IDKYCDT Session 4060
2019 Unite Conference
Adding Control Characters to WFL strings

Curt Timmerman
First National Bank Alaska
Anchorage, Alaska
ctimmerman@FNBAlaska.com

WFL Limitations

- There is currently no way to assign control characters (HEX values) to a WFL string variable
- Sometimes this could be handy for simple text formatting.
- The ClearPath EMAIL utility allows you to use '^' to start a new line.
  There is no insertion character for a TAB.
  Other EMAIL applications may not have this feature.

SETTASKSTRING Application

- Inspired by a Gregory's article (Nov 88) that added limited IO capabilities to WFL jobs.
- WFL Example:

```
STRING CRLF ;
TASK SETTASK ;
RUN SETTASKSTRING (" CR LF ") [SETTASK] ;
CRLF := SETTASK(TASKSTRING) ;
```

SETTASKSTRING parameter tokens are separated by 1 or more spaces.
The WFL CRLF variable now contains @0D25@ or the Windows equivalent of next line in a text document.

Control Character Substitutions

Parameter	Hex Value
HT	@05@
TAB	@05@
PAGE	@0C@
FF	@0C@
CR	@0D@
NL	@15@
LF	@25@
ESC	@27@
BEL	@2F@
42	@42@

The only binary value you can't use is NULL. TASKSTRING values are delimited by the NULL character.

Thinking Out Loud

- The application presented has a narrowly defined scope.
  Other uses for this interface:
- Retrieve data from DB or file
- Character translation or manipulation
- Regular expression for validation
- Message format parsing (XML/JSON)

Cobol Code

Programmer's Workbench source file will be available at: http://digm.com/ after the conference.

```
100000 IDENTIFICATION DIVISION.
101000 PROGRAM-ID. SETTASKSTRING.
102000*****************************************************************
102500*
103000* SETTASKSTRING - Sets task taskstring attribute to control
103100*                 characters specified in parameter input.
103120*
103140* Written by: Curt Timmerman (CTIMMERMAN@FNBALASKA.COM)
103200*
103300* Usage (WFL):
103400*   STRING CRLF ;
103500*   TASK SETTASK ;
103600*   RUN SETTASKSTRING (" CR LF ") [SETTASK] ;
103700*   CRLF := SETTASK(TASKSTRING) ;
103800*   %% CRLF now contains a carrage return + line feed
103900*
104000*****************************************************************
105000 ENVIRONMENT DIVISION.
106000*****************************************************************
107000 DATA DIVISION.
108000*****************************************************************
109000 WORKING-STORAGE SECTION.
110000 77  0-INPUT-PARAMETERS                  PIC X(80).
111000 01  2-WORK-AREA.
112000     05  2-CONTROL-SPECS                 PIC X(80).
113000     05  2-CONTROL-POINTER               PIC S9(11) BINARY.
114000     05  2-CONTROL-ID-PARAMETER          PIC X(4) VALUE SPACES.
115000         88  2-CONTROL-ID-END-88           VALUE SPACES.
116000     05  2-TASK-STRING.
117000         10  2-TASK-CHAR                 PIC X
118000              OCCURS 100 TIMES
119000              INDEXED BY 2-TASK-IDX.
120000     05  2-CONTROL-ID-TABLE.
120500*---- control/custom mnemonics
121000         10  FILLER                      PIC X(4) VALUE "HT  ".
122000         10  FILLER                      PIC X    VALUE @05@.
```

```
123000         10  FILLER                      PIC X(4) VALUE "TAB ".
124000         10  FILLER                      PIC X    VALUE @05@.
125000         10  FILLER                      PIC X(4) VALUE "PAGE".
126000         10  FILLER                      PIC X    VALUE @0C@.
127000         10  FILLER                      PIC X(4) VALUE "FF  ".
128000         10  FILLER                      PIC X    VALUE @0C@.
129000         10  FILLER                      PIC X(4) VALUE "CR  ".
130000         10  FILLER                      PIC X    VALUE @0D@.
131000         10  FILLER                      PIC X(4) VALUE "NL  ".
132000         10  FILLER                      PIC X    VALUE @15@.
133000         10  FILLER                      PIC X(4) VALUE "LF  ".
134000         10  FILLER                      PIC X    VALUE @25@.
135000         10  FILLER                      PIC X(4) VALUE "ESC ".
136000         10  FILLER                      PIC X    VALUE @27@.
137000         10  FILLER                      PIC X(4) VALUE "BEL ".
138000         10  FILLER                      PIC X    VALUE @2F@.
138500*---- HEX translation
139000         10  FILLER                      PIC X(4) VALUE "42  ".
140000         10  FILLER                      PIC X    VALUE @42@.
141000     05  2-CONTROL-ID-TABLE-ENTRIES
142000          REDEFINES 2-CONTROL-ID-TABLE.
143000         10  2-CONTROL-ID-TABLE-ENTRY
144000              OCCURS 10 TIMES
145000              INDEXED BY 2-CONTROL-INDEX.
146000             15  2-CONTROL-ID            PIC X(4).
147000             15  2-CONTROL-VALUE         PIC X.
148000/*****************************************************************
149000 PROCEDURE DIVISION
150000     USING 0-INPUT-PARAMETERS.
150100*
151000 0-MAIN.
152000*
153000     PERFORM 2-PROCESS
154000         THRU 2-PROCESS-EXIT.
155000*
156000     STOP RUN.
157000*
158000 0-MAIN-EXIT. EXIT.
159000*****************************************************************
160000* 2-PROCESS - Parse parameter tokens to build TASKSTRING
161000*****************************************************************
162000 2-PROCESS.
```

```
163000*----
163200*---- Bring in parameters and Upper Case the string
163400*----
164000     MOVE SPACES TO 2-CONTROL-SPECS.
165000     UNSTRING 0-INPUT-PARAMETERS
166000         DELIMITED BY LOW-VALUES
167000         INTO 2-CONTROL-SPECS.
168000     MOVE FUNCTION UPPER-CASE (2-CONTROL-SPECS)
169000         TO 2-CONTROL-SPECS.
170000*----
170200*---- Initialize pointers and fields
170400*----
171000     MOVE ALL @00@ TO 2-TASK-STRING.
172000     SET 2-TASK-IDX TO 1.
173000     MOVE 1 TO 2-CONTROL-POINTER.
174000*----
174200*---- Parse mnemonics, build output string
174400*----
175000     PERFORM 21-NEXT-CONTROL-TOKEN
176000         THRU 21-NEXT-CONTROL-TOKEN-EXIT.
177000     PERFORM UNTIL 2-CONTROL-ID-END-88
178000         SET 2-CONTROL-INDEX TO 1
179000         SEARCH 2-CONTROL-ID-TABLE-ENTRY
180000             VARYING 2-CONTROL-INDEX
181000             WHEN 2-CONTROL-ID (2-CONTROL-INDEX)
182000                     = 2-CONTROL-ID-PARAMETER
183000                MOVE 2-CONTROL-VALUE (2-CONTROL-INDEX)
184000                    TO 2-TASK-CHAR (2-TASK-IDX)
185000                SET 2-TASK-IDX UP BY 1
186000         END-SEARCH
187000         PERFORM 21-NEXT-CONTROL-TOKEN
188000             THRU 21-NEXT-CONTROL-TOKEN-EXIT
189000     END-PERFORM.
190000*----
190200*---- Return control character string to WFL job
190400*----
191000     CHANGE ATTRIBUTE TASKSTRING OF MYSELF
192000         TO 2-TASK-STRING.
193000*
194000 2-PROCESS-EXIT. EXIT.
195000*************************************************************
195200* 21-NEXT-CONTROL-TOKEN - Parse next parameter token
```

```
195400*****************************************************************
196000 21-NEXT-CONTROL-TOKEN.
197000*
198000     MOVE SPACES TO 2-CONTROL-ID-PARAMETER.
199000*
200000     UNSTRING 2-CONTROL-SPECS
201000         DELIMITED BY ALL SPACES
202000         INTO 2-CONTROL-ID-PARAMETER
203000         WITH POINTER 2-CONTROL-POINTER
204000     END-UNSTRING.
205000*
206000 21-NEXT-CONTROL-TOKEN-EXIT. EXIT.
```