSF\_FUNCTION\_MODULE\_NAME'  
    exporting  
      formname           = p\_fname  
    importing  
      fm\_name            = fm\_name  
    exceptions  
      no\_form            = 1  
      no\_function\_module = 2  
      others             = 3.  
  
  if sy-subrc ne 0 .  
    return .  
  endif .  
  
  data: st\_control\_parameters type ssfctrlop .  
  data: st\_output\_options type ssfcompop .  
  data: st\_job\_output\_info type ssfcrescl .  
  
  st\_control\_parameters-no\_dialog = abap\_true .  
  st\_control\_parameters-getotf    = abap\_true .  
  
  call function fm\_name  
    exporting  
      control\_parameters = st\_control\_parameters  
      output\_options     = st\_output\_options  
      customer           = customer  
      bookings           = bookings  
      connections        = connections  
    importing  
      job\_output\_info    = st\_job\_output\_info  
    exceptions  
      formatting\_error   = 1  
      internal\_error     = 2  
      send\_error         = 3  
      user\_canceled      = 4  
      others             = 5.  
  
  if sy-subrc eq 4 .  
    return .  
  endif .  
  
  if sy-subrc ne 0.  
    message id sy-msgid type sy-msgty number sy-msgno  
            with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.  
  endif.  
  
  data: bin\_file type xstring .  
  data: it\_lines\_dummy type tline\_tab .  
  data: bin\_filesize type i .  
  
  call function 'CONVERT\_OTF'  
    exporting  
      format                = 'PDF'  
    importing  
      bin\_file              = bin\_file  
      bin\_filesize          = bin\_filesize  
    tables  
      otf                   = st\_job\_output\_info-otfdata  
      lines                 = it\_lines\_dummy  
    exceptions  
      err\_max\_linewidth     = 1  
      err\_format            = 2  
      err\_conv\_not\_possible = 3  
      err\_bad\_otf           = 4  
      others                = 5.  
  
  data: it\_solix type solix\_tab .  
  
  call method cl\_bcs\_convert=>xstring\_to\_solix  
    exporting  
      iv\_xstring = bin\_file  
    receiving  
      et\_solix   = it\_solix.  
  
  data: attachment\_subject type so\_obj\_des .  
  
  message s000(oo) with 'attachment PDF hex' space into attachment\_subject .  
  
  try.  
      call method ob\_document\_bcs->add\_attachment  
        exporting  
          i\_attachment\_type    = 'pdf'  
          i\_attachment\_subject = attachment\_subject  
          i\_att\_content\_hex    = it\_solix.  
    catch cx\_document\_bcs .  
  endtry.  
  
endform .                    "mail\_1\_prep\_5  
\*----------------------------------------------------------------------\*  
  
\*----------------------------------------------------------------------\*  
form mail\_1\_image\_1  
  changing  
    content type xstring.  
  
  data: ob\_mr\_api type ref to if\_mr\_api .  
  
  ob\_mr\_api = cl\_mime\_repository\_api=>if\_mr\_api~get\_api( ).  
  
  data: i\_url type string value '/SAP/PUBLIC/AES/SAP Sailing.jpg' .  
  
  call method ob\_mr\_api->get  
    exporting  
      i\_url              = i\_url  
    importing  
\*     e\_is\_folder        = is\_folder  
      e\_content          = content  
\*     e\_loio             = l\_loio  
    exceptions  
      parameter\_missing  = 1  
      error\_occured      = 2  
      not\_found          = 3  
      permission\_failure = 4  
      others             = 5.  
  
endform .                    "mail\_1\_image\_1  
\*----------------------------------------------------------------------\*  
form get\_suma\_1  
  changing  
    it\_suma\_1 type tp\_suma\_1\_tab .  
  
  clear: it\_suma\_1[] .  
  
  data: it\_sflight type flighttab .  
  
  select \* into table it\_sflight  
  from sflight  
  where  
    carrid in s\_carrid and  
    connid in s\_connid and  
    fldate in s\_fldate .  
  
  field-symbols: <st\_sflight> like line of it\_sflight .  
  
  
  
  data: st\_suma\_1 like line of it\_suma\_1 .  
  
  loop at it\_sflight assigning <st\_sflight> .  
  
    clear st\_suma\_1 .  
  
    st\_suma\_1-sflight = <st\_sflight> .  
  
    clear: st\_suma\_1-sflight-fldate,  
           st\_suma\_1-sflight-connid,  
           st\_suma\_1-sflight-planetype ,  
           st\_suma\_1-sflight-currency .  
  
    collect st\_suma\_1 into it\_suma\_1 .  
  
  endloop .  
  
endform .                                                   "get\_suma\_1  
\*----------------------------------------------------------------------\*  
form do\_igs\_chart  
  changing  
    xstring type xstring.  
  
  data: ob\_igs\_chart\_engine type ref to cl\_igs\_chart\_engine .  
  data: ob\_ixml type ref to if\_ixml .  
  
  create object ob\_igs\_chart\_engine  
    exporting  
      destination = 'IGS\_RFC\_DEST'.  
  
  ob\_ixml = cl\_ixml=>create( ).  
  
  perform do\_igs\_chart\_1  
    changing  
      ob\_igs\_chart\_engine  
      ob\_ixml .  
  
  call method ob\_igs\_chart\_engine->execute.  
  
  data: it\_image\_mime type w3mimetabtype .  
  data: image\_size type w3param-cont\_len,  
        image\_type type w3param-cont\_type.  
  
  call method ob\_igs\_chart\_engine->get\_image  
    importing  
      image      = it\_image\_mime  
      image\_size = image\_size  
      image\_type = image\_type.  
  
  field-symbols: <st\_image\_mime> like line of it\_image\_mime .  
  
  loop at it\_image\_mime assigning <st\_image\_mime>.  
    concatenate xstring <st\_image\_mime>-line into xstring in byte mode.  
  endloop.  
  
endform .                    "do\_IGS\_chart  
\*----------------------------------------------------------------------\*  
form do\_igs\_chart\_1  
  changing  
    ob\_igs\_chart\_engine type ref to cl\_igs\_chart\_engine  
    ob\_ixml type ref to if\_ixml .  
  
  data: it\_suma\_1 type tp\_suma\_1\_tab .  
  
  perform get\_suma\_1  
    changing  
      it\_suma\_1 .  
  
  perform do\_igs\_chart\_1\_data\_1  
    using  
      it\_suma\_1  
    changing  
      ob\_igs\_chart\_engine  
      ob\_ixml .  
  
  perform do\_igs\_chart\_1\_cust\_1  
    changing  
      ob\_igs\_chart\_engine  
      ob\_ixml .  
  
endform .                    "do\_gui\_chart\_1  
\*----------------------------------------------------------------------\*  
form do\_igs\_chart\_1\_data\_1  
   using  
     it\_suma\_1 type tp\_suma\_1\_tab  
   changing  
     ob\_igs\_chart\_engine type ref to cl\_igs\_chart\_engine  
     ob\_ixml             type ref to if\_ixml .  
  
  data: ob\_ixml\_document type ref to if\_ixml\_document,  
        ob\_ixml\_encoding type ref to if\_ixml\_encoding.  
  
  data: ob\_simplechartdata type ref to if\_ixml\_element,  
        ob\_categories      type ref to if\_ixml\_element,  
        ob\_category        type ref to if\_ixml\_element,  
        ob\_series          type ref to if\_ixml\_element,  
        ob\_point           type ref to if\_ixml\_element,  
        ob\_value           type ref to if\_ixml\_element,  
        l\_string           type        string.  
  
  ob\_ixml\_document = ob\_ixml->create\_document( ).  
  
  ob\_ixml\_encoding = ob\_ixml->create\_encoding(  
        byte\_order    = if\_ixml\_encoding=>co\_little\_endian  
        character\_set = 'utf-8' ) .  
  
  ob\_ixml\_document->set\_encoding( ob\_ixml\_encoding ).  
  
  ob\_simplechartdata = ob\_ixml\_document->create\_simple\_element(  
            name = 'ChartData' parent = ob\_ixml\_document ).  
  
  ob\_categories = ob\_ixml\_document->create\_simple\_element(  
            name = 'Categories' parent = ob\_simplechartdata ) .  
  
  field-symbols: <st\_suma\_1> like line of it\_suma\_1 .  
  
  ob\_series = ob\_ixml\_document->create\_simple\_element(  
           name = 'Series' parent = ob\_simplechartdata ).  
  ob\_series->set\_attribute( name = 'label' value = 'Occupied seats in economy class' ).  
  ob\_series->set\_attribute( name = 'customizing' value = c\_series\_01 ).  
  
  data: a\_string type string .  
  
  loop at it\_suma\_1 assigning <st\_suma\_1> .  
  
    ob\_category = ob\_ixml\_document->create\_simple\_element(  
              name = 'Category' parent = ob\_categories ).  
  
    a\_string = cl\_str=>to\_string( <st\_suma\_1>-sflight-carrid ) .  
    ob\_category->if\_ixml\_node~set\_value( a\_string ) .  
  
    ob\_point = ob\_ixml\_document->create\_simple\_element(  
              name = 'Point' parent = ob\_series ).  
  
    a\_string = cl\_str=>to\_string( <st\_suma\_1>-sflight-seatsocc ) .  
  
    concatenate 'Seats = ' a\_string into a\_string respecting blanks .  
  
    ob\_point->set\_attribute( name = 'label' value = a\_string ) .  
  
    ob\_value = ob\_ixml\_document->create\_simple\_element(  
              name = 'Value' parent = ob\_point ).  
  
    ob\_value->set\_attribute( name = 'Type' value = 'y' ) .  
  
    a\_string = cl\_str=>to\_string( <st\_suma\_1>-sflight-seatsocc ) .  
    ob\_value->if\_ixml\_node~set\_value( a\_string ).  
  
  endloop .  
  
  ob\_igs\_chart\_engine->set\_data( data\_doc = ob\_ixml\_document ) .  
  
endform .                    "do\_chart\_1\_data\_2  
\*----------------------------------------------------------------------\*  
form do\_igs\_chart\_1\_cust\_1  
  changing  
    ob\_igs\_chart\_engine type ref to cl\_igs\_chart\_engine  
    ob\_ixml             type ref to if\_ixml .  
  
  data: ob\_ixml\_document type ref to if\_ixml\_document,  
        ob\_ixml\_encoding type ref to if\_ixml\_encoding.  
  
  data: ob\_sapchartcustomizing type ref to if\_ixml\_element,  
        ob\_globalsettings      type ref to if\_ixml\_element,  
        ob\_defaults            type ref to if\_ixml\_element,  
        ob\_charttype           type ref to if\_ixml\_element,  
        ob\_elements            type ref to if\_ixml\_element,  
        ob\_charttypes          type ref to if\_ixml\_element,  
        ob\_columns             type ref to if\_ixml\_element,  
        ob\_values              type ref to if\_ixml\_element,  
        ob\_series              type ref to if\_ixml\_element,  
        ob\_element             type ref to if\_ixml\_element . "Reused as leaf  
  
  ob\_ixml\_document = ob\_ixml->create\_document( ).  
  
  ob\_ixml\_encoding = ob\_ixml->create\_encoding(  
        byte\_order    = if\_ixml\_encoding=>co\_little\_endian  
        character\_set = 'utf-8' ) .  
  
  ob\_ixml\_document->set\_encoding( ob\_ixml\_encoding ) .  
  
  ob\_sapchartcustomizing = ob\_ixml\_document->create\_simple\_element(  
           name = 'SAPChartCustomizing' parent = ob\_ixml\_document ) .  
  ob\_sapchartcustomizing->set\_attribute( name = 'version' value = '2.0' ) .  
  
  ob\_globalsettings = ob\_ixml\_document->create\_simple\_element(  
           name = 'GlobalSettings' parent = ob\_sapchartcustomizing ) .  
  
  ob\_element = ob\_ixml\_document->create\_simple\_element(  
          name = 'Dimension' parent = ob\_globalsettings ) .  
"  ob\_element->if\_ixml\_node~set\_value( 'Three' ) .  
 ob\_element->if\_ixml\_node~set\_value( 'Two' ) .  
  
  ob\_element = ob\_ixml\_document->create\_simple\_element(  
          name = 'ColorPalette' parent = ob\_globalsettings ) .  
  ob\_element->if\_ixml\_node~set\_value( 'Streamline' ) .  
  
  ob\_defaults = ob\_ixml\_document->create\_simple\_element(  
           name = 'Defaults' parent = ob\_globalsettings ) .  
  
  ob\_charttype = ob\_ixml\_document->create\_simple\_element(  
           name = 'ChartType' parent = ob\_defaults ) .  
" ob\_charttype->if\_ixml\_node~set\_value( 'Doughnut' ).  
 ob\_charttype->if\_ixml\_node~set\_value( 'Pie' ).  
\* ob\_charttype->if\_ixml\_node~set\_value( 'Lines' ).  
"  ob\_charttype->if\_ixml\_node~set\_value( 'Columns' ) .  
  
  ob\_elements = ob\_ixml\_document->create\_simple\_element(  
           name = 'Elements' parent = ob\_sapchartcustomizing ) .  
  
  ob\_charttypes = ob\_ixml\_document->create\_simple\_element(  
         name = 'ChartTypes' parent = ob\_elements ) .  
  
  ob\_columns = ob\_ixml\_document->create\_simple\_element(  
         name = 'Columns' parent = ob\_charttypes ) .  
  
  ob\_element = ob\_ixml\_document->create\_simple\_element(  
          name = 'VaryColorsByPoint' parent = ob\_columns ) .  
  ob\_element->if\_ixml\_node~set\_value( 'true' ) .  
  
  """"  
  ob\_values = ob\_ixml\_document->create\_simple\_element(  
           name = 'Values' parent = ob\_sapchartcustomizing ) .  
  
  ob\_series = ob\_ixml\_document->create\_simple\_element(  
          name = 'Series' parent = ob\_values ) .  
  
  ob\_series->set\_attribute( name = 'id' value = c\_series\_01 ) .  
  
  ob\_element = ob\_ixml\_document->create\_simple\_element(  
         name = 'ShowLabel' parent = ob\_series ) .  
  ob\_element->if\_ixml\_node~set\_value( 'true' ) .  
  
  ob\_igs\_chart\_engine->set\_customizing( custom\_doc = ob\_ixml\_document ) .  
  
endform .                    "do\_chart\_1\_cust\_1