Navigating INFO

presented by:
Tim Haithcoat
University of Missouri
Columbia

with materials from:
Environmental Systems Research Institute
Section Five

Programming
### Odd & Even Program Sections

<table>
<thead>
<tr>
<th>ODD SECTION</th>
<th>EVEN SECTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALC MAT_COST = MILES * $NUM1</td>
<td></td>
</tr>
<tr>
<td>CALC LBR_COST = MILES * $NUM2</td>
<td></td>
</tr>
<tr>
<td>CALC TOTAL_COST = MAT_COST + LBR_COST</td>
<td></td>
</tr>
</tbody>
</table>

Total bytes read =
3 * $NOSEL * Width

PROGRAM SECTION EVEN
CALC MAT_COST = MILES * $NUM1
CALC LBR_COST = MILES * $NUM2
CALC TOTAL_COST = MAT_COST + LBR_COST

PROGRAM SECTION ODD

Total bytes read =
$NOSEL * Width
Process first related record

“NEXT 1”

Process next related record

(no more matches)

Process next selected record
ODD & EVEN PROGRAM SECTIONS

NEXT PROCESSING

Process last related record
(no more matches)

Process next selected record

“First 1”

Process first related record
Odd & Even Program Sections

- **GOTO…LABEL**
  - Cannot branch to a label that is contained in another program section

- **DO…DOEND**
  - A DO loop may not cross program sections

- **IF…ENDIF**
  - An IF block cannot cross program sections
Odd & Even Program Sections

SUBPROGRAMS

MAIN_PROG
DO UNTIL $NUM1 = 10
  RUN SUB_PROG LINK
  CALC $NUM2 = $NUM1
  CALC $NUM1 = $NUM2 + 1
DOEND

SUB_PROG
PROGRAM SECTION EVEN
  -CALCULATIONS-
PROGRAM SECTION ODD
After **COMPILE**:

<table>
<thead>
<tr>
<th></th>
<th>1 - 20</th>
<th>21 - 27</th>
<th>38 - 60</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITEM 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITEM 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITEM 3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**COMPILE** records item columns in program **OBJ** file.

After **MODIFY**:

<table>
<thead>
<tr>
<th></th>
<th>1 - 24</th>
<th>25 - 41</th>
<th>42 - 64</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITEM 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITEM 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITEM 3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Item positions recorded by **COMPILE** no longer match item definitions.

After **COMPILE**:

<table>
<thead>
<tr>
<th></th>
<th>1 - 24</th>
<th>25 - 41</th>
<th>42 - 64</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITEM 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITEM 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITEM 3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**COMPILE** updates item positions to match new file definition.
Compiling and Decoding

- Tells INFO to compile code again
- Performs no immediate action
- Info editor automatically sets decode flag
- Always use decode when debugging
Debugging Techniques

- Test Files
- USE $NM
- Run with LIST
- Displaying Checkpoints
- Breaking from a Program (Prime Users Only)
Editing INFO Programs

- INFO CHANGE COMMAND
- SRC FILES
- COMI FILES
- AML PROGRAMS
The following AML is an example:

```aml
/* MK_CALC-IDS.AML
/*
/* Recalculate user Ids for a specified
/* polygon coverage
/*
&Args Cov
&If Not [Exists %Cov% - poly) &Then ~
&Return Coverage %Cov% not found or has no topology.

&SET Have_prog := {Exists CALC-IDS - INFO]

&Data Arc Info
&If %Have_prog% &Then &Do
Erase CALC-IDS
Y
&End
PROGRAM CALC-IDS
SELECT %Cov%.PAT
CALC %COV%-ID = %COV%# - 1
END
/* This line blank to end program.
RUN CALC-IDS
ERASE CALC-IDS
Y
QUIT,STOP
&End
&Return
```

Of course, we didn’t really need a program to do this calculation, but it illustrates the point.
Running Programs with LINK

- Returns to calling program
- Do not use END
- Automatically compiled with calling program
- Use as subroutines
- Use link to run menu options
AN EXAMPLE:

PROGRAM MENU.PG
FORMAT $NUM1,22,I
CALC $NUM1 = -1
SELECT SCHED RO
DO UNTIL $NUM1 GE 0 and $NUM1 LE $NOREC
    DIS =
    DIS '
    DIS '" Option Action'
    DIS '" 
    DIS '" ,$RECNO,' -',TEXT
    DIS '" Enter the number of your choice, or <CR> to quit: 
    ACCEPT $NUM1
    DOEND
    IF $NUM1 = 0
        QUIT/STOP
    ELSE
        RESELECT BY $RECNO = $NUM1
        EXECUTE RUN_COM
        RUN MENU.PG
    ENDIF
Menu Files and Programs

- Options kept in an INFO file
- Menu program uses EXECUTE
- Compact, portable and modifiable
- Variations
  - HELP program
  - Control of access