

```
ctl-opt option(*nodebugio) dftactgrp(*no);
```

```
dcl-f weathrrsfm workstn(*ext) indds(dspindicators)
      sfile(sfl:rrn);
```

```
dcl-s   rrn           packed(4:0);
dcl-s   errtext       varchar(50);
dcl-s   ErrorCode     char(1);
dcl-s   command       char(256);
dcl-s   length        packed(15:5);
```

```
dcl-ds dspindicators;
  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);
  zipcodezero        ind pos(71);
  zipcodenotfound    ind pos(72);
  webservicedown     ind pos(73);
end-ds;
```

```
dcl-ds xmlindata      qualified dim(7);
  state              varchar(2);
  city               varchar(20);
  station            varchar(30);
  date               varchar(15);
  description         varchar(30);
  morninglow         varchar(3);
  daytimehigh        varchar(3);
  nighttime          varchar(3);
  daytime            varchar(3);
end-ds;
```

```
dcl-pr qcmdexc         extpgm('QCMDEXC');
  command            char(256);
  length             packed(15:5);
end-pr;
```

```
// Set CCSID to 037 for this job
// This is a requirement in order to properly consume
// the web service using SQL
```

```
command = 'CHGJOB CCSID(037)';
length = %len(%trim(command));
```

```
qcmdexc(command:length);
```

```
// Prompt user for zip code
dou endofjob;

exfmt screen1;

// Was E0J requested?
if endofjob;
    *inlr = *on;
    return;
endif;

// Clear error conditons
zipcodezero = *off;
zipcodenotfound = *off;
webservicedown = *off;

// Zipcode cannot be equal to all zeros
if zipcode = *zeros;
    zipcodezero = *on;
    iter;
endif;

// Call subprocedure to consume the CDYNE web service
    ConsumeWebService(zipcode:ErrorCode);

// If an error occurred display error message
    if ErrorCode = 'E';
        iter;
    endif;

// When here load subfile and display results

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

    write sflctl;

    subfileclear = *off;
    subfiledsp = *on;

// Load header fields
state = xmlindata(1).state;
city = xmlindata(1).city;
wthrsta = xmlindata(1).station;

// Capture detail data, load subfile
```

```

for rrn = 1 to 7;
  wthrdate = xmlindata(rrn).date;
  desc = xmlindata(rrn).description;
  templo = xmlindata(rrn).morninglow;
  temphi = xmlindata(rrn).daytimehigh;
  precpnt = xmlindata(rrn).nighttime;
  precpmo = xmlindata(rrn).daytime;

write sfl;
endfor;

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

enddo;

// -----
// Procedure name: ConsumeWebService
// Purpose:       To consume the Cdyne weather web service
// Parameters:    zipcode, errorcode
// *-----

dcl-proc ConsumeWebService;
  dcl-pi *n;
    zipcodein      packed(5:0) const;
    errorcodeout   char(1);
  end-pi;

  clear xmlindata;
  clear errorcodeout;

Exec SQL Declare CsrC01 Cursor For
  Select *
  from XMLTABLE(xmlnamespaces(default
    'http://ws.cdyne.com/WeatherWS/ '),
    'ForecastReturn/ForecastResult/Forecast'
    passing xmlparse(document(
      systools.httpGetClob(
        'http://wsf.cdyne.com/WeatherWS/Weather.asmx/GetCityForecastByZIP+
        Zip='concat :zipcode, '')))
Columns
State          VarChar(3)      path '../..//State',
City           VarChar(30)     path '../..//City',
Station        VarChar(30)     path '../..//WeatherStationCity',
DateTime       VarChar(50)     path 'Date',
Description    VarChar(30)     path 'Description',
MorningLow     VarChar(3)      path 'Temperatures/MorningLow',
DayTimeHigh    VarChar(3)      path 'Temperatures/DaytimeHigh',
Nighttime      VarChar(3)

```

```

Daytime      path 'ProbabilityOfPrecipiation/Nighttime',
              VarChar(3)
              path 'ProbabilityOfPrecipiation/Daytime )x;

```

```

Exec SQL Close CsrC01;
Exec SQL Open  CsrC01;

```

```
exec sql fetch CsrC01 for 7 rows into :xmlindata;
```

```
exec SQL Close CsrC01;
```

```

//  SQLCODE of < 0 means a web service error occurred
//  SQLCODE of 0 means a successful call to the web service
//  SQLCODE of 100 means no rows returned

```

```

If SQLCODE = 100;
  errorcodeout = 'E';
  zipcodenotfound = *on;
endif;

```

```

If SQLCODE < *Zeros;
  errorcodeout = 'E';
  webservicedown = *on;
endif;

```

```

// If an error occurred close the cursor
if errorcodeout = 'E';
  Exec SQL Close CsrC01;
endif;

```

```
return;
```

```
end-proc ConsumWebService;
```