```
        ctl-opt  dftactgrp(*NO) bnddir('GETITEMSVC');

         dcl-f iteminqrfm workstn sfile(sfl:rrn)  indds(dspindicators);


      /copy /WebServices/itemws/RTVITEMSVCPortType.rpgleinc

          dcl-ds  WebServiceDS  likeds(This_t);
          dcl-ds  ItemClassDS   likeds(GetItemsInput_t);

          dcl-ds ItemListDS    likeds(GetItemsResult_t);


          dcl-ds dspindicators;
           requested             ind pos(1);
           endofjob              ind pos(3);
        // optional dcl-subf is required when a field name is the same as an op code
           dcl-subf return       ind pos(12);
           subfileclear          ind pos(60);
           subfiledsp            ind pos(62);
           itemclassblank        ind pos(71);
           itemclassnotfound     ind pos(72);
           webservicedown        ind pos(73);
           lastsubfilerec        ind pos(77);
          end-ds;

          dcl-s   rrn  zoned(4:0);



       //  Prompt user for item class
          dou endofjob;

          exfmt screen1;

       //  Was EOJ requested?
           if endofjob;
            *inlr = *on;
            return;
           endif;

       //  Clear error conditons
           itemclassblank = *off;
           itemclassnotfound = *off;
           webservicedown = *off;

       //  Item class cannot be equal to all blanks
          if $itmcls = *blanks;
           itemclassblank = *on;
           iter;
           endif;
```

```
//  Prepare to call soap service
  clear WebServiceDs;
  clear ItemClassDs;
  clear ItemListDs;

  ItemClassDs.ItemClassIn.value = $itmcls;

//  Allocate space in storage
     if stub_create_RTVITEMSVCPortType(WebServiceDs);

 // Invoke the Web service operation.
if stub_op_getitems(WebServiceDs:ItemClassDs:ItemListDs);
   endif;

 // Destroy Web service stubs.
  stub_destroy_RTVITEMSVCPortType(WebServiceDs);
    endif;
 // Check if an error occurred during call to service
   if webserviceds.excoccurred;
    webservicedown = *on;
     iter;
   endif;

//  See if we have success (value of '1' will be returned)
    if ItemListDs.success.value = 'False';
    itemclassnotfound = *on;
   iter;
 endif;

//  If we made it this far, good data has been received

//  Load header data first
     $clsdsc = itemlistds.classdesc.value;
     $nbritms = itemlistds.numberofitems.value;

//  Prepare subfile control record
  rrn = *zeros;
  subfileclear = *on;
  subfiledsp = *off;
  lastsubfilerec = *off;

  write sflctl;

  subfileclear = *off;
  subfiledsp = *on;

//  Capture detail data, load subfile
 for rrn = 1 to itemlistds.numberofitems.value;

// Item Number
    $item = itemlistds.itemlist.array(rrn).itemnumber.value;
```

```
// Description
    $desc = itemlistds.itemlist.array(rrn).description.value;
//  _ref.array(rrn).temperatures.daytimehigh.value;

// Unit of Measure
    $uom = itemlistds.itemlist.array(rrn).unitofmeasure.value;

// End of subfile indicator
   if rrn = itemlistds.numberofitems.value;
    lastsubfilerec = *on;
   endif;

 write sfl;
 endfor;

//  Display subfile.
   dou return;
    exfmt sflctl;
   enddo;

     enddo;
```