

```
ctl-opt nomain;
```

```
dcl-pr UnitsToCases;  
  item          char(20)  const;  
  unitsin       zoned(9:2) const;  
  casesout      zoned(9:2);  
end-pr;
```

```
dcl-pr CasesToUnits;  
  item          char(20)  const;  
  casesin       zoned(9:2) const;  
  unitsout      zoned(9:2);  
end-pr;
```

```
// *****  
*****  
// This procedure converts units to case quantity  
// *****  
*****
```

```
dcl-proc UnitsToCases export;
```

```
dcl-pi *n;  
  item          char(20)  const;  
  units         zoned(9:2) const;  
  casesout      zoned(9:2);  
end-pi;
```

```
// Perform some very complex code here
```

```
return;
```

```
end-proc;
```

```
// *****  
*****  
// This procedure converts cases to units quantity  
// *****  
*****
```

```
dcl-proc CasesToUnits export;
```

```
  dcl-pi *n;
    item          char(20)  const;
    casesin       zoned(9:2) const;
    unitsout      zoned(9:2);
  end-pi;
```

```
  dcl-s quantity  zoned(9:2);
```

```
  // Perform some very complex code here
```

```
    unitsout = IndustrySecretConversion(item:casesin);
```

```
  return;
```

```
end-proc;
```

```
  // *****
  *****
```

```
  // This procedure contains industry secret conversion routines
```

```
  // *****
  *****
```

```
dcl-proc IndustrySecretConversion;
```

```
  dcl-pi IndustrySecretConversion zoned(9:2);
    item          char(20)  const;
    quantityin    zoned(9:2) const;
  end-pi;
```

```
  dcl-s newqty    zoned(9:2);
```

```
  // Perform some very complex code here
```

```
  return newqty;
```

```
end-proc IndustrySecretConversion;
```