This is my thought of what it might look like. I am using a procedure to explain my case and in no way be limited to just a procedure.

Fully Collapsed

```
[+]     dcl-proc reInstate ;
```

Expanding The Procedure - Step by Step

Step 1

```
[-]     dcl-proc rInstate ;
|         LastEntryRei = %lookup (' ':dsReinstate(* .rname) ;
|         LoopRei = LastEntryRei - 1 ;
[+]       if LastEntryRei > 0 ;
└       end-proc rInstate ;
```

Step 2

```
[-]     dcl-proc rInstate ;
|         LastEntryRei = %lookup (' ':dsReinstate(*).rname) ;
|         LoopRei = LastEntryRei - 1 ;
[-]         if LastEntryRei > 0 ;
[+]           If RptType <> 'D' ;
├           Endif ;
└       end-proc rInstate ;
```

Step 3

```
[-]     dcl-proc rInstate ;
|         LastEntryRei = %lookup (' ':dsReinstate(*).rname) ;
|         LoopRei = LastEntryRei - 1 ;
[-]         if LastEntryRei > 0 ;
[-]           If RptType <> 'D' ;
|               Chain (MMonth:MDay:MYear) CtpMst02 ;
[+]               Dow %Found(CtpMst02) ;
├             Endif ;
├           Endif ;
└       end-proc rInstate ;
```

Step 4

```
[-]     dcl-proc rInstate ;
|         LastEntryRei = %lookup(' ':dsReinstate(*).rname) ;
|         LoopRei = LastEntryRei - 1 ;
[-]         if LastEntryRei > 0 ;
[-]           If RptType <> 'D' ;
|               Chain (MMonth:MDay:MYear) CtpMst02 ;
[-]               Dow %Found (CtpMst02) ;
```

```
47   |                     LX(1) = 0 ;
48   [+]                   If Month = CtpMo and Day = CtpDa and
     Year = CtpYr ;
49   |                   Read CtpMst02 ;
50   |                 Enddo ;
51   |               Endif ;
52   |             Endif ;
53   |         end-proc rInstate ;
54
55   Step 5
56   [-]       dcl-proc rInstate ;
57   |           LastEntryRei = %lookup(' ':dsReinstate(*).rname) ;
58   |           LoopRei = LastEntryRei - 1 ;
59   [-]           if LastEntryRei > 0 ;
60   [-]             If RptType <> 'D' ;
61   |               Chain (MMonth:MDay:MYear) CtpMst02 ;
62   [-]               Dow %Found(CtpMst02) ;
63   |                   LX(1) = 0 ;
64   [-]                   If Month = CtpMo and Day = CtpDa and
     Year = CtpYr ;
65   [+]                     If CtpSt1 = *Blank ;
66   [+]                     If CtpSt2 = *Blank ;
67   [+]                     If CtpSt3 = *Blank ;
68   [+]                     If CtpSt4 = *Blank ;
69   [+]                     If CtpSt5 = *Blank ;
70   [+]                     If CtpCCTot > 0 ;
71   |                       TstTot = LX(1) + CTPPresFee ;
72   |                       LX(12) = LX(1) + CtpCCTot +
     CTPPresFee ;
73   |                       TL(12) = TL(12) + LX(12) ;
74   |                       FL(12) = FL(12) + LX(12) ;
75   |                       CCfpg = CCfpg + CtpCCTot ;
76   |                       CCftot = CCftot + CtpCCTot ;
77   |                       CTPFee = (LX(1) *.025) ;
78   |                       CTPRev = (LX(1) - CTPFee) ;
79   |                       CTPFeepg = CTPFeepg + CTPFee ;
80   |                       CTPRevpg = CTPRevpg + CTPRev ;
81   |                       TOTCTPFee = TOTCTPFee + CTPFee ;
82   |                       TOTCTPRev = TOTCTPRev + CTPRev ;
83   |                       RPFPg = RPFPg + CTPPresFee ;
84   |                       RPFPg1 = RPFPg ;
85   |                       RPFTot = RPFTot + CTPPresFee ;
86   |                       RPFTot1 = RPFTot ;
87   |                       *In27 = *On ;
88   |                     Endif ;
89   |                 Enddo ;
90   |               Endif ;
```

```
91    ├           Endif ;
92    └       end-proc rInstate ;
93
94    Step  1
95    [-]     dcl-proc rInstate ;
96    │         LastEntryRei = %lookup(' ':dsReinstate(*).rname) ;
97    │         LoopRei = LastEntryRei - 1 ;
98    [-]         if LastEntryRei > 0 ;
99    [-]           If RptType <> 'D' ;
100   │               Chain (MMonth:MDay:MYear) CtpMst02 ;
101   [-]             Dow %Found(CtpMst02) ;
102   │                 LX(1) = 0 ;
103   [-]               If Month = CtpMo and Day = CtpDa and
      Year = CtpYr ;
104   [-]                 If CtpSt1 = *Blank ;
105   │                     LX(1) = LX(1) + CtpAmt1 ;
106   ├                   Endif ;
107   [-]                 If CtpSt2 = *Blank ;
108   │                     LX(1) = LX(1) + CtpAmt1 ;
109   ├                   Endif ;
110   [-]                 If CtpSt3 = *Blank ;
111   │                     LX(1) = LX(1) + CtpAmt1 ;
112   ├                   Endif ;
113   [-]                 If CtpSt4 = *Blank ;
114   │                     LX(1) = LX(1) + CtpAmt1 ;
115   ├                   Endif ;
116   [-]                 If CtpSt5 = *Blank ;
117   │                     LX(1) = LX(1) + CtpAmt1 ;
118   ├                   Endif ;
119   [-]                 If CtpCCTot > 0 ;
120   │                     *In34 = *On ;
121   │                   Else ;
122   │                     *In34 = *Off ;
123   ├                   Endif ;
124   │                   TstTot = LX(1) + CTPPresFee ;
125   │                   LX(12) = LX(1) + CtpCCTot +
      CTPPresFee ;
126   │                   TL(12) = TL(12) + LX(12) ;
127   │                   FL(12) = FL(12) + LX(12) ;
128   │                   CCfpg = CCfpg + CtpCCTot ;
129   │                   CCftot = CCftot + CtpCCTot ;
130   │                   CTPFee = (LX(1) *.025) ;
131   │                   CTPRev = (LX(1) - CTPFee) ;
132   │                   CTPFeepg = CTPFeepg + CTPFee ;
133   │                   CTPRevpg = CTPRevpg + CTPRev ;
134   │                   TOTCTPFee = TOTCTPFee + CTPFee ;
135   │                   TOTCTPRev = TOTCTPRev + CTPRev ;
```

```
136                             RPFPg = RPFPg + CTPPresFee ;
137                             RPFPg1 = RPFPg ;
138                             RPFTot = RPFTot + CTPPresFee ;
139                             RPFTot1 = RPFTot ;
140                             *In27 = *On ;
141                         Endif ;
142                     Enddo ;
143                 Endif ;
144             Endif ;
145         end-proc rInstate ;
146
```